Gaussian process regression for monitoring and fault detection of wastewater treatment processes

Oscar Samuelsson, Anders Björk, Jesús Zambrano and Bengt Carlsson

ABSTRACT

Monitoring and fault detection methods are increasingly important to achieve a robust and resource efficient operation of wastewater treatment plants (WWTPs). The purpose of this paper was to evaluate a promising machine learning method, Gaussian process regression (GPR), for WWTP monitoring applications. We evaluated GPR at two WWTP monitoring problems: estimate missing data in a flow rate signal (simulated data), and detect a drift in an ammonium sensor (real data). We showed that GPR with the standard estimation method, maximum likelihood estimation (GPR-MLE), suffered from local optima during estimation of kernel parameters, and did not give satisfactory results in a simulated case study. However, GPR with a state-of-the-art estimation method based on sequential Monte Carlo estimation (GPR-SMC) gave good predictions and did not suffer from local optima. Comparisons with simple standard methods revealed that GPR-SMC performed better than linear interpolation in estimating missing data in a noisy flow rate signal. We conclude that GPR-SMC is both a general and powerful method for monitoring full-scale WWTPs. However, this paper also shows that it does not always pay off to use more sophisticated methods. New methods should be critically compared against simpler methods, which might be good enough for some scenarios.

Key words | Bayesian regression, Gaussian processes, kernel, machine learning, missing data, process monitoring

INTRODUCTION

Monitoring methods are important tools for detecting process and sensor faults in wastewater treatment plants (WWTPs). Faults that influence the treatment process have to be detected and corrected to manage increasingly strict effluent demands. At the same time, automatic process control is an essential part of an efficient treatment process, and is dependent on reliable sensor measurements.

Large amounts of process data have been available at WWTPs for a long time, and the most studied method to exploit the data for monitoring purposes, is principal component analysis (PCA). Although proposed for WWTP applications more than 30 years ago (Aarnio & Minkkinen 1986), PCA is still an active research area (Haimi 2016). Some of the research has been done to extended PCA to handle non-linear interactions in kernel-PCA (Lee et al. 2004), multiple models in Multiway PCA (Yoo et al. 2007), and in combination with other data processing methods (Qin et al. 2012).

Similarly as at WWTPs, the amount of available data has increased in many other areas, especially where the Internet has been a driving force. Discovering that data are not equivalent with useful information, large efforts have been done to develop computer-based algorithms that extract information from big data. This has resulted in a research area called machine learning, which can be defined as methods that can learn and make predictions from data (Kohavi & Provost 1998). Machine learning has successfully been applied for monitoring processes other than WWTPs. Some examples include: Gaussian process regression (GPR) for monitoring the melting index of polyethylene in multigrade process (Liu et al. 2013), deep learning for monitoring product quality in petroleum refinery process (Shang...
et al. 2014), and support vector machines and k-nearest
neighbourhood for classification of faults in a chemical pro-
cess with incomplete data (Askarian et al. 2016). Although
most machine learning methods are not new methods,
they might still be new in a WWTP monitoring context
and, therefore, interesting to study.

GPR is one machine learning method, promising for
monitoring applications at WWTPs. The method is flexible
(can handle non-linear problems), non-parametric (does
not require parameter selection), and probabilistic (predic-
tions include an uncertainty estimate) (Rasmussen &
Williams 2005). In fact, these three properties have proved
to be useful to solve problems related to monitoring includ-
ing change detection (Garnett et al. 2010), process
modelling (Ranjan et al. 2016), and fault detection (Boškoski
et al. 2015). However, few papers have studied GPR applied
to WWTPs (Ažman & Kocijan 2007; Južnić-Zonta et al.
2012; Liu et al. 2016), of which only Liu et al. (2016) studied
how GPR could be used for monitoring purposes. Thus,
GPR has not been studied to the same extent as PCA despite
its promises.

Although GPR is a promising method, it has two poten-
tial drawbacks that could limit its usage in full-scale
applications. First, a proper kernel (parametric autocovar-
iance function) has to be selected by the user, which is a
key step in GPR. This is commonly described as a strength
of GPR, since the user can include prior knowledge of the
system by selecting a proper kernel (Rasmussen &
Williams 2005). However, most papers that apply GPR only use
the standard kernel and do not consider other kernels. Also,
a separate research field has developed on how to construct
new kernels, see e.g. Duvenaud (2014). This indicates that
kernel selection may not be as straightforward as originally
claimed, which could make it hard for a WWTP engineer to
adopt GPR as a new method.

The second potential drawback concerns the estimation
of a kernel’s parameters. Commonly, the parameters are
obtained by maximum likelihood estimation (MLE). How-
ever, this estimation method is also known to give bad
solutions if several local optima exist (non-convex optimiz-
ation problem). Although the risk with local optima has
been acknowledged in both textbooks and papers, few
papers quantify its impact when GPR is applied to real
problems. Any full-scale monitoring method has to be robust
and deliver correct results every time, and it is therefore cru-
cial to evaluate the impact of local optima of GPR in WWTP
applications.

To summarize, we know that GPR is a promising
method that could improve monitoring of WWTPs, but we
lack knowledge of the impact of some potential drawbacks.
Further, we know little about GPR applied to WWTPs and
therefore, also, how good GPR is compared to existing
WWTP monitoring methods.

In this paper, we give a detailed introduction to GPR and
how it can be applied for monitoring applications at WWTPs.
Further, we study whether the potential drawbacks of GPR
can be handled. We introduce the theory of GPR and two var-
ants of GPR: (1) GPR with maximum likelihood estimation
(GPR-MLE), which is the standard approach, and (2) a state-
of-the-art variant of GPR that uses a sequential Monte Carlo
approach for estimation (GPR-SMC) (Svensson et al. 2015).
Both GPR variants use the same model structure, a Gaussian
process (GP), but they approximate and estimate the GP’s par-
rameter distribution differently.

Firstly, we evaluate the potential drawback with local
optima in a simulated case study, where the monitoring problem is
to estimate missing data in a flow rate signal. We also study
whether a priori of the flow rate signal (periodicity) is easily
included in the kernel. For both potential drawbacks, GPR-
SMC and GPR-MLE are compared in the simulated case
study. Secondly, we evaluate the most promising method,
GPR-SMC, on real data with the monitoring task to detect a
drift in ammonium on-line sensor values. As always, new
methods should be critically evaluated and compared against
existing methods. Here, we used linear interpolation in the
first case study and linear regression in the second to assess
the benefit of using GPR compared to existing simple methods.

MATERIAL AND METHODS

GPR combines properties from multivariate Gaussian dis-
tributions with Bayesian statistics to obtain a regression model
that is flexible and give an uncertainty estimate of the predic-
tions. Firstly, we introduce the theory about GPR. Secondly,
we introduce the two GPR variants, GPR-SMC and GPR-
MLE, and their differences in approximating and estimating
the kernel parameter distributions. Thirdly, we give a step-by-
step instruction for applying GPR-MLE and GPR-SMC in prac-
tice. Finally, we describe the two case studies in this paper.

GPs

GPs have been popular for two properties. Firstly, a GP is a
stochastic process that is completely defined by the multi-
variable Gaussian probability distribution. Secondly, the
best unbiased prediction is a linear combination of pre-
iously observed values.
A GP is a distribution of functions with function values \( f(x) \) at inputs \( x \) defined as

\[
f(x) \sim \mathcal{GP}(m(x), k(x_i, x_j)) = \mathcal{N}(m(x), k(x_i, x_j)),
\]

(1)

where \( x \) is size \( n \times m \) with \( n \) inputs and \( m \) variables, and \( k(x_i, x_j) \) expresses the covariance between input \( i \) and \( j \), where \( i, j = [1, \ldots, N] \). Thus, the indices \( i \) and \( j \) are not necessarily time indices, but indicate the different inputs in a specific data set. Note that \( k(x_i, x_i) \) is an \( n \times n \) matrix regardless of the number of variables. In (1),

\[
m(x) \triangleq \mathbb{E}[f(x)]
\]

(2)

is the mean function, and the covariance function is defined as

\[
k(x_i, x_j) \triangleq \mathbb{E}[f(x_i) - m(x_i)][f(x_j) - m(x_j)]^T,
\]

(3)

where \( \mathbb{E}(\cdot) \) denotes the expectation operator.

The best unbiased prediction \( f(x^*) \) at one unobserved \( x^* \), is a linear combination of previous function values \( f(x) \). This holds from the general property that two jointly Gaussian vectors, here \( f(x) \) and \( f(x^*) \)

\[
\begin{bmatrix}
    f(x) \\
    f(x^*)
\end{bmatrix}
\sim \mathcal{N}
\begin{bmatrix}
    m(x) \\
    m(x^*)
\end{bmatrix},
\begin{bmatrix}
    k(x_i, x_j) & k(x_i, x_j^*) \\
    k(x_i^*, x_j) & k(x_i^*, x_j^*)
\end{bmatrix},
\]

(4)

are also conditionally Gaussian distributed, with mean

\[
m_{x^*|x} = k(x_i^*, x_j)^T k(x_i, x_j)^{-1} f(x)
\]

(5)

and covariance matrix

\[
k_{x^*|x} = k(x_i^*, x_j^*) - k(x_i^*, x_j)^T k(x_i, x_j)^{-1} k(x_i^*, x_j).
\]

(6)

This can be written as

\[
f(x^*|x) \sim \mathcal{N}(m_{x^*|x}, k_{x^*|x}),
\]

(7)

where \( m_{x^*|x} \) should be read as the mean of \( x^* \), conditioned on \( x \). In other words, given the function values \( f(x) \), we obtain the predicted mean \( m_{x^*|x} \) as a linear combination of the covariance between observed and unobserved inputs \( k(x_i, x_j^*)^T k(x_i, x_j)^{-1} \), and of the observed function values \( f(x) \). From (6) we also get the variance \( k_{x^*|x} \) of the predicted mean, which is used as an uncertainty measure of the prediction.

One limitation with GPR is that a large \( n \times n \) covariance matrix has to be inverted to obtain the predictions in (5) and (6). The computation time increase exponentially with \( n^3 \), which limits the training data to a few thousand data. Approximations for handling large data have been suggested (Quiñonero-Candela & Rasmussen 2005) but were not considered in this paper.

**Covariance functions**

In machine learning literature, a covariance function \( k_{\theta}(x_i, x_j) \) is called a kernel, where \( \theta \) refers to its parameters. The most common kernel is the squared exponential kernel

\[
k_{\text{Single}}(x_i, x_j) = \theta_1 \exp \left( -\frac{(x_i - x_j)^2}{2\theta_2^2} \right)
\]

(8)

with the parameters \( \theta_1 \) and \( \theta_2 \). Many other kernels have been suggested, see Duvenaud (2014) for a recent work about the properties of different kernels applied to GPR.

**Regression with GPs**

In a general regression problem, we observe data \( y(x) \) and try to describe the underlying function \( f(x) \) that is corrupted by additive noise \( \varepsilon(x) \)

\[
y(x) = f(x) + \varepsilon(x).
\]

(9)

As an example, a linear regression problem would be to estimate \( \theta_1 \) and \( \theta_2 \) in \( f(x) = \theta_1 + \theta_2 x \). The standard method to solve this problem is ordinary least squares (OLS), which is described in statistical textbooks, see e.g. Kay (1998).

In GPR, both \( f(x) \) and \( \varepsilon(x) \) are modelled as two independent GPs, where \( \varepsilon(x) \) is described with a constant noise \( \theta_{\sigma^2} \)

\[
\varepsilon(x) \sim \mathcal{GP}(0, k_{\text{WN}}(x_i, x_j)), \quad k_{\text{WN}}(x_i, x_j) = \theta_{\sigma^2} I_n,
\]

(10)

where \( I_n \) is the identity matrix. Following the law of variances for independent covariance functions, the covariance of the final GP model \( \mathcal{M}_g(x) \) is the sum of the two covariance functions \( k_{\text{Single}}(x_i, x_j) \) and \( k_{\text{WN}}(x_i, x_j) \), therefore we have

\[
\mathcal{M}_g(x) \sim \mathcal{GP}(m(x), k_g(x_i, x_j)),
\]

\[
k_g(x_i, x_j) = k_{\text{Single}}(x_i, x_j) + k_{\text{WN}}(x_i, x_j),
\]

(11)

where the regression task is to estimate the parameters \( \theta \) from a data set \( X, Y \). Here, \( X \) and \( Y \) include all observations \( i = 1, \ldots, n \) for \( x_i \) and \( y_i \). See Perez-Cruz et al. (2013) for a comparison between traditional and Bayesian regression with GPR.

**Bayesian regression**

Bayesian regression relies on Bayes rule which combines prior knowledge \( p(M_g) \), together with the likelihood
The MLE uses the parameter set \( \theta \) in (14) with the maximum likelihood estimate (MLE). The MLE consists of the model \( \mathcal{M}_\theta \), see (11), with an assumed probability distribution of the parameters \( p(\theta) \). For example, a squared exponential kernel (8) as prior model, means that the data in \( X \) and \( Y \) are realizations from (8). The posterior distribution describes the probability of the model with parameters, \( \mathcal{M}_\theta \), given the data, \( \mathcal{D} \), and prior assumptions. Note that the posterior distribution (12) makes it possible to compare the probability of different models with each other, given the same data set.

In this paper, we use two different approximations of (14) and estimation method, namely sequential Monte Carlo (SMC) variant. The parameter estimation of biochemical systems. This approximation has two drawbacks. Firstly, the solution to (15) is a non-convex optimization problem with multiple solutions. Secondly, the point estimate in (15) is a poor estimate for parameter distributions with multiple modes. The impact of the second drawback depends on the optimization problem and cannot be avoided. However, the first drawback could be relieved by using a global optimization (GO) method, instead of a local optimization method. Note that GOs are not guaranteed to find the global optimum within finite time, although they commonly provide better solutions than a pure local search. See Moles et al. (2005) for a comparison of different GOs applied for parameter estimation of biochemical systems.

In this paper, we used a conjugate gradient optimization method implemented in Gaussian Processes for Machine Learning (GPML)-toolbox (Rasmussen & Nickisch 2010) to estimate (15). The method finds a local minimum based on an initial start guess. We used the method because it is the standard optimization method implemented in the well-spread GPML-toolbox for GPR (Rasmussen & Nickisch 2010) and is simple to use. A GO method would be a feasible alternative for optimization problems with multiple local optima, but would also increase the complexity for the user due to tuning efforts of the GO.

GPR-SMC: sequential Monte Carlo variant

In this paper, we also considered a less common approximation of (14) and estimation method, namely sequential Monte Carlo for Gaussian processes (GPR-SMC) described in Svensson et al. (2015). Here, the probability distribution in (14) is approximated as

\[
\hat{p}(y^*|X, Y, \mathcal{M}) = \frac{1}{K} \sum_{k=1}^{K} w^{(k)} p(y^*|X, Y, \mathcal{M}^{(k)}),
\]

where \( w^{(k)} \) are K weights that are normalized as \( \sum_{k=1}^{K} w^{(k)} = 1 \). This approximation differs most notably from (15) due to its use of multiple estimates, in contrast to the single point estimate in (15). The multiple estimates are supplied by a particle filter, which, in short, uses a set of estimates (the particles) with different weighting, to approximate the complete probability distribution in (14).

However, note that both GPR-MLE and GPR-SMC use the same original model structure, the GP, but mainly differ in how the parameter distribution in (14) is approximated and estimated.
One reason for using the more complex estimate (16) is that the predictive GPR model becomes non-parametric, or at least less dependent on the GPR’s parameter values. This should, in theory, also result in a better model in terms of mean and variance prediction of the GPR. The drawback of (16) is that it is computationally demanding (a) when many particles are used (may be required for a good approximation of the parameter distribution) and (b) for high dimensions (many parameter distributions to estimate). It is beyond the scope of this paper to introduce the research area of sequential Monte Carlo methods and particle filters, and Schön et al. (2015) and Gustafsson (2010), respectively, provide thorough introductions to the two research areas.

**Work flow – GPR in practice**

GPR-MLE and GPR-SMC were motivated from comprehensive theory. However, we can apply the methods straightforward once a suitable kernel and parameter limits have been selected. The work flow for applying GPR-MLE and GPR-SMC step-by-step, is given in Table 1.

**Case study 1: missing data in a simulated flow rate signal**

The purpose of the first case study was to a) serve as an introductory example of how GPR can be applied to univariate time series, and b) to compare the standard GPR-MLE with GPR-SMC and simple existing methods in a relevant WWTP monitoring problem. The intention was to compare the methods predictive performances, and at the same time study the impact of the aforementioned two potential drawbacks (local optima, and kernel selection). We chose the introductory problem to estimate missing data in a simulated flow rate signal for several reasons.

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Work flow for applying GPR-MLE and GPR-SMC in practice</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step</td>
<td>GPR-MLE</td>
</tr>
<tr>
<td>1)</td>
<td>Select kernel $k_0(x_i, x_j)$ and training data $X, Y$</td>
</tr>
<tr>
<td>2)</td>
<td>Set initial parameter values and optimization settings</td>
</tr>
<tr>
<td>3)</td>
<td>Run optimization method, find $\theta_{\text{MLE}}$</td>
</tr>
<tr>
<td>4)</td>
<td>Select $x^<em>$ where the distribution of $y^</em>$ will be predicted</td>
</tr>
<tr>
<td>5)</td>
<td>Predict $y^<em>$ with $M_{\text{MLE}}(x^</em>)$</td>
</tr>
</tbody>
</table>

- A simulated case study enables a simple comparison between true and estimated values. In addition, simulations allow for repeated well defined experiments.
- Missing data are a common issue in monitoring applications. Data can be missing due to sensor faults, data transfer issues, outliers or stuck values that have to be replaced (Schraa et al. 2006).
- Well established influent flow generators exist that can generate realistic flow rate signals including periodicities (daily, weekly, and seasonal variations) and stochastic extreme events (rain and snowmelt periods), see e.g. Gernaey et al. (2011).
- The flow rate signal is a representative process variable for WWTPs since the flow is correlated to many measured nutrient concentrations. Thus, a majority of the monitored process variables at WWTPs exhibit similar characteristics such as periodicities and stochastic extreme events.
- A familiar univariate estimation problem is easy to visualize and interpret as an introductory example.

Two common approaches to replace missing data are either to linearly interpolate over the data gap (referred to as Interpolation), or use the last known data point before the data gap as estimate (referred to as Last Value). Here, we used Interpolation and Last Value to assess the benefit of using GPR-MLE and GPR-SMC.

A simulated flow rate signal, resembling 10 years of influent with 15 min sampling time, was generated with the influent generator described in Gernaey et al. (2011) with the default parameter settings. Further, we added Gaussian noise with two different standard deviations:

- Low noise level: $\sigma = 135 \text{ m}^3/\text{h}$. This was equivalent to 0.5 percent of maximum flow rate, which is the recommended maximum repeatability level for measurements in Parshall flumes (Kulin 1984).
- High noise level: $\sigma = 3,000 \text{ m}^3/\text{h}$, which was considered to be a noisy but still realistic flow rate signal.

Data were removed at 100 random positions in both the low and high noise signals. This was repeated four times with gap length of 1, 8, 16, and 24 samples (15 min, 2 h, 4 h, and 6 h), which resulted in a total of eight different faulty signals.

The estimation performance was evaluated as normalized root mean squared error (NRMSE) between the true $y(i)$ and estimated value $\hat{y}(i)$ at each missing data position $i$, normalized with the full signals, $Y$, standard deviation

$$NRMSE = \sqrt{\frac{1}{M} \sum_{i=1}^{M} \left( y(i) - \hat{y}(i) \right)^2} / \text{std}(Y). \quad (17)$$
In (17), \( M \) is the length of the missing data gap times the total number of gaps. Thus, for a signal with gap length 16, \( NRMSE \) was calculated with \( M \) equal to 1,600.

Two kernels were evaluated, as follows.

**Single kernel(s):** the squared exponential kernel in (8). The single kernel is indicated with a lowercase \( s \), GPR-MLEs and GPR-SMCs.

**Combined kernel (c):** a combination of (8) and a periodic kernel

\[
k_{\text{Combined}}(x_i, x_j) = k_{\text{Single}}(x_i, x_j) + \theta_3 k_{\text{Single}}(x_i, x_j) \theta_6^2 \cos \left( \frac{2\pi}{\theta_7} (x_i - x_j) \right).
\]

and the combined kernel is indicated with a lowercase \( c \) as, GPR-MLE-c and GPR-SMC-c.

The combined kernel was used to include our prior knowledge about the periodic characteristics of the flow rate signal. In this first case study, time was the input (regressand) expressed as \( x_i \) at time instant \( i \), and flow rate was the output (regressor) expressed as \( f(x) \).

A uniform distribution was used as prior distribution for parameters in GPR-MLEs and GPR-MLEc. The same interval was used to randomly initialize the optimization for GPR-MLEs and GPR-MLEc, see step 2 in Table 1. Two days before (192 data points) and 2 days after the data gap were used as training data. Thus, for each signal with 100 data gaps, 100 local GPR-MLE and GPR-SMC models were used. Settings for GPR-SMC and prior distributions are given in the supplementary material (available with the online version of this paper).

**Case study 2: detecting abnormal air flow-nitrified ammonium ratios**

The intention of the second case study was to study GPR on a real data set and to study whether GPR could be used for model based fault detection. Here, we used GPR to build a simple process model the by describing the air requirement as a function of nitrification rate, a relationship that is not necessary linear. Thus, the promises of GPR allow us to build a well defined process model based on qualitative process knowledge and process data. Given such a model based on historic data, we could compare new measurement with the model’s predictions and detect deviations in the ammonium measurement. Drift in ammonium on-line sensor measurements have limited the performance of ammonium based controllers, still, the sensors are common in full-scale control applications (Amand 2014). In this second case study, we monitored the accuracy of an ammonium on-line sensor positioned in the pre-sedimentation basin (\( NH_{4\text{in}} \)), with GPR-SMC. We only used GPR-SMC because it had better results than GPR-MLE in case study 1.

The GPR-SMC was used to model the ratio between air flow (output/regressor) and nitrified ammonium (input/regressand) according to

\[
Q_{\text{air}}(t) = f(Q_{\text{flow}}(t) (NH_{4\text{in}}(t) - NH_{4\text{out}}(t + 5))),
\]

where \( Q_{\text{air}} \) was the total air applied to the activated sludge process at time \( t \) [hours]. We can see from (19) that \( Q_{\text{air}} \) is a function of the wastewater flow \( Q_{\text{flow}} \), and the monitored influent and effluent ammonium sensors, \( NH_{4\text{in}} \) and \( NH_{4\text{out}} \), respectively. Thus, the regressor variable, the nitrified amount ammonia, was calculated from three variables although the process model was univariate.

We used delayed measurements of the effluent ammonia concentration to compensate for the time-delay between measured influent and effluent ammonium concentration. We used a time-delay of 5 h, since this gave the highest correlation with the influent ammonia concentration. This was similar to the average hydraulic retention time and thus a reasonable approximation of the WWTPs time dynamics.

We used a linear mean function, \( m(x) \), and a squared exponential kernel combined with a white noise kernel for the covariance function, i.e.

\[
M_{\text{air}/NH_4}(x) \sim GP(m(x), k_0(x_i, x_j))
\]

\[
k_0(x_i, x_j) = k_{\text{Single}}(x_i, x_j) + k_{WN}(x_i, x_j),
\]

where \( x \) was amount of nitrified ammonium, and \( M_{\text{air}/NH_4}(x) \) was the modelled air flow. The reason for the linear mean function was prior knowledge that an increased amount of nitrified ammonium requires an increase in supplied air. However, since the relationship is not necessarily linear but at least suspected to be smooth, the squared exponential kernel seemed like a reasonable choice.

As benchmark method we used a standard linear regression model with two parameters \( \theta_3 \) and \( \theta_4 \) as the regression function \( F(x) \)

\[
F(x) = \theta_3 + \theta_4 x.
\]

The parameter estimates \( \hat{\theta}_3 \) and \( \hat{\theta}_4 \) were obtained by OLS.
In this paper, we used historic process data from Bromma WWTP in Sweden (approximately 300,000 p.e.). The original data set was 2 years with 1 h sampling time, but we reduced the data set for two reasons. Firstly, we were only interested in periods where we were certain that the influent ammonia sensor was either faulty or correct. Secondly, we wanted to restrict the study to stable and warm periods to be able to use the simple process model in (19).

The first data selection requirement was needed to assess the performance of the fault detection methods. In contrast to simulated data, it is problematic to know the true values in full-scale data. Here, we used weekly laboratory analyses to decide whether the influent ammonia sensor showed correct or faulty readings. Although laboratory measurements include uncertainties, this validation procedure is a common approach to validate most online sensors. Since the weekly laboratory samples were sampled volume proportional, we adjusted NH$_{4\text{in}}$ to be flow proportional at the equivalent week and then compared the weekly averages. The 20 weeks with largest and smallest differences in mean were defined as faulty and correct measurements, respectively. We discarded the remaining data since we considered the correctness of influent ammonia sensor to be unclear.

An extended multivariate version of (19) with suspended solids concentration and temperature could potentially cover a broader process conditions. However, in this introductory study we prefer to use a simple univariate model. We refer the interested reader to Ažman & Kocijan (2007) for examples of how GPR can be applied in the multivariate case and to Dolenc et al. (2016) for recent example of GPR applied to fault detection applications. The second selection requirement was selected to maintain a simple process model. The process model in (19) depends on temperature and suspended solids concentration, and is therefore only valid within a specific range. Here, we used stable inflow neglecting heavy rainfalls and snow melting periods ($Q_{\text{flow}}$ below 2.2 m$^3$/h), summer temperatures (temperatures above 15°C), and normal suspended solids concentration (SS) (SS between 3,200 and 3,900 mg/L).

At the end, the two selection requirements resulted in training and test data of about 4 months, compared to the original 2 year large data set. The 4 months data were categorized as: training data (1,200 h, non-faulty), non-faulty test data (1,000 h), and faulty test data (300 h), where the training data came from year one and test data from the second year.

Similarly as it is problematic to use full-scale data to evaluate fault detection evaluation, it may also be problematic to assure that the training data comes without significant errors. In this case study, we validated the training data by using the following approaches:

- NH$_{4\text{out}}$ was verified by laboratory measurements similarly as NH$_{4\text{in}}$. In contrast to NH$_{4\text{in}}$, the ammonia at the effluent was measured by an analyzer which showed good agreement with laboratory analyses.
- $Q_{\text{flow}}$ had no large changes in ratio compared to the summed flow measurements of all parallel flows at the activated sludge basin. Thus, we are confident that no large changes occurred during the 2 years, although the true value might be biased. A constant bias would, however, not affect our suggested monitoring approach.
- Q$_{\text{air}}$ was hard to validate, and significant errors in training data could result in either too many false alarms or a too restrictive fault detector. However, we expected Q$_{\text{air}}$ to be a stable measurement with minor risk to drift since the sensor measures clean air indoors and is unlikely to be affected by e.g. moisture or fouling. Also, previous experience suggests that Q$_{\text{air}}$ measurements are stable towards large drift within 2 years’ time and a constant bias would not affect our suggested monitoring approach.

We used percentage of detection $P_D$, and false alarm probability $P_{FA}$ to evaluate the fault detection performance of GPR-SMC and OLS

\[
P_D = \frac{\# \text{ Alarms}}{\# \text{ Faulty test data}} \quad (23)
\]

\[
P_{FA} = \frac{\# \text{ Alarms}}{\# \text{ Non-faulty test data}} \quad (24)
\]

A fault was detected when a new observation was outside a ±2 standard deviation prediction interval. A Gaussian 2 standard deviations prediction interval should include 95% of new values, given that the process model still applies. We assumed that this was a good trade-off between false alarm rate and detection probability. However, the magnitude of the prediction interval is a user choice that should fit the monitoring purpose. An increase to e.g. 3 standard deviations (which is also a common choice), would give a more restrictive fault detector with lower probability for false alarms with the risk of missing true faults.

**Software**

We used MATLAB for all simulations and calculations. For GPR-MLE and GPR-SMC the GPML-package (Rasmussen
and Nickisch 2010) was used with an add-on toolbox for GPR-SMC (Svensson et al. 2015). We used the influent generator described in Gernaey et al. (2011) to simulate the flow rate signals in case study 1. Computation time on a standard laptop computer was used as indication of computation complexity.

RESULTS AND DISCUSSION

In this section, results are shown and discussed for the two case studies separately.

Case study 1: estimating missing data in a flow rate signal

In this case study, we evaluated whether GPR-MLE and GPR-SMC could be used to estimate missing data in a simulated flow rate signal. The simulated flow rate signal and the missing data estimation methods were described in the methods section.

The lowest NRMSE was obtained by GPR-SMC, and the largest NRMSE was obtained by Last Value in both the low and high noise signals (Figure 1). It was interesting to note the benefit of an advanced method (GPR-SMC) compared to a basic method (Last Value) for estimating missing data, although we expected Last Value to be the worst method.

Interpolation had the shortest mean computation time of about 1 ms, disregarding Last Value that did not involve any computations. GPR-MLE and GPR-SMC were on average 3 and 5 orders of magnitude slower (5 s and 100 s), respectively. The estimation of parameters (GPR-MLE) and approximating the predictive distribution (GPR-SMC) were the reasons for the high computation time. The computation time did not depend on the estimated gap length, but on the size of training data. Computation time was only used as an indication of the different methods’ computational complexity, and was not a comparison between the specific software implementations of GPR-MLE and GPR-SMC.

Although GPR-SMC was the most computationally demanding method, it should be fast enough for an off-line batch data estimation application. For estimating data in the low noise signal, one should use Interpolation since it had comparable performance with GPR-SMC, but was both faster and easier to use (Figure 1(a)).

An increase in the noise level made it more difficult for all methods to estimate missing data (compare Figure 1(a) with 1(b)). Interestingly, the difference in NRMSE between GPR-SMC and Interpolation increased for the high noise signal, compared to the low noise signal. Thus, GPR-SMC was less sensitive to the noise level compared to Interpolation. GPR-MLE and GPR-SMC estimate both the signal and the measurement noise, i.e. the covariance function in (11). As a result, the mean function estimated will be more accurate (given that the noise estimate is accurate) compared to Interpolation, which is by construction noise sensitive.

Both GPR-MLE and GPR-SMC provided consistently good noise estimates that can be used as a quality check when the true measurement noise is known. With unknown measurement noise, the measurement noise estimate is an additional benefit of GPR-MLE and GPR-SMC. Tracking of the noise estimate could be used to detect changes in sensors measurement noise.

GPR-SMC-s gave good estimates of both the true signal (the signal without measurement noise) and missing data in the high noise signal (Figure 2(c)). Likewise, the kernels for GPR-SMC-s were close to the estimated auto covariance function (ACF), (Figure 2(d)). The ACF was estimated from the noisy data. Note that GPR-SMC used several kernels (Figure 2(b), (d)), one for each particle with a specific weight according (16).

In contrast to the good estimates by GPR-SMC-s, GPR-MLE-s gave poor estimates for missing data and the true signal (Figure 2(c)). The poor estimates were caused by a non-informative kernel with bad parameter values (Figure 2(d)), which were obtained from a local minimum during parameter optimization. In this case study, 26% of the optimization rounds of GPR-MLEs (single kernel) got stuck in a local optimum, and less than 1% of the

![Figure 1](https://iwaponline.com/wst/article-pdf/75/12/2952/452819/wst075122952.pdf)
optimization of GPR-MLEc (combined kernel). This was the reason why GPR-MLEs also had a worse NRMSE than GPR-SMCs (compare GPR-MLEs and GPR-SMCs in Figure 1). Also note that GPR-MLEc and GPR-SMC have similar NRMSE and overlap in Figure 1. We expected a kernel with many parameters, such as GPR-MLEc, to be more likely to be stuck in local optima, than GPR-MLEs. But as the results indicate, for this specific data set, the combined kernel suffered less than the squared exponential kernel from local optima.

In summary, GPR-SMC performed better than GPR-MLE because it could better account for local optima, for the two different kernels. However, it is not clear from the results whether this was an effect of the particle sampling of (16), or due to the non-parametric approximation used in GPR-SMC. In addition, we did not use a GO method in GPR-MLE that potentially could have handled the local optima in a better way than the local optimization method in the GPML-toolbox. Thus, a comparison between GPR-SMC and GPR-MLE with a GO would be an interesting future study.

It was not beneficial to use the combined kernel given in (18), compared to the standard squared exponential kernel given in (8). GPR-SMCs and GPR-SMCC had similar NRMSE (Figure 1) although the computation time was twice as long for the combined kernel as for the single kernel (126 s for GPR-SMCC and 70 s for GPR-SMCs). The reason for the equal estimation performance was that the periodic part of the combined kernel was given weights close to zero (see $\theta_3$ in (18)). In effect, we obtained the same results for both the combined and the single kernel.

A nice feature of GPR-SMC was its ability to do the trade off in model complexity automatically. In general, a too flexible model would overfit, which did not happen in this experiment. However, one should use the simplest model that serves the purpose, and in this experiment the squared exponential kernel was a good choice both for GPR-SMC and GPR-MLE.

Both GPR-MLE and GPR-SMC provide an uncertainty estimate of the mean value predictions, which is given by the covariance in (6). The uncertainty estimate is given as prediction interval with $\pm 2$ standard deviations for GPR-SMCs in Figure 2(a) and 2(c) by the thin solid black lines. The total covariance is the sum of the estimated measurement noise and the covariance added from the kernel (6). We can see that the maximum covariance of a prediction is obtained if the second term in (6) is zero. This corresponds to the lag distance when the kernel goes to zero (see arrows at x-axis in Figure 2(b) and 2(d)). Thus, the uncertainty estimate attains its maximum value when no information is used by adjacent points for the predictions. At the same time, the prediction approaches the mean function value which, in this study, was the mean value of the signal. Both effects can be seen in Figure 2(a) where the predictions (solid black line) stretch towards the mean function value (dotted dashed line in Figure 2(a), indicated with arrow), and the covariance bounds increase at the missing data gap. It is important to remark that the covariance
bound is not a measure of agreement between true (unknown) signal value and predicted value, but a prediction interval that we expect to contain the signal.

**Case study 2: detecting anomalous air flow–nitrified ammonium ratios**

The test data included a time period with two similar trends were the first one was caused by a drift in ammonium on-line sensor, and the second trend was a true decrease in ammonium concentration (Figure 3). GPR-SMC successfully detected the drift in the ammonium on-line sensor and had few false alarms during the non-faulty test period. In the full test data set, GPR-SMC had 74.5% detection percentage and 15.9% false alarm percentage at a ± 2 standard deviation prediction interval. The size of the prediction interval is a user choice, and could be increased in order to, e.g., decrease the false alarm percentage.

We used the OLS solution to the linear regression model in (22) as the benchmark method, to evaluate the benefit of using GPR. Surprisingly, both the detection percentage and false alarm percentage were equivalent to GPR-SMCs results. The reason for this was that the air flow–nitrified ammonium ratio was almost linear in the range 18–45 mg/L, which contained the majority of test data (Figure 4). Thus, we did not benefit from GPR-SMCs flexibility in this specific case study.

The magnitudes of the prediction intervals were rather constant for both OLS and GPR-SMC (Figure 4). GPR is commonly motivated by its data dependent confidence bounds, more exact the prediction intervals (Rasmussen & Williams 2005). In this case study, they did not vary much because of two reasons. Firstly, the variance due to measurement noise was much larger than the variance contribution from the kernel. Thus, the variance from measurement noise would hide potential changes in variance from the kernel. Secondly, we did not have a change in the predicted variance since we had many data. It was only for predictions far from training data (below 15 g/s and above 55 g/s) that the variance in GPR-SMC increased due to distance to data.
In future studies, we would like to study a regression problem with more profound static non-linearity to see if GPR-SMC would provide better results than linear regression. Further, such study should also be multivariate to assess GPR-SMC's performance in larger dimensions with multiple parameter distributions to estimate.

CONCLUSIONS

This paper gives new insights into the strengths and weaknesses of two variants of the machine learning method GPR, GPR-MLE and GPR-SMC. The methods were applied to two WWTP monitoring problems.

The results in the simulated missing data flow rate case, showed that GPR-SMC had better performance than the standard GPR-MLE, where GPR-MLE suffered from getting stuck in local minima during kernel parameter estimation. The local optima were mainly a problem for GPR-MLE when the single kernel was used, and indicates that the standard GPR-MLE with a standard kernel was not satisfactory for the specific case to estimate missing data in a flow rate signal. In contrast, GPR-SMC had no problems with local optima and gave at the same time better predictions than linear interpolation. In the second case study with real data, GPR-SMC was as good as linear regression, but this was because the data in the regression problem had an almost linear relationship.

It was not straightforward to include prior knowledge in the kernel, as commonly claimed in e.g. Rasmussen & Williams (2005). Although an influential flow rate signal is clearly periodic, the combined kernel with a periodic part, did not give better results than the single kernel. Thus, we argue that the common claim that prior knowledge easily can be used by selecting a proper kernel is idealized. We believe that efficient usage of GPR in full-scale applications requires thorough understanding about kernels. In this paper, we show that a graph of the autocovariance with respect to time lags can be useful to interpret the impact of different kernels. This somewhat demystifies the kernel and simplifies interpretation of the results.

We conclude that GPR-SMC is both a general and powerful method for monitoring full-scale WWTPs. It did not suffer from local optima in either the simulated or the full-scale case study and gave good results with both kernels.

ACKNOWLEDGEMENTS

The research leading to these results has received funding from Käppala Association, Syvab and Stockholm Water, Foundation for IVL Swedish Environmental Research Institute and the Swedish Water and Wastewater Association, which is gratefully acknowledged. Jesús Zambrano has received funding from the Knowledge Foundation (20140168) under the project More Sense, ABB, Mälarenergi, Flexiclean and Structor Miljöteknik.

We gratefully acknowledge Dr Erik Lindblom for providing data and valuable discussions in case study 2, and Andreas Svensson for providing the software implementation of GPR-SMC and valuable discussions about its usage. Finally, we thank Krist Gernaey for providing access to the influential flow rate simulator.

REFERENCES


First received 23 November 2016; accepted in revised form 3 March 2017. Available online 25 March 2017