Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Economic nonlinear predictive control of water distribution networks based on surrogate modeling and automatic clustering

Felix Fiedler [*,***] **Andrea Cominola** [**,***] **Sergio Lucia** [*,***]

[*] *Chair of Internet of Things for Smart Buildings, Technische Universität Berlin, Einsteinufer 17, 10587 Berlin, Germany*
[**] *Chair of Smart Water Networks, Technische Universität Berlin, Straße des 17. Juni 135, 10623, Berlin, Germany*
[***] *Einstein Center Digital Future, Technische Universität Berlin, Wilhelmstraße 67, 10117 Berlin, Germany*

Abstract: The operation of large-scale water distribution networks (WDNs) is a complex control task due to the size of the problem, the need to consider key operational, quality and safety-related constraints as well as because of the presence of uncertainties. An efficient operation of WDNs can lead to considerable reduction in the energy used to distribute the required amounts of water, leading to significant economic savings. Many model predictive control (MPC) schemes have been proposed in the literature to tackle this control problem. However, finding a control-oriented model that can be used in an optimization framework, which captures nonlinear behavior of the water network and is of a manageable size is a very important challenge faced in practice. We propose the use of a data-based automatic clustering method that clusters similar nodes of the network to reduce the model size and then learn a deep-learning based model of the clustered network. The learned model is used within an economic nonlinear MPC framework. The proposed method leads to a flexible scheme for economic robust nonlinear MPC of large WDNs that can be solved in real time, leads to significant energy savings and is robust to uncertain water demands. The potential of the proposed approach is illustrated by simulation results of a benchmark WDN model.

Keywords: model predictive control, water distribution networks, machine learning.

1. INTRODUCTION

Water distribution networks (WDNs) are large-scale interconnected systems that need to be operated reliably, while complying with operational constraints as well as water supply and quality standards, under varying climate and demand conditions. Water resources have the largest energy intensity in the urban metabolism of cities (Chini and Stillwell, 2018), with the water distribution phase often contributing the largest energy intensity share of water provision (Spang and Loge, 2015). Energy-related costs can constitute up to 65% of a utility's operating budget (Boulos et al., 2001).

The optimal management of WDNs can lead to considerable energy reduction and related economic savings. A case study in Valencia (Spain), for instance, demonstrated that potential savings of approximately 17% of the operational costs could be obtained via optimal pump scheduling and valve control, while a similar example in Wallan (Victoria, Australia) estimated a reduction in the average daily pumping costs of 20.6% with optimal operation of the WDNs (Broad et al., 2010). However, the optimal operation of such systems is a challenging task of increasing complexity and importance, due to the growing size of urban areas and the presence of multiple, potentially conflicting objectives.

Model predictive control (MPC) is an advanced control technique that can deal with nonlinear multivariable systems, including constraints and other control goals different than traditional set-point tracking (Mayne and Rawlings, 2009). For these reasons, it is an increasingly popular control method applied in many different fields and case studies and it has been already widely studied in the context of water distribution networks (Biscos et al., 2003; Cembrano et al., 2011; Ocampo-Martinez et al., 2012).

The application of MPC for WDNs faces two important challenges: (i) the presence of uncertainty and (ii) the size of the model for real-world networks. To deal with the uncertainty of the model parameters and water demands, different approaches based on chance constraints (Grosso et al., 2014), Gaussian processes (Wang et al., 2016) or scenario-tree methods (Sampathirao et al., 2018) have been recently proposed.

⋆ Felix Fiedler acknowledges the support of the Helmholtz Einstein International Berlin Research School in Data Science (HEIBRiDS), German Aerospace Center (DLR), and Technische Universität Berlin.

Possible solutions to obtain computationally tractable MPC schemes for large-scale networks include the consideration of hierarchical (Duzinkiewicz et al., 2005; Grosso et al., 2013) and decentralized MPC schemes (Ocampo-Martinez et al., 2012) that avoid the centralized detailed optimization of all network elements at the same time. Alternatively, surrogate models that simplify the hydraulic equations used in the modeling of the WDNs can be used. Surrogate models based on quadratic functions were presented by Nowak et al. (2018) while linear models were used by Sampathirao et al. (2018). Some works have used artificial neural networks (ANNs) to obtain models to be coupled with genetic optimization algorithms (Broad et al., 2005), or deep learning techniques (Wu and Rahman, 2017) to derive predictive models. However, nonlinear surrogate models based on deep learning have not been exploited to be used in a gradient-based optimization framework typical of nonlinear MPC schemes, which is one of the main contributions of our work.

Obtaining a surrogate model for each relevant node in a large network can still be a difficult task, since often the main challenge is the amount of nodes considered in the network and not the complexity of the model. A reduction of the network nodes often requires the use of heuristics (Broad et al., 2005) or detailed system knowledge (Wang et al., 2017). As an additional contribution of this work, we propose the use of data-based clustering techniques to simplify the network model that needs to be learned. Clustering techniques in the context of WDNs have been previously studied (e.g. Di Nardo et al., 2015; Kirstein et al., 2015), but were not used for the design of MPC controllers. A clustering method is especially convenient in an MPC setting as it mitigates the effect of uncertain water demand predictions, as we illustrate in the simulation results.

Compared to similar works, such as Wang et al. (2017), we propose the combination of clustering, surrogate modeling and nonlinear MPC, showing the advantages of deep learning compared to traditional ANNs. Our method is available with an open source implementation [1], does not rely on the detailed knowledge of the underlying model equations and does not need control-oriented models, but just requires a simulator of the water distribution network. We show the potential for energy savings of an economic nonlinear MPC scheme compared to a simple rule-based controller for the benchmark case study of the C-Town WDN. Finally, our proposed clustering method mitigates the effects of uncertain demands and the proposed controller is able to robustly control the system even when an error up to 20% affects the prediction of water demands.

The remainder of the paper is organized as follows. Section 2 describes the considered WDN and its modeling assumptions. The proposed clustering method is described in Section 3, while the surrogate modeling based on deep learning is presented in Section 4. The economic NMPC formulation is explained in Section 5, the simulation results for a benchmark WDN are presented in Section 6 and the paper is concluded in Section 7.

[1] Source code available at: `https://github.com/4flixt/2019_WNTR_Surrogate_Model`

2. OPTIMAL OPERATION OF LARGE-SCALE WATER DISTRIBUTION NETWORKS

The problem of optimal operation of large-scale WDNs has been investigated in the literature in the last 50 years (Mala-Jetmarova et al., 2017), with primary focus on optimal pump operation to target savings in energy use and related cost, while ensuring that demands and water quality standards in the network are satisfied.

In general, a discrete-time model of a drinking water network includes difference and algebraic equations, based on mass continuity and energy, which can be written as:

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

$$z(k) = h(x(k), u(k), d(k)), \tag{1b}$$

where the dynamic states $x(k)$ describe the levels of the tanks that form part of the water network at time step k. The control inputs $u(k)$ include the operation conditions for the valves and pumps. The algebraic states $z(k)$ include the pressure in all nodes of the network, resulting from flow balances. The equations are strongly affected by the water demands in each node, denoted by $d(k)$.

Several detailed simulators for large-scale water networks exist in the literature. Arguably, the most widely used software application for modelling drinking water distribution systems is EPANET, developed by the United States Environmental Protection Agency. EPANET can perform extended period simulation of pressurized pipe networks, computing both hydraulic and water quality simulation (Rossman et al., 2000). More recently, EPANET extensions have also been developed to enhance EPANET modeling capabilities for water security, resilience modeling, and hydraulic response to cyber-physical attacks (Taormina et al., 2019).

Yet, EPANET models are not control-oriented models and often include several switches and discrete operation conditions that make them not suitable for the direct application of gradient-based optimization approaches, which are necessary to solve large-scale problems.

2.1 C-Town benchmark problem

We formulate the optimal operation problem for the WDN of C-Town. C-Town is a benchmark medium-sized network first introduced by Ostfeld (2012) and later used in other state-of-the-art studies (e.g. Sousa et al., 2016; Taormina, 2018). The original C-Town WDN features one reservoir providing water to a network composed of 429 pipes, 388 junctions, 7 storage tanks, 5 pump stations with a total of 11 pumps, and 4 valves. In this study, we simplified the original structure of C-Town to reduce the amount of control inputs. The simplification consisted in removing redundancy in pumping stations, i.e. each pumping station is assumed to include only one pump. The resulting modified C-Town WDN, which we use here as a demonstrative case study, is composed of 419 pipes, 378 junctions, 7 storage tanks, 5 pumps, and 4 valves (see Fig. 1).

The C-Town WDN is described by the following variables, summarized in (2a) to (2d): tank levels (b), node pressure (p), node demand (d), valve setting (v), i.e. minor loss coefficient set for Throttle Control Valves and pressure

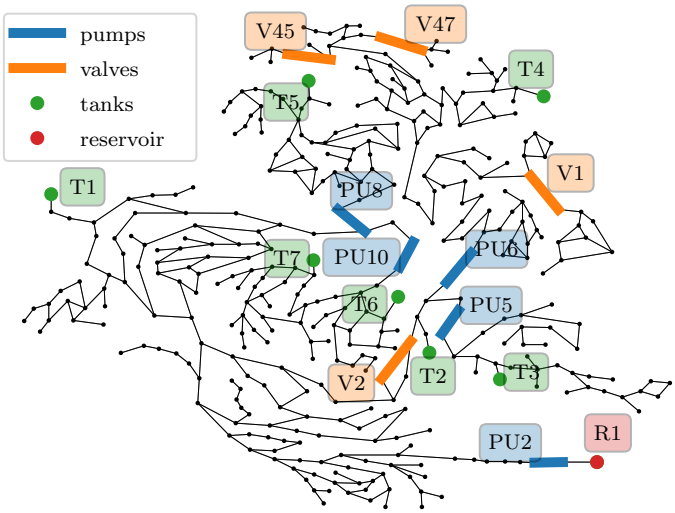

Fig. 1. Overview of the modified C-Town water distribution network, with highlighted pumps and valves (control inputs), tanks (system states) and reservoir.

setting of Pressure Reducing Valves, and relative pump speed (q):

$$x = [b_0, \ldots, b_{n_{\text{tanks}}}] \in \mathbb{R}^7, \tag{2a}$$
$$z = [p_0, \ldots, p_{n_{\text{junctions}}}] \in \mathbb{R}^{378}, \tag{2b}$$
$$u = [v_0, \ldots, v_{n_{\text{valves}}}, q_0, \ldots, q_{n_{\text{pumps}}}] \in \mathbb{R}^9, \tag{2c}$$
$$d = [d_0, \ldots, d_{n_{\text{junctions}}}] \in \mathbb{R}^{378}. \tag{2d}$$

Tank levels in x are the observable state of the C-Town network, while pump and valve settings in u are the control inputs of the system.

In this study, we modeled C-Town in EPANET and simulate it as a demand-driven system. According to the EPANET engine, demand-driven simulation implies that node demands are always satisfied.

2.2 Summary of the proposed algorithm

To deal with the large-scale nature of WDNs, we propose to use, first, a clustering method that leads to a significant reduction of the number of node pressure (2b) and demands (2d) that are explicitly considered in the prediction model. Second, we generate a surrogate model via deep-learning, to obtain a control-oriented model which is used as a prediction model in an NMPC scheme. The data obtained from subsequent NMPC simulations can be used again to improve the quality of the surrogate model in an iterative scheme, as summarized in Algorithm 1.

3. CLUSTERING ALGORITHM

The investigated network in Fig. 1 has a total of 378 distinct junctions with individual demands and pressures. Obtaining a control-oriented surrogate model for such an extensive system is challenging and does not scale well. We therefore propose a clustering-based approach to simplify the problem significantly. Among the existing techniques for clustering, the hierarchical approach with connectivity constraints appears most suited for water networks. The approach is purely data-driven, but considers structural

Algorithm 1 Clustering and surrogate-based economic NMPC of large-scale WDNs

Input: Obtain initial training data (tank levels, junction pressure, demands, pump speed and valve setting) from simulations of available EPANET models or from historical real data.

1. Run hierarchical clustering algorithm (Section 3) to obtain current cluster.
2. Obtain deep learning-based surrogate model using current training data and current clustering (Section 4).
3. Run economic nonlinear MPC using the surrogate model from Step 2 as model for the predictions and a detailed EPANET simulator as reality based emulator (Section 5).
4. If performance not as desired, include the data obtained from Step 3 in your new training data and GOTO Step 1.

information of the network in the form of connectivity constraints. The connectivity matrix $A \in \mathbb{R}^{n_{\text{junc}} \times n_{\text{junc}}}$ can be derived from the EPANET model structure and represents the connections between nodes such that:

$$A_{ij} = \begin{cases} 1 \text{ if junction i and j are connected.} \\ 0 \text{ if junction i and j are not connected.} \end{cases} \tag{3}$$

Nodes are joined based on a variance-minimizing linkage criteria (Ward Jr, 1963) which minimizes the variance of the chosen features. The algorithm stops when the number of clusters matches the user defined parameter n_{clusters}. At each step of the recursive algorithm, clusters $\mathcal{C}_i$ and $\mathcal{C}_j$ are merged, if for any $\mathcal{C}_k$, $k \neq i, j$ the variance of the features of the resulting cluster is such that $\text{var}(\mathcal{C}_i \bigcup \mathcal{C}_j) < \text{var}(\mathcal{C}_i \bigcup \mathcal{C}_k)$. We denote the final clusters as:

$$\mathcal{C}_j, \quad j = 1, \ldots, n_{\text{clusters}}, \tag{4}$$

where each cluster contains a number of $|C_j|$ junctions. As features we use normalized pressures in the individual nodes at various time steps k. We normalize the pressure at each node with an individual parameter according to:

$$\tilde{p}_i(k) = \frac{p_i(k)}{f_{p,i}}, \quad \forall i = 1, \ldots, n_{\text{junc}}. \tag{5}$$

To compute the scaling factors $f_{p,i}$ in (5) we use the mean value of the pressure for each junction from the samples used for training. Normalizing the pressure was found to be an important element to achieve a successful clustering. The scaling is motivated as connected nodes experience head loss on their linkage but will qualitatively follow the same pressure patterns. In Fig. 2, we showcase the pressure over time for two exemplary clusters, which can be identified in Fig. 3, as well as the computed normalized pressure according to (5). Fig. 2 shows that the average normalized pressure of the cluster is a good representation of the normalized pressure of each individual node. We compute the average normalized pressure for cluster j at each point in time k as:

$$\bar{p}_j(k) = \frac{1}{|\mathcal{C}_j|} \sum_{i \in \mathcal{C}_j} \tilde{p}_i(k). \tag{6}$$

The computed mean is used as a condensed representation of the original variables (2b). We evaluate the clustering performance by computing the normalized pressure variance $\sigma_j^2(k) = \text{Var}(\tilde{p}_i(k))$ with $i \in C_j$ for clusters j for all

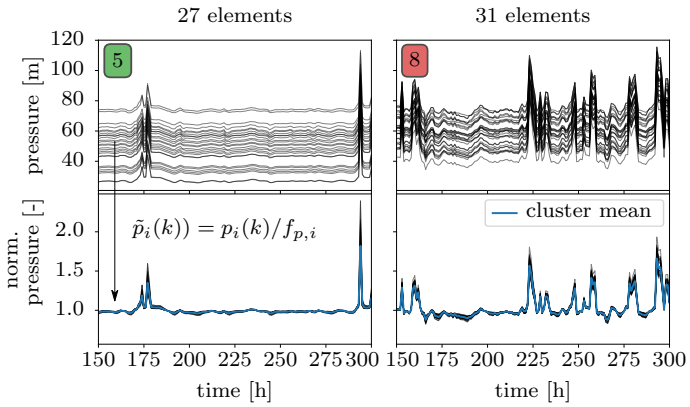

Fig. 2. Temporal evolution of pressure in clusters 5 and 8 (see Fig. 3) and the computed normalized pressure.

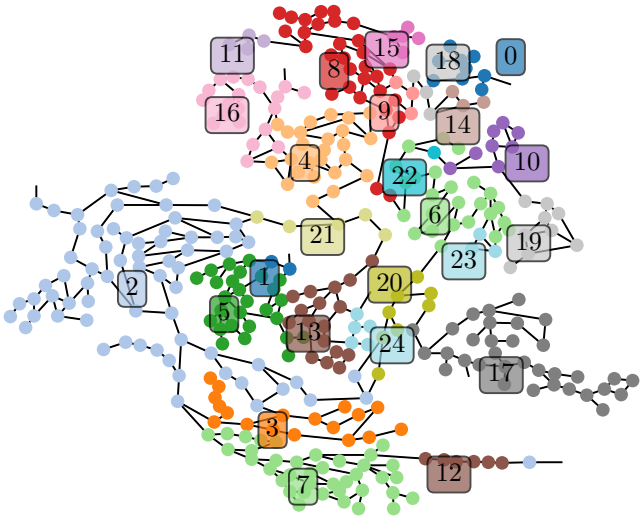

Fig. 3. Overview of the clustered junctions for the modified C-Town network.

times k. To obtain a global performance indicator we compute the mean of $\sigma_j^2(k)$ over all clusters j and times k. We choose $n_{\text{clusters}} = 25$ clusters. It shows that performance deteriorates with less individual clusters, while increasing the number has only a marginal effect. Clustering was performed in Python with scikit-learn (Pedregosa, 2011) with the feature agglomeration method. The resulting clustered nodes are presented in Fig. 3. We furthermore used the obtained clusters based on pressure similarities to group the demands in our network:

$$\bar{d}_j(k) = \sum_{i \in \mathcal{C}_j} d_i(k). \tag{7}$$

This is motivated because sections of the network which experience similar pressure patterns are likely to be affected by the same pump and valve actions, as well as tank levels. We are thus able to control these sections simultaneously and are interested in their aggregated demand.
While junction pressure and demand are clustered we investigate tank levels (b), valve setting (v) and pump speed (q) individually. In summary the relevant variables and their respective dimensions for the dynamic system used in the remainder of the paper are:

$$x = [b_0, \ldots, b_{n_{\text{tanks}}}] \in \mathbb{R}^7, \tag{8a}$$
$$\bar{z} = [\bar{p}_0, \ldots, \bar{p}_{n_{\text{clusters}}}] \in \mathbb{R}^{25}, \tag{8b}$$
$$u = [v_0, \ldots, v_{n_{\text{valves}}}, q_0, \ldots, q_{n_{\text{pumps}}}] \in \mathbb{R}^9, \tag{8c}$$
$$\bar{d} = [\bar{d}_0, \ldots, \bar{d}_{n_{\text{clusters}}}] \in \mathbb{R}^{25}. \tag{8d}$$

The clustering method significantly reduces the dimension of the algebraic states and demands as can be seen by comparing (2) and (8).

4. SURROGATE MODELING USING DEEP LEARNING

Artificial neural networks have been widely used as basis functions for surrogate modeling. Traditionally, shallow networks with only one intermediate (or hidden) layer are used, because under mild conditions they can approximate arbitrarily well any continuous function (Barron, 1993). Recently, the use of deep neural networks (DNNs) with several hidden layers has been very successful as an efficient approximation method for complex functions, mainly in the field of computer science and artificial intelligence (Silver et al., 2016). The choice of deep networks as opposed to shallow ones is motivated by recent theoretical results that support the increased representation power of deep networks (Safran and Shamir, 2017).

A standard DNN with fully connected layers is a simple sequence of function compositions of an affine transformation and a nonlinear function. A neural network $\mathcal{N} : \mathbb{R}^{n_{\text{in}}} \to \mathbb{R}^{n_{\text{out}}}$ can be written as:

$$\mathcal{N}(\zeta; \lambda) = \begin{cases} \alpha_{L+1} \circ \beta_L \circ \alpha_L \circ \cdots \circ \beta_1 \circ \alpha_1(x) & \text{for } L \geq 2, \\ \alpha_{L+1} \circ \beta_1 \circ \alpha_1(x), & \text{for } L = 1, \end{cases} \tag{9}$$

where the input of the network is $\zeta \in \mathbb{R}^{n_{\text{in}}}$ and the output is $\eta \in \mathbb{R}^{n_{\text{out}}}$, and λ denotes the network parameters. The number of hidden layers L and of neurons in each layer M determine the size of the neural network. If $L = 1$, the neural network is denoted as shallow, while deep neural networks have $L \geq 2$. Each hidden layer is composed of an affine function

$$\alpha_l(\xi_{l-1}) = W_l \xi_{l-1} + b_l, \tag{10}$$

where $\xi_{l-1} \in \mathbb{R}^M$ is the output of the previous layer and $\xi_0 = \zeta$, as well as a nonlinear nonlinear activation function β_l. The nonlinear activation function is a design parameter and usual choices include the rectifier linear units ($ReLU$), the sigmoid function and the hyperbolic tangent (tanh) which is used in this work:

$$\beta_l(\alpha_l) = \frac{e^{\alpha_l} - e^{-\alpha_l}}{e^{\alpha_l} + e^{-\alpha_l}}. \tag{11}$$

The parameters of all layers are summarized in $\lambda = \{\lambda_1, \ldots, \lambda_{L+1}\}$ with

$$\lambda_l = \{W_l, b_l\} \quad \forall l = 1, \ldots, L+1, \tag{12}$$

where W_l are the weights and b_l are the biases describing the corresponding affine functions. The neural network is trained using a set of training input-output data pairs $\{(\zeta^{(1)}, \eta^{(1)}), \ldots, (\zeta^{(N_s)}, \eta^{(N_s)})\}$ where N_s is the number of training data points. The training consists of finding the parameters W_l, b_l that minimize the mean squared error, by solving the following optimization problem for a fixed value of L and M:

$$\lambda^* = \arg\min_{\lambda} \frac{1}{N_s} \sum_{i=1}^{N_s} (\eta^{(i)} - \mathcal{N}(\zeta^{(i)}; \lambda))^2. \tag{13}$$

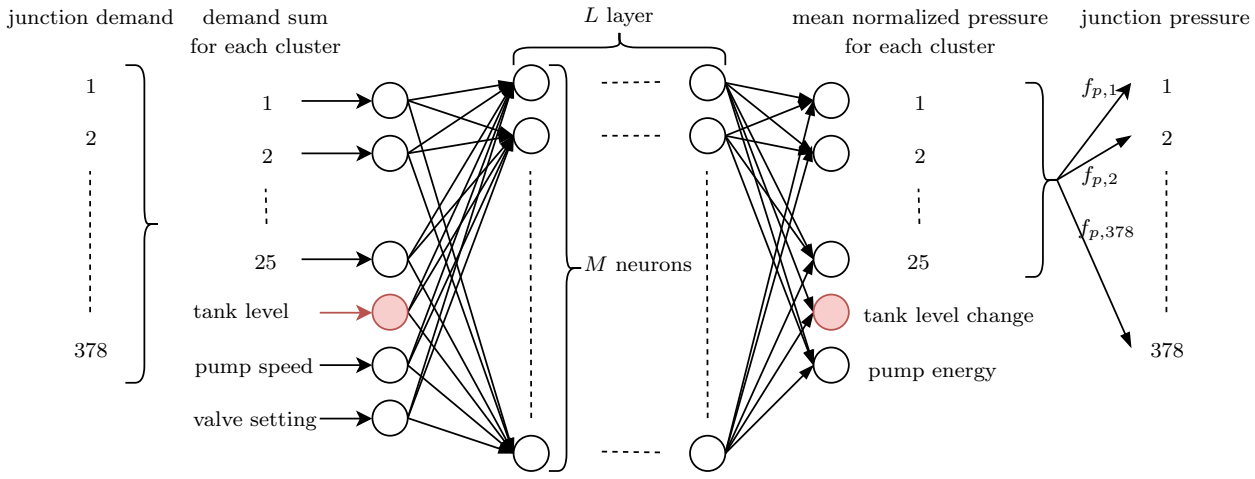

Fig. 4. Overview of the neural network inputs and outputs. The system states are marked in red.

4.1 Input and output structure

In our proposed Algorithm 1, we use DNNs to jointly learn the discrete-time and algebraic equations in (1) for the clustered variables (8). Fig. 4 illustrates the structure of the proposed model. We use as inputs the current clustered junction demands $\bar{d}(k)$, the tank level $b(k)$, the current pump speed $q(k)$ and valve position $v(k)$. As output we have the current mean value of the normalized pressures at each junction $\bar{p}(k)$, the current pump energy consumption $e(k)$ and the change of tank level $\Delta b(k)$. This allows to compute the tank level at the next time step as:

$$b(k+1) = b(k) + \Delta b(k). \tag{14}$$

Choosing the change of tank level versus the next tank level was found to be beneficial for the model accuracy and can be seen as an adaptive normalization. In practice it leads to smoother, more realistic predictions.
In summary, we have:

$$\zeta = \left[\bar{d}(k), b(k), q(k), v(k)\right] \in \mathbb{R}^{41} \tag{15a}$$

$$\eta = \left[\bar{p}(k), \Delta b(k), e(k)\right] \in \mathbb{R}^{37} \tag{15b}$$

Note that we can compute approximate values of the pressure in each junction i by rescaling the normalized pressure in cluster j value with its respective scaling factor:

$$p_i \approx f_{p,i}\bar{p}_j, \quad i \in \mathcal{C}_j \tag{16}$$

In many cases we are only interested in the lowest pressure within one cluster, for which we compute:

$$f_{\min,j} = \min(\{f_i \mid i \in \mathcal{C}_j\}), \tag{17a}$$

$$p^{\text{clust}}_{\min,j} \approx f_{\min,j}\bar{p}_j. \tag{17b}$$

The learned model can be evaluated recursively to achieve the desired prediction horizon in an MPC framework.

4.2 Training and model comparison

All data points used for training are obtained by running EPANET simulations with randomized demand scenarios and different controllers. The training is done using Tensorflow (Abadi, 2015) via Keras (Chollet et al., 2015). We are choosing Adam (Kingma and Ba, 2014), a variant of stochastic gradient descent, as optimizer to solve the problem (13). Training data was normalized, such that $|\bar{\zeta}^{(i)}| \leq 1$ and $|\bar{\eta}^{(i)}| \leq 1$ $\forall i = 1, \ldots, n_{\text{train}}$. We investigated

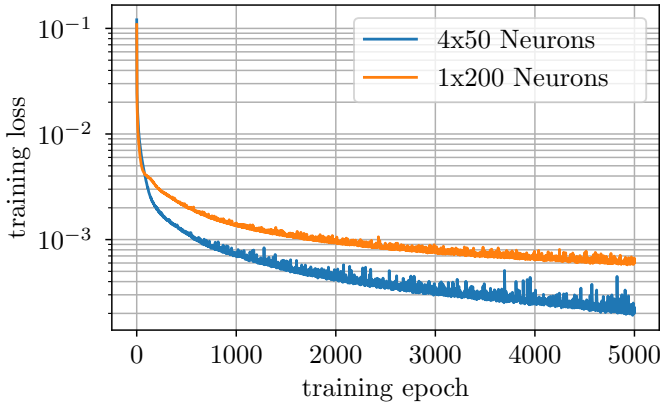

Fig. 5. Comparison of training loss for two investigated neural network architectures with the same number of neurons (nonlinear activations). Model 1 ($L = 4$, $M = 50$) with 11,637 weights and Model 2 ($L = 1$, $M = 200$) with 15,837 weights.

a number of models with different hyper parameters but limited our search space to architectures that are still feasible to optimize in real time. The training progress (mean squared error in (13)) for two investigated models that fulfil this criteria are displayed in Fig. 5. We showcase a shallow vs. a deep architecture with the same number of neurons. The deep architecture with $L = 4$, $M = 50$ has a total of 11,637 weights, whereas the shallow architecture with $L = 1$, $M = 200$ results in 15,837 weights. The training progress in Fig. 5 shows that the deep architecture achieves a significantly better performance. Similar differences between deep and shallow architectures are observed with a smaller number of neurons, leading to worse overall performance. On the other hand, larger architectures only show marginal improvements.
We also investigated a simple linear model for the same input structure and data as for the neural networks. The resulting loss was 3.67×10^{-3}, which is one order of magnitude larger than the the best obtained neural network. Because of the higher training and validation accuracy, we choose the deep learning model as prediction model embedded in the economic NMPC framework explained in the next section.

5. ECONOMIC NONLINEAR MODEL PREDICTIVE CONTROL FORMULATION

For the energy optimal control of the investigated WDN we propose the following economic nonlinear MPC formulation:

$$\underset{\substack{u(0),\dots,u(N-1)\\ x(0),\dots,x(N)\\ \epsilon(0),\dots,\epsilon(N-1)}}{\text{minimize}} \quad \sum_{k=0}^{N-1} \alpha_e e(x(k),u(k),\bar{d}(k)) + \alpha_\epsilon \epsilon(k) + \alpha_u \Delta u(k) \tag{18a}$$

$$\text{subject to:} \quad x(k+1) = f(x(k),z(k),u(k),\bar{d}(k)), \tag{18b}$$

$$z(k) = h(x(k),u(k),\bar{d}(k)), \tag{18c}$$

$$g(x(k),z(k),u(k),\bar{d}(k),\epsilon(k)) \leq 0, \tag{18d}$$

$$x_0 = x_{\text{init}}, \tag{18e}$$

$$\text{for } k = 0,\dots,N-1. \tag{18f}$$

The first term in the mixed objective cost function in (18a) denotes the energy consumption of the pumps. If available, time-varying electricity prices can be incorporated in the formulation. In this work, we consider constant prices so that the economic operation cost is determined by the pump energy consumption which is part of the DNN output (15). Furthermore, we include slack variables ϵ to allow for soft constraints in (18d). The complete set of constraints is defined in (20). The last term of the cost function in (18a) penalizes rapid changes of the control input $\Delta u(k)$. We introduce the tuning factors $\alpha_e,\ \alpha_\epsilon,\ \alpha_u$ to balance the different terms of the objective function.
The difference equation (18b) and the algebraic equations (18c) that describe the dynamic and algebraic states are described by the surrogate model of the clustered network described in the previous subsection. A DNN as depicted in Fig. 4 computes the next state $x(k+1)$ and the algebraic states $z(k)$ as

$$[x(k+1)^T, z(k)^T]^T = \mathcal{N}(\bar{d}(k), x(k), u(k)). \tag{19}$$

The constraints in (18d) describe the maximum and minimum tank levels, as well as the input constraints for pumps and valves for all steps in the prediction horizon $k = 0,\dots,N-1$:

$$b_{\min,i} \leq b_i(k) + \epsilon_{\text{tank}}(k) \leq b_{\max,i}, \quad \forall i = 1,\dots,n_{\text{tanks}}, \tag{20a}$$

$$p_{\min,i} \leq p^{\text{clust}}_{\min,i}(k) + \epsilon^{\text{clust}}_{\min,i}(k) \quad \forall i = 1,\dots,n_{\text{clusters}}, \tag{20b}$$

$$v_{\min,i} \leq v_i(k) \leq v_{\max,i}, \quad \forall i = 1,\dots,n_{\text{valves}}, \tag{20c}$$

$$q_{\min,i} \leq q_i(k) \leq q_{\max,i}, \quad \forall i = 1,\dots,n_{\text{pumps}}, \tag{20d}$$

$$0 \leq e(k), \tag{20e}$$

$$0 \leq \epsilon(k). \tag{20f}$$

As described in Section 3 and 4, we can compute the minimal pressure in each cluster from (17), which implicitly considers the constraint for all 378 junctions while only incorporating $n_{\text{clusters}} = 25$ constraints. The above mentioned slack variables for soft constraints are used for the tank level as well as the minimal pressure for each cluster. Since we are following an economic MPC formulation, softening these constraints is necessary as we are usually in their proximity. Having soft constraints on tank levels and pressures is unintuitive, but results from the demand-driven simulation that is implemented in EPANET. This modelling approach ensures that demands are satisfied, even when water cannot be supplied, thus resulting in negative pressures. Especially, when tank levels are close to zero, this scenario is bound to happen as tanks are the only storage terms in the network. For all other nodes, EPANET assumes stationary conditions under all circumstances. Therefore, we have to interpret negative pressures as not fully satisfied demand, which is undesirable but acceptable on rare occasions. To deal with unavoidable model uncertainty, and following ideas from tube-based MPC Mayne and Rawlings (2009), we tighten the constraints on the tank levels by increasing the lower bounds ($b_{\min}$).

Table 1. Comparison of different MPC solutions with the reference controller for different lower bounds of the tank levels.

	MPC for different $b_{\min}$			**Ref.**
	1.5	0.5	0.0	
energy [MWh]	106.49	105.56	99.97	111.57
energy saved [%]	4.55	5.39	10.39	-
cons. viol. [%]	0.05	0.13	0.51	0.0
avg. cons. viol. [m]	2.13	5.12	5.17	0.00

6. RESULTS

Due to the economic formulation of the MPC objective function in (18a), the main focus of investigating the proposed controller is energy consumption, while ensuring that the demand is satisfied under varying conditions. This can be realized by introducing a constraint tightening, i.e. increasing the lower bound on the tank levels in the form of a soft constraint. We investigate several options for the lower bound $b_{\min}$, which is applied to all tanks. The effect of this lower bound for the tank levels is demonstrated in Fig. 6, where we compare two different $b_{\min}$ values for the proposed MPC approach with the original rule-based controller. In this rule-based controller originally included in C-Town, pumps and valves are opened (closed) only when tank levels reach minimum (maximum) threshold values to guarantee the system can satisfy the expected demand. As expected, economic operation of the system leads to lower tank levels in general. At least one tank level (T1) remains close to zero. The effect of the increased lower bound $b_{\min}$ is visible in the mean values of the tank levels over time (horizontal lines in Fig. 6).

In Table 1 we showcase the energy consumption of the pumps over the simulation period of one month for the three investigated MPC controllers, as well as the rule-based reference controller. The computed energies are also expressed as relative savings in comparison to the reference controller. Furthermore, we display the constraint violation, expressed as percentage of instances where $p_i < 0$ over all nodes and time steps, as well as the mean constraint violation.

As we can see in Table 1, our proposed MPC controller leads to energy savings as high as 10.39% if no tightening of constraints is used ($b_{\min} = 0$). This comes at the cost of increased constraint violations, meaning that we cannot satisfy the demand in isolated instances. Nevertheless, constraint violations occur only during 0.51% of the operating time for each node. Constraint violations can be further reduced by tightening the lower bound of the tank level as shown in Table 1, achieving a trade-off between constraint violations and energy savings.

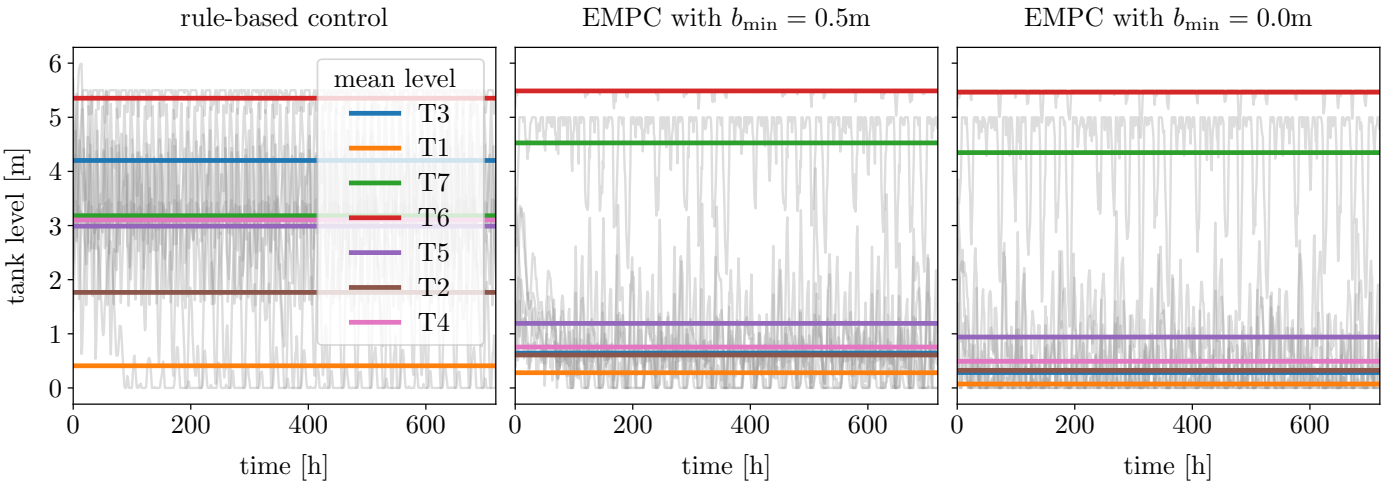

Fig. 6. Temporal evolution of the tank levels (system states) over the course of the simulation time (one month). Comparison of different values for the lower bound (soft constraint) for two EMPC solutions with the original rule-based controller. The mean value of the tank levels over time is marked with a horizontal line.

Table 2. MPC performance with $b_{\min} = 0.5$ m with error in the water demand predictions.

	Error demand predictions	
	0%	±20%
energy [MWh]	105.56	105.11
energy saved [%]	5.39	5.80
cons. viol. [%]	0.13	0.32
avg. cons. viol. [m]	5.12	9.59

Solving our proposed nonlinear MPC problem at each time step requires on average 2.5 s on a standard laptop, which enables the real-time implementation of the method.

6.1 Robust economic NMPC

Another feature of the proposed algorithm is its implicit robustness to demand uncertainty, which is a common problem for WDN. This features arises from the applied clustering method, where the demand for each node in a given cluster is summed to form an aggregated cluster demand. Fluctuations of individual nodes within a cluster are likely to compensate each other, especially for larger clusters. We investigate the robustness of our approach by disturbing the demand prediction with uniform noise in the range of ±20 %. The comparison of energy consumption and constraint satisfaction for the case of $b_{\min} = 0.5$ m are given in Table 2. The results show that the proposed NMPC scheme is able to robustly control the system even under the presence of significant uncertainty in the water demand forecasts.

7. CONCLUSIONS

Optimal operation of distribution water networks is a challenging problem due the large-scale and nonlinear nature of the system as well as the presence of important uncertainty and constraints. To deal with this problem, this paper proposes first to perform a clustering based on normalized pressures to reduce significantly the number of nodes that need to be explicitly considered in the model. We then obtain a control-oriented surrogate model based on the EPANET simulator, any other available simulator or real data and show that the use of deep learning networks leads to a better surrogate accuracy compared to standard shallow networks or linear models.

The deep learning-based surrogate model is embedded in an efficient nonlinear model predictive control framework that directly optimizes the economic performance of the system. By means of a simulation study of the C-town benchmark, we show that the proposed strategy leads to energy savings over 10% and, because of the proposed clustering, it is very robust to uncertain water demand predictions and it can be used in real time. The framework and the data used to obtain the results are openly available.

Future work includes the analysis of more advanced robust MPC strategies Lucia et al. (2013); Sampathirao et al. (2018) and stochastic approaches Lorenzen et al. (2017) that can consider other sources of uncertainty explicitly.

REFERENCES

Abadi, M. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

Barron, A.R. (1993). Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information theory*, 39(3), 930–945.

Biscos, C., Mulholland, M., Le Lann, M., Buckley, C., and Brouckaert, C. (2003). Optimal operation of water distribution networks by predictive control using MINLP. *Water Sa*, 29(4), 393–404.

Boulos, P.F., Wu, Z., Orr, C.H., Moore, M., Hsiung, P., and Thomas, D. (2001). Optimal pump operation of water distribution systems using genetic algorithms. In *Distribution system symposium*.

Broad, D.R., Dandy, G.C., and Maier, H.R. (2005). Water distribution system optimization using metamodels. *Journal of Water Resources Planning and Management*, 131(3), 172–180.

Broad, D.R., Maier, H.R., and Dandy, G.C. (2010). Optimal operation of complex water distribution systems

using metamodels. *Journal of Water Resources Planning and Management*, 136(4), 433–443.

Cembrano, G., Quevedo, J., Puig, V., Pérez, R., Figueras, J., Verdejo, J., Escaler, I., Ramón, G., Barnet, G., Rodríguez, P., et al. (2011). Plio: A generic tool for real-time operational predictive optimal control of water networks. *Water Science and Technology*, 64(2), 448–459.

Chini, C.M. and Stillwell, A.S. (2018). The state of US urban water: data and the energy-water nexus. *Water Resources Research*, 54(3), 1796–1811.

Chollet, F. et al. (2015). Keras. `https://keras.io`.

Di Nardo, A., Di Natale, M., Giudicianni, C., Musmarra, D., Santonastaso, G.F., and Simone, A. (2015). Water distribution system clustering and partitioning based on social network algorithms. *Procedia Engineering*, 119(1), 196–205.

Duzinkiewicz, K., Brdys, M., and Chang, T. (2005). Hierarchical model predictive control of integrated quality and quantity in drinking water distribution systems. *Urban Water Journal*, 2(2), 125–137.

Grosso, J., Ocampo-Martínez, C., Puig, V., and Joseph, B. (2014). Chance-constrained model predictive control for drinking water networks. *Journal of Process Control*, 24(5), 504–516.

Grosso, J., Ocampo-Martínez, C., and Puig, V. (2013). Learning-based tuning of supervisory model predictive control for drinking water networks. *Engineering Applications of Artificial Intelligence*, 26(7), 1741 – 1750.

Kingma, D.P. and Ba, J. (2014). Adam: A method for stochastic optimization.

Kirstein, J.K., Albrechtsen, H.J., and Rygaard, M. (2015). Topological clustering as a tool for planning water quality monitoring in water distribution networks. *Water Supply*, 15(5), 1011–1018.

Lorenzen, M., Dabbene, F., Tempo, R., and Allgöwer, F. (2017). Constraint-tightening and stability in stochastic model predictive control. *IEEE Transactions on Automatic Control*, 62(7), 3165–3177.

Lucia, S., Finkler, T., and Engell, S. (2013). Multi-stage nonlinear model predictive control applied to a semi-batch polymerization reactor under uncertainty. *Journal of Process Control*, 23, 1306–1319.

Mala-Jetmarova, H., Sultanova, N., and Savic, D. (2017). Lost in optimisation of water distribution systems? a literature review of system operation. *Environmental Modelling & Software*, 93, 209–254.

Mayne, D. and Rawlings, J. (2009). *Model Predictive Control: Theory and Design*. Nob Hill Publishing.

Nowak, D., Krieg, H., and Bortz, M. (2018). Surrogate models for the simulation of complex water supply networks. In *Proceedings of the 1st International WDSA/CCWI Joint Conference*.

Ocampo-Martinez, C., Barcelli, D., Puig, V., and Bemporad, A. (2012). Hierarchical and decentralised model predictive control of drinking water networks: Application to barcelona case study. *IET control theory & applications*, 6(1), 62–71.

Ostfeld, A. (2012). Battle of the water calibration networks. *Journal of Water Resources Planning and Management*, 138(5), 523–532.

Pedregosa, F. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.

Rossman, L.A. et al. (2000). Epanet 2: Users manual. *US Environmental Protection Agency. Office of Research and Development*.

Safran, I. and Shamir, O. (2017). Depth-width tradeoffs in approximating natural functions with neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, 2979–2987. JMLR. org.

Sampathirao, A.K., Sopasakis, P., Bemporad, A., and Panos Patrinos, P. (2018). GPU-Accelerated Stochastic Predictive Control of Drinking Water Networks. *IEEE Transactions on Control Systems Technology*, 26(2), 551–562.

Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, 529, 484.

Sousa, J., Muranho, J., Marques, A.S., and Gomes, R. (2016). Optimal management of water distribution networks with simulated annealing: The c-town problem. *Journal of Water Resources Planning and Management*, 142(5), C4015010.

Spang, E.S. and Loge, F.J. (2015). A high-resolution approach to mapping energy flows through water infrastructure systems. *Journal of Industrial Ecology*, 19(4), 656–665.

Taormina, R. (2018). Battle of the attack detection algorithms: Disclosing cyber attacks on water distribution networks. *Journal of Water Resources Planning and Management*, 144(8), 04018048.

Taormina, R., Galelli, S., Douglas, H., Tippenhauer, N.O., Salomons, E., and Ostfeld, A. (2019). A toolbox for assessing the impacts of cyber-physical attacks on water distribution systems. *Environmental modelling & software*, 112, 46–51.

Wang, Y., Ocampo-Martinez, C., and Puig, V. (2016). Stochastic model predictive control based on Gaussian processes applied to drinking water networks. *IET Control Theory and Applications*, 10(8), 947–955.

Wang, Y., Puig, V., and Cembrano, G. (2017). Non-linear economic model predictive control of water distribution networks. *Journal of Process Control*, 56, 23–34.

Ward Jr, J.H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301), 236–244.

Wu, Z.Y. and Rahman, A. (2017). Optimized deep learning framework for water distribution data-driven modeling. *Procedia Engineering*, 186, 261–268.