Geometric Programming-Based Control for Nonlinear, DAE-Constrained Water Distribution Networks

Shen Wang*, Ahmad F. Taha*, Nikolaos Gatsis*, and Marcio Giacomoni†

Abstract—Control of water distribution networks (WDNs) can be represented by an optimization problem with hydraulic models describing the nonlinear relationship between head loss, water flow, and demand. The problem is difficult to solve due to the non-convexity in the equations governing water flow. Previous methods used to obtain WDN controls (i.e., operational schedules for pumps and valves) have adopted simplified hydraulic models. One common assumption found in the literature is the modification of WDN topology to exclude loops and assume a known water flow direction. In this paper, we present a new geometric programming-based model predictive control approach, designed to solve the water flow equations and obtain WDN controls. The paper considers the nonlinear difference algebraic equation (DAE) form of the WDN dynamics, and the GP approach amounts to solving a series of convex optimization problems and requires neither the knowledge of water flow direction nor does it restrict the water network topology. A case study is presented to illustrate the performance of the proposed method.

I. INTRODUCTION AND PAPER CONTRIBUTIONS

Water distribution networks (WDNs) are crucial infrastructures in urban areas. With the expansion of cities, the complexity of WDNs poses challenges for utilities taking into account multiple—potentially conflicting—objectives such as minimizing economic costs, guaranteeing the stability of the network, and maintaining safe water levels in tanks. The very basic decision-making problem involved in the majority of WDN operation problems involves solving for the *water flow* and *head* in pipes given water demand forecasts. The hydraulic models of head loss and water flow across pipes, valves, and pumps are accurate, yet highly nonlinear and complex.This subsequently hinders optimally solving management/operation problems incorporating the WFP.

The literature of solving the nonconvex WFP is indeed rich and elaborate [1]–[4]; a discussion on the merits of these methods is outside the scope of this paper. Most of these methods are iterative algorithms developed to solve a set of linear and nonlinear equations to obtain the physical status (flows and heads) of WDNs. In their recent paper [5], the authors investigate the uniqueness of the WFP's solution for generic WDNs.

Several methods have been developed to solve operational and pump/valve scheduling problems incorporating the WFP in WDNs, and have been recently surveyed in [6] in great detail. Model predictive control formulations have been reported in [7], [8]. Specifically, the authors in [9] present a stochastic MPC formulation to handle uncertainty in WDNs and apply the proposed MPC to the Barcelona drinking water network

with real demands. The authors in [7] address a nonlinear economic MPC strategy to minimize the costs associated with pumping and water treatment. A nonlinear MPC controller is designed in [8] to meet consumer demands at desired pressures.

The nonlinearity present in WDNs forms a set of nonlinear difference algebraic equations (DAE) which are difficult to handle when solving operation problems. Some recent methods to deal with the nonlinearity are: linearizing the objectives and constraints [10], constructing relaxations for the nonlinear relationships [11], [12], keeping the nonlinearity and formulating the problem as a nonconvex program [13], applying convex approximations to convert the nonconvex problem into a convex one [14], [15].

The two works closest to our paper are [16] and [14]. The authors in [16] model WDNs as a directed graph, assume the directions of water flow in pipes do not change, and choose the Darcy-Weisbach equation to model head loss in pipes. In [14], the authors convert the nonconvex head loss equations into convex models using geometric programming (GP) approximations, and hence a globally optimal solution is guaranteed. An important contribution of [14] is that the proposed GP method is non-iterative. However, it only works under the two assumptions of a tree network topology and known/unchanged flow direction.

The objective of this paper is to investigate convex optimization-based methods to solve the nonconvex WFP, and to subsequently solve an MPC formulation to manage WDN controllers without restricting the WDN graph topology or assuming knowledge of the water flow direction. The proposed approach amounts to solving a series of convex optimization problems, namely, geometric programs, and is embedded within the MPC. We specifically show that the nonconvex, DAE-constrained optimal control problem of WDNs can be approximated by a convex one and solved efficiently. The organization of this paper is given next. Section II discusses control systems-oriented modeling and MPC formulation of WDNs. Section III presents the introduction to GP, and the conversion of the nonconvex hydraulic models to their corresponding convex, GP forms follows. Section IV concludes the paper with numerical tests.

II. MODELING OF WDNS

We model WDNs by a directed graph $(\mathcal{V}, \mathcal{E})$. Set $\mathcal{V}$ defines the nodes and is partitioned as $\mathcal{V} = \mathcal{J} \bigcup \mathcal{T} \bigcup \mathcal{R}$ where $\mathcal{J}$, $\mathcal{T}$, and $\mathcal{R}$ stand for the collection of junctions, tanks, and reservoirs. Let $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ be the set of links, and define the partition $\mathcal{E} = \mathcal{P} \bigcup \mathcal{M} \bigcup \mathcal{W}$, where $\mathcal{P}$, $\mathcal{M}$, and $\mathcal{W}$ stand for the collection of pipes, pumps, and valves. For the i-th

*Department of Electrical and Computer Engineering, †Department of Civil and Environmental Engineering, The University of Texas at San Antonio, TX 78249. Emails: mvy292@my.utsa.edu, {ahmad.taha, nikolaos.gatsis, marcio.giacomoni}@utsa.edu.

	Original Hydraulic Model	DAEs	GP Form	Abstracted GP
Tanks	$h_i^{\mathrm{TK}}(k+1)=h_i^{\mathrm{TK}}(k)+\dfrac{\Delta t}{A_i^{\mathrm{TK}}}\left(\sum_{j=1}^{|\mathcal{N}_i^{\mathrm{in}}|}q_{ji}(k)-\sum_{j=1}^{|\mathcal{N}_i^{\mathrm{out}}|}q_{ij}(k)\right)$ (1)	(12a)	$\hat{h}_i(k)\left(\prod_{j=1}^{\mathcal{N}_i^{\mathrm{in}}}\hat{q}_{ji}(k)\prod_{j=1}^{|\mathcal{N}_i^{\mathrm{out}}|}\hat{q}_{ij}^{-1}(k)\right)^{\frac{\Delta t}{A_i^{\mathrm{TK}}}}\hat{h}_i^{-1}(k+1)=1$ (2)	(18a)
Junction nodes	$\sum_{j=1}^{|\mathcal{N}_i^{\mathrm{in}}|}q_{ji}(k)-\sum_{j=1}^{|\mathcal{N}_i^{\mathrm{out}}|}q_{ij}(k)=d_i(k)$ (3)	(12b)	$\prod_{j=1}^{|\mathcal{N}_i^{\mathrm{in}}|}\hat{q}_{ji}(k)\prod_{j=1}^{|\mathcal{N}_i^{\mathrm{out}}|}\hat{q}_{ij}^{-1}(k)\hat{d}_i^{-1}(k)=1$ (4)	(18b)
Pipes	$h_{ij}^{\mathrm{P}}(k)=h_i(k)-h_j(k)=Rq_{ij}(k)|q_{ij}(k)|^{\mu-1}$ (5)	(12c)	$\hat{h}_i(k)\hat{h}_i^{-1}(k)[C^{\mathrm{P}}(k)]^{-1}\hat{q}_{ij}^{-1}(k)=1$ (6)	(18c)
Pumps	$h_{ij}^{\mathrm{M}}(k)=h_i(k)-h_j(k)=-s_{ij}^2(k)(h_0-r(q_{ij}s_{ij}^{-1})^\nu)$ (7)		$\hat{h}_i(k)\hat{h}_i^{-1}(k)\hat{s}_{ij}(k)^{-C_1^{\mathrm{M}}(k)}\hat{q}_{ij}(k)^{-C_2^{\mathrm{M}}(k)}=1$ (8)	(18d)
Constraints	$h_i^{\min}\leq h_i(k)\leq h_i^{\max}$ (9a) $h_i^{\mathrm{R}}(k)=h_i^{\mathrm{R}}$ (9b) $0\leq s_{ij}(k)\leq 1$ (9c) $q_{ij}^{\min}\leq q_{ij}(k)\leq q_{ij}^{\max}$ (9d)	(13)	$\hat{h}_i^{-1}(k)\hat{h}_i^{\min}\leq 1,\hat{h}_i(k)\left(\hat{h}_i^{\max}\right)^{-1}\leq 1$ (10a) $\hat{h}_i^{-1}(k)\hat{h}_i^{\mathrm{R}}=1$ (10b) $\hat{s}_{ij}^{-1}(k)\leq 1,\hat{s}_{ij}(k)b^{-1}\leq 1$ (10c) $\hat{q}_{ij}^{-1}(k)\hat{q}_{ij}^{\min}\leq 1,\hat{q}_{ij}(k)\left(\hat{q}_{ij}^{\max}\right)^{-1}\leq 1$ (10d)	(19)

node, set $\mathcal{N}_i$ collects its neighboring nodes and is partitioned as $\mathcal{N}_i=\mathcal{N}_i^{\mathrm{in}}\bigcup\mathcal{N}_i^{\mathrm{out}}$, where $\mathcal{N}_i^{\mathrm{in}}$ and $\mathcal{N}_i^{\mathrm{out}}$ stand for the collection of inflow and outflow nodes. It is worth emphasizing that the assignment of direction to each link is arbitrary, as the presented optimizations yield optimal flow direction in pipes.

A. Conservation of mass and energy

The hydraulic equations describing the flow in WDNs are derived from the principles of *conservation of mass* and *energy*. According to these basic laws, the equations that model mass and energy conservation for all components (tanks, junctions, pipes, and pumps) in WDNs can be written and provided in explicit and compact matrix-vector forms in the first three columns of Table I. The last two columns of Table I are needed in the ensuing sections of the paper.

Tanks and Junction Nodes — The water volume evolution in the i-th tank at time k can be expressed by a discrete-time difference equation (11a), while the head created by the tank can be described as (11b)

$$V_i(k+1)=V_i(k)+\Delta t\left(\sum_{j=1}^{|\mathcal{N}_i^{\mathrm{in}}|}q_{ji}(k)-\sum_{j=1}^{|\mathcal{N}_i^{\mathrm{out}}|}q_{ij}(k)\right) \text{ (11a)}$$

$$h_i^{\mathrm{TK}}(k)=\frac{V_i(k)}{A_i^{\mathrm{TK}}}+E_i^{\mathrm{TK}},\ i\in\mathcal{T}, \text{ (11b)}$$

where V_i and Δt are the volume and sampling time; $q_{ji}(k)$, $i\in\mathcal{J}$, $j\in\mathcal{N}_i^{\mathrm{in}}$ stands for the inflow of j-th neighbor, while $q_{ij}(k)$, $i\in\mathcal{J}$, $j\in\mathcal{N}_i^{\mathrm{out}}$ stands for the outflow of j-th neighbor; h_i^{TK}, A_i^{TK}, and E_i^{TK} stand for the head, cross-sectional area, and elevation of the i-th tank. Combining (11a) and (11b), the head changes from time k to $k+1$ of the i-th tank can be written as (1) in Table I. Junction nodes are the points where water flow merges or splits. The expression of mass conservation of the i-th junction at time k can be written as (3) in Table I, and $d_i(k)$ stands for the end-user demand that is extracted from node i.

Head Loss in Pipes — The major head loss of a pipe from node i to j is due to friction and is determined by (5) from Table I, where h_i and h_j stand for the head of the i-th and j-th node; h_{ij}^{P} is the head loss of the pipe from i to j; R is the resistance coefficient and μ is the constant flow exponent in the corresponding formula. Table II represents the most

TABLE II
HEAD LOSS FORMULAE.*

Formula	Resistance Coefficient (R)	Flow Exponent (μ)
Hazen-Williams	$4.727L^{\mathrm{P}}C_{\mathrm{HW}}^{-1.852}(D^{\mathrm{P}})^{-4.871}$	1.852
Darcy-Weisbach	$0.0252L^{\mathrm{P}}f(\epsilon,D^{\mathrm{P}},q)(D^{\mathrm{P}})^{-5}$	2
Chezy-Manning	$4.66L^{\mathrm{P}}C_{\mathrm{CM}}^2(D^{\mathrm{P}})^{-5.33}$	2

*C_{HW}, ϵ, C_{CM} are roughness coefficients of Hazen-Williams, Darcy-Weisbach and Chezy-Manning. D^{P} (ft) is the pipe diameter, L^{P} (ft) is the pipe length. q (cfs) is the flow rate, f is friction factor (dependent on ϵ, D^{P}, and q).

common formulae used in the literature to model the resistance coefficient R. The approach presented in this paper considers any of the three formulae in Table II. The numerical tests use the Hazen-Williams formula [17], which is typically used to analyze urban water supply systems.

Head Gain in Pumps/Reservoirs — A head increase can be generated by a pump between the suction node i and the delivery node j. The pump properties decide the relationship function between pump flow and head increase [17]. Generally, it can be expressed as (7), where h_{ij}^{M} is head increase of the pump from i to j; h_0 is the shutoff head for the pump; q_{ij} is the flow through a pump; $s_{ij}\in[0,1]$ is the relative speed of the same pump; r and ν are the pump curve coefficients. We also assume that reservoirs have infinite water supply, the head of the i-th reservoir h_i^{R} is fixed, and this can be viewed as an operational constraint.

WDN Operational Constraints — The constraints (upper and lower bounds on the heads of junctions, tanks and reservoirs, pump speeds, and flows) are expressed as equations (9a)–(9d) in Table I. Note that constraint (9c) can be replaced with binary (on/off) constraints on the pump schedules if variable speed pumps are not available in the water network. In this paper, we assume that the relative speed of all pumps can be modulated in the interval $[0,1]$, rather that needing integer constraints.

B. Difference algebraic equations (DAE) form of WDNS

The WDN model in previous section can be abstract in the form of DAEs as (12). Define $\boldsymbol{x}$, $\boldsymbol{l}$, $\boldsymbol{u}$, $\boldsymbol{v}$, and $\boldsymbol{s}$ to be vectors collecting heads at tanks, heads at junctions, flows through pumps and valves, flows through pipes, and the relative speed of pumps. And the corresponding dimensions

1471

are $n_t, n_j, n_w + n_m, n_v$, and n_m. Collecting the mass and energy balance equations of tanks (1), junctions (3), pipes (5), and pumps (7), we write the following DAE model

$$DAE: \quad \boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}_u\boldsymbol{u}(k) + \boldsymbol{B}_v\boldsymbol{v}(k) \quad (12a)$$

$$\boldsymbol{0}_{n_j} = \boldsymbol{E}_u\boldsymbol{u}(k) + \boldsymbol{E}_v\boldsymbol{v}(k) + \boldsymbol{E}_d\boldsymbol{d}(k) \quad (12b)$$

$$\boldsymbol{0}_{n_w + n_m} = \boldsymbol{E}_x\boldsymbol{x}(k) + \boldsymbol{E}_l\boldsymbol{l}(k) + \boldsymbol{\Phi}(\boldsymbol{u},\boldsymbol{v},\boldsymbol{s}) \quad (12c)$$

where $\boldsymbol{A}$, $\boldsymbol{E}$, and $\boldsymbol{B}$ are constant matrices that depend on the WDN topology and the aforementioned hydraulics. The function $\boldsymbol{\Phi}(\cdot)$ collects the nonlinear components in (5) and (7). The physical constraints pertaining to (12) can be rewritten as

Constraints:

$$\boldsymbol{x}(k) \in [\boldsymbol{x}^{\min}(k), \boldsymbol{x}^{\max}(k)], \boldsymbol{l}(k) \in [\boldsymbol{l}^{\min}(k), \boldsymbol{l}^{\max}(k)]$$
$$\boldsymbol{u}(k) \in [\boldsymbol{u}^{\min}(k), \boldsymbol{u}^{\max}(k)], \boldsymbol{v}(k) \in [\boldsymbol{v}^{\min}(k), \boldsymbol{v}^{\max}(k)]$$
$$\boldsymbol{s}(k) \in [\boldsymbol{0}_{n_m}, \boldsymbol{1}_{n_m}] \, . \quad (13)$$

The next section discusses the WDN control objectives.

C. Control objectives

Multiple objectives can be applied depending on operational considerations, and can be expressed through

$$\Gamma_1(\boldsymbol{x}(k)) = \begin{cases} (\boldsymbol{x}(k) - \boldsymbol{x}^{\text{sf}})^\top (\boldsymbol{x}(k) - \boldsymbol{x}^{\text{sf}}), & \text{if } \boldsymbol{x}(k) \leq \boldsymbol{x}^{\text{sf}} \\ 0 & \text{Otherwise} \end{cases} \quad (14)$$

$$\Gamma_2(\Delta\boldsymbol{u}(k)) = \Delta\boldsymbol{u}(k)^\top \Delta\boldsymbol{u}(k) \quad (15)$$

where $\Gamma_1(\cdot)$ enforces maintaining the safety water storage in each tank decided by the operator and $\boldsymbol{x}^{\text{sf}}$ is a vector collecting the safety head levels of tanks; $\Gamma_2(\cdot)$ enforces the smoothness of control actions through $\Delta\boldsymbol{u}(k) = \boldsymbol{u}(k) - \boldsymbol{u}(k-1)$ which stands for the flow rate changes of controllable components from time $k-1$ to k. At initial time $k = t_0$, define a vector that collects all the optimization variables as

$$\boldsymbol{\xi}[t_0] \triangleq \left\{ \boldsymbol{x}(k+1), \boldsymbol{u}(k), \boldsymbol{l}(k), \boldsymbol{v}(k), \boldsymbol{s}(k) \right\}_{k=t_0}^{k=t_0+H_p},$$

where H_p is the prediction horizon of the MPC. Note that the indexing for $\boldsymbol{x}(k)$ is different in $\boldsymbol{\xi}[t_0]$ due to the fact that the initial conditions of the tanks $\boldsymbol{x}(t_0)$ is known, unlike other optimization variables such as the flow and the pump controls which we need to solve for from $k = t_0$ through $k = t_0 + H_p$. The multi-objective cost function can be written as

$$\Gamma(\boldsymbol{\xi}[t_0]) = \sum_{i=1}^{2} \Gamma_i(\boldsymbol{\xi}[t_0]) = \boldsymbol{\xi}^\top[t_0]\boldsymbol{\Omega}\boldsymbol{\xi}[t_0] + \boldsymbol{\omega}^\top\boldsymbol{\xi}[t_0] + \omega,$$

where $\boldsymbol{\Omega}, \boldsymbol{\omega}$ and ω are the corresponding weight matrix, vector, and scalars from $\Gamma_{1,2}(\boldsymbol{\xi})$. Similar objective functions have been used before in [10], [18].

D. MPC formulation

Here, we propose using MPC to solve the WDN operation problem considering the nonlinearities and nonconvexities present in the energy balance equations in the DAE model (12). At each time instant k, MPC obtains the control input to be implemented, given a prediction of the current disturbance, output, and states, and then finds an optimal

control sequence over a prediction horizon H_p. The MPC can be written as

$$\min_{\boldsymbol{\xi}[t_0]} \Gamma\left(\boldsymbol{\xi}[t_0]\,\Big|\, \boldsymbol{x}(t_0), \{\boldsymbol{d}(k)\}_{k=t_0}^{k=t_0+H_p}\right)$$

$$\text{s.t. } DAE \text{ (12)}, \text{ Constraints (13)} \quad (16)$$

The optimization problem (16) is nonlinear and nonconvex due to the head loss models of pipes and pumps. WDN-MPC requires the knowledge or prediction of the nodal water demand for the entire prediction horizon as well as the initial levels of tanks $\boldsymbol{x}(t_0)$, and $\{\boldsymbol{d}(k)\}_{k=t_0}^{k=t_0+H_p}$.

III. GP MODELING OF WDNs

A basic introduction to GP is provided first, followed by the conversion of the nonconvex hydraulic models in (16) to their corresponding convex, GP form. A geometric program (GP) is a type of optimization problem with objective and constraint functions that are monomials and posynomials [19].

Here, we propose a GP model without assuming a known flow direction by mapping the optimization variable $\boldsymbol{\xi}[t_0]$ in (16) into its exponential form. The conversion helps to map all of the non-positive values into positive ones. Specifically, we convert the head and demand at the i-th node h_i and d_i, the flow q_{ij}, and relative speed s_{ij} into positive values $\hat{h}_i$, $\hat{d}_i$, $\hat{q}_{ij}$, and $\hat{s}_{ij}$ through exponential functions, as follows

$$\hat{h}_i \triangleq b^{h_i}, \ \hat{d}_i \triangleq b^{d_i}, \ \hat{q}_{ij} \triangleq b^{q_{ij}}, \ \hat{s}_{ij} \triangleq b^{s_{ij}}, \quad (17)$$

where $b = 1 + \delta$ is a constant base and δ is a small positive number. After conversion of variables, the $\hat{h}_i$, $\hat{d}_i$, $\hat{q}_{ij}$, and $\hat{s}_{ij}$ are positive values which can then be used to transform the nonconvex problem (16) into a GP. Converting the junction and tank physical models as well as constraints follows from the above exponential mapping (17), while converting the pipe and pump models into GP form is more complicated. The last two columns of Table I produce detailed as well as abstract versions of the conversions of all physical models.

A. Conversion of mass and energy balance equations

For the models of tanks and junctions, the conversion process is straightforward. After exponentiating both sides of (1) and (3), variables q_{ij}, h_i, and d_i are changed into $\hat{q}_{ij}$, $\hat{h}_i$, and $\hat{d}_i$, while constraints (1) and (3) are converted to monomial equality constraints (2) and (4) in Table I.

In order to clearly show the derivation for pipes, the time k is ignored at first. At time k, let $\hat{h}_{ij}^{\text{P}}$ be the GP form of head loss of a pipe, which is obtained by exponentiating both sides of (5) as follows

$$\hat{h}_i\hat{h}_i^{-1} = \hat{h}_{ij}^{\text{P}} = b^{\left(q_{ij}R|q_{ij}|^{\mu-1} - q_{ij}\right)} \cdot b^{q_{ij}} = C^{\text{P}}(q_{ij}) \cdot \hat{q}_{ij},$$

where $C^{\text{P}}(q_{ij}) = b^{q_{ij}\left(R|q_{ij}|^{\mu-1} - 1\right)}$ is a function of q_{ij}.

Note that variable $q_{ij}(k)$ is unknown at each time k. The premise is to solve a series of convex optimization problems to find the final value for each time k. Hence, we can make an initial guess denoted by $\langle q_{ij} \rangle_0$ and $\langle C^{\text{P}} \rangle_0$ for iteration 0. For the n-th iteration, the corresponding values are denoted by $\langle q_{ij} \rangle_n$ and $\langle C^{\text{P}} \rangle_n$. If the flow rates are close to each other between the $(n-1)$-th and n-th iteration, we can approximate

$\langle C^{\mathrm{P}}\rangle_n$ using $\langle C^{\mathrm{P}}\rangle_{n-1}$, that is $\langle C^{\mathrm{P}}\rangle_n \approx \langle C^{\mathrm{P}}\rangle_{n-1}$. Thus, for each iteration n,

$$\langle C^{\mathrm{P}}(k)\rangle_n = b^{\langle q_{ij}(k)\rangle_{n-1}\left(R|\langle q_{ij}(k)\rangle_{n-1}|^{\mu-1}-1\right)}$$

can be approximated by a constant given the flow value $\langle q_{ij}(k)\rangle_{n-1}$ from the previous iteration. With this approximation, the head loss constraint for each pipe can be written as a *monomial equality constraint* $\hat{h}_i(k)\hat{h}_i^{-1}(k) = C^{\mathrm{P}}(k)\hat{q}_{ij}(k)$ which is equivalently expressed as (6). If we solve flow $q_{ij}(k)$, this value can be as an initialization for the next iteration, implying that $\langle C^{\mathrm{P}}(k+1)\rangle_0 = \langle C^{\mathrm{P}}(k)\rangle_n$ which can accelerate the convergence of the successive convex approximation.

Similarly for pumps, the new variables $\hat{q}_{ij}(k) = b^{q_{ij}(k)}$ and $\hat{s}_{ij}(k) = b^{s_{ij}(k)}$ for $(i,j) \in \mathcal{M}$ are introduced. Let $\hat{h}_{ij}^{\mathrm{M}}$ be the GP form of head increase of a pump:

$$\hat{h}_i\hat{h}_i^{-1} = \hat{h}_{ij}^{\mathrm{M}} = (b^{s_{ij}})^{-s_{ij}h_0} \cdot (b^{q_{ij}})^{rq_{ij}^{\nu-1}s_{ij}^{2-\nu}} = (\hat{s}_{ij})^{C_1^{\mathrm{M}}} (\hat{q}_{ij})^{C_2^{\mathrm{M}}},$$

where $C_1^{\mathrm{M}} = -s_{ij}h_0$ and $C_2^{\mathrm{M}} = rq_{ij}^{\nu-1}s_{ij}^{2-\nu}$. Parameters $C_1^{\mathrm{M}}(k)$ and $C_2^{\mathrm{M}}(k)$ follow a similar iterative process as $C^{\mathrm{P}}(k)$. That is, they are treated at iteration n as constants based on the flow and relative speed values at iteration $n-1$. Hence, the approximating equation for the pump head increase becomes the monomial equality constraint (8), where ν is a constant parameter determined by the pump curve.

Therefore, starting with an initial guess for the flow rates and relative speeds, the constraints in the DAE model are approximated at every iteration by constraints abiding by the GP form, as listed in Table I. This process continues until a termination criterion is met. The details are further discussed in Algorithm 1, after the presentation of the abstract GP form and the conversion of the control objectives in the next section.

B. Abstract GP model

To express the GP-based form of WDN-MPC in a compact form, we define some mathematical operators.

Definition 1. For matrices $\boldsymbol{X}$ and $\boldsymbol{B} \in \mathbb{R}^{m \times n}$, the *element-wise exponential* operation on $\boldsymbol{X}$ with base $\boldsymbol{B}$ is a matrix of the same dimension with elements given by

$$\hat{\boldsymbol{X}} = \boldsymbol{B}^{\boldsymbol{X}} = \begin{bmatrix} b_{11}^{x_{11}} & \cdots & b_{1n}^{x_{1n}} \\ \vdots & \ddots & \vdots \\ b_{m1}^{x_{m1}} & \cdots & b_{mn}^{x_{mn}} \end{bmatrix} = \begin{bmatrix} \hat{x}_{11} & \cdots & \hat{x}_{1n} \\ \vdots & \ddots & \vdots \\ \hat{x}_{m1} & \cdots & \hat{x}_{mn} \end{bmatrix}.$$

Definition 2. For matrices $\boldsymbol{Y} \in \mathbb{R}^{n \times m}$ and matrix $\boldsymbol{X} \in \mathbb{R}^{m \times p}$, the *element-wise exponential matrix product* $\boldsymbol{C} = \boldsymbol{X} \star \boldsymbol{Y}$ has elements given by $c_{ij} = \prod_{k=1}^{m}(\hat{x}_{kj})^{y_{ik}}$ for $i = 1, \ldots, n$ and $j = 1, \ldots, p$, where $\hat{x}_{kj} = b^{x_{kj}}$.

Property 1. For matrices $\boldsymbol{Y}$ with size $n \times m$ and $\boldsymbol{X}$ with size $m \times p$, let $\hat{\boldsymbol{X}} = b^{\boldsymbol{X}}$, where b is base. The following holds:

$$b^{\boldsymbol{Y}\boldsymbol{X}} = \hat{\boldsymbol{X}} \star \boldsymbol{Y}.$$

With the above definitions, we now derive the GP-based DAE model and constraints of WDN. Performing an element-wise exponential operation on both sides of (12) yields

$$b^{\boldsymbol{x}(k+1)} = b^{\boldsymbol{A}\boldsymbol{x}(k)+\boldsymbol{B}_u\boldsymbol{u}(k)+\boldsymbol{B}_v\boldsymbol{v}(k)}$$

$$\mathbf{1} = b^{\boldsymbol{E}_u\boldsymbol{u}(k)+\boldsymbol{E}_v\boldsymbol{v}(k)+\boldsymbol{E}_d\boldsymbol{d}(k)}$$
$$\mathbf{1} = b^{\boldsymbol{E}_x\boldsymbol{x}(k)+\boldsymbol{E}_l\boldsymbol{l}(k)+\boldsymbol{\Phi}(\boldsymbol{u}(k),\boldsymbol{v}(k),\boldsymbol{s}(k))}.$$

Denote $\hat{\boldsymbol{x}}(k) = b^{\boldsymbol{x}(k)}$ and similarly $\boldsymbol{l}(k)$, $\boldsymbol{u}(k)$, $\boldsymbol{v}(k)$, and $\boldsymbol{s}(k)$ are converted into $\hat{\boldsymbol{l}}(k), \hat{\boldsymbol{u}}(k), \hat{\boldsymbol{v}}(k)$ and $\hat{\boldsymbol{s}}(k)$. The models of junctions and tanks can be written as monomials (18a) and (18b) directly according to Property 1.

For a pipe from node i to j, according to (6), the exponential of nonlinear function is $C^{\mathrm{P}}(k)\hat{q}_{ij}(k)$. The head loss constraints can be compactly written for all pipes using the element-wise product $\boldsymbol{F}_v(k) \circ \hat{\boldsymbol{v}}(k)$, where $\boldsymbol{F}_v(k)$ is a $n_p \times 1$ column vector collecting the $C^{\mathrm{P}}(k)$ parameters of pipes.

Similarly, for all pumps, define $\boldsymbol{F}_s(k)$ and $\boldsymbol{F}_u(k)$ as $n_m \times 1$ column vectors respectively collecting all of parameters $C_1^{\mathrm{M}}(k)$ and $C_2^{\mathrm{M}}(k)$ of pumps. In summary, (18c) and (18d) are the abstract GP form of pipe and pump models. The overall DAE-GP model is given as follows

DAE-GP:

$$\hat{\boldsymbol{x}}(k+1) = [\boldsymbol{A} \star \hat{\boldsymbol{x}}(k)] \circ [\boldsymbol{B}_u \star \hat{\boldsymbol{u}}(k)] \circ [\boldsymbol{B}_v \star \hat{\boldsymbol{v}}(k)] \quad (18a)$$

$$\mathbf{1}_{n_j} = [\boldsymbol{E}_u \star \hat{\boldsymbol{u}}(k)] \circ [\boldsymbol{E}_v \star \hat{\boldsymbol{v}}(k)] \circ [\boldsymbol{E}_d \star \hat{\boldsymbol{d}}(k)] \quad (18b)$$

$$[\boldsymbol{E}_x \star \hat{\boldsymbol{x}}(k)] \circ [\boldsymbol{E}_l \star \hat{\boldsymbol{l}}(k)] = \boldsymbol{F}_v(k) \circ \hat{\boldsymbol{v}}(k) \quad (18c)$$

$$[\boldsymbol{E}_x \star \hat{\boldsymbol{x}}(k)] \circ [\boldsymbol{E}_l \star \hat{\boldsymbol{l}}(k)] = [\hat{\boldsymbol{s}}(k)^{\boldsymbol{F}_s(k)}] \circ [\hat{\boldsymbol{u}}(k)^{\boldsymbol{F}_u(k)}]. \quad (18d)$$

In addition, the physical constraints (13) can be rewritten as

Constraints-GP:

$$\hat{\boldsymbol{x}}(k) \in [\hat{\boldsymbol{x}}^{\min}(k), \hat{\boldsymbol{x}}^{\max}(k)], \hat{\boldsymbol{l}}(k) \in [\hat{\boldsymbol{l}}^{\min}(k), \hat{\boldsymbol{l}}^{\max}(k)]$$
$$\hat{\boldsymbol{s}}(k) \in [\mathbf{1}_{n_m}, \boldsymbol{b}_{n_m}] \quad (19)$$
$$\hat{\boldsymbol{u}}(k) \in [\hat{\boldsymbol{u}}^{\min}(k), \hat{\boldsymbol{u}}^{\max}(k)], \hat{\boldsymbol{v}}(k) \in [\hat{\boldsymbol{v}}^{\min}(k), \hat{\boldsymbol{v}}^{\max}(k)].$$

C. Conversion of control objectives

In this section, we covert the control objectives in the nonconvex problem (16) to their convex, GP-based form. In (14), notice that $\boldsymbol{x}$ is a vector collecting the head h_i at tanks. The objective $(\boldsymbol{x}(k) - \boldsymbol{x}^{\mathrm{sf}})^\top (\boldsymbol{x}(k) - \boldsymbol{x}^{\mathrm{sf}})$ encourages $\boldsymbol{x}(k)$ to be close to the constant $\boldsymbol{x}^{\mathrm{sf}}$. Hence, we introduce a new auxiliary variable $\hat{\boldsymbol{z}}(k) \triangleq b^{\boldsymbol{x}^{\mathrm{sf}}-\boldsymbol{x}(k)}$ which will be pushed to be close to $\mathbf{1}$. Using the epigraph form, the original objective function $\Gamma_1(\boldsymbol{x}(k))$ in Section II-C is replaced by $\hat{\Gamma}_1(\hat{\boldsymbol{z}}(k)) = \prod_{i=1}^{n_t} \hat{z}_i(k)$ and $2n_t$ constraints are added as

$$\text{Safety-GP:} \quad \hat{\boldsymbol{z}}(k) \geq \mathbf{1}_{n_t} \quad (20a)$$
$$\hat{z}_i(k) = \hat{x}_i^{\mathrm{sf}} \hat{x}_i^{-1}(k) \quad , i \in [1, n_t] \quad (20b)$$

where $\hat{\boldsymbol{x}}^{\mathrm{sf}}$ and $\hat{\boldsymbol{x}}(k)$ are the GP form of $\boldsymbol{x}^{\mathrm{sf}}$ and $\boldsymbol{x}(k)$. If the water level of the i-th tank is above the safe level, then variable $\hat{z}_i(k)$ already satisfies the constraint, which implies no objective function is applied. Otherwise, the objective is $\hat{z}_i(k)$, and the corresponding constraint is $\hat{z}_i(k) = \hat{x}_i^{\mathrm{sf}} \hat{x}_i^{-1}(k)$. Moving to the second part of the objective function (15), $\Delta \boldsymbol{u}(k) = \boldsymbol{u}(k) - \boldsymbol{u}(k-1)$ is a vector collecting the flow changes of controllable flow $\boldsymbol{u}(k)$. We introduce a new aux-

iliary variable $\hat{\boldsymbol{p}}(k) \triangleq b^{\boldsymbol{u}(k)-\boldsymbol{u}(k-1)}$ and perform an element-wise exponential operation on both sides of (15) yielding

$$b^{[\boldsymbol{u}(k)-\boldsymbol{u}(k-1)]^\top \Delta \boldsymbol{u}(k)} = [\hat{\boldsymbol{u}}(k) \circ \hat{\boldsymbol{u}}^{-1}(k-1)]^{\Delta \boldsymbol{u}(k)}$$
$$= (\hat{p}(k))^{\Delta \boldsymbol{u}(k)}.$$

Using the epigraph form, the original objective function $\Gamma_2(\Delta \boldsymbol{u}(k))$ can be expressed as a new objective $\hat{\Gamma}_2(\hat{\boldsymbol{p}}(k)) = \prod_{i=1}^{n_u}(\hat{p}_i(k))^{\Delta u_i(k)}$ and $n_u + 1$ constraints given as

Smoothness-GP:

$$\hat{\boldsymbol{p}}_i(k) = \hat{\boldsymbol{u}}_i(k)\hat{\boldsymbol{u}}_i^{-1}(k-1), \forall\, i \in \{1, n_u\}, \; \hat{\Gamma}_2(\hat{\boldsymbol{p}}(k)) \geq 1. \quad (21)$$

Given the above derivations, the final GP form of multi-objective cost function can be rewritten as

$$\hat{\Gamma}(\hat{\boldsymbol{z}}(k), \hat{\boldsymbol{p}}(k)) = \hat{\Gamma}_1(\hat{\boldsymbol{z}}(k)) + \hat{\Gamma}_2(\hat{\boldsymbol{p}}(k)), \quad (22)$$

where $\hat{\Gamma}(\hat{\boldsymbol{z}}(k), \hat{\boldsymbol{p}}(k))$ is a posynomial function. The convex GP-based MPC can now be expressed as

$$\min_{\substack{\hat{\boldsymbol{\xi}}[t_0] \\ \hat{\boldsymbol{z}}(k), \hat{\boldsymbol{p}}(k)}} \quad \hat{\Gamma}\left(\hat{\boldsymbol{z}}(k), \hat{\boldsymbol{p}}(k) \,\middle|\, \hat{\boldsymbol{x}}(t_0), \left\{\hat{\boldsymbol{d}}(k)\right\}_{k=t_0}^{k=t_0+H_p}\right)$$

$$\text{s.t. } \textit{DAE-GP} \ (18), \ \textit{Constraints-GP} \ (19) \quad (23)$$
$$\textit{Safety-GP} \ (20), \ \textit{Smoothness-GP} \ (21).$$

In (23), two sets of optimization variables are included. The first set comprises $\hat{\boldsymbol{x}}$, $\hat{\boldsymbol{l}}$, $\hat{\boldsymbol{u}}$, $\hat{\boldsymbol{v}}$, and $\hat{\boldsymbol{s}}$ which are collected in variable $\hat{\boldsymbol{\xi}}[t_0]$. The latter is the corresponding GP form of $\boldsymbol{\xi}[t_0]$ defined in Section II-C, that is, $\hat{\boldsymbol{\xi}}[t_0] \triangleq b^{\boldsymbol{\xi}[t_0]}$. The second set includes the auxiliary variables $\hat{\boldsymbol{z}}$ and $\hat{\boldsymbol{p}}$ introduced before. Notice that the flow $\hat{q}_{ij}$ is an optimization variable while q_{ij} is not in (23), but a value used to calculate $C^{\mathrm{P}}(k)$, $C_1^{\mathrm{M}}(k)$ and $C_2^{\mathrm{M}}(k)$. Note that similar to the nonconvex problem (16), Problem (23) also requires the knowledge of the state $\hat{\boldsymbol{x}}(t_0)$ as well as a prediction of the demand pattern for an entire prediction horizon H_p. The detailed form of (23) is given in Table I. We next present how we integrate the GP iterations with the MPC windows.

As we mentioned in Section III-A, the notation $\langle q_{ij}(k) \rangle_n$ stands for the n-th iteration value of q_{ij} at time k. We use the same notation system during iterations, e.g., $\langle \hat{\boldsymbol{u}}(k) \rangle_n$ and $\langle \hat{\boldsymbol{\xi}} \rangle_n$ are the n-th iterate value of $\hat{\boldsymbol{u}}(k)$ and $\hat{\boldsymbol{\xi}}$. The algorithm of GP method for a single optimization window is presented in Algorithm 1. We initialize the flow $\langle \hat{\boldsymbol{u}}(k) \rangle_0$ and $\langle \hat{\boldsymbol{v}}(k) \rangle_0$

Algorithm 1 GP method for a single optimization window.

1: **Input:** $\hat{\boldsymbol{x}}(t_0)$, $\{\hat{\boldsymbol{d}}(k)\}_{k=t_0}^{k=t_0+H_p}$
2: **Output:** $\hat{\boldsymbol{\xi}}[t_0]$
3: Initialize $n = 0$, parameters $\langle \hat{\boldsymbol{\xi}} \rangle$; set $\hat{\boldsymbol{\xi}}_{\text{SaveLast}} = \langle \hat{\boldsymbol{\xi}} \rangle_0$
4: **while** error $\geq$ threshold **OR** $n \leq$ maxIter **do**
5: $\quad n = n + 1$
6: $\quad$ Obtain $\langle C^{\mathrm{P}} \rangle_n$, $\langle C_1^{\mathrm{M}} \rangle_n$ and $\langle C_2^{\mathrm{M}} \rangle_n$ by $\langle \hat{\boldsymbol{\xi}} \rangle_{n-1}$
7: $\quad$ Solve (23) for $\hat{\boldsymbol{\xi}}_{\text{Sol}}$; obtain error $= \|\hat{\boldsymbol{\xi}}_{\text{Sol}} - \hat{\boldsymbol{\xi}}_{\text{SaveLast}}\|$
8: $\quad$ Assign $\langle \hat{\boldsymbol{\xi}} \rangle_n = \hat{\boldsymbol{\xi}}_{\text{SaveLast}} = \hat{\boldsymbol{\xi}}_{\text{Sol}}$
9: **end while**

---

in $\langle \hat{\boldsymbol{\xi}}(k) \rangle_0$, $k \in [t_0, t_0 + H_p]$ with the historical statistical average flow in the pipes and pumps, and $\langle \hat{s}(k) \rangle_0$ is set to **1**. The parameters $\langle C^{\mathrm{P}}(k) \rangle_1$, $\langle C_1^{\mathrm{M}}(k) \rangle_1$ and $\langle C_2^{\mathrm{M}}(k) \rangle_1$ are then calculated according to Section III-A. After solving (23) and obtaining the $\hat{\boldsymbol{\xi}}_{\text{Sol}}$, we save it as $\hat{\boldsymbol{\xi}}_{\text{SaveLast}}$, obtain the iteration error, and make $\hat{\boldsymbol{\xi}}_{\text{Sol}}$ as the initial value for next iteration. In addition, we define the error as the distance between current solution $\hat{\boldsymbol{\xi}}_{\text{Sol}}$ and previous solution $\hat{\boldsymbol{\xi}}_{\text{SaveLast}}$. The iteration continues until the error is less than a predefined error threshold (threshold) or a maximum number of iterations (maxIter) is reached. The details are in Algorithm 1. Upon successfully solving for a single optimization window, we can continue the process and solve for the overall simulation time using a vintage MPC routine.

IV. NUMERICAL TESTS

In this section, we present a simple simulation example to illustrate the applicability of the GP-based MPC formulation for WDN operations. The considered water network is an 8-node network from EPANET Users Manual [17, Chapter 2]. The numerical tests are simulated using EPANET Matlab Toolkit [20] with GP solver *GGPLAB* [21]. The basic parameters in 8-node network including the elevation of nodes, length, and diameter of pipes are obtained from [17]. We now present the list of constraints and parameters used in the simulations.
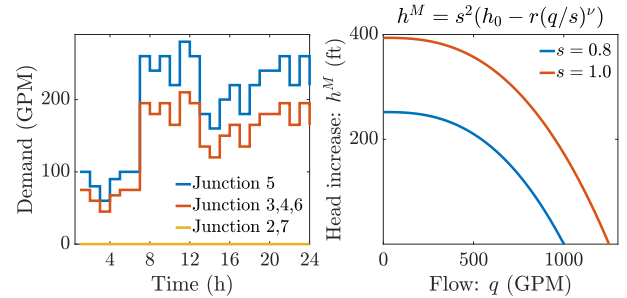

Fig. 1. Water demand at various junctions and variable-speed pump curve ($h_0 = 393.7$, $r = 3.7 \times 10^{-6}$, $\nu = 2.59$).

• The initial head of Tank 8 is 834 ft, the water level range of Tank 8 is $[830, 850]$ ft, and the safety water level x^{sf} (14) of Tank 8 from Section II-C is set to 838 ft. We set the total simulation time T_{final} to 24 hrs in Algorithm 1. The demand pattern for 24 hours at different junctions is shown in Fig. 1. This demand pattern is different from [17], as our intention is to make the demand vary more rapidly to test the performance of the presented GP-based MPC routine.

• The relationship between head increase and flow of Pump 9 defined by (7) is presented in Fig. 1. We observe that the head increase and flow provided by a pump varies with the relative speed $s \in [0, 1]$ with $s = 0$ refers to the pump being off. In (9), the physical constraints of the head imposed at the i-th junction is greater than its corresponding elevation, and the head of i-th reservoir is fixed at its elevation. Since we have only one reservoir, this implies that $h_1^{\mathrm{R}} = 700$ ft. As for the flow, the direction is unknown, and we simply constrain the flow to $q_i \in [-1000, 1000]$ GPM.

• For the geometric programming component of the presented formulations, we set the base $b = 1.005$. The parameters we

use in Algorithm 1 are selected as: error $= 0.5$ and maxIter $= 40$. We consider a sampling time of $1\,\text{hr}$, a prediction horizon $H_p = 10\,\text{hrs}$, and a simulation of $24\,\text{hrs}$.

Here, we present the solution to WDNs operation problem after running Algorithm 1. Fig. 2 shows the optimal control effort (the variable pump speed), the water level after applying the pump control, and the water flow through some pipes. During time period $k \in [1, 2]$, the water level of Tank 8 is below the safety water level 838 ft, so the relative speed of Pump 9 is set to the highest speed $s = 1$ by GP-MPC controller. Consequently, Pump 9 starts to deliver water into the network and more water begins to flow into the tank. At $k = 3$, the target setpoint is reached and the pump slows down to a relative speed of $s = 0.7566$ according to MPC algorithm. Notice that the demand at $k = 7$ increases dramatically as shown in Fig. 1, and as a result the pump speed increases to the maximum speed at $k = 4$ to prepare for this situation in advance. During time period $k \in [7, 13]$, the demand changes
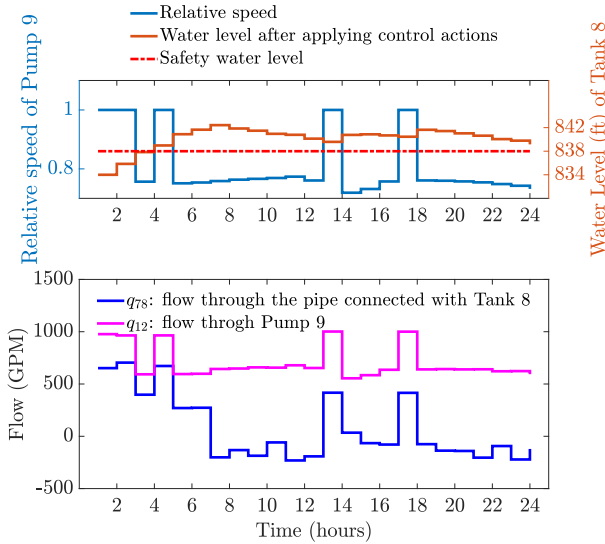

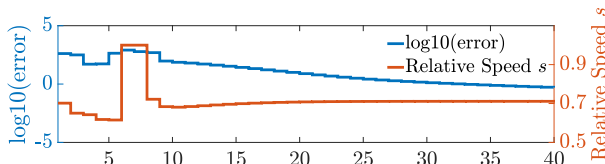Fig. 2. Relative speed of Pump 9, controlled water level of Tank 8, and flows through a pipe and Pump 9.

Fig. 3. Convergence of the Iteration error and relative speed from Algorithm 1.

slightly, and the relative speed is stabilized around $s = 0.8$. At $k = 14$, the water level decreases to 838 ft, which leads to an increase in the pump speed to maintain the desired, safe water level. During time period $k \in [18, 24]$, the increase in demand justifies the higher pump speed which is then followed by a decrease in the speed for the last stretch of the day. Fig. 3 shows the convergence of Algorithm 1—the GP method for a single optimization window when $t_0 = 23$. We can see the iteration error keeps decreasing which implies the final solution is reached slowly. For the first 7 iterations, the ups and downs of error is caused by whether the objective function (22) is activated, because the solved $\boldsymbol{x}$ from Algorithm 1 is not accurate and varies. When the solved $\boldsymbol{x}$ becomes stable, the

objective function (22) does not change, and Algorithm 1 converges to the final solution. It is noteworthy to mention that the flow $q_{12}(k)$ through Pump 9 is always in one direction, while the flow direction of $q_{78}(k)$ through the pipe connected to Tank 8 changes 5 times during the simulation in Fig. 2. This justifies the use of the *flow direction-unaware GP* component in the MPC. Future work will experiment with larger networks and investigate the impact of demand uncertainty on the WDN control.

REFERENCES

[1] H. Cross, "Analysis of flow in networks of conduits or conductors," University of Illinois at Urbana Champaign, College of Engineering. Engineering Experiment Station., Tech. Rep., 1936.

[2] D. J. Wood and C. O. Charles, "Hydraulic network analysis using linear theory," *Journal of the Hydraulics division*, vol. 98, no. 7, pp. 1157–1170, 1972.

[3] M. L. Arora, "Flows split in closed loops expending least energy," *Journal of the Hydraulics Division*, vol. 102, no. 3, pp. 455–458, 1976.

[4] M. Bazrafshan, N. Gatsis, M. Giacomoni, and A. Taha, "A fixed-point iteration for steady-state analysis of water distribution networks," in *Proc. 6th IEEE Global Conf. Signal and Information Processing*, Anaheim, CA, Nov. 2018. https://arxiv.org/abs/1807.01404

[5] M. K. Singh and V. Kekatos, "On the flow problem in water distribution networks: Uniqueness and solvers," *arXiv preprint arXiv:1901.03676*, 2019.

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

[7] Y. Wang, V. Puig, and G. Cembrano, "Non-linear economic model predictive control of water distribution networks," *Journal of Process Control*, vol. 56, pp. 23–34, 2017.

[8] G. S. Sankar, S. M. Kumar, S. Narasimhan, S. Narasimhan, and S. M. Bhallamudi, "Optimal control of water distribution networks with storage facilities," *Journal of Process Control*, vol. 32, pp. 127–137, 2015.

[9] Y. Wang, C. Ocampo-Martinez, and V. Puig, "Stochastic model predictive control based on gaussian processes applied to drinking water networks," *IET Control Theory & Applications*, vol. 10, no. 8, pp. 947–955, 2016.

[10] C. C. Sun, V. Puig, and G. Cembrano, "Combining csp and mpc for the operational control of water networks," *Engineering Applications of Artificial Intelligence*, vol. 49, pp. 126–140, 2016.

[11] J. Humpola and A. Fügenschuh, "A unified view on relaxations for a nonlinear network flow problem," 2013.

[12] M. K. Singh and V. Kekatos, "Optimal scheduling of water distribution systems," *arXiv:1806.07988*, 2018.

[13] M. Xie and M. Brdys, "Nonlinear model predictive control of water quality in drinking water distribution systems with dbps objectives."

[14] L. Sela Perelman and S. Amin, "Control of tree water networks: A geometric programming approach," *Water Resources Research*, vol. 51, no. 10, pp. 8409–8430, 2015.

[15] D. Fooladivanda and J. A. Taylor, "Energy-optimal pump scheduling and water flow," *IEEE Transactions on Control of Network Systems*, no. 3, pp. 1016–1026, Sept. 2018.

[16] A. S. Zamzam, E. Dall'Anese, C. Zhao, J. A. Taylor, and N. Sidiropoulos, "Optimal water-power flow problem: Formulation and distributed optimal solution," *IEEE Transactions on Control of Network Systems*, 2018.

[17] L. A. Rossman *et al.*, "Epanet 2: users manual," 2000.

[18] C. Ocampo-Martinez, V. Puig, G. Cembrano, and J. Quevedo, "Application of predictive control strategies to the management of complex networks in the urban water cycle applications of control." Institute of Electrical and Electronics Engineers, 2013.

[19] S. Boyd, S.-J. Kim, L. Vandenberghe, and A. Hassibi, "A tutorial on geometric programming," *Optimization and engineering*, vol. 8, no. 1, p. 67, 2007.

[20] D. Elíades and M. Kyriakou, "Epanet matlab toolkit," *University of Cyprus, Republic of Cyprus*, 2009.

[21] A. Mutapcic, K. Koh, S. Kim, L. Vandenberghe, and S. Boyd, "Ggplab: A simple matlab toolbox for geometric programming," *web page and software: http://stanford. edu/boyd/ggplab*, 2006.