

Online learning constrained model predictive control based on double prediction

J. M. Manzano¹  | D. Muñoz de la Peña¹ | J. Calliess² | D. Limon¹ 

¹Departamento de Ingeniería de Sistemas y Automática, Universidad de Sevilla, Sevilla, Spain

²Oxford-Man Institute of Quantitative Finance, University of Oxford, Oxford, UK

Correspondence

J. M. Manzano, Departamento de Ingeniería de Sistemas y Automática, Universidad de Sevilla, Camino de los Descubrimientos s/n 41092, Sevilla, Spain.
Email: manzano@us.es

Funding information

Ministerio de Economía y Competitividad, Grant/Award Number: DPI2016-76493-C3-1-R; Universidad de Sevilla, Grant/Award Number: VI-PPITUS

Summary

A data-based predictive controller is proposed, offering both robust stability guarantees and online learning capabilities. To merge these two properties in a single controller, a double-prediction approach is taken. On the one hand, a safe prediction is computed using Lipschitz interpolation on the basis of an offline identification dataset, which guarantees safety of the controlled system. On the other hand, the controller also benefits from the use of a second online learning-based prediction as measurements incrementally become available over time. Sufficient conditions for robust stability and constraint satisfaction are given. Illustrations of the approach are provided in a simulated case study.

KEYWORDS

data-based control, learning-based MPC, nonlinear MPC, robust control

1 | INTRODUCTION

When designing controllers, identifying the closed-loop behavior of the system w.r.t. the control action is crucial. There are systems whose identification via standard methods is not feasible, for instance because its complexity forbids the derivation of first-principle models, due to the large number of variables, or due to the presence of significant nonlinearities that might be unknown a priori.¹ Typically, nonlinear regression methods are employed to identify wide classes of systems based on input-output measurements of the plant. If little about the underlying (nonlinear) system is known with certainty a priori, then it is imperative to use regression methods that are flexible to learn rich classes of dynamics. Some of the most flexible regression methods for such purposes are machine learning methods, a number of which have been employed in data-based control. To name a few examples from a rich body of literature, some works have considered direct weight optimization methods,²⁻⁴ Gaussian processes,^{1,5} or random forests,⁶ among many others.

As a possible paradigm to tackle the control of such systems, data-based predictive control has recently gained increasing attention. The identified models are used by a model predictive controller (MPC) to calculate the control action that minimizes a performance index, according to the predicted evolution of the system. Inevitably, the performance of the resulting closed-loop dynamics rests on the accuracy of the predictions.

Besides, MPCs that combine robustness and stability guarantees with the data-based approach mentioned before are an open problem within the control community. An excellent survey paper that reviews learning and safety approaches for MPC can be found in Hewing et al.⁷ According to the classification therein, this article aims to learn the system dynamics in order to design a robust controller via a nonparametric model. Apart from MPC, other data-based approaches that are worth mentioning are reinforcement learning,⁸ dual control,⁹ and adaptive control.¹⁰

[Correction added on 22 September 2020, after first online publication: 'model predictive controller' has been changed to 'model predictive control' in the article title.]

As well as the machine learning methods mentioned above, Lipschitz interpolation (LI) techniques^{11,12} and their generalizations¹³⁻¹⁵ have been widely studied and applied to control, under the names *nonlinear set membership* (NSM)^{14,16} and *kinky inference* (KI),¹⁷ due to their favourable properties in data-based control. In this line, the authors have recently proposed learning predictive controllers whose predictions are inferred from this class of methods.^{18,19}

In general, the approaches reviewed²⁻¹⁹ are based on using a fixed dataset to design the model, usually collected offline from ad hoc identification experiments, and do not consider the possibility of improving the predictions online. The terms *online* and *learning* are particularly suitable when considering data-based control, since during the operation of the closed-loop plant, access to new observations of the system becomes available. It seems intuitive to add these data points to the database, in order to improve future predictions. According to T. Mitchell's definition,²⁰ an algorithm is *learning* if it improves its performance relative to some metric with increasing exposure to data. These new data points could be used to improve the estimations in real time. The improvement of the performance has been validated in multiple works within the last decade.^{21,22} For certain generalized LI techniques, the conditions under which it can be guaranteed that the prediction error vanishes (up to a factor of the level of observational error) have been studied in Reference 13.

However, when considering online learning-based predictive control settings, especially with unknown Lipschitz constants, there are few results that explicitly consider robustness issues, particularly when hard constraints are taken into account. Indeed, the design of data-based predictive controllers that include flexible online learning capabilities and guarantees of robust stability and constraint satisfaction is still an open problem.⁷

As a step toward addressing this challenge, this article extends the results presented in Manzano et al,^{18,19} which were based on offline model identification, to an online learning framework. In particular, an online learning MPC based on a double prediction model (similar in spirit to Aswani et al²³) is presented. The proposed MPC guarantees robust constraint satisfaction and stability by means of a set of tightened constraints, an appropriate terminal cost and an ad hoc designed data update policy. The controller proposed is not based on a terminal region, following a design procedure similar to the controller presented in Reference 19. This avoids the calculation of invariant sets in the controller design, which is in general a hard task. A preliminary version of this work, based on a terminal constraint and a suboptimal data update policy (in which the proofs of the stability analysis were omitted), was presented in Limon et al.²⁴

The rest of the article is structured as follows. Section 2 introduces the control problem. Section 3 presents the KI learning methodology. Section 4 presents the MPC, which includes the double prediction model and the stability guarantees. Finally, Section 5 introduces a simulated case study of a quadruple-tank process, in which the controller's properties are illustrated.

Notation

If v, w are two column vectors, (v, w) stands for $[v^T, w^T]^T$, $|v|$ denotes the vector whose components are the absolute value of the components of v , and $v < w$ implies that each component of v is smaller than its corresponding element of w . The set of integers from a to b is denoted \mathbb{I}_a^b . Given two sets A, B , the Minkowski sum $A \oplus B$ is defined as the set $\{a + b : a \in A, b \in B\}$, while the Pontryagin difference $A \ominus B$ is the set $\{c : c + b \in A, \forall b \in B\}$. Given a vector $v \in \mathbb{R}^n$, the ball $\mathcal{B}(v) \in \mathbb{R}^n$ is defined as $\mathcal{B}(v) = \{y : |y_s| \leq v_s, s \in \mathbb{I}_1^n\}$. If a continuous function $\alpha : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ is strictly increasing and $\alpha(0) = 0$, α is called a \mathcal{K} -function. The notation $\hat{y}(j|k)$ denotes the prediction of y , j time steps ahead, given the measured y at time k . The identity matrix of size n is denoted I_n . The projection of a vector x onto a set B is defined as $\text{Proj}_B(x) = \arg \min_{b \in B} \|b - x\|$. The maximum over a set of m vectors $\max_i v_i$ for all $i \in \mathbb{I}_1^m$ denotes the maximum elementwise.

2 | PROBLEM SETTING

The objective of this article is to develop stabilizing controllers that guarantee robust constraint satisfaction, while at the same time being able to improve their closed-loop performance using the data collected online. The plant to be controlled is a discrete-time system in which at sampling time $k \in \mathbb{N}$, the control actions are denoted $u(k) \in \mathbb{R}^{n_u}$ and the measured outputs $y(k) \in \mathbb{R}^{n_y}$. This plant is subject to hard constraints in the inputs and outputs, that is,

$$u(k) \in \mathcal{U}, y(k) \in \mathcal{Y}_{\text{hard}} \quad \forall k,$$

where the sets $\mathcal{U} \subset \mathbb{R}^{n_u}$ and $\mathcal{Y}_{\text{hard}} \subset \mathbb{R}^{n_y}$ are compact.

It is assumed that the evolution of the output can be characterized by the following nonlinear autonomous regression exogenous (NARX) model:

$$y(k+1) = f(x(k), u(k)) + e(k), \quad (1)$$

where $e(k) \in \mathbb{R}^{n_y}$ models the process noise, which is assumed to be upper-bounded by $\bar{e} \in \mathbb{R}^{n_y} \geq \sup_k (|e(k)|) \geq 0$. The state is defined by the following regression of previous outputs and inputs, for some memory horizons n_a and n_b :

$$x(k) = (y(k), y(k-1), \dots, y(k-n_a), u(k-1), \dots, u(k-n_b)). \quad (2)$$

In Levin and Narendra,²⁵ it is proven that under mild assumptions on the observability of a system, any system can be described by an NARX model. Besides, it is also shown that it can be posed as a state-space model as follows:

$$x(k+1) = F(x(k), u(k)) + \xi(k), \quad (3a)$$

$$y(k) = Mx(k), \quad (3b)$$

where

$$\begin{aligned} F(x(k), u(k)) &= (f(x(k), u(k)), y(k), \dots, y(k-n_a+1), u(k), \dots, u(k-n_b+1)), \\ M &= [I_{n_y}, 0, \dots, 0], \\ \xi(k) &= (e(k), 0, \dots, 0). \end{aligned}$$

It is assumed that the origin is the equilibrium point of the system (ie, $f(0,0) = 0$) where the plant must be stabilized, and that it is contained in $(\mathcal{Y}_{\text{hard}}, \mathcal{U})$.

We assume that the true target function f -and by extension, F - are unknown. Additionally, we assume that we have access to an initial dataset of measurements of the output y and that, in subsequent time steps, we might obtain additional measurements of this output (online). The objective is to design a predictive controller based on a dataset of input and output measurements, which is input-to-state stable²⁶ and that satisfies the constraints, as well as enhancing the performance of the closed-loop system by adding the online data during the operation of the system. To generate predictions we utilize the dataset to learn f (and thus, F) with a nonparametric regression approach that will be discussed in the following section.

3 | KI AS A LEARNING METHOD

This section briefly introduces the learning and prediction method used in this article, presented in Reference 13. In order to simplify notation, we note that the arguments of the function f (1) can be aggregated into the so-called regressors $w = (x, u) \in \mathcal{W} \subset \mathbb{R}^{n_w}$ which, with a slight abuse of notation, allows us to conceive the state transition function $f : \mathcal{W} \rightarrow \mathcal{Y}$ as a mapping $w(k) \mapsto y(k+1)$, which will be referred to as *ground truth function*.

Assumption 1. The ground truth function f is Lipschitz continuous, that is,

$$|f(w_1) - f(w_2)| \leq L^* \|w_1 - w_2\|, \forall w_1, w_2 \in \mathcal{W}. \quad (4)$$

The bound on the mapping is given by the smallest constant $L^* \in \mathbb{R}^{n_y}$ that satisfies (4), called the Lipschitz constant.

Then, the learning method¹³ consists of the following: Assume access to a noisy set of N_D sampled inputs and outputs, gathered in a dataset $\mathcal{D} = \{(w_i, \tilde{f}_i) | i \in \mathbb{I}_1^{N_D}\}$, where \tilde{f}_i denotes the i th noise-corrupted sample of f at input w_i . The objective is to learn the unknown function f , in order to estimate its value for unseen points $\tilde{w} \notin \mathcal{W}_D$ (where \mathcal{W}_D is the set of inputs contained in the dataset \mathcal{D}). In this learning method, the Lipschitz constant L^* is unknown. Instead, an estimation $L_D \in \mathbb{R}^{n_y}$ is computed as the minimum constant that is consistent with the data (with a regularization term to compensate for

the observational noise):¹³

$$L_D = \max_{(w, \tilde{f}) \in \mathcal{D}} \left\{ \frac{|\tilde{f}_i - \tilde{f}_j| - 2\bar{e}}{\|w_i - w_j\|} \mid i, j = 1, \dots, N_D \wedge \|w_i - w_j\| > 0 \right\}. \quad (5)$$

Using this estimation yields the so-called *lazily adapted constant kinky inference* (LACKI) predictor. After this estimate is computed from the data, the prediction $\hat{f}(\tilde{w}; L_D, \mathcal{D})$ of $f(\tilde{w})$ for a new query input \tilde{w} is computed as:

$$\hat{f}(\tilde{w}; L_D, \mathcal{D}) = \frac{1}{2} \min_{i=1, \dots, N_D} (\tilde{f}_i + L_D \|\tilde{w} - w_i\|) + \frac{1}{2} \max_{i=1, \dots, N_D} (\tilde{f}_i - L_D \|\tilde{w} - w_i\|). \quad (6)$$

This predictor $\hat{f}(\cdot; L_D, \mathcal{D})$ is Lipschitz continuous, with Lipschitz constant L_D .¹⁷ It will be used to forecast the evolution of the plant, yielding the prediction model

$$\hat{y}(k+1) = \hat{f}(x(k), u(k); L_D, \mathcal{D}). \quad (7)$$

4 | ONLINE LEARNING-BASED MPC

This section presents the proposed learning-based predictive controller. As we have already stated, it will be able to ensure robust stability and constraints satisfaction of the closed-loop system, while including new data points acquired during the plant operation in order to enhance its performance.

4.1 | Double prediction framework

This article considers an online learning set-up. In a first offline design stage, an initial dataset $\mathcal{D}(0)$ is available, obtained via specific experiments or given historical data. Once the controller is designed and applied to the plant, access to new measurements $y(k) = \tilde{f}(w(k-1)) = \tilde{f}(x(k-1), u(k-1))$ becomes available during its operation, allowing one to update the dataset up to the current time step k , yielding $\mathcal{D}(k)$. An update method $\mathcal{D}(k-1) \rightarrow \mathcal{D}(k)$ will be presented, heuristically tailored to the proposed control law and learning method. Similar to Aswani et al.,²³ the proposed controller will use two different prediction models, one for safety and one for performance.

The safe model

$$\hat{y}_s(k+1) = \hat{f}_s(x(k), u(k)) = \hat{f}(x(k), u(k); L_D, \mathcal{D}(0)), \quad (8)$$

is obtained applying \hat{f} in Equation (7) with $\mathcal{D}(0)$, and where L_D is obtained as in (5). It is derived offline from the initial data available before closing the loop. A state-space version can be obtained using Equation (3a), denoted $\hat{x}_s(k+1) = \hat{F}_s(x(k), u(k))$.

Given the Lipschitz continuity of the function of the real system, it can be proven that the estimation error of the proposed method is bounded.¹³ Under the assumption that this bound is known, the model provides a safe prediction that allows us to design deterministic robust controllers able to guarantee stability and constraint satisfaction. The error bound can be calculated from the following expression:¹³

$$\mu = (L^* + L_D)R_D + 2\bar{e}, \in \mathbb{R}^{n_y}, \quad (9)$$

where $R_D = \sup_{w \in \mathcal{W}_D} \inf_{\tilde{w} \notin \mathcal{W}_D} \|w - \tilde{w}\|$, for $\mathcal{D}(0)$. In this article, we assume that the only a priori knowledge from the plant is the input-output data collected from the experiments, with which the Lipschitz constant L_D can be estimated. However, the prediction error bound (9) still requires knowledge of the true Lipschitz constant of the plant, L^* , and the noise bound \bar{e} . Therefore, instead of using (9), the prediction error bound will be obtained in practice via validation tests, as it is customary in identification. In order to prove robust stability, it is assumed that a guaranteed bound of the estimation error is determined, as stated in the following assumption.

Assumption 2. The prediction error of the safe model $e_s(\cdot) \in \mathbb{R}^{n_y}$, which depends on the dataset $\mathcal{D}(0)$ and the estimated Lipschitz constant L_D , is bounded by some known $\mu \in \mathbb{R}^{n_y}$. That is, for all admissible (x, u) ,

$$e_s(k) = |y(k+1) - \hat{f}_s(w(k); L_D, \mathcal{D}(0))| \leq \mu. \tag{10}$$

The safe model and its bound e_s are necessary to prove robust constraint satisfaction, and therefore, safety of the controlled system. Note that $e_s(\cdot)$ accounts not only for the error induced by the underestimation of the Lipschitz constant, but also for the lack of information on the dataset, as well as for the effect of the noise in the sampled data.

Property 1. According to Assumption 2, the real output $f(w)$ lies in a ball centered in $\hat{f}_s(w)$ and width μ , which is defined as the set

$$\mathcal{Y}_s(x, u) = \{\hat{f}_s(x, u) \oplus B(\mu)\}. \tag{11}$$

The online model

$$\hat{y}_p(k+1) = \hat{f}_p(x(k), u(k)) = \text{Proj}_{\mathcal{Y}_s(x(k), u(k))} \left(\hat{f}(x(k), u(k); L_D, \mathcal{D}(k)) \right), \tag{12}$$

provides the prediction with the updated dataset $\mathcal{D}(k)$, as long as it is contained in \mathcal{Y}_s . If not, a guaranteed prediction is obtained by projection, according to Property 1. The state-space online model is denoted $\hat{x}_p(k+1) = \hat{F}_p(x(k), u(k))$.

Since new information is added to the dataset, it is sensible to think that it will provide better predictions. As proven in Reference 13, the estimation error of the online model, which will be denoted $e_p(k)$, decreases as the density of the dataset increases. Thus, this model will be used to enhance the closed-loop performance of the plant. However, the guarantee on the bound of the prediction error μ might not be valid for \hat{f}_p , and then it would not be suitable for safety. For this reason, a double model framework is used.

4.2 | Proposed controller

In this section, the proposed MPC, the required design ingredients, and the stability analysis are presented. Additionally, we prove that the closed-loop system is input to state stable w.r.t. the estimation error of the updated prediction model.

In order to ensure the satisfaction of hard constraints in the outputs (ie, $y(k) \in \mathcal{Y}_{\text{hard}} \forall k$), the proposed controller is based on a set of tightened constraints,²⁷ to counteract the effect of the prediction error. This set of constraints varies for each prediction step j , and it is defined as

$$\mathcal{Y}_j = \mathcal{Y}_{\text{hard}} \ominus B(d_j(\mu)), \tag{13}$$

where $d_j(c_1)$ (known as back-off in MPC literature) is obtained by the recursion

$$d_j(c_1) = \sum_{i=1}^j c_i(c_1), \tag{14a}$$

$$c_{j+1}(c_1) = L_D r_j(c_1), \tag{14b}$$

$$r_j(c_1) = \sum_{i=\sigma_j}^j \|c_i(c_1)\|, \tag{14c}$$

with $\sigma_j = \max(1, j - n_a)$, for all $j \in \mathbb{I}_1^N$.

The proposed predictive controller is defined in the following optimization problem, denoted as $P_N(x(k); L_D, \mathcal{D}(0), \mathcal{D}(k))$.^{* †} Its solution yields the control law $\mathbf{u}^*(k) = \kappa_{\text{MPC}}(x(k); L_D, \mathcal{D}(0), \mathcal{D}(k))$.

^{*}The abbreviation for $P_N(x(k); L_D, \mathcal{D}(0), \mathcal{D}(k))$ will be denoted $P_N(k)$.

[†]For the sake of conciseness, we omit the dependence of the functions $V_{N, \ell}$ and V_f with L_D and $\mathcal{D}(0), \mathcal{D}(k)$.

$$\begin{cases}
\min_{\mathbf{u}} & V_N(k) = \sum_{j=0}^{N-1} \ell(\hat{y}_p(j|k), u(j)) + \lambda V_f(\hat{x}_p(N|k)) & (15a) \\
\text{s.t.} & \hat{x}_p(0|k) = x(k) & (15b) \\
& \hat{x}_p(j+1|k) = \hat{F}_p(\hat{x}_p(j|k), u(j)), j \in \mathbb{I}_0^{N-1} & (15c) \\
& \hat{y}_p(j|k) = M\hat{x}_p(j|k), j \in \mathbb{I}_0^{N-1} & (15d)
\end{cases}$$

$$\begin{cases}
& \hat{x}_s(0|k) = x(k) & (15e) \\
& \hat{x}_s(j+1|k) = \hat{F}_s(\hat{x}_s(j|k), u(j)), j \in \mathbb{I}_0^{N-1} & (15f) \\
& \hat{y}_s(j|k) = M\hat{x}_s(j|k), j \in \mathbb{I}_1^N & (15g) \\
& u(j) \in \mathcal{U}, j \in \mathbb{I}_0^{N-1} & (15h) \\
& \hat{y}_s(j|k) \in \mathcal{Y}_j, j \in \mathbb{I}_0^{N-1} & (15i)
\end{cases}$$

where $x(k)$ is the current state, $\mathbf{u} = (u(0), \dots, u(N-1))$ is the future input trajectory, and $\hat{x}_p(j|k)$, $\hat{y}_p(j|k)$, $\hat{x}_s(j|k)$, and $\hat{y}_s(j|k)$ are the state and input trajectories predicted from the current state using the online and safe models, respectively.

Note how the optimization problem reflects the purpose of each model: the performance index is calculated using the online model, while the constraints must be satisfied by the predicted trajectory using the safe model. Notice that there is no terminal constraint in the optimization problem. Instead, stability is achieved by means of a weighting factor $\lambda \geq 1$ for the terminal cost, following the method presented in Manzano et al.¹⁹ To ensure robust stability and constraint satisfaction, the following assumptions must hold:

Assumption 3.

1. The stage cost function $\ell(y, u)$ is a continuous positive definite function for all $y \in \mathcal{Y}_{\text{hard}}$ and $u \in \mathcal{U}$ such that $\ell(y, u) \geq \alpha_y(\|y\|) + \alpha_u(\|u\|)$ and $|\ell(y_1, u) - \ell(y_2, u)| \leq \theta_\ell(\|(y_1, u) - (y_2, u)\|)$, for certain \mathcal{K} -functions $\theta_\ell, \alpha_y, \alpha_u$.
2. There exists a local control law $u = \kappa_f(x)$, a terminal cost function V_f and a set $\Omega_\gamma = \{x : V_f(x) \leq \gamma\} \subseteq \mathbb{R}^{n_x}$, for $\gamma > 0$ such that for all $x \in \Omega_\gamma$, the following conditions holds:

(a)

$$\kappa_f(x) \in \mathcal{U}, \quad (16a)$$

$$Mx \oplus \mathcal{B}(a_N) \subseteq \mathcal{Y}_N, \quad (16b)$$

where a_j is given by the recursion $a_{j+1} = L_D \|a_j\| + \mu$, with $a_1 = \mu$.

- (b) V_f is a continuous positive definite function such that $|V_f(x_1) - V_f(x_2)| \leq \theta_f(\|x_1 - x_2\|)$ and

$$\alpha_f(\|x\|) \leq V_f(x) \leq \beta_f(\|x\|), \quad (17a)$$

$$V_f(\hat{F}_p(x, \kappa_f(x))) - V_f(x) \leq -\ell(Mx, \kappa_f(x)), \quad (17b)$$

for certain \mathcal{K} -functions $\alpha_f, \beta_f, \theta_f$.

Remark 1. Note that (16b) implicitly states a condition on the estimation error bound μ . It can be proven that if μ is small enough to ensure that $\mathcal{B}(a_N) \oplus \mathcal{B}(d_N(\mu)) \subseteq \epsilon \mathcal{Y}$, for $0 < \epsilon < 1$, then (16b) is solvable.

Remark 2. Equation (17b) in Assumption 3 requires that the Lyapunov condition holds for the online prediction model. Since the terminal ingredients must be computed offline, the online model is not available for this design. Taking into account that the safe model can be regarded as a sampled subset of the updated model, since $\mathcal{D}(0) \subseteq \mathcal{D}(k)$, the online model is contained in a difference inclusion centered in the safe model and given by the estimation error of the LACKI method, which is explicitly known. Therefore, robust design methods can be used to calculate the terminal ingredients satisfying Assumption 3.

Definition 1. Making use of the definition of $c_j(c_1)$ and $r_j(c_1)$ in Equation (14a) and Assumption 3, the following function, which is a \mathcal{K} -function for every input, is defined:

$$\theta_a(c_1) = \sum_{j=0}^{N-1} \theta_\ell(\|c_{j+1}(c_1)\|) + \theta_f(r_{N+1}(c_1)). \quad (18)$$

Assumption 4. The set

$$Y = \{x : \ell(Mx, 0) \leq \theta_a(2\mu)\} \quad (19)$$

is contained in Ω_γ .

Definition 2. ϕ is a positive constant such that

$$\ell(Mx, 0) > \phi, \quad \forall x \notin \Omega_\gamma. \quad (20)$$

Similarly to Manzano et al,¹⁹ it can be proven that ϕ is such that $\phi \geq \theta_a(2\mu)$.

The region of feasibility of the problem (15) is denoted X_N , and a level set of its optimal cost is defined as:

$$\Gamma = \{x \in X_N : V_N^*(x) \leq N\phi + \lambda\gamma\}. \quad (21)$$

The control law is given by Algorithm 1, where the updating policy of the online dataset guarantees closed-loop robust stability. At each time step k , the online dataset is updated with the current data point, that is $\mathcal{D}(k) = \mathcal{D}(k-1) \cup (y(k), w(k-1))$, subject to the following triple updating criteria:

1. The current data point must not be *close* to any point already contained in $\mathcal{D}(k-1)$, for a given threshold $\tau \geq 0$. This rule prevents the cardinality of \mathcal{D} from becoming large, yielding the method computationally expensive. A reasonable estimate for τ is the noise level $\bar{\epsilon}$.
2. Adding the new data point must not increase the cost calculated with the shifted input sequence which, together with the control policy and update rule, is defined in Algorithm 1.
3. The new candidate $(y(k), w(k-1))$ must be consistent with the Lipschitz constant obtained in Equation (5). This rule is optional, included so that the online model maintains the properties of the LACKI method.¹³

In the following theorem, it is proven that the proposed control algorithm guarantees that the closed-loop system is input-to-state stable (ISS).

Theorem 1. Consider that Assumptions 2-4 hold, and let $\kappa_{\text{MPC}}(x)$ be the control law derived from the solution of $P_N(k)$ applied using Algorithm 1. Then, for any feasible state $x(0) \in \Gamma$, the system controlled by the control law $u(k) = \kappa_{\text{MPC}}(x(k))$ is input-to-state stable w.r.t. the estimation error $e_p(k)$, and the constraints are fulfilled along the operation, that is, $y(k) \in \mathcal{Y}_{\text{hard}}, \forall k$.

Proof. In this proof, different predicted trajectories will be considered, as shown in Figure 1. These trajectories are obtained predicting from $x(k)$ with $\mathbf{u}^*(k)$ or from $x(k+1)$ with $\bar{\mathbf{u}}(k+1)$, and using the safe or the online models. ■

It is first proven that Γ is an invariant set of the closed-loop system, that is, if $x(k) \in \Gamma$, then $x(k+1) \in \Gamma$. To this end, as it is standard in MPC proofs, it is proven that the shifted trajectory presented in Algorithm 1 is feasible for $x(k+1)$.

Provided that $x(k) \in \Gamma$, due to its definition in Equation (21), it follows that

$$V_N^*(x(k)) \leq N\phi + \lambda\gamma, \quad (22)$$

and therefore it can be proven²⁸ that

$$\hat{x}_p(N|k) \in \Omega_\gamma. \quad (23)$$

Algorithm 1. Online update and control law**while** automatic control is on **do** Read $y(k)$ $x(k) \leftarrow (y(k), \dots, y(k - n_a), u(k - 1), \dots, u(k - n_b))$

▷ State

function SHIFTED SEQUENCE($\bar{\mathbf{u}}$)

▷ Shifted Sequence

 $\bar{u}(j|k) \leftarrow u^*(j + 1|k - 1), j \in \mathbb{I}_0^{N-2}$ $\bar{u}(N|k) \leftarrow \kappa_f(\hat{x}_p(N|k - 1))$ **return** $\bar{\mathbf{u}}$ **end function** $\bar{D}(k) \leftarrow D(k - 1) \cup (y(k), w(k - 1))$

▷ Updated dataset

if $\left(\min_{w_i \in \mathcal{W}_D(k)} \|w(k) - w_i\| < \tau \right)$ **and** $(\bar{V}_N(x(k), \bar{\mathbf{u}}; \bar{D}(k)) \leq \bar{V}_N(x(k), \bar{\mathbf{u}}; D(k - 1)))$ **and** $\left(\max_{i \in \mathbb{I}_1^{N_D(k)}} \frac{|y(k) - y_i| - 2\bar{e}}{\|w(k) - w_i\|} \leq L_D \right)$ **then** $D(k) \leftarrow \bar{D}(k)$

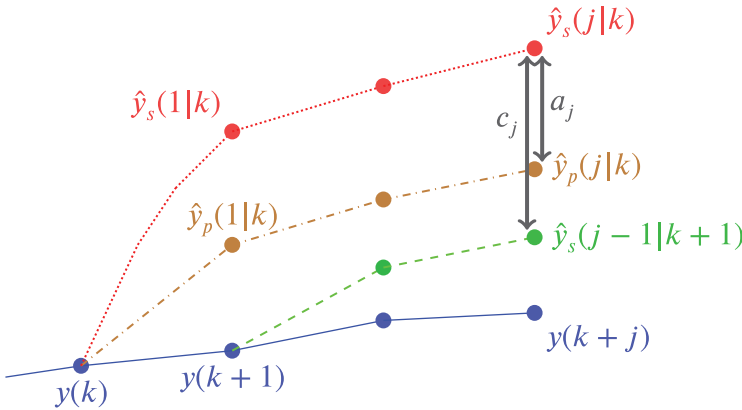
▷ Update

else $D(k) \leftarrow D(k - 1)$

▷ Do not update

end if $\mathbf{u}^*(k) \leftarrow \kappa_{\text{MPC}}(x(k); L_D, D(0), D(k))$ ▷ Solution of $P_N(k)$ $u(k) \leftarrow u^*(0)$

▷ Receding horizon

 Apply $u(k)$ to the system**end while****FIGURE 1** Propagation of the predictions

Feasibility of the shifted sequence $\bar{\mathbf{u}}(k + 1)$ is proven for any $x(k) \in \Gamma$ demonstrating that

1. $\bar{u}(j|k + 1) \in \mathcal{U}, \forall j \in \mathbb{I}_0^{N-1}$
2. $\hat{y}_s(j|k + 1) \in \mathcal{Y}_j, \forall j \in \mathbb{I}_0^{N-1}$.

Constraint (i) holds because $\mathbf{u}^*(k) \in \mathcal{U}$, and since $\hat{x}_p(N|k) \in \Omega_\gamma$, because of Equation (16a), the shifted sequence defined in Algorithm 1 is feasible, that is, $\bar{u}(j|k + 1) \in \mathcal{U}, \forall j \in \mathbb{I}_0^{N-1}$.

To address (ii), the following Lemmas, proven in Appendix A1, are used:

Lemma 1. *The following inequalities hold:*

$$|\hat{y}_p(j - 1|k + 1) - \hat{y}_p(j|k)| \leq c_j(e_p(k)), \quad (24a)$$

$$\|\hat{x}_p(j - 1|k + 1) - \hat{x}_p(j|k)\| \leq r_j(e_p(k)), \quad (24b)$$

where $c_j(c_1)$ and $r_j(r_1)$ are obtained from the recursion (14a).

The bounds (24) hold for the safe model, taking $c_1 = \mu$ instead of $e_p(k)$.

Lemma 2. For all $y \in \mathcal{Y}_j$ and all $\Delta y \in \mathcal{B}(c_j(\mu))$, the sets \mathcal{Y}_j are such that $y + \Delta y \in \mathcal{Y}_{j-1}$.

It is known that $\hat{y}_s(j|k) \in \mathcal{Y}_j$ up to $N - 1$, and that

$$|\hat{y}_s(j|k) - \hat{y}_s(j-1|k+1)| \leq c_j(\mu). \quad (25)$$

Hence, because of Lemma 2, $\hat{y}_s(j|k+1) \in \hat{y}_s(j+1|k) \oplus \mathcal{B}(c_{j+1}(\mu)) \subseteq \mathcal{Y}_{j+1} \oplus \mathcal{B}(c_{j+1}(\mu)) \subseteq \mathcal{Y}_j$. For the last prediction, the fact that $\hat{x}_p(N|k) \in \Omega_\gamma$ implies that $\hat{y}_p(N|k) \in \mathcal{Y}_N$ (Equation (16b)).

Another lemma is introduced (proven in Appendix A1):

Lemma 3. The following inequalities hold:

$$|\hat{y}_p(j|k) - \hat{y}_s(j|k)| \leq a_j, \quad (26)$$

where a_j is given by the recursion

$$a_{j+1} = L_D \|a_j\| + \mu, \quad (27)$$

with $a_1 = \mu$.

Given the definition of $c_j(\mu)$ in Lemma 1 and a_j in Lemma 3, the following bound is obtained:

$$\begin{aligned} |\hat{y}_p(j|k) - \hat{y}_s(j-1|k+1)| &\leq |\hat{y}_p(j|k) - \hat{y}_s(j|k)| + |\hat{y}_s(j|k) - \hat{y}_s(j-1|k+1)| \\ &\leq a_j + c_j(\mu). \end{aligned} \quad (28)$$

Using Lemma 3 and Equation (28), it follows that the error between the predictions of the safe and the online models is bounded, so

$$|\hat{y}_p(N|k) - \hat{y}_s(N-1|k+1)| \leq a_N + c_N(\mu).$$

Then, because of Equation (16b),

$$\hat{y}_s(N-1|k+1) \in M\hat{x}_p(N|k) \oplus \mathcal{B}(a_N) \oplus \mathcal{B}(c_N(\mu)) \subseteq \mathcal{Y}_N \oplus \mathcal{B}(c_N(\mu)) \subseteq \mathcal{Y}_{N-1}, \quad (29)$$

which completes the proof of (ii).

Given the definition of the cost in Equation (15a), the measurements of $y(k)$ and $y(k+1)$, and the optimal and shifted sequences $\mathbf{u}^*(k)$ and $\bar{\mathbf{u}}(k+1)$ (see Algorithm 1), the following equality holds:

$$\begin{aligned} V_N(\hat{x}_p(1|k), \bar{\mathbf{u}}(k+1); \mathcal{D}(k)) - V_N^*(x(k), \mathbf{u}^*(k); \mathcal{D}(k)) &= \ell(\hat{y}_p(N|k), \kappa_f(\hat{x}_p(N|k))) + V_f(\hat{F}_p(\hat{x}_p(N|k), \kappa_f(\hat{x}_p(N|k)))) \\ &\quad - \ell(y(k), u^*(k)) - V_f(\hat{x}_p(N|k)). \end{aligned} \quad (30)$$

Since $\hat{x}_p(N|k) \in \Omega_\gamma$ and taking into account Equation (17b), the following inequality is obtained:

$$V_f(\hat{F}_p(\hat{x}_p(N|k), \kappa_f(\hat{x}_p(N|k)))) - V_f(\hat{x}_p(N|k)) + \ell(\hat{y}_p(N|k), \kappa_f(\hat{x}_p(N|k))) \leq 0, \quad (31)$$

which implies that

$$V_N(\hat{x}_p(1|k), \bar{\mathbf{u}}(k+1); \mathcal{D}(k)) - V_N^*(x(k), \mathbf{u}^*(k); \mathcal{D}(k)) \leq -\ell(y(k), u^*(k)). \quad (32)$$

In addition,

$$\begin{aligned} V_N(x(k+1), \bar{\mathbf{u}}(k+1); \mathcal{D}(k)) - V_N(\hat{x}_p(1|k), \bar{\mathbf{u}}(k+1); \mathcal{D}(k)) &= \sum_{j=0}^{N-1} \ell(\hat{y}_p(j|k+1), \bar{u}(j|k+1)) - \sum_{j=0}^{N-1} \ell(\hat{y}_p(j+1|k), \bar{u}(j|k+1)) \\ &\quad + V_f(\hat{x}_p(N|k)) - V_f(\hat{F}_p(\hat{x}_p(N|k), \kappa_f(\hat{x}_p(N|k)))). \end{aligned} \quad (33)$$

Using Lemma 1 and taking into account continuity of ℓ and V_f , there exists certain \mathcal{K} -functions θ_ℓ and θ_f and a function θ_a such that

$$V_N(x(k+1), \bar{\mathbf{u}}(k+1); \mathcal{D}(k)) - V_N(\hat{x}_p(1|k), \bar{\mathbf{u}}(k+1); \mathcal{D}(k)) \leq \sum_{j=0}^{N-1} \theta_\ell(\|e_{j+1}(e_p(k))\|) + \theta_f(r_{N+1}(e_p(k))) = \theta_a(e_p(k)). \quad (34)$$

Now, merging Equations (32) and (34) yields

$$V_N(x(k+1), \bar{\mathbf{u}}(k+1); \mathcal{D}(k)) - V_N^*(x(k), \mathbf{u}^*(k); \mathcal{D}(k)) \leq -\ell(y(k), u^*(k)) + \theta_a(e_p(k)). \quad (35)$$

Moreover, because of the updating policy of Algorithm 1,

$$V_N(x(k+1), \bar{\mathbf{u}}(k+1); \mathcal{D}(k+1)) \leq V_N(x(k+1), \bar{\mathbf{u}}(k+1); \mathcal{D}(k)), \quad (36)$$

and by optimality,

$$V_N^*(x(k+1); \mathcal{D}(k+1)) \leq \bar{V}_N(x(k+1), \bar{\mathbf{u}}(k+1); \mathcal{D}(k+1)). \quad (37)$$

Summing up,

$$V_N^*(x(k+1), \mathbf{u}^*(k+1); \mathcal{D}(k+1)) - V_N^*(x(k), \mathbf{u}^*(k); \mathcal{D}(k)) \leq -\ell(y(k), u^*(k)) + \theta_a(e_p(k)). \quad (38)$$

Next, recursive feasibility is proven ensuring that $x(k+1) \in \Gamma$, for which it is necessary that $V_N^*(x(k+1); \mathcal{D}(k+1)) \leq N\phi + \lambda\gamma$. Consider that $x(k) \notin \Upsilon$. Then $\ell(y(k), u^*(k)) > \theta_a(2\mu)$, provided that the maximum online prediction error satisfies $e_p \leq 2\mu$, given the definition of \hat{f}_p in Equation (12). Therefore, using Equations (22) and (38), it is derived that $x(k+1) \in \Gamma$.

If $x(k) \in \Upsilon$, then $x(k) \in \Omega_\gamma$ (Assumption 4). From standard arguments of MPC, it follows that

$$V_N^*(x(k)) \leq \lambda V_f(x(k)) \leq \lambda\gamma, \quad (39)$$

which taking into account $\theta_a(e_p(k)) \leq \phi$ implies that

$$V_N^*(x(k+1), \mathbf{u}^*(k+1); \mathcal{D}(k+1)) \leq \phi + \lambda\gamma - \ell(y(k), u(k)) \leq N\phi + \lambda\gamma, \quad (40)$$

and hence $x(k+1) \in \Gamma$.

To prove input-to-state stability, Equation (38) and the continuity of the stage cost is used to derive that

$$V_N^*(x(k+1); \mathcal{D}(k+1)) - V_N^*(x(k); \mathcal{D}(k)) \leq -\alpha_y(\|y(k)\|) - \alpha_u(\|u(k)\|) + \theta_a(e_p(k)). \quad (41)$$

Defining

$$W(x(k); \mathcal{D}(k)) = \sum_{j=0}^n V_N^*(x(k-j); \mathcal{D}(k-j)), \quad (42)$$

with $n = \max(n_a, n_b + 1)$, it follows that $\alpha_x(\|x(k)\|) \leq \sum_{j=0}^n \alpha_y(\|y(k-j)\|) + \alpha_u(\|u(k-j-1)\|)$,¹⁸ and hence

$$W(x(k+1); \mathcal{D}(k+1)) - W(x(k); \mathcal{D}(k)) \leq -\alpha_x(\|x(k)\|) + (n+1)\theta_a(\max_{j=0, \dots, n} e_p(k-j)), \quad (43a)$$

$$\alpha_1(\|x\|) \leq W(x(k)) \leq \alpha_2(\|x\|). \quad (43b)$$

Hence, $W(x(k))$ is an ISS Lyapunov function.

The proposed MPC is proven to be stable and to robustly satisfy the output constraints under certain assumptions. Note that it lacks a terminal constraint, avoiding the calculation of a robust invariant set, unlike what is common in the design of robust MPCs. The performance of the controller is enhanced by the inclusion of fresh data from the operation of the plant, as it will be illustrated in the case study of the next section. Not only does the behavior improve, but also the convergence rates are decreased as the updated estimation error decreases, because the closed-loop system is proven to be input-to-state stable with respect to the estimation error of the updated model. This error decreases in average with time, for increasing datasets.¹³

If the terminal cost is designed offline with the safe model, the following corollary proves that input-to-state practical stability can still be derived:

Corollary 1. *Consider that Assumptions 2 and 4 hold, and that Assumption 3 holds replacing Equation (17b) by*

$$V_f(\hat{F}_s(x, \kappa_f(x))) - V_f(x) \leq -\ell(Mx, \kappa_f(x)), \quad (44)$$

that is, satisfying the Lyapunov condition for the safe model. Let $\kappa_{\text{MPC}}(x)$ be the control law derived from the solution of $P_N(k)$ applied using Algorithm 1. Then, for any feasible state $x(0) \in \Gamma$, the system controlled by the control law $u(k) = \kappa_{\text{MPC}}(x(k))$ is input-to-state stable w.r.t. $e_p(k)$ and μ . Besides the constraints are fulfilled along the operation, that is, $y(k) \in \mathcal{Y}_{\text{hard}}$.

Proof. The proof is similar to the one of Theorem 1, but replacing Equation (31) as follows. First, see that from the uniform continuity of V_f and Lemma 3 (Equation (A7)), we have that

$$V_f(\hat{F}_p(\hat{x}_p(N|k), \kappa_f(\hat{x}_p(N|k)))) - V_f(\hat{F}_s(\hat{x}_p(N|k), \kappa_f(\hat{x}_p(N|k))))$$

is upper bounded by

$$\theta_f(\|\hat{F}_p(\hat{x}_p(N|k), \kappa_f(\hat{x}_p(N|k))) - \hat{F}_s(\hat{x}_p(N|k), \kappa_f(\hat{x}_p(N|k)))\|),$$

which is equal to

$$\theta_f(\|\hat{f}_p(\hat{x}_p(N|k), \kappa_f(\hat{x}_p(N|k))) - \hat{f}_s(\hat{x}_p(N|k), \kappa_f(\hat{x}_p(N|k)))\|) \leq \theta_f(\|\mu\|).$$

Taking this into account, Equation (31) can be rewritten in this case as

$$V_f(\hat{F}_p(\hat{x}_p(N|k), \kappa_f(\hat{x}_p(N|k)))) - V_f(\hat{x}_p(N|k)) + \ell(\hat{y}_p(N|k), \kappa_f(\hat{x}_p(N|k))) \leq \theta_f(\|\mu\|). \quad (45)$$

Following the subsequent steps, Equation (43a) would be rewritten as follows, given $n = \max(n_a, n_b + 1)$:

$$W(x(k+1); D(k+1)) - W(x(k); D(k)) \leq -\alpha_x(\|x(k)\|) + (n+1)\theta_a(\max_{j=0, \dots, n} e_p(k-j)) + \theta_f(\|\mu\|). \quad (46)$$

Consequently, the controlled system would be ISS w.r.t. e_p and μ . ■

Remark 3. The domain of attraction of the controller is defined by the feasibility region Γ in Equation (21). This set increases as the weighting factor λ increases.¹⁸ Hence, λ can be chosen arbitrarily big in order to enlarge the domain of attraction of the MPC.

Remark 4. Soft constraints in the outputs, $y(k) \in \mathcal{Y}_{\text{soft}}$, may also be considered, so that the performance will be penalized if the outputs go beyond $\mathcal{Y}_{\text{soft}}$. To account for the soft constraint, we will make use of barrier functions, adding a penalizing term to the stage cost. Hence, we can describe the stage cost $\ell(y, u)$ as the standard cost to track the reference plus the barrier function:¹⁸

$$\ell(y, u) = \ell_t(y, u) + \ell_b(y), \quad (47)$$

where $\ell_b(y) = \psi \rho(y, \mathcal{Y}_{\text{soft}})$. ψ is a large constant, and $\rho(y, \mathcal{Y}_{\text{soft}})$ is a measurement of the distance of y to the set $\mathcal{Y}_{\text{soft}}$.

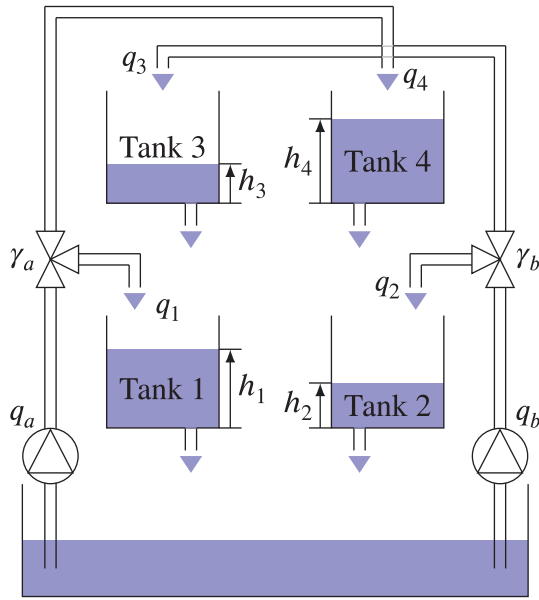


FIGURE 2 Scheme of the quadruple-tank process

5 | CASE STUDY

The proposed controller will be applied to the quadruple-tank process presented in Johansson,²⁹ reproduced in Figure 2. Here, two upper tanks discharge on two lower tanks (tank number 3 into tank number 1, and tank 4 into 2). Two pumps send water through two 3-ways-valves, such that pump *a* feeds a fraction γ_a to tank 1 and the rest to tank 4, while a flow $q_b\gamma_b$ goes to tank 2 and $q_b(1 - \gamma_b)$ to tank 3.

The control inputs are the flows, q_a, q_b (m^3/h); the states are the heights of the liquid in the tanks, denoted h_i (m), $i \in \mathbb{I}_1^4$; and the measurable outputs are the heights of the lower tanks, h_1, h_2 . The dynamics of the plant are modeled by the following set of differential equations:

$$A_1 \frac{dh_1(t)}{dt} = -a_1 \sqrt{2gh_1(t)} + a_3 \sqrt{2gh_3(t)} + \gamma_a \frac{q_a(t)}{3600} \quad (48a)$$

$$A_2 \frac{dh_2(t)}{dt} = -a_2 \sqrt{2gh_2(t)} + a_4 \sqrt{2gh_4(t)} + \gamma_b \frac{q_b(t)}{3600} \quad (48b)$$

$$A_3 \frac{dh_3(t)}{dt} = -a_3 \sqrt{2gh_3(t)} + (1 - \gamma_b) \frac{q_b(t)}{3600} \quad (48c)$$

$$A_4 \frac{dh_4(t)}{dt} = -a_4 \sqrt{2gh_4(t)} + (1 - \gamma_a) \frac{q_a(t)}{3600}, \quad (48d)$$

where A_i (m^2) denotes the area of tank i and a_i (m^2) is the equivalent area of the hole of tank i . The parameters of the model are given in Table 1. The constraints are given by $1.5 \leq q_a \leq 1.9$, $1.4 \leq q_b \leq 1.8$ m^3/h and $0.38 \leq h_1 \leq 0.62$, $0.45 \leq h_2 \leq 0.73$ m.

The sampling time is set to $\tau_s/30 = 5$ s, where τ_s stands for the mean settling time of a sequence of steps applied to the system. A set of experiments is carried out to obtain a training dataset. The experiments are designed using the methodologies presented in Rivera and Jun³⁰ to identify the dynamics of a system: a sequence of chirp signals covering the workspace are applied to generate the original dataset containing trajectories of inputs and outputs, with $N_0 = 5030$ data points. This dataset $\mathcal{D}(0)$ is used for predicting. This simulation is represented in Figure 3. The output sensors introduce noise that follows an uniform distribution with $\bar{\epsilon}$ equal to 1% of the measurement.

In addition, several tests with random input signals (square signals with random mean, amplitude and frequency) are carried out in order to obtain datasets for cross-validation, as represented partially in Figure 4.

The signals are scaled between 0 and 1 w.r.t. the span of the constraints. The regressor $w(k)$ is constructed for different values of n_a and n_b . Cross validation tests are used to estimate the prediction error, which as shown in Figure 5, is minimized for $n_a = 2$ and $n_b = 3$. This results in $\mu = [0.98 \ 1.24](\text{cm})$, for which $L_{\mathcal{D}} = [1.231 \ 0.836]$.

TABLE 1 Parameters of the system

Param.	Definition	Value
A	Area of the four tanks	0.03 m^2
a_1	Eq. area of the hole of tank 1	$1.31 \times 10^{-4} \text{ m}^2$
a_2	Eq. area of the hole of tank 2	$1.51 \times 10^{-4} \text{ m}^2$
a_3	Eq. area of the hole of tank 3	$9.27 \times 10^{-5} \text{ m}^2$
a_4	Eq. area of the hole of tank 4	$8.82 \times 10^{-5} \text{ m}^2$
γ_a	Fraction of three-ways valve a	0.3
γ_b	Fraction of three-ways valve b	0.4
g	Gravity acceleration	9.8 m/s^2

FIGURE 3 Chirp signals applied

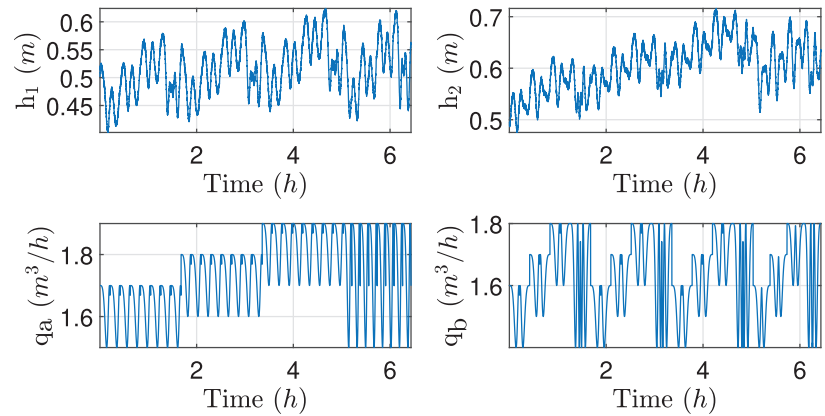
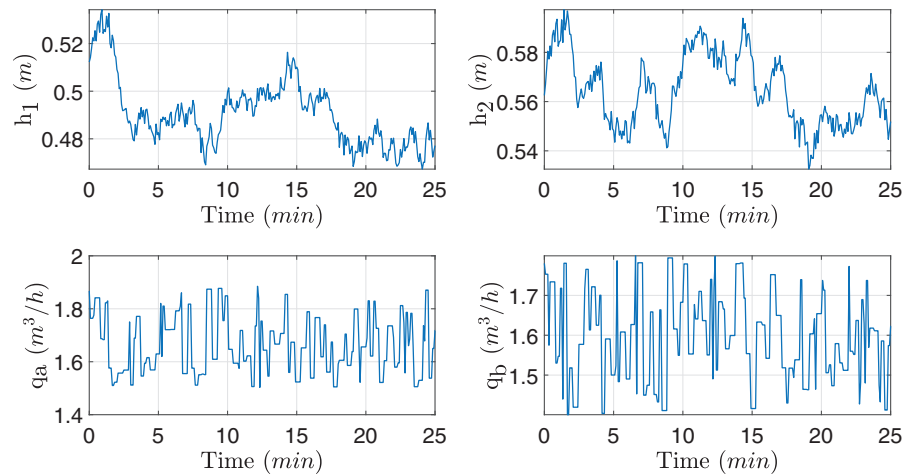


FIGURE 4 PRB signals applied



Then, the MPC presented in Equation (15) is applied with $N = 4$, $\lambda = 10$, and $\tau = 0.01$. The resulting back-off for the safe model is $d_4 = [9.956.43](\text{cm})$, and $a_4 = [5.533.86](\text{cm})$. The stage and terminal cost are defined as follows:

$$\ell(y, u) = \|y - y^{\text{ref}}\|_Q + \|u - u^{\text{ref}}\|_R + \psi \left(1 - \exp \left(\frac{-\max(y - y^s, 0)}{\epsilon} \right) \right) \quad (49)$$

$$V_f(x) = \|x - x^{\text{ref}}\|_P, \quad (50)$$

where the height of the second tank is penalized with $\psi = 999$ if it goes beyond $y^s = 0.61 \text{ m}$, with $\epsilon = 3 \times 10^{-3}$. Q is set to $100I_2$, $R = I_2$, and P is obtained by a LQR linearizing the NARX model around the reference. The back-off of the tightened constraints results in $\mathcal{Y}_N = \{y : [0.48 \ 0.514] \leq y \leq [0.52 \ 0.666](\text{m})\}$.

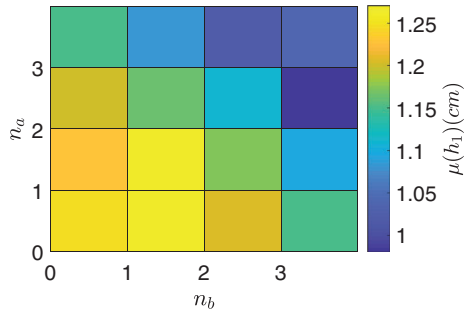


FIGURE 5 Maximum prediction error of h_1 (cm) for different values of n_a, n_b

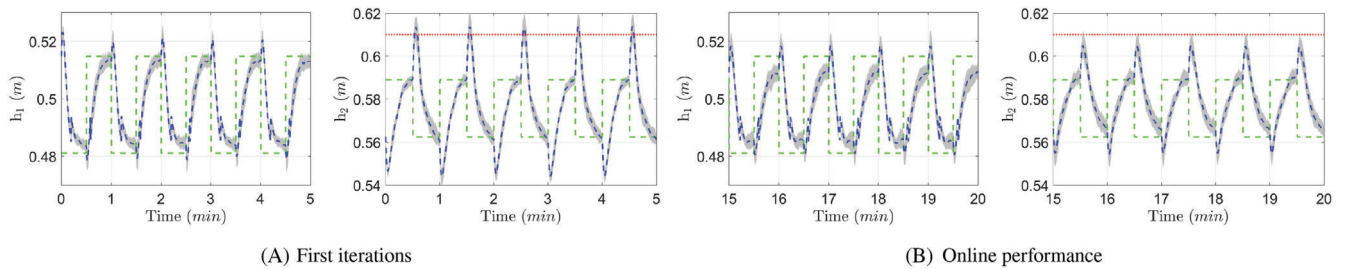


FIGURE 6 Online learning MPC applied to a quadruple tank process

In the first set of simulations, the reference alternates every 5 seconds between $y_1^{\text{ref}} = [0.481 \ 0.589]$ and $y_2^{\text{ref}} = [0.515 \ 0.562]$. The parameters γ and ϕ are obtained as in Rawlings & Mayne,²⁷ resulting in $\gamma = 11698$ and $\phi = 5.8359 \times 10^7$, satisfying the assumptions of the controller.

The results are shown in Figure 6 for 100 simulations subject to the sensors' random noise. The red dotted line represents the soft constraint, the green dashed line represents the reference, the gray band represents the set of signals, and the blue line represents its mean.

Figure 6 illustrates the triple contribution of the article. The set of tightened constraints, applied to the safe model, prevents the closed-loop system from violating the hard constraints (out of scope in the figure). The system is ISS and converges to the reference. Besides, the performance is enhanced by adding the online data to the prediction model used to minimize the cost. For instance, note that after 15 minutes, the model is able to learn the dynamics well enough to avoid the penalty of the soft constraint (red dotted line).

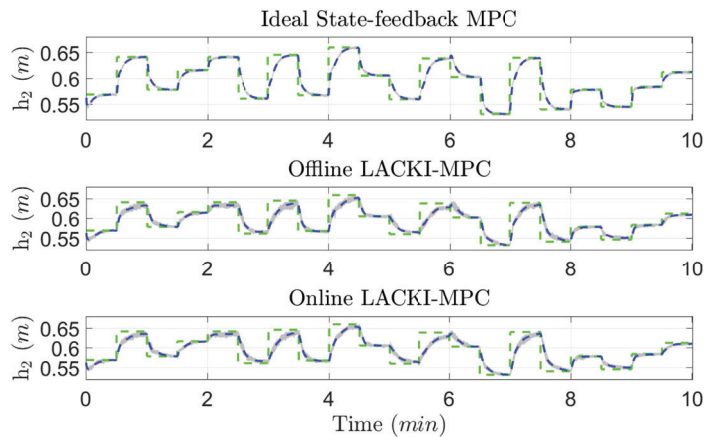
In the simulation presented, the reference alternated between two values. However, the proposed controller can be applied to any admissible reference. In a second set of simulations, represented in Figure 7, the reference changes randomly between different reachable values. In addition, in order to compare the proposed controller with other strategies, we apply the same setting to: (i) an MPC based on the ideal state-feedback model (identical to the plant), given by the set of ODEs of Equation (48a), and (ii) the LACKI model without updating the dataset, that is, the offline version. We compare the three controllers in Figure 7, representing the overall cost for 100 simulations, by means of the closed-loop performance index, which is defined as

$$\Phi = \sum_{i=1}^{t_{\text{sim}}} \ell(y(i), u(i)). \quad (51)$$

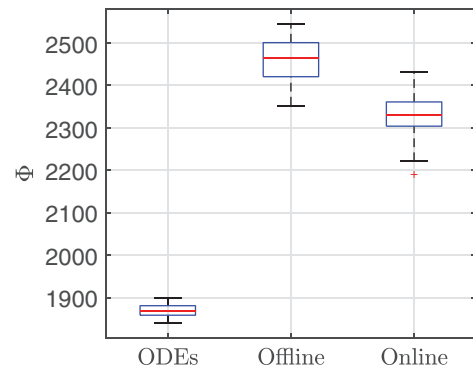
In the results, it can be seen that the performance improves as more data is obtained for the online prediction model, and that the constraints are satisfied along the operation.

6 | CONCLUSIONS

An output-feedback online learning predictive controller based on KI was proposed. It exploits the data collected online to enhance the performance. Under the assumption of known bounded errors, the controller guarantees robust stability



(A) Height of the second tank



(B) Boxplot comparison of the performance index

FIGURE 7 Same setting applied to 100 simulations of three controllers, subject to random noise, in which the models are the set of ODEs, the offline version of the LACKI, and the online one

and constraint satisfaction while being able to improve its performance online using the data obtained from new measurements. The proposed controller does not require a terminal constraint. The main contributions of the article were illustrated using a simulated quadruple tank process.

ACKNOWLEDGMENTS

The authors would like to thank Pablo Krupa for his helpful comments during the elaboration of this paper. This work was supported by the MINECO-Spain and FEDER funds under project DPI2016-76493-C3-1-R and the VI-PPIT of the University of Seville.

ORCID

J. M. Manzano  <https://orcid.org/0000-0003-2932-5581>

D. Limon  <https://orcid.org/0000-0001-9334-7289>

REFERENCES

1. Fisac JF, Akametalu AK, Zeilinger MN, Kaynama S, Gillula J, Tomlin CJ. A general safety framework for learning-based control in uncertain robotic systems. *IEEE Trans Automat Control*. 2018;64(7):2737-2752.
2. Piga D, Formentin S, Bemporad A. Direct data-driven control of constrained systems. *IEEE Trans Control Syst Technol*. 2017;26(4):1422-1429.
3. Salvador JR, Ramirez DR, Alamo T, Muñoz de la Peña D. Offset free data driven control: application to a process control trainer. *IET Control Theory Appl*. 2019;13(18):3096-3106.
4. Salvador JR, Muñoz de la Peña D, Ramirez DR, Alamo T. Predictive control of a water distribution system based on process historian data. *Opt Control Appl Methods*. 2019;41(2):571-586.
5. Maiworm M, Limon D, Manzano JM, Findeisen R. Stability of Gaussian process learning based output feedback model predictive control. *IFAC-PapersOnLine*. 2018;51(20):455-461.
6. Smarra F, Jain A, Ambrosini D, D'Innocenzo A, Mangharam R. Data-driven model predictive control using random forests for building energy optimization and climate control. *Appl Energy*. 2018;226:1252-1272.
7. Hewing L, Wabersich KP, Menner M, Zeilinger MN. Learning-based model predictive control: toward safe learning in control. *Ann Rev Control Robotics Autonom Syst*. 2020;3:269-296.
8. Sutton RS, Barto AG. *Introduction to Reinforcement Learning*. Cambridge, MA: MIT Press; 1998.
9. Wittenmark B. *Adaptive Dual Control Methods: An Overview*. Amsterdam, Netherlands: Elsevier; 1995:67-72.
10. Benosman M. Model-based vs data-driven adaptive control: an overview. *Int J Adapt Control Signal Process*. 2018;32(5):753-776.
11. Beliaikov G. Interpolation of Lipschitz functions. *J Comput Appl Math*. 2006;196(1):20-44.
12. Sukharev AG. Optimal method of constructing best uniform approximations for functions of a certain class. *USSR Comput Math Math Phys*. 1978;18(2):21-31.
13. Calliess JP. Lazily adapted constant kinky inference for nonparametric regression and model-reference adaptive control; 2016. arXiv preprint arXiv:1701.00178.

14. Milanese M, Novara C. Set membership identification of nonlinear systems. *Automatica*. 2004;40(6):957-975.
15. Calliess JP. Lipschitz optimisation for Lipschitz interpolation. Paper presented at: Proceedings of the American Control Conference (ACC); vol 2017, 2017:3141-3146; IEEE.
16. Canale M, Fagiano L, Signorile MC. Nonlinear model predictive control from data: a set membership approach. *Int J Robust Nonlinear Control*. 2014;24(1):123-139.
17. Calliess JP. Conservative Decision-Making and Inference in Uncertain Dynamical Systems [PhD thesis]. University of Oxford; 2014.
18. Manzano JM, Limon D, Muñoz de la Peña D, Calliess J-P. Output feedback MPC based on smoothed projected kinky inference. *IET Control Theory Appl*. 2019;13(6):795-805.
19. Manzano JM, Limon D, Muñoz de la Peña D, Calliess J-P. Robust learning-based MPC for nonlinear constrained systems. *Automatica*. 2020;117:108948.
20. Mitchell TM. Machine learning and data mining. *Commun ACM*. 1999;42(11):30-36.
21. Di Cairano S, Bernardini D, Bemporad A, Kolmanovsky IV. Stochastic MPC with learning for driver-predictive vehicle control and its application to HEV energy management. *IEEE Trans Control Systems Technol*. 2013;22(3):1018-1031.
22. Ostafew CJ, Schoellig AP, Barfoot TD, Collier J. Learning-based nonlinear model predictive control to improve vision-based mobile robot path tracking. *J Field Robot*. 2016;33(1):133-152.
23. Aswani A, Gonzalez H, Sastry SS, Tomlin C. Provably safe and robust learning-based model predictive control. *Automatica*. 2013;49(5):1216-1226.
24. Limon D, Calliess J, Maciejowski JM. Learning-based nonlinear model predictive control. *IFAC-PapersOnLine*. 2017;50(1):7769-7776.
25. Levin AU, Narendra KS. *Identification of Nonlinear Dynamical Systems Using Neural Networks*. Amsterdam, Netherlands: Elsevier; 1997:129-160.
26. Limon D, Alamo T, Raimondo DM, et al. Input-to-state stability: an unifying framework for robust model predictive control. *Nonlinear Model Predictive Control*. Berlin, Heidelberg: Springer; 2008.
27. Rawlings JB, Mayne DQ. *Model Predictive Control: Theory and Design*. 1st ed. Madison, Wisconsin: Nob-Hill Publishing; 2009.
28. Limon D, Alamo T, Salas F, Camacho EF. On the stability of MPC without terminal constraint. *IEEE Trans Automat Control*. 2006;42:832-836.
29. Johansson KH. The quadruple-tank process. *IEEE Trans Control Syst Technol*. 2000;8:456-465.
30. Rivera DE, Jun KS. An integrated identification and control design methodology for multivariable process system applications. *IEEE Control Syst*. 2000;20(3):25-37.

How to cite this article: Manzano JM, Muñoz de la Peña D, Calliess J, Limon D. Online learning constrained model predictive control based on double prediction. *Int J Robust Nonlinear Control*. 2021;31:8813–8829. <https://doi.org/10.1002/rnc.5124>

APPENDIX A. PROOFS OF THE LEMMAS

Proof of Lemma 1. Given the triangular inequality that applies to the definition of the state vector x in Equation (2),

$$\|x(k+1) - \hat{x}_p(1|k)\| \leq \|y(k+1) - \hat{y}_p(1|k)\| = \|e_p(k)\| = \|c_1\| = r_1. \quad (\text{A1})$$

Assuming that $r_{j-1}(c_1)$ and $c_{j-1}(c_1)$ are known, we can derive, from the Lipschitz continuity of \hat{f}_p , that

$$\begin{aligned} |\hat{y}_p(j-1|k+1) - \hat{y}_p(j|k)| &= |\hat{f}_p(\hat{x}_p(j-2|k+1), u(k+j-1)) - \hat{f}_p(\hat{x}_p(j-1|k), u(k+j-1))| \\ &\leq L_D \|\hat{x}_p(j-2|k+1) - \hat{x}_p(j-1|k)\| \\ &\leq L_D r_{j-1} = c_j. \end{aligned} \quad (\text{A2})$$

The error of the estimated state can be bounded as follows, provided that $\hat{x}_p(j|k)$ contains $\min(n_a + 1, j)$ estimated outputs (and real measurements otherwise)

$$\|\hat{x}_p(j-1|k+1) - \hat{x}_p(j|k)\| \leq \|\hat{y}_p(j-1|k+1) - \hat{y}_p(j|k)\| + \dots + \|y(k+1) - \hat{y}_p(j|k)\| \leq \sum_{i=\sigma_j}^j \|c_i\| = r_j. \quad (\text{A3})$$

■

Note that the same holds if the safe model is used, with \hat{x}_s and \hat{y}_s , being $c_1 = \mu$.

Proof of Lemma 2. First, it is proven that $\mathcal{B}(\delta_1) \oplus \mathcal{B}(\delta_2) \subseteq \mathcal{B}(\delta_1 + \delta_2)$. Indeed, for all $y = y_1 + y_2$, with $y_1 \in \mathcal{B}(\delta_1)$ and $y_2 \in \mathcal{B}(\delta_2)$, we have that

$$\|y\| \leq \|y_1\| + \|y_2\| \leq \delta_1 + \delta_2, \quad (\text{A4})$$

and thus $y \in \mathcal{B}(\delta_1 + \delta_2)$.

Since (for any c_1) $d_j = c_j + d_{j-1}$, it is inferred that $\mathcal{B}(d_{j-1}) \oplus \mathcal{B}(c_j) \subseteq \mathcal{B}(d_j)$, and then,

$$\mathcal{Y}_j = \mathcal{Y}_{\text{hard}} \ominus \mathcal{B}(d_j(\mu)) \subseteq \mathcal{Y}_{\text{hard}} \ominus \mathcal{B}(d_{j-1}(\mu)) \ominus \mathcal{B}(c_j(\mu)) = \mathcal{Y}_{j-1} \ominus \mathcal{B}(c_j(\mu)). \quad (\text{A5})$$

Given that $\Delta y \in \mathcal{B}(c_j(\mu))$ and $y \in \mathcal{Y}_j$, it is derived that

$$y + \Delta y \in \mathcal{Y}_j \oplus \mathcal{B}(c_j(\mu)) \subseteq \mathcal{Y}_{j-1} \ominus \mathcal{B}(c_j(\mu)) \oplus \mathcal{B}(c_j(\mu)) \subseteq \mathcal{Y}_{j-1}. \quad (\text{A6})$$

■

Proof of Lemma 3. This lemma is proven by recursion. For $j=1$, the condition holds, given the definition of \hat{f}_p in Equation (12):

$$|\hat{y}_p(1|k) - \hat{y}_s(1|k)| \leq a_1 = \mu. \quad (\text{A7})$$

Then, assuming that a_{j-1} is known, we can obtain.

$$|\hat{y}_s(j|k) - \hat{y}_p(j|k)| = |\hat{f}_s(\hat{x}_s(j-1|k), u(j-1)) - \hat{f}_p(\hat{x}_p(j-1|k), u(j-1))| \quad (\text{A8a})$$

$$\begin{aligned} &\leq |\hat{f}_s(\hat{x}_s(j-1|k), u(j-1)) - \hat{f}_s(\hat{x}_p(j-1|k), u(j-1))| \\ &\quad + |\hat{f}_s(\hat{x}_p(j-1|k), u(j-1)) - \hat{f}_p(\hat{x}_p(j-1|k), u(j-1))| \end{aligned} \quad (\text{A8b})$$

$$\leq L_D \|\hat{y}_s(j-1|k) - \hat{y}_p(j-1|k)\| + \mu \quad (\text{A8c})$$

$$= L_D \|a_{j-1}\| + \mu = a_j. \quad (\text{A8d})$$

■