Learning-Based Predictive Control with Gaussian Processes: An Application to Urban Drainage Networks

Balla, Krisztian Mark; Eringis, Deividas; Ahdab, Mohamad Al; Bendtsen, Jan Dimon; Kallesøe, Carsten; Ocampo-Martinez, Carlos

Published in:
Learning-Based Predictive Control with Gaussian Processes: An Application to Urban Drainage Networks

Publication date:
2022

Document Version
Accepted author manuscript, peer reviewed version

Learning-Based Predictive Control with Gaussian Processes: An Application to Urban Drainage Networks

Krisztian Mark Balla[1,2],Deividas Eringis[1], Mohamad Al Ahdab[1], Jan Dimon Bendtsen[1], Carsten Skovmose Kallesøe[1,2], Carlos Ocampo-Martinez[3],

***Abstract*— Many traditional control solutions in urban drainage networks suffer from unmodelled nonlinear effects such as rain and wastewater infiltrating the system. These effects are challenging and often too complex to capture through physical modelling without using a high number of flow sensors. In this article, we use level sensors and design a stochastic model predictive controller by combining nominal dynamics (hydraulics) with unknown nonlinearities (hydrology) modelled as Gaussian processes. The Gaussian process model provides residual uncertainties trained via the level measurements and captures the effect of the hydrologic load and the transport dynamics in the network. To show the practical effectiveness of the approach, we present the improvement of the closed-loop control performance on an experimental laboratory setup using real rain and wastewater flow data.**

I. INTRODUCTION

Real-time control in Urban Drainage Networks (UDNs) allows for the systematic mitigation of water volumes, typically exploiting the available sensor measurements, weather forecasts, and, in some cases, the available physical description of the network. In an urban drainage context, sensors typically measure flow and level, furthermore, the most common actuators are pumps and gates. In combined UDNs, the disturbances are considered as the meteorological and human waste water loads. In this article, we focus on combined UDNs, where the actuators are pumps and both rain and wastewater gravitate from station to station in open pipes until reaching the Waste Water Treatment Plant (WWTP). To predict the volumes, e.g., avoid overflows and utilize the system capacity equally, it is essential to measure levels and flows. Here, we distribute only easy-accessible level sensors in the network and use a Gaussian Process (GP)-based system identification. The proposed approach uses water level residuals to capture the nonlinearities coming from the behavior of the pipes and the infiltration of the disturbances. Furthermore, the approach captures the uncertainties in modelling and the disturbance forecasts.

Modelling UDNs for control is a complex task, given the delays and nonlinearities imposed by the flow transport between pumping stations. The flow-to-level translation inside the sewers requires either a High Fidelity (HiFi) model or a large number of flow measurements, often economically out of reach for smaller water utilities. Traditional techniques on conceptual modelling are reported in [1] and [2], where the capacity of pipes are collectively modelled as virtual buffers, and in [3], where the flow-to-level translation is modelled by polynomials. Grey-box modelling for level propagation in open pipes has been reported in [4], [5] and [6].

Model Predictive Control (MPC) for UDNs has been reported in [3], [2], [1] and [7], where the operational constraints and the weather forecasts have been considered deterministic. Taking into account disturbance uncertainties with ensemble forecasts has been reported in [5], while [8] reported on using chance constraints to track the daily demands of drinking water consumption.

GP regression has been widely used in machine learning [9] for applications where an unknown system is given without structural information. A systematic framework for uncertainty propagation in real-time control of dynamic systems has been proposed in [10], while in [11], flow forecasting has been done with the use of GPs in drinking water applications. Learning with data via GPs and using a nominal model allows identifying the nonlinearities and provides a measure of uncertainty without any prior knowledge.

The contribution of the article is the following. A model and control methodology is proposed for UDNs that can characterize the uncertainty along with the predictions and reject the meteorological disturbances. To this end, we utilize level sensors to generate residuals between the unknown part (pipe dynamics, model, and disturbance uncertainty) and the nominal part (storage tanks). We incorporate the topological structure between the GP outputs and select the regressors based on prior process knowledge. To show the practical feasibility of the problem in UDNs, we carry out experimental tests on a scaled laboratory setup, where the dimensionality of the uncertainty propagation via the GPs is reduced by actively selecting the points used for prediction.

The article is organized as follows. In Section II, an overview of the operation of UDNs is presented. Section III-A first presents the GP regression, followed by the formulation of the uncertainty propagation. In Section IV, the stochastic MPC design is presented, whereupon the point selection algorithm is introduced. In Section V, the experiment on the laboratory setup is detailed and results are presented, using disturbance data from a real-world network. This is followed by Section VI, where conclusions, and future research directions are drawn.

[1] Section for Automation and Control, Department of Electronic Systems, Aalborg University, Fredrik Bajers vej 7c, DK-9220 Aalborg, Denmark, Email: {kmb,der,mah,dimon,csk}@es.aau.dk

[2] Technology Innovation, Controls Department , Grundfos Holding A/S, Poul Due Jensens vej 7, DK-8850 Bjerringbro, Denmark, Email: {kballa, ckallesoe}@grundfos.com

Automatic Control Department, Universitat Politècnica de Catalunya, Llorens i Artigas 4-6, Planta 2, 08028, Barcelona, Spain, Email: {carlos.ocampo}@upc.edu

Nomenclature: Let $\mathbb{R}, \mathbb{R}^n, \mathbb{R}^{m\times n}$ denote the field of real numbers, the set of real column vectors of length n and the set of $m \times n$ matrices composed of entries in the real numbers, respectively. The superscript $^\intercal$ denotes transposition and I is the identity matrix of suitable dimensions. A normally distributed vector x with mean μ and variance σ is denoted by $x \sim \mathcal{N}(\mu, \sigma)$, and the expected value of a random variable by $\mathbb{E}\{\}$.

II. UDN MODEL

A. Network representation

We consider stations, where pumps are placed in tanks acting as the actuators. Moreover, the main piping layout defines the topology of the network. In UDNs, the tree topology is the most common in practice [12], where the collected wastewater, rain, and groundwater are pumped from station to station until they reach the root of the network. The root is an outlet point, where the water is discharged either to a WWTP or to the environment. An illustration of main transport lines in a UDNs are shown in Fig. 1.

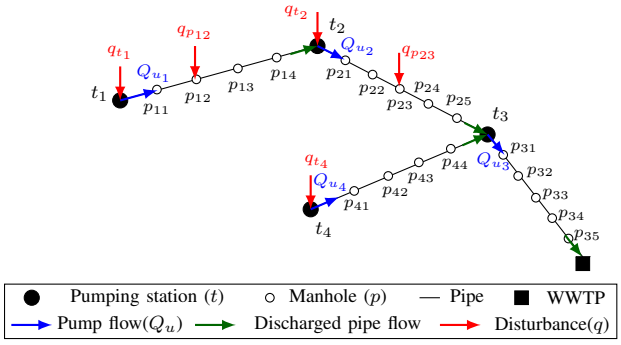

Fig. 1. Tree topology of UDNs, where filled nodes represent storage tanks.

Disturbances represent the meteorological load on the sewer network. In this work, we use forecasts of runoff flow due to rain combined with the wastewater produced by households. These disturbances enter the piping network through the nodes representing storage tanks or manholes. Note that our representation of the network takes into account only the main pipelines, which connect the pumping stations.

Remark 1: To measure level variation in both tanks and pipes, we distribute level sensors along the network nodes. The location of the level sensors is based on the high-level piping layout, meaning that we aim to deploy sensors at network nodes where the disturbances act on the network, e.g., where urban areas or catchments are discharging.

B. Physical component model

Flow propagating in open-channel pipes is most commonly approximated by the Saint-Venant nonlinear PDEs, describing the mass and the momentum of the fluid [2]:

$$\frac{\partial A(x,t)}{\partial t} + \frac{\partial Q(x,t)}{\partial x} = \tilde{q}(x,t), \tag{1a}$$

$$\frac{\partial Q(x,t)}{\partial t} + \frac{\partial}{\partial x}\left(\frac{Q(x,t)^2}{A(x,t)}\right) + gA(x,t)\left(\frac{\partial h(x,t)}{\partial x} + S_f - S_b\right) = 0, \tag{1b}$$

where $Q(x,t)$ is the flow propagating inside the pipe at location x and time t and $\tilde{q}(x,t) = q(x,t)/dx$ is the lateral inflow per unit length, where we refer to $q(x,t)$ lateral inflow as a disturbance. $A(x,t)$ is a function describing the wetted pipe area while $h(x,t)$ is the level of water inside the channel. Besides, $Q(x,t), q(x,t), A(x,t)$ and $h(x,t)$ are functions from $(0,L)\times\mathbb{R}_+ \to \mathbb{R}_+$, and L, S_b and S_f are the length, slope and friction parameters, respectively.

The stored volumes in the network are modelled by linear tanks, for which the change in level per time unit is computed as the sum of all in- and outflows, i.e.,

$$\tau\frac{dh_t(t)}{dt} = q_t(t) + Q(t) - Q_u(t), \tag{2}$$

where $Q_u(t)$ denotes the sum of flows generated by pumps, $h_t(t)$ is the level in the tank, $q_t(t)$ is disturbance inflow to storage tanks, while $Q_t(t)$ is the incoming gravitated discharge from upstream stations. Besides, τ is the tank parameter, representing the geometry and size of the tank.

C. Data-driven model

The first-principle dynamics of transport pipes in (1) are coupled by means of boundary conditions and the full PDEs can be computationally expensive to solve for complex network topology. As an alternative to [10], we consider a discrete-time representation of the entire UDN in the form,

$$\begin{aligned} x(k+1) = {} & f(x(k), u(k), d(k)) \\ & + B_d\big(g(x(k), u(k), d(k)) + w(k)\big), \end{aligned} \tag{3}$$

where the model is composed of a nominal part f, describing integrators, i.e., the discrete-time storage tanks in (2), whereas the remaining part defined by g represents the time and spatially discretized dynamics, i.e., the transport model for pipes in (1). Besides, $x(k) \in \mathbb{R}^{N_x}$, $u(k) \in \mathbb{R}^{N_u}$ and $d(k) \in \mathbb{R}^{N_d}$ are the system state, input and disturbance at time step k, respectively. The process noise $w \sim \mathcal{N}(0, \Sigma^w)$ is considered independent identically distributed, with Σ^w being a diagonal variance matrix. Besides, B_d is a matrix mapping states corresponding to the pipe dynamics from the full state vector x. Furthermore, we consider the nominal dynamics to be linear in the standard state-space form:

$$x(k+1) = f(x(k), u(k), d(k)) = Ax(k) + Bu(k) + Ed(k), \tag{4}$$

where A, B, E are constant matrices of suitable dimensions.

III. GAUSSIAN PROCESS REGRESSION

Similarly to [10], we use GPs to identify the unknown dynamics g and the uncertainty w in (3). A GP model is a probabilistic, non-parametric framework, most commonly used in supervised machine learning for predicting the distribution of output variables [9]. In order to formulate output data (or target points) for the GPs, we create residuals between our measurements and the output of the nominal model dynamics, using the formulation in (3). We assume that state measurements x_i are available at time step i:

$$y_i \triangleq g(x_i, u_i, \hat{d}_i) + w_i = B_p^\dagger\big(x_{i+1} - f(x_i, u_i, \hat{d}_i)\big), \tag{5}$$

where $\hat{d}_i \in \mathbb{R}^{N_d}$ is the vector of forecasted disturbances and $B_p^\dagger$ is the Moore-Penrose pseudo-inverse. The training set $\mathcal{D}$ is constructed from the inputs z and outputs y by collecting data with a nominal controller

$$\begin{aligned}\mathcal{D} = \{ & y = (y_1, \dots, y_M)^\intercal \in \mathbb{R}^{M \times N_y} \\ & z = (z_1, \dots, z_M)^\intercal \in \mathbb{R}^{M \times N_z} \}\,,\end{aligned} \tag{6}$$

where $z_i \triangleq [x_i^\intercal \; u_i^\intercal \; \hat{d}_i^\intercal]^\intercal$, $N_z = N_x + N_u + N_d$ and M denotes the number of collected data points. Note that we use the GPs not only for capturing the model uncertainties w but to take care of the error between the forecasted and actual disturbances (d_i and $\hat{d}_i$) as well. Furthermore, we assume that the residuals for each state are independent, and therefore, we perform a GP regression for each residual y^a with $a \in \{1, \dots, N_y\}$. Since each finite collection of y^a is normally distributed, we can write, for each y^a,

$$y^a \sim \mathcal{N}\left(\mu^a(z), K_{zz}^a + I\sigma_a^2\right), \tag{7}$$

where y^a is the a^{th} column of y and σ_a^2 is the process noise variance. Besides, $\mu^a(z)$ is the mean and K_{zz}^a is the Gram matrix such that $K_{ij}^a = k^a(z_i, z_j)$, with $k^a(z_i, z_j)$ being a kernel function [9]. We use the kernel function to describe the prior of the GP distribution, e.g., the covariance K between the points belonging to set $\mathcal{D}$ in (6). The choice of the kernel function k^a is determined based on knowledge of the physical process. The squared exponential kernel function is chosen assuming dynamics that exhibit smooth and continuous behavior [13], i.e.,

$$k^a(z_i, z_j) = \sigma_{f,a}^2 \exp\left(-\frac{1}{2}(z_i - z_j)^\intercal S_a^\intercal \Lambda_a^{-1} S_a (z_i - z_j)\right), \tag{8}$$

where the hyper-parameter $\sigma_{f,a}$ is the signal variance and $\Lambda_a^{-1} = \operatorname{diag}(\sigma_{L,1}^{-2}, ..., \sigma_{L,N_z}^{-2})$ is the length scale matrix. Note that different length-scale parameters are used on each dimension of z, thereby determining the relative importance of the contributions made by each input.

Remark 2: Mapping matrices $S_{i_1,\dots,i_n}^a \in \mathbb{R}^{n \times N_z}$ are introduced for each output dimension a, picking states x_i, inputs u_i and disturbances $\hat{d}_i$ for each residual y_i. The mapping is determined based on the structure of the network, i.e., based on the physical network model in (1) and (2). This allows to reduce the training set for each output dimension, thereby easing the computational cost due to the high dimension of the training set $\mathcal{D}$.

Given a testing point z_*, we aim to predict the residual $y_* = (g(z_*) + w_*)$ given the training set $\mathcal{D}$, i.e., we aim to find the distribution of $p(y_*^a | y^a)$. The joint distribution is

$$\begin{pmatrix} y^a \\ y_*^a \end{pmatrix} \sim \mathcal{N}\left(\mu^a, \begin{bmatrix} K_{zz}^a + I\sigma_a^2 & K_{zz^*}^a \\ K_{z^*z}^a & K_{z^*z^*}^a \end{bmatrix}\right), \tag{9}$$

where $[K_{zz^*}^a]_i = k^a(z_i, z^*)$, $K_{z^*z}^a = (K_{zz^*}^a)^\intercal$, and $K_{z^*z^*}^a = k^a(z^*, z^*)$. The conditional distribution of the residuals is Gaussian [14], where the mean and variance are given as

$$y_*^a | y^a \sim \mathcal{N}(\mu_a^p(z^*), \Sigma_a^p(z^*)), \tag{10a}$$

$$\mu_a^p(z^*) = K_{z^*z}^a \left(K_{zz}^a + I\sigma_a^2\right)^{-1} y^a, \tag{10b}$$

$$\Sigma_a^p(z^*) = K_{z^*z^*}^a - K_{z^*z}^a \left(K_{zz}^a + I\sigma_a^2\right)^{-1} K_{zz^*}^a, \tag{10c}$$

where μ_a^p and Σ_a^p are the mean and variances of the GP for output dimension a, respectively. By stacking the predicted residuals y_*^a in a single vector $p(z^*)$, we can write

$$p(z^*) \sim \mathcal{N}\left(\mu^p(z^*), \Sigma^p(z^*)\right), \tag{11}$$

with mean $\mu^p(z^*) = [\mu_1^p(z^*), \dots, \mu_{N_x}^p(z^*)]^\intercal$, and variance $\Sigma^p(z^*) = \operatorname{diag}(\Sigma_1^p(z^*), \dots, \Sigma_{N_x}^p(z^*))$.

A. Uncertainty propagation

The iterative, multi-step ahead prediction with the GP model is done by feeding back the mean and the variance of the predicted states, making each input a Gaussian random variable. Hence, the prediction of the states is in general non-Gaussian [14], as the probability distribution of the GP needs to be propagated through the nonlinear kernel function in (8). In this work, the states and the GP are approximated as jointly Gaussians, where the predicted mean μ_i^x and covariance Σ_i^x of the states are given by

$$\mu_{i+1}^x = f(\tilde{z}_i) + B_d \mu_i^p, \tag{12a}$$

$$\Sigma_{i+1}^x = [\nabla_x f(\tilde{z}_i) \quad B_p] \Sigma_i [\nabla_x f(\tilde{z}_i) \quad B_p]^\intercal, \tag{12b}$$

where $\tilde{z}_i \triangleq (\mu_i^x, u_i, \hat{d}_i)$, the mean μ^p and variance Σ^p are given by (10), and Σ_i is the covariance of the jointly Gaussian approximation of the states and the GP. Note that the inputs are assumed to be known and therefore treated as deterministic variables. The mean μ_i and the covariance Σ_i of the joint distribution are then given by

$$\begin{pmatrix} x_i \\ p_i + w_i \end{pmatrix} \sim \mathcal{N}(\mu_i, \Sigma_i) = \left(\begin{bmatrix} \mu_i^x \\ \mu_i^p \end{bmatrix}, \begin{bmatrix} \Sigma_i^x & \Sigma_i^{xp} \\ \Sigma_i^{px} & \Sigma_i^p + \Sigma_i^w \end{bmatrix}\right),$$

where $\Sigma^{px} = (\Sigma^{px})^\intercal$ are the cross-covariances between the states and the GP. Due to the linear nominal dynamics in (4), the mean and variance dynamics can be simply written with the cross-covariances

$$\mu_{i+1}^x = A\mu_i^x + Bu_i + E\hat{d}_i + B_p \mu_i^p, \tag{13a}$$

$$\Sigma_{i+1}^x = A\Sigma_i^x A^\intercal + B_p \Sigma_i^{px} A^\intercal + A\Sigma_i^{xp} B_p^\intercal + B_p(\Sigma_i^p + \Sigma_i^w) B_p^\intercal. \tag{13b}$$

To solve the above approximation of the state distribution in a tractable way, the dynamics of the Gaussian distribution in (12) are found through Taylor series expansion of (10) around the mean μ^x. The covariance matrix Σ_i^p for each GP and the cross-covariances Σ_i^{xp} between the GP and the states are then updated such that

$$\mu_i^p = \mu^p(\tilde{z}_i), \tag{14a}$$

$$\Sigma_i^{xp} = \Sigma_i^x \left(\nabla_x \mu^p(\tilde{z}_i)\right)^\intercal \tag{14b}$$

$$\Sigma_i^p = \Sigma^p(\tilde{z}_i) + \nabla_x \mu^p(\tilde{z}_i) \Sigma_i^x \left(\nabla_x \mu^p(\tilde{z}_i)\right)^\intercal, \tag{14c}$$

where $\mu^p(\tilde{z}_i)$ and $\Sigma^p(\tilde{z}_i)$ are for each GP dimension a, as stated in (10). The Taylor approximation used in this work is detailed in [14]. Different methods for approximating the posterior of a GP from Gaussian input have been researched, see for example [9].

IV. Stochastic MPC design

A. Tractable GP-MPC

Introducing the nonlinear kernel and propagating the uncertainties with the states following a Gaussian distribution $x_i \sim \mathcal{N}(\mu_i^x, \Sigma_i^x)$ adds complexity to the optimization problem behind the GP-MPC. To solve the optimization problem in a tractable way, we formulate the problem as follows:

$$\mathcal{L}(k) = \mathbb{E}\Big\{ \sum_{i=k}^{k+H_p-1} ||x_i - x_i^r||_Q^2 + ||\Delta u_i||_R^2 + ||\epsilon||_S^2 \Big\}, \quad (15)$$

where H_p denotes the length of the prediction horizon, the Δu term is introduced for integral action on flow control of the pumps and x^r is a reference signal for water level in the tanks. The reference is introduced in the state penalty term to show the effectiveness of the closed-loop behavior. Although this is somewhat unrealistic regarding the application, we artificially create reference scenarios to push the controller to its limits when testing under uncertain forecast signals. The slack variable ϵ is introduced for state constraint relaxation for water level violation in storage tanks, where the amount of level violation is related to the overflow volumes escaping the system. Furthermore, the weights Q, R and S represent a prioritization between the different objectives, and the individual terms in (15) are normalized, such that water level and flow terms are comparable in magnitude. Although the evolution of the states is stochastic, we chose to implement the optimization problem by simply considering deterministic state constraints where the slack variables for overflow provide recursive feasibility to the problem.

Using the approximate distribution of the states and the expected value of the objective function, we formulate the deterministic optimization problem behind the GP-MPC as

$$\min_{\Delta u_i} \sum_{i=k}^{k+Hp-1} ||\mu_i^x - x_i^r||_Q^2 + \mathrm{tr}(Q\Sigma_i^x) + ||\Delta u_i||_R^2 + ||\epsilon_i||_S^2 \quad (16a)$$

$$\text{s.t.} \quad \mu_{i+1}^x = f(\tilde{z}_i) + B_d \mu^p(\tilde{z}_i), \quad (16b)$$

$$\Sigma_{i+1}^x = [\nabla_x f(\tilde{z}_i)\ B_p] \Sigma_i [\nabla_x f(\tilde{z}_i)\ B_p]^\intercal, \quad (16c)$$

$$\Delta u_i = u_i - u_{i-1}, \quad (16d)$$

$$H_x \mu_i^x \leq b_x + H_\epsilon \epsilon_i, \quad (16e)$$

$$H_u u_i \leq b_u, \quad (16f)$$

$$\mu_i^p,\ \Sigma_i \text{ according to (13b), (14)}, \quad (16g)$$

$$\mu_0^x = x(k), \quad \Sigma_0^x = 0, \quad (16h)$$

where $i = k, ..., k + H_p - 1$, furthermore (16e) and (16f) are $2N_x$ and $2N_u$ dimensional polytopes representing state and input constraints, respectively, where $H_x \in \mathbb{R}^{2N_x \times Nx}$, $b_x \in \mathbb{R}^{N_x}$, $H_u \in \mathbb{R}^{2N_u \times N_u}$, $b_u \in \mathbb{R}^{N_u}$ and $H_\epsilon \in \mathbb{R}^{2N_x \times Nx}$.

The optimization problem in (16) is solved in a receding horizon fashion where the dynamics f are discretized with the fixed step, 4th order Runge-Kutta method. Moreover, the single-shooting method is used in the symbolic framework `CasADI` [15] and a primal-dual interior point solver, `IPOPT` [16] is used to solve the non-convex optimization problem. The optimization is solved with warm start at each time step, given the process is slowly-varying.

B. Subset of data approximation

The computational complexity of solving the optimization problem presented in (16) is highly influenced by the GP model representing the unknown pipe dynamics and the uncertainty of the disturbance load on the network. The computational burden of propagating the uncertainty depends on the number of data points M used in (6). To lower the computational complexity, the data set used for the prediction needs to be used in a computationally efficient way. Several sparse methods exist for approximating the distribution of GPs [17], among which we use the Subset of Data (SoD) method, where the computation is reduced to $\mathcal{O}(\tilde{M}^3)$ from $\mathcal{O}(M^3)$, by using a subset of $\tilde{M} < M$ data points [18]. Similar point selection methods have been used in [19]. The SoD method is shown in Fig. 2.

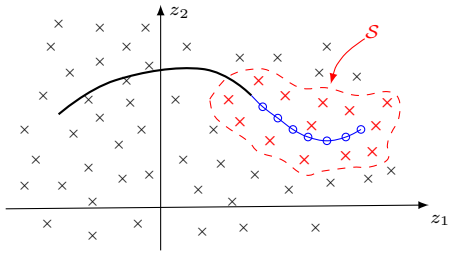

Fig. 2. Conceptual point selection scheme, where $\mathcal{S}$ is the selected subset.

As opposed to more advanced sparse approximation methods, the computation with SoD can be reduced drastically, however, to the cost of degrading the prediction quality. Therefore, we spend the extra resources on using a larger subset $\tilde{M}$ and selecting new points at each sampling time [20]. The proposed approach is detailed in Algorithm 1.

Algorithm 1: Subset of Data point selection

$\mathcal{P} = z,\ i = 0\ ,\ \mathcal{I} = \emptyset\ ;$
while $i < \tilde{M}$ **do**
  for $j = 0$ *to* $H_p - 1$ **do**
    i = i + 1;
    r = argmin$_r$ $||\mathcal{P}(r) - \hat{z}_j||^2$;
    $\mathcal{I} = \mathcal{I} \cup r\ ;$
    $\mathcal{P} = \mathcal{P} \setminus r\ ;$
    if $i = \tilde{M}$ **then**
      break
    end
  end
end
return $\mathcal{S} = \{y(\mathcal{I}),\ z(\mathcal{I})\};$

The method considers the predictions provided by solving the finite-horizon optimization problem in (16).

At each time step we choose a subset $\mathcal{S}$ of the training set $\mathcal{P}$, such that all points in $\mathcal{S}$ are close to the previously predicted trajectory $\hat{z}$. In Algorithm (1), the index set $\mathcal{I}$ is a set containing the indices r. Here, $\mathcal{I}$ is used to index into the full training data set $\mathcal{P}$, e.g., to find the locations of the data points closest to the previously predicted trajectory $\hat{z}_j$, where $j = (0, ..., H_p - 1)$. The SoD method assumes

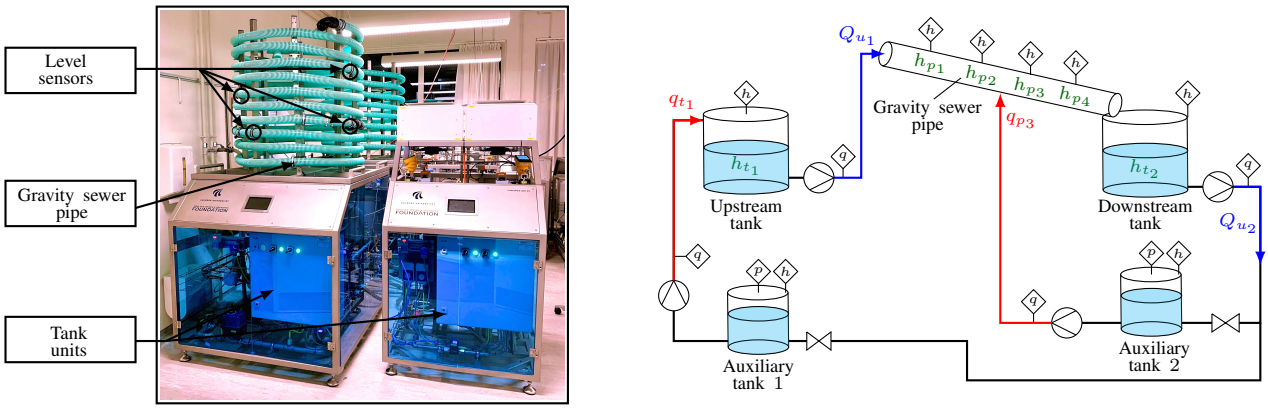

Fig. 3. Sewer modules of the Smart Water Laboratory setup at Aalborg university (left). Schematics of the experimental setup (right). The sensor placements are indicated for each individual module with pressure sensor (p), level sensor (h) and flow sensor (q).

that the selected points are close enough to the solution trajectory calculated at the previous time step such that the prediction accuracy for the current time step is sufficient. Using the previous solution trajectory as the selection criteria is a fair assumption in case of the control of UDNs, as the dynamics and disturbances are slowly varying with respect to the sampling time of the controller.

V. CASE STUDY

The experimental setup for testing the GP-MPC in UDN control is shown in Fig. 3 [21]. This setup of a UDN represents a $1:80$ scale model of a section of a real-life UDN, meaning that the typical resolution and time scale of the disturbances, control steps, tank emptying, and sampling times have been scaled down accordingly. An upstream and downstream pumping station are connected by a gravity-driven sewer line, most commonly found in real-life pumped sewer infrastructures [22]. The open-channel pipeline is equipped with four level sensors equivalent to water levels measured in manholes in a real network. As shown in Fig. 3, auxiliary tanks are utilized to pump the disturbances to the points where they act on the system. Note that in our practical setup we generate these disturbances by pumping them to the inflow locations, while our controller knows only the forecast. The level sensor measurements of the pipe and the two tanks, furthermore the flow of the pumps are obtained and locally managed at each unit with a Codesys soft-PLC in real-time [23]. The data acquisition is done at every 0.5 s, while the control input is applied at every 10 minute. The periodic component of the disturbances is equal to 17 minutes, which corresponds to one day in real life.

A. UDN model

Following the methodology in Section II, the states, inputs and disturbances are given by the physical variables

$$x = [h_{t_1} \quad h_{t_2} \quad h_{p_1} \quad h_{p_4}]^\intercal, \tag{17}$$
$$u = [Q_{u_1} \quad Q_{u_2}]^\intercal, \tag{18}$$
$$\hat{d} = [q_{t_1} \quad q_{p_3}]^\intercal, \tag{19}$$

where the input set $z = [x, u, \hat{d}]^\intercal$ is constructed as in (6), where $N_x = 4$, $N_u = 2$ and $N_d = 2$, following the notation shown in Fig. 3. Note that we utilize only h_{p_1} and h_{p_4} at the up- and downstream end of the pipeline, for the reason that the input flow Q_{u_1} from station one and the lateral disturbance flow q_{p_3} in the middle of the pipeline can be captured indirectly on these two level measurements. Hence, excluding h_{p_2} and h_{p_3} eases computation.

The nominal dynamics in (4) are given in the form

$$A = \begin{bmatrix} I_{2\times 2} & 0_{2\times 2} \\ 0_{2\times 2} & 0_{2\times 2} \end{bmatrix}, B = \begin{bmatrix} \frac{T_s}{\tau_{t_1}} & 0 \\ 0 & \frac{T_s}{\tau_{t_2}} \\ & 0_{2\times 2} \end{bmatrix}, E = \begin{bmatrix} -\frac{T_s}{\tau_{t_1}} & 0 \\ 0 & -\frac{T_s}{\tau_{t_2}} \\ & 0_{2\times 2} \end{bmatrix}$$

where $A \in \mathbb{R}^{N_x \times N_x}$, $B \in \mathbb{R}^{N_x \times N_u}$ and $E \in \mathbb{R}^{N_x \times N_d}$. Moreover, T_s is the sampling time of the controller. Note that the nominal dynamics include the two integrator states h_{t_1} and h_{t_2}, whereas the remaining entries in the state vector are zeros, meaning that the nonlinear pipe dynamics are not part of the nominal dynamics. Hence, the GPs are chosen to take into account the pipe dynamics, the model uncertainty of the integrator states and the uncertainty of the forecasted disturbances $\hat{d}$. As such, the matrix B_p in (3) is an identity of suitable dimensions. The uncertainty on the forecasted disturbances $\hat{d}$ is designed as an additive white noise on top of the mean of the forecast. Furthermore, due to the scaled nature of the Smart Water Laboratory setup, uncertainty and additional dynamics are present due to the pumps.

The mapping matrices defined in (8) are given for each GP dimension $a = 1, ..., 4$, respectively:

$$S^1_{i_1,\dots,i_{n_1}} \in \mathbb{R}^{n_1 \times N_z}, \quad i \in \{1, 5, 7\}$$
$$S^2_{i_1,\dots,i_{n_2}} \in \mathbb{R}^{n_2 \times N_z}, \quad i \in \{2, 4, 6\}$$
$$S^3_{i_1,\dots,i_{n_3}} \in \mathbb{R}^{n_3 \times N_z}, \quad i \in \{3, 5\}$$
$$S^4_{i_1,\dots,i_{n_4}} \in \mathbb{R}^{n_4 \times N_z}, \quad i \in \{3, 4, 8\}$$

where n defines the total number of dimensions we pick from the original N_z dimensional data set.
Furthermore, i specifies the number of entry we pick from z for each output dimension a, respectively.

B. Experimental results

Following the model and data collection methodology in Section II, the residuals y are constructed between the level

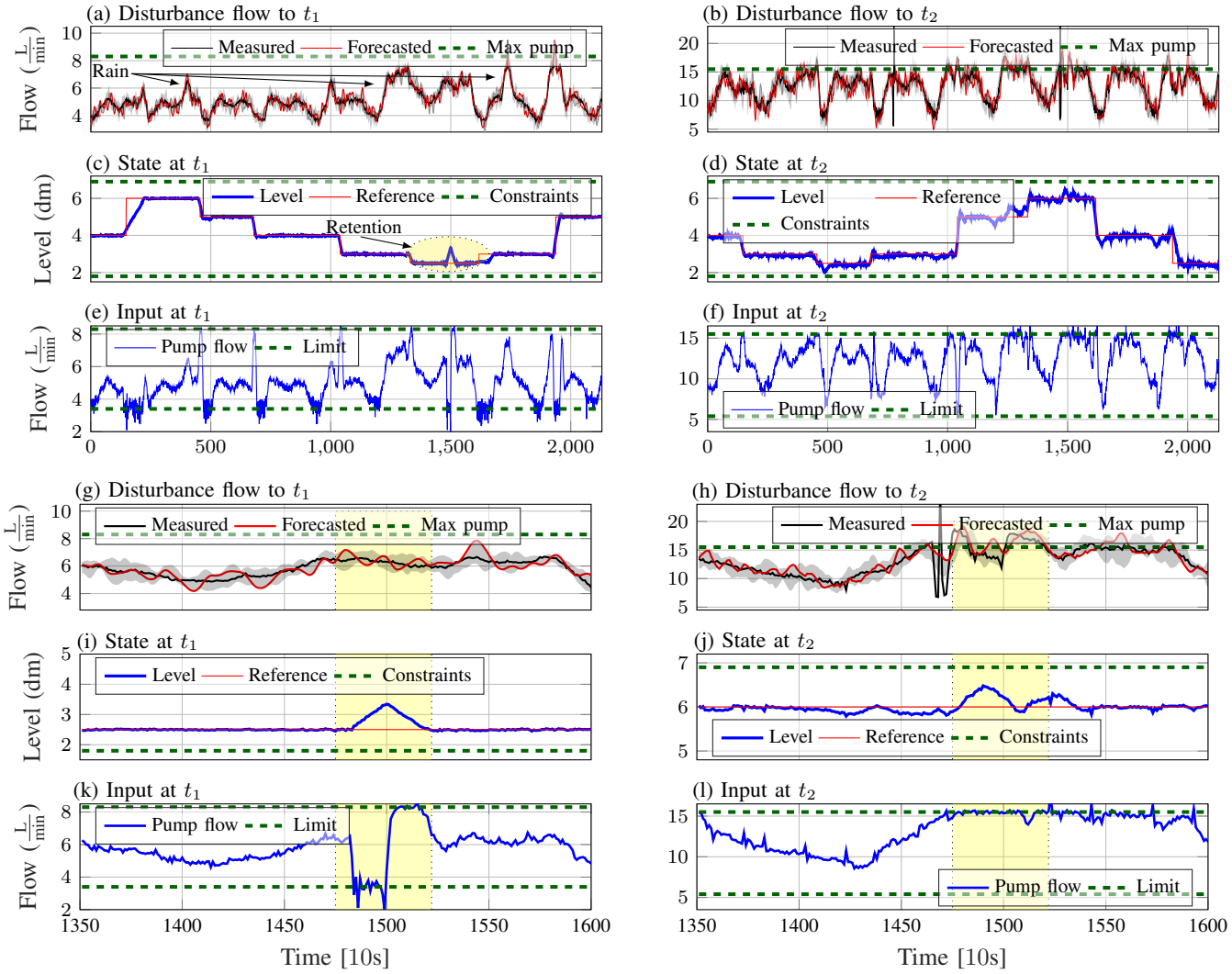

Fig. 4. Closed-loop control results of the GP-MPC for disturbance rejection with water level reference tracking under uncertain weather forecasts.

measurements x in (17) and the nominal tank dynamics in (4). The data is collected from the laboratory setup under a nominal, threshold-based level controller, most commonly found in practice [12]. The hyper-parameters σ_f^2 and Λ are found for each output dimension $a = 1, ..., N_a$, using Bayesian optimization with the `fitrgp` toolbox in MATLAB. Moreover, $H_p = 25$ steps are used for prediction, corresponding to 5 hours in real life. This is a reasonable horizon length in UDNs, as rainfall radar predictions can provide sufficient accuracy only up to $2 - 4$ hours. The point selection is done on a data set, where $\tilde{M} = 80$ points are selected at each time step for the residual predictions. Besides, the full dataset z from which we select the points for prediction is continuously updated.

The experimental results of the optimization problem in (16) are shown in Fig. 4. To evaluate the capabilities of the GP-MPC under uncertainty, $n = 10$ scenarios of imperfect rain and wastewater flow forecasts have been created by adding Gaussian distributed random data on top of the historic real flow data. The imperfect forecast is provided to the GP-MPC, whereas the historical flow data is implemented on the laboratory setup. Note that the pumps at both stations have a lower constraint different from zero, although in reality pumps can be shut down completely. This is for the reason, that below the lower limits the flow-based PI controllers cannot keep the given reference due to the small pressure drop in the test setup. As shown in Fig. 4 (c) and (d), the level references at the two stations are tracked, indicating that the effect of disturbances and the unknown dynamics are learned well. As expected, the water level in t_2 indicates a higher spread of the uncertainty in the predictions. This is partly because we use the GPs to model both the disturbance uncertainty and the nonlinear pipe dynamics providing the transport between the two stations.

In Fig. (4) (g-l), we show an event, which we observe at $t = 1500$ time steps in Fig. 4 (c) and (d). The pumps at the upstream t_1 tank reduce the flow to the lower limit, aiming to retent as much volume upstream as possible. Hence, the load on tank t_2 is eased, where at $t = 1500$ we operate the station close to its upper constraint. Note that even though the pumps at t_2, shown in Fig. 4 (l), run at full speed, the water level h_{t_2} is approaching a level, where constraint violation is likely under the uncertain flow forecasts. The constraint violation of the upper tank level is highly undesirable, as it

relates to an overflow event, where the slacks ϵ have to be used in the optimization to lift the physical upper level of the tanks. The controller recognizes that the slacks in (16) need to be used to avoid the infeasibility of the optimization problem, hence the controller rather violates the tracking of the reference in the tank t_1 and shuts down the pump, as shown in Fig. 4 (i) and (k). As shown in the left column of the results in Fig. 4, the GP can learn the uncertainty on the forecasts and thereby control the pumps to the reference.

VI. CONCLUSIONs AND FUTURE WORK

In this paper, we discussed a predictive control approach in urban drainage networks, where Gaussian process regression has been used to model the unknown dynamics and the disturbance uncertainties. For this purpose, we utilized the Gaussian process regression framework to learn only the parts where the traditional predictive control lacks a good first principle modelling approach. To this end, we used level sensors distributed along the network to learn the disturbance uncertainty and the pipe dynamics from the level variation under a rule-based nominal controller. The residual prediction through uncertainty propagation along with the deterministic model predictive controller based on the known dynamics has been solved in a receding horizon fashion on an experimental laboratory setup, using disturbance flow forecast data extracted from a real-world waste water utility. The performance of the reference tracking underpinned the feasibility of the practical utility of the Gaussian process-based predictive controller in sewer networks and showed the robustness capabilities towards uncertain disturbance forecasts.

An investigation into how the periodic behavior of human disturbances can be learned from the level sensor data with Gaussian processes and using rain intensity instead of flow is a matter of future work.

ACKNOWLEDGMENT

The authors would like to thank the Poul Due Jensen Foundation for providing the Smart Water Lab at Aalborg University. The project was supported by Innovation Fund Denmark and Grundfos Holding A/S as part of a Danish Industrial Ph.D. Project [Application number: *9065-00018A*]. The work of C. Ocampo-Martinez has been supported by the L-BEST project (Ref. PID2020-115905RB-C21) from the Spanish Ministry of Science and Innovation.

REFERENCES

[1] C. Ocampo-Martinez, *Model Predictive Control of Wastewater Systems*. Barcelona: Springer, 1st ed., 2010.

[2] X. Litrico and V. Fromion, *Modeling and control of hydrosystems*. Springer, 2009.

[3] B. Joseph-Duran, C. Ocampo-Martinez, and G. Cembrano, "Output-feedback control of combined sewer networks through receding horizon control with moving horizon estimation," *Water Resources Research*, vol. 51, no. 10, pp. 8129–8145, 2015.

[4] K. M. Balla, C. Schou, J. D. Bendtsen, C. Ocampo-Martínez, and C. S. Kallesøe, "A Nonlinear Predictive Control Approach for Urban Drainage Networks Using Data-Driven Models and Moving Horizon Estimation," *IEEE Transactions on Control Systems Technology (Early Access)*, pp. 1–16, 2022.

[5] K. M. Balla, C. H. Knudsen, A. Hodzic, J. D. Bendtsen, and C. S. Kallesøe, "Nonlinear Grey-box Identification of Sewer Networks with the Backwater Effect: An Experimental Study," in *2021 IEEE Conference on Control Technology and Applications (CCTA)*, (San Diego), pp. 1202–1207, IEEE, 2021.

[6] M. Xu, P. Van Overloop, and N. Van De Giesen, "On the study of control effectiveness and computational efficiency of reduced saint-venant model in model predictive control of open channel flow," *Advances in Water Resources*, vol. 34, no. 2, pp. 282–290, 2011.

[7] L. García, J. Barreiro-Gomez, E. Escobar, D. Téllez, N. Quijano, and C. Ocampo-Martínez, "Modeling and real-time control of urban drainage systems: A review," *Advances in Water Resources*, vol. 85, pp. 120–132, 2015.

[8] J. Grosso, C. Ocampo-Martínez, V. Puig, and B. Joseph, "Chance-constrained model predictive control for drinking water networks," *Journal of process control*, vol. 24, no. 5, pp. 504–516, 2014.

[9] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*, vol. 2. MIT press Cambridge, MA, 2006.

[10] L. Hewing, J. Kabzan, and M. N. Zeilinger, "Cautious model predictive control using gaussian process regression," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 6, pp. 2736–2743, 2020.

[11] 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.

[12] N. S. V. Lund, A. K. V. Falk, M. Borup, H. Madsen, and P. Steen Mikkelsen, "Model predictive control of urban drainage systems: A review and perspective towards smart real-time water management," *Critical Reviews in Environmental Science and Technology*, vol. 48, no. 3, pp. 279–339, 2018.

[13] J. Kocijan, *Modelling and control of dynamic systems using Gaussian process models*. Springer, 2016.

[14] A. Girard, C. E. Rasmussen, J. Quinonero-Candela, and R. Murray-Smith, "Gaussian process priors with uncertain inputs: Application to multiple-step ahead time series forecasting," 2003.

[15] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "Casadi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

[16] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, no. 1, pp. 25–57, 2006.

[17] E. Snelson and Z. Ghahramani, "Sparse gaussian processes using pseudo-inputs," *Advances in Neural Information Processing Systems*, vol. 18, pp. 1259–1266, 2006.

[18] K. Chalupka, C. K. Williams, and I. Murray, "A framework for evaluating approximation methods for gaussian process regression," *Journal of Machine Learning Research*, vol. 14, pp. 333–350, 2013.

[19] J. Kabzan, L. Hewing, A. Liniger, and M. N. Zeilinger, "Learning-based model predictive control for autonomous racing," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3363–3370, 2019.

[20] J. Quinonero-Candela and C. E. Rasmussen, "A unifying view of sparse approximate gaussian process regression," *The Journal of Machine Learning Research*, vol. 6, pp. 1939–1959, 2005.

[21] J. Val Ledesma, R. Wisniewski, and C. S. Kallesøe, "Smart water infrastructures laboratory: Reconfigurable test-beds for research in water infrastructures management," *Water*, vol. 13, no. 13, p. 1875, 2021.

[22] D. Butler, C. J. Digman, C. Makropoulos, and J. W. Davies, *Urban drainage*. Crc Press, 2018.

[23] 3S-Smart Software Solutions GmbH, "Codesys."