Contents lists available at ScienceDirect

Environmental Modelling & Software

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

Uncertainty-aware demand management of water distribution networks in deregulated energy markets

P. Sopasakis [a, *], A.K. Sampathirao [b, 1], A. Bemporad [c], P. Patrinos [a, 2]

[a] KU Leuven, Department of Electrical Engineering (ESAT), STADIUS Center for Dynamical Systems, Signal Processing and Data Analytics & Optimization in Engineering (OPTEC), Kasteelpark Arenberg 10, 3001 Leuven, Belgium
[b] Technische Universität Berlin, Fachgebiet Regelungssysteme, Einsteinufer 17, D-10587 Berlin, Germany
[c] IMT School for Advanced Studies Lucca, Piazza San Ponziano 6, Lucca 55100, Italy

ARTICLE INFO

Article history:
Received 15 May 2017
Received in revised form
13 October 2017
Accepted 28 November 2017
Available online 15 December 2017

Keywords:
Drinking water networks
Bid-based energy market
Stochastic model predictive control
Graphics processing units
Scenario trees
Open-source software

ABSTRACT

We present an open-source solution for the operational control of drinking water distribution networks which accounts for the inherent uncertainty in water demand and electricity prices in the day-ahead market of a volatile deregulated economy. As increasingly more energy markets adopt this trading scheme, the operation of drinking water networks requires uncertainty-aware control approaches that mitigate the effect of volatility and result in an economic and safe operation of the network that meets the consumers' need for uninterrupted water supply. We propose the use of scenario-based stochastic model predictive control: an advanced control methodology which comes at a considerable computation cost which is overcome by harnessing the parallelization capabilities of graphics processing units (GPUs) and using a massively parallelizable algorithm based on the accelerated proximal gradient method.

Software availability

Name of software: RapidNet.
Hardware required: CUDA-compliant GPU (tested on NVIDIA Tesla 2075).
Software required: CUDA framework v6.0 or higher (including cuBLAS) and rapidjson (http://rapidjson.org/).
Availability: Open-source software, licence: GNU LGPL v3.0. Available online at https://github.com/GPUEngineering/RapidNet.
Program language: CUDA-C++/C++.
First release: 2017

1. Introduction

1.1. State of the art

The explosive proliferation of interconnected sensing, computing and communication devices has marked the advent of the concept of *cyber-physical systems* — ensembles of computational and physical components. In drinking water networks this trend has ushered in new control paradigms where the profusion of data, produced by a network of sensors and stored in a database, is used to prescribe informed control actions (Eggimann et al., 2017; Meseguer and Quevedo, 2017; Solomatine, 2003; Lobbrecht and Solomatine, 2002). Nevertheless, as these data, be they water demand values or electricity prices, cannot be modeled perfectly, the associated uncertainty is shifted to the decision making process.

The high uncertainty in the operation of drinking water networks, as a result of the volatility of future demands as well as energy prices (in a deregulated energy market) is likely to lead to a rather expensive operating mode with poor quality of service (the network may not always be able to provide the necessary amount of water to the consumers). In control engineering practice, this uncertainty is often addressed in a worst-case fashion (Sampathirao et al., 2014; Wang et al., 2015) — if not neglected at all

* Corresponding author.
E-mail addresses: pantelis.sopasakis@kuleuven.be (P. Sopasakis), sampathirao@control.tu-berlin.de (A.K. Sampathirao), alberto.bemporad@imtlucca.it (A. Bemporad), panos.patrinos@esat.kuleuven.be (P. Patrinos).

[1] The work of A. K. Sampathirao was supported by the German Federal Ministry for Economic Affairs and Energy (BMWi), Project No. 0324024A.
[2] The work of P. Patrinos was supported by the KU Leuven Research Council under BOF/STG-15-043.

https://doi.org/10.1016/j.envsoft.2017.11.035
1364-8152/© 2017 Elsevier Ltd. All rights reserved.

— leading to conservative and suboptimal control policies. It is evident that it is necessary to devise control methods which take into account the probabilistic nature of the underlying uncertainty making use of the wealth of available historical data aiming at a proactive and foresightful control scheme which leads to an improved closed-loop performance. These requirements necessitate the use of *stochastic model predictive control*: an advanced control methodology where at every time instant we determine a sequence of *control laws* which minimizes the *expected value* of a performance index taken with respect to the distribution of the uncertainty (Mesbah, 2016). Optimization-based approaches for the operational management of water networks have been studied and are well established in engineering practice (Mala-Jetmarova et al., 2017).

Indeed, scenario-based stochastic model predictive control (SSMPC) has been shown to lead to remarkable decrease in the operating cost and improvement in the quality of service of drinking water networks (Sampathirao et al., 2017a). In SSMPC, the uncertain disturbances are treated as random variables on a discrete sample space without assuming any parametric form for their distribution (Calafiore and Campi, 2006). The scenario approach was identified in a recent review as a powerful method for mitigating uncertainty in environmental modeling related to water management (Horne et al., 2016). Although this approach offers a realistic control solution as it is entirely data-driven, this comes with considerable computational burden as the resulting optimization problems are of particularly large scale (Goryashko and Nemirovski, 2014; Grosso et al., 2014a). This has rendered the use of SSMPC prohibitive and has hindered its applicability. Indeed, hitherto there have been used only conventional model predictive control approaches (Ocampo-Martinez et al., 2010; Bakker et al., 2013), robust worst-case formulations (Sampathirao et al., 2014; Ocampo-Martinez et al., 2009; Tran and Brdys, 2009) and stochastic formulations where the underlying uncertainty is assumed to be normally identically independently distributed (Wang et al., 2016a; Grosso et al., 2016). Note that it has been observed that demand prediction errors are typically follow heavy-tail distributions which cannot be well approximated by normal ones (Hutton and Kapelan, 2015).

In this paper, we present a software for the fast and efficient solution of such problems harnessing the immense computational capabilities of graphics processing units (GPU) building up on our previous work (Sampathirao et al., 2017a, 2016, 2015).

There has been recently a lot of interest in the development of efficient methods for stochastic optimal control problems such as stochastic gradient methods (Themelis et al., 2016a), the alternating directions method of multipliers (ADMM) (Kang et al., 2015) and various decomposition methods which can lead to parallelizable methods (Carpentier et al., 2010; Defourny et al., 2012) (the most popular being the stochastic dual approximate dynamic programming (Jiang et al., 2014), the progressive hedging approach (Carpentier et al., 2013) and dynamic programming (Bertsekas, 2000)). There have been proposed parallelizable interior point algorithms for two-stage stochastic optimal control problems such as (Klintberg and Gros, 2017; Lubin et al., 2011; Kang et al., 2014; Chiang et al., 2014) and an *ad hoc* interior point solver for multistage problems (Hübner et al., 2017). However, interior point algorithms involve complex steps and are not suitable for an implementation on GPUs which can make the most of the capabilities of the hardware. Additionally, interior point methods cannot accommodate complex non-quadratic terms in the cost function such as soft constraints (distance-to-set functions).

At large, not many software and libraries are available for stochastic optimal control; one of the very few one may find on the web is JSPD, a generic Java stochastic dynamic programming library. QUASAR is a commercial tool for scenario-based stochastic optimization. One of the most popular tools in the toolbox of the water networks engineer is PLIO (Cembrano et al., 2011), which implements MPC algorithms. This work covers the yawning gap between engineering practice and the latest developments in control and optimization theory for drinking water networks. These results can also be applied for the control of other infrastructure with similar structure such as power grids (Hans et al., 2015).

1.2. Contributions and novelty

Despite the fact that SSMPC problems typically involve millions of decision variables, the associated optimization problems possess a rich structure which can be exploited to devise parallelizable *ad hoc* methods to solve the problem more than an order of magnitude faster than commercial solvers running on CPU.

The architecture of our implementation comprises three independent modules: (i) the *network* module, (ii) the *energy prices and water demands forecaster* and (iii) the *control module*. The network module provides a dynamical system model which describes the flow of water across the network together with the storage limits of the tanks and the constraints on pumping capacities. The network module defines a *safety storage level* for each tank — a level which ensures the availability of water in case of high demand and the maintenance of a minimum required pressure. The forecasters produce a *scenario tree*, that is, a tree of likely future water demands and energy prices, upon which a contingency plan is made by minimizing a cost function which quantifies the operating cost and the quality of service. Such scenario trees are constructed from historical data of energy prices and water demands. The control module computes flow set-points, which are sent to the pumping stations and valves, by solving a scenario-based stochastic model predictive control problem over a finite prediction horizon.

The proposed stochastic model predictive controller leads to measurable benefits for the operation of the water network. It leads to a more economic operation compared to methods which do not take into consideration the stochastic nature of the energy prices and water demands. In this paper, we assess the performance of the controlled network using three key performance indicators: (i) the *economic index*, (ii) the *safety index*, which quantifies the extent of violation of the safety storage level requirement and (iii) the *computational complexity index* with which we assess the computational feasibility of the controller. Simulation results are provided using data from the water network of Barcelona and the energy market of Austria. The advantages of the adopted control methodology are combined with the computational power of GPUs, which enables us to solve problems of very large scale.

1.3. Software

Our implementation is available as an open-source and free software which can be readily tailored to the needs of different water networks modifying the parameters of its modules. The implementation is done entirely in CUDA-C++ and it can be configured either programmatically or using configuration files. The adopted object-oriented programming model is amenable to extensions and users may specify their own predictive models, scenario trees, cost functions, dynamical models and constraints. Our results are accompanied by extensive benchmarks and the software is verified with unit tests.

2. Modeling

2.1. Hydraulic modeling

The water network involves four types of elements: water tanks, active elements (pumps and valves), mixing nodes and demand sectors. We focus on flow-based water networks where all flows are manipulated variables based on the modeling approaches presented in (Batchabani and Fuamba, 2014; Ocampo-Martinez et al., 2010; Papageorgiou, 1984; Grosso et al., 2014a; Sampathirao et al., 2014).

Water tanks which play a crucial role in demand management as they ensure water supply, obviate the need for continuous pumping and allow water to be pumped when the price of electricity is lower and provide water in cases of unexpected peaks in demand. Tank dynamics are modeled by simple mass balance equations: let $V^j(t)$ be the volume of tank $j=1,\ldots,N_t$ at time t. We denote all flows in the network by q^i, $i=1,\ldots,N_f$. Let $q^i(t)$, $i\in\mathscr{I}_j$ be the controlled inflowing streams to tank j and $q^i(t)$, $i\in\mathscr{J}_j$ be the controlled outflowing streams. Then, the mass balance equation becomes (see Fig. 1a)

$$\frac{\mathrm{d}V^j(t)}{\mathrm{d}t}=\sum_{i\in\mathscr{I}_j}q^i(t)-\sum_{i\in\mathscr{J}_j}q^i(t). \tag{1}$$

The volume in each tank should never exceed a maximum limit V^j_{max} and it should always be above a hard lower limit V^j_{min}, that is,

$$V^j_{\text{min}}\leq V^j(t)\leq V^j_{\text{max}}. \tag{2}$$

In addition, for the sake of service reliability (availability of water when demand rises unexpectedly) and safety, it is required that the level of water remains above a certain level V^j_{safe}. This is allowed to be violated occasionally, when the demand happens to be too high.

Mixing nodes are intersections where flows of water are merged or separated. The mass balance equations for mixing nodes give rise to algebraic constraints of the form

$$\sum_{i\in\mathscr{K}_s}q^i(t)=\sum_{i\in\mathscr{C}_s}q^i(t), \tag{3}$$

where $\mathscr{K}_s$ and $\mathscr{C}_s$ are the sets of indices of the incoming and outgoing flows at node $s=1,\ldots,N_s$ as shown in Fig. 1b.

Pumps and *valves* are used to control the flow of water in the network and transfer it across tanks and to the demand sectors. We treat these as controlled systems — indeed, pumping stations and valves are equipped with local controllers — to which we prescribe flow set-points. The local control systems operate at a sampling rate of about 1HZ, while the operational management of the network updates its decisions at a much slower rate (e.g., hourly). It is reasonable to assume that the local control system equilibrates fast enough to neglect its dynamics in the context of operational control. That said, the flow determined by each pump $i\in\mathscr{P}$ is equal to its prescribed set-point u^i. As shown in Fig. 1c, that is

$$q^i(t)=u^i(t),\ i\in\mathscr{P}. \tag{4}$$

Similarly, as shown in Fig. 1d the flow through each valve is

$$q^i(t)=u^i(t),\ i\in\mathscr{V}. \tag{5}$$

All flows in the network are unidirectional, so we require that $q^i\geq 0$ for all $i=1,\ldots,N_f$. Each pump $i\in\mathscr{P}$ has a maximum *pumping capacity* q^i_{max}, that is we require that

$$0\leq q^i(t)\leq q^i_{\text{max}},\ i\in\mathscr{P}. \tag{6}$$

Demand sectors are the exit nodes of the water from the network towards the consumers as shown in Fig. 1e. At each demand sector $i\in\mathscr{D}$, the mass balance yields

$$q^i(t)=d^i(t),\ i\in\mathscr{D}. \tag{7}$$

We treat $d^i(t)$ as a random process and elaborate on that in Section 2.2.

We may now describe the system dynamics in terms of the state variable $x(t)=(V^j(t))_{j=1,\ldots,N_t}$, the input variable $u(t)=(u^j(t))_{j\in\mathscr{P}\cup\mathscr{V}}$ and the disturbance $d(t)=(d^i(t))_{i\in\mathscr{D}}$. Let $n_u=|\mathscr{P}\cup\mathscr{V}|$ be the dimension of $u(t)$, that is, the total number of pumps and valves. Let $n_d=|\mathscr{D}|$ be the total number of demand nodes. By discretizing the dynamical equation (1) using the exact discretization method [(Chi-Tsong, 1999), Sec. 4.2.1] and taking into account the algebraic constraints stated above, we may write the system dynamics in the following form of a discrete-time linear time-invariant system

$$x_{k+1}=Ax_k+Bu_k+G_dd_k, \tag{8a}$$

$$0=Eu_k+E_dd_k, \tag{8b}$$

where $A\in\mathbb{R}^{N_t\times N_t}$, $B\in\mathbb{R}^{N_t\times n_u}$, $G_d\in\mathbb{R}^{N_t\times n_d}$, $E\in\mathbb{R}^{N_s\times n_u}$ and $E_d\in\mathbb{R}^{N_s\times n_d}$.

The constraints on x_k and u_k can be concisely written as

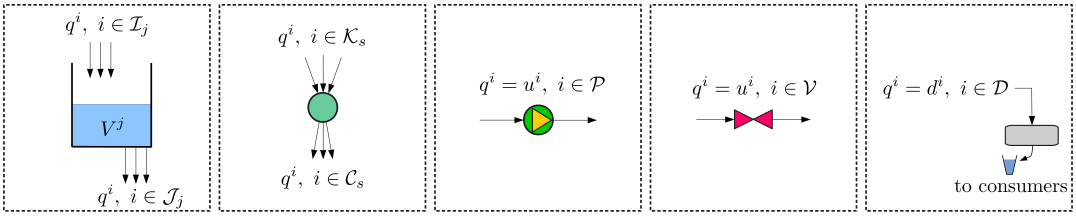
$$x_{\text{min}}\leq x_k\leq x_{\text{max}}, \tag{9a}$$

$$u_{\text{min}}\leq u_k\leq u_{\text{max}}, \tag{9b}$$

with $x_{\text{min}},\ x_{\text{max}}\in\mathbb{R}^{N_t}$ and $u_{\text{min}},\ u_{\text{max}}\in\mathbb{R}^{n_u}$ and $\leq$ is meant in the element-wise sense. These constraints encompass (2) and (6).

2.2. Demands and electricity prices

Water demand has been the main source of uncertainty for the


$q^i,\ i\in\mathcal{I}_j$
V^j
$q^i,\ i\in\mathcal{J}_j$
$q^i,\ i\in\mathcal{K}_s$
$q^i,\ i\in\mathcal{C}_s$
$q^i=u^i,\ i\in\mathcal{P}$
$q^i=u^i,\ i\in\mathcal{V}$
$q^i=d^i,\ i\in\mathcal{D}$
to consumers

(a) Water tank (b) Mixing node (c) Controlled pump (d) Controlled valve (e) Demand sector

Fig. 1. Elements of the water network and the associated flows.

operation of drinking water networks and a lot of attention has been paid on the development of models for its prediction. Prediction methodologies span from simple linear models (Pedregal and Trapero, 2007) to neural networks (Ghiassi et al., 2008; Romano and Kapelan, 2014) and support vector machines (Peña-Guzmán et al., 2016; Msiza et al., 2007), nonlinear multiple linear regression (Yasar et al., 2012), Holt-Winters-type models (Wang et al., 2016b), as well as more complex neuro-fuzzy models (Papageorgiou et al., 2016; Lertpalangsunti et al., 1999). Increased predictive ability can be obtained using exogenous information such as weather forecasts (Bakker et al., 2014) and calendar data (Sampathirao et al., 2014).

In the context of a deregulated wholesale energy market, prices on the day-ahead market are volatile and are often decided on the basis of an auction (bid-based market) among energy companies instead of bilateral agreements with an energy provider. In such cases, energy prices may change on a daily or hourly basis (Puglia et al., 2013). It is then necessary to be able to predict the day-ahead evolution of the prices using past data; several time series analysis methodologies have been developed for that purpose — see (Weron, 2014) and references therein.

In our approach the prediction procedure is decoupled from the control which allows the use of any forecasting methodology without having to modify the controller parameters or implementation. An independent *forecaster* provides estimates of future demands and electricity prices along with an estimation of their uncertainty which is discussed in the next section. At time k, the predicted demands for the future time $k+j$ are estimated by a model which computes $\widehat{d}_{k+j|k}$. Likewise, we denote the predicted electricity prices by $\widehat{\alpha}_{k+j|k}$. Let d_{k+j} and α_{k+j} denote the actual, but unknown at time k, values of the water demands and prices. Then

$$\begin{bmatrix} d_{k+j} \\ \alpha_{k+j} \end{bmatrix} = \begin{bmatrix} \widehat{d}_{k+j|k} \\ \widehat{\alpha}_{k+j|k} \end{bmatrix} + \varepsilon_{j|k}, \tag{10}$$

where $\varepsilon_{j|k}$ is a random variable which corresponds to the j-step-ahead prediction error at time k. At time k, a forecaster provides finite-horizon estimates of the upcoming water demands and electricity prices

$$\widehat{\boldsymbol{d}}_k = \left(\widehat{d}_{k+1|k}, \widehat{\alpha}_{k+1|k}, \ldots, \widehat{d}_{k+H_p|k}, \widehat{\alpha}_{k+H_p|k}\right), \tag{11}$$

where H_p is a *prediction horizon*. This information will then be provided to the controller as we shall discuss in Section 3.

2.3. Uncertainty

It is common in stochastic control-oriented modeling to assume that the errors $\varepsilon_{j|k}$ are independently distributed (Wang et al., 2016a; Grosso et al., 2016). This assumption however neglects the covariance across the times stages — indeed, if at the future time $j=1$ the model has a large prediction error we would rather expect that the prediction error at time $j=2$ is likely to be large too. This motivates the use of scenario trees: discrete representations of the random processes $(\varepsilon_{j|k})_j$ which capture such multistage covariances (Shapiro et al., 2009).

To date, stochastic modeling for drinking water networks in presence of price uncertainty has received little attention — to the best of the authors' knowledge (Eck et al., 2014), is the only relevant reference — and the scenario tree approach has not been used previously.

A scenario tree is a structure such as the one illustrated in Fig. 2. The scenario tree is organized into time instants $j=0,\ldots,H_p$ called *stages* and a number of *nodes* at each stage denoted by $\varepsilon^i_{j|k}$ — these are treated as the possible values of $\varepsilon_{j|k}$. At time $j=0$, the prediction error is always equal to 0 assuming that we observe the current water demands and electricity prices. The corresponding unique node is called the *root* node of the tree. The nodes of the tree at the last stage are known as *leaf nodes*. The number of nodes at stage j is denoted by μ_j. All nodes are identified by a pair $\alpha=(j,i)$, where $j=0,\ldots,H_p$ is the stage index and i is the index of the node in that stage. Each non-leaf node defines a (nonempty) set of *children* $\mathrm{ch}(j,i)$ which are those nodes in next $j+1$ which are linked to (j,i). Conversely, each not except for the root node defines a unique *ancestor* denoted by $\mathrm{anc}(j,i)$. The probability of visiting a node (j,i) starting from the root node and following the tree structure is denoted by p^i_j.

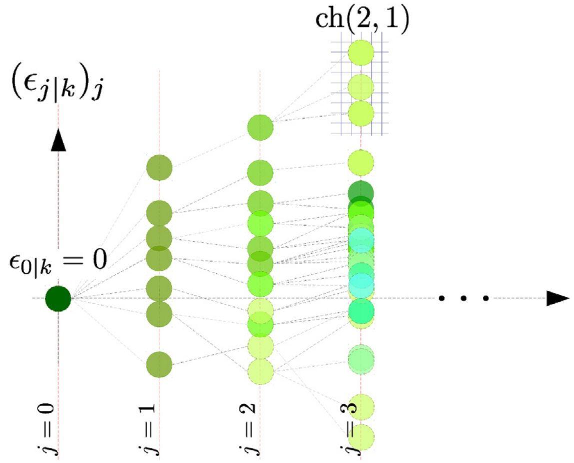


Fig. 2. Scenario tree of the random process $(\varepsilon_{j|k})_j$. At time $j=0$, the prediction error is $\varepsilon_{0|k}=0$; this defines the *root node* of the tree. The children nodes $\mathrm{ch}(2,1)$ of node $(2,1)$ are highlighted in the figure.

Note that the joint demand-price modeling of the uncertainty allows us to cast possible demand-price correlations as in cases of uncertain volume-based pricing.

Using the scenario tree structure, Equation (10) yields

$$\begin{bmatrix} d^i_{k+j} \\ \alpha^i_{k+j} \end{bmatrix} = \begin{bmatrix} \widehat{d}_{k+j|k} \\ \widehat{\alpha}_{k+j|k} \end{bmatrix} + \varepsilon^i_{j|k}, \tag{12}$$

where $j=0,\ldots,H_p$ and $i=1,\ldots,\mu_j$. Here we see that the tree structure of $\varepsilon^i_{j|k}$ induces a corresponding tree structure upon the water demands and electricity prices, namely d^i_{k+j} and α^i_{k+j}. These are the contingent future water demand and electricity price values associated with the prediction error $\varepsilon^i_{j|k}$.

Similarly, Equation (8b) gives

$$Eu^i_{k+j|k} + E_d d^i_{k+j|k} = 0, \tag{13}$$

where $(u^i_{k+j|k})_{j,i}$ become the decision variables of the stochastic optimal control problem we shall present in the following section. The discrete-time system dynamics (8a) becomes

$$x^l_{k+j+1|k} = Ax^i_{k+j|k} + Bu^l_{k+j|k} + G_d d^l_{k+j|k}, \tag{14}$$

where $(j+1,l)\in\mathrm{ch}(j,i)$.

Scenario trees can be constructed from observed sequences of prediction errors which can be easily obtained in practice using methodologies such as (Pflug and Pichler, 2015) or the popular *scenario reduction* method (Heitsch and Römisch, 2009). There exist several other scenario generation algorithms such as clustering-based algorithms (Latorre et al., 2007; Chen and Xu, 2014) and simulation and optimization-based approaches (Beraldi et al., 2010; Glpnar et al., 2004). It is not necessary, however, to update the scenario tree for $\varepsilon_{j|k}$ at every time instant k — it should be updated occasionally to detect changes in the predictive ability of the forecaster or whenever the predictive model is updated.

3. Scenario-based stochastic optimal control

3.1. Control objectives

The cost for the operation of the water network is quantified in terms of three individual costs which have been proposed in the literature (Sampathirao et al., 2014; Cong Cong et al., 2014; Ocampo-Martinez et al., 2009; Ocampo-Martinez and Negenborn, 2015): the economic cost which is related to the treatment cost and electricity required for pumping, the smooth operating cost which penalizes the abrupt operation of pumps and valves and the safety storage cost which penalizes the use of water from the reserves (i.e., allowing the level in the tanks to drop below the safety level).

The *economic cost* quantifies the *production* and *distribution* cost and it is computed by

$$\ell^w(u_k, k) = W_\alpha(\alpha_0 + \alpha_k)' u_k, \tag{15}$$

where $\alpha_0' u_k$ is the water production cost (treatment and acquisition fees), $\alpha_k' u_k$ is the uncertain pumping cost and W_α is a positive scaling factor.

The *smooth operation cost* is defined as

$$\ell^\Delta(\Delta u_k) = \Delta u_k' W_u \Delta u_k, \tag{16}$$

where $\Delta u_k = u_k - u_{k-1}$ and $W_u \in \mathbb{R}^{n_u \times n_u}$ is a symmetric positive definite weight matrix.

The total *stage cost* at a time instant k is the summation of the above costs and is given by

$$\ell(u_k, u_{k-1}, k) = \ell^w(u_k, k) + \ell^\Delta(\Delta u_k).$$

The *safety storage cost* penalizes the drop of water level in the tanks below a given *safety level* x_s. An elevation above this safety level ensures that there will be enough water in unforeseen cases of unexpectedly high demand and also maintains a minimum pressure for the flow of water in the network. This is given by

$$\ell^S(x_k) = W_s \|\max\{0, x_s - x_k\}\|, \tag{17}$$

where W_s is a positive scaling factor.

The state constraints (9a) should be satisfied at all times without however jeopardizing the feasibility of the optimal control problem we have to solve at every time instant. For that, we introduce an additional cost which penalizes the violation of the state constraints as follows

$$\ell^x(x_k) = W_x(\|\max\{0,\ x_{\min} - x_k\}\| + \|\max\{0, x_k - x_{\max}\}\|) \tag{18}$$

where W_x is a positive weight factor.

The scaling factors W_α, W_u, W_s and W_x are the tuning knobs of stochastic MPC as we shall discuss in the following section.

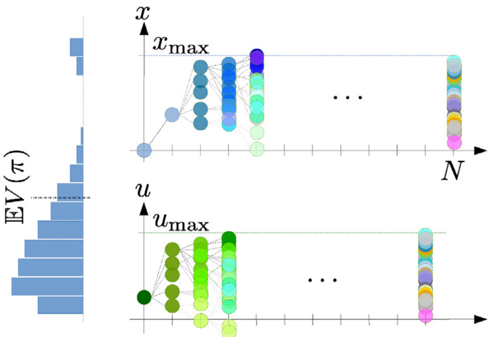


Fig. 3. The concept of scenario-based stochastic optimal control: at every time instant k, we make an optimal contingency plan by minimizing the *expectation* of a cost function V which encodes the operation cost along a finite prediction horizon.

3.2. Stochastic optimal control: problem formulation

In scenario-based stochastic MPC, at every time instant k we solve a stochastic optimal control problem which consists in determining an optimal *contingency plan* for the future course of actions in a *causal fashion*, that is, our future decisions $u_{k+j|k}$ are only allowed to depend on information that will be available to the controller at time $k+j$ (Bertsekas and Shreve, 1996). This was tacitly stated in Equation (14). This concept is illustrated in Fig. 3.

We formulate the following scenario-based stochastic MPC problem with a prediction horizon H_p and decision variables $\boldsymbol{x} = \{u_{k+j|k}, x_{k+j+1|k}\}_{j=0,\ldots,H_p-1}$ where we minimize the expected total cost along the prediction horizon

$$\underset{\boldsymbol{x}}{\text{minimize}}\ \mathbb{E}[V(\boldsymbol{x}, p, q, k)] + V_s(\boldsymbol{x}), \tag{19a}$$

subject to the following constraints

$$x_{k|k} = p,\ u_{k-1|k} = q, \tag{19b}$$

$$x^l_{k+j+1|k} = Ax^i_{k+j|k} + Bu^l_{k+j|k} + G_d d^l_{k+j|k}, \tag{19c}$$

$$Eu^i_{k+j|k} + E_d d^i_{k+j|k}(\varepsilon_j) = 0, \tag{19d}$$

$$u_{\min} \le u_{k+j|k} \le u_{\max}. \tag{19e}$$

In (19a), $\mathbb{E}$ is the expectation operator, V is the cost function given by

$$V(\boldsymbol{x}, p, q, k) = \sum_{j=0}^{H_p-1} \ell\Big(u_{k+j|k}, u_{k+j-1|k}, k+j\Big), \tag{20}$$

and V_s is the total state constraint violation penalty defined as

$$V_s(\boldsymbol{x}) = \sum_{j=0}^{H_p} \sum_{i=1}^{\mu_{j-1}} \ell^x\left(x^i_{k+j|k}\right) + \ell^S\left(x^i_{k+j|k}\right). \tag{21}$$

Note that we have a time-varying cost as the electricity prices change with time.

4. Numerical algorithm

4.1. Problem reformulation

Most modern numerical optimization algorithms such as the (accelerated) proximal gradient algorithm, the alternating directions method of multipliers (ADMM) (Parikh and Boyd, 2014), the Pock-Chambolle method (Chambolle and Pock, 2011), Tseng's forward-backward-forward algorithm (Tseng, 2000) and many another require that the optimization problem be first written in a form

$$\mathscr{P} : \operatorname*{minimise}_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}) + g(H\boldsymbol{x}), \tag{22}$$

where $f : \mathbb{R}^n \to \overline{\mathbb{R}} := \mathbb{R}\cup\{+\infty\}$ and $g : \mathbb{R}^m \to \overline{\mathbb{R}}$ are convex, lower semi-continuous extended-real-valued functions and $H : \mathbb{R}^n \to \mathbb{R}^m$ is a linear operator. Functions f and g are allowed to return the value $+\infty$ to encode constraints; for example, the constraint $x\in C$ is encoded by the *indicator function* of the set C which is

$$\delta(x|C) = \begin{cases} 0, & \text{if } x\in C \\ +\infty, & \text{otherwise} \end{cases} \tag{23}$$

The key question is how to split the optimization problem in (19) so that the resulting formulation is amenable to a fast numerical solution with massive parallelization. For reasons that will be elucidated in Section 4.3 we choose f to be the *smooth* part of the cost (which corresponds to the linear function ℓ^w and the quadratic function ℓ^Δ) plus the indicator of the input-disturbance coupling given in (8b) plus the indicator of the system dynamics in (8a), that is $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ is defined as

$$\begin{aligned} f(\boldsymbol{x}) = {} & \delta\left(u_j^i \middle| \Phi_1\left(d_j^i\right)\right) + \delta\left(x_{j+1}^i, u_j^i, x_j^{\mathrm{anc}(j+1,i)} \middle| \Phi_2\left(d_j^i\right)\right) \\ & + \sum_{j=0}^{H_p-1} \sum_{i=1}^{\mu_j} p_j^i\left(\ell^w\left(u_j^i\right) + \ell^\Delta\left(\Delta u_j^i\right)\right), \end{aligned} \tag{24}$$

where $\Delta u_j^i = u_j^i - u_{j-1}^{\mathrm{anc}(j,i)}$ and $\Phi_1(d)$ is the affine subspace of $\mathbb{R}^{n_u}$

$$\Phi_1(d) = \{u : Eu + E_d d = 0\}, \tag{25}$$

and $\Phi_2(d)$ is the affine subspace of $\mathbb{R}^{2n_x+n_u}$ defined by the system dynamics

$$\Phi_2(d) = \{(z, x, u) : z = Ax + Bu + G_d d\}. \tag{26}$$

Function g is naturally chosen to be the indicator of the set of input constraints plus the total constraint violation penalty function V_s. Note, however, that the same variable $x_{k+j|k}^i$ participates in both functions ℓ^x and ℓ^S. As we shall explain in Section 4.3, this complicates any computations thereon. For that reason we introduce a linear operator $H : \boldsymbol{x} \mapsto \boldsymbol{y} := H(\boldsymbol{x})$ which maps $x_{k+j|k}^i$ to $(x_{k+j|k}^i, x_{k+j|k}^i)$ and $u_{k+j|k}^i$ to itself, that is

$$y_{k+j|k}^i = \left(x_{k+j|k}^i, x_{k+j|k}^i, u_{k+j|k}^i\right), \tag{27a}$$

for $j = 0, \ldots, H_p - 1$ and

$$y_{k+H_p|k}^i = \left(x_{k+H_p|k}^i, x_{k+H_p|k}^i\right). \tag{27b}$$

Then, we define the function $g : \mathbb{R}^m \to \overline{\mathbb{R}}$

$$g(\boldsymbol{y}) = \sum_{\substack{j=0,\ldots,H_p \\ i=1,\ldots,\mu_{j-1}}} \ell^x\left(y_{k+j|k}^{i,(1)}\right) + \ell^S\left(y_{k+j|k}^{i,(2)}\right) + \sum_{\substack{j=0,\ldots,H_p-1 \\ i=1,\ldots,\mu_{j-1}}} \delta\left(y_{k+j|k}^{i,(3)} \middle| \mathscr{U}\right), \tag{28}$$

where $\mathscr{U} := \{u\in\mathbb{R}^{n_u} | u_{\min} \leq u \leq u_{\max}\}$.

The scenario-based optimization problem (19) is now in the form (22) with f and g given by (24) and (28) respectively and H given by (27).

4.2. Convex conjugates and proximal operators

Before we can proceed with the statement of the numerical algorithm for the solution of problem (22) we need to introduce a few mathematical notions. A function $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ is called *proper* if it is not everywhere equal to $+\infty$. It is called *lower semi-continuous* if for every $x\in\mathbf{R}^n$, $\liminf_{z\to x} f(z) = f(x)$. The *domain* of f is the set $\operatorname{dom} f := \{x\in\mathbb{R}^n | f(x) < \infty\}$. It is called κ-strongly convex, for some $\kappa > 0$, if the function $f(x) - \kappa/2\|x\|^2$ is convex.

For a proper, convex, lower semi-continuous function f, we define its *convex conjugate* to be the convex function $f^* : \mathbb{R}^n \to \overline{\mathbb{R}}$ defined as (Rockafellar and Wets, 2009, Chapter 11).

$$f^*(y) = \sup_{x\in\mathbb{R}^n} y'x - f(x). \tag{29}$$

An important property is that if f is κ-strongly convex, then f^* is differentiable with ${}^1/_\kappa$-Lipschitz gradient [(Rockafellar and Wets, 2009), Prop. 12.60] and the gradient of f^* is

$$\nabla f^*(y) = \operatorname*{argmax}_{x\in\Re^n} y'x - f(x). \tag{30}$$

By means of the convex conjugate we may derive the *Fenchel dual* optimization problem of (22) which is [(Bauschke and Combettes, 2011), Sec. 15.3]

$$\mathscr{D} : \min_{y\in\Re^m} f^*(-H'y) + g^*(y). \tag{31}$$

Fenchel duality offers a more powerful framework as compared to the classical Lagrangian duality approach as it allows us to dualize *functions* (by means of their convex conjugates) rather than merely constraints.

Under certain conditions on f and g (see Section 4.3), the two problems are equivalent: their optimal values are equal and given a dual-optimal point $y^\star$ which is a minimizer of (31), the optimal solution of $\mathscr{P}$ is $x^\star = \nabla f^*(-H'y^\star)$. This is referred to as *strong duality*. The reason why we formulate the dual optimization problem $\mathscr{D}$ is because it possesses a favorable structure which can be exploited in the development of fast and parallelizable numerical algorithms (See Section 4.3).

Lastly, for a proper, lower semi-continuous, extended-real valued function $g : \mathbb{R}^m \to \overline{\mathbb{R}}$ we define its *proximal operator* with parameter $\gamma > 0$ to be a function $\operatorname{prox}_{\gamma g} : \mathbb{R}^m \to \mathbb{R}^m$ defined as

$$\operatorname{prox}_{\gamma g}(v) := \operatorname*{argmin}_{x\in\mathbb{R}^m} g(x) + \frac{1}{2\gamma}\|x - v\|^2. \tag{32}$$

Proximal mappings act as generalized projections. For example, the proximal mapping of the indicator function $\delta(\,\cdot \mid C)$ — cf. (23) — of a nonempty, closed, convex set is the projection onto that set, i.e., $\operatorname{prox}_{\gamma\delta(\cdot|D)}(v) = \operatorname{proj}(v|C)$.

The proximal operators of many convex functions (such as Euclidean norm, norm-1, quadratic and linear functions, distance-to-set functions) are easy to compute and typically consist in element-wise operations which can be fully parallelized on a GPU. We shall refer to such functions as *prox-friendly* (Wytock, 2016).

So long as $\operatorname{prox}_{\gamma g}$ is easy to compute, so is $\operatorname{prox}_{\gamma g^*}$ and it can be obtained from the *Moreau decomposition formula* which is

$$\mathrm{prox}_{\gamma g}(v) + \gamma \mathrm{prox}_{\gamma^{-1} g^*}\left({}^{v}\!/_{\gamma}\right) = v. \tag{33}$$

4.3. Accelerated proximal algorithm on the dual optimization problem

The function $f(\boldsymbol{x}) + g(H\boldsymbol{x})$, with f and g defined by (24) and (28) respectively, is proper, convex and piecewise linear-quadratic on its domain, so, following [(Rockafellar and Wets, 2009), Thm. 11.42] there is strong duality. As discussed above, since f is strongly convex, f^* is differentiable with Lipschitz gradient. Function g is written in the form of a *separable sum* — a sum of prox-friendly functions of different arguments (Parikh and Boyd, 2014). Function g is indeed prox-friendly. Let $(\mathrm{prox}_{\gamma g}(v))_{j,i,(s)}$ denote the part of the vector $\mathrm{prox}_{\gamma g}(v) \in \mathbb{R}^m$, indexed by j, i and s, which corresponds to $y^{i,(s)}_{k+j|k}$, for $s = 1, 2, 3$. Then, $(\mathrm{prox}_{\gamma g}(v))_{j,i,(1)}$ and $(\mathrm{prox}_{\gamma g}(v))_{j,i,(2)}$ are computed by virtue of the formula

$$\mathrm{prox}_{\gamma \mathrm{dist}(\cdot|C)}(v) = \begin{cases} v + \dfrac{\mathrm{proj}_C(v) - v}{\mathrm{dist}(v|C)}, & \text{if } \mathrm{dist}(v|C) > \gamma \\ \mathrm{proj}_C(v), & \text{otherwise} \end{cases} \tag{34}$$

where $\mathrm{dist}(\cdot|C)$ is the distance-to-set function and the fact that $\ell^x(x) = W_x \mathrm{dist}(x|[x_{\min}, x_{\max}])$ and $\ell^S(x) = W_s \mathrm{dist}(x|[x_s, +\infty))$. The proximal operator $(\mathrm{prox}_{\gamma g}(v))_{j,i,(3)}$ is simply the projection on $\mathscr{U}$.

Note that although g is prox-friendly, g composed with the linear operator H — as it is in (22) — is not. This is the main reason why we resort to the dual problem (31).

Given the properties of functions f^*, being differentiable with Lipschitz gradient, and g^*, being prox-friendly, we may use Nesterov's accelerated proximal gradient method on the dual problem which produces the sequence

$$w^\nu = y^\nu + \beta_\nu \left(y^\nu - y^{\nu-1}\right), \tag{35a}$$

$$y^\nu = \mathrm{prox}_{\gamma g^*}(w^\nu + \gamma H \nabla f^*(-H' w^\nu)), \tag{35b}$$

with $y^0 = 0$, $y^{-1} = 0$, $\beta^0 = 0$. In (35a) we perform an *extrapolation step* for some $\beta_\nu > 0$ and we perform a dual gradient projection update on the extrapolated vector w^ν. The extrapolation parameters β_ν are parametrized as $\beta^\nu = \theta_\nu(\theta_{\nu-1}^{-1} - 1)$ with $\theta_0 = \theta_{-1} = 1$. Any choice of θ_ν so that $1 - \theta_{\nu+1} \leq \theta^2_{\nu+1}/\theta^2_\nu$. A simple choice is

$$\theta_\nu = \frac{2}{\nu + 2}, \tag{36}$$

for $\nu \geq 1$. Here we choose $\theta_{\nu+1} = {}^1\!/\!_2 \left(\sqrt{\theta_\nu^4 + 4\theta_\nu^2} - \theta_\nu^2\right)$ which satisfies the above requirement with equality.

This algorithm fits into Tseng's Alternating Minimization framework (Tseng, 1990, 1991). It has been found to be suitable for embedded applications as it is relatively simple to implement and it has good convergence properties (the dual variable y^ν converges with rate $O\left({}^1\!/_{\nu^2}\right)$ and an averaged primal iterate converges at $O\left({}^1\!/_{\nu^2}\right)$ as well) (Patrinos and Bemporad, 2014). It involves only matrix-vector operations (additions and multiplications) and it is numerically stable. In the following section we discuss how the involved operations can be massively parallelized in a lock-step fashion (performing the exact same operation on different memory positions) and how the algorithm can be implemented on a GPU.

4.4. Implementation

Because of the definition of f in (24), the computation of the gradient of f^* as defined in problem (30) boils down to the solution of a scenario-based *optimal control* problem where the only constraints are the ones defined by the system dynamics. This problem can be solved by dynamic programming leading to a Riccati-type recursion from stage $k = H_p$ to stage $k = 0$. At each stage operations across all nodes can be fully parallelized. In particular, such a parallelization — assuming that full parallelization is supported by the hardware — equalizes the complexity of the scenario-based Riccati recursion to that of a deterministic one. A detailed exposition of the details of this procedure is available in (Sampathirao et al., 2015, 2017a).

All operations involved in the computation of $\mathrm{prox}_{\gamma g}$ are element-wise operations and can be fully parallelized on a GPU; therefore, the computational cost for applying $\mathrm{prox}_{\gamma g}$ — or, what is the same — $\mathrm{prox}_{\gamma^{-1} g^*}$ via (33) is negligible.

5. Functionality

In this section we present `RapidNet`, a `CUDA-C++` implementation of the accelerated proximal gradient method for the solution of scenario-based stochastic optimal control problems, particularly tailored to the needs of a drinking water network.

5.1. Software structure

The entities involved in `RapidNet` and the relationships among each other are illustrated in Fig. 4 which correspond to classes in the `CUDA-C++` implementation of `RapidNet` (see also Table 1).

The `DwnNetwork` class stores data related to the network topology, physical constraints and network dynamics matrices in equations (8a) and (8b). The user can create instances of `DwnNetwork` simply by passing a JSON file, `network.json`, with the network information.

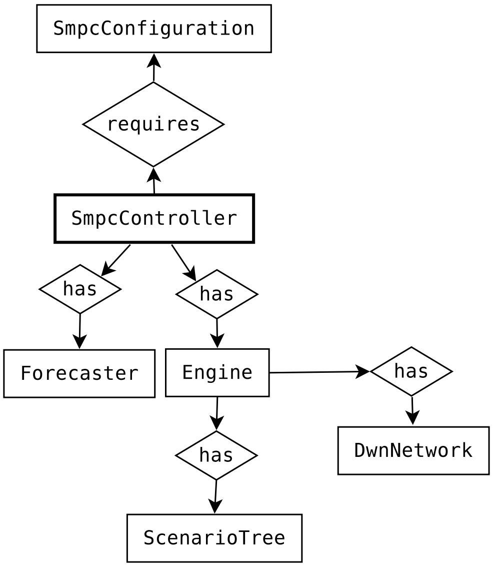


Fig. 4. Entity-Relationship (ER) diagram of the entities in `RapidNet` which reflects the underlying class structure.

Table 1
Description of the major classes and the associated JSON files in RapidNet.

Class name	Description	JSON file
SmpcController	Runs the accelerated dual proximal gradient algorithm and computes a control action to be applied to the water network. Computations are carried out on GPU and the output (flow set-points) can be stored in a JSON file.	controlOutput.json
DwnNetwork	Encapsulates all information related to the topology, dynamics and constraints of the water network.	network.json
ScenarioTree	Scenario-tree representation of the uncertainty in electricity prices and water demands.	scenarioTree.json
Forecaster	An abstract forecaster which predicts the upcoming electricity prices and water demands using some predictive model (implemented by subclassing Forecaster) or reads the forecasts from a JSON file (so that the user can use forecasts from third-party software).	forecaster.json
Engine	Provides essential functionality to SmpcController and manages the GPU-side memory.	
SmpcConfiguration	Contains configuration parameters that are relevant for SmpcController (tuning parameters, solver tolerance, maximum number of iterations).	controllerConfig.json

The ScenarioTree class models the scenario tree structure that represents the uncertainty associated with the volatile energy prices and water demands. This class describes the structure of the scenario tree by assigning a unique index to each node and storing the indexes of the children of each non-leaf node, the ancestor of all nodes except for the root node and the values $d^i_{k+j|k}$ and $\alpha^i_{k+j|k}$ at each node (j,i). An instance of ScenarioTree can be generated using a JSON file, scenarioTree.json.

An SmpcController executes the accelerated proximal gradient algorithm to solve a stochastic optimal control problem at every time instant and compute the control actions to be applied to the water network. Certain functionality is delegated to Engine — a collection of utility methods — which precomputes certain quantities which are associated with the Riccati-type recursion (details can be found in (Sampathirao et al., 2017a)) for the computation of the dual gradient and manages the associated memory on GPU.

The Engine is, in turn, linked with to a DwnNetwork entity which provides all necessary technical specifications for the network (topology, dynamics, constraints) and a ScenarioTree entity which encodes the probabilistic information associated with the prediction errors. Note that the end-user does not have to create instances of Engine or directly interact with it.

A Forecaster provides to the controller estimates of the upcoming water demands and electricity prices. This is an abstract class which can be subclassed with particular model implementations (e.g., ARIMA, SVM, or any other), or the user can provide custom forecasts using any third-party software which exports its forecasts in a JSON file.

SmpcController requires certain configuration parameters which is provided by the entity SmpcConfiguration. There, the user specifies the desired tolerance, maximum number of iterations and can override other solver-specific properties.

Overall, the flow of information in RapidNet is shown in Fig. 5. The end-user initializes an SmpcController object by providing the network topology, the controller configuration and a scenario tree. During real-time operation, the controller receives the network state x_k (which can be provided in a JSON file) and, using a demand/price forecaster, computes a control action which is applied to the system.

In Table 2 we list the main methods of RapidNet. Additional getter methods are available in each class for more advanced use-case scenarios.

5.2. GPU implementation

GPUs were first developed for video applications and, due to the high demand in high-performance graphics, rapidly evolved to powerful hardware featuring hundreds of computation cores. Nowadays, GPUs are used for more than video processing and they are becoming popular for computational purposes including, but not limited to, environmental modeling (Xia and Liang, 2016; Le et al., 2015; Guidolin et al., 2016). By design they are well-suited for data-parallel lockstep applications where the same type of operation is applied to different memory positions. Instructions are sent to the GPU (from the CPU) in the form of *compute kernels*. GPUs offer unprecedented parallelization capabilities provided that the program can be parallelized in a lockstep fashion (the same operation is executed on different memory positions).

CUDA is a parallel computing framework and application programming interface for NVIDIA GPUs used for general-purpose computing. Part of the CUDA framework is cuBLAS, a parallel counterpart of the popular linear algebra library BLAS.

In RapidNet, at every time instant k, the method controlAction in SmpcController returns the control action that is to be applied to the water network (pump and valve set points). All computations involved in this method are either summations or matrix-vector multiplications which can be parallelized across the nodes of a stage. These multiplications are implemented using the function cublasSgemmBatched of cuBLAS and vector additions

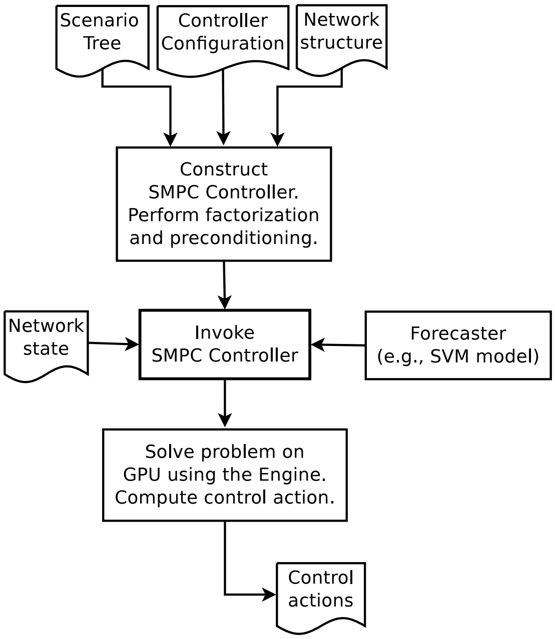

Scenario Tree
Controller Configuration
Network structure
Construct SMPC Controller. Perform factorization and preconditioning.
Network state
Invoke SMPC Controller
Forecaster (e.g., SVM model)
Solve problem on GPU using the Engine. Compute control action.
Control actions

Fig. 5. Flow of information in RapidNet.

Table 2
Key methods in RapidNet.

Class	Method	Description
`SmpcController`	`controlAction`	Computes control action using the accelerated proximal gradient algorithm to solve the scenario-based stochastic optimal control problem.
`Forecaster`	`predictDemand` `predictPrice`	Returns water demand forecasts. Returns energy price forecasts.
`Engine`	`factorStep`	Precomputes certain quantities that facilitate and accelerate the computation of the dual gradient in the algorithm.

are performed using the `cublasSaxpy` of cuBLAS. Apart from the standard cuBLAS methods, we have defined custom kernels for the summation over the set of nodes and to evaluate projections with respect to the box constraints and the proximal operator of the distance function from the set that is, to compute the proximal operator of g as discussed in Section 4.3.

5.3. *Software verification*

The validation of the software is done through *unit testing.* A unit represents the smallest functional part of the software and unit testing involves verification of its functionality through predefined inputs and expected outputs. It assists in the debugging and maintenance of the code and facilitates the integration of the various units reliably. In lack of a standardized testing framework for CUDA applications, we developed our in-house testing framework. Moreover, using `cudaMemCheck` we have thoroughly tested for GPU-side memory leaks.

6. RapidNet in action: simulation results

In this section, we present the application of `RapidNet` for the management of the drinking water network of the city of Barcelona, whose schematic diagram is shown in Fig. 6, using the demand data provided in (Grosso et al., 2014b; Sampathirao et al., 2017a). The network counts a total of 63 tanks, 114 controlled flows (by means of 75 pumps and 39 valves), 88 demand sectors and 17 mixing nodes.

6.1. *Forecasting of water demands*

Upcoming water demands are predicted using a radial-basis-function support vector machines (SVM) model from the literature with good predictive ability (Sampathirao et al., 2014). The model predicts the water demand using past demand data together with calendar data (day of the week). Validation information is provided in (Sampathirao et al., 2014). In Fig. 7 we show an instance of a prediction using this SVM model.

6.2. *Forecasting of energy prices*

Among other European countries such as Denmark and Sweden, Austria implements a deregulated energy market which induces a volatile stream of electricity prices. Time series of energy prices in

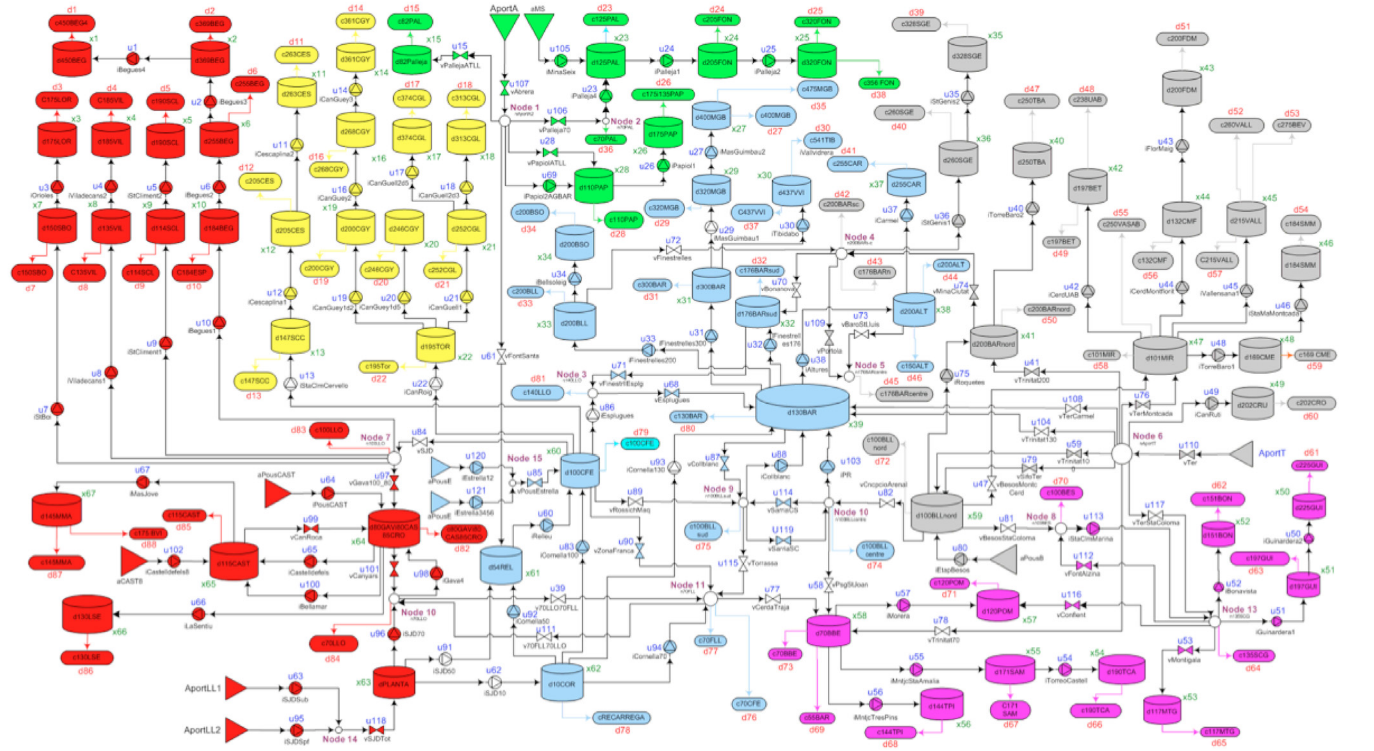


Fig. 6. Schematic diagram of the drinking water network of Barcelona. Overall, this distribution network involves 63 tanks, 114 controlled flows and, in particular, 75 pumps and 39 valves, 88 demand sectors and 17 mixing nodes.

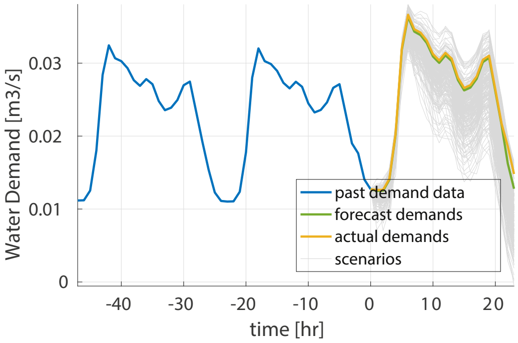


Fig. 7. Water demands predicted with a radial-basis-function SVM model. The current time instant is positioned at 0.

Austria have become public by EXAA (Energy Exchange Austria; a central European energy exchange) and are available online at http://www.exaa.at/de/marktdaten/historische-daten (Accessed on November 29, 2017). In lack of electricity price data from Spain or relevant data from the water network of Barcelona, we used price data from Austria as an indicative dataset.

Out of 8784 hourly price data which are available for the year 2016, we excluded the last 2000 data points to be used for testing and using the rest of the data we built an ARIMA$(24, 1, 4)$ model. ARIMA models have been previously used for the short-term prediction of electricity prices in the day-ahead market (Weron, 2014). Using a Monte-Carlo method, a set of 10^4 independent scenarios were generated and, subsequently, these were reduced into a scenario tree using the method described in (Heitsch and Römisch, 2009). An instance of a prediction using the trained ARIMA model along with the associated scenario tree is shown in Fig. 8.

The residuals of the model where found to be uncorrelated at the confidence level of 99.9%. Indeed, the residuals pass the Ljung-Box Q-test of uncorrelatedness with p-value equal to 1.0000. The model was selected using the Akaike information criterion (AIC) with value 1.523.

It is natural to expect that the upcoming energy prices can be only predicted up to moderate accuracy as they do not follow a regular pattern and are influenced by many market-related parameters. Despite our limited predictive capacity, we shall show in Section 6.3 that by taking into account this volatility using stochastic model predictive control we do mitigate the effect of the price uncertainty leading to a more economic operation of the water network.

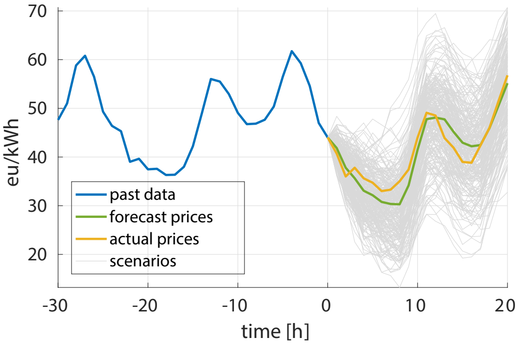


Fig. 8. Past data of energy prices (blue line) together with the nominal 24-h-ahead forecast produced by the ARIMA model (green line) and the actual upcoming price data (orange line). A set of 250 scenarios in also shown with thin gray lines. The current time instant is positioned at 0. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

6.3. *Closed-loop simulations*

In this section we present closed-loop simulation results on the water network where the sampling time is equal to 1 hr and the prediction horizon of the SSMPC was fixed to $H_p = 24$. The weight parameters used to tune the SSMPC are $W_\alpha = 10^6$, $W_u = 1.3 \cdot 10^7 \cdot I$, $W_s = 10^5$ and $W_x = 10^8$. Note that all units used in this section are SI units (flows in $m^3/_s$ and volumes in m^3).

The system was simulated for a period of $H_s = 168$ time instants, which corresponds to one week of operation. In order to assess the performance of the closed-loop operation, we use three KPIs, in particular: (i) the *economic index* which is defined as

$$\mathrm{KPI}_E = {}^1/_{H_s} \sum_{k=1}^{H_s} (\alpha_0 + \alpha_k)' u_k, \tag{37}$$

which provides an estimation of the average hourly cost of operation of the water network, (ii) the *safety index*, which is defined as

$$\mathrm{KPI}_S = \sum_{k=1}^{H_s} ||\max\{x_s - x_k, 0\}||_1, \tag{38}$$

which quantifies the total weekly violation of the safety constraint (the “soft” requirement that $x_k \geq x_s$), and, last, (iii) the *complexity index* which is the worst-case time required to solve the SSMPC optimization problem up to the specified accuracy, which is defined as

$$\mathrm{KPI}_\tau = \max_{k=1,\ldots,H_s} \tau_k, \tag{39}$$

where τ_k is the computation time required by the solver for solving the SSMPC problem at time instant k. The maximum runtime is of higher importance that the average runtime in applications to verify that a decision can be made within the available time period. All three indices are defined so that low values are preferred.

In order to justify the need for and underline the importance of taking into account the uncertainty associated with electricity prices, we performed two sets of simulations where in one case we disregarded that volatility (the SSMPC used only the nominal predictions of the electricity prices). We may observe in Fig. 9 that the hourly operation of the network becomes more expensive by approximately +5%, therefore, the proposed stochastic control approach can lead to significant economic savings for the network operator. For example, for the case of 631 scenarios, the operation cost when the price uncertainty is disregarded is €4947/hour, whereas with the proposed approach which takes into account the price uncertainty it drops to €4748/hour — a saving of €199/hour, that is, a cost reduction of 4%.

We may also see that as more scenarios are considered in the SSMPC formulation, the average cost of operation plummets at around 194 scenarios, where, however, the safety index is still high. In order to obtain a safer operation we need to pay the “cost of safe operation”: Indeed, at 631 scenarios, the operation of the network is more expensive compared to the case of 194 scenarios, but KPI_S is at a minimum. This reflects a trade-off between the economic and safe operation of the network.

At 631 scenarios, the SSMPC optimization problem involves as many as 2,306,133 primal variables and 3,126,960 dual variables. `RapidNet` solves it with $\mathrm{KPI}_\tau = 82.7$s as compared to $\mathrm{KPI}_\tau = 719$s for the popular commercial solver Gurobi v7.0 at the same level of

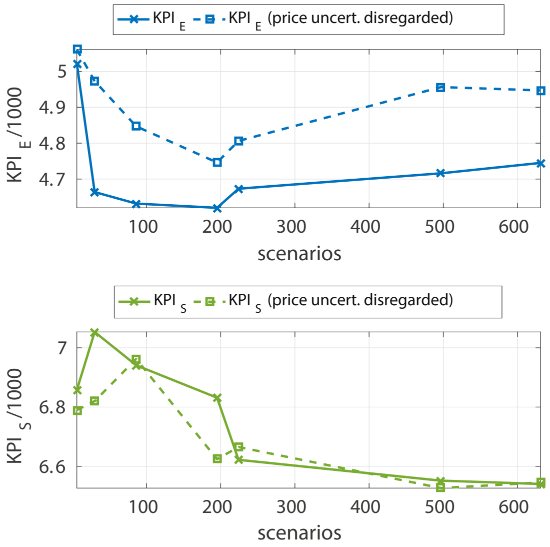


Fig. 9. (Up) The economic index KPI$_E$ in euros (€) as a function of the number of scenarios considered in the SSMPC formulation, (Down) The safety index KPI$_S$. The dashed lines correspond to SSMPC's which do not consider the volatility in electricity prices.

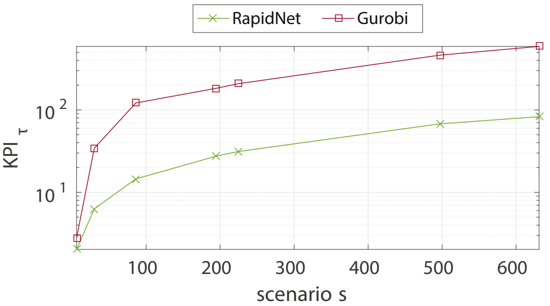


Fig. 10. KPI$_\tau$ for Gurobi (executed on an Intel Core i7-6600U machine with 4 × 2.60 GHz CPUs, 12 GB RAM running 64-bit Ubuntu 16.04) and RapidNet running on an NVIDIA Tesla C2075 GPU.

accuracy (tolerance $5 \cdot 10^{-2}$). The solvers CPLEX and MOSEK were also tested, but Gurobi outperformed them consistently. As we may see in Fig. 10, RapidNet exhibits a lower complexity index by approximately an order of magnitude.

7. Conclusions and future work

We presented the integrated software solution RapidNet for the control of drinking water networks which accounts for uncertainty both in predicted water demand and in predicted electricity prices in the day-ahead energy market.

RapidNet is a highly inter-operable software as it can be interfaced using JSON files which follow a standard API which is detailed in the software documentation. It can be combined with any forecaster as the controller does not need to know the mechanism with which the forecasts are produced. RapidNet features a parser implemented in MATLAB which allows the conversion of EPANET.inp files to JSON files. It is an open-source and free software which is distributed under the terms of the GNU LGPL v3.0 licence and can be downloaded at https://github.com/GPUEngineering/RapidNet.

In this paper we have shown via simulations that when a water network is operated in the context of a volatile energy market, considerable savings can be obtained by using a predictor of the upcoming energy prices and taking into account the associated price-related uncertainty in the formulation of the scenario tree. Alongside, we advocate that RapidNet can transform SSMPC from a powerful (but too complex) theoretical development to control engineering practice and enabling the solution of very large SSMPC scale problems and their seamless integration into the control system of the water network.

This work focuses on flow-based water distribution networks such as the network of the city of Barcelona. Pressure-based networks lead to optimization problems with nonconvex constraints. These can be approximated by solving a constraints satisfaction problem (CSP) as discussed in (Cong Cong et al., 2014). The problem can be then transformed into a convex SSMPC problem which can be solved by RapidNet. The operator splitting concept has been proven to apply to sums of nonconvex functions which allows the solution of nonconvex optimal control problems with input constraints (Themelis et al. 2016b). This allows the application of recent developments in nonconvex optimization such as PANOC (Stella et al., 2017) for the solution of nonlinear MPC problems.

Future developments in RapidNet will involve the implementation of new faster parallelizable algorithms which make use of quasi-Newton directions based on our recent theoretical work (Sampathirao et al., 2017b) and the exploitation of multiple-GPU architectures. We shall introduce a library of available water network entities from the literature as well as reported water demand and energy price models for different water networks and energy markets.

References

Bakker, M., Vreeburg, J.H.G., Palmen, L.J., Sperber, V., Bakker, G., Rietveld, L.C., 2013. Better water quality and higher energy efficiency by using model predictive flow control at water supply systems. J. Water Supply Res. Technol. - Aqua 62 (1), 1–13. https://doi.org/10.2166/aqua.2013.063.

Bakker, M., van Duist, H., van Schagen, K., Vreeburg, J., Rietveld, L., 2014. Improving the performance of water demand forecasting models by using weather input. Procedia Eng. 70, 93–102. https://doi.org/10.1016/j.proeng.2014.02.012.

Batchabani, E., Fuamba, M., 2014. Optimal tank design in water distribution networks: review of literature and perspectives. J. Water Resour. Plan. Manag. 140 (2), 136–145. https://doi.org/10.1061/(asce)wr.1943-5452.0000256.

Bauschke, H.H., Combettes, P.L., 2011. Convex Analysis and Monotone Operator Theory in Hilbert Spaces. Springer. https://doi.org/10.1007/978-1-4419-9467-7.

Beraldi, P., Simone, F.D., Violi, A., 2010. Generating scenario trees: a parallel integrated simulation-optimization approach. J. Comput. Appl. Math. 233 (9), 2322–2331. https://doi.org/10.1016/j.cam.2009.10.017.

Bertsekas, D.P., 2000. Dynamic Programming and Optimal Control, second ed. Athena Scientific.

Bertsekas, D.P., Shreve, S.E., 1996. Stochastic Optimal Control: the Discrete-time Case. Athena Scientific.

Calafiore, G., Campi, M., 2006. The scenario approach to robust control design. Automatic Control, IEEE Trans. 51 (5), 742–753. https://doi.org/10.1109/TAC.2006.875041.

Carpentier, P.-L., Barty, K., Girardeau, P., 2010. Decomposition of large-scale stochastic optimal control problems. RAIRO - Operations Res. 44, 167–183. https://doi.org/10.1051/ro/2010013.

Carpentier, P.-L., Gendreau, M., Bastin, F., 2013. Optimal scenario set partitioning for multistage stochastic programming with the progressive hedging algorithm. Tech. Rep. http://www.optimization-online.org/DB_HTML/2013/10/4065.html.

Cembrano, G., Quevedo, J., Puig, V., Pérez, R., Figueras, J., Verdejo, J.M., Escaler, I., Ramón, G., Barnet, G., Rodríguez, P., Casas, M., 2011. PLIO: a generic tool for real-time operational predictive optimal control of water networks. Water Sci. Technol. 64 (2), 448–459. https://doi.org/10.2166/wst.2011.431.

Chambolle, A., Pock, T., 2011. A first-order primal-dual algorithm for convex problems with applications to imaging. J. Math. Imag. Vis. 40 (1), 120–145. https://doi.org/10.1007/s10851-010-0251-1.

Chen, Z., Xu, D., 2014. Knowledge-based scenario tree generation methods and application in multiperiod portfolio selection problem. Appl. Stoch. Model. Bus. Ind 30 (3), 240–257. https://doi.org/10.1002/asmb.1970.

Chi-Tsong, C., 1999. Linear System Theory and Design, third ed. Saunders College Publishing, Philadelphia, PA, USA.

Chiang, N., Petra, C.G., Zavala, V.M., 2014. Structured nonconvex optimization of large-scale energy systems using PIPS-NLP. In: Power Systems Computation Conference (PSCC), 2014, pp. 1–7. https://doi.org/10.1109/PSCC.2014.7038374.

Cong Cong, S., Puig, S., Cembrano, G., 2014. Combining CSP and MPC for the operational control of water networks: application to the Richmond case study. In: 19th IFAC World Congress, Cape Town, pp. 6246–6251. https://doi.org/10.1016/j.engappai.2015.12.006.

Defourny, B., Ernst, D., Wehenkel, L., 2012. Multistage stochastic programming: a scenario tree based approach to planning under uncertainty. In: Decision Theory Models for Applications in Artificial Intelligence, IGI Global, pp. 97–143. http://hdl.handle.net/2268/80246.

Eck, B., McKenna, S., Akrhiev, A., Kishimoto, A., Palmes, P., Taheri, N., van den Heever, S., 2014. Pump scheduling for uncertain electricity prices. In: World Environmental and Water Resources Congress 2014. American Society of Civil Engineers. https://doi.org/10.1061/9780784413548.046.

Eggimann, S., Mutzner, L., Wani, O., Schneider, M.Y., Spuhler, D., Moy de Vitry, M., Beutler, P., Maurer, M., 2017. The potential of knowing more: a review of data-driven urban water management. Environ. Sci. Technol. 51 (5), 2538–2553. https://doi.org/10.1021/acs.est.6b04267 pMID: 28125222.

Ghiassi, M., Zimbra, D.K., Saidane, H., 2008. Urban water demand forecasting with a dynamic artificial neural network model. J. Water Resour. Plan. Manag. 134 (2), 138–146. https://doi.org/10.1061/(asce)0733-9496(2008)134:2(138).

Glpnar, N., Rustem, B., Settergren, R., 2004. Simulation and optimization approaches to scenario tree generation. J. Econ. Dyn. Control 28 (7), 1291–1315. https://doi.org/10.1016/S0165-1889(03)00113-1.

Goryashko, A., Nemirovski, A., 2014. Robust energy cost optimization of water distribution system with uncertain demand. Automation Remote Control 75 (10), 1754–1769. https://doi.org/10.1134/S000511791410004X.

Grosso, J.M., Maestre, J.M., Ocampo-Martinez, C., Puig, V., 2014. On the assessment of tree-based and chance-constrained predictive control approaches applied to drinking water networks. IFAC Proc. Vol. 47 (3), 6240–6245. https://doi.org/10.3182/20140824-6-ZA-1003.01648.

Grosso, J., Ocampo-Martinez, C., Puig, V., Joseph, B., 2014. Chance-constrained model predictive control for drinking water networks. J. Process Control 24 (5), 504–516. https://doi.org/10.1016/j.jprocont.2014.01.010.

Grosso, J.M., Velarde, P., Ocampo-Martinez, C., Maestre, J.M., Puig, V., 2016. Stochastic model predictive control approaches applied to drinking water networks. Optim. Control Appl. Methods 1–18. https://doi.org/10.1002/oca.2269.

Guidolin, M., Chen, A.S., Ghimire, B., Keedwell, E.C., Djordjevic, S., Savic, D.A., 2016. A weighted cellular automata 2D inundation model for rapid flood analysis. Environ. Model. Softw. 84, 378–394. https://doi.org/10.1016/j.envsoft.2016.07.008.

Hans, C., Sopasakis, P., Bemporad, A., Raisch, J., Reincke-Collon, C., 2015. Scenario-based model predictive operation control of islanded microgrids. In: 54 IEEE Conf. Decision and Control, Osaka, Japan, pp. 3272–3277. https://doi.org/10.1109/CDC.2015.7402711.

Heitsch, H., Römisch, W., 2009. Scenario tree modeling for multistage stochastic programs. Math. Program. 118 (2), 371–406. https://doi.org/10.1007/s10107-007-0197-2.

Horne, A., Szemis, J.M., Kaur, S., Webb, J.A., Stewardson, M.J., Costa, A., Boland, N., 2016. Optimization tools for environmental water decisions: a review of strengths, weaknesses, and opportunities to improve adoption. Environ. Model. Softw. 84, 326–338. https://doi.org/10.1016/j.envsoft.2016.06.028.

Hübner, J., Schmidt, M., Steinbach, M.C., 2017. A distributed interior-point KKT solver for multistage stochastic optimization. Optimization Online. http://www.optimization-online.org/DB_HTML/2016/02/5318.html.

Hutton, C.J., Kapelan, Z., 2015. A probabilistic methodology for quantifying, diagnosing and reducing model structural and predictive errors in short term water demand forecasting. Environ. Model. Softw. 66, 87–97. https://doi.org/10.1016/j.envsoft.2014.12.021.

Jiang, D.R., Pham, T.V., Powell, W.B., Salas, D.F., Scott, W.R., 2014. A comparison of approximate dynamic programming techniques on benchmark energy storage problems: does anything work?. In: 2014 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL), pp. 1–8. https://doi.org/10.1109/ADPRL.2014.7010626.

Kang, J., Cao, Y., Word, D.P., Laird, C., 2014. An interior-point method for efficient solution of block-structured NLP problems using an implicit Schur-complement decomposition. Comput. Chem. Eng. 71, 563–573. https://doi.org/10.1016/j.compchemeng.2014.09.013.

Kang, J., Raghunathan, A.U., Di Cairano, S., 2015. Decomposition via ADMM for scenario-based model predictive control. In: 2015 American Control Conference (ACC), pp. 1246–1251. https://doi.org/10.1109/ACC.2015.7170904.

Klintberg, E., Gros, S., 2017. A parallelizable interior point method for two-stage robust MPC. In: IEEE Transactions on Control Systems Technology, vol. 99, pp. 1–11. https://doi.org/10.1109/TCST.2016.2638680.

Latorre, J.M., Cerisola, S., Ramos, A., 2007. Clustering algorithms for scenario tree generation: application to natural hydro inflows. Eur. J. Operational Res. 181 (3), 1339–1353. https://doi.org/10.1016/j.ejor.2005.11.045.

Le, P.V., Kumar, P., Valocchi, A.J., Dang, H.-V., 2015. GPU-based high-performance computing for integrated surface — sub-surface flow modeling. Environ. Model. Softw. 73, 1–13. https://doi.org/10.1016/j.envsoft.2015.07.015.

Lertpalangsunti, N., Chan, C.W., Mason, R., Tontiwachwuthikul, P., 1999. A toolset for construction of hybrid intelligent forecasting systems: application for water demand prediction. Artif. Intell. Eng. 13 (1), 21–42. https://doi.org/10.1016/S0954-1810(98)00008-9.

Lobbrecht, A.H., Solomatine, D.P., 2002. Machine learning in real-time control of water systems. Urban Water 4 (3), 283–289. https://doi.org/10.1016/S1462-0758(02)00023-7.

Lubin, M., Petra, C.G., Anitescu, M., Zavala, V., 2011. Scalable stochastic optimization of complex energy systems. In: Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis, SC '11, vols. 1–64. ACM, New York, NY, USA, p. 64. https://doi.org/10.1145/2063384.2063470.

Mala-Jetmarova, H., Sultanova, N., Savic, D., 2017. Lost in optimisation of water distribution systems? a literature review of system operation. Environ. Model. Softw. 93, 209–254. https://doi.org/10.1016/j.envsoft.2017.02.009.

Mesbah, A., 2016. Stochastic model predictive control: an overview and perspectives for future research. IEEE Control Syst. 36 (6), 30–44. https://doi.org/10.1109/MCS.2016.2602087.

Meseguer, J., Quevedo, J., 2017. Real-time monitoring and control in water systems. In: Puig, V., Ocampo-Martínez, C., Pérez, R., Cembrano, G., Quevedo, J., Escobet, T. (Eds.), Real-time Monitoring and Operational Control of Drinking-water Systems. Springer, Cham, pp. 1–19. https://doi.org/10.1007/978-3-319-50751-4_1.

Msiza, I.S., Nelwamondo, F.V., Marwala, T., 2007. Artificial neural networks and support vector machines for water demand time series forecasting. In: 2007 IEEE International Conference on Systems, Man and Cybernetics, pp. 638–643. https://doi.org/10.1109/ICSMC.2007.4413591.

Ocampo-Martinez, C., Negenborn, R., 2015. Transport of Water versus Transport over Water : Exploring the Dynamic Interplay of Transport and Water. Springer, Cham. https://doi.org/10.1007/978-3-319-16133-4.

Ocampo-Martinez, C., Puig, V., Cembrano, G., Creus, R., Minoves, M., 2009. Improving water management efficiency by using optimization-based control strategies: the Barcelona case study. Water Sci. Technol. Water Supply 9 (5), 565–575. https://doi.org/10.2166/ws.2009.524.

Ocampo-Martinez, C., Fambrini, V., Barcelli, D., Puig, V., 2010. Model predictive control of drinking water networks: a hierarchical and decentralized approach. In: Proceedings of the 2010 American Control Conference, pp. 3951–3956. https://doi.org/10.1109/ACC.2010.5530643.

Papageorgiou, M., 1984. Optimal control of generalized flow networks. In: Thoft-Christensen, P. (Ed.), System Modelling and Optimization: Proceedings of the 11th IFIP Conference Copenhagen, Denmark, July 25–29, 1983. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 373–382. https://doi.org/10.1007/BFb0008911.

Papageorgiou, E.I., Poczęta, K., Laspidou, C., 2016. Hybrid model for water demand prediction based on fuzzy cognitive maps and artificial neural networks. In: 2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp. 1523–1530. https://doi.org/10.1109/FUZZ-IEEE.2016.7737871.

Parikh, N., Boyd, S., 2014. Proximal algorithms. Found. Trends Optim. 1 (3), 127–239. https://doi.org/10.1561/2400000003.

Patrinos, P., Bemporad, A., 2014. An accelerated dual gradient-projection algorithm for embedded linear model predictive control. IEEE Trans. Automatic Control 59 (1), 18–33. https://doi.org/10.1109/TAC.2013.2275667.

Pedregal, D.J., Trapero, J.R., 2007. Electricity prices forecasting by automatic dynamic harmonic regression models. Energy Convers. Manag. 48 (5), 1710–1719. https://doi.org/10.1016/j.enconman.2006.11.004.

Peña-Guzmán, C., Melgarejo, J., Prats, D., 2016. Forecasting water demand in residential, commercial, and industrial zones in Bogotá, Colombia, using least-squares support vector machines. Math. Problems Eng. 2016, 1–10. https://doi.org/10.1155/2016/5712347.

Pflug, G.C., Pichler, A., 2015. Dynamic generation of scenario trees. Comput. Optim. Appl. 62 (3), 641–668. https://doi.org/10.1007/s10589-015-9758-0.

Puglia, L., Patrinos, P., Bernardini, D., Bemporad, A., 2013. Reliability and efficiency for market parties in power systems. In: 2013 10th International Conference on the European Energy Market (EEM), pp. 1–8. https://doi.org/10.1109/EEM.2013.6607416.

Rockafellar, R., Wets, J., 2009. Variational Analysis, third ed. Springer-Verlag, Berlin. https://doi.org/10.1007/978-3-642-02431-3.

Romano, M., Kapelan, Z., 2014. Adaptive water demand forecasting for near real-time management of smart water distribution systems. Environ. Model. Softw. 60, 265–276. https://doi.org/10.1016/j.envsoft.2014.06.016.

Sampathirao, A.K., Grosso, J.M., Sopasakis, P., Ocampo-Martinez, C., Bemporad, A., Puig, V., 2014. Water demand forecasting for the optimal operation of large-scale drinking water networks: the Barcelona case study. In: 19th IFAC World Congress, Cape Town, South Africa, pp. 10457–10462. https://doi.org/10.3182/20140824-6-ZA-1003.01343.

Sampathirao, A.K., Sopasakis, P., Bemporad, A., Patrinos, P., 2015. Distributed solution of stochastic optimal control problems on GPUs. In: 54 IEEE Conf. Decision and Control, Osaka, Japan, pp. 7183–7188. https://doi.org/10.1109/CDC.2015.7403352.

Sampathirao, A.K., Sopasakis, P., Bemporad, A., Patrinos, P., 2016. Fast parallelizable scenario-based stochastic optimization. In: 4th European Conference on Computational Optimization, Leuven, Belgium.

Sampathirao, A.K., Sopasakis, P., Bemporad, A., Patrinos, P., 2017. GPU-accelerated stochastic predictive control of drinking water networks. IEEE Trans. Control Syst. Technol. 99, 1–12. https://doi.org/10.1109/TCST.2017.2677741.

Sampathirao, A.K., Sopasakis, P., Bemporad, A., Patrinos, P., 2017. Proximal quasi-Newton methods for scenario-based stochastic optimal control. IFAC WC 2017.

Shapiro, A., Dentcheva, D., Ruszczynski, A., 2009. Lectures on Stochastic Programming: Modeling and Theory, Society for Industrial and Applied Mathematics, Philadelphia. https://doi.org/10.1137/1.9780898718751.

Solomatine, D.P., 2003. Applications of data-driven modelling and machine learning in control of water resources. In: Computational Intelligence in Control, IGI

Global, pp. 197–217.

Stella, L., Themelis, A., Sopasakis, P., Patrinos, P., 2017. A simple and efficient algorithm for nonlinear model predictive control. In: IEEE CDC. Melbourne, Australia.

Themelis, A., Villa, S., Patrinos, P., Bemporad, A., 2016. Stochastic gradient methods for stochastic model predictive control. In: 2016 European Control Conference (ECC), pp. 154–159. https://doi.org/10.1109/ECC.2016.7810279.

Themelis, A., Stella, L., Patrinos, P., 2016. Forward-backward envelope for the sum of two nonconvex functions: Further properties and nonmonotone line-search algorithms. ArXiv preprint arXiv:1606.06256.

Tran, V., Brdys, M., 2009. Optimizing control by robustly feasible model predictive control and application to drinking water distribution systems. In: Alippi, C., Polycarpou, M., Panayiotou, C., Ellinas, G. (Eds.), Artificial Neural Networks ICANN 2009, Vol. 5769 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 823–834. https://doi.org/10.1007/978-3-642-04277-5_83.

Tseng, P., 1990. Further applications of a splitting algorithm to decomposition in variational inequalities and convex programming. Math. Program. 48 (1), 249–263. https://doi.org/10.1007/BF01582258.

Tseng, P., 1991. Applications of a splitting algorithm to decomposition in convex programming and variational inequalities. SIAM J. Control Optim. 29 (1), 119–138. https://doi.org/10.1137/0329006.

Tseng, P., 2000. A modified forward-backward splitting method for maximal monotone mappings. SIAM J. Control Optim. 38 (2), 431–446. https://doi.org/10.1137/s0363012998338806.

Wang, Y., Ocampo-Martinez, C., Puig, V., 2015. Robust model predictive control based on Gaussian processes: application to drinking water networks. In: 2015 European Control Conference (ECC), pp. 3292–3297. https://doi.org/10.1109/ECC.2015.7331042.

Wang, Y., Ocampo-Martinez, C., Puig, V., 2016. Stochastic model predictive control based on Gaussian processes applied to drinking water networks. IET Control Theory Appl. 10 (8), 947–955. https://doi.org/10.1049/iet-cta.2015.0657.

Wang, Y., Ocampo-Martínez, C., Puig, V., Quevedo, J., 2016. Gaussian-process-based demand forecasting for predictive control of drinking water networks. In: Panayiotou, C.G., Ellinas, G., Kyriakides, E., Polycarpou, M.M. (Eds.), Critical Information Infrastructures Security: 9th International Conference, CRITIS 2014, Limassol, Cyprus, October 13-15, 2014, Revised Selected Papers. Springer International Publishing, Cham, pp. 69–80. https://doi.org/10.1007/978-3-319-31664-2_8.

Weron, R., 2014. Electricity price forecasting: a review of the state-of-the-art with a look into the future. Int. J. Forecast. 30 (4), 1030–1081. https://doi.org/10.1016/j.ijforecast.2014.08.008.

Wytock, M., 2016. Optimizing Optimization: Scalable Convex Programming with Proximal Operators. Ph.D. thesis. Carnegie Mellon University, Pittsburgh. http://repository.cmu.edu/dissertations/785.

Xia, X., Liang, Q., 2016. A GPU-accelerated smoothed particle hydrodynamics (SPH) model for the shallow water equations. Environ. Model. Softw. 75, 28–43. https://doi.org/10.1016/j.envsoft.2015.10.002.

Yasar, A., Bilgili, M., Simsek, E., 2012. Water demand forecasting based on stepwise multiple nonlinear regression analysis. Arabian J. Sci. Eng. 37 (8), 2333–2341. https://doi.org/10.1007/s13369-012-0309-z.