2018 European Control Conference (ECC)
June 12-15, 2018. Limassol, Cyprus

Historian Data Based Predictive Control of a Water Distribution Network

Jose R. Salvador, David Muñoz de la Peña, Daniel R. Ramirez and Teodoro Alamo[1]

Abstract— **In this paper, a novel historian data based predictive control strategy is presented and used to control a water distribution network simulated using the EPANET software. 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. The predictive controller computes the current control actions as a weighted sum of past control actions so that a performance cost over a prediction horizon is minimized. This predictive strategy does not need an explicit model of the process and it is well suited to control applications of large and complex processes such as water distribution networks. To limit the computational burden, 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 a initial state close to the current state of the process. Furthermore, other parameters different from the initial state can be considered when choosing the historian subset (e.g., the set point values).**

I. INTRODUCTION

Model predictive control (MPC) techniques rely on the use of a model to predict the evolution of the system state over a prediction horizon and thus compute the values of control signals that minimizes a certain performance cost over that horizon. Therefore, the use of a model is a key concept in these techniques. While there are many applications in which obtaining a prediction model is not a problem, there are also large and complex processes that makes the task of identifying a good model very difficult. Moreover, in those cases, the resulting model (if identified) may be too complex to be used with most MPC techniques, as they result in an optimization problem difficult to solve on line each sampling time, even if efficient MPC code is used [1]. This situation is common in the control of large infrastructures such as water distribution networks in which simplified models are often used [2], [3], [4]. However a simplified model may not capture the nonlinear behavior of, eg., node pressures.

The idea of designing a predictive controller without an explicit model has been proposed as a possible solution to the problems mentioned before. This idea has been developed using different strategies, that in a broad sense can be considered based on machine learning techniques. In [5] the controller is given the ability to predict the process future evolution from current process measures by embedding an identification procedure, in this case the subspace identification method. In [6], a different technique, named nonlinear set membership ([7]) is used to obtain an approximate model, derived directly from data, with the advantage of getting 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 [8]. Other machine learning techniques like regression trees and ensemble learning [9] or reinforcement learning [10] have been also applied to MPC.

This paper presents a novel approach to data driven predictive control that it is purely model free in the sense that a process model is never obtained directly or indirectly as a byproduct of the application of some learning method. The proposed strategy relies on the assumption that the historian data of a process control system that has been operated long enough, contains enough knowledge to predict the future closed loop evolution of the process. This knowledge is in the form of data from closed loop trajectories of the process operated possibly by different control strategies. These data can be used to obtain a future control sequence optimal in relation to some performance cost. The key idea is to look in the historian for a set of operating conditions that are close or similar to the current ones and compute the optimal control sequence from a combination of all the closed loop trajectories that evolve from the set of operating conditions. This would be a predictive control strategy in the sense that use 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. On the other hand the proposed strategy could be conceptually linked to approaches on explicit MPC based on interpolation among a set of off-line computed solutions, like [11], but in this case the measured data, and not the controllers, are considered.

Furthermore, the proposed control strategy can be applied to large processes provided that there is enough data, for example water distribution networks. Water distribution networks 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 network (a well known case study) using the EPANET software.

The paper is organized as follows: In Section II the problem formulation is presented. The controller formulation is introduced in Section III. The use of hyperparameters in the controller design is addressed in Section IV. The results of the application of the controller to the water distribution network case study are shown in Section V. The paper ends with some conclusions in Section VI.

II. PROBLEM FORMULATION

The system considered throughout the paper will be represented by a discrete nonlinear model:

$$x(t+1) = f(x(t), u(t)) \quad (1)$$

*This work was supported by the MINECO-Spain and FEDER Funds under project DPI2013-48243-C2-2-R and by the University of Seville under grant 2014/425.

[1]The authors are with the Department of Systems Engineering and Automation, Sevilla University, Spain email: {jrsalvador,dmunoz,danirr,talamo}@us.es

TABLE I

DATABASE STRUCTURE.

	TraId	k	x	u
1	1	0	$x_1(0)$	$u_1(0)$
2	1	1	$x_1(1)$	$u_1(1)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
p	1	p	$x_1(p)$	$u_1(p)$
$p+1$	2	0	$x_2(0)$	$u_2(0)$
$p+2$	2	1	$x_2(1)$	$u_2(1)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$2 \cdot p$	2	p	$x_2(p)$	$u_2(p)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$M \cdot (p-1)+1$	M	0	$x_M(0)$	$u_M(0)$
$M \cdot (p-1)+2$	M	1	$x_M(1)$	$u_M(1)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$M \cdot p$	M	p	$x_M(p)$	$u_M(p)$

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^m$ is the input and f is an unknown 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

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

where $\ell(\cdot, \cdot)$ is a positive definite stage cost function and N defines the prediction horizon. Note that k is used instead t to indicate that stage and performance cost are computed with predictions as they are in future.

Standard model predictive control [12] is based on solving an optimal control finite horizon control problem in which the cost of the predicted trajectory (2) is minimized at each time step to obtain an optimal control sequence that is applied in a receding horizon manner. However, in this work we assume that the model is unknown, so a finite horizon control problem cannot be formulated. Instead we propose in Section III an historian data based predictive control algorithm.

A. Database structure

As the function that models the system f is unknown, a database will be used to predict the behavior of the system. It is assumed that the database stores M different closed loop trajectories. These trajectories contain the state vector x, the input signal vector u and the discrete time interval k of different controllers applied in closed loop to the system. The nature of these controllers affect considerably the effectiveness of the historian predictive control.

All the trajectories that form the database have an unique identifier $TraId$ and a length of p discrete time intervals. This fact is translated in p database entries or rows for each trajectory which contains p $\{x(\cdot), u(\cdot)\}$ consecutive pairs (Table I shows the general structure of the database).

III. CONTROLLER FORMULATION

The proposed strategy is similar to other predictive control techniques. The main difference with standard predictive controllers is that a database is used instead a model of the system to compute the optimal control input. In order to predict the future evolution of the system, a combination of the data from different closed-loop trajectories will be used.

The proposed strategy has to solve the following problems:

1) Select a subset of the database with the closed loop trajectories that are likely to give the best prediction of the future evolution of process state. The trajectories selected to solve the problem should have an state close to the current state of the system $x(t)$, and should drive the system towards the origin to minimize the cost function.
2) Calculate the performance cost for each selected candidate trajectory. Note that only a portion of length N of each trajectory is used in the performance cost.
3) Solve an optimization problem to compute the convex combination of the selected trajectories with the lowest cost.

A. Historian data-based predictive control algorithm

In this subsection, we present a distance function to solve the candidate trajectory selection problem, a cost function to evaluate the optimality of each candidate trajectory and an optimization problem to obtain the optimal control sequence.

1) Distance function: Given a state vector $x(t)$, a database DB and a row of the database n_r, the distance function is defined as

$$dist_{DB}(x(t), n_r) = |x(t) - x_{n_r}|_\lambda^2 \tag{3}$$

where x_{n_r} is the state vector associated to row nr and $\lambda \in \mathbb{R}^n$ is a weight vector that relates state components. The distance function $dist_{DB}(x(t), n_r) \in \mathbb{R}$ evaluates how far is a state stored in the data base x_{n_r} from the state vector $x(t)$.

Distance (3) will be used to find the Q rows of the database with the closest stored states to $x(t)$ that also have at least $N-1$ consecutive remaining rows in their trajectories. These rows will be considered the initial states of Q closed loop trajectories with a length of N time steps. Assuming that the first state of the candidate trajectory q is stored in row n_r with a slight abuse of the notation we define the states of the q-th candidate trajectory $x_q(k)$ as the k-th row ahead state after the initial row n_r of the candidate trajectory. This notation is extended to the stored control inputs as $u_q(k)$.

2) Performance index function: Once the selection of the database trajectories is made and the Q candidate trajectories are obtained, a performance cost associated to each candidate trajectory is computed. To this end, a cost function similar to (2) is used to evaluate the cost J_q of each candidate:

$$J_q = \sum_{k=0}^{N-1} \ell(x_q(k+1), u_q(k)). \tag{4}$$

3) Optimization problem of the proposed strategy: The optimal control sequence will be the convex combination of the control sequences of each candidate trajectory which has the smallest cost, thus it will be computed as the solution of:

$$\begin{aligned} \min_{\beta \in \mathbb{R}^Q} \quad & \sum_{q=1}^{Q} \beta_q \cdot J_q \\ s.t. \quad & x(t) = \sum_{q=1}^{Q} \beta_q \cdot x_q(0) \\ & \sum_{q=1}^{Q} \beta_q = 1 \\ & \beta_q \geq 0 \quad \forall\, q \in [1, \ldots, Q] \end{aligned} \tag{5}$$

where β_q is the optimization variable that indicates the weigh vector of the convex combination. The optimal control action to be applied $u^*(t)$ is computed as:

$$u^*(t) = \sum_{q=1}^{Q} \beta_q \cdot u_q(0) \tag{6}$$

while, like in any MPC scheme, the solution of (5) is applied in a receding horizon manner. Note that the constraint $x(t) = \sum_{q=1}^{Q} \beta_q \cdot x_q(0)$ ensures that the convex combination of the trajectories starts from $x(t)$ and thus the convex combination of the candidate trajectories is an estimation of the process state trajectory if the convex combination of the candidate input trajectories is applied.

B. Feasibility issues

The cardinality of the candidate trajectories Q is important because larger values carry higher computational burdens in order to possibly take into account candidates that are farther from $x(t)$ in distance terms. On the opposite side, smaller values of Q could produce feasibility problems in (5).

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 $x(t) \notin S_3$, 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, $x(t) \in S_4$, so the minimization problem is feasible.

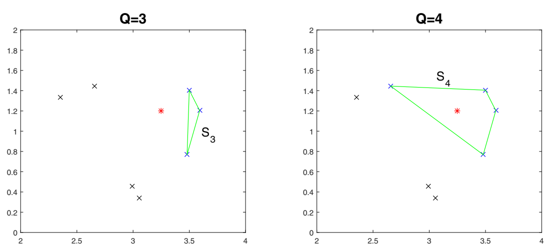

Fig. 1. Feasibility problem in $\mathbb{R}^2$: left unfeasible, right feasible.

The solution to feasibility problems could be to increase the value of Q to find a convex envelope that contains $x(t)$. 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 the input corresponding to the nearest point of the convex envelope of the candidates or some form of extrapolation procedure. Assuming that the function f that model the system is linear, the stage cost function is convex and the trajectories $x_q(k)$ and $u_q(k)$ are feasible and converge to the origin, then it would be relatively straightforward to prove that a convex combination of $x_q(k)$ and $u_q(k)$, like the one produced by the solution of (5) results in a stable and feasible convergent solution. Similar ideas have been used previously [13], [14].

IV. HYPERPARAMETERS

Given the database structure defined in section II-A, hyperparameters are defined as additional information of the controllers that generate the trajectories that can be used to improve the effectiveness of the proposed controller[1]. This additional information can be stored in a separate table. Examples of hyperparameters could be 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.

In order to take into account the hyperparameters in the proposed predictive control scheme:

1) The distance (3) should take into account the stored hyperparameters in the database in order to select the Q candidate trajectories.
2) The hyperparameters should be taken into account in the optimization problem (5) either in the constraints or in the functional cost.

In section V we present an example in which the information of the reference of the controller of each trajectory stored in the database is used as an hyperparameter in the proposed controller.

V. EXAMPLES OF THE HISTORIAN PREDICTIVE CONTROL IN A WATER DISTRIBUTION NETWORK

In order to illustrate the proposed strategy, an example of a water distribution network is proposed. The Richmond water distribution network is a documented and well known case study ([15]) that can be simulated, using the EPANET hydraulic simulation software ([16]). This case study describes a system that it 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, see [17]. Figure 2 shows the Richmond water distribution network diagram that is composed by 6 tanks, 7 pumps, 41 nodes of which 11 are demand nodes, 44 pipes of which 8 are unidirectional pipes and 1 source. Note that for a given tank

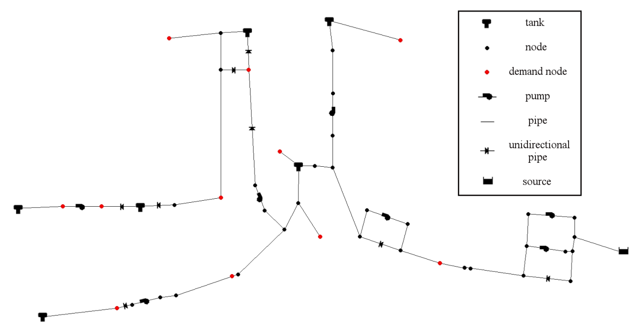

Fig. 2. Richmond water distribution network diagram.

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 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 6 tanks, that is $x \in \mathbb{R}^6$. Demands are considered disturbances, modelled by the disturbance signal vector $d \in \mathbb{R}^{11}$.

It is assumed that the disturbance signal vector, in this example the water demand vector, is composed by a periodic

[1]Without loss of generality we assume here that all the trajectories in the historian have values for all the hyperparameters

TABLE II

MAXIMUM AND MINIMUM SAFETY TANK WATER LEVELS (METERS) IN RICHMOND NETWORK.

	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

signal with a random component, that is:

$$d(t) = d_p(t) + d_r(t) \tag{7}$$

where d_p is the periodic signal that satisfies $d_p(t) = d_p(t+T)$ with $T \in \mathbb{N}$ and d_r is the random signal with mean $\mu_{dr} = 0$ and standard deviation $\sigma_{dr} << \max d_p$. Furthermore, demands signals have a daily period.

In order to attain the control objective, water flows are used as manipulated variables, thus the input signal vector $u \in \mathbb{R}^7$ will contain the water flows that have to be attained using each pump in the network. Note that in the Richmond case study (as in many water distribution networks), pumps are usually operated with an ON-OFF mode which defines the control laws as switching logic. 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 are considered discrete time intervals of 1 hour ($t, k \in \mathbb{N}$) and low level switching logic intervals of $\frac{1}{24}$ hours $= 2.5$ minutes. To minimize the number of pump switches, a duty cycle policy consisting on applying all the control effort in a single pulse is used.

The Richmond water distribution network has 6 tanks, but only 5 of them can be directly controlled. Thus, in the controller only those five tanks are considered for the purposes of computing the control signals. Note, however, that the EPANET simulation uses the whole system.

The cost function in these examples will be a pure tracking error penalty stage cost:

$$\ell(x_c(k)) = (x_c(k) - x_r)^2 \tag{8}$$

where $x_c(k), x_r \in \mathbb{R}^5$ are the levels of the 5 controllable tanks and the reference values for the levels of each tank respectively.

The database stores the closed loop trajectories of four different controllers each one based on a different set of relays with the same hysteresis but different set points. For each controller there are stored in the database 100 trajectories each one with 96 hours of closed loop simulated operation of the network. Each of the trajectories starts with random initial values of the tank level that satisfy the minimum and maximum safety constraints of table II.

Notice that it is not necessary to know the controller structure in order to create the database. The database follows the format shown in table I. In this case, the time stamp k refers the hour of the row within the trajectory, with $k \in [0, 23]$, repeated for four days (96 hours) for each trajectory. All the trajectories start at 7 AM of day 1 of the simulation (thus each trajectory ends at 6 AM of day 4).

Note that 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 $\mathbb{R}^6$ due to the 5 controllable tanks and the time stamp.

A. Example 1

This first example uses only the process state (i.e., $x(t)$) and the current time stamp when building the candidate subset of the database. The example is a closed loop simulation of 120 hours with initial state:

$$x(0) = [3, 2.44, 1.58, 1.5, 0.99, 1.51]$$

and the reference used is:

$$x_r = [2.9403, 3.4206, 1.3018, 1.7808, 1.9066].$$

This reference is equal to the reference of one of the controllers used to create the database. The distance function expressed in (3) is applied to each row n_r of the database DB and for this example it takes the following form:

$$dist_{DB}(x(t), t, n_r) = |(x(t) - x_{n_r})|^2_\lambda + \lambda_t tp_{DB}(t, n_r) \tag{9}$$

where t is the current simulation hour in the $[0, 23]$ range, x_{n_r} is the state vector stored in the corresponding row n_r of the database, λ is a weighted vector for the deviation between the current tank levels and those in the row n_r of the database which, in these examples, takes the value:

$$\lambda = [1.0831, 1, 1.825, 1.7299, 1.6667]$$

, while $\lambda_t = 0.2$ is the value that weights the time term $tp_{DB}(t, n_r)$ which penalize the distance between the current time t and the time k of the row n_r defined as:

$$tp_{DB}(t, n_r) = \min\{(t - k_{n_r})^2, (t + 24 - k_{n_r})^2\} \tag{10}$$

where k_{n_r} is the time stamp k of the corresponding row of the database. Note that because both t and k_{n_r} are in the $[0, 23]$ range, tp_{DB} must take into account this and ensure that the 23^{th} hour of a day has the same distance to the following hour (hour 0 of next day) than the distance between hour 0 and hour 1 of the next day. Hence the definition of tp_{DB} in (10).

The performance index function defined in (4) is used taking into account the stage cost defined in (8). Although the reference of each really controller has not been taken into account, the controller will retain its set point abilities because the current tank level reference is already taken into account in the stage cost (8) and the cost function penalizes candidate trajectories that do not finish near x_r at the end of the prediction horizon.

The optimization problem solved in each time instant is defined in (5) and the control input applied is calculated as in (6).

Figure 3 shows the water levels in the six tanks of the Richmond network during the closed loop simulation using EPANET. Note that the minimum and maximum levels and the reference level are represented for each tank (in red dashed and black dashed respectively). The tank x_5 is not controllable and there is not any reference signal in its level graphic.

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 $d(t)$ computed at each hour of the simulation as:

$$PC(x_c(t)) = \sum_{k=0}^{N-1} \ell(x_c(t+k)) \tag{11}$$

where $x_c(t+k)$ is the real value of the controllable tank levels in the closed loop simulation and $N = 24$ because the daily periodicity. Note that in this system, the instantaneous

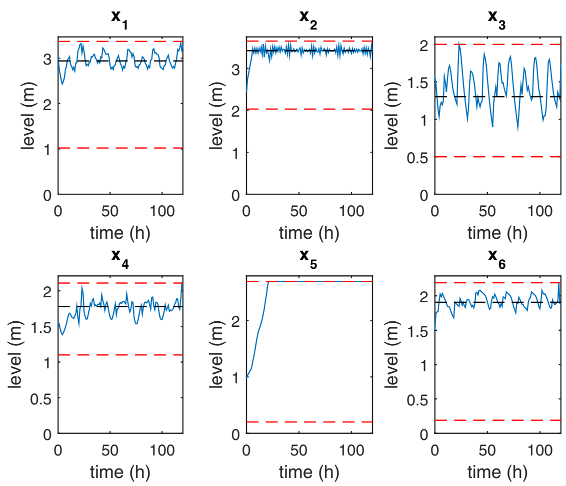

Fig. 3. Tank water levels and state references with proposed controller in closed loop.

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 $d(t)$ will have an impact on the behavior of $PC(x_c(t))$.

Figure 4 shows the evolution of $PC(x_c(t)$ for the 4 relay based controllers and the proposed strategy. It is clear that the historian predictive controller obtains the best results, even better than the best relay controller which is the one with the set point equal to x_r.

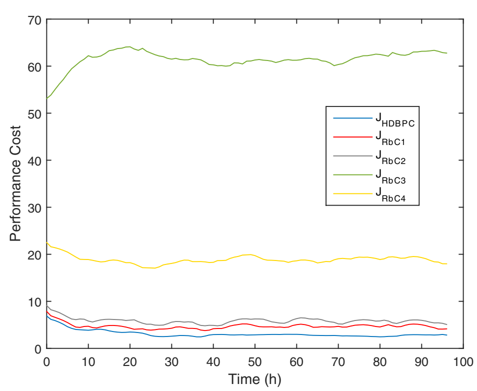

Fig. 4. Performance cost comparison with proposed controller and database relay control laws.

B. Example 2

In this example the proposed strategy is applied taking into account the set point value of the closed loop trajectories stored of the database. The reference x_r for each relay based controller is shown in table III and those values are the mid point between the lower and upper commutation levels of each pump, These values are not shown because of the lack of space (note that pumps 1, 2 and 3 controls tank 1, so x_r for that tank is based on the average switching levels of those pumps). So, an hyperparameter table is stored and it

TABLE III
REFERENCE (METERS) FOR THE DATABASE CONTROLLERS.

	x_r1	x_r2	x_r3	x_r4	x_r6
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

relates each trajectory of the database with the set point of the controller.

According to Section IV, 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.

Firstly, hyperparameters are used when building the subset of Q candidate trajectories. Defining $x^r_{n_r}$ as the set point value of the trajectory that the row n_r of the database belongs, the distance function in (9), can be modified adding a term that penalizes candidate trajectories that reaches references $x^r_{n_r}$ that are far to the reference of the problem x_r, that is:

$$\begin{aligned} dist_{DB}(x(t), x_r, t, n_r) = \quad & |x(t) - x_{n_r}|^2_\lambda \\ & + \lambda_t tp_{DB}(t, n_r) \\ & + |x_r - x^r_{n_r}|^2_{\lambda_r} \end{aligned} \tag{12}$$

where λ_r is a weighted vector for the deviation between the current reference and the reference of the row n_r of the database which, in these examples, takes the value $\lambda_r = 10\lambda$.

Secondly, the functional cost is also affected by the hyperparameters. In the previous example, the stage cost take into account the reference, but the reference x_r is not a convex combination of the optimal sequence of β_i values. This can affect the set point tracking capabilities of the controller. Instead of forcing that the reference will be a convex combination of the optimal solution as a hard constraint, compromising the feasibility of (5), it is proposed to use it as a soft constraint, solving the following optimization problem:

$$\begin{aligned} \min_{\beta} \quad & \sum_{q=1}^{Q} \beta_q \cdot J_q + (x_r - \sum_{q=1}^{Q} \beta_q \cdot x^r_q)^2 \\ s.t. \quad & x(t) = \sum_{q=1}^{Q} \beta_q \cdot x_q(0) \\ & \sum_{q=1}^{Q} \beta_q = 1 \\ & \beta_q \geq 0 \quad \forall\, q \in [1, \dots, Q] \end{aligned} \tag{13}$$

Figure 5 shows the level trajectories of the closed loop simulation with hyperparameters. Figure 6 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 easily be seen that the use of hyperparameters leads to better set point regulation and to a lower performance cost.

C. Example 3

In the previous subsections all the simulations were carried using a set point that matches the stored x_r of one of the relay based controllers. If the set point is different from anyone stored in the database, the set point regulation will be affected. In this section it is shown a closed loop simulation of 120 hours with initial state:

$$x(0) = [3, 2.44, 1.58, 1.5, 0.99, 1.51]$$

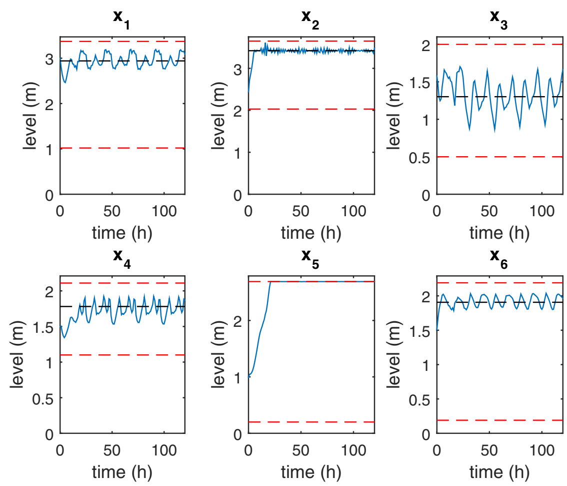

Fig. 5. Tank water levels and state references with proposed strategy with hyperparameters in closed loop.

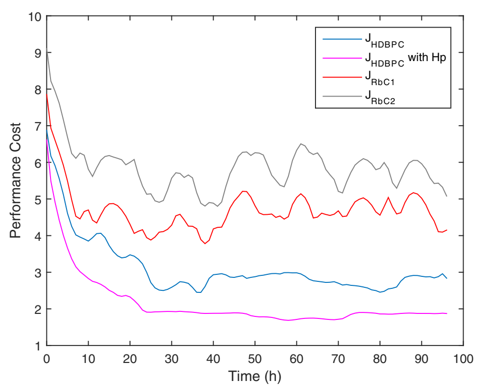

Fig. 6. Performance cost comparison with relay based controllers and historian predictive control with and without hyperparameters.

and reference:

$$x_r = [2.7236, 3.2539, 1.2685, 1.7025, 1.6566].$$

This reference does not match anyone of the controllers (see table III) used to create the database. Figure 7 shows the performance cost comparison for both proposed and relay based controllers. As expected, the controller that uses the hyperparameters performs better than the other as it takes into account the previously unused reference when building subset Q and solving the optimization problem. Both, with and without hyperparameters, improve the performance cost obtained with controllers that conform the database.

VI. CONCLUSIONS

This paper has presented an efficient algorithm to solve a predictive control problem that it is tailored for systems with an unknown model function. It solves the problem using past closed-loop trajectories and control actions stored in a database. The proposed algorithm can increase the effectiveness of the solution using additional information of the controllers used to generate the database historians.

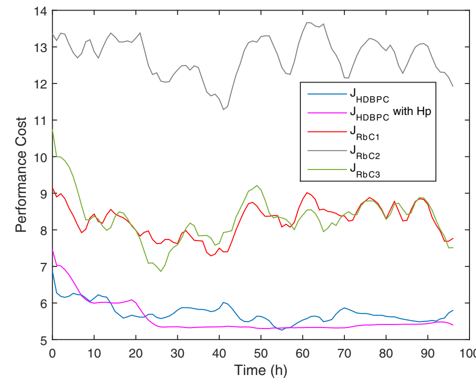

Fig. 7. Performance cost comparison with relay based controllers and historian predictive control with and without hyperparameters for a reference not present in the database.

REFERENCES

[1] B. Houska, H. J. Ferreau, and M. Diehl, “Acado toolkitan open-source framework for automatic control and dynamic optimization,” *Optimal Control Applications and Methods*, vol. 32, no. 3, pp. 298–312, 2011.

[2] M. Brdys and B. Ulanicki, *Operational Control of Water Systems: Structures, Algorithms and Applications*. Prentice Hall, 1994.

[3] G. Cembrano, G. Wells, J. Quevedo, R. Pérez, and R. Argelaguet, “Optimal control of a water distribution network in a supervisory control system,” *Control Engineering Practice*, vol. 8, no. 10, pp. 1177 – 1188, 2000.

[4] C. Ocampo-Martinez, V. Puig, G. Cembrano, and J. Quevedo, “Application of predictive control strategies to the management of complex networks in the urban water cycle [applications of control],” *IEEE Control Systems*, vol. 33, no. 1, pp. 15–41, 2013.

[5] R. Kadali, B. Huang, and A. Rossiter, “A data driven subspace approach to predictive controller design,” *Control Engineering Practice*, vol. 11, no. 3, pp. 261 – 278, 2003.

[6] M. Canale, L. Fagiano, and M. Signorile, “Nonlinear model predictive control from data: a set membership approach,” *International Journal of Robust and Nonlinear Control*, vol. 24, no. 1, pp. 123–139, 2014.

[7] M. Milanese and C. Novara, “Unified set membership theory for identification, prediction and filtering of nonlinear systems,” *Automatica*, vol. 47, no. 10, pp. 2141 – 2151, 2011.

[8] D. Limon, J. Calliess, and J. Maciejowski, “Learning-based nonlinear model predictive control,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 7769 – 7776, 2017.

[9] A. Jain, F. Smarra, and R. Mangharam, “Data predictive control using regression trees and ensemble learning,” in *Proceedings of the 2017 Conference on Decision and Control*. IEEE, 2017.

[10] H. Shah and M. Gopal, “Model-free predictive control of nonlinear processes based on reinforcement learning,” *IFAC-PapersOnLine*, vol. 49, no. 1, pp. 89 – 94, 2016, 4th IFAC Conference on Advances in Control and Optimization of Dynamical Systems ACODS 2016.

[11] J. A. Rossiter, Y. Ding, B. Pluymers, J. A. K. Suykens, and B. D. Moor, “Interpolation based robust MPC with exact constraint handling,” in *Proceedings of the 44th IEEE Conference on Decision and Control*, 2005, pp. 302–307.

[12] J. Rawlings and D. Mayne, *Model Predictive Control:Theory and Design*. Madison, Wisconsin: Nob Hill Publishing, 2009.

[13] G. Valencia-Palomo and J. Rossiter, “Efficient suboptimal parametric solutions to predictive control for PLC applications,” *Control Engineering Practice*, vol. 19, no. 7, pp. 732 – 743, 2011.

[14] P. Colaneri, R. H. Middleton, Z. Chen, D. Caporale, and F. Blanchini, “Convexity of the cost functional in an optimal control problem for a class of positive switched systems,” *Automatica*, vol. 50, no. 4, pp. 1227 – 1234, 2014.

[15] J. E. van Zyl, D. A. Savic, and G. A. Walters, “Operational optimization of water distribution systems using a hybrid genetic algorithm,” *Journal of Water Resources Planning and Management*, vol. 130, no. 2, pp. 160–170, 2004.

[16] “Epanet: Software that models the hydraulic and water quality behavior of water distribution piping systems,” https://www.epa.gov/water-research/epanet.

[17] Y. Wang, J. R. Salvador, D. M. de la Peña, V. Puig, and G. Cembrano, “Periodic nonlinear economic model predictive control with changing horizon for water distribution networks,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 6588 – 6593, 2017.