IET Control Theory & Applications

Research Article

Offset free data driven control: application to a process control trainer

ISSN 1751-8644
Received on 17th April 2019
Revised 19th August 2019
Accepted on 2nd September 2019
E-First on 1st October 2019
doi: 10.1049/iet-cta.2019.0376
www.ietdl.org

Jose R. Salvador[1] ✉, Daniel Rodriguez Ramirez[1], Teodoro Alamo[1], David Muñoz de la Peña[1]
[1]*Departamento de Ingeniería de Sistemas y Automática, Universidad de Sevilla, Escuela Superior de Ingenieros, Camino de los Descubrimientos s/n. 41092 Sevilla, Spain*
✉ *E-mail: jrsalvador@us.es*

Abstract: This work presents a data driven control strategy able to track a set point without steady-state error. The control sequence is computed as an affine combination of past control signals, which belong to a set of trajectories stored in a process historian database. This affine combination is computed so that the variance of the tracking error is minimised. It is shown that offset free control, that is zero mean tracking error, is achieved under the assumption that the state is measurable, the underlying dynamics are linear and the trajectories of the database share the same error dynamics and are in turn offset free. The proposed strategy learns the underlying controller stored in the database while maintaining its offset free tracking capability in spite of differences in the reference, disturbances and operating conditions. No training phase is required and newly obtained process data can be easily taken into account. The proposed strategy, related to direct weight optimisation learning techniques, is tested on a process control trainer.

1 Introduction

Tracking variable set points without steady-state error, even in presence of constant disturbances, is one of the most desirable capabilities of a feedback control system. Classical control methods for linear systems achieve this by using integral action controllers, as in the case of the ever popular proportional–integral (PI) or proportional–integral–derivative control algorithms. Generally speaking, offset-free control is a well-understood problem, although still researched in more ambitious control formulations [1].

In recent years, the terms data driven control have been applied to different control strategies based on completely different paradigms. Data driven approaches based on time-domain models [2] and frequency-domain methods [3] have been researched, as well as modern paradigms like behavioural [4] or big data based control [5]. Data driven controller tuning methods which directly synthesise a controller with an iterative procedure [6, 7] or which obtain desirable properties as to ensure closed-loop stability [8] have been developed. Data driven predictive control [9, 10] and iterative learning control [11] have also been proposed.

In this class of approaches, offset free control is often achieved by following a reference model and assuming an integrator mode [12], controller or set point adaptation [5], reinforcement learning [13] or exploiting the linear dependence in input–output data of linear processes [2]. Other works in the literature focus on learning the control law from data as in [14, 15] or on identifying the process dynamics [16]. In [17], a data-driven control design technique which is based on the on-line inversion of the model and copes with MIMO non-linear system is presented. Other works, as in [18], focus on achieving high tracking performance through learning for unknown LTI systems subject to unknown disturbances. Non-linear systems with output saturation are addressed in [19]. Also recent tendencies in data driven control are using approximate Q-learning methods [20, 21] and applying distributed optimisation algorithms to tackle adaptative dynamic programming problems [22].

Many of these data driven control problems often rely on learning off-line either a model of the system or directly a control law, in some cases because the learning/estimation algorithms are too complex to be carried out online. Direct weight approximation methods provide low computational burden algorithms [23–25] carried out using affine combinations of locally weighted past data. This class of algorithms have been used for example in lazy learning approaches [26, 27], in which training is deferred until a query is to be answered. Other works that have considered similar ideas in control applications include [28], in which local weighted projection regression is used together with partial least squares combined with a predictive controller [29], where local learning is used in a data driven control method that performs a model free dynamic linearisation within the context of a an adaptive predictive control and [30] which tackles the problem of trajectory tracking with a hierarchical three-level controller that relies on past memorised optimal input–output pairs that are adaptively merged using a similarity measurement.

Following this line of research, in this work (that is an expanded version of [31]) we present a data driven control strategy based on a database of past state trajectories that aim to produce a similar offset free closed-loop response in spite of different operating conditions. The control laws used to generate the database trajectories are assumed to be unknown, so that the proposed strategy will learn the underlying unknown control law that obtain similar closed-loop responses for different operating conditions. The proposed controller computes the input signal to be applied as an affine combination of the control signals. Zero mean tracking error with minimum variance is achieved under the assumption that the state is measurable, the underlying dynamics are linear and the trajectories of the database share the same error dynamics and are in turn offset free.

The idea of using affine combinations of stored trajectories has been used, in a different context, by the authors in previous works [32, 33] in which offset free tracking was not achieved except for the ideal case of a noise and disturbance free database. With respect to other data driven control approaches, the proposed method does not perform an identification step, avoiding the potential problems that can arise in this phase. Furthermore, being related to the lazy learning techniques, no training phase is required to learn the underlying control law in the database, thus it possible to include new data that is made available online. The results of the paper are illustrated by means of an application to a well-known process control trainer [34, Chapter 4].

The paper is organised as follows. The problem formulation is presented in Section 2. The characterisation of steady states and the main contributions of the paper are presented in Sections 3 and 4.

A general formulation for the proposed controller is presented in Section 5, whereas Section 6 shows the experimental results with the scaled laboratory process. The paper ends with the conclusions and future works in Section 7.

2 Problem statement

In this work, we consider a system for which a model of its dynamics is not available, but a particular state representation is known and measurable. This implies that although we propose a model-free approach, some knowledge of the system is needed to define this state. In some cases, the measured state vector corresponds to physical measurements chosen based on first principles; while in others, for those systems in which the state is not completely measurable, the state will be considered composed by present and past values of the system inputs and outputs, following a standard input–output modelling procedure and assuming that an estimation of the order of the system and the delays is known.

Although unknown, we assume that the system is a linear system subject to bounded state and output disturbances, hence, the measured state satisfies the following model:

$$\boldsymbol{x}(t+1) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{w}(t) \tag{1}$$

$$\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{D}\boldsymbol{u}(t) + \boldsymbol{v}(t) \tag{2}$$

where t is the discrete time variable, $\boldsymbol{x}(t) \in \mathbb{R}^{n_x}$, $\boldsymbol{u}(t) \in \mathbb{R}^{n_u}$ and $\boldsymbol{y}(t) \in \mathbb{R}^{n_y}$ are the measured state, input and output of the system at time step t, respectively, $\boldsymbol{A} \in \mathbb{R}^{n_x \times n_x}$, $\boldsymbol{B} \in \mathbb{R}^{n_x \times n_u}$, $\boldsymbol{C} \in \mathbb{R}^{n_y \times n_x}$ and $\boldsymbol{D} \in \mathbb{R}^{n_y \times n_u}$ are the unknown system matrices and $\boldsymbol{w}(t) \in \mathbb{R}^{n_x}$ and $\boldsymbol{v}(t) \in \mathbb{R}^{n_y}$ are unknown state and output disturbances, respectively, with non-zero mean, that is

$$E\{\boldsymbol{w}(t)\} = \boldsymbol{w}^e, \quad E\{\boldsymbol{v}(t)\} = \boldsymbol{v}^e. \tag{3}$$

Note that the state and output disturbances include all the discrepancies between measured and real states due to noises and, to some extent, uncertainties and slight non-linearities.

The control objective is to track a reference $\boldsymbol{r} \in \mathbb{R}^{n_y}$ without offset. The stochastic disturbances considered in (1) and (2) make impossible to reach true offset free control. Thus, by offset free control we mean that $\boldsymbol{y}$ is probabilistically ultimately bounded into a set with mean equal to $\boldsymbol{r}$ [35]. Also, to ensure a well-posed control problem, we assume the following.

Assumption 1: The system given by (1) and (2) is assumed to have full state and output controllability. Furthermore, the reference $\boldsymbol{r} \in \mathbb{R}^{n_y}$ is assumed to be reachable.

2.1 Historian database

In this work, instead of using a model to define the controller, we present a procedure to take a decision on behalf of the information stored in a historian database. This historian database has a large number of past offset free state, input, output and reference trajectories. Each trajectory stored in the database, which may be of different length, represents the closed-loop behaviour of the system given by (1) and (2) controlled with a different unknown control law and constant reference. In addition, we assume that the disturbances of each stored trajectory are characterised by a possibly different mean value. Using this framework, the disturbance may account for time-varying and state-dependent perturbations.

If the measured state for each trajectory is given by $\tilde{\mathbf{x}}$ and its corresponding steady-state value is denoted as $\tilde{\mathbf{x}}^e$, we assume that all the trajectories in the historian database satisfy the following property:

Assumption 2: It is assumed that the dynamics of the trajectories of the database satisfy

$$\tilde{\mathbf{x}}(t+1) - \tilde{\mathbf{x}}^e = \mathbf{A}_c(\tilde{\mathbf{x}}(t) - \tilde{\mathbf{x}}^e) + \tilde{\boldsymbol{\tau}}(t), \tag{4}$$

where $\mathbf{A}_c$ is a Schur stable matrix and $\tilde{\boldsymbol{\tau}}(t)$ a zero mean error term with bounded covariance.

The objective of this work is to learn from the historian database the underlying controller defined by Assumption 2, while preserving its offset free property in the presence of different mean perturbations. Note that using a standard function approximation procedure to determine a function that relates state, reference and output would yield a static controller that would not provide offset free constant reference tracking for different mean perturbations.

There are different areas of applicability for the proposed control scheme depending on the origin of the historian database. The trajectories stored can be obtained from real operation (which may include manual operation and different controllers) of the system in the past, or from dedicated tests. In the first case, the controller objective is to learn the underlying control law that has provided good performance in the past, in spite of changing operating conditions. This procedure may be of interest in complex systems for which great amounts of data are available.

In the second case, using closed-loop testing may be a benefit, for example when trying to control a open-loop unstable system, because identification is avoided. In this case, because the transient closed-loop response will depend on those of the trajectories stored in the database, it is important to store trajectories that exhibit a good control performance so that the controller inherits it. The design criteria can be anything from a performance cost to transient response measures (e.g. overshoot) that can be used to characterise the good trajectories. In some sense, tuning is carried out using extensive off-line experimentation, which depending on the application, may or may be not possible or be less appropriate than using standard identification modelling techniques. This procedure is shown in the temperature control application example.

Besides closed-loop testing and real past behaviours, it is also possible to include open-loop tests for stable systems in which for a predefined input sequence, obtained for example from step tests with filtering or lead/lag, the reference is defined for the steady state reached. In addition, these results can be combined with trajectories obtained from past operation or closed-loop testing.

Remark 1: The requirement that all the trajectories of the database satisfy (4) could seem very restrictive but it is consistent with the standard control practice in which the process is desired to have the same performance in spite of the different operating conditions. For a real process this implies that different controllers used to generate the historian database were characterised by a similar closed-loop dynamic behaviour (e.g. similar rise time and overshoot). Note also that the zero mean error term is not bounded, which provides a certain degree of robustness for closed-loop trajectories with slightly different error dynamics. Furthermore, the bounded covariance implies that the probability of getting high errors is small.

2.2 Building the candidate set

The information stored in the historian database is used to generate a set of possible candidate tuples $j \in S$. Each tuple consists of the inputs, states, outputs and corresponding reference of a particular trajectory and sampling time of the historian database. These tuples will be used to define the optimisation problem that has to be solved at each sampling time to implement the proposed controller and is built off-line using all the trajectories available in the database excluding those that are the first stored element of a particular trajectory. The reason of this is that, in the proposed strategy, for each candidate tuple, it is necessary to know the state and input value of the previous sample time in its corresponding trajectory.

For each candidate, we denote as $\boldsymbol{x}_j(0), \boldsymbol{u}_j(0), \boldsymbol{y}_j(0), \boldsymbol{w}_j(0)$ and $\boldsymbol{v}_j(0)$ the state, input, output and disturbances at the corresponding sample time of the candidate, and $\boldsymbol{x}_j(k), \boldsymbol{u}_j(k), \boldsymbol{y}_j(k), \boldsymbol{w}_j(k)$ and $\boldsymbol{v}_j(k)$ the corresponding values shifted k sample times (i.e. $\boldsymbol{x}_j(-1)$ denotes the state at the sample time before the candidate's

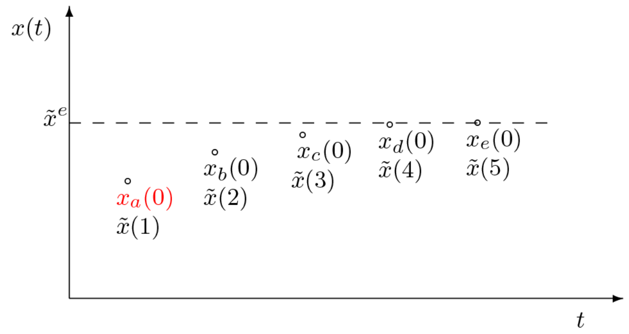

Fig. 1 *Example of the notation employed to describe the candidates that conform the set S*

corresponding sample time). Moreover, for each candidate, we denote as $\boldsymbol{r}_j$ the corresponding reference of its trajectory (note that the reference does not depend on the time because it is assumed constant for each trajectory). The database does not store the values of the disturbances, and hence the variables $\boldsymbol{w}_j(\cdot)$ and $\boldsymbol{v}_j(\cdot)$ are unknown for the controller. Although the disturbances are not stored in the database, their effects are indirectly stored by means of the state and output measurements.

Sampling time index variable t refers to the sampling time of a real trajectory, either from the historian data base or online implementation, in particular, in the controller implementation $x(t)$ refers to the current state measurement. On the other hand, sampling time index variable k refers to displacement relative to a candidate of the set S. Note that the only values of k needed to define the proposed controller are $k = 0$ and $k = -1$. In the proof of the main theorem, $k = 1$ is also used.

To clarify how to define the candidates from the trajectories stored in the historian database, consider Fig. 1, which shows an example of a one-dimensional state trajectory with five sampled values, $\tilde{x}(1)$ to $\tilde{x}(5)$ that reach its corresponding target state $\tilde{x}^e$. From this trajectory, four different state candidates can be defined, one for each state in the trajectory that has a predecessor. In the figure, the possible candidates are denoted as $x_j(0)$ with $j \in a, b, c, d, e$. In this particular case, $x_a(0)$ cannot be included in S because $x_a(-1)$ does not exists. The value of each candidate is defined by a different sample time; that is

$$\begin{aligned} x_a(0) &= \tilde{x}(1) \\ x_b(0) &= \tilde{x}(2) \\ x_c(0) &= \tilde{x}(3) \\ x_d(0) &= \tilde{x}(4) \\ x_e(0) &= \tilde{x}(5)\,. \end{aligned} \tag{5}$$

The corresponding previous state of each candidate are also defined by the states in this trajectory; that is

$$\begin{aligned} x_a(-1) &= NA \\ x_b(-1) &= \tilde{x}(1) \\ x_c(-1) &= \tilde{x}(2) \\ x_d(-1) &= \tilde{x}(3) \\ x_e(-1) &= \tilde{x}(4)\,. \end{aligned} \tag{6}$$

With this notation, the candidates satisfy

$$\begin{aligned} \boldsymbol{x}_j(k+1) &= \boldsymbol{A}\boldsymbol{x}_j(k) + \boldsymbol{B}\boldsymbol{u}_j(k) + \boldsymbol{w}_j(k) \\ \boldsymbol{y}_j(k) &= \boldsymbol{C}\boldsymbol{x}_j(k) + \boldsymbol{D}\boldsymbol{u}_j(k) + \boldsymbol{v}_j(k)\,. \end{aligned} \tag{7}$$

For each candidate, we denote its corresponding reference as $\boldsymbol{r}_j$ which has a constant value for the whole trajectory, thus shared with the other candidates that are from the same trajectory. In addition, for each candidate, we denote the constant mean disturbance values of its corresponding trajectory as $\boldsymbol{w}_j^e$ and $\boldsymbol{v}_j^e$; that is

$$E\{\boldsymbol{w}_j\} = \boldsymbol{w}_j^e, \quad E\{\boldsymbol{v}_j\} = \boldsymbol{v}_j^e\,. \tag{8}$$

Taking into account that all the trajectories in the database satisfy Assumption 2, the candidates also satisfy

$$\mathbf{x}_j(k+1) - \mathbf{x}_j^e = \mathbf{A}_c(\mathbf{x}_j(k) - \mathbf{x}_j^e) + \boldsymbol{\tau}_j(k)\,. \tag{9}$$

Assumption 2 and (9) implies a direct consequence for all the candidates trajectories. Given $0 < p_j \le 1$ and $\Omega_j \subset \mathbb{R}^{n_x + n_u}$, there exists N_j such that for its corresponding future state and input trajectories satisfy

$$Pr\big[\boldsymbol{x}_j(k) - \boldsymbol{x}_j^e \in \Omega_j\big] \ge p_j, \quad \forall k \ge N_j \tag{10}$$

where Ω_j is a probabilistic ultimate bound ([35]) which is a neighbourhood around the steady state $\boldsymbol{x}_j^e$ whose size is related to the covariance of $\boldsymbol{\tau}_j(k)$. Notice that N_j depends on the initial state of the candidate trajectory $\boldsymbol{x}_j(0)$. We also remark that if the disturbances $\boldsymbol{\tau}_j(k)$ are bounded, it is possible to find a deterministic ultimate bound, that is a Ω_j which satisfies (10) with probability $p_j = 1$ [36, Chapter 4].

Remark 2: The size of the candidates set influences the learning capabilities of the proposed approach and it is also directly related to the computational burden. On the other hand, as in other data based and learning approaches, the dimension of the state vector (and other variables) affects the necessary size of the set S. From a practical point of view, this leads to use the minimum dimension state representation necessary for the control objectives considered.

3 Steady-state characterisation

Although the proposed controller is based on state feedback, we have considered an output reference tracking problem. In this section, we consider the notion of steady state for the output equal to a given reference $\boldsymbol{r}$. This characterisation will be used on the proof of the main result of this paper.

The pair $(\boldsymbol{x}^e, \boldsymbol{u}^e)$, with $\boldsymbol{x}^e \in \mathbb{R}^{n_x}$, $\boldsymbol{u}^e \in \mathbb{R}^{n_u}$, represents a steady state and steady control input if and only if

$$\begin{aligned} \boldsymbol{x}^e &= \boldsymbol{A}\boldsymbol{x}^e + \boldsymbol{B}\boldsymbol{u}^e + \boldsymbol{w}^e \\ \boldsymbol{r} &= \boldsymbol{C}\boldsymbol{x}^e + \boldsymbol{D}\boldsymbol{u}^e + \boldsymbol{v}^e\,. \end{aligned} \tag{11}$$

We assume that each of the trajectories of the database has been generated to track a particular reference and moreover, we assume that the control input trajectory drives the output to this reference in spite of the disturbances. Thus, similarly to (11), a pair $(\boldsymbol{x}_j^e, \boldsymbol{u}_j^e)$, with $\boldsymbol{x}_j^e \in \mathbb{R}^{n_x}$, $\boldsymbol{u}_j^e \in \mathbb{R}^{n_u}$, represents a steady state and steady control input for the trajectory of the jth candidate if and only if

$$\begin{aligned} \boldsymbol{x}_j^e &= \boldsymbol{A}\boldsymbol{x}_j^e + \boldsymbol{B}\boldsymbol{u}_j^e + \boldsymbol{w}_j^e, \\ \boldsymbol{r}_j &= \boldsymbol{C}\boldsymbol{x}_j^e + \boldsymbol{D}\boldsymbol{u}_j^e + \boldsymbol{v}_j^e, \end{aligned} \tag{12}$$

The following assumption is a necessary controllability condition and it is required to ensure that a steady state pair $(\boldsymbol{x}_j^e, \boldsymbol{u}_j^e)$ exists for a given reference $\boldsymbol{r}$.

Assumption 3: Let $\boldsymbol{G}$ be defined as

$$\boldsymbol{G} = \begin{bmatrix} \mathrm{I} - \boldsymbol{A} & -\boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix}. \tag{13}$$

It is assumed that $\boldsymbol{G}$ has full column rank.

It is possible to provide a characterisation of the pair $(\boldsymbol{x}_j^e, \boldsymbol{u}_j^e)$, through Theorems 1 and 2 given in the following.

Theorem 1: Suppose that Assumption 3 holds, and that $\boldsymbol{G}$ is a square matrix. Then, the pair $(\boldsymbol{x}_j^e, \boldsymbol{u}_j^e)$ is uniquely given by a linear expression of $\boldsymbol{r}_j - \boldsymbol{v}_j^e$ and $\boldsymbol{w}_j^e$.

Proof: The equality constraints (12) can be rewritten as

$$\begin{bmatrix} \mathrm{I} - \boldsymbol{A} & -\boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_j^e \\ \boldsymbol{u}_j^e \end{bmatrix} = \begin{bmatrix} \boldsymbol{w}_j^e \\ \boldsymbol{r}_j - \boldsymbol{v}_j^e \end{bmatrix}. \tag{14}$$

Since it is assumed that $\boldsymbol{G}$ has full column rank, the previous system of equations has a unique solution equal to

$$\begin{bmatrix} \boldsymbol{x}_j^e \\ \boldsymbol{u}_j^e \end{bmatrix} = \boldsymbol{G}^{-1} \begin{bmatrix} \boldsymbol{w}_j^e \\ \boldsymbol{r}_j - \boldsymbol{v}_j^e \end{bmatrix}, \tag{15}$$

and this completes the proof. □

Theorem 2: Suppose that Assumption 3 holds, and that the number of columns of $\boldsymbol{G}$ is larger than the number of rows, and that the steady-state pair $(\boldsymbol{x}_j^e, \boldsymbol{u}_j^e)$ is defined as the solution of

$$\begin{aligned} &\min_{\boldsymbol{x}_j^e, \boldsymbol{u}_j^e} \frac{1}{2} \parallel \boldsymbol{x}_j^e \parallel_{\boldsymbol{Q}}^2 + \frac{1}{2} \parallel \boldsymbol{u}_j^e \parallel_{\boldsymbol{R}}^2 \\ &s.t. \\ &\quad \boldsymbol{G} \begin{bmatrix} \boldsymbol{x}_j^e \\ \boldsymbol{u}_j^e \end{bmatrix} = \boldsymbol{b}_j^e, \end{aligned} \tag{16}$$

where

$$\boldsymbol{b}_j^e = \begin{bmatrix} \boldsymbol{w}_j^e \\ \boldsymbol{r}_j - \boldsymbol{v}_j^e \end{bmatrix}, \quad \boldsymbol{Q} > 0 \quad \text{and} \quad \boldsymbol{R} > 0.$$

Then the pair $(\boldsymbol{x}_j^e, \boldsymbol{u}_j^e)$ is uniquely given by a linear expression of $\boldsymbol{r}_j - \boldsymbol{v}_j^e$ and $\boldsymbol{w}_j^e$.

Proof: Let $\boldsymbol{z}_j^e$ be

$$\boldsymbol{z}_j^e = \begin{bmatrix} \boldsymbol{x}_j^e \\ \boldsymbol{u}_j^e \end{bmatrix}, \tag{17}$$

and $\boldsymbol{0}_{a,b}$ the zero matrix with a rows and b columns. It is well known ([37, Chapter 10]) that if the block diagonal matrix

$$\boldsymbol{\Upsilon} = \begin{bmatrix} \boldsymbol{Q} & \boldsymbol{0}_{n_x, n_u} \\ \boldsymbol{0}_{n_u, n_x} & \boldsymbol{R} \end{bmatrix} \tag{18}$$

is strictly definite positive and $\boldsymbol{G}$ has full rank, then the optimal value for $\boldsymbol{z}_j^e$, defined as in (17) and solution to (16), is given by the following system of equations:

$$\begin{bmatrix} \boldsymbol{\Upsilon} & \boldsymbol{G}^\top \\ \boldsymbol{G} & \boldsymbol{0}_{n_G, n_G} \end{bmatrix} \begin{bmatrix} \boldsymbol{z}_j^e \\ \boldsymbol{\beta}_j^e \end{bmatrix} = \begin{bmatrix} \boldsymbol{0}_{n_z, 1} \\ \boldsymbol{b}_j^e \end{bmatrix}, \tag{19}$$

where $n_z = n_x + n_u$ is $\boldsymbol{\Upsilon}$ dimension, $n_G < n_z$ is the number of rows of $\boldsymbol{G}$ and $\boldsymbol{\beta}_j^e$ denotes the optimal value for the dual variables of the optimisation problem. From

$$\boldsymbol{\Upsilon} \boldsymbol{z}_j^e + \boldsymbol{G}^\top \boldsymbol{\beta}_j^e = \boldsymbol{0}_{n_z, 1} \tag{20}$$

we obtain

$$\boldsymbol{z}_j^e = -\boldsymbol{\Upsilon}^{-1} \boldsymbol{G}^\top \boldsymbol{\beta}_j^e. \tag{21}$$

From (19) we also have

$$\boldsymbol{G} \boldsymbol{z}_j^e = \boldsymbol{b}_j^e. \tag{22}$$

Substituting (21) into (22), we obtain the optimal value for $\boldsymbol{\beta}_j^e$

$$\begin{aligned} -\boldsymbol{G}\boldsymbol{\Upsilon}^{-1}\boldsymbol{G}^\top \boldsymbol{\beta}_j^e &= \boldsymbol{b}_j^e \\ \boldsymbol{\beta}_j^e &= -(\boldsymbol{G}\boldsymbol{\Upsilon}^{-1}\boldsymbol{G}^\top)^{-1} \boldsymbol{b}_j^e. \end{aligned} \tag{23}$$

From (21) and (23) results

$$\boldsymbol{z}_j^e = \boldsymbol{\Upsilon}^{-1}\boldsymbol{G}^\top (\boldsymbol{G}\boldsymbol{\Upsilon}^{-1}\boldsymbol{G}^\top)^{-1} \boldsymbol{b}_j^e. \tag{24}$$

We now denote

$$\boldsymbol{M}_\Upsilon = \boldsymbol{\Upsilon}^{-1}\boldsymbol{G}^\top (\boldsymbol{G}\boldsymbol{\Upsilon}^{-1}\boldsymbol{G}^\top)^{-1}, \tag{25}$$

to get

$$\boldsymbol{z}_j^e = \boldsymbol{M}_\Upsilon \boldsymbol{b}_j^e. \tag{26}$$

Thus, it is shown that there is a linear relationship between vector $\boldsymbol{b}_j^e$ and vector $\boldsymbol{z}_j^e$. □

4 Offset free data driven control

In this paper, we propose to use a control law derived from the control signals in the candidates set S. The control signal to be applied at time t will be computed as a weighted sum of the initial control signals of every candidate of S, that is

$$\boldsymbol{u}(t) = \sum_{j \in S} \lambda_j \boldsymbol{u}_j(0). \tag{27}$$

The following sections discuss the conditions that $\{\lambda_j\}$ must meet in order to recover the underlying closed-loop properties of the trajectories stored in the historian database. Offset free with minimum variance must be achieved for all possible references $\boldsymbol{r}$ and mean disturbance values $\boldsymbol{w}^e, \boldsymbol{v}^e$, not only those included in the candidate's data.

At a first approximation, similar to that of [32], suppose that we compute $\{\lambda_j\}$ so that the current state, output and reference are a combination of the candidates

$$\boldsymbol{x}(t) = \sum_{j \in S} \lambda_j \boldsymbol{x}_j(0), \tag{28}$$

$$1 = \sum_{j \in S} \lambda_j. \tag{29}$$

$$\boldsymbol{y}(t) = \sum_{j \in S} \lambda_j \boldsymbol{y}_j(0), \tag{30}$$

$$\boldsymbol{r} = \sum_{j \in S} \lambda_j \boldsymbol{r}_j. \tag{31}$$

Taking into account (27) and (28) in the state equation (1)

$$\begin{aligned} \boldsymbol{x}(t+1) &= \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{w}(t) \\ &= \boldsymbol{A} \sum_{j \in S} \lambda_j \boldsymbol{x}_j(0) + \boldsymbol{B} \sum_{j \in S} \lambda_j \boldsymbol{u}_j(0) + \boldsymbol{w}(t) \\ &= \sum_{j \in S} \lambda_j \big(\boldsymbol{A}\boldsymbol{x}_j(0) + \boldsymbol{B}\boldsymbol{u}_j(0) \big) + \boldsymbol{w}(t) \\ &= \sum_{j \in S} \lambda_j \boldsymbol{x}_j(1) + \sum_{j \in S} \lambda_j (\boldsymbol{w}(t) - \boldsymbol{w}_j(0)). \end{aligned} \tag{32}$$

We notice that approximating the next state $\boldsymbol{x}(t+1)$ as a combination of the next states of the candidates yields an estimation error given by

$$\sum_{j \in S} \lambda_j(\boldsymbol{w}(t) - \boldsymbol{w}_j(0)), \tag{33}$$

which is not guaranteed to have zero mean. This means that if we use a control strategy similar to that of [32], the state in $t+1$ will be a combination of the states of every trajectory in S, but it will have an offset caused by the approximation error even if the trajectories in the database are offset free.

To obtain an offset free control, additional constraints have to be imposed on $\{\lambda_j\}$ in order to force the controller to have memory, in the sense that the previous state and applied control signal have to be related with the previous ones of every trajectory in S through the values of $\{\lambda_j\}$. The constraints

$$\boldsymbol{x}(t-1) = \sum_{j \in S} \lambda_j \boldsymbol{x}_j(-1), \tag{34}$$

$$\boldsymbol{u}(t-1) = \sum_{j \in S} \lambda_j \boldsymbol{u}_j(-1), \tag{35}$$

take this issue into account. The next properties demonstrate that the estimation error of the next state obtained using a set of weights that satisfy the above-mentioned constraints has a zero mean error term. This property will be used to prove offset free tracking.

Property 1: Assuming that (27)–(29), (34) and (35) hold; then

$$\boldsymbol{x}(t+1) = \sum_{j \in S} \lambda_j \boldsymbol{x}_j(1) + \boldsymbol{e}_x(t), \tag{36}$$

where $\boldsymbol{e}_x(t)$ is a zero mean error term.

Proof: From the state equation (1) shifted backwards and the constraints (28), (34) and (35)

$$\begin{aligned} \boldsymbol{w}(t-1) &= \boldsymbol{x}(t) - \boldsymbol{A}\boldsymbol{x}(t-1) - \boldsymbol{B}\boldsymbol{u}(t-1) \\ &= \sum_{j \in S} \lambda_j \boldsymbol{x}_j(0) - \boldsymbol{A} \sum_{j \in S} \lambda_j \boldsymbol{x}_j(-1) \\ &\quad -\boldsymbol{B} \sum_{j \in S} \lambda_j \boldsymbol{u}_j(-1) \\ &= \sum_{j \in S} \lambda_j \big(\boldsymbol{x}_j(0) - \boldsymbol{A}\boldsymbol{x}_j(-1) - \boldsymbol{B}\boldsymbol{u}_j(-1)\big) \\ &= \sum_{j \in S} \lambda_j \boldsymbol{w}_j(-1). \end{aligned} \tag{37}$$

From (27) and (28), as mentioned before, we obtain the estimated value of the next time step

$$\begin{aligned} \boldsymbol{x}(t+1) &= \sum_{j \in S} \lambda_j \boldsymbol{x}_j(1) + \sum_{j \in S} \lambda_j(\boldsymbol{w}(t) - \boldsymbol{w}_j(0)) \\ &= \sum_{j \in S} \lambda_j \boldsymbol{x}_j(1) + \boldsymbol{w}(t) - \sum_{j \in S} \lambda_j \boldsymbol{w}_j(0). \end{aligned} \tag{38}$$

Defining the prediction error as

$$\boldsymbol{e}_x(t) = \boldsymbol{x}(t+1) - \sum_{j \in S} \lambda_j \boldsymbol{x}_j(1), \tag{39}$$

and substituting in (38), we have

$$\boldsymbol{e}_x(t) = \boldsymbol{w}(t) - \sum_{j \in S} \lambda_j \boldsymbol{w}_j(0). \tag{40}$$

From (37) we obtain

$$-\boldsymbol{w}(t-1) + \sum_{j \in S} \lambda_j \boldsymbol{w}_j(-1) = 0. \tag{41}$$

Adding this equality to (40) yields

$$\begin{aligned} \boldsymbol{e}_x(t) &= \boldsymbol{w}(t) - \boldsymbol{w}(t-1) - \sum_{j \in S} \lambda_j(\boldsymbol{w}_j(0) - \boldsymbol{w}_j(-1)) \\ &= \Delta\boldsymbol{w}(t) - \Delta\boldsymbol{w}(t-1) - \sum_{j \in S} \lambda_j(\Delta\boldsymbol{w}_j(0) - \Delta\boldsymbol{w}_j(-1)), \end{aligned} \tag{42}$$

where $\Delta\boldsymbol{w}(\cdot) = \boldsymbol{w}(\cdot) - \boldsymbol{w}^e$ and $\Delta\boldsymbol{w}_j(\cdot) = \boldsymbol{w}_j(\cdot) - \boldsymbol{w}_j^e$. Notice that $\boldsymbol{e}_x(t)$ is a zero mean error term because, by construction, $\Delta\boldsymbol{w}_j(\cdot)$ and $\Delta\boldsymbol{w}(\cdot)$ have zero mean. □

The weights $\{\lambda_j\}$ obtained can be used not only to obtain an estimation of the future state, but also to estimate the mean value of the current perturbations with zero mean error. This property is proved next.

Property 2: Assuming that (27)–(30), (34) and (35) hold; then

$$\boldsymbol{w}^e = \sum_{j \in S} \lambda_j \boldsymbol{w}_j^e + \boldsymbol{e}_w(t), \tag{43}$$

$$\boldsymbol{v}^e = \sum_{j \in S} \lambda_j \boldsymbol{v}_j^e + \boldsymbol{e}_v(t), \tag{44}$$

where $\boldsymbol{e}_w(t)$ and $\boldsymbol{e}_v(t)$ are zero mean error terms.

Proof:

$$\begin{aligned} \boldsymbol{x}(t) &= \boldsymbol{A}\boldsymbol{x}(t-1) + \boldsymbol{B}\boldsymbol{u}(t-1) + \boldsymbol{w}(t) \\ &= \boldsymbol{A}\boldsymbol{x}(t-1) + \boldsymbol{B}\boldsymbol{u}(t-1) + \boldsymbol{w}^e + \Delta\boldsymbol{w}(t). \end{aligned} \tag{45}$$

Thus

$$\begin{aligned} \boldsymbol{w}^e &= \boldsymbol{x}(t) - \boldsymbol{A}\boldsymbol{x}(t-1) - \boldsymbol{B}\boldsymbol{u}(t-1) - \Delta\boldsymbol{w}(t) \\ &= \sum_{j \in S} \lambda_j(\boldsymbol{x}_j(0) - \boldsymbol{A}\boldsymbol{x}_j(-1) - \boldsymbol{B}\boldsymbol{u}_j(-1)) - \Delta\boldsymbol{w}(t) \\ &= \sum_{j \in S} \lambda_j \boldsymbol{w}_j(-1) - \Delta\boldsymbol{w}(t) \\ &= \sum_{j \in S} \lambda_j(\boldsymbol{w}_j^e + \Delta\boldsymbol{w}_j(-1)) - \Delta\boldsymbol{w}(t) \\ &= \sum_{j \in S} \lambda_j \boldsymbol{w}_j^e + \sum_{j \in S} \lambda_j \Delta\boldsymbol{w}_j(-1) - \Delta\boldsymbol{w}(t) \\ &= \sum_{j \in S} \lambda_j \boldsymbol{w}_j^e + \boldsymbol{e}_w(t), \end{aligned} \tag{46}$$

where $\boldsymbol{e}_w(t)$ is a zero mean error term. In a similar way

$$\begin{aligned} \boldsymbol{y}(t) &= \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{D}\boldsymbol{u}(t) + \boldsymbol{v}(t) \\ &= \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{D}\boldsymbol{u}(t) + \boldsymbol{v}^e + \Delta\boldsymbol{v}(t). \end{aligned} \tag{47}$$

Thus

$$\begin{aligned} \boldsymbol{v}^e &= \boldsymbol{y}(t) - \boldsymbol{C}\boldsymbol{x}(t) - \boldsymbol{D}\boldsymbol{u}(t) - \Delta\boldsymbol{v}(t) \\ &= \sum_{j \in S} \lambda_j(\boldsymbol{y}_j(0) - \boldsymbol{C}\boldsymbol{x}_j(0) - \boldsymbol{D}\boldsymbol{u}_j(0)) - \Delta\boldsymbol{v}(t) \\ &= \sum_{j \in S} \lambda_j \boldsymbol{v}_j(0) - \Delta\boldsymbol{v}(t) \\ &= \sum_{j \in S} \lambda_j(\boldsymbol{v}_j^e + \Delta\boldsymbol{v}_j(0)) - \Delta\boldsymbol{v}(t) \\ &= \sum_{j \in S} \lambda_j \boldsymbol{v}_j^e + \sum_{j \in S} \lambda_j \Delta\boldsymbol{v}_j(0) - \Delta\boldsymbol{v}(t) \\ &= \sum_{j \in S} \lambda_j \boldsymbol{v}_j^e + \boldsymbol{e}_v(t), \end{aligned} \tag{48}$$

where $\boldsymbol{e}_v(t)$ is a zero mean error term. □

The control objective is to drive the output to the reference $\boldsymbol{r}$. Taking into account Assumption 2, this implies that, for each candidate, the state and the input have to reach the corresponding steady-state values $\boldsymbol{x}_j^e, \boldsymbol{u}_j^e$. In the next result, it is proved that the evolution of the deviation of the state from its target state can be estimated from the weighted evolution of the deviation of each of the candidate trajectories from its corresponding target states with zero mean error. This relation will be used to prove convergence and zero mean tracking error.

Theorem 3: Suppose that

$$\begin{bmatrix} \boldsymbol{x}^e \\ \boldsymbol{u}^e \end{bmatrix} = \boldsymbol{M} \begin{bmatrix} \boldsymbol{w}^e \\ \boldsymbol{r} - \boldsymbol{v}^e \end{bmatrix}, \tag{49}$$

$$\begin{bmatrix} \boldsymbol{x}_j^e \\ \boldsymbol{u}_j^e \end{bmatrix} = \boldsymbol{M} \begin{bmatrix} \boldsymbol{w}_j^e \\ \boldsymbol{r}_j - \boldsymbol{v}_j^e \end{bmatrix}, \tag{50}$$

and that (27)–(31), (34) and (35) holds; then

$$\boldsymbol{x}(t+1) - \boldsymbol{x}^e = \sum_{j \in S} \lambda_j (\boldsymbol{x}_j(1) - \boldsymbol{x}_j^e) + \boldsymbol{\eta}(t), \tag{51}$$

where $\boldsymbol{\eta}(t)$ is a zero mean error term.

Proof: From property 2 it holds that

$$\begin{aligned} \begin{bmatrix} \boldsymbol{x}^e \\ \boldsymbol{u}^e \end{bmatrix} &= \boldsymbol{M} \begin{bmatrix} \boldsymbol{w}^e \\ \boldsymbol{r} - \boldsymbol{v}^e \end{bmatrix} \\ &= \boldsymbol{M} \begin{bmatrix} \sum_{j \in S} \lambda_j \boldsymbol{w}_j^e + \boldsymbol{e}_w(t) \\ \sum_{j \in S} \lambda_j \boldsymbol{r}_j - \sum_{j \in S} \lambda_j \boldsymbol{v}_j^e - \boldsymbol{e}_v(t) \end{bmatrix} \\ &= \sum_{j \in S} \lambda_j \boldsymbol{M} \begin{bmatrix} \boldsymbol{w}_j^e \\ \boldsymbol{r}_j - \boldsymbol{v}_j^e \end{bmatrix} + \boldsymbol{M} \begin{bmatrix} \boldsymbol{e}_w(t) \\ -\boldsymbol{e}_v(t) \end{bmatrix} \\ &= \sum_{j \in S} \lambda_j \begin{bmatrix} \boldsymbol{x}_j^e \\ \boldsymbol{u}_j^e \end{bmatrix} + \boldsymbol{M} \begin{bmatrix} \boldsymbol{e}_w(t) \\ -\boldsymbol{e}_v(t) \end{bmatrix}. \end{aligned} \tag{52}$$

From the previous equation

$$\boldsymbol{x}^e = \sum_{j \in S} \lambda_j \boldsymbol{x}_j^e + \boldsymbol{e}_s(t), \tag{53}$$

where $\boldsymbol{e}_s(t)$ is a zero mean error term. From property 1 and subtracting (53) to (36) we finally obtain

$$\begin{aligned} \boldsymbol{x}(t+1) - \boldsymbol{x}^e &= \sum_{j \in S} \lambda_j (\boldsymbol{x}_j(1) - \boldsymbol{x}_j^e) + \boldsymbol{e}_x(t) - \boldsymbol{e}_s(t) \\ &= \sum_{j \in S} \lambda_j (\boldsymbol{x}_j(1) - \boldsymbol{x}_j^e) + \boldsymbol{\eta}(t), \end{aligned} \tag{54}$$

where $\boldsymbol{\eta}(t) = \boldsymbol{e}_x(t) - \boldsymbol{e}_s(t)$. □

Theorem 3 and the stability of the error dynamics of all the candidates [that is a direct consequence of Assumption 2.] will be used in the following to prove the main result of the paper. Since the disturbances $\boldsymbol{w}(t)$ and $\boldsymbol{v}(t)$ are random variables with non-zero mean, offset free tracking will be attained if we can prove that the closed-loop trajectory converges to a neighbourhood of $\boldsymbol{x}^e$.

Theorem 4: Under the assumptions of Theorem 3, there exist $\gamma \in (0, 1)$ such that

$$\| \boldsymbol{x}(t+1) - \boldsymbol{x}^e \|_{\boldsymbol{P}} \leq \| \boldsymbol{\psi}(t) \|_{\boldsymbol{P}} + \sqrt{\gamma} \| \boldsymbol{x}(t) - \boldsymbol{x}^e \|_{\boldsymbol{P}}, \tag{55}$$

where $\boldsymbol{\psi}(t)$ is a zero mean error term, which implies that the state converges to a neighbourhood of $\boldsymbol{x}^e$.

Proof: Under the assumptions of Theorem 3 we have that

$$\boldsymbol{x}(t+1) - \boldsymbol{x}^e = \sum_{j \in S} \lambda_j (\boldsymbol{x}_j(1) - \boldsymbol{x}_j^e) + \boldsymbol{\eta}(t). \tag{56}$$

From Assumption 2 we also have

$$\boldsymbol{x}_j(k+1) - \boldsymbol{x}_j^e = \boldsymbol{A_c}(\boldsymbol{x}_j(k) - \boldsymbol{x}_j^e) + \boldsymbol{\tau}_j(k), \tag{57}$$

where $\boldsymbol{A_c}$ is Schur stable and $\boldsymbol{\tau}_j(t)$ has zero mean. Therefore, each trajectory satisfies

$$\boldsymbol{x}_j(1) - \boldsymbol{x}_j^e = \boldsymbol{A_c}(\boldsymbol{x}_j(0) - \boldsymbol{x}_j^e) + \boldsymbol{\tau}_j(0). \tag{58}$$

Substituting (58) into equation (56)

$$\boldsymbol{x}(t+1) - \boldsymbol{x}^e = \boldsymbol{\eta}(t) + \sum_{j \in S} \lambda_j (\boldsymbol{A_c}(\boldsymbol{x}_j(0) - \boldsymbol{x}_j^e) + \boldsymbol{\tau}_j(0)), \tag{59}$$

We now denote $\boldsymbol{\xi}(t)$ the aggregation of all the error terms. That is

$$\boldsymbol{\xi}(t) = \boldsymbol{\eta}(t) + \sum_{j \in S} \lambda_j (\boldsymbol{\tau}_j(0)). \tag{60}$$

With this notation

$$\boldsymbol{x}(t+1) - \boldsymbol{x}^e = \boldsymbol{\xi}(t) + \boldsymbol{A_c} \sum_{j \in S} \lambda_j (\boldsymbol{x}_j(0) - \boldsymbol{x}_j^e). \tag{61}$$

Since $\boldsymbol{x}(t) = \sum_{j \in S} \lambda_j \boldsymbol{x}_j(0)$ and taking into account (53), we obtain

$$\begin{aligned} \boldsymbol{x}(t+1) - \boldsymbol{x}^e &= \boldsymbol{\xi}(t) + \boldsymbol{A_c} \sum_{j \in S} \lambda_j (\boldsymbol{x}_j(0) - \boldsymbol{x}_j^e) \\ &= \boldsymbol{\xi}(t) + \boldsymbol{A_c}(\boldsymbol{x}(t) - \boldsymbol{x}^e) + \boldsymbol{A_c} \boldsymbol{e}_s(t). \end{aligned} \tag{62}$$

Aggregating again the error terms in $\boldsymbol{\psi}(t) = \boldsymbol{\xi}(t) + \boldsymbol{A_c}\boldsymbol{e}_s(t)$, we have

$$\boldsymbol{x}(t+1) - \boldsymbol{x}^e = \boldsymbol{\psi}(t) + \boldsymbol{A_c}(\boldsymbol{x}(t) - \boldsymbol{x}^e), \tag{63}$$

where $\boldsymbol{\psi}(t)$ is a zero mean error term. This is enough to ensure the existence of a probabilistic ultimate bound set [35]. Consider now the weighted norm $\| \boldsymbol{x}(t+1) - \boldsymbol{x}^e \|_{\boldsymbol{P}}$. The triangle inequality yields

$$\begin{aligned} \| \boldsymbol{x}(t+1) - \boldsymbol{x}^e \|_{\boldsymbol{P}} &= \| \boldsymbol{\psi}(t) + \boldsymbol{A_c}(\boldsymbol{x}(t) - \boldsymbol{x}^e) \|_{\boldsymbol{P}} \\ &\leq \| \boldsymbol{\psi}(t) \|_{\boldsymbol{P}} + \| \boldsymbol{A_c}(\boldsymbol{x}(t) - \boldsymbol{x}^e) \|_{\boldsymbol{P}}. \end{aligned} \tag{64}$$

Since $\boldsymbol{A_c}$ is assumed to be Schur stable, there is $\boldsymbol{P} > 0$ and $\gamma \in (0, 1)$ such that

$$\boldsymbol{A_c}^\top \boldsymbol{P} \boldsymbol{A_c} < \gamma \boldsymbol{P}, \tag{65}$$

Taking into account this into (64)

$$\| \boldsymbol{x}(t+1) - \boldsymbol{x}^e \|_{\boldsymbol{P}} \leq \| \boldsymbol{\psi}(t) \|_{\boldsymbol{P}} + \sqrt{\gamma} \| \boldsymbol{x}(t) - \boldsymbol{x}^e \|_{\boldsymbol{P}}. \tag{66}$$

This means that the trajectory converges to a neighbourhood of $\boldsymbol{x}^e$. □

Notice that the distance to the desired steady state $\boldsymbol{x}^e$ decreases at each sample time provided that

$$\| \boldsymbol{\psi}(t) \|_{\boldsymbol{P}} < (1 - \sqrt{\gamma}) \| \boldsymbol{x}(t) - \boldsymbol{x}^e \|_{\boldsymbol{P}}. \tag{67}$$

From here we infer that the size of the set in which $\boldsymbol{x}(t)$ is ultimately bounded can be characterised by an upper bound on $\| \boldsymbol{\psi}(t) \|_{\boldsymbol{P}}$.

5 Controller formulation

In this section, a general formulation for the proposed strategy and an implementation procedure (see Algorithm 1 in Fig. 2) are presented. Furthermore, we focus on some details of the algorithm that provide a simplification in its implementation and a relaxation on some theoretical assumptions made in Sections 3 and 4.

1. *Build* $z(t)$ *as in* (73).
2. *For each candidate* $j \in S$ *compute a suitable distance function* $d(z(t), z_j)$ *(e.g., like* (74)*).*
3. *Build* $\hat{S}$ *from the* n_c *closest candidates.*
4. *Solve problem* (71) *using the constraints defined by* $\hat{S}$.
5. *Compute and apply the control signal* $u(t)$ *using* (72) *with* $\hat{S}$.

Fig. 2 *Algorithm 1: reducing the candidates set implies that at each sampling time, the procedure given in this algorithm has to be implemented*

The objective of the proposed controller is to minimise at each sampling time the the variance of the tracking error which following the results of the previous section can be defined as

$$\sum_{j \in S} \lambda_j^2 \mathrm{E} \| \boldsymbol{e}_j(0) \|^2 . \tag{68}$$

where the $\boldsymbol{e}_j(0)$ represents the part of the error term $\boldsymbol{\psi}(t)$ related to the tracking error in the stored trajectory j. It can be difficult to obtain the expectation $\mathrm{E}\| \boldsymbol{e}_j(0) \|^2$, but if we assume an upper bound

$$\mathrm{E}\| \boldsymbol{e}_j(0) \|^2 \le \boldsymbol{\sigma}_j, \quad \forall j \in S, \tag{69}$$

then the optimisation problem to solve is to minimise

$$\sum_{j \in S} \boldsymbol{\sigma}_j^2 \lambda_j^2, \tag{70}$$

subject to the equality constraints presented in the previous section. Furthermore, if it is considered that the values for $\boldsymbol{\sigma}_j$ are all equal to an unknown value $\boldsymbol{\sigma}$, the optimisation problem can be rewritten as

$$\lambda_j^*(t) = \arg \min_{\lambda_j} \sum_{j \in S} \lambda_j^2 \tag{71a}$$

$$\text{s.t.} \sum_{j \in S} \lambda_j \boldsymbol{x}_j(0) = \boldsymbol{x}(t) \tag{71b}$$

$$\sum_{j \in S} \lambda_j \boldsymbol{x}_j(-1) = \boldsymbol{x}(t-1) \tag{71c}$$

$$\sum_{j \in S} \lambda_j \boldsymbol{u}_j(-1) = \boldsymbol{u}(t-1) \tag{71d}$$

$$\sum_{j \in S} \lambda_j \boldsymbol{y}_j(0) = \boldsymbol{y}(t) \tag{71e}$$

$$\sum_{j \in S} \lambda_j \boldsymbol{r}_j = \boldsymbol{r} \tag{71f}$$

$$\sum_{j \in S} \lambda_j = 1 \tag{71g}$$

Using the solution obtained with the previous problem, the control signal to be applied is

$$\boldsymbol{u}(t) = \sum_{j \in S} \lambda_j^*(t) \boldsymbol{u}_j(0) \tag{72}$$

where $\lambda_j^*(t)$ are obtained every sampling time from the solution of (71a) taking into account the current values of $\boldsymbol{x}(t)$, $\boldsymbol{x}(t-1)$, $\boldsymbol{u}(t-1)$ and $\boldsymbol{y}(t)$. This control law can be obtained using a simple explicit equation and will result in an offset free tracking trajectory as shown in the previous section.

5.1 Reducing the candidate set

Given a database, a set S of candidate trajectories that includes all the information available can be obtained. In general, the cardinality of this set can be very high if the database is large (note that for each trajectory a number of candidates can be obtained with different initial states along such trajectory). In this section, we propose to use, at each sampling time t, not all the candidates available, but a reduced subset denoted $\hat{S}(t)$. In particular, we propose to use only the n_c candidates closer, in a sense, to the current state of the system. The cardinality of $\hat{S}(t)$ becomes a tuning parameter that provides a tradeoff between the amount of information used, the computational burden and the estimation error due to non-linearities. In practice, there are several reasons that justify using *local* information including the high computational burden with a large database and the low information value between trajectories or data repetition. In addition, using local information reduces the estimation error produced when the proposed approach is applied to a non-linear system. Using local information is akin to carrying out a linearisation in the current state [32, 38].

In order to reduce the candidate set, a selection criteria has to be specified. We propose to use a distance function that evaluates the trajectories stored in the database with respect to the current situation taking into account not only the current state, but also the current output, the reference and the past state and input; that is, all the information used to define the optimisation problem of the proposed controller. This information is condensed in the following vectors:

$$z(t) = \begin{bmatrix} \boldsymbol{r}(t) \\ \boldsymbol{y}(t) \\ \boldsymbol{x}(t) \\ \boldsymbol{x}(t-1) \\ \boldsymbol{u}(t-1) \end{bmatrix} \quad z_j = \begin{bmatrix} \boldsymbol{r}_j \\ \boldsymbol{y}_j(0) \\ \boldsymbol{x}_j(0) \\ \boldsymbol{x}_j(-1) \\ \boldsymbol{u}_j(-1) \end{bmatrix} \tag{73}$$

with $j \in S$. At each sampling time, the n_c candidates from S that yield the lowest value of a given weighted distance function are selected. This distance function can be defined as

$$d(z(t), z_j) = \| z(t) - z_j \|_{\boldsymbol{\alpha}} , \tag{74}$$

where $\boldsymbol{\alpha}$ is the weight matrix that is tuned to normalise and prioritise each entry of the deviation vector.

Problem (71) can be posed as a quadratic programming problem subject to equality constraints:

$$\begin{aligned} \boldsymbol{\lambda}^*(t) = \arg \quad & \min_{\boldsymbol{\lambda}} \quad \boldsymbol{\lambda}^\top \boldsymbol{\lambda} \\ & \text{s.t.} \quad \boldsymbol{H}\boldsymbol{\lambda} = \boldsymbol{b}, \end{aligned} \tag{75}$$

where $\boldsymbol{\lambda}$ is a vector that includes the n_c weights, matrix $\boldsymbol{H} \in \mathbb{R}^{n_h \times n_c}$ depends on the reduced candidate set $\hat{S}$ and vector $\boldsymbol{b} \in \mathbb{R}^{n_h}$ depends on the current sample time data $z(t)$. The number of equality constraints is $n_h = 2 \cdot n_x + n_u + 2 \cdot n_y + 1$, which in general is much lower than the number of candidates of the reduced set n_c (see subsection 10.1.1. in [37]). The solution to this optimisation problem is well known and given by

$$\boldsymbol{\lambda}^*(t) = \boldsymbol{H}^\top (\boldsymbol{H}\boldsymbol{H}^\top)^{-1} \boldsymbol{b} . \tag{76}$$

Note that this solution has to be calculated at each sampling time, because matrix $\boldsymbol{H}$ changes with the candidates selected and vector b depends on the current measurements. Note however that the most time consuming calculation is the inversion of matrix $\boldsymbol{H}\boldsymbol{H}^\top$, whose dimension is n_h. This implies that the proposed procedure avoids the use of iterative optimisation algorithms and can be implemented on a wide range of applications. In the next section, we apply this procedure to a scaled laboratory process with fast dynamics.

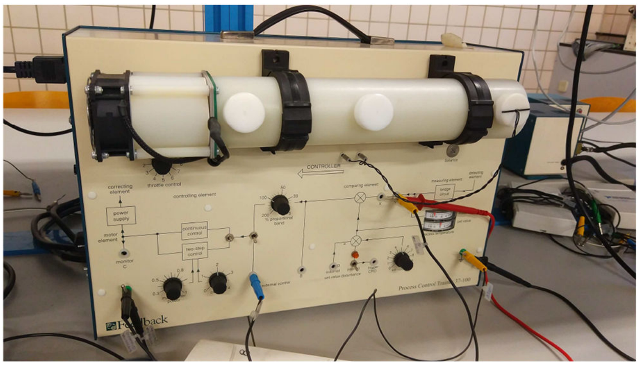

Fig. 3 *Feedback Process Trainer 37-100 unit used in this work*

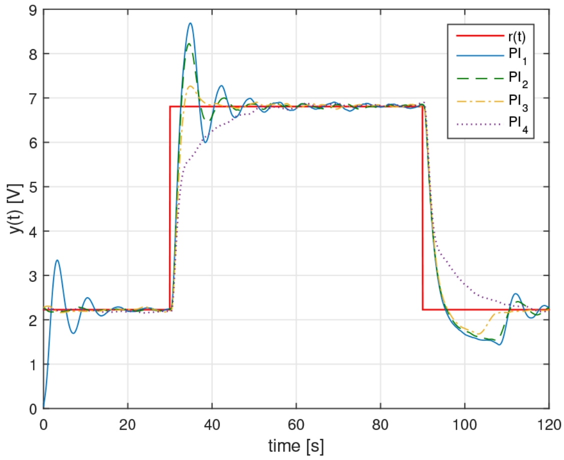

Fig. 4 *Tracking experiments in closed loop with the PI controllers that generate the database*

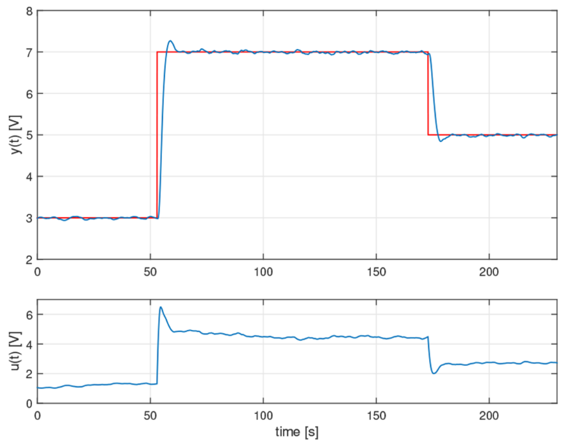

Fig. 5 *Tracking test results on the Feedback Process Trainer 37-100*

6 Application to the feedback process trainer 37-100

The proposed controller has been tested on the Feedback Process Trainer 37-100 (see Fig. 3), a renewed version of the Feedback PT-326 [34]. In this equipment, an axial fan is used to circulate air through a heating element inside a propylene tube. The heating element can be controlled using a voltage input and the air temperature is measured at the end of the tube by a bead thermistor. The system exhibits air and tube thermal time constants. The dynamic characteristics of the system can be changed by manually changing the fan speed using a potentiometer. The controlled variable in these experiments is the voltage output of the bead thermistor, whereas the manipulated variable is the voltage input that controls the heating element.

Although an scaled laboratory process, this equipment is a challenging test bed for data driven algorithms or other complex control methods because of its fast dynamics, that require sampling times of a few hundredths of a second. A sampling time of 0.07 s has been used through all the experiments shown in this section. The delay is neglected because it is lower than the sample time, thus no delay compensation has been taken into account.

Table 1 Parameters for PI controllers of the database

	PI_1	PI_2	PI_3	PI_4
K_p	1	1	1	1
K_i	1.5	1.07	0.64	0.21

Following the nomenclature, the input u is the voltage that controls the heating element, that is $u(t) = V(t) \in [0, 10]$ V, and the output y is the temperature measured by the sensor represented in a voltage, that is $y(t) = V_T(t) \in [0, 10]$ V. As it is well known that the process control trainer can be characterised by a first-order model, in this experiments the state is equal to the output, so $x(t) = y(t)$.

A total of 300 8 min trajectories have been generated with the fan speed potentiometer set to 50%. Each trajectory is defined by a random initial set point and a step set point change of random amplitude, computed in a way such that the initial and final set point values differ between 1 and 8 V. For each trajectory, four different PI controllers have been tested (each one for 2 min, changing from the initial set point value to the step value and back again to the initial value). Table 1 shows the parameters for each PI controller while Fig. 4 shows the behaviour of the transient response for each PI controller with four different closed-loop tracking experiments. The database comprises a total of 300 2 min closed-loop offset free trajectories which results in 51400 different candidates in S. It is noteworthy that the database took 40 h to be generated, thus the ambient temperature changed quite a bit during the morning and night hours. This implies that the process dynamics are not constant through the database leading to different perturbations mean values for each trajectory.

Following the procedure presented in Algorithm 1 (Fig. 2), the distance function (74), with $\boldsymbol{\alpha} = I$ has been used to select the $n_c = 6000$ nearest points to the current state in the database. It is noteworthy that the solution of (71) and the control law (72) can be computed within the sampling time of 0.07 s in Matlab with a Intel Core-i3 running Windows.

Fig. 5 shows the results of a set point tracking experiment with two reference changes using the proposed approach. The controller achieves offset free tracking in each set point value (plotted in red), despite the obvious noise and disturbances. It can be seen how the controller adjusts continually, in a clear trend, the control effort to keep the controlled variable near the set point. The reason of this trend is the heating of the propylene tube, much slower than the heating of the air, but nonetheless able to affect the controlled variable.

To demonstrate the disturbance rejection properties of the controller, three different experiments have been considered. First, a constant error of amplitude 2 V is added artificially to the temperature measure after 3 min. Fig. 6 shows the trajectory of the measured temperature and how it converges again to the reference value because the proposed controller successfully rejects the disturbance. Note that the input has to modify its steady-state value to compensate for the effect of the additive disturbance.

Second, the fan speed potentiometer has been increased from 50%, the value used to generate the database trajectories, to 80%. As a result of the increased fan speed, the temperature drops and the controller is forced to raise the voltage applied to the heater. After the disturbance is rejected, the fan speed is changed back to its previous value, which is again another disturbance that it is also effectively rejected. Fig. 7 shows the experimental results. It is noteworthy that the changes induced in the system by increasing the fan speed are more severe than the additive measurement disturbance included in the previous simulation. Despite this disturbance, the non-linearity of the system and the variations in the ambient temperature, the controller is able to track set point changes and reject disturbances.

Finally, an even more difficult case is shown in Fig. 8 where the fan speed was reduced from 50 to 30%. This case is more difficult

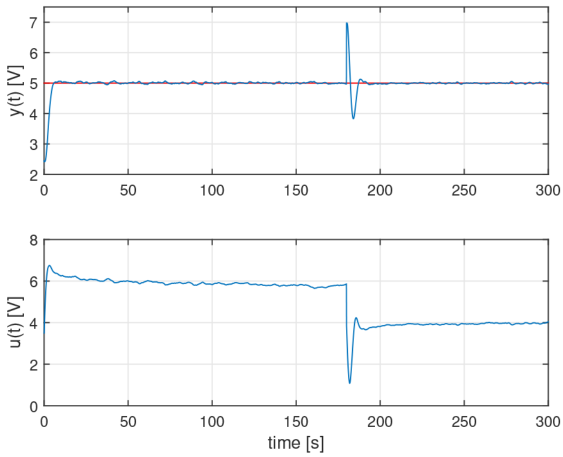

Fig. 6 *Disturbance rejection test with the Feedback Process Trainer 37-100: Artificial additive disturbance*

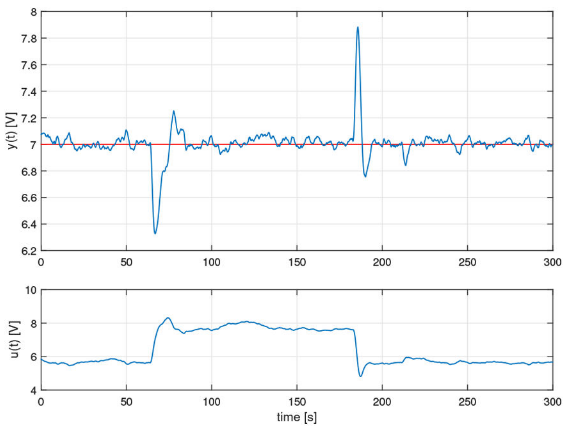

Fig. 7 *Disturbance rejection test the Feedback Process Trainer 37-100: Fan speed increased*

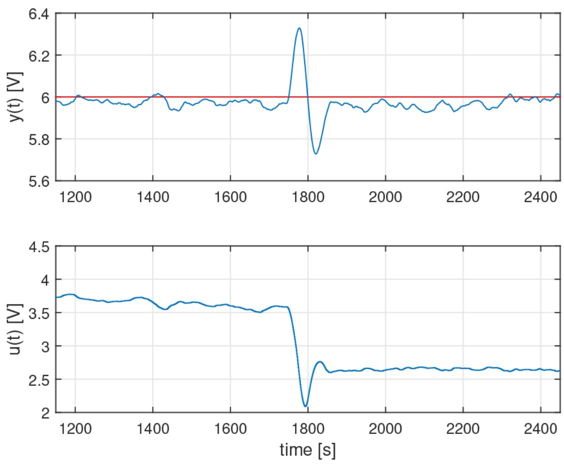

Fig. 8 *Disturbance rejection test the Feedback Process Trainer 37-100: Fan speed decreased*

than the previous one because in addition to the changes in the process dynamics, the dead-time is increased. Nevertheless, as shown in Fig. 8, the controller is able to track the set-point with minimal steady-state error while compensating the slow drifts in the temperature.

In order to study the effect of the number of candidates n_c on the closed-loop performance, we will compare the tracking error of a set of controllers with n_c taking values $n_c = \{1000, 1500, 2000, 2500, 3000\}$. For each controller, 15 closed-loop tracking experiments with length 60 s (857 samples) have been carried out with different reference values, in particular, five experiments with reference $r = 3.5$, five with reference $r = 5$

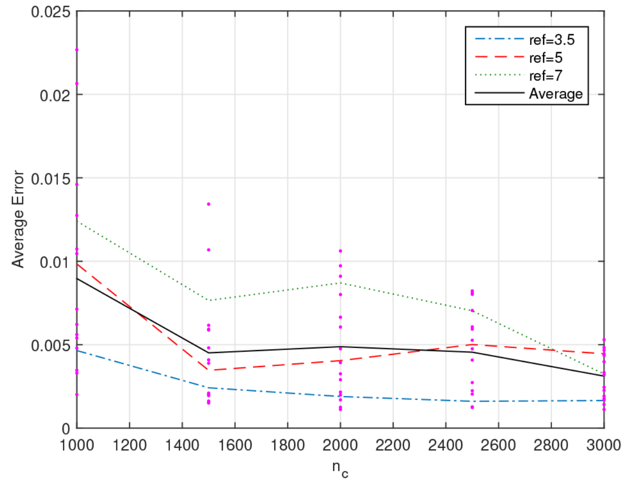

Fig. 9 *Average steady-state error against the number of candidates n_c for different value of the reference*

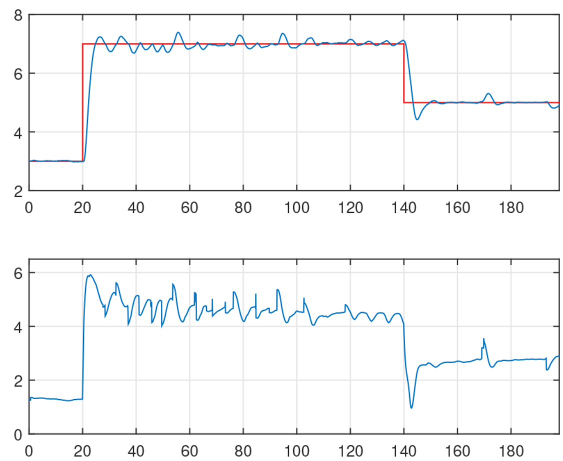

Fig. 10 *Tracking experiment with a low number of candidates ($n_c = 50$)*

and another five with reference $r = 7$. For each experiment the mean value of the squared error is computed for 172 samples once the closed-loop system has converged to its corresponding reference.

Fig. 9 shows the steady-state error for the 75 experiments as magenta dots. Furthermore, the average of the steady-state error for the five closed-loop experiments with the same constant reference signal and n_c is computed. The blue dashed-dotted line shows this result for reference $r = 3.5$, the red dashed line represents the average of the experiments with $r = 5$ and the green dotted one is with $r = 7$. The average of the 15 experiments with the same n_c is represented with the black solid line. We can observe that the steady-state error for all the experiments is always upper-bounded by $2.5 \cdot 10^{-2}$. Moreover, the steady-state error decreases when n_c is increased which is the expected behaviour.

It is important to remark that reducing too much the number of candidates has a negative impact on the performance of the controller. In those cases, the controller is forced to extrapolate, which results in worse control. Fig. 10 shows a tracking experiment where only the 50 closest candidates are considered. In this experiment, the average tracking error is quite large (almost 0.4 V with a standard deviation of 0.36), but as shown in Table 2, these numbers drop as expected if the experiment is repeated with an increasingly higher number of candidates (note that these experiments are different from that of Fig. 9 as they contain several set point changes, hence the greater average error and standard deviation).

Finally, it is demonstrated that the proposed strategy performance is strongly dependent on the database trajectories considered. Fig. 11 shows the output of the data-based controller in closed-loop using two different databases for the same reference

Table 2 Average tracking error and standard deviation in varying set-point tracking experiment for low numbers of candidates

n_c	50	100	500	1500
$\bar{e}$	0,3998	0,1423	0,0889	0,0872
σ	0,3686	0,1487	0,1434	0,1480

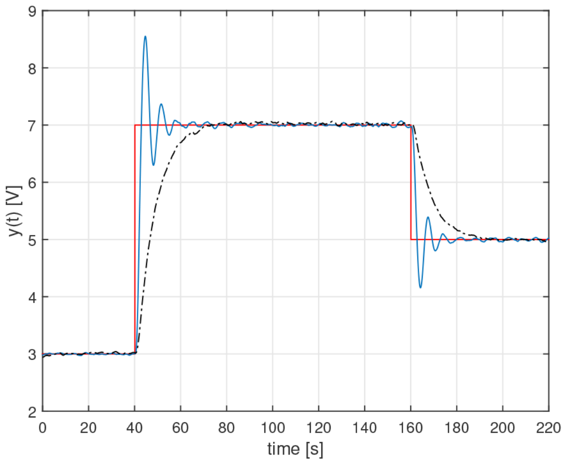

Fig. 11 *Comparison of the proposed controller with a database using only a PI controller with $K_p = 1$ and $K_i = 1.5$ (solid blue) and with a database using only a PI controller with $K_p = 1$ and $K_i = 0.21$ (dashed-dotted black). Red dashed line represents the reference signal*

signal (dashed red). In the first case, a more oscillating PI controller with parameters $K_\text{p} = 1$ and $K_\text{i} = 1.5$ is used to obtain the database. On the other hand, a PI controller less aggressive with parameters $K_\text{p} = 1$ and $K_\text{i} = 0.21$ generates the second database. Note that the closed-loop trajectory of the data-based controller reaches the reference independently of the database used, however its behaviour in the transient response is inherited from the database trajectories.

7 Conclusion

This paper has presented an efficient strategy to solve a tracking control problem that it is tailored for systems with an unknown model function. It solves the problem using past closed-loop offset free trajectories and control actions stored in a database while minimising the variance of the tracking error. Moreover, the optimisation problem can be solved efficiently which fits with fast dynamic process control problems. The proposed strategy has obtained offset free tracking with a real scaled laboratory process with fast dynamics and these results show its effectiveness, even with non-linear systems.

The proposed strategy will be improved in future works taking into account issues such as periodic changing set-points, large process delays, extension to non-linear systems (either general or with some predefined structure as in Volterra or Hammerstein–Wiener models), online maintenance of the database (in which overparameterisation versus data reusability should have to be considered, specially when few data are available) and how to design appropriate open-loop tests to be included in the database instead of or mixed with proper closed-loop tests.

8 Acknowledgments

This work was supported by the MINECO-Spain and FEDER Funds under project DPI2016-76493-C3-1-R and by the University of Seville under grant 2014/425.

9 References

[1] Maeder, U., Morari, M.: ‘Offset-free reference tracking with model predictive control’, *Automatica*, 2010, **46**, (9), pp. 1469–1476

[2] Ikeda, M., Fujisaki, Y., Hayashi, N.: ‘A model-less algorithm for tracking control based on input-output data’, *Nonlinear Anal., Theory, Methods Appl.*, 2001, **47**, (3), pp. 1953–1960. Proceedings of the Third World Congress of Nonlinear Analysts

[3] Kergus, P., Poussot-Vassal, C., Demourant, F., *et al.*: ‘Frequency-domain data-driven control design in the loewner framework’, *IFAC-PapersOnLine*, 2017, **50**, (1), pp. 2095–2100. 20th IFAC World Congress

[4] Maupong, T.M., Rapisarda, P.: ‘Data-driven control: a behavioral approach’, *Syst. Control Lett.*, 2017, **101**, pp. 37–43

[5] Stanley, G.: ‘Big data approximating control (BDAC) – a new model-free estimation and control paradigm based on pattern matching and approximation’, *J. Process Control*, 2018, **67**, pp. 141–159

[6] Campi, M.C., Lecchini, A., Savaresi, S.M.: ‘Virtual reference feedback tuning: a direct method for the design of feedback controllers’, *Automatica*, 2002, **38**, (8), pp. 1337–1346

[7] Hjalmarsson, H., Gevers, M., Gunnarsson, S., *et al.*: ‘Iterative feedback tuning: theory and applications’, *IEEE Control Syst. Mag.*, 1998, **18**, (4), pp. 26–41

[8] Van Heusden, K., Karimi, A., Bonvin, D.: ‘Data-driven model reference control with asymptotically guaranteed stability’, *Int. J. Adapt. Control Signal Process.*, 2011, **25**, (4), pp. 331–351

[9] Piga, D., Formentin, S., Bemporad, A.: ‘Direct data-driven control of constrained systems’, *IEEE Trans. Control Syst. Technol.*, 2018, **26**, (4), pp. 1422–1429

[10] Tanaskovic, M., Fagiano, L., Novara, C., *et al.*: ‘Data-driven control of nonlinear systems: an on-line direct approach’, *Automatica*, 2017, **75**, pp. 1–10

[11] Chi, R., Liu, X., Zhang, R., *et al.*: ‘Constrained data-driven optimal iterative learning control’, *J. Process Control*, 2017, **55**, pp. 10–29

[12] Radac, M.B., Precup, R.E., Roman, R.C.: ‘Data-driven model reference control of mimo vertical tank systems with model-free VRFT and Q-learning’, *ISA Trans.*, 2018, **73**, pp. 227–238

[13] Zhang, Y., Ding, S.X., Yang, Y., *et al.*: ‘Data-driven design of two-degree-of-freedom controllers using reinforcement learning techniques’, *IET Control Theory Appl.*, 2015, **9**, (7), pp. 1011–1021

[14] Canale, M., Fagiano, L., Signorile, M.C.: ‘Nonlinear model predictive control from data: a set membership approach’, *Int. J. Robust Nonlinear Control*, 2014, **24**, (1), pp. 123–139

[15] Fagiano, L., Novara, C.: ‘Learning a nonlinear controller from data: theory, computation, and experimental results’, *IEEE Trans. Autom. Control*, 2016, **61**, (7), pp. 1854–1868

[16] Kadali, R., Huang, B., Rossiter, A.: ‘A data driven subspace approach to predictive controller design’, *Control Eng. Pract.*, 2003, **11**, (3), pp. 261–278

[17] Novara, C., Milanese, M.: ‘Control of mimo nonlinear systems: a data-driven model inversion approach’, *Automatica*, 2019, **101**, pp. 417–430

[18] de Rozario, R., Oomen, T.: ‘Data-driven iterative inversion-based control: achieving robustness through nonlinear learning’, *Automatica*, 2019, **107**, pp. 342–352

[19] Bu, X., Wang, Q., Hou, Z., *et al.*: ‘Data driven control for a class of nonlinear systems with output saturation’, *ISA Trans.*, 2018, **81**, pp. 1–7

[20] Li, Y., Yang, C., Hou, Z., *et al.*: ‘Data-driven approximate Q-learning stabilization with optimality error bound analysis’, *Automatica*, 2019, **103**, pp. 435–442

[21] Bertsekas, D.P.: ‘*Reinforcement learning and optimal control*’ (Athena Scientific, Amsterdam, The Netherlands, 2019)

[22] Tang, W., Daoutidis, P.: ‘Distributed adaptive dynamic programming for data-driven optimal control’, *Syst. Control Lett.*, 2018, **120**, pp. 36–43

[23] Roll, J., Nazin, A., Ljung, L: ‘A non-asymptotic approach to local modelling’. 2002 Proc. of the 41st IEEE Conf. on Decision and Control, Las Vegas, NV, USA, 2002, vol. 1, pp. 638–643

[24] Roll, J., Nazin, A., Ljung, L: ‘Local modelling with a priori known bounds using direct weight optimization’. 2003 European Control Conf. (ECC), Cambridge, UK, 2003, pp. 2138–2143

[25] Roll, J., Nazin, A., Ljung, L.: ‘Nonlinear system identification via direct weight optimization’, *Automatica*, 2005, **41**, (3), pp. 475–490

[26] Atkeson, C.G., Moore, A.W., Schaal, S.: ‘Locally weighted learning’, *Artif. Intell. Rev.*, 1997, **11**, (1), pp. 11–73

[27] Atkeson, C.G., Moore, A.W., Schaal, S.: ‘Locally weighted learning for control’, *Artif. Intell. Rev.*, 1997, **11**, (1), pp. 75–113

[28] Gao, T., Yin, S., Gao, H., Yang, X., Qiu, J., Kaynak, O.: ‘A locally weighted project regression approach-aided nonlinear constrained tracking control’, *IEEE Trans. Neural Netw. Learn. Syst.*, 2018, **29**, (12), pp. 5870–5879

[29] Hou, Z., Liu, S., Yin, C.: ‘Local learning-based model-free adaptive predictive control for adjustment of oxygen concentration in syngas manufacturing industry’, *IET Control Theory Applic.*, 2016, **10**, (12), pp. 1384–1394

[30] Radac, M.B., Precup, R.E.: ‘Three-level hierarchical model-free learning approach to trajectory tracking control’, *Eng. Appl. Artif. Intell.*, 2016, **55**, pp. 103–118

[31] Salvador, J.R., Ramirez, D.R., Alamo, T., *et al.*: ‘Data driven control: an offset free approach’. Proc. of the European Control Conf., Napoli, Italy, 2019

[32] Salvador, J.R., Muñoz de la Peña, D., Ramirez, D.R., Alamo, T: ‘Historian data based predictive control of a water distribution network’. Proc. of the European Control Conf., Limassol, Cyprus, 2018

[33] Salvador, J.R., Muñoz de la Peña, D., Alamo, T., *et al.*: ‘Data-based predictive control via direct weight optimization’. 6th IFAC Conf. on Nonlinear Model Predictive Control, Madison, Wisconsin, USA, 2018

[34] Ljung, L.: ‘*System identification, theory for the user*’ (Prentice-Hall, Upper Saddle River, NJ, USA, 1987)

[35] Kofman, E., De Doná, J.A., Seron, M.M.: ‘Probabilistic set invariance and ultimate boundedness’, *Automatica*, 2012, **48**, (10), pp. 2670–2676

[36] Blanchini, F., Miani, S.: '*Set-theoretic methods in control*' (Springer, New York, USA, 2008)
[37] Boyd, S., Vandenberghe, L.: '*Convex optimization*' (Cambridge University Press, Cambridge, UK, 2004)
[38] Bravo, J.M., Alamo, T., Vasallo, M., *et al.*: 'A general framework for predictors based on bounding techniques and local approximation', *IEEE Trans. Autom. Control*, 2017, **62**, (7), pp. 3430–3435