Journal of Process Control 132 (2023) 103109

Contents lists available at ScienceDirect

Journal of Process Control

journal homepage: www.elsevier.com/locate/jprocont

Robust data-based predictive control of systems with parametric uncertainties: Paving the way for cooperative learning

Eva Masero [a,b,*], José M. Maestre [a], José R. Salvador [c], Daniel R. Ramirez [a], Quanyan Zhu [d]

[a] Department of Ingeniería de Sistemas y Automática, Universidad de Sevilla, Seville 41092, Spain
[b] Department of Electronics, Information and Bioengineering, Politecnico di Milano, Milan 20133, Italy
[c] Department of Ingeniería, Universidad Loyola Andalucía, Dos Hermanas, Seville, 41704, Spain
[d] Department of Electrical and Computer Engineering, New York University, Brooklyn, NY 11201, United States

ARTICLE INFO

Keywords:
Predictive control
Data-driven control
Tube-based control
Robustness
Cooperative learning

ABSTRACT

This article combines data and tube-based predictive control to deal with systems with bounded parametric uncertainty. This integration generates robustly feasible control sequences that can also be exploited in cooperative scenarios where controllers learn from each other's data. In particular, the approach is based on a database that contains information from previous executions of the same and other controllers handling similar systems. By the combination of feasible histories plus an auxiliary control law that deals with bounded uncertainties, which only needs to be stabilizing for at least one of the system realizations within the uncertainty set, this scheme provides a finite-horizon predictive controller that guarantees exponential stability and robust constraint satisfaction. The validity and benefits of the proposed scheme are shown in case studies with linear and non-linear dynamics.

1. Introduction

Multiple applications of predictive model control (MPC) in industry, *e.g.*, [1–3], have stimulated renewed interest in approaches combining data-based and learning methods with predictive control schemes [4]. For example, those based on neural networks [5], adaptive models [6], scenarios [7], and the extension of terminal regions [8], to name a few examples. In the recent literature, we can find MPC schemes for repetitive tasks that learn from previous executions to improve its performance. For instance, in [9], the theoretical properties are derived from the use of safe sets, which can be defined as regions of the state space where there exists a control law that guarantees constraint satisfaction for all successive time steps. This topic is also addressed in [10,11], where methods to expand regions of safe states are proposed. Learning is also used to infer the plant model, as in [12], where experimental data on system inputs and outputs are used to feed a non-parametric machine learning method. Multi-step predictors learned from the data for robust MPC are used in [13].

Another strategy is to use a model-free approach such as [14], where the control signal is calculated using combinations of closed-loop trajectories stored from the past in such a way that the combined performance cost is minimized. This approach is related to direct weight optimization (DWO) techniques [15] and also to kriging methods [16, 17]. In general, both techniques rely on expressing a given datum as a linear combination of data stored in the database, while some measurement of the combined weights is minimized. The DWO approach has been used in the context of identification in [18], and for data-based control purposes in [19]. On the other hand, kriging methods have been mainly used as an interpolation technique, first in geostatistics [20], but also in some control-related applications, *e.g.*, [21] uses a basic kriged black box prediction model to design a non-linear MPC controller, and [22] introduces a new type of state space modeling technique based on kriging weights. The strategy presented in [14] was extended in [23] to obtain offset-free control under some limiting conditions. However, these strategies are highly dependent on the quality of the data and lack the robustness to track setpoints not included in the database.

To address this challenge, several robust data-driven MPC techniques have emerged. For example, for the control of Type 1 diabetes, [24] combines data-driven learning with min–max MPC to obtain patterns of disturbance sets from historical data and calculate insulin infusions in the worst-case disturbances. Moreover, [25] proposes a robust data-driven MPC method that uses Hankel matrices to predict the system behavior by its input–output trajectories and includes a slack variable to relax the constraints and account for noise, *i.e.*, the differences between the online measurements and the data used for prediction. This method is extended in [26] using constraint tightening to enforce closed-loop output constraint satisfaction. Another

* Corresponding author at: Department of Ingeniería de Sistemas y Automática, Universidad de Sevilla, Seville 41092, Spain.
E-mail address: evamasero@us.es (E. Masero).

https://doi.org/10.1016/j.jprocont.2023.103109
Received 21 November 2022; Received in revised form 21 September 2023; Accepted 10 October 2023
Available online 20 October 2023

way of handling disturbances is to rely on stochastic approaches. For instance, [27] proposes a stochastic MPC based on data driven for linear systems subject to unknown additive disturbances. It is also considered a distributionally robust optimization with a Wasserstein ambiguity set based on a dataset consisting of disturbance trajectories. Then, a scenario-based tube MPC is employed to reduce the online complexity of the classical scenario-based stochastic MPC.

In this article, we deal with the reutilization of past historical information following the method presented in [14], where data-based predictive control is proposed to manage linear systems through a convex combination of previously applied trajectories assuming that successive executions of the system remain constant and are disturbance-free. The key contribution of our work is to extend [14] to overcome its lack of robustness regarding constraint satisfaction, *e.g.*, due to process noise or parametric uncertainties. To this end, we provide a set of conditions that must be met by the trajectories employed by the data-based controller to ensure robust constraint satisfaction. Here we apply a strategy analogous to tube-based MPC [28], but it is a data-based predictive controller that provides us with the *nominal* disturbance-free trajectory for the current system. Therefore, the control problem is twofold: (i) to calculate the nominal control inputs by a convex combination of previously trajectories collected in the database, and (ii) to apply a stabilizing feedback controller to reject the differences between the real and nominal dynamics. Regarding the former, it is employed a direct weight optimization based on [14], an idea aligned with the rationale of kriging methods [21,22] and recent data-driven MPC approaches [26,29], which rely on the *fundamental lemma* by Willems et al. [29, Lemma 2]. The lemma establishes that a discrete-time LTI system can be fully characterized by its trajectories as long as the input is persistently exciting [30]. The main differences with regard to the previously mentioned articles are that the newly proposed method guarantees recursive feasibility and robust exponential stability, and can even manage one system with trajectories generated from different instances of the system being controlled (*e.g.*, a set of controllers sharing a cloud-based database), leading to a *cooperative learning* approach. To learn from each other's trajectories, controllers need trajectories generated in a wide range of conditions and system realizations so that they are representative of the behavior of the system; also, trajectories should be stored in a format that is easily accessible and can be processed efficiently by the cooperative learning algorithm.

The idea of using data from one or more systems to design a stabilizing controller for another system is also addressed in [31] in a data-based manner for the particular case of transferring a stabilizing control policy from a source system to a target system. The commonalities of our method and that of [31] are a prior knowledge of the maximum distance between two linear systems, significant data (with noise) from the source system, and the design of a stabilizing controller based on linear matrix inequalities (LMI) for the target system. However, our twist is the integration of data-driven control and predictive control to inherit the optimal and robustness properties of tube-based MPC. Our method also provides robust exponential stability and robust constraint satisfaction, which is a major issue in reusing past historical data. Another approach related to our proposal is the interpolation-based MPC [32], which relies on the convex combination of feedback control laws for a set of uncertain systems. Although this idea has some connections with the current work, note that our trajectories may have been generated by different types of controllers. Furthermore, robust constraint satisfaction is attained in [32] leveraging the well-known approach of Kothare et al., which is known to be conservative. The referred article also points out some significant differences with standard predictive control approaches.

To sum up, the novelty of our work is the combination of a data-driven MPC with a tube-based approach to address systems with polytopic uncertainty. This integration presents a novel contribution that can be leveraged to have multiple controllers learning in a cooperative fashion. It also allows for the design of controllers that are both data-driven and robust, filling a gap in the existing literature.

Finally, a preliminary version of the proposed strategy was accepted for presentation [33]. There are significant differences between the current article and the conference version: (i) the problem formulation is now less conservative, with fewer constraints required; (ii) the optimization problem has now more degrees of freedom, improving performance; (iii) the current article does include theoretical analysis of the recursive feasibility and stability; (vi) new simulations and an additional example are also included in the current version.

Index of contents: Section 2 presents the problem setting and the proposed control strategy. Sections 3 and 4 detail the design and implementation of the nominal and auxiliary controllers, respectively. Section 5 analyzes the recursive feasibility and stability guarantees. Sections 6 and 7 present the case studies. Section 8 ends the paper with concluding remarks.

Notation: Sets $\mathbb{R}$ and $\mathbb{N}$ refer to real numbers and integers, respectively. The distance between vectors $x, y \in \mathbb{R}^n$ is $\mathrm{d}(x,y) = \|x-y\| = \sqrt{(x_1-y_1)^2+\cdots+(x_n-y_n)^2}$, while the square of the distance is denoted as $\|x-y\|^2$. The image of a set $\mathcal{X} \subseteq \mathbb{R}^n$ under a linear mapping $A : \mathbb{R}^n \mapsto \mathbb{R}^m$ is given by $A\mathcal{X} \triangleq \{Ax : x \in \mathcal{X}\}$. For sets $\mathcal{X}, \mathcal{Y} \subseteq \mathbb{R}^n$, the Minkowski sum is $\mathcal{X} \oplus \mathcal{Y} \triangleq \{x+y : x \in \mathcal{X},\ y \in \mathcal{Y}\}$, and the Pontryagin difference is $\mathcal{X} \ominus \mathcal{Y} \triangleq \{z \in \mathbb{R}^n : \mathcal{Y} \oplus \{z\} \subseteq \mathcal{X}\}$ for $\mathcal{Y} \subseteq \mathcal{X}$. The matrix $A \succ 0$ ($A \succeq 0$) is positive definite (semi-definite). The identity matrix is denoted as I. The set $\Omega \in \mathbb{R}^n$ is a robust positively invariant (RPI) set for system $x^+ = f(x,w)$ with constraints $\mathcal{X}$ and $\mathcal{W}$ if $\forall x \in \Omega \subset \mathcal{X}$ and $\forall w \in \mathcal{W}$, its evolution satisfies $x^+ \in \Omega$.

2. Problem formulation

We consider that the system dynamics are described by the following discrete-time linear time-invariant (LTI) model with an unknown vector of parameters $\theta \in \Theta \subseteq \mathbb{R}^{n_\theta}$:

$$x^+ = A(\theta)\,x + B(\theta)\,u, \tag{1}$$

where $x,\ x^+ \in \mathbb{R}^{n_x}$ and $u \in \mathbb{R}^{n_u}$ are, respectively, the state, the successor state, and the input of the system. The matrices $A(\theta) \in \mathbb{R}^{n_x \times n_x}$ and $B(\theta) \in \mathbb{R}^{n_x \times n_u}$ are the state-transition and input-to-state matrices, which depend on the realization of parametric uncertainty θ.

Assumption 1. All systems $(A(\theta), B(\theta))$ are controllable.

Assumption 2. The differences between the system realizations are bounded by polytopic sets ΔA_θ and ΔB_θ that account for possible parameter variations.

Two different realizations of θ, say $\theta_i, \theta_j \in \Theta$, lead to two distinct system realizations (A_i, B_i) and (A_j, B_j), with

$$A_i - A_j \in \Delta A_\theta, \quad B_i - B_j \in \Delta B_\theta. \tag{2}$$

The parametric uncertainty of the system can be reinterpreted as an unknown disturbance w. In particular, the dynamic evolution of the system realizations i and j can be related as follows by (2)

$$\begin{aligned} x^+ &= A_i\,x + B_i\,u \\ &\in (A_j + \Delta A_\theta)\,\mathcal{X} + (B_j + \Delta B_\theta)\,\mathcal{U} \\ &= A_j\,x + B_j\,u + w, \end{aligned} \tag{3}$$

where the state and inputs are subject to constraints

$$x \in \mathcal{X}, \quad u \in \mathcal{U}, \tag{4}$$

with the disturbance term w bounded by

$$w \in \mathcal{W} \triangleq \Delta A_\theta\,\mathcal{X} \oplus \Delta B_\theta\,\mathcal{U}. \tag{5}$$

Note that, in case of need, $\mathcal{W}$ can be enlarged to deal with additional sources of uncertainty.

Assumption 3. Sets $\mathcal{X} \subseteq \mathbb{R}^{n_x}$, $\mathcal{U} \subseteq \mathbb{R}^{n_u}$, and $\mathcal{W} \subseteq \mathbb{R}^{n_w}$ are polytopic convex sets that contain the origin in their interiors.

Table 1
Database structure.

Traj. ID	k	$x_t(k)$	$u_t(k)$
1	1	$x_t(1)$	$u_t(1)$
⋮	⋮	⋮	⋮
1	k^1_{end}	$x_t(k^1_{\text{end}})$	$u_t(k^1_{\text{end}})$
2	1	$x_t(1)$	$u_t(1)$
⋮	⋮	⋮	⋮
2	k^2_{end}	$x_t(k^2_{\text{end}})$	$u_t(k^2_{\text{end}})$
⋮	⋮	⋮	⋮

2.1. Database structure and control goal

Any controller has access to a database that contains a set of T trajectories generated by different instances of system (1). Each trajectory is identified using indices within set $\mathcal{T} \triangleq \{1,\dots,T\}$ and is assumed to be feasible for the system realization that generated it, but does not need to be feasible for other system realizations. In particular, a trajectory $t \in \mathcal{T}$ is composed of a sequence of samples $\big(x_t(k), u_t(k)\big)$, for $k \in \mathcal{K}_t \triangleq \{1,\dots,k^t_{\text{end}}\}$, that satisfy (1) and (4) for the particular system realization. See Table 1 for an example regarding the structure of the considered database.

Remark 1. Although we consider the general case of the database being cooperatively updated by multiple realizations of system (1), it is worth noticing that there is no difference if the database is generated by collecting data from a single plant that uses different parametric configurations.

Assumption 4. At some time instant k^t_{end}, the sequences of all the trajectories $t \in \mathcal{T}$ finish at a bounded distance from the origin smaller than a threshold $\varepsilon \in \mathbb{R}^+$.

Remark 2. The threshold ε defines a ball around the origin, and must be small enough to satisfy basic conditions such as constraints. It can be empirically chosen as a function of the amplitude of noise and random disturbances that affect the tracking errors. The larger ε, the greater the uncertainty of the system's state relative to the origin.

The control objective is to regulate the system to the origin while minimizing a global infinite-horizon cost:

$$J^\infty = \sum_{n=0}^{\infty} \ell\big(x(n), u(n)\big), \tag{6}$$

where $\ell(\cdot,\cdot)$ is the stage cost, which is defined by positive definite weighting matrices $Q \in \mathbb{R}^{n_x\times n_x}$ and $R \in \mathbb{R}^{n_u\times n_u}$:

$$\ell\big(x(n), u(n)\big) = x(n+1)^\top Q\, x(n+1) + u(n)^\top R\, u(n). \tag{7}$$

For the sake of implementability, it is desirable to consider a finite horizon N, so the cost function (6) is replaced by:

$$J^N = \sum_{n=0}^{N-1} \ell\big(x(n), u(n)\big) + f\big(x(N)\big), \tag{8}$$

where $f(\cdot) = x(N)^\top P\, x(N)$ with $P \succ 0$ is a terminal cost function. In this regard, let us define the N-length partial sequences of $\mathbf{x}$ and $\mathbf{u}$, which are obtained by taking N consecutive states and inputs from trajectory $t \in \mathcal{T}$:

$$p \triangleq \begin{cases} \mathbf{x} = \{x_t(k), x_t(k+1), \dots, x_t(k+N)\} \\ \mathbf{u} = \{u_t(k), u_t(k+1), \dots, u_t(k+N-1)\} \end{cases},$$

and the performance of this sequence measured by (8) can be denoted as $J^N(\mathbf{x},\mathbf{u})$.

2.2. Dual control law

We control the system realization (A_i, B_i) using a data-based controller that combines the trajectories contained in $\mathcal{T}$. A robust feedback controller is also employed to deal with disturbance differences w due to uncertain system realizations. Therefore, the control law becomes

$$u(x) = v(x) + v^e(x,z), \tag{9}$$

where $v(x)$ corresponds to the first element of a sequence of control actions generated by the data-based law, and $v^e(x,z)$ rejects the differences between the system state x and the nominal state z, which is obtained from the database and follows disturbance-free dynamics: $z^+ = A_j\, z + B_j\, v$ (recall (3)). A block diagram of the proposed dual control law is shown in Fig. 1. Sections 3 and 4 detail, respectively, how $v(x)$ and $v^e(x,z)$ are obtained.

3. Data-based control law $v(x)$

Let us consider a particular system realization (A_i, B_i). In the most restrictive case, there is no previous information in the database about previous executions of this system, *i.e.*, we need to rely on information from other system realizations.

3.1. Trajectory robustness check

Since the sequences of any other system realization (A_j, B_j), with $j \neq i$ might not be feasible for (A_i, B_i), we need to guarantee their robust feasibility.

Therefore, any candidate trajectory $\{x_t(k), u_t(k)\}$, with $k \in \mathcal{K}_t = \{1,\dots,k^t_{\text{end}}\}$, must have enough margin respect to the constraints to accommodate uncertainties given by the closed-loop dynamics of the errors due to system discrepancies. One way to do this is to check the following conditions:

$$\begin{aligned} x_t(k) &\in \mathcal{X} \ominus \Omega_x, \\ u_t(k) &\in \mathcal{U} \ominus \Omega_u, \end{aligned} \tag{10}$$

for $k \in \mathcal{K}_t$, where Ω_x and Ω_u are given sets that define the desired margin with respect to the state and input constraints, respectively. Details on the calculation of Ω_x and Ω_u are given in Section 4.

By checking (10), we can find the set of robustly feasible trajectories in the database.[1] Let $\mathcal{F} \subseteq \mathcal{T}$ contain the IDs of these feasible trajectories.

Assumption 5. The set of robustly feasible trajectories $\mathcal{F}$ is assumed to be nonempty.

3.2. Optimization problem

Conventional tube-based MPC employs a nominal model with tighter constraints. If we considered a database that only contains information about the nominal system, we could define, for nominal dynamics $z^+ = A_j\, z + B_j\, v$, an optimal control problem $\mathbb{P}_N(z)$ as [14]:

$$\begin{aligned} J_N^o(z) = \min_{\Lambda\in\mathbb{R}^M} \quad & \sum_{m\in\bar{\mathcal{T}}} \lambda_m\, J^N(\mathbf{x}_m, \mathbf{u}_m), \\ \text{s.t.} \quad & \sum_{m\in\bar{\mathcal{T}}} \lambda_m x_m(0) = z, \\ & \sum_{m\in\bar{\mathcal{T}}} \lambda_m = 1, \\ & \lambda_m \geq 0,\ \forall m \in \bar{\mathcal{T}}, \end{aligned} \tag{11}$$

where z is the current state measurement, and $\Lambda \in \mathbb{R}^M$ is the set of weights $\{\lambda_1,\dots,\lambda_M\}$ for the convex combination of M partial control

[1] Feasible sequences can be extended with zeros so that their length becomes greater or at least equal to N, *i.e.*, $k^t_{\text{end}} \geq N$.

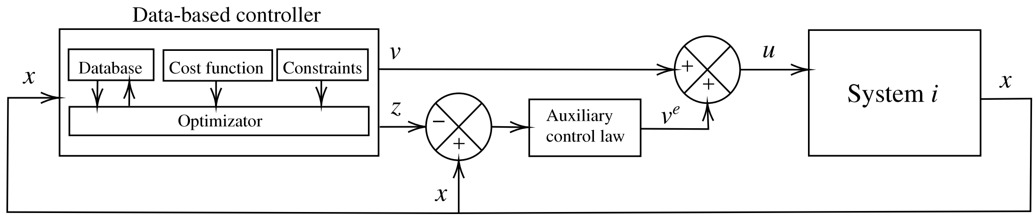


Fig. 1. Block diagram of the combined data-based and feedback controller.

sequences listed in set $\bar{\mathcal{T}}$, which contains the indices of $M \le |\mathcal{K}_t| \cdot |\mathcal{T}|$ partial trajectories.

Here, however, we consider a database that lacks previous information on the uncertain system $x^+ = A_j\,x + B_j\,u + w$ being controlled. Therefore, the previous problem $\mathbb{P}_N(z)$ is transformed into $\mathbb{P}^*_N(x)$:

$$J^*_N(x) = \min_{z,\,\Lambda} \sum_{m\in\bar{\mathcal{F}}} \lambda_m\, J^N(\mathbf{x}_m, \mathbf{u}_m), \tag{12a}$$

$$\text{s.t.} \sum_{m\in\bar{\mathcal{F}}} \lambda_m x_m(0) = z, \tag{12b}$$

$$\sum_{m\in\bar{\mathcal{F}}} \lambda_m = 1, \tag{12c}$$

$$x \in z \oplus \Omega_x, \tag{12d}$$

$$z \in \mathcal{X} \ominus \Omega_x, \tag{12e}$$

$$\lambda_m \ge 0,\ \forall m \in \bar{\mathcal{F}}, \tag{12f}$$

where x is the current state measurement, $z \in \mathbb{R}^{n_x}$ is the associated nominal state, which is now a decision variable, and $\Lambda \in \mathbb{R}^M$ is the set of weights for the combination of M partial control sequences contained in the set of robustly feasible trajectories $\bar{\mathcal{F}}$, which gathers the indices of $M \le |\mathcal{K}_t| \cdot |\mathcal{F}|$ partial trajectories. Here, we bestow problem $\mathbb{P}^*_N(x)$ on an additional degree of freedom by optimizing the nominal state, similar to [34,35]. Problem $\mathbb{P}^*_N(x)$ can be solved very efficiently, although in the case of stringent timing requirements and large datasets, it may be necessary to perform parallel searches in the database using GPU computing or specific hardware, *e.g.*, FPGA.

Note that the feasible nominal sets of any given instance of Problem (12) are, respectively, a set of states and inputs that are obtained, respectively, by convex combinations of database state and input trajectories using coefficients of Λ that satisfy the constraints. In particular, let z^* and $\Lambda^* = \{\lambda^*_1, \lambda^*_2, \ldots, \lambda^*_M\}$ be the minimizer of (12), provided that it exists,[2] then the nominal input partial trajectory for (3) is given by:

$$\mathbf{v}^* = \sum_{m\in\bar{\mathcal{F}}} \lambda^*_m \mathbf{u}_m, \tag{13}$$

with the first element of $\mathbf{v}^* = \{v^*(0), \ldots, v^*(N-1)\}$ defining the nominal control law:

$$v(x) = v^*(0), \tag{14}$$

and the optimized current nominal state $z^*(0)$ is defined as:

$$z^*(0) = \sum_{m\in\bar{\mathcal{F}}} \lambda^*_m x_m(0).$$

At the next time instant, it is produced a new combination of partial sequences based on the most recent information available in the database following a receding-horizon strategy.

[2] The existence of the minimizer requires the database to be large enough and have data representative of the operating conditions of the system. This issue can also be mitigated by increasing the number of samples considered for the combination (*i.e.*, M) and using soft constraints to solve Problem $\mathbb{P}^*_N(x)$.

3.3. Control algorithm

We summarize the robust predictive controller proposed based on historical data in Algorithm 1, which is executed at each time instant. Note that, as a selection criterion to limit the computational burden, we consider that only the samples listed in $\bar{\mathcal{F}}$ can be combined to compute the control inputs.

Let $\mathrm{d} : \mathcal{X} \times \mathcal{F} \to \mathbb{R}$ be a function that computes the distance between a system's state x_i and a trajectory's partial state x_t. The set $\bar{\mathcal{F}}$ contains the indices of the M feasible partial trajectories with the smallest $\mathrm{d}\big(x_i, x_t(k)\big)$ for all $k \in \mathcal{K}_t \triangleq \{1, \ldots, k^t_{\text{end}}\}$ and $\forall t \in \mathcal{F}$, and $|\bar{\mathcal{F}}| = M$ with $M \le \sum_{t\in\mathcal{F}} |\mathcal{K}_t|$.

Algorithm 1 Robust data-based predictive controller

Inputs: $x_i, \mathcal{X}, \mathcal{U}, Q, R, P, N, \mathcal{F}, M, K, \Omega$. **Output:** u.

1: For all $t \in \mathcal{F}$, compute the distance $\mathrm{d}\big(x_i, x_t(k)\big)$ for $k \in \mathcal{K}_t$.
2: Gather in $\bar{\mathcal{F}}$ the indices of the M feasible partial sequences $(\mathbf{x}_m, \mathbf{u}_m)$ with the lowest distance to x_i .
3: Evaluate the cost $J^N(\mathbf{x}_m, \mathbf{u}_m)$ of each candidate $m \in \bar{\mathcal{F}}$.
4: Solve (12) to optimize the weight vector Λ^* for the convex combination of the M control sequences, and the nominal state z^*.
5: Obtain $\mathbf{v}^*$ by (13), and set the first element as $v = v^*(0)$.
6: Calculate the auxiliary control input term v^e that penalizes the deviation between x_i and z^* (see Section 4 for more details).
7: Obtain (9) and apply u to the system.
8: Wait for the next sampling time and repeat from Step 1.

4. Auxiliary control law $v^e(x, z)$

The tube-based strategy requires the auxiliary control law to keep the evolution of the error confined within set Ω_x using control actions in set Ω_u despite the role played by parametric uncertainties, which lie within set $\mathcal{W}$. Typically, the design of sets Ω_x and Ω_u is achieved using a feedback controller K and a robust positively invariant (RPI) set, which, together with system (3), guarantee

$$(A_j + B_j K)\,\Omega_x \oplus \mathcal{W} \subseteq \Omega_x, \tag{15}$$

where $\Omega_x \subset \mathcal{X}$ and $\Omega_u \triangleq K\,\Omega_x \subset \mathcal{U}$. That is, considering $v^e(x, z) = K(x - z)$ as the auxiliary control law, the system dynamics (A_i, B_i) from the viewpoint of jth system realization (3) and the double control law (9) are

$$x^+ = A_j\,x + B_j\,(v + K\,e) + w, \tag{16}$$

where $e = (x - z) \in \Omega_x$ captures the difference between the real and nominal values of the state, and $w \in \mathcal{W}$ accounts for the model mismatch between systems j and i. As a consequence, the trajectory of system realization i will lie inside a tube around the *nominal* trajectory generated by the data-based controller. The system will be recursively feasible if there is a feasible solution in the database for its trajectory toward the origin, thanks to the combination of the database sequences. Finally, the system state will lie within a set Ω_x around the origin.

The cornerstone of this approach is feedback gain K. The methods for designing K vary based on the availability and extent of prior information about the system. In the following subsections, we will discuss several strategies that can be effectively employed in this context.

4.1. Redundancy of feedback gain

The role of K is to guarantee that the closed-loop dynamics $(A_j + B_j K)$ are stable. However, some systems are inherently stable (*e.g.*, passive systems) and therefore, Eq. (15) holds even if no feedback gain is applied, *i.e.*, $K = 0$. This means that robustness can be attained if the set of system realizations $\big(A(\theta), B(\theta)\big)$ is known to be stable. That being said, even in this case, feedback is often used in practice to improve the system's performance and robustness features.

4.2. Empirical approach

For some systems, it may be possible to find a feedback controller K for all system realizations empirically. In such a case, Ω_x and Ω_u could be designed as balls of ratio $r_x, r_u > 0$, respectively, *i.e.*, $\Omega_x = \mathbb{B}_x \triangleq \{x \in \mathcal{X} : \text{dist}(x, 0) \leq r_x\}$ and $\Omega_u = \mathbb{B}_u \triangleq \{u \in \mathcal{U} : \text{dist}(u, 0) \leq r_u\}$. Here, r_x and r_u become tuning parameters that might be adjusted considering the database trajectories to maintain system constraints in a robust manner.

4.3. Data-based design

The design of the auxiliary control law and the invariant set can also be derived from data. Nevertheless, typical approaches are not *model-free* because they often involve the implicit or explicit identification of a model based on the available data, which is then used to compute the invariant set and the control law. For example, in [36] a semidefinite programming problem (SDP) is formulated to simultaneously find the uncertain model, the feedback gain, and the corresponding invariant sets. A concurrent approach is also proposed in [37,38] to compute an approximation of a minimal robust control invariant set (mRCI) from experimental data via linear programming (LP) and select a suitable model given a model structure.

4.4. Model-based design

There are multiple well-known model-based design methods to generate a feedback gain that yields stable closed-loop dynamics for a particular realization, e.g., pole placement. In this regard, note that this is all that is required by the proposed method. Likewise, given the robust nature of the problem, it may be desirable to employ a robust feedback controller K valid for all system realizations, assuming that information regarding the vertices of the uncertainty set is available. This can be performed using linear matrix inequalities (LMIs) [39] leveraging Assumption 2, which makes the disturbance set of system realizations $\big(A(\theta), B(\theta)\big)$ polytopic. In particular, let set Y denote the vertices of this set, so that the dynamics of the extreme realizations can be expressed as:

$$x^+ = A_v x + B_v u, \quad v \in Y. \tag{17}$$

Lemma 1 (*Optimal Feedback Controller [40]*)**.** *Given system (A_v, B_v) and stage cost matrices Q and R. If there are matrices $H = H^\top \in \mathbb{R}^{n_x \times n_x}$ and $Y \in \mathbb{R}^{n_u \times n_x}$, such that the following constraints are satisfied*

$$\begin{bmatrix} H & HA_v^\top + Y^\top B_v^\top & HQ^{1/2} & Y^\top R^{1/2} \\ A_v H + B_v Y & H & 0 & 0 \\ Q^{1/2} H & 0 & \mathrm{I} & 0 \\ R^{1/2} Y & 0 & 0 & \mathrm{I} \end{bmatrix} \geq 0, \quad \forall v \in Y, \tag{18}$$

then there exist an optimal feedback control $K = YH^{-1}$, maximizing the trace of H, that stabilizes the closed-loop system, and a Lyapunov function $f(x) = x^\top P x$, with $P = H^{-1}$.

See [40, Appendix A] for the proof of Lemma 1.

Given the feedback gain K and the bounded disturbances $\mathcal{W}$ leveraging Assumption 2, which makes the disturbance set of system realizations $\big(A(\theta), B(\theta)\big)$ polytopic, we seek to find now an RPI set (Ω_x) as small as possible to reduce conservatism. For polytopic disturbances, methods such as the linear programming method proposed in [41] or the outer approximation presented in [42] can be employed to compute a common minimal RPI set for the system realizations. Similarly, one can find an RPI set for each vertex and then use the union of the RPI sets in the checking steps (10).

5. Recursive feasibility and stability

Regardless of the method employed to find feedback gain K and sets Ω_x and Ω_u, it is possible to prove recursive feasibility and stability.

Assumption 6. There exists, at least, a feedback controller K that can stabilize system (3) and a robust positively invariant (RPI) set Ω_x for constraint sets $(\mathcal{X}, \mathcal{U}, \mathcal{W})$ under the control law $u = Kx$ that satisfies (15).

Without lost of generality, we here consider that the database trajectories end in the origin although, as state in Assumption 4, it is not necessary. The only difference is the nominal state will be confined into a ball centered at the origin, therefore the uncertain system will be ultimately bounded in that ball plus the set Ω_x.

The robust feasibility and stability properties of the proposed algorithm are inherited from tube-based MPC controllers [28], which impose the following constraints in the computation of the nominal trajectory: (i) the *linear* model, (ii) the set of *tighter constraints* (10) and (12e), and (iii) a *terminal region* to guarantee stability and recursive feasibility. Here, the data-based controller provides a nominal trajectory, which accounts for condition (i). The set of Eqs. (10) guarantees that only robustly feasible sequences are considered, and (12e) ensures robustly feasible nominal states (condition ii). Likewise, there is no need for a terminal region because the generated trajectories reach the origin, so that recursive feasibility is guaranteed (condition iii). Finally, the auxiliary control law is identical to that of tube-based MPC and deals with the *disturbances* arising from the use of a nominal trajectory that belongs to a combination of trajectories from possibly different realizations of system (1).

5.1. Recursive feasibility

For the feasibility analysis of the closed-loop system, let us first define the domain of the optimization problem $\mathbb{P}_N^*(x)$. The feasible input set of problem (12), denoted as $\mathbb{V}^N(z)$, is a set of nominal input trajectories $\{v^*(0), v^*(1), \dots, v^*(N-1)\}$, which are obtained from a convex combination of the database input trajectories using coefficients of Λ^* (recall (13)). Likewise, the feasible state set $\mathbb{Z}^N = \{z \in \mathcal{X} \ominus \Omega_x : \mathbb{V}^N(z) \neq \emptyset\}$ is composed of nominal states that are obtained from a convex combination of the database state trajectories using coefficients of Λ^*. The feasible state set for the *real* dynamics is $\mathbb{X}^N = \mathbb{Z}^N \oplus \Omega_x$. A state $x = (z + e) \in \mathbb{X}^N$ is said to be recursively feasible if $\mathbb{V}^N(z^+) \neq \emptyset$ and $e^+ \in \Omega_x$.

Theorem 1 (*Recursive Feasibility*). *Consider that Assumptions 1–6 hold. Suppose also $x = (z + e) \in \mathbb{X}^N$ at instant k so that $(z^*(0), \mathbf{v}^*)$, where $\mathbf{v}^* = \{v^*(0), v^*(1), \dots, v^*(N-1)\}$ and its associated nominal state trajectory $\mathbf{z}^* = \{z^*(0), z^*(1), \dots, z^*(N)\}$ are feasible for problem $\mathbb{P}_N^*(x)$. Then, for all $x^+ = A_j x + B_j u + w$ with $w \in \mathcal{W}$, problem $\mathbb{P}_N^*(x^+)$ is also feasible for the next instants.*

Proof. Given a feasible state $x \in z^*(0) \oplus \Omega_x$, the successor state $x^+ \in z^*(1) \oplus \Omega_x$ remains within the *tube* with the applied control $u = v^*(0) + K(x - z^*(0))$ (for a detailed proof, see [43, Proposition 1]). Since there is a feasible solution $(z^*(0), \mathbf{v}^*)$ at instant k, the system trajectory toward the origin is recursively feasible for the next instants as it is the combination of database trajectories.

5.2. Robust exponential stability

Here, we analyze the robust stability of the controlled uncertain system $x^+ = A_j x + B_j u + w$, $\forall w \in \mathcal{W}$, which will inherit the properties of the database trajectories, so let us consider the next assumption to prove stability.

Assumption 7. The trajectories in the database $\mathcal{T}$ are generated using stabilizing controllers, and for all $z \in \mathbb{Z}^N$, the value function $J_N^o(z)$ (recall Eq. (11)) is Lyapunov, *i.e.*,

$$J_N^o(z^+) \leq J_N^o(z) - \ell(z, v), \tag{19}$$

so there exist positive constants $b > a > 0$ such that

$$\begin{aligned} J_N^o(z) &\geq a\,\|z\|^2, \\ J_N^o(z) &\leq b\,\|z\|^2, \\ J_N^o(z^+) &\leq J_N^o(z) - a\,\|z\|^2. \end{aligned} \tag{20}$$

In this way, the origin is exponentially stable for the controlled nominal dynamics $z^+ = A_j\, z + B_j\, v$ for all $z \in \mathbb{Z}^N$. Should there be trajectories that do not satisfy Assumption 7, they can be removed from the database. Although Assumption 7 is restrictive, it is necessary to inherit stability from the nominal trajectories generated by the data-based controller.

Proposition 1. *(i) For all $x \in \mathbb{X}^N$, $J_N^*(x) = J_N^o(z^*(0))$ and their corresponding control solutions $\mathbf{v}^*(x) = \mathbf{v}^o(z^*(0))$. (ii) $J_N^*(\cdot)$ is a Lyapunov function for dynamics $x^+ = A_j\, x + B_j\, u + w$, i.e.,*

$$J_N^*(x^+) \leq J_N^*(x) - \ell(z^*(0), v^*(0)). \tag{21}$$

Proof. The result (i) follows from a direct consequence of the definition of problems $\mathbb{P}_N(z)$ and $\mathbb{P}_N^*(x)$. To prove (ii), consider Theorem 1 and note that $x^+ = z^*(1) \oplus \Omega_x$ such that $(z^*(1), \mathbf{v}^o(z^*(1)))$ is also feasible for problem $\mathbb{P}_N^*(x^+)$, but $J_N^o(z^*(1)) \geq J_N^*(x^+)$. From (19), we have $J_N^o(z^*(1)) \leq J_N^o(z^*(0)) - \ell(z^*(0), v^*(0))$, which is lower bounded by $J_N^o(x^+) \leq J_N^o(z^*(1)) \leq J_N^o(z^*(0)) - \ell(z^*(0), v^*(0))$. Since $J_N^*(x) = J_N^o(z^*(0))$ by (i), we conclude that (21) holds.

Theorem 2 (*Robust Exponential Stability*). *Consider that Assumption 7 holds. For all $x \in \mathbb{X}^N$ and $w \in \mathcal{W}$, the control law $u = v^*(0) + K(x - z^*(0))$ stabilizes the uncertain system $x^+ = A_j\, x + B_j\, u + w$ exponentially in the set Ω_x.*

Proof. As proved in [34], since $J_N^*(x) = J_N^o(z^*(0))$ (let us denote $z^*(0) = z_0^*(x)$ to make explicit the dependence of the state x) for all $x \in \mathbb{X}^N$, we obtain the following set of equations using (20) and (21):

$$J_N^*(x) = J_N^o(z_0^*(x)) \geq a\,\|z_0^*(x)\|^2, \tag{22a}$$

$$J_N^*(x) = J_N^o(z_0^*(x)) \leq b\,\|z_0^*(x)\|^2, \tag{22b}$$

$$J_N^*(x^+) - J_N^*(x) \leq -\ell(z_0^*(x), v_0^*(x)) \leq -a\,\|z_0^*(x)\|^2, \tag{22c}$$

where $b > a > 0$. Let $x(k) = x^+$ denote the state system at instant k given an initial state $x(0) = x$ and $w \in \mathcal{W}$; from Theorem 1, $x(k) \in \mathbb{X}^N$ for all k. Let us define $S_\alpha \triangleq \{x : J_N^*(x) \leq \alpha\}$ for all $\alpha \geq 0$. Then, $S_0 \in \Omega_x$ for $\alpha = 0$, and there is an $\alpha > 0$ such that $S_\alpha \subset \mathcal{X}$. From (22), we have $J_N^*(x(k)) \leq \rho^k J_N^*(x(0))$, where $\rho^k \triangleq (1 - (a/b)) \in (0, 1)$. Iterating from $k = 0$ to k and following $J_N^*(x(k)) \leq \rho^k J_N^*(x(0))$, we have that $a\,\|z_0^*(x(k))\|^2 \leq \rho^k\, b\,\|z_0^*(x(0))\|^2$, *e.g.*, $\|z_0^*(x(k))\| \leq \delta^k c\,\|z_0^*(x(0))\|$, where $\delta^k = \sqrt{\rho}$ and $c = \sqrt{b/a} < \infty$ for all $x(0) = x \in \mathcal{S}_\alpha$. Since $\mathbb{X}^N$ is bounded, it yields from (22c) the existence of a finite integer D such that $x(k) \in \mathcal{S}_\alpha$ for all $k \in D$, $x \in \mathbb{X}^N$ and $w \in \mathcal{W}$. Due to $x(k) \in z_0^*(x(k)) \oplus \Omega_x$, the set Ω_x is robustly exponentially stable for the uncertain system with all $x \in \mathbb{X}^N$.

6. Cooperative learning example

A potential application, for which the proposed approach can be used as-is, is that of cooperative learning, where multiple controllers share their previous experiences to attain better performance. Let us consider a set of robots moving in a plane, so that their position and velocities are, respectively, controlled and manipulated variables. Moreover, the set of robots share a cloud-based database that collects information from different robots' controllers. We may model such robots as a system with parametric uncertainties that is formed *nominally* by two discrete-time linear integrators as follows:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^+ = A(\theta) \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + B(\theta) \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \tag{23}$$

with state $x = [x_1, x_2]^\top$ and control input $u = [u_1, u_2]^\top$ both subject to constraints:

$$[-2, -2]^\top \leq \{x, u\} \leq [2, 2]^\top.$$

Matrices $A(\theta)$ and $B(\theta)$ are, respectively, composed of the *nominal* matrices $A = \mathrm{I}$ and $B = \mathrm{I}$ plus parametric uncertainties

$$\Delta A_\theta = \begin{bmatrix} \Delta A_{\theta 1} & 0 \\ 0 & \Delta A_{\theta 2} \end{bmatrix}, \; \Delta B_\theta = \begin{bmatrix} \Delta B_{\theta 1} & 0 \\ 0 & \Delta B_{\theta 2} \end{bmatrix},$$

where $\Delta A_{\theta 1}, \Delta A_{\theta 2}, \Delta B_{\theta 1}, \Delta B_{\theta 2} \in [0, 0.4]$.

For simulations, we consider the realizations of the nominal system detailed in Table 2. The simulation length is $N_{\mathrm{end}} = 30$, the prediction horizon is $N = 5$, and the cost function is defined by (8) with $Q = 0.5 \cdot \mathrm{I}$ and $R = 5 \cdot \mathrm{I}$. The optimal feedback controller K for all system realizations $(A(\theta), B(\theta), Q, R)$ is calculated by solving the set of LMIs (18) obtaining $K = -0.7257 \cdot \mathrm{I}$. The RPI set Ω_x is calculated with the `MPT toolbox` of MATLAB taking into account the set $\mathcal{W}$ defined by (5). Furthermore, we have created a database of $T = 100$ trajectories from the nominal system (A, B) by using different proportional controllers. Note that we consider the most restrictive case in which there are no data from other realizations. These trajectories start from different feasible initial states and reach the origin, *i.e.*, $x_{\mathrm{obj}} = [0, 0]^\top$. Each trajectory has k_{end}^t–step information, where the value of each k_{end}^t depends on when its state error was below the threshold $\epsilon = 10^{-3}$. Hence, there is a total of $\sum_{t \in \mathcal{T}} k_{\mathrm{end}}^t$ partial trajectories, but the algorithm only considers $M = 1500$ to combine them.

We test our robust data-based controller (Robust DBC) with the data-based controller (DBC) previously proposed in [14]. For initial state $x(0) = [-1.8, 1.8]^\top$, Fig. 2 shows a comparison of state trajectories $\{x(k)\}$ obtained by implementing both controllers for realizations $\{(A, B), (A_1, B_1), \ldots, (A_4, B_4)\}$. We also show all trajectories in the database ($\mathcal{T}$), the sequence of the optimized nominal state $\{z_0^*(x(k))\}$ centered in the RPI set Ω_x, and the constraint set $\mathcal{X}$. Furthermore, for all considered system realizations, Table 2 displays a numerical comparison in terms of the maximum computing time to solve the control problem (denoted as $CT_{\max}$ [s]), the absolute average error (E_{abs}):

$$E_{\mathrm{abs}} = \sum_{k=1}^{N_{\mathrm{end}}} \frac{\|x(k) - x_{\mathrm{obj}}\|}{N_{\mathrm{end}}},$$

and the cumulative performance cost (J_{acc}):

$$J_{\mathrm{acc}} = \sum_{k=1}^{N_{\mathrm{end}}} x(k)^\top Q\, x(k) + u(k)^\top R u(k).$$

As shown in Table 2, the implementation of the Robust DBC results in lower absolute average error and cumulative performance cost than the DBC method for all system realizations.

Moreover, as depicted in Fig. 2a, the DBC method fails to control the system realizations (A_2, B_2) and (A_4, B_4) driving the states far from the origin. In contrast, the Robust DBC control method successfully steers the state toward the goal for all system realizations. These results

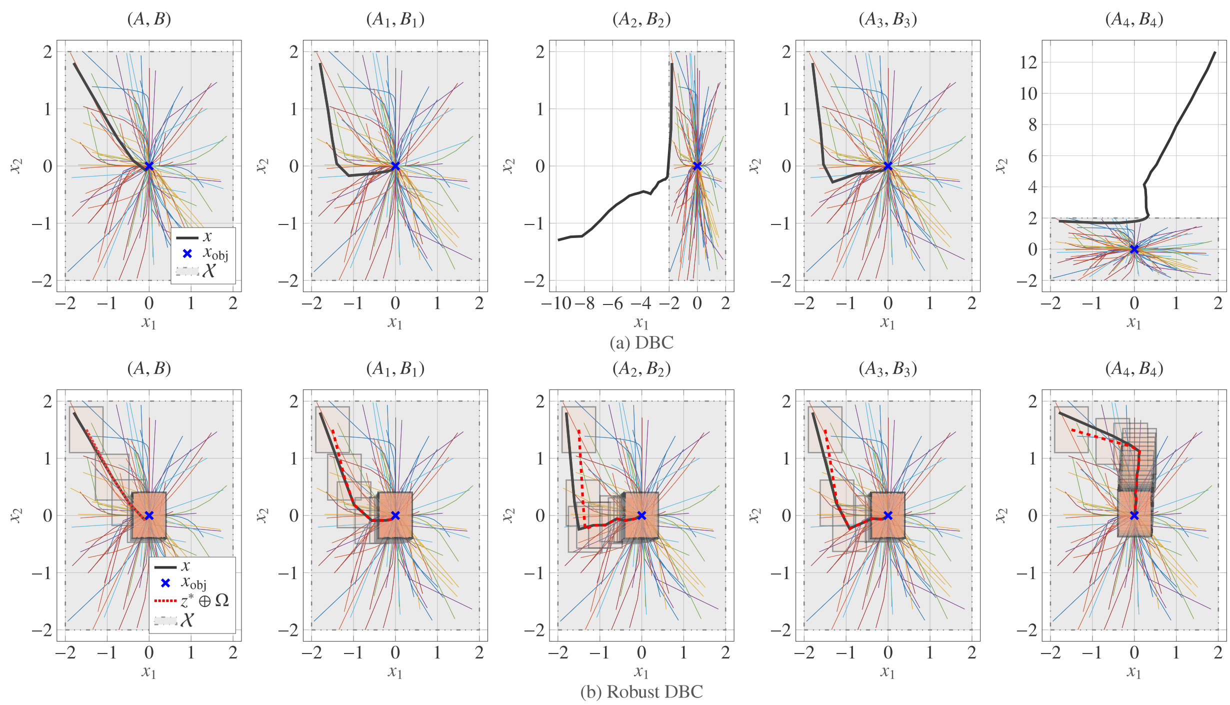


Fig. 2. State trajectories with initial state $x(0) = [-1.8, 1.8]^{\top}$ for realizations $\{(A, B), (A_1, B_1), \ldots, (A_4, B_4)\}$ by implementing our Robust DBC and the DBC previously proposed in [14].

Table 2
R2.4 Numerical results comparison between the DBC method previously proposed in [14], and our Robust DBC strategy.

System realizations	System parametric uncertainties				DBC			Robust DBC		
	$\Delta A_{\theta 1}$	$\Delta A_{\theta 2}$	$\Delta B_{\theta 1}$	$\Delta B_{\theta 2}$	CT_{max} [s]	E_{abs}	J_{acc}	CT_{max} [s]	E_{abs}	J_{acc}
(A, B)	0	0	0	0	0.0036	0.13	11.17	0.0256	0.11	10.30
(A_1, B_1)	0.4	0	0.4	0	0.0022	0.21	20.83	0.0217	0.14	13.88
(A_2, B_2)	0.4	0	0	0.4	0.0015	2.35	613.49	0.0221	0.20	22.77
(A_3, B_3)	0.4	0	0.4	0.4	0.0022	0.18	18.66	0.0203	0.12	12.67
(A_4, B_4)	0	0.4	0.4	0	0.0013	2.69	870.78	0.0192	0.29	24.58
(A_5, B_5)	0	0.4	0	0.4	0.0014	0.17	14.94	0.0187	0.12	11.70
(A_6, B_6)	0	0.4	0.4	0.4	0.0009	0.17	13.85	0.0192	0.12	10.13
(A_7, B_7)	0.4	0.4	0.4	0	0.0017	0.60	55.45	0.0242	0.24	23.56
(A_8, B_8)	0.4	0.4	0	0.4	0.0013	2.04	369.57	0.0195	0.27	28.92
(A_9, B_9)	0.4	0.4	0.4	0.4	0.0013	0.27	25.82	0.0186	0.16	15.57

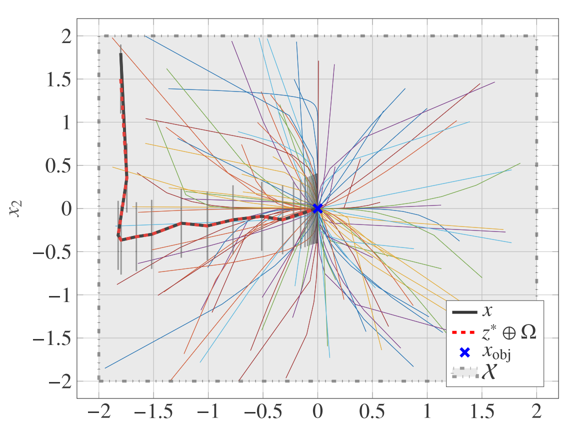


Fig. 3. State trajectory with initial state $x(0) = [-1.8, 1.8]^{\top}$ for realization (A_2, B_2) by implementing our Robust DBC with a feedback gain K designed solving the LMIs [39] for a particular different realization.

demonstrate that the original method can be very sensitive to problems such as uncertainties or noise in the data, where the proposed method is inherently more robust.

Finally, Fig. 3 illustrates how the controller is capable to stabilize the system even if the feedback gain is poorly design for a different particular realization. Specifically, we show how the proposed method is applied to system realization: (A_2, B_2). Here, it has been used as an auxiliary controller the feedback gain $K = -0.2475 \cdot \mathrm{I}$, which was designed specifically for a different system realization within the uncertain set, *e.g.*, (A_{10}, B_{10}):

$$A_{10} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad B_{10} = \begin{bmatrix} 1.4 & 0 \\ 0 & 1.4 \end{bmatrix},$$

solving the LMIs [39]. This case is interesting because this feedback gain would yield unstable closed-loop dynamics with eigenvalues 0.6536 and 1.1525 if it was used directly as main controller. Therefore, this example illustrates one of the remarkable features of the proposed method, which is to require only a feedback gain that stabilizes at least one of the system realizations. Note also that the change of gain in this example motivates the different shape of the invariant set.

7. Nonlinear-dynamics example

We consider the quadruple-tank plant presented in [44] as the case study with non-linear dynamics.

The plant is formed by four interconnected tanks, where the upper tanks (#3 and #4) discharge flow rates to the lower tanks (#1 and #2),

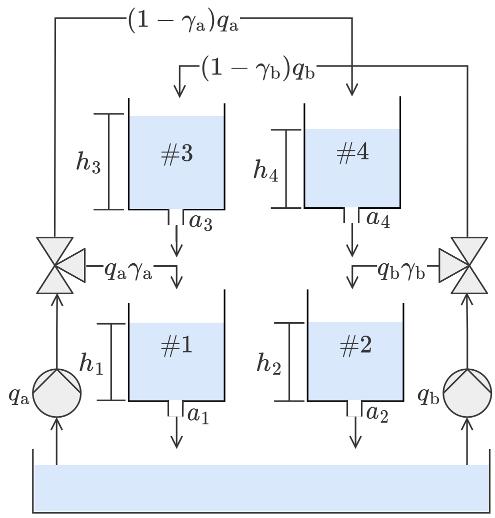


Fig. 4. Scheme of the quadruple-tank plant.

and these, in turn, into a sinking tank. The plant is controlled by two pumps that keep the water circulation between the tanks. There are also two three-way valves that divide the flow into two ways (see Fig. 4). Applying the mass balance and Bernoulli's law to the plant, the non-linear dynamics are:

$$\begin{aligned} S\frac{dh_1}{dt} &= -a_1\sqrt{2\,gh_1} + a_3\sqrt{2\,gh_3} + \gamma_a\frac{q_a}{3600}, \\ S\frac{dh_2}{dt} &= -a_2\sqrt{2\,gh_2} + a_4\sqrt{2\,gh_4} + \gamma_b\frac{q_b}{3600}, \\ S\frac{dh_3}{dt} &= -a_3\sqrt{2\,gh_3} + (1-\gamma_b)\frac{q_b}{3600}, \\ S\frac{dh_4}{dt} &= -a_4\sqrt{2\,gh_4} + (1-\gamma_a)\frac{q_a}{3600}, \end{aligned} \tag{24}$$

where $S = 0.03$ m^2 is the cross section of the tanks, $\{h_1, h_2, h_3, h_4\}$ are the water levels of the tanks, and $\{a_1, a_2, a_3, a_4\}$ are the cross section of the outlet pipes. The parameters $\{\gamma_a, \gamma_b\} \in [0, 1]$ refer to the opening of the three-way valves ($\gamma_a = 0.3$, $\gamma_b = 0.4$), $g = 9.81$ m/s^2 is the gravitational constant, and $\{q_a, q_b\}$ are the pump flow rate.

The dynamics of the non-linear system (24) can be linearized by its Taylor expansion as:

$$\hat{x}^+ = A\hat{x} + B\hat{u}, \tag{25}$$

where $\hat{x} = [h_1 - h_1^0, \dots, h_4 - h_4^0]^\top$ and $\hat{u} = [q_a - q_a^0,\ q_b - q_b^0]^\top$ around the following reference point:

$$\begin{bmatrix} h_1^0 \\ h_2^0 \\ h_3^0 \\ h_4^0 \end{bmatrix} = \begin{bmatrix} 0.5005 \\ 0.4704 \\ 0.5206 \\ 0.4319 \end{bmatrix} \text{m}, \quad \begin{bmatrix} q_a^0 \\ q_b^0 \end{bmatrix} = \begin{bmatrix} 1.5355 \\ 1.6794 \end{bmatrix} \text{m}^3/\text{h}.$$

Therefore, the discrete-time linear nominal system (A, B) with a sampling time $T_s = 30$ s is characterized by:

$$A = \begin{bmatrix} 0.6654 & 0 & 0.1919 & 0 \\ 0 & 0.5971 & 0 & 0.2250 \\ 0 & 0 & 0.7643 & 0 \\ 0 & 0 & 0 & 0.7077 \end{bmatrix}, \quad B = \begin{bmatrix} 0.0684 & 0.0179 \\ 0.0254 & 0.0868 \\ 0 & 0.1461 \\ 0.1644 & 0 \end{bmatrix}.$$

We consider the following state and control input constraints:

$$\begin{aligned} 0\ [\text{m}] &\le \{h_1, h_2, h_3, h_4\} &&\le 1.2\ [\text{m}], \\ 0\ [\text{m}^3/\text{h}] &\le \{q_a, q_b\} &&\le 3\ [\text{m}^3/\text{h}]. \end{aligned} \tag{26}$$

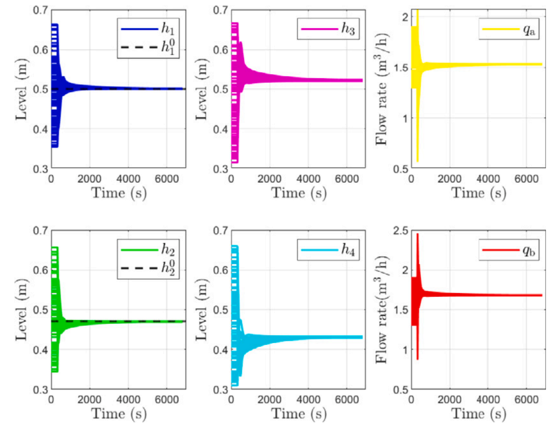


Fig. 5. Set of 100 trajectories of the nominal plant (A, B) that compose the database.

7.1. Control parameters and database

The control objective is to drive the states of tanks #1 and #2 to the reference point:

$$h^0 \triangleq \begin{bmatrix} h_1^0 \\ h_2^0 \end{bmatrix} = \begin{bmatrix} 0.5005 \\ 0.4704 \end{bmatrix} \text{ m}, \tag{27}$$

while minimizing the control cost (8) subject to constraints (26). The prediction horizon is $N = 5$, matrices Q and R are:

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad R = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix},$$

the length of simulations is $N_{end} = 2010$ s, and the sampling time of the controller is $T_s = 30$.

The database is composed of $T = 100$ trajectories, which have been obtained from the *nominal* plant (A, B) using PI controllers. These trajectories start at different points and end at the reference point (27), as illustrated in Fig. 5. Since the plant is a multiple-input multiple-output (MIMO) system, we employ two discrete PI controllers, one to manage tank level h_1 via pump q_a and another to manage h_2 through pump q_b, whose parameters are randomized in the intervals: $K_p \in [2.25, 3.25]$ and $T_i \in [275, 475]$. The procedure carried out to obtain the trajectories can be divided into two steps. First, the system is brought to a point that meets the constraints (26). Once in steady state, we apply the PI controller to bring the system to the reference point (27) until the sum of 100 consecutive absolute errors of $h_1(k)$ plus $h_2(k)$ is below the threshold $\varepsilon = 0.35$. Note that despite using PIs to generate the database, we may encounter minor offset in some of the trajectories.

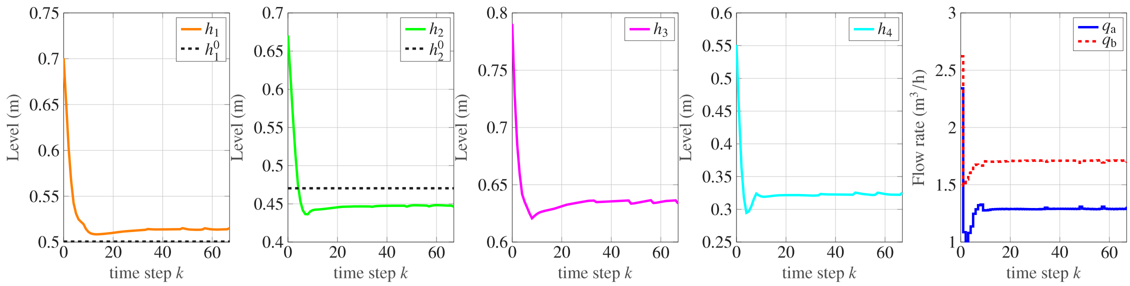
Each trajectory has 7000 seconds of information on the plant operation. Therefore, there is a total of $T \cdot 7000/T_s$ candidate partial trajectories at each time step, but only $M = 1500$ partial trajectories are considered by the algorithm to compute the final control sequence.

Furthermore, we consider five different system realizations, $\{(A_1, B_1), \dots, (A_5, B_5)\}$, to test the proposed control method, which are obtained by slightly changing the cross section of the outlet pipes of nominal system (A, B), as detailed in Table 3. Note that, for a practical application of the control method, the difference between systems should be small enough to apply robustness arguments.

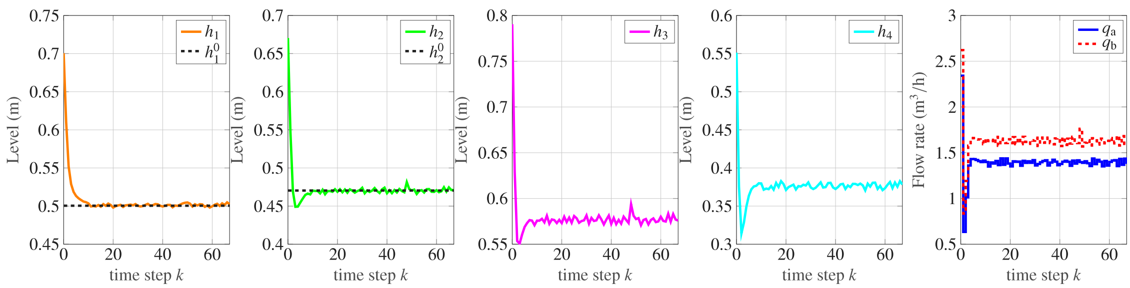
The optimal feedback controller K for all system realizations obtained by solving LMIs (18) is

$$K = \begin{bmatrix} -2.9758 & -0.7966 & -0.1686 & -1.5448 \\ -1.7242 & -1.7282 & -2.3000 & 0.3660 \end{bmatrix},$$

and Ω_x is calculated with `MPT toolbox` taking into account K and the polytopic set $\mathcal{W} \triangleq \{w \in \mathbb{R}^{n_w} : w < \mathbf{0.1}\}$.



(a) Simulation results with DBC [14]



(b) Simulation results with Robust DBC

Fig. 6. Results for system realization (A_2, B_2) by implementing our Robust DBC and the previously proposed DBC.

Table 3
R2.4 Numerical results comparison between the DBC method proposed in [14] and our Robust DBC strategy.

System realizations	Cross section of outlet pipes [m]				DBC			Robust DBC		
	$a_1 \cdot 10^{-4}$	$a_2 \cdot 10^{-4}$	$a_3 \cdot 10^{-4}$	$a_4 \cdot 10^{-4}$	CT_{max} [s]	E_{abs} (%)	J_{acc}	CT_{max} [s]	E_{abs} (%)	J_{acc}
(A, B)	1.301	1.597	0.876	1.026	0.0078	2.162	0.131	0.0236	0.751	0.039
(A_1, B_1)	1.341	1.627	0.896	1.108	0.0021	2.162	0.116	0.0238	0.706	0.038
(A_2, B_2)	1.236	1.488	0.807	0.997	0.0023	2.534	0.137	0.0320	0.751	0.041
(A_3, B_3)	1.236	1.627	0.807	1.108	0.0023	1.341	0.089	0.0267	0.714	0.038
(A_4, B_4)	1.683	1.691	0.964	0.940	0.0156	2.460	0.175	0.0219	0.869	0.053
(A_5, B_5)	1.475	1.845	0.896	1.208	0.0022	2.197	0.116	0.0243	0.668	0.037

7.2. Simulation results

We performed simulations for different system realizations to provide a numerical comparison between the data-based controller (DBC) presented in [14], and our robust data-based controller (Robust DBC).

Fig. 6 shows the trajectories of tank levels and flow rates $\{q_a, q_b\}$, considering the initial states ($h_1(0) = 0.7005$ m and $h_2(0) = 0.6704$ m, with their corresponding h_3 and h_4 levels) for the system realization (A_2, B_2). Table 3 displays the mean absolute error (%) of the tank levels $n = \{1, 2\}$ calculated as:

$$E_{abs}^n = \frac{\sum_{k=1}^{N_{end}/T_s} \left| h_n(k) - h_n^0 \right|}{N_{end}/T_s} \cdot 100,$$

$$E_{abs}(\%) = (E_{abs}^1 + E_{abs}^2)/2,$$

and also the accumulated cost:

$$J_{acc} = \sum_{k=1}^{N_{end}/T_s} \hat{h}(k)^\top Q\, \hat{h}(k) + \hat{q}(k)^\top R\, \hat{q}(k),$$

where

$$\hat{h}(k) = [h_1(k), h_2(k)] - [h_1^0, h_2^0],$$

$$\hat{q}(k) = [q_a(k), q_b(k)] - [q_a^0, q_b^0].$$

As displayed in Table 3, the mean absolute errors obtained with Robust DBC are lower than those obtained with the original algorithm. In terms of the accumulated cost, Robust DBC also outperforms the original algorithm for all system realizations.

Finally, Figs. 6(a) and 6(b) show the simulated trajectories of state and input for system realization (A_2, B_2). Our robust data-based controller is able to manage the plant with the information of the database (see Fig. 6(b)), while the original controller offers a worse control, as shown in Fig. 6(a). These results demonstrate the effectiveness of the proposed robust data-based approach in dealing with challenging aspects of systems, such as noise and parametric uncertainties.

8. Conclusions

The proposed approach extends a previously presented historian-based method with tube-based MPC ideas to obtain new properties such as robust constraint satisfaction, also paving the way for extensions to linear parameter-varying systems and applications such as cooperative learning. The main advantage of the new method is that it is straightforward to implement, as it does not require additional experiments, since historical data is typically available in most control systems. Finally, our outcomes in the two case studies show promising results and highlight the potential for data exchange between different realizations of the system in this context.

In particular, our proposal uses closed-loop trajectories and an auxiliary control law; two ingredients analogous to the ones of classical MPC methods (closed-loop predictions and semi-feedback strategies [45]) to reduce conservatism, exploiting the duality between parametric uncertainties and additive uncertainties in the proposed setup.

The adopted approach somehow navigates between the model-based and model-free paradigms. In particular, it is based on a dual control approach where a nominal and an auxiliary controller are employed. While the former can be implemented using only the trajectories contained in the database, the latter has to be designed empirically if absolutely no prior information regarding the system being controlled is available. However, as has been shown, there are also methods to derive models, control laws and invariant sets from data. In a less restrictive information setup, we only require the control designer to have access to a particular system realization within the uncertainty set – a very mild requirement – so that a feedback gain for that system realization can be computed together with the corresponding invariant set. Remarkably, as we have shown in the simulations, this is enough to control other realizations that yield unstable closed-loop dynamics with the previously mentioned feedback gain. As for the set of possible system realizations and the corresponding additive uncertainty, empirical bounds might be obtained if no prior information is available. Note that this is often done in robust predictive control approaches, since uncertainty sets are rarely known with precision.

Lastly, we discuss some of the potential limitations of our method. The availability of one robust trajectory is necessary to guarantee at least a feasible solution for the nominal problem. This assumption is somewhat equivalent to the assumption that standard MPC has solution at the first time instant to prove stability. Moreover, stable trajectories are required to finish at a bounded distance from the origin, so that trajectories with offset are also considered, in addition to the previously mentioned need for a feedback gain K for the auxiliary control law and knowledge about the uncertainty set to apply the tube-based controller.

CRediT authorship contribution statement

Eva Masero: Conceptualization, Methodology, Investigation, Formal analysis, Software, Data curation, Visualization, Writing – original draft, Writing – review & editing. **José M. Maestre:** Conceptualization, Methodology, Investigation, Formal analysis, Visualization, Writing – original draft, Writing – review & editing, Supervision, Funding acquisition. **José R. Salvador:** Conceptualization, Methodology, Writing – original draft. **Daniel R. Ramirez:** Conceptualization, Methodology, Writing – original draft, Writing – review & editing, Supervision, Funding acquisition. **Quanyan Zhu:** Conceptualization, Methodology.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The authors appreciate the financial support received from the European Research Council for the Advanced Research Grant OCONTSOLAR (ref. 789051), from MCIN/AEI/10.13039/501100011033 for the grants PID2020-119476RB-I00 (project C3PO-R2D2) and PID2022-141159OB-I00, and from the Spanish Ministry of Science, Innovation and Universities under the FPU programme for University Staff (ref. FPU18/04476).

References

[1] J. Richalet, Industrial applications of model based predictive control, Automatica 29 (5) (1993) 1251–1274.

[2] S. Vazquez, J.I. Leon, L.G. Franquelo, J. Rodriguez, H.A. Young, A. Marquez, P. Zanchetta, Model predictive control: A review of its applications in power electronics, IEEE Ind. Electron. Mag. 8 (1) (2014) 16–31.

[3] S.J. Qin, T.A. Badgwell, A survey of industrial model predictive control technology, Control Eng. Pract. 11 (7) (2003) 733–764.

[4] D. Görges, Relations between model predictive control and reinforcement learning, IFAC-PapersOnLine 50 (1) (2017) 4920–4928.

[5] P. Masding, B. Lennox, Use of dynamic modelling and plant historian data for improved control design, Control Eng. Pract. 18 (1) (2010) 77–83.

[6] N. Bhat, T.J. McAvoy, Use of neural nets for dynamic modeling and control of chemical process systems, Comput. Chem. Eng. 14 (4–5) (1990) 573–582.

[7] G.C. Calafiore, M.C. Campi, The scenario approach to robust control design, IEEE Trans. Automat. Control 51 (5) (2006) 742–753.

[8] C.J. Ong, D. Sui, E.G. Gilbert, Enlarging the terminal region of nonlinear model predictive control using the support vector machine method, Automatica 42 (6) (2006) 1011–1016, http://dx.doi.org/10.1016/j.automatica.2006.02.023.

[9] U. Rosolia, F. Borrelli, Learning model predictive control for iterative tasks. A data-driven control framework, IEEE Trans. Automat. Control 63 (7) (2018) 1883–1896.

[10] K.P. Wabersich, M.N. Zeilinger, Linear model predictive safety certification for learning-based control, in: Proceedings of the 57th Conference on Decision and Control, CDC, IEEE, 2018, pp. 7130–7135.

[11] T. Koller, F. Berkenkamp, M. Turchetta, A. Krause, Learning-based model predictive control for safe exploration, in: Proceedings of the 57th Conference on Decision and Control, CDC, IEEE, 2018, pp. 6059–6066, http://dx.doi.org/10.1109/CDC.2018.8619572.

[12] D. Limon, J. Calliess, J.M. Maciejowski, Learning-based nonlinear model predictive control, IFAC-PapersOnLine 50 (1) (2017) 7769–7776.

[13] E. Terzi, L. Fagiano, M. Farina, R. Scattolini, Learning-based predictive control for linear systems: A unitary approach, Automatica 108 (2019) 108473, http://dx.doi.org/10.1016/j.automatica.2019.06.025.

[14] J.R. Salvador, D. Muñoz de la Peña, D.R. Ramirez, T. Alamo, Predictive control of a water distribution system based on process historian data, Optim. Control Appl. Methods 41 (2) (2020) 571–586.

[15] J. Roll, A. Nazin, L. Ljung, Nonlinear system identification via direct weight optimization, Automatica 41 (3) (2005) 475–490.

[16] J.P. Kleijnen, Kriging metamodeling in simulation: A review, European J. Oper. Res. 192 (3) (2009) 707–716.

[17] N. Cressie, Kriging nonstationary data, J. Amer. Statist. Assoc. 81 (395) (1986) 625–634.

[18] J.M. Bravo, T. Alamo, M. Vasallo, M.E. Gegundez, A general framework for predictors based on bounding techniques and local approximation, IEEE Trans. Automat. Control 62 (7) (2016) 3430–3435.

[19] J.R. Salvador, D. Muñoz de la Peña, T. Alamo, A. Bemporad, Data-based predictive control via direct weight optimization, IFAC-PapersOnLine 51 (20) (2018) 356–361.

[20] G. Matheron, Principles of geostatistics, Econ. Geol. 58 (8) (1963) 1246–1266.

[21] J. Marzat, H. Piet-Lahanier, Design of nonlinear MPC by Kriging-based optimization, IFAC Proc. Vol. 45 (16) (2012) 1490–1495.

[22] A.D. Carnerero, D.R. Ramirez, T. Alamo, State-space Kriging: A data-driven method to forecast nonlinear dynamical systems, IEEE Control Syst. Lett. 6 (2022) 2258–2263, http://dx.doi.org/10.1109/LCSYS.2021.3140167.

[23] J.R. Salvador, D.R. Ramirez, T. Alamo, D. Muñoz de la Peña, Offset free data driven control: Application to a process control trainer, IET Control Theory Appl. 13 (18) (2019) 3096–3106.

[24] N. Paoletti, K.S. Liu, H. Chen, S.A. Smolka, S. Lin, Data-driven robust control for a closed-loop artificial pancreas, IEEE/ACM Trans. Comput. Biol. Bioinform. 17 (6) (2019) 1981–1993.

[25] J. Berberich, J. Köhler, M.A. Müller, F. Allgöwer, Data-driven model predictive control with stability and robustness guarantees, IEEE Trans. Automat. Control 66 (4) (2020) 1702–1717.

[26] J. Berberich, J. Köhler, M.A. Müller, F. Allgöwer, Robust constraint satisfaction in data-driven MPC, in: Proceedings of the 59th IEEE Conference on Decision and Control, CDC, IEEE, 2020, pp. 1260–1267.

[27] C. Mark, S. Liu, Data-driven distributionally robust MPC: An indirect feedback approach, 2021, arXiv preprint arXiv:2109.09558.

[28] W. Langson, I. Chryssochoos, S.V. Raković, D.Q. Mayne, Robust model predictive control using tubes, Automatica 40 (1) (2004) 125–133.

[29] I. Markovsky, F. Dörfler, Behavioral systems theory in data-driven analysis, signal processing, and control, Annu. Rev. Control 52 (2021) 42–64.

[30] C. Verhoek, R. Tóth, S. Haesaert, A. Koch, Fundamental lemma for data-driven analysis of linear parameter-varying systems, in: Proceedings of the 60th IEEE Conference on Decision and Control, CDC, IEEE, 2021, pp. 5040–5046.

[31] L. Li, C. De Persis, P. Tesi, N. Monshizadeh, Data-based transfer stabilization in linear systems, 2022, arXiv preprint arXiv:2211.05536.

[32] J.A. Rossiter, Y. Ding, Interpolation methods in model predictive control: An overview, Internat. J. Control 83 (2) (2010) 297–312.

[33] J.M. Maestre, E. Masero, J.R. Salvador, D.R. Ramirez, Q. Zhu, On data reutilization for historian based predictive control, in: Proccedings of the 61st IEEE Conference on Decision and Control, CDC, IEEE, 2022, pp. 474–478.

[34] D.Q. Mayne, M.M. Seron, S. Raković, Robust model predictive control of constrained linear systems with bounded disturbances, Automatica 41 (2) (2005) 219–224.

[35] Z. Dong, D. Angeli, Tube-based robust economic model predictive control on dissipative systems with generalized optimal regimes of operation, in: Proceedings of the 57th IEEE Conference on Decision and Control, CDC, IEEE, 2018, pp. 4309–4314.

[36] S.K. Mulagaleti, A. Bemporad, M. Zanon, Data-driven synthesis of robust invariant sets and controllers, IEEE Control Syst. Lett. 6 (2021) 1676–1681.

[37] Y. Chen, H. Peng, J. Grizzle, N. Ozay, Data-driven computation of minimal robust control invariant set, in: Proceedings of the 57th IEEE Conference on Decision and Control, CDC, IEEE, 2018, pp. 4052–4058.

[38] Y. Chen, N. Ozay, Data-driven computation of robust control invariant sets with concurrent model selection, IEEE Trans. Control Syst. Technol. 30 (2) (2021) 495–506.

[39] S. Boyd, L. El Ghaoui, E. Feron, V. Balakrishnan, Linear Matrix Inequalities in System and Control Theory, SIAM, 1994.

[40] M.V. Kothare, V. Balakrishnan, M. Morari, Robust constrained model predictive control using linear matrix inequalities, Automatica 32 (10) (1996) 1361–1379.

[41] P.A. Trodden, A one-step approach to computing a polytopic robust positively invariant set, IEEE Trans. Automat. Control 61 (12) (2016) 4100–4105.

[42] S.V. Raković, E.C. Kerrigan, K.I. Kouramas, D.Q. Mayne, Invariant approximations of the minimal robust positively invariant set, IEEE Trans. Automat. Control 50 (3) (2005) 406–410.

[43] D.Q. Mayne, W. Langson, Robustifying model predictive control of constrained linear systems, Electron. Lett. 37 (23) (2001) 1422–1423.

[44] K.H. Johansson, The quadruple-tank process: A multivariable laboratory process with an adjustable zero, IEEE Trans. Control Syst. Technol. 8 (3) (2000) 456–465.

[45] D.Q. Mayne, Control of constrained dynamic systems, Eur. J. Control 7 (2–3) (2001) 87–99.