Received: 24 July 2018 | Revised: 17 September 2019 | Accepted: 28 October 2019
DOI: 10.1002/oca.2559

RESEARCH ARTICLE

WILEY

# Predictive control of a water distribution system based on process historian data

**Jose R. Salvador** | **D. Muñoz de la Peña** | **D.R. Ramirez** | **T. Alamo**

Departamento de Ingeniería de Sistemas y Automática, Escuela Técnica Superior de Ingeniería, Universidad de Sevilla, Sevilla, Spain

**Correspondence**
Jose R. Salvador, Departamento de Ingeniería de Sistemas y Automática, Escuela Técnica Superior de Ingeniería, Universidad de Sevilla, 41092 Sevilla, Spain.
Email: jrsalvador@us.es

**Funding information**
MINECO-Spain; FEDER Funds, Grant/Award Number: DPI2013-48243-C2-2-R and DPI2016-76493-C3-1-R; University of Seville, Grant/Award Number: 2014/425

**Summary**

In this paper, a heuristic historian data-based predictive control strategy is presented and used to control a water distribution system simulated using the EPANET software, in particular, the Richmond water distribution system. The control actions are computed based on past historian data. The historian stores closed-loop operation data of the process with different controllers used in the past, which may not provide sufficient information for a precise system nor controller identification. The proposed predictive controller computes the current control actions as a weighted sum of past control actions so that an estimation of the performance cost over a prediction horizon is minimized. Only a subset of the past control actions in the historian close to the current state of the process is considered in the current control computations to carry out a local linearization. This predictive strategy is well suited to control applications of large and complex processes for which it is difficult to carry out identification experiments such as water distribution systems. In the application example, the trajectories of a set of relay controllers are used through the proposed approach to take into account pressure constraints and periodic references.

**KEYWORDS**

database, historian data-based control, predictive control, water distribution system

## 1 | INTRODUCTION

In optimal and model predictive control (MPC), the use of a model is a key concept. While there are many applications in which obtaining a prediction model is not a problem using standard identification techniques,[1] there are also large and complex processes for which the task of identifying a good model can be very difficult. Moreover, in those cases, the resulting model (if identified) may be too complex to be used with most techniques, although there are many different strategies to simplify the problems such as linear-convex and linear-quadratic convex approximations[2,3] or, more recently, learning-based approximations (see the work of Karg and Lucia[4] and references therein). This situation appears in the control of large infrastructures such as water distribution networks in which simplified models are often used.[5-7]

Nowadays, there has been an increasing interest of the control community on developing controllers based on machine learning techniques because of the recent theoretical and technological groundbreaking advances in this field. There are several data-driven approaches in optimal control such as the work of Zhang et al[8] in which a neural network is used to infer a model which is used later in an approximate dynamic programming problem with a robustifying term. A similar approach has been applied to water-gas shift reaction systems in the work of Wei and Liu.[9] Other data-driven techniques like J-learning and Q-learning have been applied to approximate dynamic programming in the work of Lee and Lee.[10]

 © 2019 John Wiley & Sons, Ltd. 

Data-based learning techniques have been applied to other aspects of optimal control such as predicting the evolution of control targets.[11] Furthermore, learning from data has been also used to improve optimal controllers in iterative learning control schemes.[12,13] All these works use the data, in different ways, in the solution of the corresponding approximate dynamic programming problem. However, in this paper, we focus on applying machine learning techniques to MPC without using a dynamic programming formulation.

In this setting, a widely used approach is to derive an explicit model from data to later use it in the controller. Regression trees and ensemble learning have been used by Jain et al[14] to obtain a prediction model from data. A major breakthrough in system identification was the developing of subspace identification methods (see the works of Van Overschee and De Moor[15-17]), which have also been used in the context of data-driven predictive control.[18] A different technique used by Canale et al[19] based on nonlinear set membership,[20] was used to obtain an approximate model with a bound on the worst-case model error that can be used to infer closed-loop-stability properties. Prediction models are also inferred from experimental data of inputs and outputs of the plant in the work of Limon et al.[21] Prediction methods can also return an interval obtained from historic input-output measurements that bounds the system output as in the work of Bravo et al.[22]

A different approach is to completely avoid the model estimation phase. Favoreel et al[23] used this approach to derive, by means of matrix decompositions, an LQG control law directly from data. Model-free data-driven MPC has been presented by Piga et al,[24] in which no model is explicitly obtained from data and a hierarchical structure with an inner linear controller is used. Other strategy without explicit model was presented by Tanaskovic et al,[25] where past data obtained from the system are used off-line to guarantee a predictable closed-loop behavior, whereas on-line collected data is used to adapt the controller.

Finally, a fourth approach would be the use of some form of machine learning technique to learn the controller directly from data, like in the work of Fagiano and Novara[26] where a $l_1$-norm regularized learning algorithm based on convex programming was presented, or to solve iteratively infinite horizon control problems by using approximate dynamic programming[27,28] or approximate Q-learning techniques[29] in which the direct computation of the performance function is avoided by using a suitable approximation.[30]

Most of these approaches assume that, although the model is unknown, appropriate data for identifying the system dynamics or a specific control law is available, either from appropriate identification experiments or through the extensive use of a simulator, which may not be possible for some applications. In this paper,* however, we do not assume that the trajectories stored in the database have been chosen or designed to provide sufficient information for a precise identification of the system dynamics or the corresponding control law. This is the case when historian trajectories are available from the closed-loop operation of the system under different conditions, controllers, and even control objectives. This issue limits the applicability of standard identification procedures in which the quality of the information available greatly affects the quality of the resulting identified models.

In this work, we take a different approach. Instead of first identifying a model and then find the best future input trajectory based on this model, we limit the set of possible input trajectories to a convex combination of past trajectories with an initial state close (in some sense) to the current state of the system. We will then use the information in the database to estimate the performance of the chosen trajectory using a heuristic linear approximation based on a particular extreme case of direct weight optimization methods[31] that results in the solution of a quadratic programming problem (or linear programming if only regulation around a set point is considered). This is a predictive control strategy in the sense that it uses the future in the past to predict the future evolution of the process. Note that this concept is very general and it does not impose almost any condition on the closed-loop trajectories stored in the database, although it would be logical to consider only those that resulted in a good control performance.

The proposed control strategy can be applied to large processes, eg, water distribution systems. Water distribution systems pose control problems due to their size, diverse nature of the process, and manipulated variables and disturbance rich operating conditions. Motivated by these issues, the proposed controller has been applied to the Richmond case study[32] using the EPANET software, assuming that the only historian available is obtained from a set of relay controllers. Different scenarios are considered, including regulation, periodic reference tracking, and pressure constraints.

This paper is organized as follows. In Section 2, the problem formulation is presented. The controller formulation is introduced in Section 3. The results of the application of the controller to the water distribution system case study are shown in Section 4. This paper ends with some conclusions in Section 5.

*A preliminary version of this paper has been presented at the European Control Conference, Limassol, Cyprus, June 2018.

## 2 | PROBLEM FORMULATION

The system considered throughout this paper will be represented by a discrete-time invariant system

$$x(k+1) = f(x(k), u(k)), \tag{1}$$

where $x \subset \mathbb{R}^{n_x}$ is the state, $u \subset \mathbb{R}^{n_u}$ is the input, and $f$ is the unknown transition function.

A regulation problem is considered along this paper; thus, the control objective is to regulate the system to the origin while minimizing the performance defined by a stage cost $\ell(\cdot,\cdot)$. Standard MPC[33,34] is based on solving an optimal control finite horizon problem in which the cost of a predicted trajectory of length $N$ is minimized at each time step to obtain an optimal control sequence that is applied in a receding horizon manner. In order to approximate an infinite horizon problem, often, a terminal cost function $F(\cdot)$ is also considered, which leads to the finite horizon performance cost $J : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u N} \rightarrow \mathbb{R}$:

$$J(x(k), U) = \sum_{i=0}^{N-1} \ell(x(k+i|k), u(k+i|k)) + F(x(k+N|k)), \tag{2}$$

where $x(k+i|k)$ is the predicted state obtained applying the future input sequence $u(k+i|k)$ with $i = \{0, \dots, N-1\}$ from the initial state $x(k)$, $\ell(\cdot,\cdot)$ and $F(\cdot)$ are convex positive definite functions, $N$ defines the prediction horizon, and

$$U = \begin{bmatrix} u(k|k) \\ u(k+1|k) \\ \vdots \\ u(k+N-1|k) \end{bmatrix} \tag{3}$$

is the optimization variable.

Standard MPC solves an optimization problem based on the model. In this work, as the function that models the system $f$ is unknown, a database will be used to obtain a straightforward estimation of the cost $J$, which implicitly predict the behavior of the system. The database stores different closed-loop trajectories. These trajectories contain the state and the input trajectories of different controllers applied in the past. The information stored in the historian database is used to generate a set of candidate trajectories of appropriate length. This set of candidates is built using all the sample times in the historian database for which an $N$-step trajectory is available. Each candidate trajectory $q$ is defined by its state and input after $i$ time steps, $x_q(i), u_q(i)$ with $i = 0, \dots N$, where $x_q(0)$ is the initial state of the candidate trajectory and they satisfy

$$x_q(i+1) = f(x_q(i), u_q(i)). \tag{4}$$

We denote its corresponding input trajectory as

$$U_q = \begin{bmatrix} u_q(0) \\ u_q(1) \\ \vdots \\ u_q(N-1) \end{bmatrix} \tag{5}$$

and its corresponding cost for the objective function is considered as

$$J_q = \sum_{i=0}^{N-1} \ell(x_q(i), u_q(i)) + F(x_q(k+N)). \tag{6}$$

In this paper, we propose to use a control law derived from the control trajectories in the candidates set. Following a receding horizon approach, at each sampling time, the control signal to be applied will be computed as a weighted sum of the initial control signals of the candidates considered, ie,

$$u^*(k) = \sum_q \beta_q^* u_q(0), \tag{7}$$

where the optimal values of the weights $\beta_q$ are chosen to minimize an estimation of $J(x(k), \sum \beta_q U_q)$. The following section discuss how to define this estimation and implement the proposed controller.

## 3 | CONTROLLER FORMULATION

This section presents the main contributions of the paper, which are based on using an approach closely related to direct weight optimization to forecast the future performance cost, and the controller formulation itself, in which only a subset of the past control actions in the historian are considered in the control computations to carry out a local linearization. The proposed predictive controller computes the current control actions as a weighted sum of past control actions. In order to minimize the computational burden and to provide good cost estimations based on local data, only a subset of the past control actions in the historian are considered in current control computations. This subset is comprised of closed-loop trajectories starting from an initial state close to the current state of the process.

In particular, at time $k$, we propose to choose the $Q$ candidates with an initial state $x_q(0)$ closest to the current state $x(k)$ taking into account a given metric.† Once the $Q$ candidate trajectories are obtained, the optimal control sequence will be chosen among the convex combination of the control sequences of the candidate trajectories with an initial state equal to $x(k)$, ie, the optimization variables are the $Q$ weights $\beta_q$ with $q = 1, \dots, Q$ such that

$$\begin{aligned} x(k) &= \sum_{q=1}^{Q} \beta_q x_q(0), \\ \sum_{q=1}^{Q} \beta_q &= 1, \\ \beta_q &\geq 0, \ \forall q \in \{1, \dots, Q\}. \end{aligned} \tag{8}$$

For a given choice of candidates weights $\beta_q$, we consider the following estimation of its corresponding cost:

$$J\left(x(k), \sum \beta_q U_q\right) = J\left(\sum \beta_q x_q(0), \sum \beta_q U_q\right) \simeq \sum \beta_q J_q. \tag{9}$$

This approach is related to direct weight optimization nonlinear identification methods.[31] Direct weight optimization methods are based on postulating an estimator that is linear in the observed outputs of the estimated function (in this case the predicted cost) and then determining the weights in this estimation by direct optimization of a suitable chosen criterion. Different criteria can be defined depending on the properties of the estimated function to take into account the nonlinearities of the function by penalizing far away points and the effect of noise. In the proposed approach, we assume that, for the set of $Q$ closest candidates, the function can be approximated by a linear function and we neglect the effect of noise. In practice, using local trajectories is similar to carrying out an online linearization of the dynamics around the current state. Under these assumptions, (9) is a valid estimation of the cost for any particular choice of weights (see remark 4 in the work of Roll et al[31]). The optimization is then carried out to optimize the expected predicted performance as in standard MPC. Note that, because of the receding horizon implementation, the estimation of the future cost is recalculated at each sampling time, reducing the effect of the prediction error through feedback.

Summing up, the proposed controller is based on solving the following linear programming optimization problem based on minimizing the upper bound on the trajectory defined by the convex combination of the candidate trajectories that start from the current state:

$$\begin{aligned} \min_{\beta \in \mathbb{R}^Q} \ & \sum_{q=1}^{Q} \beta_q J_q \\ \text{s.t.} \ & x(k) = \sum_{q=1}^{Q} \beta_q x_q(0), \\ & \sum_{q=1}^{Q} \beta_q = 1, \\ & \beta_q \geq 0, \ \forall q \in \{1, \dots, Q\}. \end{aligned} \tag{10}$$

†Candidates with the same distance value will be randomly selected if necessary so that only Q candidates are included in the set.

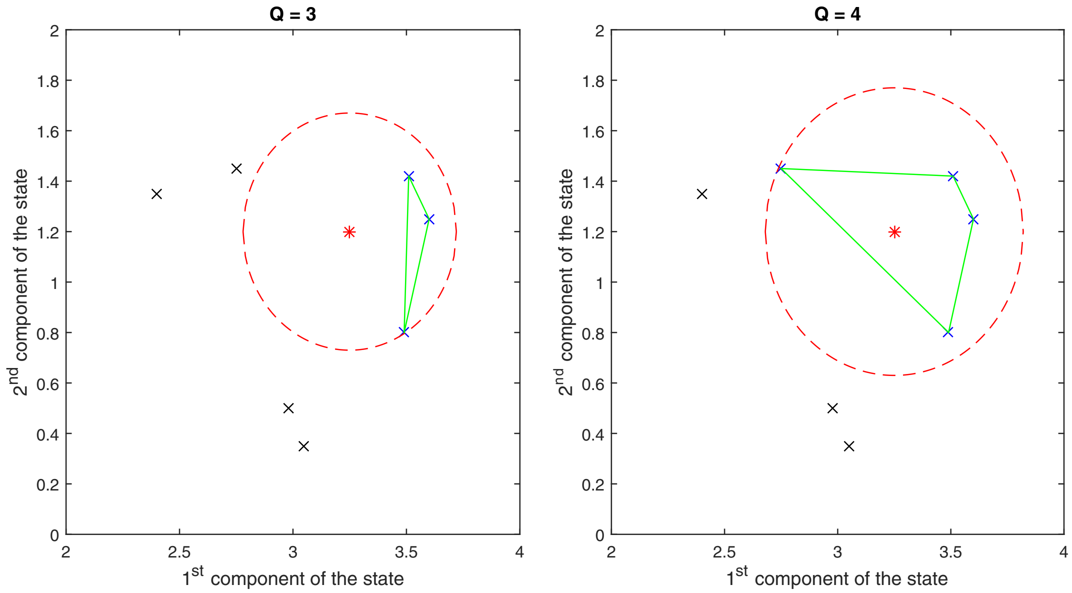


**FIGURE 1** Feasibility problem in $\mathbb{R}^2$: left infeasible, right feasible [Colour figure can be viewed at wileyonlinelibrary.com]

Problem (10) is a linear programming problem that can be solved using off-the-shelf algorithms, even for large number of optimization variables. In the next section, in order to illustrate the proposed strategy, it is applied to control the Richmond water distribution system.

*Remark* 1. We notice that, if $J : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u N} \to \mathbb{R}$ is a convex function in $\mathbb{R}^{n_x} \times \mathbb{R}^{n_u N}$, then, in view of Jensen's inequality,[35] we have

$$J\left(x(k), \sum \beta_q U_q\right) = J\left(\sum \beta_q x_q(0), \sum \beta_q U_q\right) \leq \sum \beta_q J_q. \tag{11}$$

Thus, the proposed approach, under the convexity assumption, minimizes an upper bound of the cost for the chosen future input trajectory. Note that $J$ is convex for linear systems and quadratic cost functions, which are widely used in the MPC literature.

*Remark* 2. Although we have considered a state feedback setting, the proposed approach can also be used in an output feedback setting in which the output measurements are stored in the database. In addition, constraints can be taken into account using the weighted predicted state and input trajectories. In Section 4, we present an example in which a constraint on an output of the system (the pressure at one of the demand nodes of the water drinking network) is considered.

*Remark* 3. The cardinality of the candidate trajectories $Q$ is important because larger values carry higher computational burdens. Moreover, a large number of candidates increases the distance from $x(k)$ of the last candidate selected with a local linearity loss (if the system were nonlinear). On the opposite side, smaller values of $Q$ could produce feasibility problems. Figure 1 shows the feasibility problem in $\mathbb{R}^2$. On the left side, $S_3$ is the subset contained in the convex envelope formed by states $x_q(0)$ with $Q = 3$. It is shown that $x(k) \notin S_3$, so the optimization problem is infeasible. On the right side, $S_4$ is the subset contained in the convex envelope formed by states $x_q(0)$ with $Q = 4$. In this case, $x(k) \in S_4$, so the minimization problem is feasible. The solution to feasibility problems could be to increase the value of $Q$ to find a convex envelope that contains $x(k)$. However, sometimes this may not be possible because there is not enough information in the database or because the current state is close to the system operating boundaries and the trajectories in the database operate far from these boundaries. In these cases, a different solution has to be considered. In this paper, we propose to apply the $u_q(0)$ of the closest candidate. However, there are other options like using the input corresponding to the nearest point of the convex envelope of the candidates or consider some form of extrapolation procedure.

*Remark* 4. Hyperparameters of the trajectories stored in the data base are defined as additional information of the controllers that generate these trajectories. Examples of hyperparameters are the tuning values of a PID controller, reference and hysteresis of a relay controller or the prediction horizon, and the matrices that define the cost of a MPC controller. Hyperparameters can be taken into account in the distance function, the cost function, or in the constraints of the optimization problem to improve the performance of the proposed approach. In Section 4, we present an example in which the information of the reference of the controller of each trajectory stored in the database is used as a hyperparameter in the proposed controller.

## 4 | APPLICATION TO A WATER DISTRIBUTION SYSTEM

The Richmond water distribution system is a well-known case study[32,36] that can be simulated using the EPANET hydraulic simulation software.[37] This case study describes a system that is a good candidate for the historian data based predictive control strategy presented in this paper, which has also been used in a standard MPC framework.[38] Although the system is nonlinear and complex, it can be approximated by a linear water balance based model with sufficiently large sampling times, in this case, 1 hour. Figure 2 shows the Richmond water distribution system diagram, which is composed by six tanks, seven pumps, 41 nodes of which 11 are demand nodes, and 44 pipes of which eight are unidirectional pipes and one source.

Note that, for a given tank, there are several demand nodes that withdraw water from that tank. Water is introduced by the pumps from a single water source and demand nodes consume this water, lowering the levels in the tanks. The control objective in this paper is to keep the water levels in each tank around a specified set-point, while satisfying the demands. The state vector $x$ is composed by the levels of the six tanks, ie, $x \in \mathbb{R}^6$. Demands are considered disturbances, modeled by the disturbance signal vector $d \in \mathbb{R}^{11}$.

In order to attain the control objective, water flows are used as manipulated variables; thus, the input signal vector $u \in \mathbb{R}^7$ contains the water flows that have to be attained using each pump in the system. Note that, in the Richmond case study (as in many water distribution systems), pumps are operated with an ON-OFF mode; thus, the necessity of a low level switching logic that transform each real component of $u$ into an equivalent logic sequence for each particular pump. In this paper, discrete time intervals of 1 hour ($k \in \mathbb{N}$) and low-level switching logic intervals of $\frac{1}{24}$ hours = 2.5 minutes are considered. To minimize the number of pump switches, a duty cycle policy consisting on applying the control effort in a single pulse is used.

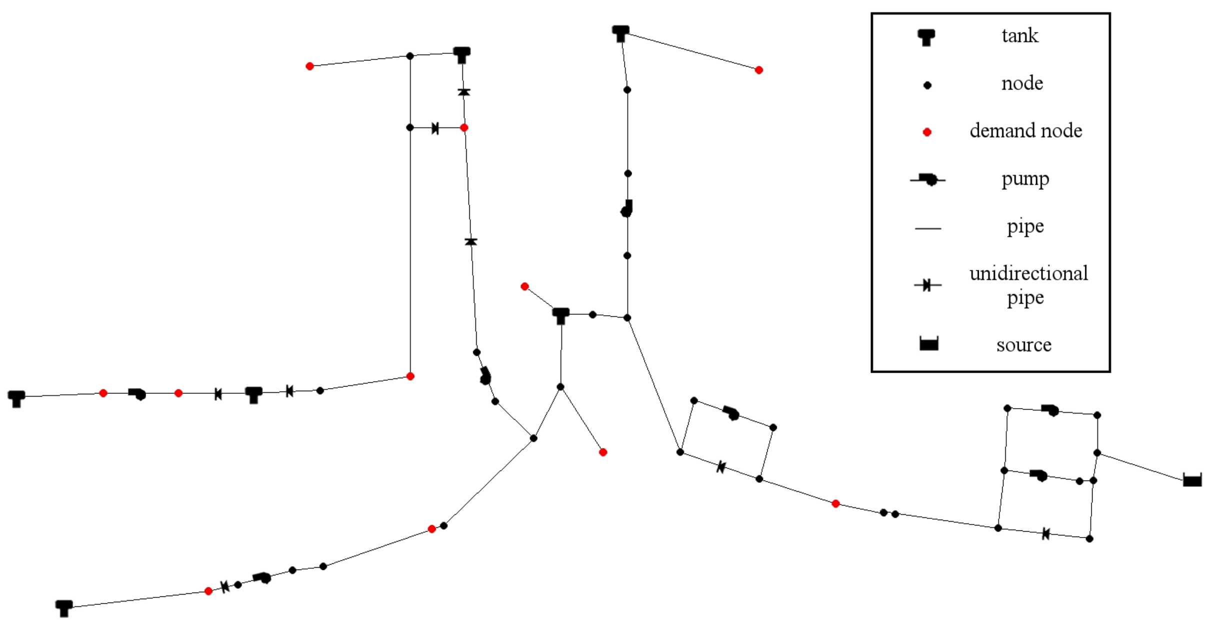


**FIGURE 2** Richmond water distribution network diagram [Colour figure can be viewed at wileyonlinelibrary.com]

**TABLE 1** Switching levels (in meters) of the relays of each of the controllers used to generate the data base

| | | RbC 1 | | RbC 2 | | RbC 3 | | RbC 4 | |
|---|---|---|---|---|---|---|---|---|---|
| Pump | Tank | Level ON | Level OFF | Level ON | Level OFF | Level ON | Level OFF | Level ON | Level OFF |
| $u1$ | $x1$ | 2.3685 | 2.9799 | 2.4785 | 3.0899 | 1.5018 | 2.1132 | 1.9352 | 2.5466 |
| $u2$ | $x1$ | 3.0405 | 3.2513 | 3.1505 | 3.3613 | 2.1738 | 2.3846 | 2.6072 | 2.8180 |
| $u3$ | $x1$ | 2.8888 | 3.1126 | 2.9988 | 3.2226 | 2.0221 | 2.2459 | 2.4555 | 2.6793 |
| $u4$ | $x2$ | 3.2623 | 3.5789 | 3.3323 | 3.6489 | 2.5956 | 2.9122 | 2.9290 | 3.2456 |
| $u5$ | $x3$ | 0.7185 | 1.8850 | 0.8285 | 1.9950 | 0.5852 | 1.7517 | 0.6518 | 1.8183 |
| $u6$ | $x4$ | 1.5907 | 1.9708 | 1.7207 | 2.1008 | 1.2774 | 1.6575 | 1.4340 | 1.8141 |
| $u7$ | $x6$ | 1.7037 | 2.1095 | 1.7837 | 2.1895 | 0.7037 | 1.1095 | 1.2037 | 1.6095 |

**TABLE 2** Set-points (in meters) of the controllers used to generate the data base

| | $x1_r$ | $x2_r$ | $x3_r$ | $x4_r$ | $x6_r$ |
|---|---|---|---|---|---|
| RbC1 | 2.9403 | 3.4206 | 1.3018 | 1.7808 | 1.9066 |
| RbC2 | 3.0503 | 3.4906 | 1.4118 | 1.9108 | 1.9866 |
| RbC3 | 1.6403 | 2.4206 | 1.1018 | 1.3108 | 0.4066 |
| RbC4 | 2.5069 | 3.0873 | 1.2351 | 1.6241 | 1.4066 |

**TABLE 3** Maximum and minimum safety tank water levels in Richmond system

| | Tank 1 | Tank 2 | Tank 3 | Tank 4 | Tank 5 | Tank 6 |
|---|---|---|---|---|---|---|
| Min. level | 1.02 | 2.03 | 0.5 | 1.1 | 0.2 | 0.19 |
| Max. level | 3.37 | 3.65 | 2 | 2.11 | 2.69 | 2.19 |

It is assumed that the water demand is composed by a periodic signal with a random component, ie,

$$d(k) = d_p(k) + d_r(k), \tag{12}$$

where $d_p \in \mathbb{R}^{11}$ is a set of periodic signals that satisfy $d_p(k) = d_p(k + T)$ with $T = 24$ obtained from the demand profiles used by Van Zyl et al[32,36] and $d_r(k)$ is a set of random zero mean signals, which added to $d_p(k)$ result in a demand signal with 5% of variation around $d_p(k)$.

The Richmond water distribution system has six tanks; however, because of the network structure, in order to maintain the desired levels, Tank 5 is always full in normal operating conditions. For this reason, only the rest of the tanks are considered for the purposes of computing the control signals. Note, however, that the EPANET simulation uses the whole state information.

The cost function in these examples is the following tracking error penalty stage cost that only depend on $x$ (ie, in these examples $l(x, u) = l(x)$):

$$\ell(x) = |x_c - x_r|_2^2, \tag{13}$$

where $x_c \in \mathbb{R}^5$ are the levels of the five tank levels considered of $x$ and $x_r \in \mathbb{R}^5$ are their corresponding reference values. Furthermore, the terminal cost is equal to the stage cost, ie, $F(\cdot) = \ell(\cdot)$.

The database stores the closed-loop trajectories of four different controllers, each one based on a different set of relays, denoted RbC1, RbC2, RbC3, and RbC4, respectively. Every pump is switched on and off depending on the level of the tank that is directly downstream. Table 1 shows this relation and the switching on and off levels for every pump of each controller.

Table 2 shows set-points values for each controller. The set-points are obtained as the middle point of the corresponding pump switching on and off levels. In the case of tank 1, denoted as $x1$, which is the first component of $x$, the set-point is the average of the three middle switching levels of pumps 1, 2, and 3, denoted as $u1$, $u2$, and $u3$ respectively.

For each controller, there are 100 trajectories stored in the data base, each one with 96 hours of closed-loop simulated operation of the system. Each of the trajectories start with different random initial values of the tank levels that satisfy the minimum and maximum safety constraints of Table 3. In addition, the head at the demand node four, denoted $z(k)$, is also included in the database. Head information will be used to include a soft constraint to limit the maximum pressure in that particular demand node to demonstrate that historic data can be used to model complex, possibly nonlinear, outputs and take them into account in the control decision. The amount of information stored in the database is equivalent to 4.38 years of historian information. Although a realistic size for a historian, the database is very small in relation to the dimensionality of the problem, which is $\mathbb{R}^6$, which could lead to identification problems.

Using the information of these simulations, over 38 000 24-hour candidates trajectories are defined, for which at each sampling time; only $Q$ will be considered to formulate the controller. The choice of the number of candidates considered depends mainly on the density and quality of the historian trajectories stored in the database. There is a trade-off between complexity, feasibility, and relevance. Higher number of candidates in general avoid feasibility issues, but may take into account candidates whose information is not relevant because its initial state is too far away from the current state leading to lower quality cost estimations. In this example, a value of $Q = 500$ has been found to be appropriate in regard to the aforementioned trade-off.

## 4.1 | Example 1

The example is a closed-loop simulation of 120 hours with initial state

$$x(0) = [3, 2.44, 1.58, 1.5, 0.99, 1.51], \tag{14}$$

and the reference used is

$$x_r = [3.0503, 3.4906, 1.4118, 1.9108, 1.9866]. \tag{15}$$

This reference is equal to the reference of one of the controllers used to create the database, in particular, the relay-based controller 2 (RbC 2). In order to take into account the periodic nature of the demand, the distance function takes the following form:

$$dist_q = \|x(k) - x_q(0)\|_\lambda^2 + \lambda_p \Delta_q(k)^2, \tag{16}$$

where $\lambda$ is a diagonal matrix given by

$$\lambda = \text{diag}([\ 1.0831\ \ 1\ \ 1.825\ \ 1.7299\ \ 1.6667\ ]), \tag{17}$$

and $\lambda_p = 0.2$. The term $\Delta_q(k)$ is the time difference, in hours, between the candidate initial time and the current time. This distance function (16) takes into account the periodic nature of the demands of the system, penalizing candidates trajectories that start at a different hour because the demand differs.

The optimization problem solved at each time instant is defined in (10) and the control input applied is calculated as in (7). A standard linear programming solver can be used to obtain the solution as, in our case, *linprog* in Matlab. Figure 3 shows the water levels in the six tanks of the Richmond system during the closed-loop simulation using EPANET. Minimum, maximum, and reference levels are represented for each tank (in red-dashed lines and black-dashed lines, respectively). Tank 5 is not controllable and there is not any reference signal in its level graphic.

The pump flows, in liters per second, can be observed in Figure 4. Notice that the solution obtained with the proposed controller tends to a quasi-periodic behavior. Since demand signals periodicity produces periodic trajectories and control action stored in the database when system is controlled with relay controllers, the convex combination of the control actions in the database is almost periodic as well.

To evaluate the performance of the controller, it is necessary to take into account the periodic nature of the system. The performance metric will be the summation of the closed-loop performance cost over a period of $d(k)$ computed at each hour of the simulation as

$$PC(x(k)) = \sum_{i=0}^{N} \ell(x(k+i)), \tag{18}$$

where $x(k+i)$ are the values of the tank levels of the closed-loop simulation; $\ell(\cdot)$ is defined as in (13) and $N = 24$. Note that, in this system, the instantaneous performance cost has no meaning, as it will go up and down as the periodic disturbance changes. The summation of the closed-loop performance cost over a period is a sensible choice as it should converge to a constant value when the closed-loop system reaches a quasi-steady state periodic trajectory, provided that the controller is working fine. Note that the random part of $d(k)$ has an impact on the behavior of $PC(x(k))$.

Figure 5 shows the evolution of $PC(x(k))$ for two of the relay-based controllers and the proposed strategy. The relay-based controllers 3 and 4 are not represented because their performance costs are much higher. In particular, their mean costs are 76.99 and 27.7, respectively, while the mean cost of the proposed historian data–based predictive control is 3.36. The historian data–based predictive controller improves the controllers included in the database, although the performance and behavior is similar to a relay controller.

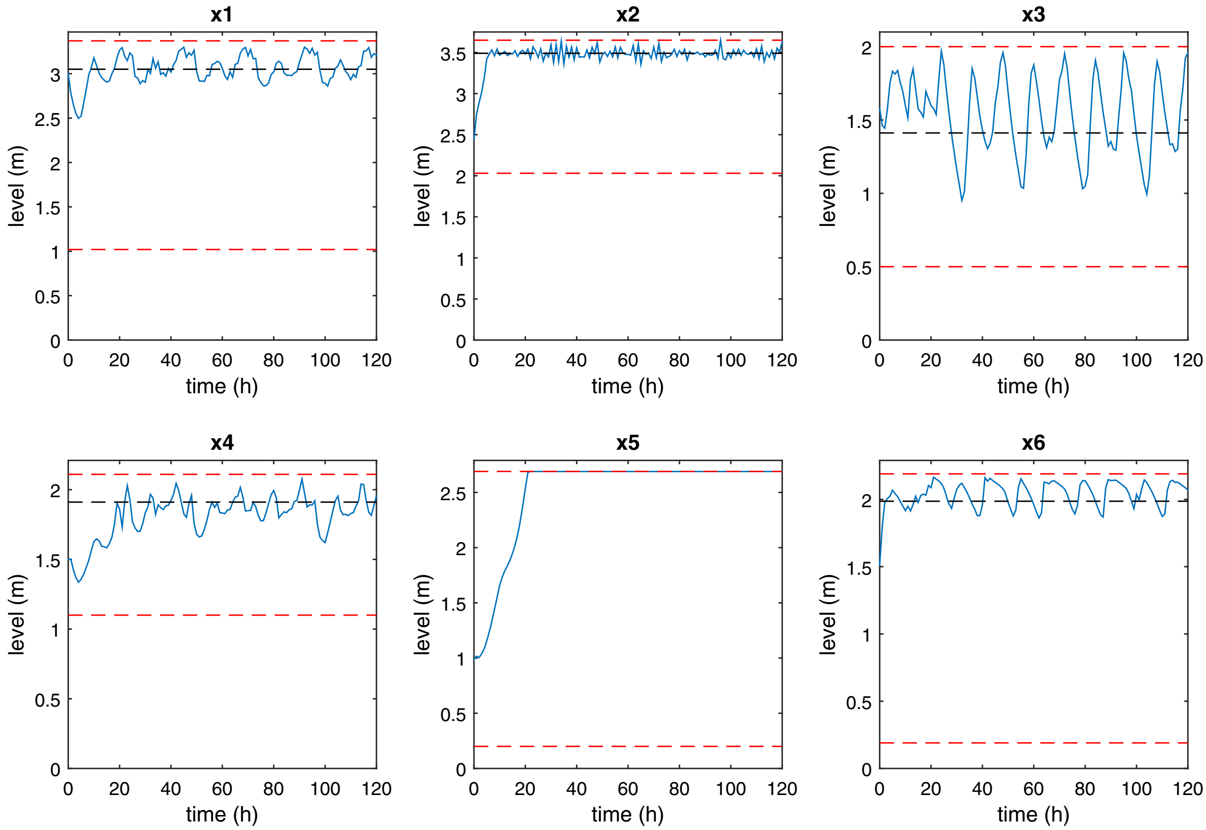


**FIGURE 3** References (black dash) and closed-loop trajectories (blue solid) of the tank levels with the proposed controller. Maximum and minimum physical-level constraints (red dash) are represented for each tank [Colour figure can be viewed at wileyonlinelibrary.com]

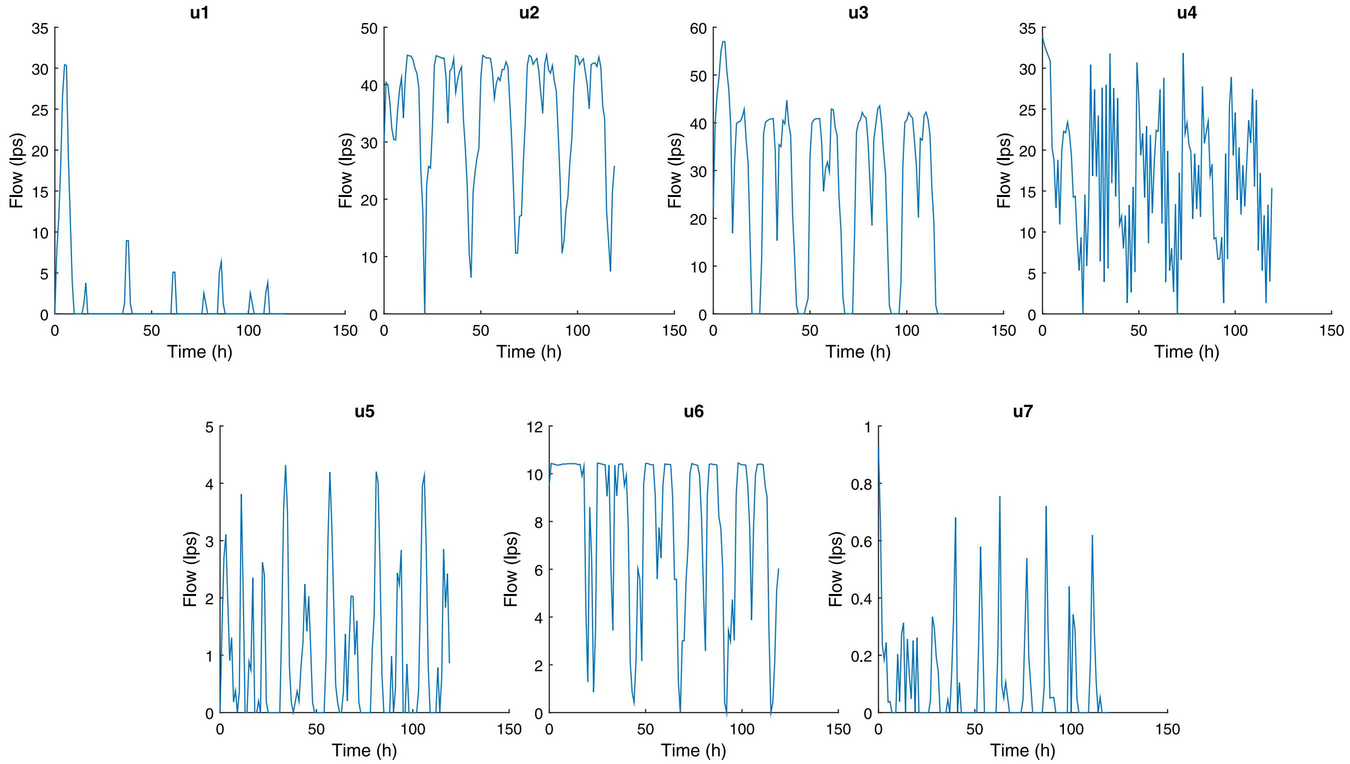


**FIGURE 4** Closed-loop control actions for each pump with the proposed controller [Colour figure can be viewed at wileyonlinelibrary.com]

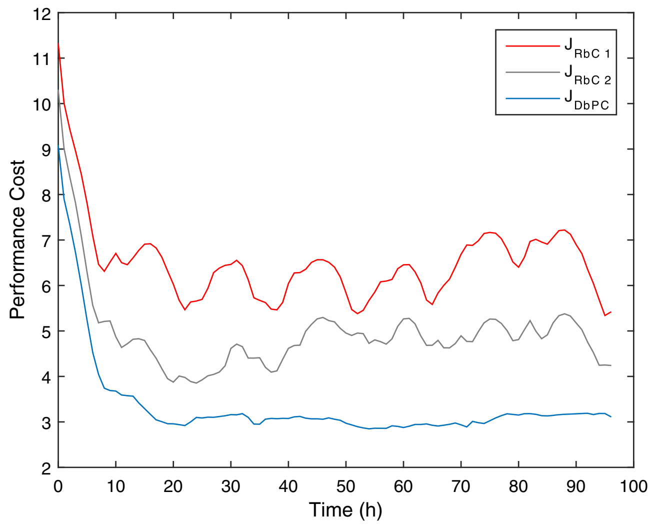


**FIGURE 5** Performance cost comparison of the proposed controller ($J_{DbPC}$) and the relay-based controllers 1 and 2 ($J_{RbC1}$ and $J_{RbC2}$) [Colour figure can be viewed at wileyonlinelibrary.com]

## 4.2 Example 2

In this example, the proposed strategy is applied taking into account the set-point value of each closed-loop trajectory stored in the database. The reference of each relay-based controller is shown in Table 2. According to Remark 4, which presents hyperparameters and the way to use them in the proposed strategy, this example focuses on applying the hyperparameter information in both the distance and the cost function.

First, hyperparameters are used when building the subset of $Q$ candidate trajectories. Defining $x_q^r$ as the set-point value of candidate $q$, the distance function in (16) is modified, adding a term, which penalizes candidate trajectories whose references are far from the reference of the problem $x_r$, ie,

$$dist_q = \|x(k) - x_q(0)\|_\lambda^2 + \lambda_p \Delta_q(k)^2 + \lambda_r \left\|x_r - x_q^r\right\|_2^2, \tag{19}$$

where $\lambda_r = 10$ weighs the deviation between the current reference and the reference of the candidate.

Second, the cost function is also modified to take into account the hyperparameters. In the previous example, the stage cost does not consider the particular target of the candidates. This can affect the set-point tracking capabilities of the controller. One possibility is to add a constraint to optimization problem (10) to guarantee that the convex combination of the candidates corresponding references $x_q^r$ is equal to the target reference of the predictive controller $x_r$, ie,

$$\sum_{q=1}^{Q} \beta_q x_q^r = x_r.$$

This constraint aims at taking into account, not only where a given trajectory is, but also where it is headed. However, adding this constraint may compromise the feasibility of the optimization problem. For this reason, in this example, we propose to add it as a soft constraint modifying the objective function. To this end, at each sampling time, the historian-based controller solves the following optimization problem:

$$\begin{aligned} \min_{\beta} \ & \sum_{q=1}^{Q} \beta_q J_q + \rho_r \left\| x_r - \sum_{q=1}^{Q} \beta_q x_q^r \right\|_2^2 \\ \text{s.t.} \ & x(k) = \sum_{q=1}^{Q} \beta_q x_q(0), \\ & \sum_{q=1}^{Q} \beta_q = 1, \\ & \beta_q \geq 0, \quad \forall q \in \{1, \ldots, Q\}. \end{aligned} \tag{20}$$

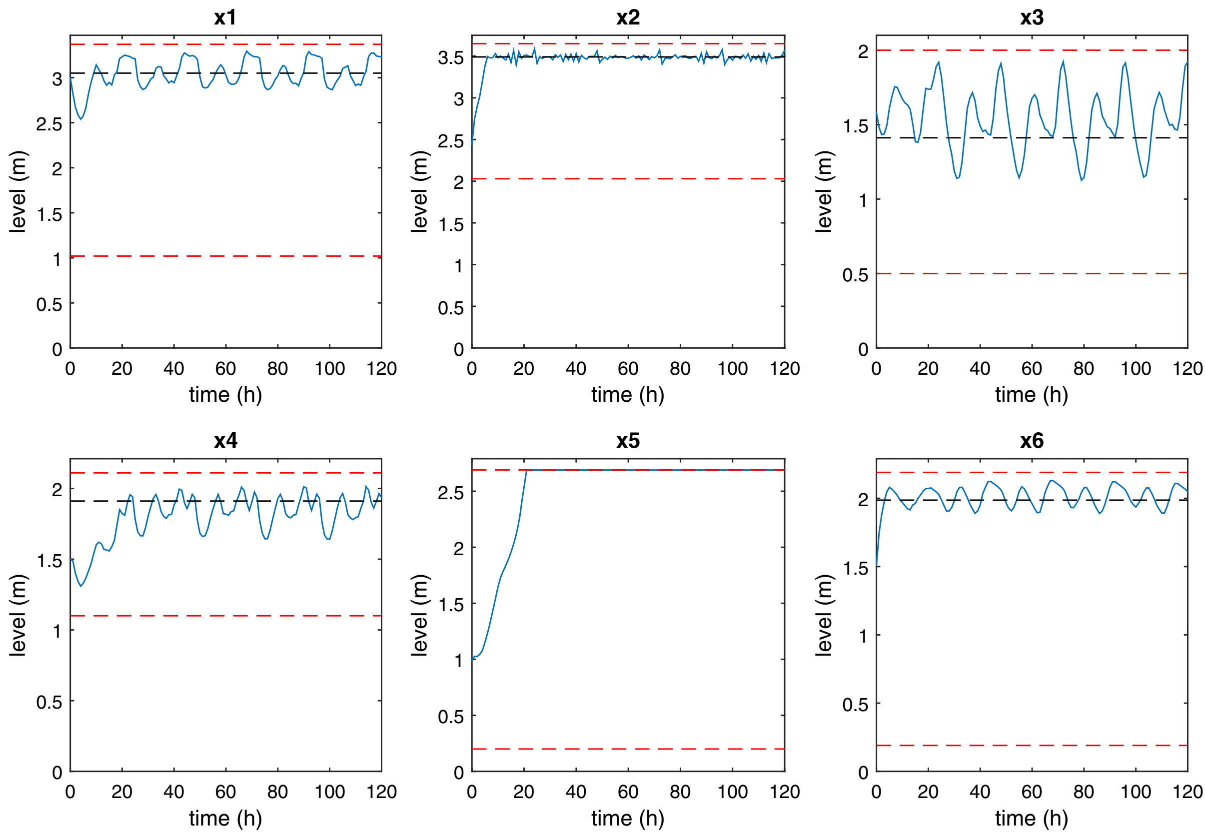


**FIGURE 6** References (black dash) and closed-loop trajectories (blue solid) of the tank levels with the proposed controller using hyperparameters [Colour figure can be viewed at wileyonlinelibrary.com]

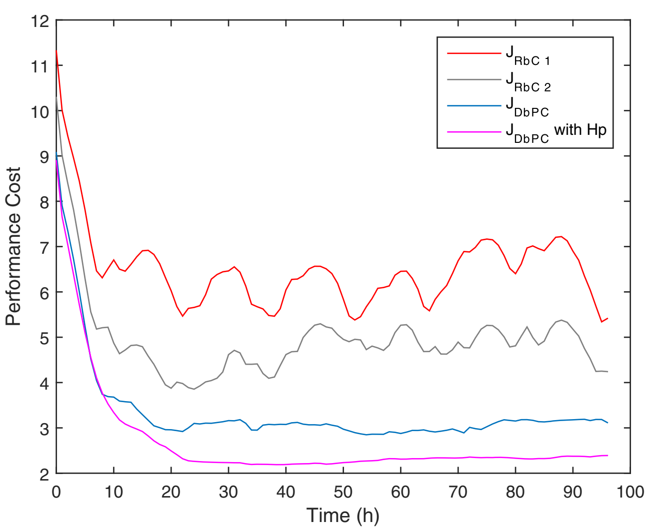


**FIGURE 7** Performance cost comparison with relay-based controllers and historian predictive control with and without hyperparameters [Colour figure can be viewed at wileyonlinelibrary.com]

Any standard quadratic programming solver can be used as, in our case, *quadprog* in Matlab. Figure 6 shows the level trajectories of the closed-loop simulation of the proposed control strategy using hyperparameters with weight $\rho_r = 1$. Figure 7 shows the closed-loop performance of the proposed controller with and without hyperparameters and the cost of the two best relay-based controllers. It can be seen that the use of hyperparameters leads to better set-point regulation and to a lower performance cost.

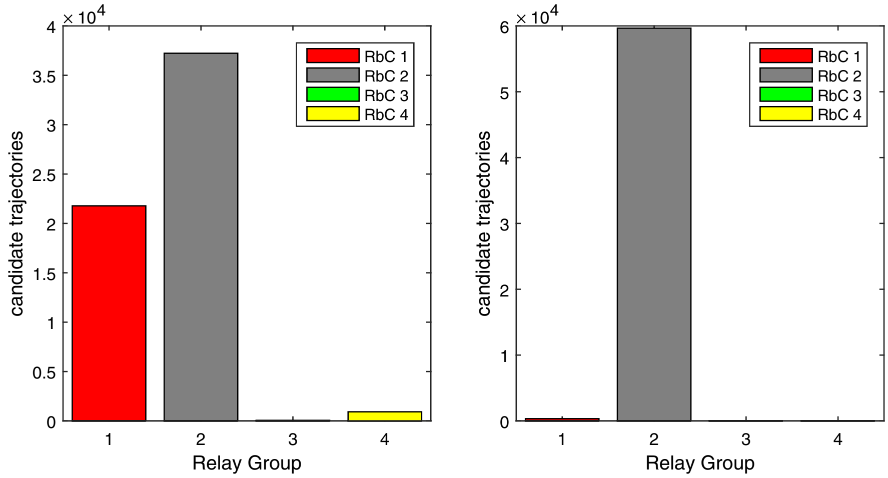


**FIGURE 8** Comparison of number of candidates provided by each relay-based (RBC) controller along the whole simulation between proposed strategy without hyperparameters (left side) and with hyperparameters (right side) [Colour figure can be viewed at wileyonlinelibrary.com]

Figure 8 shows the number of candidates provided by each relay controller along the whole simulation for the proposed controller with and without the use of the information provided by the hyperparameters.

Note that, in both cases, most candidates trajectories belong to relay-based controller 1 and 2 because their set-points are close to the reference; in fact, relay-based controller 2 has the same reference and it can be seen that, when hyperparameters are considered, almost all candidates belong to this controller. Note also that, even if most of the trajectories are of a single relay controller, the proposed strategy achieves a better performance cost.

### 4.3 | Example 3

In this section, constraints are taken into account, in particular, a maximum average head constraint in demand node four. Hydraulic head is a specific measurement of liquid pressure above a geodetic datum and it relates the energy in an incompressible fluid to the height of an equivalent static column of the fluid. As mentioned before, the historian data includes the average head in this node for all the trajectories, which we denote as $z$.

There exists a nonlinear relation between head and flow, which we can observe if we consider the head loss Hazen-William formula[39]

$$z = \gamma u^{1.852}, \tag{21}$$

where $\gamma$ is a parameter calculated with the length of the pipe, the pipe roughness coefficient, and the inside pipe diameter. Despite this nonlinear relation, the estimated average pressure sequence,

$$\hat{z}_{k+i} = \sum_{q=1}^{Q} \beta_q z_q(i), \tag{22}$$

provides a good approximation based on local data.

Head constraints are included in the optimization problem as soft constraints (slack variables) instead of hard constraints to eliminate feasibility issues caused by this sort of constraints. Note that these feasibility issues are related to the high dimension of the state in relation to the database size in this example. Thus, for lower dimensional systems, it could be possible to use hard constraints. The number of slack variables added to the optimization problem is equal to the prediction horizon $N$ to ensure that all the average heads in the estimated average head sequence, denoted as $\hat{z}$ and obtained as the convex combination of the candidate head sequences, satisfy this soft constraints.

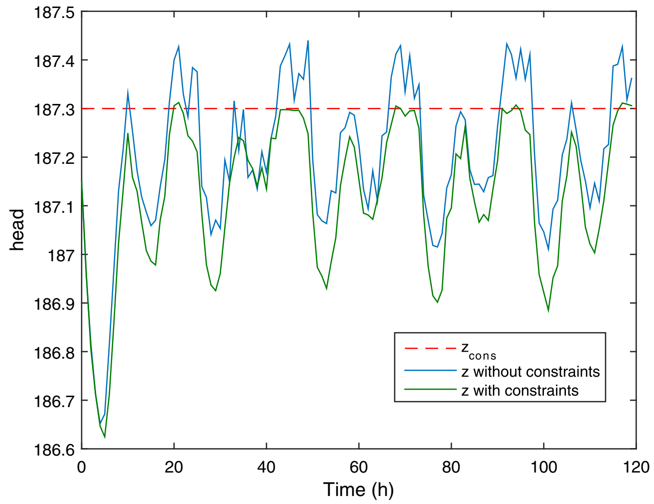


**FIGURE 9** Comparison between the average head in demand node 4 with and without constraints [Colour figure can be viewed at wileyonlinelibrary.com]

Including constraints in $z$ modifies the optimization problem (20) in the following form:

$$\begin{aligned}
\min_{\beta,\tau} \; & \sum_{q=1}^{Q} \beta_q J_q + \rho_r \left\| x_r - \sum_{q=1}^{Q} \beta_q x_q^r \right\|_2^2 + \rho_z \sum_{i=0}^{N-1} \left(\tau_i^2 + \tau_i\right) \\
\text{s.t. } & x(k) = \sum_{q=1}^{Q} \beta_q x_q(0), \\
& \sum_{q=1}^{Q} \beta_q = 1, \\
& \beta_q \geq 0, \quad \forall q \in \{1, \ldots, Q\}, \\
& z_{cons} \geq \left( \sum_{q=1}^{Q} \beta_q z_q(i) \right) - \tau_i, \quad \forall i \in \{0, \ldots, N-1\}, \\
& \tau_i \geq 0, \quad \forall i \in \{0, \ldots, N-1\},
\end{aligned} \tag{23}$$

where $\tau$ is a set of slack variables, which lets a small violation of the average head constraints; $\rho_z$ is the weight of the slack variables with respect to the other terms; $z_{cons}$ are the average head constraint values; and $z_q(i)$ is the average head of the $q$th candidate sequence at instant $i$.[‡] Similar to problem (20), a standard quadratic programming solver can be used to obtain the solution of (23).

The optimization problem solved tracks the same reference in level as in (15) and implement the proposed controller with hyperparameters using a number of candidates $Q = 500$ and without hyperparameters. The value of the head constraints weight is $\rho_z = 10^5$. For simplicity, head constraint is considered only in one demand node, ie, demand node 4, and the average head constraint values in meters is

$$z_{cons} = 187.3. \tag{24}$$

Figure 9 shows a comparison between the average head in demand node 4 obtained with and without head average constraints. In case of the proposed controller without constraints, the constraint in average head is clearly violated by the optimal solution. When the controller takes into account this constraint, the optimal solution mostly satisfies it. Figure 10 shows the average head estimation signal error for the demand node 4, which is calculated as the difference between the real average head, obtained by simulation, and the estimation of the average head, obtained as in (22). We can observe that the error signal has approximately zero mean (0.21%).

[‡]Notice that, with the same slight abuse of the notation used before, we define the head of the $q$th candidate trajectory $z_q(i)$ as the $i$th row ahead average head after the initial row $n_r$ of the candidate trajectory.

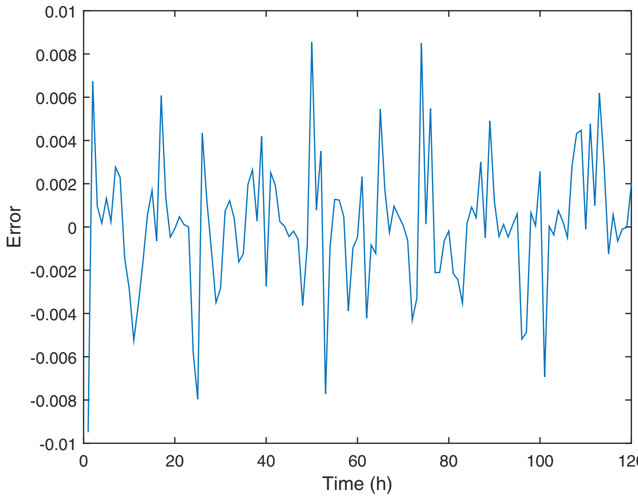


**FIGURE 10** Difference between the estimated average head and the real average head of demand node 4 [Colour figure can be viewed at wileyonlinelibrary.com]

**TABLE 4** Amplitude and offset of the sinusoidal reference signal of each controllable tank

| | Tank 1 | Tank 2 | Tank 3 | Tank 4 | Tank 6 |
|---|---|---|---|---|---|
| Amplitude | 1.41 | 1.07 | 0.31 | 0.6 | 1.58 |
| Offset | 2.3453 | 2.9556 | 1.2568 | 1.6108 | 1.1966 |

## 4.4 | Example 4

In this section, we consider a set of periodic level reference signals instead of the constant signals used in the previous examples. Typically, water distribution networks are controlled taking into account economic issues such as the tariff pattern related to the electricity price. In the Richmond benchmark, the electricity costs considered had a different tariff

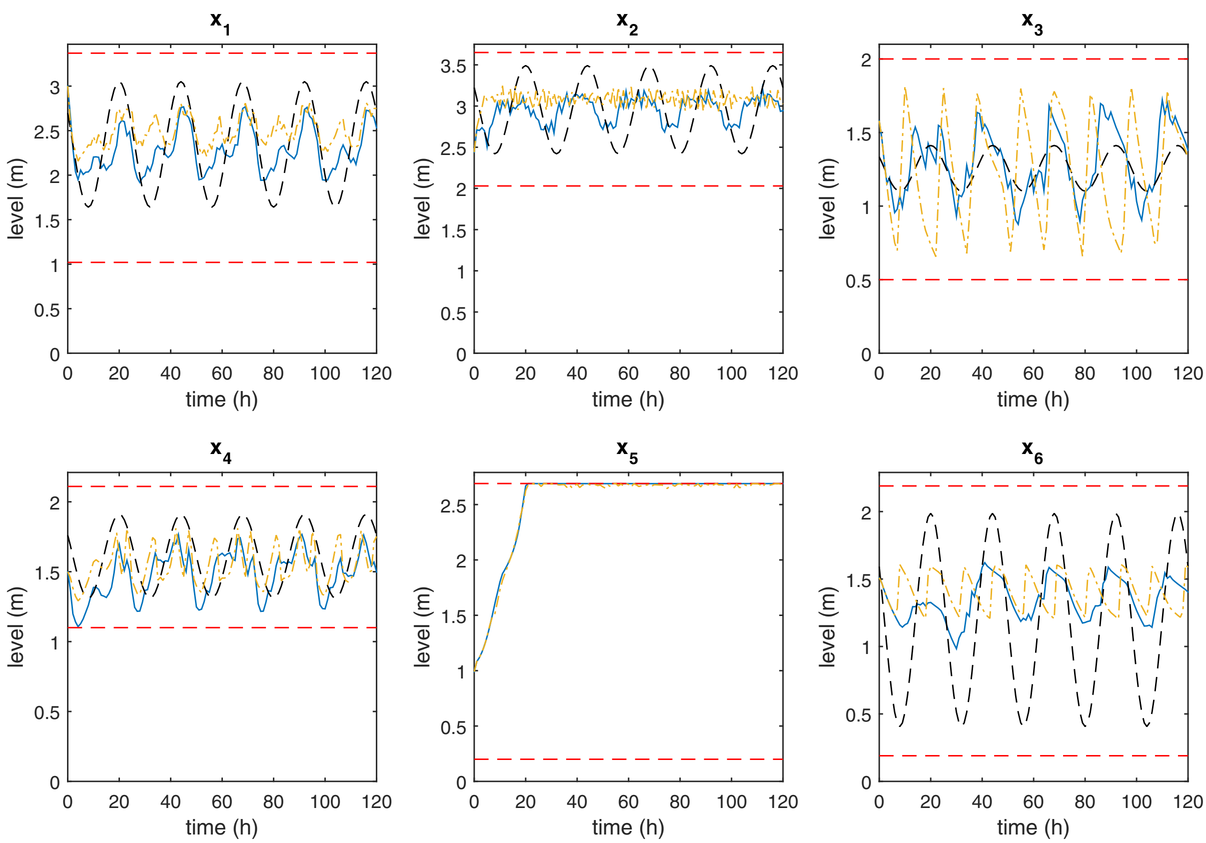


**FIGURE 11** Periodic reference for every tank (dashed black), closed-loop simulation with the proposed controller (blue), and with the relay controller 4 (dashed-dotted yellow) [Colour figure can be viewed at wileyonlinelibrary.com]

during day and night hours. In general, this leads to nonsteady level in the tanks, which fill during the night in which pumping water is more economic, and discharge to satisfy the demands during the day. According to this idea and considering the period of the tariff pattern and demand signals, a sinusoidal signal of 24-hour period is taken into account with its maximal at 3 a.m. and its minimal at 15 p.m. to build the reference signals for the tank levels. Table 4 shows the amplitude and offset for every reference signal considered.

Figure 11 shows the level references considered and the closed-loop trajectories of the proposed controller, together with that of the relay-based controller 4 (which provided the best performance of all the relay based-controllers). This figure shows that the proposed controller provides a quasi-periodic closed-loop trajectory, which is almost in phase with the reference, while, clearly, the relay-based trajectory (which is akin to the trajectories of the database) is not correlated. This implies that the proposed strategy does not simply learn (or identify) the control law in the database, but, rather than that, it uses the stored trajectories to fulfill as best as possible the current control objective, which can be different from that used to build the database.

## 5 | CONCLUSIONS

In this paper, we have proposed a heuristic data–based predictive controller based, optimizing an estimation of the performance over a convex combination of past trajectories. This approach can be used in complex systems in which models or enough data for identification are not available. The proposed approach is applied to a water distribution network, demonstrating that it is able to consider different issues such as periodic level references and pressure constraints explicitly in its formulation. Future works include the development of stabilizing designs and estimation error bounds.

### ACKNOWLEDGEMENTS

This work was supported by the MINECO-Spain and FEDER Funds under projects DPI2013-48243-C2-2-R and DPI2016-76493-C3-1-R and by the University of Seville under grant 2014/425.

### ORCID

*Jose R. Salvador* https://orcid.org/0000-0002-1568-1318
*D. Muñoz de la Peña* https://orcid.org/0000-0002-6696-8490
*D.R. Ramirez* https://orcid.org/0000-0002-8962-5984
*T. Alamo* https://orcid.org/0000-0002-0623-8146

### REFERENCES

1. Ljung L. *System Identification: Theory for the User*. Upper Saddle River, NJ: Prentice-hall, Inc.; 1987.
2. Liberzon D. *Calculus of Variations and Optimal Control Theory: A Concise Introduction*. Princeton, NJ: Princeton University Press; 2011.
3. Kirk DE. *Optimal Control Theory: An Introduction*. North Chelmsford, MA: Courier Corporation; 2012.
4. Karg B, Lucia S. Learning-based approximation of robust nonlinear predictive control with state estimation applied to a towing kite. In: Proceedings of the 18th European Control Conference (ECC); 2019; Naples, Italy.
5. Brdys M, Ulanicki B. *Operational Control of Water Systems: Structures, Algorithms and Applications*. Upper Saddle River, NJ: Prentice Hall; 1994.
6. Puig V, Ocampo-Martinez C, Pérez R, Cembrano G, Quevedo J, Escobet T. *Real-Time Monitoring and Operational Control of Drinking-Water Systems*. Cham, Switzerland: Springer; 2017.
7. Ocampo-Martinez C, Puig V, Cembrano G, Quevedo J. Application of predictive control strategies to the management of complex networks in the urban water cycle. *IEEE Control Syst*. 2013;33(1):15-41.
8. Zhang H, Cui L, Zhang X, Luo Y. Data-driven robust approximate optimal tracking control for unknown general nonlinear systems using adaptive dynamic programming method. *IEEE Trans Neural Netw*. 2011;22(12, 2):2226-2236.
9. Wei Q, Liu D. Data-driven neuro-optimal temperature control of water-gas shift reaction using stable iterative adaptive dynamic programming. *IEEE Trans Ind Electron*. 2014;61(11):6399-6408.
10. Lee J, Lee J. Approximate dynamic programming-based approaches for input-output data-driven control of nonlinear processes. *Automatica*. 2005;41(7):1281-1288.
11. Qiu J, Wang T, Yin S, Gao H. Data-based optimal control for networked double-layer industrial processes. *IEEE Trans Ind Electron*. 2017;64(5):4179-4186.
12. Chi R, Wang D, Hou Z, Jin S. Data-driven optimal terminal iterative learning control. *J Process Control*. 2012;22(10):2026-2037.

13. Janssens P, Pipeleers G, Swevers J. A data-driven constrained norm-optimal iterative learning control framework for LTI systems. *IEEE Trans Control Syst Technol*. 2013;21(2):546-551.
14. Jain A, Smarra F, Mangharam R. Data predictive control using regression trees and ensemble learning. In: Proceedings of the 2017 Conference on Decision and Control. 2017; Melbourne, Australia.
15. Van Overschee P, De Moor B. *Subspace Identification for Linear Systems: Theory–Implementation–Applications*. New York, NY: Springer Science & Business Media; 2012.
16. Van Overschee P, De Moor B. Subspace algorithms for the stochastic identification problem. *Automatica*. 1993;29(3):649-660.
17. Van Overschee P, De Moor B. A unifying theorem for three subspace system identification algorithms. *Automatica*. 1995;31(12):1853-1864.
18. Kadali R, Huang B, Rossiter A. A data driven subspace approach to predictive controller design. *Control Eng Pract*. 2003;11(3):261-278.
19. 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.
20. Milanese M, Novara C. Unified set membership theory for identification, prediction and filtering of nonlinear systems. *Automatica*. 2011;47(10):2141-2151.
21. Limon D, Calliess J, Maciejowski JM. Learning-based nonlinear model predictive control. *IFAC-PapersOnLine*. 2017;50(1):7769-7776.
22. Bravo JM, Alamo T, Vasallo M, Gegúndez ME. A general framework for predictors based on bounding techniques and local approximation. *IEEE Trans Autom Control*. 2017;62(7):3430-3435.
23. Favoreel W, De Moor B, Van Overschee P, Gevers M. Model-free subspace-based LQG-design. In: Proceedings of the 1999 American Control Conference; 1999; San Diego, CA.
24. Piga D, Formentin S, Bemporad A. Direct data-driven control of constrained systems. *IEEE Trans Control Syst Technol*. 2017;26(4):1422-1429.
25. Tanaskovic M, Fagiano L, Novara C, Morari M. Data-driven control of nonlinear systems: an on-line direct approach. *Automatica*. 2017;75:1-10.
26. Fagiano L, Novara C. Learning a nonlinear controller from data: theory, computation, and experimental results. *IEEE Trans Autom Control*. 2016;61(7):1854-1868.
27. Bertsekas DP. *Dynamic Programming and Optimal Control*. Belmont, MA: Athena scientific; 1995.
28. Powell WB. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Hoboken, NJ: John Wiley & Sons; 2007.
29. Li Y, Yang C, Hou Z, Feng Y, Yin C. Data-driven approximate q-learning stabilization with optimality error bound analysis. *Automatica*. 2019;103:435-442.
30. Van Hasselt H. Reinforcement learning in continuous state and action spaces. In: *Reinforcement Learning*. Berlin, Germany: Springer; 2012:207-251.
31. Roll J, Nazin A, Ljung L. Nonlinear system identification via direct weight optimization. *Automatica*. 2005;41(3):475-490.
32. Van Zyl JE. Richmond water distribution network EPANET model. 2001. http://emps.exeter.ac.uk/engineering/research/cws/resources/benchmarks/operation/richmond.php
33. Rawlings JB, Mayne DQ. *Model Predictive Control: Theory and Design*. Madison, WI: Nob Hill Publishing; 2009.
34. Fernandez-Camacho E, Bordons-Alba C. *Model Predictive Control*. 2nd ed. London, UK: Springer; 2004.
35. Boyd S, Vandenberghe L. *Convex Optimization*. Cambridge, UK: Cambridge University Press; 2004.
36. Van Zyl JE, Savic DA, Walters GA. Operational optimization of water distribution systems using a hybrid genetic algorithm. *J Water Resour Plan Manag*. 2004;130(2):160-170.
37. Enviromental Protection Agency. EPANET: software that models the hydraulic and water quality behavior of water distribution piping systems. 2018. https://www.epa.gov/water-research/epanet
38. Wang Y, Salvador JR, de la Peña DM, Puig V, Cembrano G. Economic model predictive control based on a periodicity constraint. *J Process Control*. 2018;68:226-239.
39. Williams GS, Hazen A. *Hydraulic Tables: Showing the Loss of Head due to the Friction of Water Flowing in Pipes, Aqueducts, Sewers, Etc. and the Discharge Over Weirs*. New York, NY: John Wiley & Sons; 1905.


**How to cite this article:** Salvador JR, Muñoz de la Peña D, Ramirez DR, Alamo T. Predictive control of a water distribution system based on process historian data. *Optim Control Appl Meth*. 2020;41:571–586. https://doi.org/10.1002/oca.2559