

# Data-driven control of nonlinear systems: an on-line direct approach

Marko Tanaskovic<sup>a,d</sup>, Lorenzo Fagiano<sup>b</sup>, Carlo Novara<sup>c</sup>, Manfred Morari<sup>a</sup>

<sup>a</sup>*Automatic Control Laboratory, Swiss Federal Institute of Technology, Zurich, Switzerland.*

<sup>b</sup>*ABB Switzerland Ltd., Corporate Research, Baden-Daettwil, Switzerland.*

<sup>c</sup>*Dipartimento di Automatica e Informatica, Politecnico di Torino, Italy.*

<sup>d</sup>*Maxon motor ag., Brünigstrasse 220, Sachseln, Switzerland*

---

## Abstract

A data-driven method to design reference tracking controllers for nonlinear systems is presented. The technique does not derive explicitly a model of the system, rather it delivers directly a **time-varying state-feedback** controller by combining an on-line and an off-line scheme. Like in other on-line algorithms, the measurements collected in closed-loop operation are exploited to modify the controller in order to improve the tracking performance over time. At the same time, a predictable closed-loop behavior is guaranteed by making use of a batch of available data, which is a characteristic of off-line algorithms. The feedback controller is parameterized with kernel functions and the design approach exploits results in set membership identification and learning by projections. **Under the assumptions of Lipschitz continuity and stabilizability of the system's dynamics, it is shown that if the initial batch of data is informative enough then the resulting closed-loop system is guaranteed to be finite gain stable.** In addition to the main theoretical properties of the approach, the design algorithm is demonstrated experimentally on a water tank system.

*Key words:* Data-driven control, Dynamic inversion, Nonlinear control, Identification for control, Adaptive control

---

## 1 Introduction

Model-based control design approaches require the derivation of a mathematical model of the plant to be controlled, the identification of the model parameters and the design of a controller based on the derived model. This approach is widely used and gives good results in many applications. However, in several cases, building a detailed and accurate model of a nonlinear plant can be difficult, costly and time-consuming. In these situations, data-driven design techniques represent a possible alternative approach, since they do not require a detailed knowledge of the physics of the system and rely only on the available measured data and relatively little prior information (e.g. qualitative information on the relations between the involved variables, approximate knowledge of the system order, knowledge on the system

block structure and on the most important internal states, etc.). In particular, in direct data-driven approaches the controller is designed directly from the measured data, eliminating completely the need for a model of the plant.

The existing direct data-driven approaches can be divided into on-line and off-line ones. In on-line schemes the controller is modified with each new measurement obtained in closed-loop operation. Examples of on-line direct techniques are the perturbation stochastic approximation control (see [20]), the model free adaptive control (see e.g. [6–8]) and the unfalsified control (see [4,16]). The main advantage of on-line techniques is the ability to improve the control performance over time using the measured data. However, since the controller can change at any time, its behavior is often hard to predict. In addition, guaranteeing stability of these control schemes is quite challenging and requires restrictive assumptions on the controlled system.

In off-line procedures the design is based on a batch of measurements, collected in preliminary experiments before the controller becomes operational, and no further modification is carried out during operation. Such techniques include the iterative feedback tuning (see e.g. [5,18]), the correlation based tuning (see [12]), the virtual reference feedback tun-

---

\* The material in this paper has been partly presented at 17th IFAC Symposium on System Identification (SYSID 2015) held in Beijing, Oct 19–21, 2015. Corresponding author M. Tanaskovic. ©2017 Elsevier 10.1016/j.automat.2016.09.032

*Email addresses:* tmarko@control.ee.ethz.ch (Marko Tanaskovic), lorenzo.fagiano@ch.abb.com (Lorenzo Fagiano), carlo.novara@polito.it (Carlo Novara), morari@control.ee.ethz.ch (Manfred Morari).

ing (see [1,3]) and the direct inversion based control (see e.g. [13] and the references therein). In most of these techniques, stability is not considered in the design phase and it is assessed by simulations or experimental verification before the controller becomes operational. Recently, an off-line direct technique that relies on nonlinear set-membership identification (see e.g. [10]) has been proposed in [14]. The approach guarantees theoretically finite-gain stability of the closed-loop system, as the number of data used for the design approaches infinity. The main disadvantage of off-line algorithms is that, unlike the on-line schemes, they do not exploit the additional measurements obtained during controller operation in order to improve the performance. On the other hand, the behavior of a controller designed off-line is usually more predictable.

In this paper, we propose a direct, data-driven design approach that combines the advantages of on-line and off-line techniques. The technique makes use of the theory of learning by projections (see [22] and the references therein) to update the controller on-line. At the same time, **under the assumptions of Lipschitz continuity and stabilizability of the system's dynamics, it is shown that if the initial batch of data is informative enough then the resulting closed-loop system is guaranteed to be finite gain stable.** In particular, stability is achieved by enforcing a robust constraint on the control input; such a constraint is derived by means of set-membership identification. The mentioned theoretical results are obtained by considering the control design problem as a static inversion, where one aims to derive, from experimental data, an approximate inverse of the system's function. This is the same theoretical framework as in [14], but the design approach and the results are completely different in order to enable on-line learning while retaining the stability guarantee. In addition, unlike the scheme in [14], the proposed on-line design algorithm leads to closed-loop stability even when the number of initially available data is finite. After describing the approach and its properties, we present the experimental results obtained on a laboratory water tank system, where we compare our technique with a purely off-line direct design approach and a well tuned linear controller.

The paper is organized as follows. The control problem is defined in Section 2 and the design algorithm is presented in Section 3. The theoretical analysis of the proposed scheme is described in Section 4.1, while Section 4.2 discusses the tuning of the involved parameters. The experimental results are presented in Section 5 and conclusions are drawn in Section 6.

## 2 Problem Statement

**We consider a discrete-time nonlinear system with one input and  $n_x$  states, represented by the following state equation:**

$$x_{t+1} = g(x_t, u_t) + e_{t+1}, \quad (1)$$

**where  $t \in \mathbb{Z}$  is the discrete time variable,  $u_t \in \mathbb{R}$  is the control input,  $x_t \in \mathbb{R}^{n_x}$  is the vector of measured states and  $e_t \in \mathbb{R}^{n_x}$  is the vector of disturbance signals that accounts for both the measurement noise and process disturbances.**

**Assumption 1** *The noise and disturbance term  $e_t$  is bounded in magnitude:*

$$e_t \in B_\epsilon \doteq \{e_t : \|e_t\| \leq \epsilon, \forall t \in \mathbb{Z}\}, \quad (2)$$

for some  $\epsilon > 0$ .

For a given compact domain  $Y$  and image set  $Z$ , let us denote the class of Lipschitz continuous functions over  $Y$ , with Lipschitz constant  $\gamma$ , with:

$$\mathcal{F}(\gamma, Y) \doteq \{f : Y \rightarrow Z : \|f(y^a) - f(y^b)\| \leq \gamma \|y^a - y^b\|, \forall y^a, y^b \in Y\}$$

**Remark 1** *Throughout the paper, the notation  $\|\cdot\|$  stands for a suitable vector norm chosen by the user (typically 2- or  $\infty$ -norm); the presented results hold for any specific norm.*

We further consider that during the controller operation the state trajectory shall be confined inside a compact set  $X \subset \mathbb{R}^{n_x}$ , and that input constraints are present in the form of a compact interval  $U \subset \mathbb{R}$ . The system at hand is assumed to enjoy the following regularity property over these sets:

**Assumption 2** *For any  $x \in X$ , the function  $g$  is Lipschitz continuous with respect to  $u$ , i.e.*

$$\forall x \in X, g(x, \cdot) \in \mathcal{F}(\gamma_g, U) \quad (3)$$

The function  $g$  in (1) is unknown to the control designer, but a set  $\mathcal{D}_N$  of past input and state measurements is available at time  $t = 0$ :

$$\mathcal{D}_N \doteq \{u_t, \omega_t\}_{t=-N}^{-1} \quad (4)$$

where

$$\omega_t \doteq (x_t, x_{t+1}). \quad (5)$$

**Assumption 3** *The batch of data  $\mathcal{D}_N$  is such that  $u_t \in U$  and  $\omega_t \in X \times X$ ,  $\forall t = -N, \dots, -1$ .*

**Remark 2** *If the system is open-loop unstable, a pre-stabilizing controller (eventually a human operator, as in [2]) can be used to carry out the initial experiments to collect the data  $\mathcal{D}_N$ . Such data are usually collected and commonly used also in model-based approaches, to identify the parameters of the mathematical model of the system. Another scenario to which our approach applies is when a high-fidelity model of  $g$  is available, but it is too complex to carry out model-based control design. In this case, the data  $\mathcal{D}_N$  can be also generated through simulations with such a model.*

**Remark 3** *The problem settings introduced so far can be extended to the case of output-feedback control design, if the system state is not known or not fully measured. In this case, one can replace the state in (1) with a regressor or pseudo-state, composed by present and past values of the input and of the output. Then, under reasonable controllability/observability conditions, the dynamics can still be written in the form (1), and algorithms and results similar to those presented in the following can be derived. Another option is to employ a state observer, when available, to obtain an estimate of the state, which would be affected by estimation errors that can be embedded in the term  $e_t$ . Similarly, when the system involves dynamics that are neglected, i.e. there are additional states with respect to those contained in  $x$ , our formulation and results would be still valid as long as one assumes to embed the effects of such neglected dynamics into the additive term  $e_t$ .*

In our theoretical derivations we consider the notion of finite gain stability (see e.g. [9]).

**Definition 1** (*Finite gain stability*) *A system with input  $u_t \in U$ , state  $x_t \in X$  and disturbance  $e_t \in B_\epsilon$  is finite gain stable if there exist finite and nonnegative constants  $\lambda_1$ ,  $\lambda_2$  and  $\beta$  such that:*

$$\|\mathbf{x}\|_\infty \leq \lambda_1 \|\mathbf{u}\|_\infty + \lambda_2 \|\mathbf{e}\|_\infty + \beta, \quad (6)$$

where  $\mathbf{x} = (x_1, x_2, \dots)$ ,  $\mathbf{u} = (u_1, u_2, \dots)$ ,  $\mathbf{e} = (e_1, e_2, \dots)$  are the time sequences of the input, state and disturbance variables, and  $\|\mathbf{x}\|_\infty \doteq \sup_t \|x_t\|$  is the  $\ell_\infty$  signal norm.

Let us consider a reference signal  $r_t$  to be tracked, assumed to belong to a compact set  $B_{\bar{r}} \subseteq X$ :

$$B_{\bar{r}} \doteq \{x \in \mathbb{R}^{n_x} : \|x\| \leq \bar{r}\}. \quad (7)$$

This means that the reference is bounded in norm by the scalar  $\bar{r} \in \mathbb{R}$ ,  $\bar{r} > 0$  and it is never outside the set where the state trajectory shall be confined. We can now introduce the notion of  $\gamma$ -stabilizability.

**Definition 2** ( $\gamma$ -stabilizability) *The system (1) is  $\gamma$ -stabilizable if there exists a  $\gamma < \infty$  and a control function  $f \in \mathcal{F}(\gamma, X \times X)$  such that the closed-loop system:*

$$x_{t+1} = g(x_t, f(x_t, r_{t+1})) + e_{t+1} \quad (8)$$

is finite gain stable, with input  $r_t \in B_{\bar{r}}$ , state  $x_t \in X$  and disturbance  $e_t \in B_\epsilon$ .

**Assumption 4** *The system (1) is  $\gamma$ -stabilizable for some  $\gamma < \infty$ .*

We can finally state the problem addressed in this paper.

**Problem 1** *Use the batch of data  $\mathcal{D}_N$ , collected up to  $t = 0$ , to design a feedback controller whose aim is to track a desired reference signal  $r_t \in B_{\bar{r}}$  for  $t > 0$ . Once the controller is in operation, carry out on-line refinements of the design by exploiting the incoming input and state measurements, while keeping the closed-loop system finite gain stable.*

### 3 On-line direct control design method

We approach Problem 1 from the point of view of data-driven, direct inversion techniques. In this context, we assume the existence of an “optimal” (in a sense that will be shortly specified) inverse of the system’s function  $g$  in (1) among the functions that, if used as controller, stabilize the closed-loop system. Then, we build from the available prior knowledge and data a set of functions that is guaranteed to contain the optimal inverse, and we exploit such a set to derive an approximated inverse, which we use as feedback controller. This approach involves several preliminary ingredients, explained in the following sub-sections.

#### 3.1 Optimal inverse and controller structure

Following the definitions and notation introduced in [14], for a given control function  $f$  we define the point-wise inversion error as:

$$IE(f, r, x, e) \doteq \|r - g(x, f(x, r)) - e\|, \quad (9)$$

and the global inversion error as:

$$GIE(f) =_L \|IE(f, \cdot, \cdot, \cdot)\|, \quad (10)$$

where  $_L \|\cdot\|$  in (10) is a suitable function norm (e.g.  $L_\infty$ ) evaluated on  $X \times B_{\bar{r}} \times B_\epsilon$ . Based on Assumption 4, there exist a set  $\mathcal{S}$  containing all functions  $f$  that stabilize the closed loop system. Then, we define the optimal inverse controller function  $f^*$  as:

$$f^* = \arg \min_{\mathcal{S} \cap \mathcal{F}_{X \times X}} GIE(f), \quad (11)$$

where  $\mathcal{F}_{X \times X}$  denotes the set of all Lipschitz continuous functions on  $X \times X$ . We denote the Lipschitz constant of  $f^*$  with  $\gamma^*$ , and the related constants  $\lambda_1$ ,  $\lambda_2$  and  $\beta$ , obtained if the controller  $f^*$  were used in closed-loop (see (6)), by  $\lambda_1^*$ ,  $\lambda_2^*$  and  $\beta^*$ .

**Deriving function  $f^*$  is challenging. In fact, even if the function  $g$  were known exactly (which is not the case in our settings, since we assume its existence but we don’t make explicit use of it in our algorithm), just to compute the global inversion error  $GIE$  (10) would entail solving a multi-variable integral of a general nonlinear function over a compact set. Then, to find the function**

$f^*$  (11) would be even more difficult, since the related optimization problem is generally non-convex and infinite-dimensional. To cope with this issue, we aim to calculate an approximation of  $f^*$  from the available data. In particular, we will derive a time-varying nonlinear function  $f_t \approx f^*$  which we will employ as feedback controller. Considering the measured data available up to a generic time  $t$ , we can write the control input as:

$$u_t = f^*(\omega_t) + d_t, \quad (12)$$

where  $d_t$  is a signal accounting for the unmeasured noise and disturbances and possible inversion errors. From Assumptions 1 and 2, it holds that as long as the state and input trajectories evolve in the sets  $X$  and  $U$ , respectively, the scalar  $d_t$  has to be bounded, i.e.  $d_t \in B_\delta \subset \mathbb{R}$  with  $\delta$  being a positive constant. Assumption 3 guarantees that this is the case up to  $t = 0$ , i.e. considering the initial batch of data  $\mathcal{D}_N$ , and let us assume that for  $t > 0$  the control input is such that this property is still valid (we will show that indeed our approach guarantees that  $x_t \in X, \forall t > 0$ ). Then, following a set membership identification approach (see e.g. [23,10]), we consider the set of feasible inverse functions at time step  $t$  ( $FIFS_t$ ), i.e. the set of all functions  $f \in \mathcal{F}_{X \times X}$  that are consistent with the available data and prior information:

$$FIFS_t \doteq \bigcap_{j=-N, \dots, t-1} H_j, \quad (13)$$

where:

$$H_j \doteq \{f \in \mathcal{F}_{X \times X} : |u_j - f(\omega_j)| \leq \delta\}. \quad (14)$$

The inequality in (14) stems from the observation that the difference between the measured input  $u_t$  and the value of function  $f^*$  evaluated at the corresponding  $\omega_t$  can not be larger than the bound on the amplitude of the signal  $d_t$ .

Under Assumptions 1–4, if  $u_t \in U$  and  $x_t \in X \times X, \forall t \geq -N$ , then the optimal inverse  $f^*$  belongs to  $FIFS_t$ , i.e.  $f^* \in FIFS_t$  for all  $t$ . In set membership identification, an estimate  $f \approx f^*$  belonging to the set  $FIFS_t$  enjoys a guaranteed worst-case approximation error not larger than twice the minimal one that can be achieved (see e.g. [23] for details). Motivated by this accuracy guarantee, we update the controller  $f_t$  on-line in order to approach the set  $FIFS_t$ . First, in order to have a tractable computational problem, we parameterize the controller  $f_t$  with a finite sum of kernel functions:

$$f_t(\omega) = a_t^T K(\omega, W_t),$$

where  $a_t \in \mathbb{R}^{L_t}$  is the vector of weights, and  $K(\omega, W_t) = [\kappa(\omega, \tilde{\omega}_1), \dots, \kappa(\omega, \tilde{\omega}_{L_t})]^T$  is a vector of kernel functions  $\kappa(\cdot, \tilde{\omega}_i) : \mathbb{R}^{2n_x} \rightarrow \mathbb{R}, i = 1, \dots, L_t$  belonging to a dictionary that is uniquely determined by the  $L_t$  kernel function centers  $W_t = \{\tilde{\omega}_1, \dots, \tilde{\omega}_{L_t}\}$ . Then, at each time step  $t$  we update the set  $W_t$ , which determines the kernel function dictionary, and we also recursively update the weights  $a_t$  exploiting the knowledge of  $FIFS_t$  (13), with an approach inspired by the projection-based learning scheme presented in

[19] in the context of signal processing. Moreover, in order to achieve finite gain stability of the closed-loop system, we exploit the information that  $f^* \in FIFS_0$  to derive a robust constraint on the vector of weights  $a_t$ , which we impose in the on-line procedure.

In the following, we provide the details of these steps and we then summarize the overall design method and discuss its computational and memory requirements.

### 3.2 Robust inequality to enforce closed-loop stability

In order to enforce the stability of the closed-loop system obtained when the time varying feedback controller  $f_t$  is used, we require the approximated inverse,  $f_t$ , to satisfy the following inequality at each time step  $t \geq 0$ :

$$\begin{aligned} |f_t(\omega_t^+) - f^*(\omega_t^+)| &\leq \gamma_\Delta \|x_t\| + \sigma, \\ \forall f^* \in \mathcal{F}(\gamma^*, X \times X) \cap FIFS_0, \forall t &\geq 0, \end{aligned} \quad (15)$$

where  $\omega_t^+ = [x_t, r_{t+1}]^T$  and  $\gamma_\Delta, \sigma \in \mathbb{R}, \gamma_\Delta, \sigma > 0$ , are design parameters. Sufficient conditions on these parameters that guarantee finite gain stability of the closed-loop are given in Section 4. The idea behind (15) is to limit the discrepancy between the input computed by the approximate inverse  $f_t$  at time step  $t$ , i.e.  $u_t = f_t(\omega_t^+)$ , and the one given by the optimal inverse  $f^*$  to a sufficiently small value, which depends linearly on the norm of the current state. However, since the optimal inverse  $f^*$  is not known, we require the inequality (15) to be satisfied robustly for all functions in  $FIFS_0$  that have Lipschitz constant equal to  $\gamma^*$ . As mentioned above, such a function set is in fact guaranteed to contain  $f^*$  under our working assumptions.

To translate the inequality (15) into a computationally tractable constraint on the parameters  $a_t$ , we exploit the information that  $f^* \in \mathcal{F}(\gamma^*, X \times X) \cap FIFS_0$  to compute tight upper and lower bounds on  $f^*(\omega_t^+)$  using the following result from [10].

**Theorem 1** (Theorem 2 in [10]) *Given a nonlinear function  $f^* \in \mathcal{F}(\gamma^*, X \times X) \cap FIFS_0$ , the following inequality holds:*

$$\underline{f}(\omega) \leq f^*(\omega) \leq \overline{f}(\omega),$$

where:

$$\begin{aligned} \overline{f}(\omega) &= \min_{k=-N, \dots, -1} (u_k + \delta + \gamma^* \|\omega - \omega_k\|) \\ \underline{f}(\omega) &= \max_{k=-N, \dots, -1} (u_k - \delta - \gamma^* \|\omega - \omega_k\|). \end{aligned} \quad (16)$$

Exploiting Theorem 1, the robust constraint (15) can be satisfied by enforcing the following two inequalities on the vector of weights  $a_t$ :

$$\begin{aligned} -\gamma_\Delta \|x_t\| - \sigma + \overline{f}(\omega_t^+) &\leq a_t^T K(\omega_t^+, W_t) \\ a_t^T K(\omega_t^+, W_t) &\leq \gamma_\Delta \|x_t\| + \sigma + \underline{f}(\omega_t^+) \end{aligned} \quad (17)$$

Note that the value of  $\gamma^*$  that is required in order to enforce the constraints in (17) can be estimated from the available data  $\mathcal{D}_N$  by using the methods presented in [14] and [21] (see also Remark 5). A detailed analysis on how the enforcement of (17) guarantees finite-gain stability of the closed-loop is provided in our main result in Section 4. We describe next the approach to update the dictionary of kernel functions and the vector of weights  $a_t$ .

### 3.3 Updating the dictionary of kernel functions

As stated in Section 3.1, the two main components that are used to calculate the time varying feedback controller  $f_t$  are the dictionary of kernel functions  $W_t$  and the vector of weights  $a_t$ . In this Section, we describe the algorithm to update the dictionary over time, such that it accounts for the incoming measurement data, while at the same time keeping its size bounded.

Kernel functions are widely used in machine learning for parametrization in nonlinear approximation and learning tasks (see e.g. [17]). Without loss of generality, we will consider the use of one of the most common of such functions, i.e. the Gauss one:

$$\kappa(\omega, \tilde{\omega}_i) = e^{-p\|\omega - \tilde{\omega}_i\|_2^2}, \quad (18)$$

where  $p \in \mathbb{R}, p > 0$  is a design parameter that determines the width of the Gauss function and  $\tilde{\omega}_i$  is its center.

The data generated by any Lipschitz continuous nonlinear function evaluated at a finite number of points can be well approximated by a dictionary of kernel functions centered at the same points. In our on-line controller design, we let the dictionary grow and incorporate new kernel functions as new input and state measurements are collected. However, adding a new function to the dictionary at each time step would lead to an unlimited growth of the dictionary size  $L_t$  over time. Moreover, this would result in a dictionary that is not sparse, i.e. with many functions that are similar (centered at points close to each other), and with possible over-fitting of the measurement data. To avoid these problems, we choose to add a new function only if it is sufficiently different from those already contained in the dictionary. As indicator of similarity, we use the so-called coherence factor (see e.g. [15] for more details):

$$\mu(\omega, W_t) = \max_{i=1, \dots, L_t} |\kappa(\omega, \tilde{\omega}_i)|. \quad (19)$$

Note that  $\mu(\omega, W_t) \in (0, 1]$ , and that  $\mu(\omega, W_t) = 1$  if and only if  $\omega \in W_t$ . Hence, the larger the coherence value in (19), the more similar is the kernel function centered at  $\omega$  to some function already in the dictionary. In our design technique, we set a threshold  $\bar{\mu} \in (0, 1)$  and we add a particular data point  $\omega$  to the set of function centers  $W_t$  if  $\mu(\omega, W_t) \leq \bar{\mu}$ . This approach guarantees that the size of the dictionary will remain bounded over time, as we recall

in Section 4 (see Lemma 1). The tuning parameter  $\bar{\mu}$  determines the size and sparsity of the dictionary. A discussion on how to choose the parameters  $p$  in (18) and  $\bar{\mu}$  is given in Section 4.2.

### 3.4 Updating the vector of weights

**The vector of weights  $a_t$  is updated over time such that the corresponding function  $f_t$ , evaluated at a finite number of measured regressor values  $\omega_j = (x_{j+1}, x_j)$ ,  $j = t-q, \dots, t-1$ , is consistent with the assumed prior information, in particular with the fact that  $|u_j - f_t(\omega_j)| \leq \delta$ , see (14). This update strategy yields, on the one hand, a computationally tractable algorithm to adapt the weights (since enforcing the inequality (14) for all time instants would lead to a problem whose size grows unlimited with time), and, on the other hand, guaranteed theoretical properties regarding the distance of the resulting function from the set  $FIFS_t$  (13). At the same time, we also enforce the robust inequalities required for closed-loop stability as described in Section 3.2.**

As a preliminary step to the recursive update of the weights  $a_t$  note that, as discussed above, the size of the dictionary can expand from time step  $t-1$  to time step  $t$  and therefore in general it will hold that  $a_{t-1} \in \mathbb{R}^{L_{t-1}}$  and  $a_t \in \mathbb{R}^{L_t}$  with  $L_{t-1} \leq L_t$ . Therefore, in order to properly define the updating algorithm at time  $t$ , we consider the vector  $a_{t-1}^+ \in \mathbb{R}^{L_t}$ :

$$a_{t-1}^+ = [a_{t-1}^T, \underbrace{0, \dots, 0}_{L_t - L_{t-1}}]^T, \quad (20)$$

obtained by initializing the weights corresponding to the kernel functions that are added to the dictionary to zero.

To introduce the updating of the vector  $a_t \in \mathbb{R}^{L_t}$ , we note that each pair  $(u_j, \omega_j)$ ,  $j = -N, \dots, t-1$  defines, together with the dictionary of kernel functions at time step  $t$ , the following set:

$$S_{jt} \doteq \{a \in \mathbb{R}^{L_t} : |a^T K(\omega_j, W_t) - u_j| \leq \delta\}, \quad (21)$$

which is a strip (hyperslab) in  $\mathbb{R}^{L_t}$ . If  $a_t \in S_{jt}$ , then the corresponding function  $f_t$  in (15) belongs to the set  $H_j$  defined in (14). We further define the projection of a point in  $\mathbb{R}^{L_t}$  onto the strip  $S_{jt}$  as:

$$P_{jt}(a) \doteq \min_{\hat{a} \in S_{jt}} \|a - \hat{a}\|_2. \quad (22)$$

Note that calculating the projection (22) amounts to solving a very simple quadratic program, whose solution can be explicitly derived (see e.g. [22]):

$$P_{j_t}(a) = a + \begin{cases} \frac{u_j - \delta - a^T K(\omega_j, W_t)}{\|K(\omega_j, W_t)\|_2^2} K(\omega_j, W_t) & \text{if } u_j - \delta > a^T K(\omega_j, W_t) \\ 0 & \text{if } |a^T K(\omega_j, W_t) - u_j| \leq \delta \\ \frac{u_j + \delta - a^T K(\omega_j, W_t)}{\|K(\omega_j, W_t)\|_2^2} K(\omega_j, W_t) & \text{if } u_j + \delta < a^T K(\omega_j, W_t). \end{cases} \quad (23)$$

Therefore, calculating the projection of any point in  $\mathbb{R}^{L_t}$  onto a measurement strip as in (22) can be done computationally very efficiently. Finally, we consider the hyperslab defined by the stability constraint (17):

$$S_t^+ \doteq \left\{ a \in \mathbb{R}^{L_t} : \begin{aligned} & a^T K(\omega_t^+, W_t) \geq -\gamma_\Delta \|x_t\| - \sigma + \bar{f}(\omega_t^+) \\ & a^T K(\omega_t^+, W_t) \leq \gamma_\Delta \|x_t\| + \sigma + \underline{f}(\omega_t^+) \end{aligned} \right\}, \quad (24)$$

and we denote the corresponding projection operator with  $P_t^+(\cdot)$ . Such a projection can also be calculated explicitly by using a formula similar to (23).

From the definitions of the hyperslabs  $S_{j_t}$  and  $S_t^+$  in (21) and (24) it follows that if

$$a_t \in S_t^+ \cap \left( \bigcap_{j=-N, \dots, t-1} S_{j_t} \right),$$

then the corresponding function  $f_t$  belongs to the set  $FIFS_t$  and satisfies the stabilizing constraint (15). However, to find a point that belongs to the intersection of all  $S_{j_t}$ ,  $j = -N, \dots, t-1$  at each time step is computationally challenging. Therefore, we exploit the idea at the basis of projection learning algorithms, that by repeatedly applying the projection operators to a point, the result will eventually fall in the intersection of the considered hyperslabs. In particular, we update the vector of weights  $a_t$  in two steps: first, following the idea of [19], we calculate a convex combination of its projections onto the hyperslabs defined by a finite number  $q \geq 1$  of the latest measurements; then, we project the obtained point onto the hyperslab  $S_t^+$  in order to ensure the satisfaction of the stabilizing constraint (15). To be more specific, let the set of indexes  $J_t = \{\max\{-N, t-q\}, \dots, t-1\}$  contain the time instants of the last  $q$  state and input measurements, and let  $I_t = \{j \in J_t : a_{t-1}^+ \notin S_{j_t}\}$  be the subset of indexes such that the weighting vector  $a_{t-1}^+$  does not belong to the corresponding hyperslabs. Then, we compute our update of the weighting vector  $a_t$  from  $a_{t-1}^+$  as:

$$a_t = P_t^+ \left( a_{t-1}^+ + \sum_{j \in I_t} \frac{1}{\text{card}(I_t)} (P_{j_t}(a_{t-1}^+) - a_{t-1}^+) \right), \quad (25)$$

where  $\text{card}(I_t)$  denotes the number of elements in  $I_t$ . This update can be computed very efficiently with the explicit formulas for vector projections (see e.g. (23)) and eventually by parallelizing the projection operations.

### 3.5 Summary of the proposed design algorithm

The described procedures to update the dictionary of kernel functions and the weights  $a_t$  form our on-line scheme to compute the feedback controller  $f_t$ , summarized in Algorithm 1.

---

**Algorithm 1** Feedback control algorithm based on the on-line direct control design scheme

---

- 1) Collect the state measurement  $x_t$ . If  $t < 0$ , set  $\omega_t^+ = [x_t, x_{t+1}]^T$ , otherwise set  $\omega_t^+ = [x_t, r_{t+1}]^T$ .
  - 2) Update the dictionary  $W_t$  starting from  $W_{t-1}$  by adding  $\omega_{t-1}$  if  $\mu(\omega_{t-1}, W_{t-1}) \leq \bar{\mu}$ . Form the vector  $a_{t-1}^+$  according to (20).
  - 3) Calculate  $a_t$  according to (25).
  - 4) If  $t \geq 0$ , calculate the input  $u_t = a_t^T K(\omega_t^+, W_t)$  and apply it to the plant.
  - 5) Set  $t = t + 1$  and go to 1).
- 

For  $t \geq 0$ , such an algorithm is both a controller and a design algorithm, while for  $t < 0$  it only acts as a design algorithm.

**Remark 4** Note that instead of running Algorithm 1 for  $t < 0$ , other off-line techniques for controller design (like the one in [14]) could be used in order to derive the initial controller. Algorithm 1 could then be used on-line (for  $t \geq 0$ ) in order to further refine the initial controller during the closed-loop operation.

The proposed scheme has moderate computational requirements, since many operations can be parallelized, and it does not require the solution of complex mathematical problems. However, in order to evaluate the bounds (16) needed to compute the projection operator  $P_t^+(\cdot)$  onto the hyperslab  $S_t^+$  in (24), the training data  $\mathcal{D}_N$  (or a subset of it) need to be stored in memory and made available on-line.

## 4 Properties of the approach and tuning guidelines

### 4.1 Theoretical properties of the proposed algorithm

We will now present results on the following aspects: closed-loop stability, growth of the dictionary of kernel functions over time, and performance in terms of boundedness of the tracking error. The first result provides sufficient conditions for a time-varying controller  $f_t$ , designed according to our Algorithm 1, to render the closed-loop system finite-gain stable. Let us consider the following assumption on the tuning parameter  $\gamma_\Delta$ .

**Assumption 5**

$$\gamma_\Delta \in \left( 0, \frac{1}{\gamma_g \lambda_2^*} \right).$$

Assumption 5 requires the tuning parameter  $\gamma_\Delta$  to be small enough. With this assumption, we can define the maximal achievable state amplitude as:

$$\bar{x} \doteq \frac{\lambda_1^* \bar{r} + \gamma_g \lambda_2^* \sigma + \lambda_2^* \epsilon + \beta^*}{1 - \gamma_g \lambda_2^* \gamma_\Delta}, \quad (26)$$

and the sets  $B_{\bar{x}}$  and  $B_{\bar{x}^r}$  as:

$$\begin{aligned} B_{\bar{x}} &\doteq \{x \in \mathbb{R}^{n_x} : \|x\| \leq \bar{x}\}, \\ B_{\bar{x}^r} &\doteq \{\omega = (x, r) : x \in B_{\bar{x}}, r \in B_{\bar{r}}\}. \end{aligned} \quad (27)$$

We consider also a technical assumption on the relationship between the sets  $B_{\bar{x}}$ ,  $B_{\bar{x}^r}$  and the sets  $U$  and  $X$ .

**Assumption 6**  $B_{\bar{x}} \subseteq X$ . Moreover,  $\forall \omega \in B_{\bar{x}^r}, \forall \Delta u \in [-\gamma_\Delta \bar{x} - \sigma, \gamma_\Delta \bar{x} + \sigma], f^*(\omega) + \Delta u \in U$ .

Namely, Assumption 6 requires the compact sets  $X$  and  $U$ , in which the state and the input of the plant (1) evolve, to be sufficiently large, such that they include, respectively, the set  $B_{\bar{x}}$  and all possible control inputs that are consistent with the robust constraint (15), for all pairs of state measurement and reference values in  $B_{\bar{x}^r}$ . In particular, this means that the function  $g$  should be Lipschitz continuous with respect to the input  $u$  over sets  $X$  and  $U$  that are sufficiently large.

Moreover, we denote with  $D_0$  the diameter of information derived from the set membership condition  $f^* \in \mathcal{F}(\gamma^*, X \times X) \cap FIFS_0$ .  $D_0$  is the maximal difference, evaluated over the set  $B_{\bar{x}^r}$ , between the bounds in (16):

$$D_0 \doteq \sup_{\omega \in B_{\bar{x}^r}} (\bar{f}(\omega) - \underline{f}(\omega)). \quad (28)$$

We assume that  $D_0$  and the tuning parameter  $\sigma$  satisfy the following condition.

**Assumption 7**

$$\sigma \geq \frac{D_0}{2}.$$

**Remark 5** Assumptions 5, 6 and 7 depend on both system-related quantities, like  $\epsilon, \delta, \gamma_g, \gamma^*, D_0, \lambda_1^*, \lambda_2^*$  and  $\beta^*$ , and design parameters, i.e.  $\sigma$  and  $\gamma_\Delta$ , that have been introduced so far. We note that, in order to apply our method, most of the involved system-related quantities need not to be known explicitly, rather they are instrumental to derive theoretical guarantees on the closed-loop system and on the behavior of the approach. The only quantities that need to be estimated are the bound  $\delta$  and the Lipschitz constant  $\gamma^*$ . As a matter of fact, such estimates can be obtained from the collected data, see e.g. [14], with guaranteed convergence under assumptions on the informative content of the collected data. As an additional extension of the approach, we also explored the possibility to compute and refine such estimates on-line, with guaranteed convergence properties. We omit here such extensions for the sake of clarity and brevity; for the full details the interested reader is referred to [21].

We now have all the ingredients to state the result on the finite gain stability of the closed-loop.

**Theorem 2** Let the Assumptions 1–7 hold, and let  $S_0^+ \neq \emptyset$  and  $x_0 \in B_{\bar{x}}$ . Then, for any reference signal  $r_t \in B_{\bar{r}}, \forall t > 0$ , Algorithm 1 guarantees that  $S_t^+ \neq \emptyset, \forall t \geq 0$  and that the closed-loop system is finite gain stable.

**Proof.** We prove the theorem by induction. First, we note that the closed loop system obtained by using the approximate controller  $f_t$  can be represented as:

$$\begin{aligned} x_{t+1} &= g(x_t, f_t(x_t, r_{t+1})) + e_{t+1} = g(x_t, f^*(x_t, r_{t+1})) + e_{t+1} + v_t, \\ v_t &= g(x_t, f_t(x_t, r_{t+1})) - g(x_t, f^*(x_t, r_{t+1})). \end{aligned} \quad (29)$$

From Assumption 4 and the definition of the optimal inverse controller  $f^*$  in (11), it holds that:

$$\|x\|_\infty \leq \lambda_1^* \|r\|_\infty + \lambda_2^* \|v\|_\infty + \lambda_2^* \|e\|_\infty + \beta^*, \quad (30)$$

where  $v = (v_1, v_2, \dots)$ . Moreover, we note that from Assumptions 2 and 6, it follows that:

$$\|v_t\| \leq \gamma_g |f_t(x_t, r_{t+1}) - f^*(x_t, r_{t+1})|, \forall (x_t, r_{t+1}) \in B_{\bar{x}^r}. \quad (31)$$

We now employ the inductive argument to show that if  $S_0^+ \neq \emptyset$  and  $x_0 \in B_{\bar{x}}$ , then  $S_t^+ \neq \emptyset$  and  $x_t \in B_{\bar{x}}, \forall t \geq 0$ . The condition is satisfied for  $t = 0$  by the Theorem assumption. Let us assume, for the sake of inductive argument, that  $S_k^+ \neq \emptyset$  and  $x_k \in B_{\bar{x}}, \forall k \in [0, t-1]$ . From this assumption and the way the weighting vector  $a_t$  is updated in (25), it follows that  $a_k \in S_k^+, \forall k \in [0, t-1]$ . From Assumptions 3 and 6, it follows that  $\omega_k \in X \times X$  and  $u_k \in U, \forall k \in [-N, t-1]$ . From the definition of  $S_t^+$  in (24), Assumptions 1–2 and Theorem 1, it then follows that:

$$|f_k(x_k, r_{k+1}) - f^*(x_k, r_{k+1})| \leq \gamma_\Delta \|x_k\| + \sigma, \forall k \in [0, t-1]. \quad (32)$$

From (31) and (32) it then holds that:

$$\|v_k\| \leq \gamma_g \gamma_\Delta \|x_k\| + \gamma_g \sigma, \forall k \in [0, t-1]. \quad (33)$$

From Assumption 5, it follows that  $\gamma_g \lambda_2^* \gamma_\Delta < 1$ . From this and from (30) and (33) it further follows that:

$$\begin{aligned} \|x_t\|_\infty &\leq \frac{\lambda_1^*}{1 - \gamma_g \lambda_2^* \gamma_\Delta} \|r_t\|_\infty + \frac{\lambda_2^*}{1 - \gamma_g \lambda_2^* \gamma_\Delta} \|e_t\|_\infty \\ &\quad + \frac{\gamma_g \lambda_2^* \sigma + \beta^*}{1 - \gamma_g \lambda_2^* \gamma_\Delta}, \end{aligned} \quad (34)$$

where  $x_t = (x_1, \dots, x_t)$ ,  $r_t = (r_1, \dots, r_t)$  and  $e_t = (e_1, \dots, e_t)$ . From the facts that  $\|r_t\|_\infty \leq \bar{r}$  and that  $\|e_t\|_\infty \leq \epsilon$ , and from (26), it then follows that  $\|x_t\| \leq \bar{x}$ , i.e.  $x_t \in B_{\bar{x}}$ . Therefore,  $\omega_t^+ \in B_{\bar{x}^r}$  and from (28), it holds that  $\bar{f}(\omega_t^+) - \underline{f}(\omega_t^+) \leq D_0$ . From Assumption 7, it then follows that:

$$-\gamma_\Delta \|x_t\| - \sigma + \bar{f}(\omega_t^+) \leq \gamma_\Delta \|x_t\| + \sigma + \underline{f}(\omega_t^+),$$

which implies that  $S_t^+ \neq \emptyset$ . Repeating this inductive argumentation for all  $t \geq 0$ , it follows that  $S_t^+ \neq \emptyset, \forall t \geq 0$ . In addition, (34) will hold for all  $t \geq 0$  which implies that the closed-loop system is finite gain stable (see e.g. Definition 1).  $\blacksquare$

**Remark 6** *It is hard to use Theorem 2 in a constructive way in order to precisely select the tuning parameters and the initial batch of data that guarantee closed-loop stability, since, as mentioned above, many of the quantities involved in Assumptions 1–7 are in general not known. Nevertheless, the result provides useful insights on the behavior of the approach, as well as tuning guidelines for the parameters  $\sigma$  and  $\gamma_\Delta$ . In particular, from Assumptions 5 and 7 it follows that, for the sake of closed-loop stability,  $\gamma_\Delta$  should be chosen as a small value, and  $\sigma$  as a large one. A larger value of  $\sigma$  implies that larger discrepancies between the approximated inverse and the optimal one can occur (see (15)), with consequent potentially larger values of the state variable in closed-loop, represented by  $\bar{x}$  in (26). However, the latter quantity shall be kept small, as required by Assumption 6, for the state trajectory to remain inside the set  $X$  and for the input to remain within the set  $U$ . Hence, the value of  $\sigma$  should be chosen as the smallest such that Assumption 7 holds. The latter condition can be indeed always satisfied for any fixed value of  $\sigma$  provided that the initial batch of data  $\mathcal{D}_N$  is large and informative enough, since in this case the diameter of information  $D_0$  can be made arbitrarily small, see e.g. [10]. Thus, ultimately Theorem 2 can be also interpreted as a sufficient condition on the initial data set, such that the designed controller guarantees closed-loop stability.*

A direct consequence of Theorem 2 and of its proof is that the tracking error is bounded  $\forall t \geq 0$ . Namely, from (9), (15), (29) and (31) it follows that  $\forall t \geq 0$ :

$$\|r_t - x_t\| \leq IE(f^*, r_t, x_{t-1}, e_t) + \gamma_g \gamma_\Delta \|x_{t-1}\| + \gamma_g \sigma. \quad (35)$$

Therefore, despite the fact that the controller  $f_t$  can change at each time step during operation, an upper bound on the tracking error can be theoretically derived.

Next, we address the question on whether the size of the dictionary remains bounded or grows unlimited over time, which would imply prohibitive computations with the proposed algorithm. To answer this question, we recall an important property of the coherence measure, demonstrated in [15].

**Lemma 1** (Proposition 2 in [15]) *Let  $\mathcal{W}$  be a compact set. Then for any  $\bar{\mu} \in [0, 1)$ , the dictionary obtained by adding a kernel function centered at  $\omega_t \in \mathcal{W}$  to it when  $\mu(\omega_t, W_t) \leq \bar{\mu}$  has a finite number of elements for any sequence  $\{\omega_t\}_{t=-N}^\infty$ .*

As shown by Theorem 2, under our working assumptions it holds that  $\omega_t \in B_{\bar{x}r}, \forall t \geq 0$  and from Assumption 3 it

follows that  $\omega_t \in X \times X$  for  $t \in [-N, -1]$ . Therefore, all incoming data points that are used to evaluate the coherence factor  $\mu$  in (19) belong to a compact set and we can apply Lemma 1 to conclude that the size of the dictionary used under the Algorithm 1 remains bounded.

More precisely, there exists a finite time step  $\underline{t} < \infty$  such that the dimension  $L_t$  of vector  $a_t$  reaches an upper bound  $L_t = \bar{L}, \forall t \geq \underline{t}$  and therefore  $S_{jt} \subset \mathbb{R}^{\bar{L}}, \forall j \in I_t$  and  $S_t^+ \subset \mathbb{R}^{\bar{L}}$  for  $\forall t \geq \underline{t}$ . Moreover,  $\forall t^a, t^b, \geq \underline{t}$ , it holds that  $S_{jt^a} = S_{jt^b}$ , therefore for  $t \geq \underline{t}$  we can denote  $S_{jt}$  just by  $S_j$ , **since the centers of the kernel functions do not depend on time anymore.**

In addition, we state the following result, which is a direct consequence of the properties of the projection based update rule (25) analyzed in [19].

**Lemma 2** (Slight modification of Theorem 4.2 in [19]) *Let*

$\Omega \doteq \bigcap_{t \geq \bar{t}} \left( S_t^+ \cap \left( \bigcap_{j \in I_t} S_j \right) \right) \neq \emptyset$  for some finite  $\bar{t} \in \mathbb{N}, \bar{t} \leq \underline{t} < \infty$ . Then, the recursive update rule (25) of Algorithm 1 is guaranteed to bring the point  $a_t \in \mathbb{R}^{\bar{L}}$  closer to the set  $\Omega$  with each time step  $t \geq \bar{t}$ , i.e.  $\min_{a \in \Omega} \|a_t - a\|_2 \leq \min_{a \in \Omega} \|a_{t-1} - a\|_2, \forall t \geq \bar{t}$  and in the limit it holds that:

$$\lim_{t \rightarrow \infty} \min_{a \in \Omega} \|a_t - a\|_2 = 0. \quad (36)$$

Lemma 2 states that if the set  $\Omega$  is nonempty, then, in the limit, the vector of weights  $a_t$  is guaranteed to belong to the intersection of all but finitely many hyperslabs  $S_t^+$  and  $S_j, j \in I_t, t \geq \underline{t}$ . Therefore, from the definitions of  $S_{jt}$  in (21) and the control function  $f_t$  in (15), it follows that in the limit, the control law  $f_t$  belongs to the intersection of the sets  $H_k, k \geq \bar{t}$  (see e.g. (14)). Note also that for Lemma 2 to hold, the set  $\Omega$  needs to be nonempty, which is a reasonable assumption if the values of the disturbance bound  $\delta$  and the Lipschitz constant  $\gamma^*$  are correctly estimated or overestimated (see also Remark 5) and if the parameters related to the parametrization of the controller are chosen properly. Guidelines on how to select the tuning parameters related to controller parametrization are given in Section 4.2.

Finally, we present a result pertaining to the behavior of the tracking error. On the basis of the properties of Algorithm 1 described so far, we can show that in the limit, i.e. as  $t \rightarrow \infty$ , there is a bound on the tracking error that might be tighter than the one provided by (35). To introduce this result, for any point  $\omega \in B_{\bar{x}r}$ , we denote by  $v_\omega$  the smallest positive constant for which there exists a time step  $t, \bar{t} \leq t < \infty$  such that

$$\|\omega_t - \omega\| \leq v_\omega,$$

where  $\omega_t$  is the state measurement pair acquired at time step  $t$ . In addition, we denote by  $\gamma_{f_t}$  the Lipschitz constant of



the approximate controller  $f_t$ . Note that the function  $f_t$  is indeed Lipschitz continuous as it is a sum of Gauss kernel functions (see (15)). In addition, we note that if the set  $\Omega$  is bounded, then according to Lemma 2 it follows that as  $t \rightarrow \infty$ ,  $f_t$  is given as a weighted sum of Lipschitz functions with bounded weights and therefore there exists a constant  $\bar{\gamma}_f \in \mathbb{R}$ ,  $\bar{\gamma}_f > 0$  such that  $\lim_{t \rightarrow \infty} \sup \gamma_{f_t} \leq \bar{\gamma}_f$ . The condition that  $\Omega$  is bounded is satisfied if the acquired measurements are informative, i.e. if the hyperslabs generated by the incoming measurements form a bounded set. Based on the introduced definitions, we state the following result.

**Theorem 3** *If  $\Omega$  is nonempty and bounded, then:*

$$\lim_{t \rightarrow \infty} \|r_t - x_t\| \leq \lim_{t \rightarrow \infty} \text{here???} IE(f^*, r_t, x_{t-1}, e_{t-1}) + 2\gamma_g \delta + \gamma_g (\bar{\gamma}_f + \gamma^*) v_\omega, \forall (x_{t-1}, r_t) \in B_{\bar{x}}, \quad (37)$$

where  $\omega = (x_{t-1}, r_t)$ .

**Proof 1** *For any  $\omega \in B_{\bar{x}}$  it holds that*

$$|f_t(\omega) - f^*(\omega)| \leq |f_t(\omega_k) - f^*(\omega_k)| + |f_t(\omega_k) - f_t(\omega)| + |f^*(\omega_k) - f^*(\omega)|, \quad (38)$$

where  $\omega_k$  is the state measurement pair acquired at time step  $k$ ,  $\bar{t} \leq k < \infty$  that satisfies

$$\|\omega_k - \omega\| \leq v_\omega.$$

In addition, from the definition of  $\Omega$  it holds that  $a \in S_k$ , for any  $a \in \Omega$  and any  $k \geq \bar{t}$ , where  $S_k$  is the hyperslab defined by the measurement point  $\omega_k$ . Moreover, from Lemma 1 it follows that  $W_t = W_{\bar{t}}, \forall t \geq \bar{t}$ . From this and the fact that  $|f^*(\omega_k) - u_k| \leq \delta$  (see e.g. (12)), it follows that

$$|a^T K(\omega_k, W_t) - f^*(\omega_k)| \leq 2\delta, \forall a \in \Omega, \forall k \geq \bar{t}. \quad (39)$$

From Lemma 2 it follows that there exists a point  $a \in \Omega$  such that  $\lim_{t \rightarrow \infty} a_t = a$  and hence from the definition of the controller  $f_t$  and (39) it follows that:

$$\lim_{t \rightarrow \infty} |f_t(\omega_k) - f^*(\omega_k)| \leq 2\delta, \forall k \geq \bar{t}. \quad (40)$$

In addition, from the fact that both  $f_t$  and  $f^*$  are Lipschitz continuous, and from the definitions of  $v_\omega$  and  $\bar{\gamma}_f$  it follows that

$$\begin{aligned} \lim_{t \rightarrow \infty} |f_t(\omega_k) - f_t(\omega)| &\leq \bar{\gamma}_f v_\omega \\ \lim_{t \rightarrow \infty} |f^*(\omega_k) - f^*(\omega)| &\leq \gamma^* v_\omega. \end{aligned} \quad (41)$$

Moreover, as shown in the proof of Theorem 2, it holds that

$$\|r_t - x_t\| \leq IE(f^*, r_t, x_{t-1}, e_{t-1}) + \gamma_g |f_t(x_{t-1}, r_t) - f^*(x_{t-1}, r_t)|. \quad (42)$$

Therefore, from (38)–(42) the claim of the theorem follows directly. ■

As an interpretation of Theorem 3, consider the case when the system operates close to the reference. As more and more data points are collected over time, the value of the constant  $v_\omega$  for the points close to the reference trajectory is reduced and it eventually tends to zero as the state space around the desired reference trajectory is well explored. In such a case, the tracking error in the limit is bounded by the inversion error of the optimal inverse function  $f^*$ , which can be assumed to be small, plus the product  $2\gamma_g \delta$ , which can be also assumed to be small in practice. Thus, if the neighborhood of the reference trajectory is well explored during operation, then good tracking performance can be expected as more and more measurement data are collected. This is typically the case when the reference trajectory is periodic or constant for some time, as we show in our experimental results in Section 5.

## 4.2 Tuning guidelines for the learning algorithm

In order to implement Algorithm 1, few tuning parameters need to be selected. These parameters can be split into two groups. The first group consists of the scalars  $\gamma_\Delta$  and  $\sigma$ , related to closed-loop stability and to the amount of information contained in the initial data, which we already commented on in the previous Section (see Remark 6). Calculating these parameters such that the assumptions required for stability would be satisfied might be challenging in general and is a topic of current research. Some initial considerations on how to select these parameters, on the basis of the available measurement data, such that closed-loop stability is guaranteed are given in [21].

The second group of parameters also consists of two scalars, the Gauss kernel function parameter  $p$  (see (18)) and the value of the coherence bound  $\bar{\mu}$  (see (19)), which are related to the parametrization of the controller  $f_t$ . We briefly elaborate here about the choice of such parameters. A possible way to select the right combination of  $p$  and  $\bar{\mu}$  is to run the Algorithm 1 off-line, by using the available training data  $\mathcal{D}_N$ , for a set of different parameter values and, after each execution, to calculate the average estimation error with respect to the data  $\mathcal{D}_N$  as:

$$E_e = \frac{1}{N} \sum_{k=-N}^{-1} |a_{-1}^T K(\omega_k, W_{-1}) - u_k|. \quad (43)$$

There are several outcomes of these trials, which can be analyzed in order to select the parameter values. Namely, if  $E_e$  is much larger than  $\delta$ , it means that the particular choice of parameters yields a poor approximation of the function  $f^*$ , hence large estimation error. If  $E_e$  is small (around  $\delta$  or smaller) and the resulting dimension of the weighting vector  $a_{-1}$ , i.e.  $L_{-1}$ , is large (close to  $N$ ), then the choice of the corresponding parameters is also poor, since a large dictionary is required to fit the data well. On the other hand, if both  $E_e$  and  $L_{-1}$  are small, then the selected parameter combination approximates  $f^*$  well with a relatively small

dictionary, hence such a combination of tuning parameters is preferable. Note that although this analysis might be computationally expensive and time consuming, it can be carried out off-line before the controller becomes operational.

## 5 Experimental example

We tested experimentally the proposed control design algorithm on a water tank system. The latter consists of a round water tank with a water inlet at the top and a small opening at the bottom, through which the water leaks out of the tank. Water is injected from a reservoir into the tank by an electric pump, whose flow can be controlled with the voltage  $v$  applied to the motor. The water flow that the pump generates depends nonlinearly on the applied voltage. The water level in the tank,  $h$ , is measured by a pressure sensor located at its bottom. The pump voltage  $v$  is the input variable and the water level  $h$  is the main internal state, hence referring to the notation used in the previous sections we have  $x = h$  and  $u = v$ . A schematic of the described experimental setup and its picture are given in Fig. 1. The described system is nonlinear and the pump characteristic and the cross section of the water outlet are not known. Therefore, the proposed on-line control design algorithm represents a useful way to build a controller that would regulate the tank water level without the need of going through detailed modeling and parameter identification procedures.

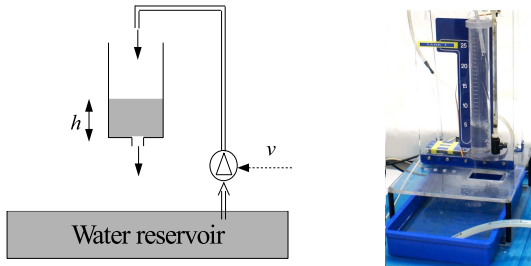


Fig. 1. Schematic of the experimental setup (left) and its picture (right)

A sampling time of 3 seconds was selected to control the system. In order to implement the proposed design scheme, we first generated the training data  $\mathcal{D}_N$  by applying the input sequence shown in Fig. 2 and recording the resulting water level measurements. Then, we employed our theoretical results, in particular Theorem 2, to tune the control algorithm. Specifically, starting with the recorded data, we applied Algorithms 2 and 3 in [21] to estimate the noise bounds  $\epsilon = 0.1$ ,  $\delta = 0.15$  and the Lipschitz constants  $\gamma_g = 1.25$  and  $\gamma^* = 3.6$ . We then assumed the values of  $\lambda_1^* = \lambda_2^* = 1.12$ ,  $\beta^* = 0$  and chose the value of  $\gamma_\Delta = 0.15$ . In addition, from the available training data  $\mathcal{D}_N$  and the estimated values of  $\delta$  and  $\gamma^*$ , by using the method proposed in [11] we calculated the value  $D_0 = 5.8$  in (28) considering

the set  $B_{\bar{x}\bar{r}}$  with  $\bar{x} = 18$  and  $\bar{r} = 9$ . A summary of the parameters that we used is provided in Table 1; we note that all the tuning parameters were selected such that Assumptions 5–7 are valid. The algorithm was implemented and run on a laptop with Intel i7 processor. The worst case execution time during experiments was never larger than 0.8 seconds and the maximal number of kernel functions in the dictionary never exceeded 250.

Table 1  
Tuning parameters.

$\delta$	$\gamma^*$	$\gamma_\Delta$	$\sigma$	$\bar{\mu}$	$p$	$q$
0.15	3.6	0.15	2.9	0.82	1.8	20

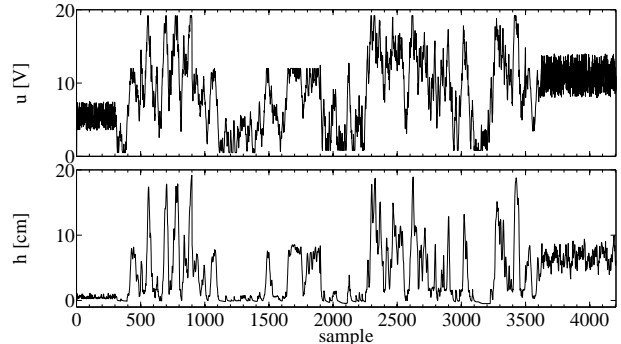


Fig. 2. Initial training data  $\mathcal{D}_N$ . The upper plot shows the applied pump voltages and the lower plot shows the resulting tank water levels.

To illustrate the advantage of the proposed on-line scheme, we compared its performance with that of an approach where the updating of the controller  $f_t$  is stopped before the controller is put into operation (at time step  $t = 0$ ) and the performance of a well tuned proportional-integral (PI) controller. The reference tracking performance for the three cases is compared in Fig. 3 and the applied voltage signals during the experiments are shown in Fig. 4.

In both the experiments with directly designed nonlinear controllers a reasonable tracking performance is achieved. However, for the case when the controller is not updated on-line, the tracking error is significant when the tank water level is low, which is the operating condition with strongest nonlinear behavior.

The PI controller was tuned to achieve good performance when the set point is equal to XX cm, which corresponds to the middle of the range that the reference signal spanned during the initial experiments. As illustrated in Fig. 3, for this value and for all reference values above it, the control performance of the PI controller is quite good, since the system dynamics are close to linear. On the other hand, due to a strong nonlinearity when the water level is below 3 cm, it is not possible to achieve good reference tracking with the PI controller in this range, as illustrated in Fig. 3. However, with the nonlinear controller obtained directly from the data it is possible to achieve good reference tracking performance for the tank water levels in a very wide

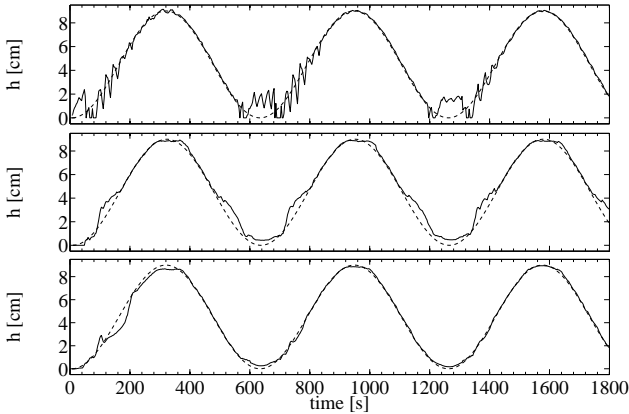


Fig. 3. Measured tank water levels (solid lines) compared with the desired reference (dashed line) for the case when a well tuned PI controller (upper plot) is used and the case when nonlinear controllers directly designed from data with no on-line update (middle plot) and with an on-line update (lower plot) are used.

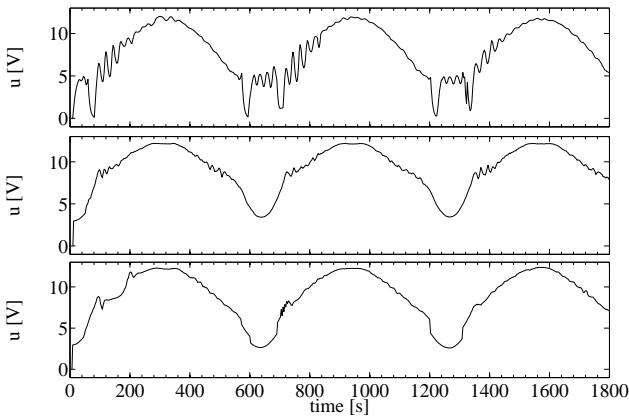


Fig. 4. Pump voltage applied during the experiment with a well tuned PI controller (upper plot), nonlinear controller directly designed from data when there is no on-line update (middle plot) and when the controller is updated during operation (lower plot).

range. In addition, when the controller is updated on-line based on the incoming closed loop measurements, the tracking performance is improved over time and after 1200 seconds very good reference tracking is achieved even for low water levels. The absolute average tracking error during the last 600 seconds of the experiment for the controller that is updated on-line is 0.08 cm, compared to 0.30 cm for the controller that is designed on the basis of initial training data only and 0.61 cm for the PI controller. This illustrates the main advantage of the proposed scheme, which is the ability to exploit the incoming measurements obtained during controller operation in order to improve the tracking performance over time while at the same time having a predictable behavior.

## 6 Conclusion

We proposed a novel on-line direct control design method. The algorithm exploits results from set membership theory for nonlinear systems and the theory of learning by projections, and it retains the advantages of the existing on-line and off-line schemes. We presented theoretical results concerned with closed-loop stability, controller complexity and tracking performance, and experimental results showing the advantages of the method with respect to a purely off-line direct design approach and a well tuned linear controller. **Future research efforts will aim to derive automatic ways to tune and update the design parameters (particularly  $\gamma\Delta$  and  $\sigma$ ) on-line, exploiting the available data, and to extend the approach to systems with multiple inputs.**

## References

- [1] M. C. Campi and S. M. Savaresi. Direct nonlinear control design: the virtual reference feedback tuning (VRFT) approach. *IEEE Transactions on Automatic Control*, 51:14–27, 2006.
- [2] L. Fagiano and C. Novara. Automatic crosswind flight of tethered wings for airborne wind energy: a direct data-driven approach. *Proceeding of the 19th IFAC World Congress, Cape Town, South Africa*, 21:4927–4932, 2014.
- [3] S. Formentin, P. De Filippi, M. Corno, M. Tanelli, and S. M. Savaresi. Data-driven design of braking control systems. *IEEE Transactions on Control Systems Technology*, 21:186–193, 2013.
- [4] J. Van Helvoort, B. de Jager, and M. Steinbuch. Direct data-driven recursive controller unfalsification with analytic update. *Automatica*, 43:2034–2046, 2007.
- [5] H. Hjalmarsson. Iterative feedback tuning—an overview. *International Journal of Adaptive Control and Signal Processing*, 16:373–395, 2002.
- [6] Z. Hou and S. Jin. A novel data-driven control approach for a class of discrete-time nonlinear systems. *IEEE Transactions on Control Systems Technology*, 19:1549–1558, 2011.
- [7] Z. Hou and S. Jin. Controller-dynamic-linearization-based model free adaptive control for discrete-time nonlinear systems. *IEEE Transactions on Industrial Informatics*, 9:2301–2309, 2013.
- [8] Z. Hou and S. Jin. *Model free adaptive control: theory and applications*. CRC Press, 2013.
- [9] H. K. Khalil. *Nonlinear systems*. Prentice Hall, 1996.
- [10] M. Milanese and C. Novara. Set membership identification of nonlinear systems. *Automatica*, 40:957–975, 2004.
- [11] M. Milanese and C. Novara. Computation of local radius of information in SM-IBC identification of nonlinear systems. *Journal of Complexity*, 23:937–951, 2007.
- [12] L. Miskovic, A. Karimi, D. Bonvin, and M. Gevers. Corelation-based tuning of decoupling multivariable controllers. *Automatica*, 43:1481–1494, 2007.
- [13] M. Norgaard, O. Ravn, N. Poulsen, and L. K. Hansen. *Neural networks for modeling and control of dynamic systems*. Springer, 2000.
- [14] C. Novara, L. Fagiano, and M. Milanese. Direct feedback control design for nonlinear systems. *Automatica*, 49:849–860, 2013.
- [15] C. Richard, J. C. M. Bermudez, and P. Honeine. Online prediction of time series data with kernels. *IEEE Transactions on Signal Processing*, 57:1058–1067, 2009.

- [16] M. G. Safonov and T.C. Tsao. The unfalsified control concept and learning. *IEEE Transactions on Automatic Control*, 42:843–847, 1997.
- [17] B. Schölkopf and A. J. Smola. *Learning with kernels*. MIT Press, Cambridge MA, 2001.
- [18] J. Sjöberg, F. De Bruyne, M. Agarwal, B. D. O. Anderson, M. Gevers, F. J. Kraus, and N. Linard. Iterative controller optimization for nonlinear systems. *Control Engineering Practice*, 11:1079–1086, 2003.
- [19] K. Slavakis and I. Yamada. The adaptive projected subgradient method constrained by families of quasi-nonexpansive mappings and its application to online learning. *SIAM Journal of Optimization*, 23:126–152, 2013.
- [20] J. C. Spall and J. A. Cristion. Model-free control of nonlinear stochastic systems with discrete-time measurements. *IEEE Transactions on Automatic Control*, 43:1198–1210, 1998.
- [21] M. Tanaskovic, L. Fagiano, C. Novara, and M. Morari. On-line direct data driven controller design approach with automatic update for some of the tuning parameters, 2015. arXiv:1506.05267 (available online).
- [22] S. Theodoridis, K. Slavakis, and I. Yamada. Adaptive learning in a world of projections. *IEEE Signal Processing Magazine*, 28:97–123, 2011.
- [23] F. J. Traub and H. Wozniakowski. *A general theory of optimal algorithms*. Academic Press, New York, 1980.