Estimation of Node Pressures in Water Distribution Networks by Gaussian Process Regression

Ildeberto Santos-Ruiz
and Francisco-Ronay López-Estrada

Tecnológico Nacional de México
Instituto Tecnológico de Tuxtla Gutiérrez.
TURIX-Dynamics Diagnosis and Control Group
Carretera Panamericana km 1080 S/N,
29050, Tuxtla Gutiérrez, México
Email: idelossantos, frlopez@ittg.edu.mx

Vicenç Puig
and Joaquim Blesa

Universitat Politècnica de Catalunya
Institut de Robòtica i Informàtica Industrial, CSIC-UPC.
Parc Tecnològic de Barcelona. C/ Llorens i Artigas 4-6,
08028, Barcelona, Spain.
Tel: +34 93 401 5751
Email: vicenc.puig, joaquim.blesa@upc.edu

***Abstract*— This paper proposes a method to predict pressures in all nodes of a water distribution network (WDN) by Gaussian process regression (GPR) from pressure measurements in a subset of selected nodes. The pressure sensors are placed in the nodes where, together, they capture the maximum pressure variance and also have a minimum sensitivity to measurement noise. As a case study, the proposed method was tested on a dataset obtained from simulations with the hydraulic model of the Hanoi WDN. Using only three pressure sensors, the GPR estimation error in the pressures of the unmeasured nodes are comparable to the error due to measurement noise in physical pressure sensors.**

I. Introduction

Drinking water is one of the critical resources in modern life. Since water processing from its collection to its distribution to final consumers is generally expensive, it is important to manage it efficiently, in order to reduce economic losses and avoid shortages of this vital resource. The assurance of water quality and the reduction of losses due to leaks in the pipes of distribution systems are two of the main problems related to water management [1]. Concerning leakage losses, the volume of water leaked is around 30% in most cities; however, in some towns such as Tuxtla Gutiérrez (Mexico), it reaches over 60% [2]. Controlling leaks is a challenging task due to the difficulty of locating them since leaks are usually not visible directly. However, pressure variations in the network caused by leaks provide evidence that can help determine them [3]–[5].

In order to facilitate monitoring and control of water distribution networks (WDNs), these are often divided into subnetworks called district metered areas (DMAs) [6]. This sectioning in small systems with at most 3 000 nodes facilitates the isolation and control of leaks [7]. Nevertheless, in the best cases, the available instrumentation is limited to monitoring the inflow/outflow of the DMA and pressures of the inlet node and the critical node where the minimum pressure of the DMA is recorded. These few pressure measurements are not sufficient for the precise location of leaks, so it is necessary to add a higher number of sensors at other points in the network [4], [8], which is not possible due to economic limitations.

Since the monitoring of the pressure on all nodes of the network is technically and economically unfeasible, several methods have been proposed to select the optimal positions to place the pressure sensors [9]–[12]. Although some leak location methods don't requiere the knowledge of all the node pressures, e.g. classifiers and other machine learning techniques [13]–[16], it is also essential to estimate the pressure in sensor-free nodes, which could help to determine if a leak or other failure causes that a pressure fall below the minimum or exceed a tolerable value.

This problem has been studied in [17] and [18] where the authors proposed the calculation of the pressure in the non-sensed nodes using Kriging interpolation and data of the hydraulic topology of the network. Now, in this paper, a method that calculates the pressures in unmeasured nodes based on only statistical dependence between node pressures is proposed. An argument supporting this proposal is the fact that pressures at different nodes of a WDN are highly correlated in leak and non-leak scenarios (see Fig. 1). This was confirmed with a principal component analysis, which reveals that about 85% of the pressure variance on the analyzed network can be explained with only the first principal component, which proves the high correlation between all the pressures. This statistical dependence suggests directly using measured pressures to estimate unmeasured pressures, and has motivated this work.

In this paper, a method is proposed to predict the unmeasured node pressures in WDNs, from a specified set of pressure sensors placed in nodes in such a way the maximum pressure variance is captured. These pressure variations can be due to leaks and changes on demand in the consumption nodes throughout the day, among other causes. The unmeasured pressures are estimated from regression models with Gaussian processes, which can be seen as virtual sensors that provide an optimal unbiased estimate that, as will be shown, have an estimation error comparable to the error due to the measurement noise in a physical pressure sensor.

The paper is organized as follows. Section II presents the mathematical basis for fitting the GPR models that estimate

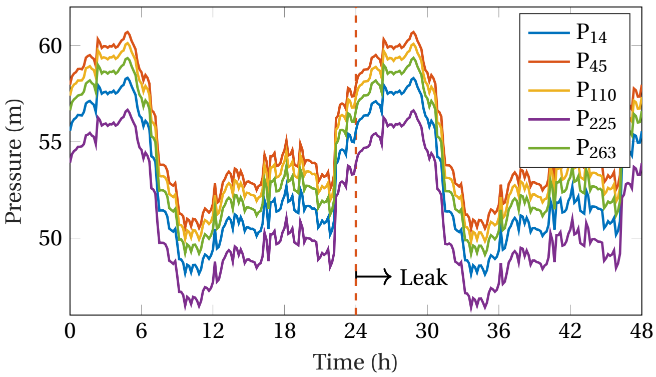

Fig. 1. Evolution of node pressures in a sector of Madrid WDN. A leak occurs at node 50 from time $t = 24$ h. The test sector consists of 312 nodes, but only the pressures in five nodes were plotted.

the node pressures. Section III shows the results of applying the GPR models to estimate pressures in Hanoi WDN, which is used as a case study. Finally, in Section IV, some conclusions are presented, as well as the direction that the research that motivated the presentation of this paper will take.

II. Foundation and Methodology

This section describes the Gaussian process regression (GPR) and its application to the prediction of pressures in sensor-free nodes. First, the Gaussian process is defined; Subsequently, it describes how to perform regressions with Gaussian processes; Finally, a method is proposed to estimate unmeasured pressures in a WDN using GPR.

A. Gaussian process

Gaussian processes (GPs) are data-driven machine learning models that have been used in regression and classification tasks. The GP provides a mechanism to make inferences about new data from previously known data sets. By modeling the data as Gaussian distributions, it is possible to make an estimation of new data (predictions) using the mean of these distributions. In addition, they also offer a measure of the uncertainty in the forecasts from the variance of the data distributions. The following description is limited to what is required for this work. A detailed and friendly explanation of GP and its applications for learning and control can be found in [19].

A GP is defined as a collection of random variables where any finite subset of these variables has a normal (Gaussian) multivariate distribution. Formally: If $\left\{f(\mathbf{x}), \mathbf{x} \in \mathbb{R}^d\right\}$ is a GP, then given n observations $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$, the joint distribution of the random variables $f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_n)$ is Gaussian. A GP is defined by the functions that determine its mean $m(\mathbf{x})$ and its covariance $k(\mathbf{x}, \mathbf{x}')$:

$$E\left(f(\mathbf{x})\right) = m(\mathbf{x}) \tag{1}$$

$$E\left(\left[f(\mathbf{x}) - m(\mathbf{x})\right]\left[f(\mathbf{x}) - m(\mathbf{x})\right]^{\mathsf{T}}\right) = k(\mathbf{x}, \mathbf{x}') \tag{2}$$

Unlike normal distributions over vectors that are characterized by a finite number of parameters, usually a mean $\boldsymbol{\mu}$ and a covariance $\boldsymbol{\Sigma}$, Gaussian processes are not parametric distributions, but they are considered distributions over functions in an infinite-dimensional vector space (often called Hilbert space). The function $f : \mathbb{R}^d \to \mathbb{R}$ that characterizes a GP is denoted by

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \,. \tag{3}$$

In most GP applications, the mean function $m(\mathbf{x})$ is assumed to be zero, and the covariance function $k(\mathbf{x}, \mathbf{x}')$ is selected according to the context or problem. Thus, for example, the polynomial kernel is frequently used in classification problems with many dimensions ($d \gg 1$), while the squared exponential kernel is frequently used in regression problems:

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\!\left(-(\mathbf{x} - \mathbf{x}')^{\mathsf{T}}(\mathbf{x} - \mathbf{x}')/(2\sigma_l^2)\right), \tag{4}$$

where σ_f and σ_l are kernel parameters, which are often called hyperparameters and grouped in a vector $\boldsymbol{\theta} = [\sigma_f, \sigma_l]$. Since the regression reliability is dependent on how well the covariance function is selected, if its hyperparameters are not chosen sensibly, the result is nonsense. The covariance function is often written as $k(\mathbf{x}, \mathbf{x}' \,|\, \boldsymbol{\theta})$ to indicate its dependence on hyperparameters $\boldsymbol{\theta}$. A compendium of the most used kernels in applications can be found in [20]–[22]. In the proposal presented, a squared exponential function is used.

B. Gaussian process regression

GPR is an application of GPs to infer (predict or estimate) values of variables that are indexed in time (e.g. Kalman filter), in space (e.g. Kriging), or in any other dimension not necessarily space-time but that has a useful meaning in applications. In regards to prediction, GPR provides an optimal unbiased estimate. It is optimal because minimizes the variance of estimation errors, and unbiased because it ensures that the mean of estimation errors towards zero. In the context of Machine Learning, GPR uses supervised learning and a measure of the similarity between points (the covariance function) to predict response for input points not included in the training data.

A GPR model is constructed from a training dataset $\{(\mathbf{x}_i, y_i);\, i = 1, 2, \ldots, n\}$, where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$. In this dataset, $\mathbf{x}_i$ are known values of the predictor variables (features), while y_i are the desired response for the corresponding inputs $\mathbf{x}_i$. The training dataset is used to fit the GPR model, which consists of tuning the parameters and hyperparameters of the model. Then, the fitted GPR model is used to predict the value of the response variable y_{new} given a new input vector $\mathbf{x}_{\text{new}}$.

In classical linear regression, the prediction model is of the form

$$y = \mathbf{x}^{\mathsf{T}}\boldsymbol{\beta} + \varepsilon, \tag{5}$$

where $\varepsilon \sim \mathcal{N}(0, \sigma^2)$. The error variance σ^2 and the coefficients $\boldsymbol{\beta}$ in (5) are estimated from the data. In a way similar to (5), GPR explains the response y by means of the following model:

$$\boldsymbol{h}(\mathbf{x})^{\mathsf{T}}\boldsymbol{\beta} + f(\mathbf{x}), \tag{6}$$

where $f(\mathbf{x}) \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}'))$. Equation (6) expresses that the data is close to a lineal model with the residuals being modeled by a GP. The functions $\boldsymbol{h}(\cdot)$ are used to project the original feature vector $\mathbf{x} \in \mathbb{R}^d$ into a new feature vector $\boldsymbol{h}(\mathbf{x}) \in \mathbb{R}^p$, and $\boldsymbol{\beta} \in \mathbb{R}^p$ is the vector of basis functions coefficients. The basis functions $\boldsymbol{h}(\cdot)$ are proposed looking for the GPR model to fit the data better than the standard linear model, first projecting the inputs in a high-dimensional space and then applying the linear model in this space instead of directly on the inputs themselves. Since the projections are fixed functions (i.e. independent of $\boldsymbol{\beta}$), the model is still linear in the parameters and, therefore analytically tractable [23]. When fitting the model, it must be optimized over the parameters $\boldsymbol{\beta}$ jointly with the hyperparameters $\boldsymbol{\theta}$ of the covariance function.

In order to create the GPR model, one latent variable $f(\mathbf{x}_i)$ is defined for each observation $\mathbf{x}_i$ in the training set, so that a response instance y_i can be modeled as

$$P(y_i \,|\, f(\mathbf{x}_i), \mathbf{x}_i) \sim \mathcal{N}\big(y_i \,|\, \boldsymbol{h}(\mathbf{x}_i)^\mathsf{T}\boldsymbol{\beta} + f(\mathbf{x}_i), \sigma^2\big)\,, \tag{7}$$

where σ^2 is the noise variance. Like $\boldsymbol{\beta}$, σ^2 is fitted in the training process.

Using a more compact notation, the probability distribution of the response (7) and the corresponding Gaussian process, given the training data, can be written in the following matrix form:

$$P(\mathbf{y} \,|\, \mathbf{f}, \mathbf{X}) \sim \mathcal{N}\big(\mathbf{y} \,|\, \mathbf{H}\boldsymbol{\beta} + \mathbf{f}, \sigma^2 \mathbf{I}_n\big) \tag{8}$$

$$P(\mathbf{f} \,|\, \mathbf{X}) \sim \mathcal{N}(\mathbf{f} \,|\, \mathbf{0}, \mathbf{K}(\mathbf{X}, \mathbf{X})) \tag{9}$$

where

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^\mathsf{T} \\ \mathbf{x}_2^\mathsf{T} \\ \vdots \\ \mathbf{x}_n^\mathsf{T} \end{bmatrix}, \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \mathbf{H} = \begin{bmatrix} \boldsymbol{h}(\mathbf{x}_1)^\mathsf{T} \\ \boldsymbol{h}(\mathbf{x}_2)^\mathsf{T} \\ \vdots \\ \boldsymbol{h}(\mathbf{x}_n)^\mathsf{T} \end{bmatrix}, \mathbf{f} = \begin{bmatrix} f(\mathbf{x}_1) \\ f(\mathbf{x}_2) \\ \vdots \\ f(\mathbf{x}_n) \end{bmatrix},$$

and $\mathbf{K}(\mathbf{X}, \mathbf{X})$ denotes the matrix containing the covariances between all input pairs of the training set, for a given set of hyperparameters $\boldsymbol{\theta}$, i.e. $\mathbf{K}_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$. This covariance matrix is also denoted by $\mathbf{K}(\mathbf{X}, \mathbf{X} \,|\, \boldsymbol{\theta})$ to indicate its dependence on the hyperparameters. Both the hyperparameters ($\boldsymbol{\theta}$), implicitly included in (9), and the parameters ($\boldsymbol{\beta}$ and σ^2) included in (8), are calculated offline during the training process. After this, the GPR model fitted to the training data $(\mathbf{X}, \mathbf{y})$ can be used online for predictions with new input data.

In order to estimate $\boldsymbol{\beta}$, $\boldsymbol{\theta}$ and σ^2 of a GPR model, the likelihood $P(\mathbf{y} \,|\, \mathbf{X})$ is maximized as a function of those parameters and hyperparameters. To facilitate the optimization process, the logarithm of the likelihood is used: $\ell(\boldsymbol{\beta}, \boldsymbol{\theta}, \sigma^2) \overset{\text{def}}{=} \log(P(\mathbf{y} \,|\, \mathbf{X}, \boldsymbol{\beta}, \boldsymbol{\theta}, \sigma^2))$. In [23] it has been shown that the log-likelihood is given by

$$\ell(\boldsymbol{\beta}, \boldsymbol{\theta}, \sigma^2) = -\frac{1}{2}\log\left(\left|\mathbf{K}(\mathbf{X}, \mathbf{X} \,|\, \boldsymbol{\theta}) + \sigma^2 \mathbf{I}_n\right|\right) - \frac{n}{2}\log 2\pi - \frac{1}{2}\left(\mathbf{y} - \mathbf{H}\boldsymbol{\beta}\right)^\mathsf{T}\left(\mathbf{K}(\mathbf{X}, \mathbf{X} \,|\, \boldsymbol{\theta}) + \sigma^2 \mathbf{I}_n\right)^{-1}\left(\mathbf{y} - \mathbf{H}\boldsymbol{\beta}\right). \tag{10}$$

From (10), the parameters and hyperparameters of the GPR model are estimated by

$$\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\theta}}, \widehat{\sigma}^2 = \arg\max_{\boldsymbol{\beta}, \boldsymbol{\theta}, \sigma^2} \ell(\boldsymbol{\beta}, \boldsymbol{\theta}, \sigma^2) \tag{11}$$

using a multivariate optimization algorithm (e.g. conjugate gradients, Nelder-Mead, etc.) on (10).

Once the parameters and hyperparameters of the GPR model are known, it is possible to make predictions with it, this means estimating the response y_{new} for each new input $\mathbf{x}_{\text{new}}$. To make predictions, it is necessary to know the probability density $P(y_{\text{new}} \,|\, \mathbf{y}, \mathbf{X}, \mathbf{x}_{\text{new}})$. From the definition of conditional probability, and assuming that each response y_i only depends on the feature vector $\mathbf{x}_i$ and its corresponding latent variable $f(\mathbf{x}_i)$, it can be shown [23] that the density of the response y_{new} at a new point $\mathbf{x}_{\text{new}}$, given $\mathbf{y}$, $\mathbf{X}$, is given by

$$P(y_{\text{new}} \,|\, \mathbf{y}, \mathbf{X}, \mathbf{x}_{\text{new}}) = \frac{P(y_{\text{new}}, \mathbf{y} \,|\, \mathbf{X}, \mathbf{x}_{\text{new}})}{P(\mathbf{y} \,|\, \mathbf{X}, \mathbf{x}_{\text{new}})} = \mathcal{N}\big(y_{\text{new}} \,|\, \boldsymbol{h}(\mathbf{x}_{\text{new}})^\mathsf{T}\boldsymbol{\beta} + \mu, \sigma^2_{\text{new}} + \Sigma\big)\,, \tag{12}$$

where

$$\mu = \mathbf{k}(\mathbf{x}_{\text{new}}, \mathbf{X})^\mathsf{T}\left(\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I}_n\right)^{-1}\left(\mathbf{y} - \mathbf{H}\boldsymbol{\beta}\right), \tag{13}$$

$$\Sigma = k(\mathbf{x}_{\text{new}}, \mathbf{x}_{\text{new}}) - \mathbf{k}(\mathbf{x}_{\text{new}}, \mathbf{X})^\mathsf{T}\left(\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I}_n\right)^{-1}\mathbf{k}(\mathbf{x}_{\text{new}}, \mathbf{X}), \tag{14}$$

and $\mathbf{k}(\mathbf{x}_{\text{new}}, \mathbf{X})$ is a vector containing the covariances between the new input vector and all input vectors of the training set.

From (12) and (13), the expected value of y_{new} for the new feature vector $\mathbf{x}_{\text{new}}$ is given by

$$E(y_{\text{new}} \,|\, \mathbf{y}, \mathbf{X}, \mathbf{x}_{\text{new}}, \boldsymbol{\beta}, \boldsymbol{\theta}, \sigma^2) = \boldsymbol{h}(\mathbf{x}_{\text{new}})^\mathsf{T}\boldsymbol{\beta} + \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_{\text{new}}, \mathbf{x}_i), \tag{15}$$

where

$$\boldsymbol{\alpha} = \left(\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I}_n\right)^{-1}(\mathbf{y} - \mathbf{H}\boldsymbol{\beta}). \tag{16}$$

Equations (15) and (16) that allow to statistically infer the response y_{new} for an input vector $\mathbf{x}_{\text{new}}$ are the same as the kriging equations used in geostatistics [24], but in these the spatial coordinates are taken as predictor variables.

The computation of predictions using GPR is mainly determined by the calculation of $\boldsymbol{\alpha}$. The complexity of this computation is $\mathcal{O}(n^3)$, so the computational cost is high for large values of n [25]. Some techniques to reduce the computational cost by approximating $\boldsymbol{\alpha}$ are discussed in [22] and [26].

C. Estimation of unmeasured pressures using GPR

The proposed method for estimating unmeasured pressures in a WDN requires a matrix of pressure measurements at the N nodes of the network. This matrix, $\mathbf{P} = [\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_N]$, is constructed with measurements of the node pressures for different operating conditions (at a different time of day or different node demands), with leaks and without leaks. If a well-calibrated hydraulic model of the network is available, then, this information can be used to generate $\mathbf{P}$ through an extended-period simulation.

A subset of nodes is selected as "measured nodes", whose pressures will be used as the predictor variable $\mathbf{x}$ defined in the previous subsection. The pressures on the remaining nodes act as response variables y, so a GPR model will be created for each node. Thus, for a network with N nodes and d sensors, a total of $N-d$ models will be built, which will function as virtual sensors to estimate unmeasured pressures. Correctly, the pressure matrix $\mathbf{P}$ is rearranged and partitioned as follows:

$$\mathbf{P}^* = [\mathbf{x}(1), \ldots, \mathbf{x}(d), \mathbf{y}(1), \ldots, \mathbf{y}(N-d)] \qquad (17)$$

Although any subset of node pressures can be used as predictor variables, it is recommended to select the nodes that capture the maximum pressure variance in the entire network. This is due to the fact that the pressure variations contain the information useful for detect leaks and other failures in the WDN. The selection of the predictor variables (nodes with sensors) is performed through a principal components analysis (PCA), discarding the node pressures that in PCA are mainly mapped to the last components, because they do not capture the essential variability of the hydraulic phenomenon, but instead capture mostly noise [27]. In this way, a robust predictor selection is obtained.

The computations of the training process to fit the parameters and hyperparameters of GPR models, as described from (7) to (11), and the computations to estimate the kth pressure $P_k = y(j)_{\text{new}}$ for $j = 1, \ldots, N-d$, as described from (12) to (16), were implemented in MATLAB using the built-in matrix operations and subroutines of the *Statistics and Machine Learning Toolbox*.

Because statistical inference is used to estimate the node pressures, in addition to the expected pressure $\widehat{P}_k$ (mean of the Gaussian distribution) it is possible to determine a confidence interval for the prediction. From elementary statistics, it is known that the amplitude of the confidence interval is determined by the standard deviation (square root of the variance) of the probability distribution. Thus, for example, the 95% confidence interval is given by

$$\widehat{P}_k - 1.96s \leq P_k \leq \widehat{P}_k + 1.96s, \qquad (18)$$

where s is the standard deviation in $P(y_{\text{new}} \,|\, \mathbf{y}, \mathbf{X}, \mathbf{x}_{\text{new}})$ according to (12) and (14). So GPR models not only allow to predict the non-measured pressures, they also allow to estimate the prediction uncertainty.

III. Results

The prediction of node pressures using GPR models, as described in the preceding section, was tested using a dataset obtained by simulation with the hydraulic model of Hanoi WDN [28]. This network has 31 consumption nodes and 34 pipes organized in 3 loops. No pumping facilities are considered since only a single fixed head source at elevation of $100\,\mathrm{m}$ is available. This WDN was simulated using the EPANET package [29] through the EPANET-MATLAB Toolkit [30]. To construct the pressure matrix $\mathbf{P}$, the network was first simulated under leak-free nominal conditions and then with leaks of $\{1, 2, \ldots, 10\}$ $\mathrm{l/s}$ in each of the 31 nodes, creating a 311×31 matrix.

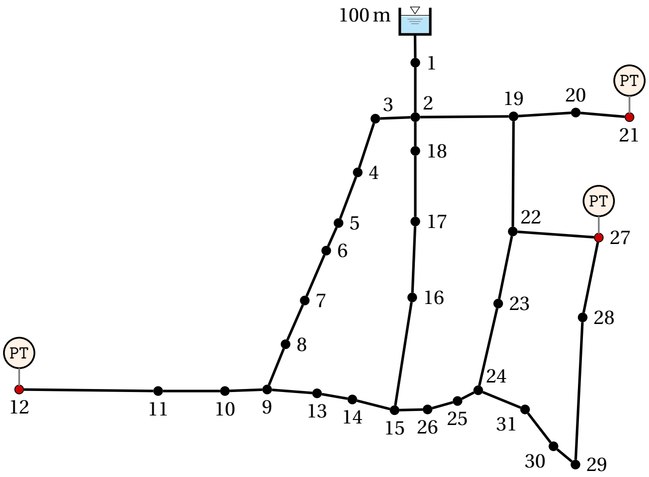

Fig. 2. The Hanoi WDN. Only in nodes marked with "PT" are sensors placed to measure the pressures used as predictors in the GPR models.

Using PCA, it was determined that if only three sensors are available, they should be placed in the nodes numbered 12, 21 and 27, for Hanoi WDN. These positions are marked with "PT" on the network map in Fig. 2. If a greater number of predictor variables are desired, to improve the accuracy of the predictions with the GPR model, through PCA it was determined that the ten most suitable nodes to sense the network pressures are, in order of importance, $\{21, 12, 27, 16, 1, 13, 31, 17, 26, 20\}$. From only the three predictor variables P_{12}, P_{21} and P_{27}, using GPR models for the pressures of the remaining 28 nodes it was possible to reconstruct the entire pressure map of the WDN with RMS error of $0.0017\,\mathrm{m}$ (0.01% of the network average pressure) in no leakage condition, and $0.0070\,\mathrm{m}$ (0.06% of the network average pressure) with leaks of $10\,\mathrm{l/s}$ maximum. In Fig. 3, the reconstructed pressure map for the non-leakage condition is shown.

Typically, only pressures at the supply point (node 1, the higher pressure) and at the critical point (node 29, the lower pressure) would be monitored. However, as determined by PCA, it is the pressures on the nodes $\{12, 21, 27\}$ that provide as much information as possible in order to reconstruct the entire map of network pressures.

Figures 4, 5 and 6 compare the pressure values estimated by GPR models against the true values obtained by simulation for three test nodes (11, 14 and 25) of Hanoi WDN. For these experiments were considered three different magnitudes of leak, $Q_{\text{leak}} = \{10, 20, 30\}$ $\mathrm{l/s}$, in each network node. To numerically assess the accuracy in the predictions, Table I shows some estimated pressures at node 11 compared to the true (target) pressures under varying pressure conditions. In all reported results, affine basis functions $\boldsymbol{h}(\mathbf{x})^{\mathsf{T}} = \left[1, \mathbf{x}^{\mathsf{T}}\right]$ (named "linear" in MATLAB) were used. Quadratic basis functions were also tested, but no significant improvements

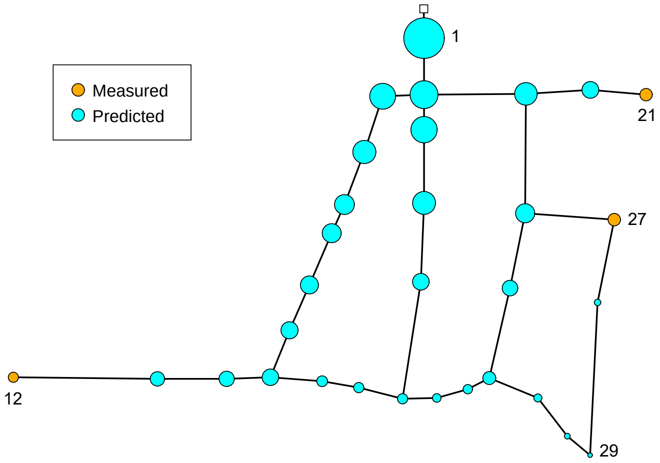

Fig. 3. Predicted pressures in all the nodes of the Hanoi WDN from only three measurements. In this "ball plot" the areas of circles are proportional to the pressure magnitude at each node.

TABLE I

PREDICTED VALUES OF P_{11} FROM MEASUREMENTS OF $\{P_{12}, P_{21}, P_{27}\}$.

Predictors			**Response**	
Sensor 1	**Sensor 2**	**Sensor 3**	**Target**	**Predicted**
P_{12}	P_{21}	P_{27}	P_{11}	$\widehat{P}_{11}$
4.1477	6.2606	6.3014	8.3558	8.3556
4.0235	6.1364	6.1772	8.2315	8.2309
3.9907	6.1350	6.1738	8.1987	8.1992
3.9494	6.1332	6.1696	8.1574	8.1587
3.9016	6.1311	6.1648	8.1096	8.1105
⋮	⋮	⋮	⋮	⋮
3.6765	5.6358	4.9594	7.8845	7.8845
3.6725	5.6384	5.1660	7.8805	7.8806

were achieved in the regression performance.

Finally, Fig. 7 shows the graph of the standard deviation that represents the uncertainty of the estimates for each node, according to (12) and (14). In this graph it is noted that the greatest uncertainty in the estimated pressure corresponds to node 16, which was expected because this node is topologically distant from the three measured nodes. The standard deviation in this case is $s = 0.0484\,\mathrm{m}$ and the estimated pressure is $\widehat{P}_{16} = 11.2984\,\mathrm{m}$, so that the 95% confidence interval, according to (18), results

$$11.2035\,\mathrm{m} \leq P_{16} \leq 11.3933\,\mathrm{m}, \tag{19}$$

which contains the true value $P_{16} = 11.3057\,\mathrm{m}$.

Considering the worst case (19), if the measurements of the predictor variables are assumed to be accurate, the uncertainty in the estimated pressure is almost comparable to the measurement noise that would result from physically measuring the variable, so it is feasible to estimate the non-measured pressures through GPR.

IV. CONCLUSIONS

A method to estimate unmeasured pressures in WDNs from a subset of pressure measurements was presented. The prediction method based on data-driven GPR models showed good results with a dataset obtained by simulation on Hanoi

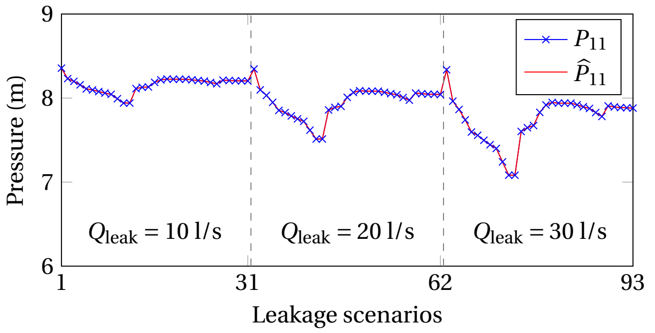

Fig. 4. Predicted pressure at node 11 of Hanoi WDN from measures at nodes $\{12, 21, 17\}$.

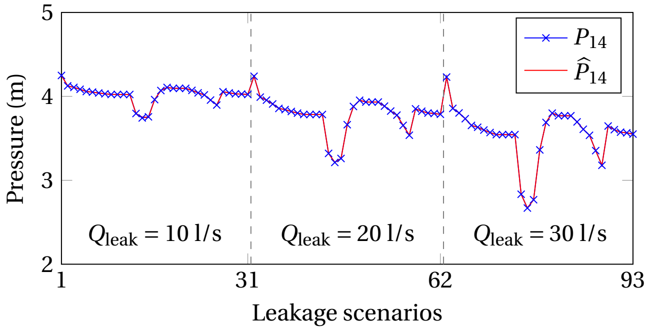

Fig. 5. Predicted pressure at node 14 of Hanoi WDN from measures at nodes $\{12, 21, 17\}$.

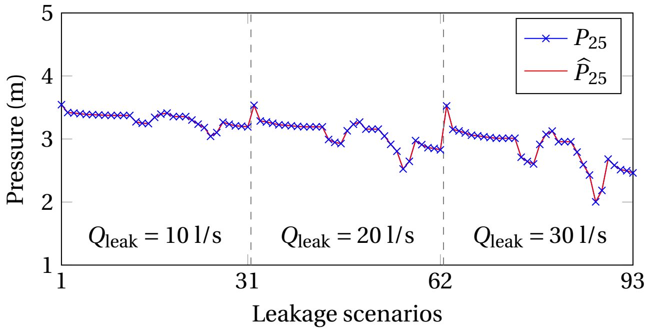

Fig. 6. Predicted pressure at node 25 of Hanoi WDN from measures at nodes $\{12, 21, 17\}$.

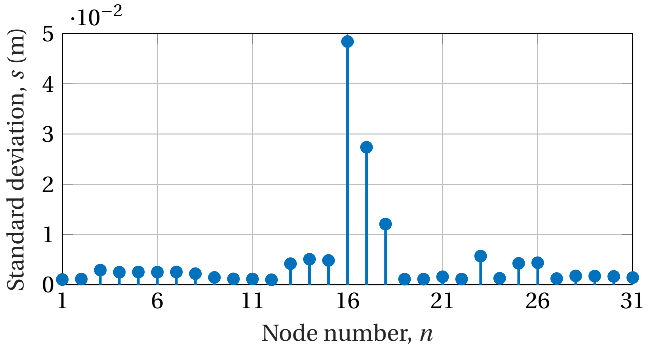

Fig. 7. Standard deviation that determines the uncertainty in the pressure estimates.

WDN which was used as a benchmark. The good results are also attributed to the fact that the available pressure sensors are not placed arbitrarily but their optimal position is calculated so that they capture the maximum pressure variance in the network. In the near future, this method will be tested with measurements of a real hydraulic network.

It is important to note that these results can be considered to develop methods for detecting and locating leaks in WDNs using a "pressure map" continuously updated through GPR. Two possibilities have been considered: one where two-dimensional pressure maps of the network are compared (with leakage and without leakage) using techniques for image processing, and another where the node pressures are processed as one-dimensional lists.

Acknowledgment

This work has been funded by scholarship of the National Institute of Technology of Mexico (TecNM) and financial support of the National Council for Science and Technology (CONACYT) of Mexico, grant number PN-2016/3595.

References

[1] J. Meseguer and J. Quevedo, *Real-Time Monitoring and Control in Water Systems*. Springer International Publishing, 2017, pp. 1–19. [Online]. Available: https://doi.org/10.1007/978-3-319-50751-4_1

[2] OECD, "Water Governance in Cities," *OECD Studies on Water*, feb 2016. [Online]. Available: http://dx.doi.org/10.1787/9789264251090-en

[3] R. Puust, Z. Kapelan, D. A. Savic, and T. Koppel, "A review of methods for leakage management in pipe networks," *Urban Water Journal*, vol. 7, no. 1, pp. 25–45, 2010.

[4] R. Pérez, V. Puig, J. Pascual, J. Quevedo, E. Landeros, and A. Peralta, "Methodology for leakage isolation using pressure sensitivity analysis in water distribution networks," *Control Engineering Practice*, vol. 19, no. 10, pp. 1157–1167, 2011.

[5] A. Soldevila, J. Blesa, S. Tornil-Sin, E. Duviella, R. M. Fernandez-Canti, and V. Puig, "Leak localization in water distribution networks using a mixed model-based/data-driven approach," *Control Engineering Practice*, vol. 55, pp. 162–173, 2016.

[6] J. Saldarriaga, J. Bohorquez, D. Celeita, L. Vega, D. Paez, D. Savic, G. Dandy, Y. Filion, W. Grayman, and Z. Kapelan, "Battle of the water networks district metered areas," *Journal of Water Resources Planning and Management*, vol. 145, no. 4, p. 04019002, 2019. [Online]. Available: https://ascelibrary.org/doi/abs/10.1061/(ASCE)WR.1943-5452.0001035

[7] A. Di Nardo, M. Di Natale, and A. Di Mauro, *Water District Metering policy and practice*. Springer Vienna, 2013, pp. 11–24. [Online]. Available: https://doi.org/10.1007/978-3-7091-1493-3_2

[8] R. Sarrate, F. Nejjari, and J. Blesa, *Sensor Placement for Monitoring*. Springer International Publishing, 2017, pp. 153–173. [Online]. Available: https://doi.org/10.1007/978-3-319-50751-4_9

[9] M. V. Casillas, L. E. Garza-Castañón, and V. Puig, "Optimal sensor placement for leak location in water distribution networks using evolutionary algorithms," *Water*, vol. 7, no. 11, pp. 6496–6515, 2015.

[10] A. Soldevila, S. Tornil, R. M. Fernandez, J. Blesa, and V. Puig, "Optimal sensor placement for classifier-based leak localization in drinking water networks," in *3rd Conference on Control and Fault-Tolerant Systems (SysTol)*, Barcelona, Spain, September 2016, pp. 325–330.

[11] J. Blesa, F. Nejjari, and R. Sarrate, "Robust sensor placement for leak location: analysis and design," *Journal of Hydroinformatics*, vol. 18, no. 1, pp. 136–148, 2016.

[12] M. À. Cugueró-Escofet, V. Puig, and J. Quevedo, "Optimal pressure sensor placement and assessment for leak location using a relaxed isolation index: Application to the barcelona water network," *Control Engineering Practice*, vol. 63, pp. 1–12, 2017.

[13] J. Mashford, D. De Silva, D. Marney, and S. Burn, "An approach to leak detection in pipe networks using analysis of monitored pressure values by support vector machine," in *Third International Conference on Network and System Security, 2009. NSS'09*. IEEE, 2009, pp. 534–539.

[14] K. Aksela, M. Aksela, and R. Vahala, "Leakage detection in a real distribution network using a SOM," *Urban Water Journal*, vol. 6, no. 4, pp. 279–289, 2009.

[15] A. Soldevila, S. Tornil-Sin, J. Blesa, R. M. Fernandez-Canti, and V. Puig, *Leak Localization in Water Distribution Networks Using Pressure Models and Classifiers*. Cham: Springer International Publishing, 2017, ch. 10, pp. 191–212. [Online]. Available: https://doi.org/10.1007/978-3-319-55944-5_10

[16] A. Soldevila, J. Blesa, S. Tornil-Sin, R. M. Fernandez-Canti, and V. Puig, "Sensor placement for classifier-based leak localization in water distribution networks using hybrid feature selection," *Computers & Chemical Engineering*, vol. 108, pp. 152–162, 2018.

[17] A. Soldevila, T. N. Jensen, J. Blesa, S. Tornil-Sin, R. M. Fernandez-Canti, and V. Puig, "Leak localization in water distribution networks using a kriging data-based approach," in *2018 IEEE Conference on Control Technology and Applications (CCTA)*, Aug 2018, pp. 577–582.

[18] M. Javadiha, J. Blesa, V. Puig, and A. Soldevila, "Leak localization in water distribution networks using deep learning," in *6th International Conference on Control, Decision and Information Technologies*, April 2019.

[19] M. Liu, G. Chowdhary, B. C. da Silva, S.-Y. Liu, and J. P. How, "Gaussian processes for learning and control: A tutorial with examples," *IEEE Control Systems Magazine*, vol. 38, no. 5, pp. 53–86, 2018.

[20] C. R. Souza, "Kernel functions for machine learning applications," *Creative Commons Attribution-Noncommercial-Share Alike*, vol. 3, p. 29, 2010.

[21] A. Wilson and R. Adams, "Gaussian process kernels for pattern discovery and extrapolation," in *International Conference on Machine Learning*, 2013, pp. 1067–1075.

[22] J. L. Rojo-Álvarez, M. Martínez-Ramón, J. Muñoz-Marí, and G. Camps-Valls, *Digital Signal Processing with Kernel Methods*, 1st ed. Wiley-IEEE Press, 2018.

[23] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[24] N. A. Cressie, "Spatial prediction and kriging," *Statistics for Spatial Data (Cressie NAC, ed). New York: John Wiley & Sons*, pp. 105–209, 1993.

[25] D. Barber, *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012.

[26] L. Bo and C. Sminchisescu, "Greedy block coordinate descent for large scale Gaussian process regression," in *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 2008, pp. 43–52.

[27] I. T. Jolliffe, *Choosing a Subset of Principal Components or Variables*, 2nd ed. New York, NY: Springer, 2002, pp. 111–149.

[28] O. Fujiwara and D. B. Khang, "A two-phase decomposition method for optimal design of looped water distribution networks," *Water resources research*, vol. 26, no. 4, pp. 539–549, 1990.

[29] L. A. Rossman, "EPANET 2: Users manual," US Environmental Protection Agency, Tech. Rep. EPA/600/R-00/057, September 2000.

[30] D. G. Eliades, M. Kyriakou, S. Vrachimis, and M. M. Polycarpou, "EPANET-MATLAB Toolkit: An Open-Source Software for Interfacing EPANET with MATLAB," in *Proc. 14th International Conference on Computing and Control for the Water Industry (CCWI)*, The Netherlands, Nov 2016, p. 8.