Dynamic modeling and optimization of sustainable algal production with uncertainty using multivariate Gaussian processes

Eric Bradford[a], Artur M. Schweidtmann[b], Dongda Zhang[c,d,*], Keju Jing[e], Ehecatl Antonio del Rio-Chanona[c,**]

[a]*Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway*
[b]*Aachener Verfahrenstechnik Process Systems Engineering, RWTH Aachen University, Aachen, Germany*
[c]*Centre for Process Systems Engineering, Department of Chemical Engineering, Imperial College London, London, England*
[d]*Centre for Process Integration, School of Chemical Engineering and Analytical Science, University of Manchester, Manchester, England*
[e]*Department of Chemical and Biochemical Engineering, College of Chemistry and Chemical Engineering, Xiamen University, Xiamen, China*

Abstract

Dynamic modeling is an important tool to gain better understanding of complex bioprocesses and to determine optimal operating conditions for process control. Currently, two modeling methodologies have been applied to biosystems: kinetic modeling, which necessitates deep mechanistic knowledge, and artificial neural networks (ANN), which in most cases cannot incorporate process uncertainty. The goal of this study is to introduce an alternative modeling strategy, namely Gaussian processes (GP), which incorporates uncertainty but does not require complicated kinetic information. To test the performance of this strategy, GPs were applied to model microalgae growth and lutein production based on existing experimental datasets and compared against the results of previous ANNs. Furthermore, a dynamic optimization under uncertainty is performed, avoiding over-optimistic optimization outside of the model's validity. The results show that GPs possess comparable prediction capabilities to ANNs for long-term dynamic bioprocess modeling, while accounting for model uncertainty. This strongly suggests their potential applications in bioprocess systems engineering.

Keywords: Optimization under uncertainty; Gaussian Process; Artificial neural network; Machine Learning; Bioprocess

1. Introduction

The synthesis of sustainable bioproducts from microalgae through photosynthetic related metabolic pathways has become a promising research field because of its outstanding advantages over traditional fossil fuel based processes (Brennan and Owende, 2010; Zhang et al., 2015a). Specifically, in the energy and food sectors the development and deployment of microalgae based technologies have seen substantial interests within the last decade (Kuddus et al., 2013; Mata et al., 2010). For example, these emerging technologies represent a variety of promising alternatives for the next generation of renewable and environmentally friendly transportation fuels such as biodiesel and biohydrogen (Tamburic et al., 2011; Adesanya et al., 2014). Meanwhile, they have been recently adopted by different countries such as the United States, China and Mexico to produce nutritious food supplements and animal feeds, of which the global market has been predicted to undergo considerable growth (Chu, 2012; Zhang et al., 2015b). Furthermore, they have been successfully

*Corresponding author: dongda.zhang@manchester.ac.uk
**Corresponding author: a.del-rio-chanona@imperial.ac.uk

industrialized to produce different high-value bioproducts that are widely used in the cosmetic, pharmaceutical and food industries (*e.g.* lutein, C-phycocyanin and astaxanthin), which are otherwise produced from expensive, energy intensive and low efficient manufacturing routes using non-renewable sources (Fábregas et al., 2001; Sun et al., 2015).

In particular, the biorenewable product investigated in the current study is lutein, which is of great interest to the health, pharmaceutical, and food sectors. In the United States the demand of lutein is predicted to increase from $150 million in 2000 to $309 million in 2018, with an annual growth rate of over 6% up to 2024 (Ho et al., 2015; del Rio-Chanona et al., 2017b). However, the current feedstock for lutein production is marigold, a plant with extremely low lutein content (0.02-0.1% wt (fresh flowers)) and low growth rate requiring large separation costs (Yen et al., 2011). A promising alternative to produce lutein is from microalgae due to their rapid growth rate, higher lutein content (up to 0.5% wt) and capability of utilizing abundant sustainable resources such as solar energy, atmospheric CO_2 and waste water for their growth and product synthesis (Xie et al., 2013).

To drive the industrialization of sustainable lutein production from microalgae, a robust mathematical model is required for precise process control over a long-term time horizon, so that both process safety and efficiency can be guaranteed. Moreover, by utilizing state-of-the-art process optimization strategies for mathematical models, dynamic optimization can be further carried out to increase process profitability (del Rio-Chanona et al., 2015). As bioprocesses are in general sensitive to changes of operating conditions, it is expected that by implementing dynamic optimization a significant improvement on product yield can be obtained compared to the recent literature results (Malek et al., 2016; Xie et al., 2013; del Rio-Chanona et al., 2016a).

So far, two modeling methodologies have been employed to simulate dynamic behavior of the underlying biosystem for microalgal biomass growth and lutein synthesis, namely kinetic modeling and artificial neural networks (ANN) (García-Camacho et al., 2016; del Rio-Chanona et al., 2016a). In this paper, however, a third methodology, Gaussian process (GP) regression, is proposed to simulate this system and compared against the previous two, so that its feasibility and capability for bioprocess modeling can be thoroughly explored for the first time. Recently, GPs have become an increasingly popular non-parametric method for both regression and classification problems (Rasmussen and Williams, 2006). GP regression was first proposed by O'Hagan and Kingman (1978) and then popularized in Neal (2012).

The GP regression framework not only provides a prediction for unknown outputs, but also provides a measure for prediction uncertainty, which is a distinct advantage compared to other commonly used black-box methods. Furthermore, in Rasmussen (1996) it was shown that GPs are able to forecast outputs with comparable performance to other modeling approaches like ANNs or local learning methods. GPs have been shown to be a powerful tool for derivative-free optimization, since the uncertainty measure can be exploited to evaluate functions more efficiently for both single-objective optimization (Jones et al., 1998; Shahriari et al., 2016) and multi-objective optimization (Bradford et al., 2018). Although GPs have been predominantly used to model static nonlinearities, it is notable that they have also been demonstrated and applied to simulate dynamic systems in several studies (Kocijan et al., 2005; Brahim-Belhouari and Bermak, 2004; Girard et al., 2003; Urtasun et al., 2006; Wang et al., 2005; Bradford and Imsland, 2018). Particularly for studies of long-term bioprocess modeling and optimization, despite the fact that GPs have never been adopted in this domain, they are expected to possess two outstanding advantages over the most commonly used methods (*i.e.* kinetic model and ANN), which are:

1. Compared to ANNs, GPs provide a clear measure of prediction uncertainty which is crucial when modeling complex biological systems. In addition, although ANNs might be preferred over GPs in some cases (*e.g.* when there is a large amount of training data) because of the matrix inversion that is required for the construction of GP predictions, in macro-scale bio-manufacturing studies it is infeasible to obtain datasets in the order of hundreds of thousands or millions. Hence, GPs are clearly a comparable or even superior tool to ANNs in this domain.
2. Compared to kinetic models, GPs do not need a full understanding of the complex metabolic mechanisms that take place in the specific biosystem. Specific to bioprocess applications, in most cases, intensive collaborative effort of the scientific community is required to identify the essential biochemical kinetic information before a kinetic model is ready to be constructed. On the contrary, GPs are black-box models that can be used to simulate and optimize processes in the early stage of research, hence efficiently forwarding the assessment and prototyping stages for process design and scale-up.

As a result, in the current research GPs are set as the default black-box modeling strategy for algal lutein production. Given that for bioprocesses there is a need to predict long time horizons (in the order of days), two approaches have been proposed in literature to accomplish this (del Rio-Chanona et al., 2017b):

One approach is to train a model on **all** control inputs and the initial state to obtain predictions at the full-time horizon. While this approach is easy-to-implement, it has several considerable disadvantages. Once trained, the model cannot be extended to make predictions at time horizons with different lengths. In addition, for large time horizons the number of inputs quickly becomes large, requiring a high dimensionality of the GP to be learnt and hence too many data-points. Alternatively, the iterative method can be used, which trains a GP to predict one-step ahead and applies this GP recursively to obtain a prediction for the full time horizon. The iterative method has several advantages compared to one-step ahead predictions. It can be easily used for different time horizons with different lengths and provide any k-step ahead forecast including the joint probability distribution of the states at the desired points. In addition, the dimensionality of GPs is much smaller in the iterative method when control inputs are present and therefore more data efficient.

In this paper, a general procedure to execute dynamic modeling using GPs is outlined. For this the *iterative method* was selected and applied to predict the evolution of multivariate states for lutein production. In Girard et al. (2003) it was shown that the noise in the one-step ahead prediciton needs to be propagated to be conservative enough. Therefore, we propagated the resulting probability densities of the GPs using exact moment matching (by moments we refer to the mean and covariance) for the squared exponential covariance function. The methodology exhibited in Kocijan et al. (2005) given for single variate state systems was extended to the multivariate case by using results from machine learning (*e.g.* reinforcement learning) in Deisenroth and Rasmussen (2011).

The paper is structured as follows. In Section 2 we introduce the reader to Gaussian processes. Section 3 outlines the experimental set-up of the algal lutein production process with various operating conditions and shows how this data can be used to build GPs for the dynamic modeling of biosystems. In Section 4 a description of the recently constructed ANNs for comparison to the GP is given. In the Results and Discussion section (Section 5) the GP regression results are presented and compared against the ANNs. Meanwhile, a dynamic optimization with stochastic constraints was performed in this section to maximize lutein yield by varying flow rate and light intensity, while taking advantage of the probabilistic nature of the GP.

2. Introduction to Gaussian process regression

2.1. Multivariate Gaussian distribution

To explain the principle of the GP modeling framework, we need to first introduce some concepts commonly used in probability theory. A random variable follows a univariate Gaussian distribution if its probability density function is given by Equation 1, where $x \in \mathbb{R}$ is the result of a single test. A Gaussian distribution is defined by its mean μ (expectation), and its variance σ^2. This is generally written as $x \sim \mathcal{N}\left(\mu, \sigma^2\right)$.

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{1}$$

Let us generalize this definition into higher dimensions. For a given random n-dimensional vector $\mathbf{x} = [x_1, ..., x_n]^T$ with mean $\boldsymbol{\mu} = [\mu_1, ..., \mu_n]^T$, its covariance matrix $\boldsymbol{\Sigma} = \mathrm{cov}(\mathbf{x})$ is defined as a $n \times n$ matrix of which the entry at the i^{th} row and j^{th} column is calculated by Equation 2, where μ_i and μ_j denote the expectation of x_i and x_j, respectively. In general, any symmetric positive semidefinite matrix can be used as a covariance matrix. This necessitates the diagonal elements to be non-negative, since these are variances. The vector $\mathbf{x} \in \mathbb{R}^n$ is said to have a multivariate Gaussian distribution if every linear combination of its components $(x_1, ..., x_n)$ is a univariate Gaussian distribution. The probability density function of this multivariate Gaussian distribution is written as Equation 3, and commonly denoted as $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the mean and covariance matrix of these random variables, respectively. It is worth mentioning that the covariance $\mathrm{cov}\left(x_i, x_j\right)$ is a measure of the correlation between the component x_i and the component x_j. Thus, if these components are independent, the covariance coefficient becomes 0. For example, a standard multivariate Gaussian distribution is defined such that each component is independent, the mean $\boldsymbol{\mu} = \mathbf{0}$ and the covariance matrix is given by $\boldsymbol{\Sigma} = \mathbf{I}_{n\times n}$.

$$\operatorname{cov}(x_i, x_j) = E\left[(x_i - \mu_i)(x_j - \mu_j)\right] \tag{2}$$

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{n/2} \mid \mathbf{\Sigma} \mid^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \mathbf{\Sigma} (\mathbf{x} - \boldsymbol{\mu})\right) \tag{3}$$

where $|\cdot|$ denotes the determinant

There are two important identities for multivariate Gaussian distributions, which are essential in GP regression. Let us assume that we are given a joint vector of $\mathbf{x}$ and $\mathbf{y}$ that are distributed as a multivariate normal distribution as shown in Equation 4.

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \sim \mathcal{N}\left(\begin{matrix} \boldsymbol{\mu}_{\mathbf{x}} \\ \boldsymbol{\mu}_{\mathbf{y}} \end{matrix}, \begin{bmatrix} \mathbf{\Sigma}_{\mathbf{x}} & \mathbf{\Sigma}_{\mathbf{x},\mathbf{y}} \\ \mathbf{\Sigma}_{\mathbf{y},\mathbf{x}} & \mathbf{\Sigma}_{\mathbf{y}} \end{bmatrix} \right) \tag{4}$$

The marginal distribution of $\mathbf{x}$ (the distribution of $\mathbf{x}$ alone) is then given by Equation 5.

$$\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{x}}, \mathbf{\Sigma}_{\mathbf{x}}) \tag{5}$$

The conditional distribution of $\mathbf{x}$ given $\mathbf{y}$ can be denoted by Equation 6.

$$\mathbf{x}|\mathbf{y} \sim \mathcal{N}\left(\boldsymbol{\mu}_{\mathbf{x}} + \mathbf{\Sigma}_{\mathbf{x},\mathbf{y}} \mathbf{\Sigma}_{\mathbf{y}}^{-1} (\mathbf{y} - \boldsymbol{\mu}_{\mathbf{y}}), \mathbf{\Sigma}_{\mathbf{x}} - \mathbf{\Sigma}_{\mathbf{x},\mathbf{y}} \mathbf{\Sigma}_{\mathbf{y}}^{-1} \mathbf{\Sigma}_{\mathbf{y},\mathbf{x}}\right) \tag{6}$$

where $\mathbf{x}|\mathbf{y}$ denotes the probability distribution of $\mathbf{x}$ given that we know the value of $\mathbf{y}$. Therefore, the distribution on the right-hand side of Equation 6 has lower variances (the diagonal elements of the covariance matrix) than the marginal distribution of $\mathbf{x}$ in Equation 5, since we are exploiting the knowledge that the value of $\mathbf{y}$ gives us on $\mathbf{x}$.

2.2. Introduction to Gaussian process regression

In this section we give a short introduction to GP regression. For a more detailed and complete overview please refer to Rasmussen and Williams (2006); Ebden (2015); Jones et al. (2001). GPs generalize multivariate Gaussian distribution to infinite dimensions defining a functional space and hence describe a distribution over infinite dimensional vector functions. Formally, a GP is a collection of random variables of which any finite subset follows a Gaussian distribution. GP regression aims to model an unknown latent function $f(\mathbf{x})$ using noisy observations y of $f(\mathbf{x})$, which are related as follows:

$$y = f(\mathbf{x}) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2) \tag{7}$$

where $\mathbf{x} \in \mathbb{R}^n$ denotes an arbitrary input vector and ϵ is Gaussian distributed measurement noise with a variance of σ_ϵ^2.

Assume we want to make a prediction of $f(\mathbf{x})$ at some arbitrary input $\mathbf{x}$. Before we have sampled the function at this point, *i.e.* before we obtain observations of $f(\mathbf{x})$ at this input $\mathbf{x}$, this value will be uncertain. For GPs we model this uncertainty of the value of the function at $\mathbf{x}$ as the realization of a normally distributed random variable $f(\mathbf{x})$ with mean μ and variance σ^2. Intuitively, we are assuming the function value at $\mathbf{x}$ to have a typical value of μ, which can be expected to lie with a probability of 99.7% in the range $[\mu - 3\sigma, \mu + 3\sigma]$. The mean μ of $f(\mathbf{x})$ in the most general case may be given by an arbitrary function $m(\mathbf{x})$, which defines the "average" shape of the function.

To define a covariance function, we consider two arbitrary input vectors $\mathbf{x}$ and $\mathbf{x}'$, which again have not been sampled and consequently the values of the function at these points are uncertain. However, if we assume the unknown function, which we wish to model, to be continuous, then the function values $f(\mathbf{x})$ and $f(\mathbf{x}')$ will be close if the distance between $\mathbf{x}$ and $\mathbf{x}'$ is small. This prior information can be modeled statistically by assuming that the random variables $f(\mathbf{x})$ and $f(\mathbf{x}')$ are strongly correlated if the distance $||\mathbf{x} - \mathbf{x}'||$ is small. Correlation means that $f(\mathbf{x})$ will tend to be large if $f(\mathbf{x}')$ is large, as long as $\mathbf{x}$ and $\mathbf{x}'$ are close together. On the other hand if $\mathbf{x}$ and $\mathbf{x}'$ are far apart, then the values of $f(\mathbf{x})$ and $f(\mathbf{x}')$ are virtually independent. In particular, in this paper we assume the correlation between the random variables is given by the squared-exponential (SE) covariance function, which is a stationary covariance function. A stationary covariance function is a function of $\mathbf{x} - \mathbf{x}'$, such that the covariance function is translation invariant

$k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x} - \mathbf{x}', \mathbf{0})$. The SE covariance function can be defined as follows (Rasmussen and Williams, 2006):

$$\text{cov}_f\left(f(\mathbf{x}), f(\mathbf{x}')\right) = \mathbb{E}_f\left(\left(f(\mathbf{x}) - m(\mathbf{x})\right)\left(f(\mathbf{x}') - m(\mathbf{x}')\right)\right) = k(\mathbf{x}, \mathbf{x}') = \alpha^2 \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{x}')^T \mathbf{\Lambda} (\mathbf{x} - \mathbf{x}')\right) \quad (8)$$

where $\mathbf{x}$,$\mathbf{x}'$ are arbitrary inputs, $\mathbf{\Lambda} = \text{diag}([\lambda_1^{-2}, \ldots, \lambda_n^{-2}])$ is a diagonal matrix with a length scale λ_i for each input and α^2 is the signal variance. $\mathbb{E}_f$ is the expectation over the function space. The mean function can be viewed as the 'typical' shape of the function, while the covariance function specifies the covariance between any two function values at two separate inputs. The SE covariance function is such that if $\mathbf{x} = \mathbf{x}'$ then $k(\mathbf{x}, \mathbf{x}') = \alpha^2$ reaches the maximum; while if $||\mathbf{x} - \mathbf{x}'|| \to \infty$ the correlation tends to zero as required. The parameters λ_i determine how fast the correlation tends to zero as one moves in the i^{th} dimension of the input vector. Small values of λ_i model functions that are significantly dependent on the i^{th} dimension, *i.e.* the function value can rapidly change when varying the i^{th} variable of such functions. Conversely large values of λ_i lead to functions that are close to invariant with respect to the i^{th} variable. In addition, the SE covariance function not only assumes the latent function to be continuous, but also smooth since it is infinitely differentiable.

A GP generalizes the Gaussian distribution to infinite dimensions and describes a distribution over functions. It is fully specified by a mean function $m(\mathbf{x})$ and a covariance function $k(\mathbf{x}, \mathbf{x}')$. We write f is distributed as a GP as follows:

$$f(\mathbf{x}) \sim GP(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \quad (9)$$

The noisy observations y also follows a GP due to the additive property of Gaussian distributions with the same mean, but with a different covariance function to account for the measurement noise:

$$y \sim GP(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}') + \sigma_\epsilon^2 \delta(\mathbf{x}, \mathbf{x}')) \quad (10)$$

where $\delta(\mathbf{x}, \mathbf{x}') = 1$ **iff** $\mathbf{x} = \mathbf{x}'$ and else $\delta(\mathbf{x}, \mathbf{x}') = 0$, known as the Kronecker-delta.

Equations 9 and 10 define the prior of the function, since no data has been used yet. Afterwards this prior is updated using input-output data available from observations. For GP regression we need to first define the prior of the GP by choosing the mean function and covariance function, which encapsulate our prior beliefs, if available, about the function to be modeled. In this report we assume a mean function of zero as given in Equation 11, which is a common choice in Machine Learning (Rasmussen and Williams, 2006). A zero-mean of the data is achieved in this report by scaling the data. In essence this means that we are assuming the function to be overall zero mean, such as a sine function.

$$m(\mathbf{x}) = 0 \quad (11)$$

We assume the covariance function to be given by the SE defined in Equation 8. As previously described this encapsulates our belief that the function to be modeled is smooth.

Next we assume that N observations are available at N different inputs given by the following two quantities:

$$\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N], \qquad \mathbf{y} = [y_1, \ldots, y_N]^T \quad (12)$$

We can then represent the uncertainty of n function values based on the prior from the mean and covariance functions with the help of a random vector $\mathbf{F} = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N)]^T$ at n separate input vectors given by the matrix $\mathbf{X}$. This random vector has a mean vector defined as $\mathbf{0}$ and a covariance matrix equal to:

$$\mathbf{\Sigma_F} = [k(\mathbf{x}_i, \mathbf{x}_j)]_{N \times N} \quad (13)$$

where $\mathbf{\Sigma_F}$ is a $N \times N$ matrix with (i, j) element given by Equation 8.

Equation 13 gives us the covariance matrix for the latent function values. We however observe y and not $f(\mathbf{x})$, which is perturbed by Gaussian distributed measurement noise with a variance of σ_ϵ^2 as shown in

Equation 7. The uncertainty of the observation matrix $\mathbf{y}$ can then be expressed in the same way as $\mathbf{F}$ with a mean function of $\mathbf{0}$ and a covariance matrix given by:

$$\mathbf{\Sigma_y} = \left[k\left(\mathbf{x}_i, \mathbf{x}_j\right) + \sigma_\epsilon^2 \delta(\mathbf{x}_i, \mathbf{x}_j)\right]_{N\times N} \tag{14}$$

The hyperparameters defining the prior GP are commonly unknown *a priori*, and hence an important step in GP regression is the determination of the hyperparameters from the available data. The hyperparameters that define the GP are given by the parameters of the covariance function in Equation 8 and by the noise of y in Equation 7. These are jointly denoted by the vector $\mathbf{\Theta} = [\log(\lambda_1), \ldots, \log(\lambda_n), \log(\alpha), \log(\sigma_\epsilon)]^T$, where the parameters were log-transformed to ensure positiveness. The hyperparameters of the GPs in this study were efficiently found using a *maximum a posteriori* (MAP) estimate, which is more efficient for smaller data sets than the more commonly used *maximum likelihood* (ML) approach (Rasmussen and Williams, 2006). This is due to the prior introduced in MAP preventing overfitting compared to ML (Sundararajan and Keerthi, 2001). In this work, we assume independent Gaussian distributions on the hyperparameters:

$$\Theta_j \sim \mathcal{N}(\mu_{\Theta_j}, \sigma^2_{\Theta_j}) \tag{15}$$

where μ_{Θ_j} is the mean and $\sigma^2_{\Theta_j}$ the variance of the prior Gaussian distribution for the hyperparameter Θ_j.

Following from the uncertainty expression of the data as multivariate Gaussian distribution with covariance matrix as in Equation 14 and the prior distribution of the hyperparameters in Equation 15, the log-likelihood of the posterior density of the hyperparameters can be stated as follows (Rasmussen and Williams, 2006):

$$\begin{aligned}\mathcal{L}(\mathbf{\Theta}) = &-\frac{1}{2}\log(|\mathbf{\Sigma_y}|) - \frac{1}{2}\mathbf{y}^T\mathbf{\Sigma_y^{-1}}\mathbf{y} - \frac{N}{2}\log(2\pi) + \\ &\sum_j(-\frac{1}{2}\log(2\pi) - \frac{1}{2}\log(\sigma^2_{\Theta_j}) - \frac{1}{2\sigma^2_{\Theta_j}}(\Theta_j - \mu_{\Theta_j})^2)\end{aligned} \tag{16}$$

Notice that function $\mathcal{L}(\mathbf{\Theta})$ is still a function of the training targets $\mathbf{y}$, and hence the difference between the predicted outputs and the target outputs (the $\mathbf{y}$s) will be minimized in a similar fashion as would happen with an ANN training framework. The covariance matrix $\mathbf{\Sigma_y}^{-1}$ can be efficiently factorized using Cholesky decomposition, since it is a symmetric positive semidefinite matrix. Once the hyperparameters are fixed, the inverse matrix product can then be solved efficiently. Since the elements of the covariance are however nonlinear functions of the hyperparameters, the factorization needs to be recalculated for each iteration of these.

The MAP estimate of $\mathbf{\Theta}$ is then given by:

$$\mathbf{\Theta}_{MAP} \in \arg\max_{\mathbf{\Theta}} \mathcal{L}(\mathbf{\Theta}) \tag{17}$$

Once we have calculated the MAP estimate of $\mathbf{\Theta}$, we can use GPs to predict the value of $f(\mathbf{x})$ and y at unknown inputs. First consider the joint distribution of the data and the function value $f(\mathbf{x})$, which can be established using the mean and covariance function of the prior:

$$\begin{bmatrix} f(\mathbf{x}) \\ \mathbf{y} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \Sigma_f & \mathbf{\Sigma}_{f,\mathbf{y}} \\ \mathbf{\Sigma}_{\mathbf{y},f} & \mathbf{\Sigma_y} \end{bmatrix}\right)$$

where $\mathbf{\Sigma_y}$ is given in Equation 14, $\Sigma_f = k(\mathbf{x}, \mathbf{x})$, $\mathbf{\Sigma}_{f,\mathbf{y}} = [k(\mathbf{x}, \mathbf{x}_1), \ldots, k(\mathbf{x}, \mathbf{x}_N)]$ and $\mathbf{\Sigma}_{\mathbf{y},f} = \mathbf{\Sigma}^T_{f,\mathbf{y}}$.

As set out at the beginning, we want to know the value of $f(\mathbf{x})$ given the data available to us, which is represented by the random vector $\mathbf{y}$. We can now apply the identity given in Equation 6 that gives us the distribution of $f(\mathbf{x})$ given the observations of $\mathbf{y}$, which can be stated as follows:

$$f(\mathbf{x})\,|\mathbf{y} \sim \mathcal{N}\left(\mathbf{\Sigma}_{f,\mathbf{y}}\mathbf{\Sigma_y^{-1}}\mathbf{y}, \Sigma_f - \mathbf{\Sigma}_{f,\mathbf{y}}\mathbf{\Sigma_y}^{-1}\mathbf{\Sigma}^T_{f,\mathbf{y}}\right) \tag{18}$$

The mean in this case is the best estimate of $f(\mathbf{x})$ given the data available, while the variance gives us a measure of uncertainty to this estimate. To obtain the posterior of the observation of $f(\mathbf{x})$ we simply need

to add the observation noise to the variance:

$$y|\mathbf{y} \sim \mathcal{N}\left(\mathbf{\Sigma}_{f,\mathbf{y}}\mathbf{\Sigma}_{\mathbf{y}}^{-1}\mathbf{y}, \Sigma_f - \mathbf{\Sigma}_{f,\mathbf{y}}\mathbf{\Sigma}_{\mathbf{y}}^{-1}\mathbf{\Sigma}_{f,\mathbf{y}}^{T} + \sigma_\epsilon^2\right) \tag{19}$$

where y is the observation of $f(\mathbf{x})$ according to Equation 7.

We have now shown how GPs can be used to obtain predictions at arbitrary inputs. The overall procedure involves these three steps:

1. Choose mean and covariance function depending on the prior knowledge of the underlying function.
2. Determine the hyperparameter values by maximizing likelihood using observations of the underlying function.
3. Make predictions at arbitrary inputs using Equations 18 and 19, where the mean represents the prediciton and the variance the corresponding uncertainty.

An example of GP regression can be seen in Figure 1 with the prior and posterior shown.

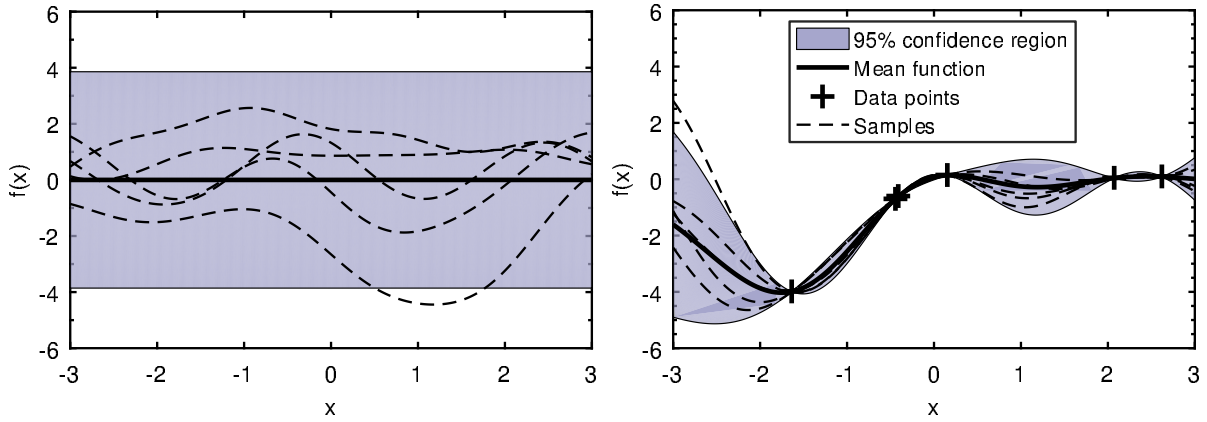

Figure 1: Illustration of a GP of a 1-dimensional function perturbed by noise. On the left the prior of the GP is shown with mean 0 and standard deviation of ~ 2 with 5 samples drawn from the GP prior, each of which corresponds to a separate function. On the right the GP was given additional information (8 observations of the latent function) and fitted to these observations to obtain the posterior. Again the mean and 5 samples are shown. One can notice that close to these observations the uncertainty is greatly reduced, however areas far from observations exhibit greater uncertainty.

3. Gaussian process dynamic modeling for bioprocesses

3.1. Algal lutein production process experimental set-up

The experiment of microalgal lutein production consists of 3 states and 2 control variables. The 3 states are biomass concentration, nitrate concentration and lutein production and the 2 control variables are incident light intensity and nitrate inflow rate. The microalgae species *Desmodesmus* sp. F51 was used for lutein production and experimental temperature was fixed at 35°C. A 1 L photobioreactor (15.5 cm in length and 9.5 cm in diameter) was used in these experiments with an external light source on both sides. Initial biomass concentrations were kept constant and incident light intensities were varied between 150 μmol $m^{-2}s^{-1}$ to 600 μmol $m^{-2}s^{-1}$. Nitrate influent was supplied to the reactor to compensate for the culture nitrate consumption from the 60^{th} hour until the end of the experiment with a fixed inflow rate of 3 mL hr^{-1}. Influent nitrate concentration was chosen as 0.1M or 0.5M. All the runs were carried out over 6 days.

The states were measured every 12 hours over 144 hours, hence 12 measurements were taken for each experimental run. In total 7 different experiments were conducted. All of the experiments were replicated twice and the detailed presentation of experimental design and measurement techniques can be found in del Rio-Chanona et al. (2017a). The operating conditions of these experiments are summarized in Table 1.

Table 1: Operating conditions of 7 algal lutein production experiments

Operating conditions	Exp1	Exp2	Exp3	Exp4	Exp5	Exp6	Exp7
Initial Biomass (g L^{-1})	0.07	0.07	0.07	0.07	0.07	0.07	0.07
Initial Nitrate (mM)	8.8	30	8.8	8.8	8.8	30	8.8
Inflow rate (mL h^{-1})	3.0	3.0	3.0	3.0	3.0	3.0	3.0
Influent nitrate (M)	0.5	0.5	0.1	0.1	0.1	0.5	0.1
Light intensity (μmol $m^{-2}s^{-1}$)	300	600	150	480	600	480	300

The aim of this section is to introduce GP regression in the context of a discrete time, dynamic black-box model for a biosystem given a set of time series measurements (data sets). It is emphasized that the measurements are taken at a constant sampling frequency. Therefore, in this paper we consider the dynamic system to be in the form of:

$$\mathbf{x}(t) = \boldsymbol{F}(\mathbf{x}(t-1), \mathbf{u}(t-1)) \tag{20}$$

where t is the discrete time, $\mathbf{x} \in \mathbb{R}^n$ denotes the states, $\mathbf{u} \in \mathbb{R}^m$ denotes the control inputs and $\boldsymbol{F} : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^n$ resembles the nonlinear transition dynamics. It is assumed that the control inputs $\mathbf{u}$ are deterministic.

In simple words Equation 20 means that the system at time step t will be predicted using measurements and inputs at the previous time step $t-1$. This is the general approach when a real experiment is conducted, where past data are used to predict and hence optimize the process at a future time.

For a biosystem the states are commonly given by concentrations, while a common input is the feed rate of a substrate. For example for the lutein case study the state vector is given by $\mathbf{x} = [C_X, C_N, C_L]^T$, where C_X represents the concentration of algal biomass, C_N the concentration of nitrate and C_L the concentration of lutein; while the control inputs are given by $\mathbf{u} = [Li, F_N]^T$, where Li denotes the light intensity and F_N the inflow rate of nitrate.

3.2. Data preparation

To model the multi-input, multi-output system in Equation 20, we employ independent GPs for each output, *i.e.* each output is modeled by a separate GP. The training procedure is therefore the same as outlined in Section 1.2. The data is consequently given, such that each GP can be trained with the same multivariate inputs, but with different single dimensional outputs.

GPs are identified from input-output data pairs, and can be adopted to approximate the dynamic behavior described by Equation 20 given a set of measurements. The first step consists of preparing available data points for the GP training. Assuming several laboratory experiments have been conducted, we are commonly given the initial conditions $\mathbf{x}(0)$ and the measurements of $\mathbf{x}(t)$ at constant time intervals over different experimental runs with known controls $\mathbf{u}(t)$. From s distinct experimental runs, we obtain data over s time series (data sets), which gives us data in the form of $\left\{\mathbf{x}^{(i)}(0), \ldots, \mathbf{x}^{(i)}(T_i)\right\}_{i\in\{1,\ldots,s\}}$ and $\left\{\mathbf{u}^{(i)}(0), \ldots, \mathbf{u}^{(i)}(T_i - 1)\right\}_{i\in\{1,\ldots,s\}}$, where T_i denotes the number of time intervals in experimental run i. We assume that the sampling rate of all experiments remains constant. In this research we use the augmented vector $\mathbf{x}_a(t) = [\mathbf{x}(t), \mathbf{u}(t)]^T \in \mathbb{R}^{n+m}$ as inputs and the differences $\boldsymbol{\Delta}_{\mathbf{x}}(t) = \mathbf{x}(t) - \mathbf{x}(t-1) + \boldsymbol{\epsilon} \in \mathbb{R}^n$ as regression targets, where $\boldsymbol{\epsilon}$ denotes measurement noise. The regression targets define what we want to predict, *i.e.* we aim to predict the change of the states at each stage using GPs.

The input-output data is consequently given by collecting the measurements of all experiments in the

matrices $\mathbf{X}'$ and $\mathbf{Y}'$:

$$\mathbf{X}' = \begin{bmatrix} [\mathbf{x}^{(1)}(0)^T, \mathbf{u}^{(1)}(0)^T] \\ \vdots \\ [\mathbf{x}^{(1)}(T_1-1)^T, \mathbf{u}^{(1)}(T_1-1)^T] \\ [\mathbf{x}^{(2)}(0)^T, \mathbf{u}^{(2)}(0)^T] \\ \vdots \\ [\mathbf{x}^{(2)}(T_2-1)^T, \mathbf{u}^{(2)}(T_2-1)^T] \\ \vdots \\ [\mathbf{x}^{(s)}(0)^T, \mathbf{u}^{(s)}(0)^T] \\ \vdots \\ [\mathbf{x}^{(s)}(T_s-1)^T, \mathbf{u}^{(s)}(T_s-1)^T] \end{bmatrix}^T, \mathbf{Y}' = \begin{bmatrix} \mathbf{x}^{(1)}(1)^T - \mathbf{x}^{(1)}(0)^T \\ \vdots \\ \mathbf{x}^{(1)}(T_1)^T - \mathbf{x}^{(1)}(T_1-1)^T \\ \mathbf{x}^{(2)}(1)^T - \mathbf{x}^{(2)}(0)^T \\ \vdots \\ \mathbf{x}^{(2)}(T_2)^T - \mathbf{x}^{(2)}(T_2-1)^T \\ \vdots \\ \mathbf{x}^{(s)}(1)^T - \mathbf{x}^{(s)}(0)^T \\ \vdots \\ \mathbf{x}^{(s)}(T_s)^T - \mathbf{x}^{(s)}(T_s-1)^T \end{bmatrix}^T \tag{21}$$

where $\mathbf{X}' \in \mathbb{R}^{(n+m)\times N}$ are the training inputs, $\mathbf{Y}' \in \mathbb{R}^{n\times N}$ are the training targets and is the total number N of input-output data pairs.

Note that what we are proposing is for the GP to predict the change of the states over a fixed time interval given previous states and inputs (*e.g.* given biomass concentration, lutein concentration, nitrate concentration, light intensity and nitrate input at time $t-1$, we can predict the increase or decrease on biomass concentration, lutein concentration and nitrate concentration from time $t-1$ to time t).

The GPs were trained with transformed data. To train the GPs, the inputs were scaled to lie in $[0, 1]$. The input scaling was chosen as a popular feature scaling procedure that have been shown to improve the prediction quality (Aksoy and Haralick, 2001). Unlike the output, the equations used for the input do not assume zero mean and instead account for the mean of the input, consequently a zero mean scaling is not required. The outputs were scaled to have mean 0 to match the zero mean assumption introduced in Section 1.2 and a standard deviaiton of 1. Transformations also help to set the priors of the hyperparameters introduced in Section 1.2 (Equation 15), since normalized data behave in a more predictable fashion. The described transformations are accomplished as follows:

$$\mathbf{X}^{(i)} = \mathbf{A}\mathbf{X}'^{(i)} - \mathbf{b} \tag{22}$$

where $\mathbf{X}_i$ and $\mathbf{X}^{(i)}$ are the i^{th} row and column of matrix $\mathbf{X}$ respectively, $\mathbf{X}'_i$ and $\mathbf{X}'^{(i)}$ are the i^{th} row and column of matrix $\mathbf{X}'$ respectively, $\mathbf{b} = [\min \mathbf{X}'_1, \ldots, \min \mathbf{X}'_{n+m}]^T$ and
$\mathbf{A} = \text{diag}([1/(\max \mathbf{X}'_1 - \min \mathbf{X}'_1), \ldots, 1/(\max \mathbf{X}'_{n+m} - \min \mathbf{X}'_{n+m})])$

$$\mathbf{Y}^{(i)} = \mathbf{C}\mathbf{Y}'^{(i)} - \mathbf{d} \tag{23}$$

where $\mathbf{Y}^{(i)}$ is the i^{th} column of matrix $\mathbf{Y}$, $\mathbf{Y}'^{(i)}$ the i^{th} column of the matrix $\mathbf{Y}'$, $\mathbf{d} = [\overline{\mathbf{Y}'}_1, \ldots, \overline{\mathbf{Y}'}_n]^T$ and $\mathbf{C} = \text{diag}([1/std_1, \ldots, 1/std_n])$, where $\overline{\mathbf{Y}'}_i$ is the sample mean and std_i the sample standard deviation of row i of matrix $\mathbf{Y}'$.

3.3. Training of Gaussian processes

The inputs of the GP is the concatenated vector of states and deterministic control inputs $[\mathbf{x}^T, \mathbf{u}^T]^T$, where $\mathbf{x} = [C_X, C_N, C_L]^T$ represents the concentration of biomass (C_X), concentration of nitrate (C_N) and concentration of lutein (C_L), while $\mathbf{u} = [Li, F_N]^T$ denotes the light intensity (Li) and inflow rate of nitrate (F_N). The training outputs are given by the differences of the states at each time step in the training data. The three independent GPs, one for each state, were constructed based on the procedure outlined in the subsequent sections. The parameters were optimized over their log-values with priors set on their log-values as well to ensure positiveness as was shown in Section 1.2 in Equation 15. The same Gaussian priors were used for all states with mean and variances given in Table 2, which were set to ensure that the parameters do not take too large or too small values and hence to prevent overfitting. It is possible to use the same Gaussian priors for all states due to the data transformation outlined in Section 2.2. The standard deviation of the

initial state was taken to be 5% of the initial state, based on the current experimental measurement accuracy. The initial covariance matrix is hence given by $\mathbf{\Sigma_x}(0) = \text{diag}([(0.05C_X(0))^2, (0.05C_N(0))^2, (0.05C_L(0))^2])$.

Table 2: Variance and mean of Gaussian prior distributions on the log of the hyperparameters

Hyperparameter	Mean	variance
$\log(\lambda_1), \ldots, \log(\lambda_n)$	0.0	1.0
$\log(\alpha)$	0.0	2.0
$\log(\sigma_\epsilon)$	-6.0	4.0

The detailed training procedure is explained in the following subsections.

3.4. Gaussian process prior

The GP regression framework is designed for one-step ahead predictions by identifying a latent function $\mathbf{f}(\cdot)$ to predict $\mathbf{\Delta_x}(t)$ given $\mathbf{x}_a(t-1)$:

$$\mathbf{\Delta_x}(t)|\mathbf{x}_a(t-1) = \mathbf{f}(\mathbf{x}_a(t-1)) + \boldsymbol{\epsilon},\ \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma_\epsilon}) \tag{24}$$

to approximate the subsequent states by:

$$\mathbf{x}(t)|\mathbf{x}_a(t-1) \approx \mathbf{x}(t-1) + \mathbf{f}(\mathbf{x}_a(t-1)) \tag{25}$$

where $\boldsymbol{\epsilon} \in \mathbb{R}^n$ represents the measurement noise, which is assumed to be normally distributed with covariance matrix $\mathbf{\Sigma_\epsilon} = \text{diag}([{\sigma_\epsilon^{(1)}}^2, \ldots, {\sigma_\epsilon^{(n)}}^2])$.

In Figure 2 an illustration is shown for a latent function representing a time series of state x that is modeled by a GP using several noisy observations.

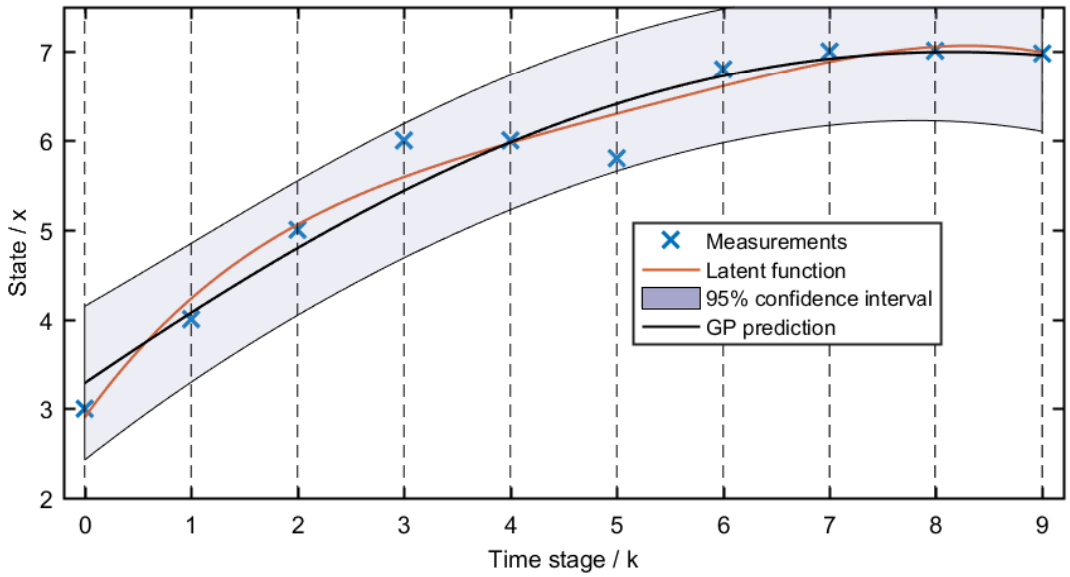

Figure 2: Illustration of a latent function of a times series modeled by a GP through a finite number of measurements. The confidence region predicted by the GP is also shown.

Commonly, GPs are employed for multi-input, single-output problems as was introduced in Section 1.2. An effective extension proposed in this research to multi-outputs is to use a separate, independent GP for each output (Rasmussen and Williams, 2006), where independence means that the outputs are assumed to be uncorrelated. The latent function in Equation 25, is therefore given by:

$$\mathbf{f}(\mathbf{x}_a) = [f^{(1)}(\mathbf{x}_a), \ldots, f^{(n)}(\mathbf{x}_a)]^T \tag{26}$$

where each component $f^{(i)}(\mathbf{x}_a)$) is modeled separately by a GP and $\mathbf{x}_a$ is an arbitrary input.

Given that essentially the same process is carried out n-times with different output data, a superscript (i) was added to refer to the separate GPs for each output dimension. We can therefore write that $f^{(i)}(\mathbf{x}_a)$ is distributed as a GP as follows, the same as Section 1.2:

$$f^{(i)}(\mathbf{x}_a) \sim GP(m^{(i)}(\mathbf{x}_a), k^{(i)}(\mathbf{x}_a, \mathbf{x}'_a)) \tag{27}$$

where $\mathbf{x}'_a$ is an arbitrary input, $m^{(i)}(\mathbf{x}_a)$ and $k^{(i)}(\mathbf{x}_a, \mathbf{x}'_a)$ are separate mean and covariance functions for each component $f^{(i)}(\mathbf{x}_a)$ with different parameter values.

As shown in Equation 24, $\Delta_x^{(i)}$ is a noisy observation to $f^{(i)}(\mathbf{x}_a)$ perturbed by additive Gaussian noise:

$$\Delta_x^{(i)} = f^{(i)}(\mathbf{x}_a) + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, {\sigma_\epsilon^{(i)}}^2) \tag{28}$$

Due to the additive property of Gaussian distributions, the observation $\Delta_x^{(i)}$ of $f^{(i)}(\mathbf{x}_a)$ also follows a GP with the same mean, but larger covariance, see Equation 10:

$$\Delta_x^{(i)} \sim GP(m^{(i)}(\mathbf{x}_a), k^{(i)}(\mathbf{x}_a, \mathbf{x}'_a) + \sigma_\epsilon^2 \delta(\mathbf{x}_a, \mathbf{x}'_a)) \tag{29}$$

Without loss of generality we consider the prior mean function to be zero for each GP, $m^{(i)}(\mathbf{x}_a) := 0$. We propose to use the squared-exponential (SE) covariance function for all the GPs, which is a frequently applied stationary covariance function (O'Hagan and Kingman, 1978). The SE covariance function can then be stated as follows for each GP (Rasmussen and Williams, 2006):

$$k^{(i)}(\mathbf{x}_a, \mathbf{x}'_a) = {\alpha^{(i)}}^2 \exp\left(-\frac{1}{2}(\mathbf{x}_a - \mathbf{x}'_a)^T \mathbf{\Lambda}^{(i)} (\mathbf{x}_a - \mathbf{x}'_a)\right) \tag{30}$$

where each SE function is parametrized with different parameter values to model the separate GPs indicated by i, $\mathbf{\Lambda}^{(i)} = \text{diag}([{\lambda_1^{(i)}}^{-2}, \ldots, {\lambda_n^{(i)}}^{-2}])$ and ${\alpha^{(i)}}^2$ is the signal variance.

The hyperparameters that define the GPs are given in Equations 28 and 30 and are given by the vectors $\mathbf{\Theta^{(i)}} = [\log(\lambda_1^{(i)}), \ldots, \log(\lambda_n^{(i)}), \log(\alpha^{(i)}), \log(\sigma_\epsilon^{(i)})]^T$. The GPs for one-step ahead predictions are obtained by using the data defined in Equations 22 and 23. For each output dimension in $\mathbf{Y}$, *i.e.* for each row in $\mathbf{Y}$, a separate GP needs to be trained (fitted) . In particular, given N training points, n independent GPs are trained with the same input data $\mathbf{X}$ and different response data $\mathbf{y}^{(i)} = \mathbf{Y}_i^T$, where $\mathbf{y}^{(i)} \in \mathbb{R}^N$ is the transpose of the i^{th} row of $\mathbf{Y}$, $\mathbf{Y}_i^T$. The following steps need to be carried out for all $i = 1, \ldots, n$ independent GPs and are basically implemented based on the steps outlined in Section 1.2.

3.5. Gaussian process posterior

The prior GP of $f^{(i)}(\mathbf{x}_a)$ and the observation $\Delta_x^{(i)}$ can be expressed as:

$$f^{(i)}(\mathbf{x}_a) \sim GP(0, k^{(i)}(\mathbf{x}_a, \mathbf{x}'_a)), \quad \Delta_x^{(i)} \sim GP(0, k^{(i)}(\mathbf{x}_a, \mathbf{x}'_a) + {\sigma_\epsilon^{(i)}}^2 \delta(\mathbf{x}_a, \mathbf{x}'_a)) \tag{31}$$

To build the posterior distribution of these functions knowledge of the observations needs to be incorporated. This is accomplished by considering the joint distribution of observations $\mathbf{y}^{(i)}$ and the response at an arbitrary input $\mathbf{x}_a$, $f^{(i)}(\mathbf{x}_a)$, of the latent function. We assume the input $\mathbf{x}_a$ to be deterministic. This is the same process we used for the derivation in Section 1.2 and can be denoted as follows:

$$\begin{bmatrix} f^{(i)}(\mathbf{x}_a) \\ \mathbf{y}^{(i)} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \Sigma_f^{(i)} & \mathbf{\Sigma}_{f,\mathbf{y}}^{(i)} \\ \mathbf{\Sigma}_{\mathbf{y},f}^{(i)} & \mathbf{\Sigma}_{\mathbf{y}}^{(i)} \end{bmatrix}\right) \tag{32}$$

where $\Sigma_f^{(i)} = k^{(i)}(\mathbf{x}_a, \mathbf{x}_a)$, $\mathbf{\Sigma}_{f,\mathbf{y}}^{(i)} = [k^{(i)}(\mathbf{x}_a, \mathbf{x}_1), \ldots, k^{(i)}(\mathbf{x}_a, \mathbf{x}_N)]$, $\mathbf{\Sigma}_{\mathbf{y},f}^{(i)} = \mathbf{\Sigma}_{f,\mathbf{y}}^{(i)T}$ and $\mathbf{\Sigma}_{\mathbf{y}}^{(i)} \in \mathbb{R}^{N \times N}$ is the covariance matrix of the data whose entries are given by $\Sigma_{\mathbf{y}jk}^{(i)} = k^{(i)}(\mathbf{x}_{aj}, \mathbf{x}_{ak}) + {\sigma_\epsilon^{(i)}}^2 \delta(\mathbf{x}_{aj}, \mathbf{x}_{ak})$, where $\mathbf{x}_{aj}$ refers to the vector of the j^{th} column of $\mathbf{X}$.

By conditioning $f^{(i)}(\mathbf{x}_a)$ on the observations according to the joint Gaussian distribution given in Equation 32 and using the identity given in Equation 6, we obtain the posterior predictive distribution of $f^{(i)}(\mathbf{x}_a)$

(Rasmussen and Williams, 2006):

$$f^{(i)}(\mathbf{x}_a)|\mathbf{y}^{(i)} \sim \mathcal{N}(m_f^{(i)}(\mathbf{x}_a), {\sigma_f^{(i)}}^2(\mathbf{x}_a)) \tag{33}$$

$$m_f^{(i)}(\mathbf{x}_a) = \mathbf{\Sigma}_{f,\mathbf{y}}^{(i)}(\mathbf{\Sigma}_{\mathbf{y}}^{(i)})^{-1}\mathbf{y}^{(i)} \tag{34}$$

$${\sigma_f^{(i)}}^2(\mathbf{x}_a) = \Sigma_f - \mathbf{\Sigma}_{f,\mathbf{y}}^{(i)}(\mathbf{\Sigma}_{\mathbf{y}}^{(i)})^{-1}\mathbf{\Sigma}_{\mathbf{y},f}^{(i)} \tag{35}$$

where the mean $m_f^{(i)}$ refers to the best-estimate of the latent function value, while the variance ${\sigma_f^{(i)}}^2$ is a measure of the uncertainty of this prediction.

The predictive distribution of the observation $\Delta_x^{(i)}$ at the test-input is given by the same expressions, except that the term ${\sigma_\epsilon^{(i)}}^2$ needs to be added to the right hand side of Equation 35 (Ebden, 2015):

$$\Delta_x^{(i)}|\mathbf{y}^{(i)} \sim \mathcal{N}(m_f^{(i)}(\mathbf{x}_a), {\sigma_f^{(i)}}^2(\mathbf{x}_a) + \sigma_\epsilon^{2(i)}) \tag{36}$$

3.6. Hyperparameter training

The first step for GP regression is to determine hyperparameter values using MAP, since these are generally unknown *a priori*. This again has to be carried out for every GP separately. In this work, we assume independent Gaussian distributions on the hyperparameters:

$$\Theta_j^{(i)} \sim \mathcal{N}(\mu_{\Theta_j}^{(i)}, {\sigma_{\Theta_j}^{(i)}}^2) \tag{37}$$

where $\mu_{\Theta_j}^{(i)}$ is the mean and ${\sigma_{\Theta_j}^{(i)}}^2$ the variance of the prior Gaussian distribution for the hyperparameter Θ_j, which are specified in Table 2 for the lutain case study.

Following from Equation 32, the log-likelihood of the posterior density of the hyperparameters can be stated as follows (Rasmussen and Williams, 2006):

$$\begin{aligned}\mathcal{L}^{(i)}(\mathbf{\Theta}) = &-\frac{1}{2}\log(|\mathbf{\Sigma}_{\mathbf{y}}^{(i)}|) - \frac{1}{2}\mathbf{y}^{(i)T}(\mathbf{\Sigma}_{\mathbf{y}}^{(i)})^{-1}\mathbf{y}^{(i)} - \frac{N}{2}\log(2\pi) + \\ &\sum_j(-\frac{1}{2}\log(2\pi) - \frac{1}{2}\log({\sigma_{\Theta_j}^{(i)}}^2) - \frac{1}{2{\sigma_{\Theta_j}^{(i)}}^2}(\Theta_j - \mu_{\Theta_j}^{(i)})^2)\end{aligned} \tag{38}$$

The MAP estimates of $\mathbf{\Theta}$ are then given by:

$$\mathbf{\Theta}_{MAP}^{(i)} \in \arg\max_{\mathbf{\Theta}} \mathcal{L}^{(i)}(\mathbf{\Theta}) \tag{39}$$

We now have separate optimal hyperparameter vectors $\mathbf{\Theta}_{MAP}^{(i)}$ for each output $\mathbf{y}^{(i)}$. We will refer to $k^{(i)}(\mathbf{x}_a, \mathbf{x}'_a)$ as the covariance function with hyperparameter values according to $\mathbf{\Theta}_{MAP}^{(i)}$ and hyperparameters with superscript (i) as optimal values from $\mathbf{\Theta}_{MAP}^{(i)}$ in the subsequent sections.

3.7. One-step ahead predictions

Next with the hyperparameters determined previously, predictions can be made one-step ahead through the GP framework, using the predictive distribution given in 33-36. The functions $f^{(i)}(\mathbf{x_a})$ were defined to be able to predict the difference vector at each time-stage, which can be used for one-step ahead predictions as follows (Deisenroth and Rasmussen, 2011):

$$\mathbf{x}(t)|\mathbf{x}(t-1) \sim \mathcal{N}(\mathbf{m_x}(t), \mathbf{\Sigma_x}(t)) \tag{40}$$

$$\mathbf{m_x}(t) = \mathbf{m_x}(t-1) + \mathbf{C}\mathbf{m_f}(\mathbf{x}_a(t-1)) - \mathbf{d} \tag{41}$$

$$\mathbf{\Sigma_x}(t) = \mathbf{C}\mathbf{\Sigma_f}(\mathbf{x}_a(t-1))\mathbf{C}^T \tag{42}$$

$$\mathbf{x}_a(t-1) \sim \mathcal{N}(\mathbf{A}^{-1}([{\mathbf{m_x}}^T(t-1), {\mathbf{m_u}}^T(t-1)]^T + \mathbf{b}), \qquad \mathbf{A}^{-1}(\mathrm{diag}(\mathbf{\Sigma_x}(t-1), \mathbf{\Sigma_u}(t-1))\mathbf{A}^{-T}) \tag{43}$$

where $\mathbf{m_f}(\mathbf{x}_a(t-1)) = [m_f^{(1)}(\mathbf{x}_a(t-1)), \ldots, m_f^{(n)}(\mathbf{x}_a(t-1))]^T$ is a stacked vector of independent predictions and the corresponding diagonal matrix of the variances at $\mathbf{x}(t-1)$ is $\mathbf{\Sigma_f}(\mathbf{x}_a(t-1)) = \text{diag}([{\sigma_f^{(1)}}^2(\mathbf{x}_a(t-1)), \ldots, {\sigma_f^{(n)}}^2(\mathbf{x}_a(t-1))])$. The equations for $m_f^{(i)}$ and ${\sigma_f^{(i)}}^2$ can be found in Equations 34 and 35. Let us emphasize that $\mathbf{m_f}(\mathbf{x}_a(t-1))$ refers to a difference and hence it needs to be added to $\mathbf{m_x}(t-1)$ to calculate the mean of the state at the following time step $\mathbf{m_x}(t)$. The defintions of $\mathbf{A}, \mathbf{b}, \mathbf{C}$ and $\mathbf{d}$ can be found in Equations 22 and 23, and are used to transform the predictions of the GP to obtain predicitons of the true states denoted by $\mathbf{x}$, while $\mathbf{x}_a$ needs to be transformed to match the transformation of the input data.

Let $\mathbf{y_x}$ be the observation of $\mathbf{x}(t)$, then $\mathbf{y_x}$ has the same mean as $\mathbf{x}(t)$, but a larger covariance:

$$\mathbf{y_x}(t)|\mathbf{x}(t-1) \sim \mathcal{N}\left(\mathbf{m_x}(t), \mathbf{\Sigma_x}(t) + \mathbf{\Sigma_\epsilon}\right) \tag{44}$$

where $\mathbf{y_x}(t)$ has the same distribution as $\mathbf{x}(t)$ with the difference that the measurement noise $\mathbf{\Sigma_\epsilon}$ needs to be added to the covariance matrix.

3.8. Multi-step ahead prediction

Using the one-step ahead predictions from the GPs we wish to make multi-step ahead predictions by repeatedly applying Equations 41, 42 and 43. It is, however, important to emphasize that the input to the GP is now a normally distributed random variable, while in GP regression the input is generally assumed to be deterministic. In the one-step ahead predictions in Section 2.7 the input was essentially deterministic, since we conditioned on it.

In other words, the propagation of uncertainty for a multi-step prediction is demonstrated here. In particular, we assume a joint Gaussian distribution on the test input $\mathbf{x}_a$, $p(\mathbf{x}_a) = \mathcal{N}(\mathbf{m_{x_a}}, \mathbf{\Sigma_{x_a}})$. Obtaining the predictive distribution $p(\mathbf{f}(\mathbf{x}_a)|\mathbf{Y}, \mathbf{m_{x_a}}, \mathbf{\Sigma_{x_a}})$ of $\mathbf{f}(\mathbf{x}_a)$ at the test input $\mathbf{x}_a$ is now obtained by integrating out $\mathbf{x}_a$, which is however analytically intractable, so that an approximation method is needed (Deisenroth and Rasmussen, 2011). We assume that the predictive distribution of $\mathbf{f}(\mathbf{x}_a)|\mathbf{Y}, \mathbf{m_{x_a}}, \mathbf{\Sigma_{x_a}}$ is Gaussian, such that the distribution is fully specified by its mean and covariance. For the SE covariance function in Equation 30 exact moment matching is possible, *i.e.* the predictive distribution $p(\mathbf{f}(\mathbf{x}_a)|\mathbf{Y}, \mathbf{m_{x_a}}, \mathbf{\Sigma_{x_a}})$ is approximated by a Gaussian which has the same mean and covariance as the true distribution (Deisenroth et al., 2009). In the multivariate case the predictive mean vector $\mathbf{m_f}(\mathbf{x}_a)$ for an uncertain input $\mathbf{x}_a$ is given by Equation 45. The target dimensions in general co-vary such that the covariance matrix $\mathbf{\Sigma_f}(\mathbf{x}_a)$ is not diagonal anymore. The covariances on the diagonal can be found using Equation 47, while the cross-covariances can be determined using Equation 48. The expressions for the mean and covariance of $\mathbf{f}(\mathbf{x}_a)|\mathbf{Y}, \mathbf{m_{x_a}}, \mathbf{\Sigma_{x_a}})$ are then given by equations involving quantities of all GPs (Deisenroth and Rasmussen, 2011):

$$\mathbf{f}(\mathbf{x}_a)|\mathbf{Y}, \mathbf{m_{x_a}}, \mathbf{\Sigma_{x_a}} \sim \mathcal{N}(\mathbf{m_f}(\mathbf{x}_a), \mathbf{\Sigma_f}(\mathbf{x}_a)) \tag{45}$$

$$\mathbf{m_f}(\mathbf{x}_a) = \mathbf{q}^{(i)T}\boldsymbol{\beta}^{(i)} \tag{46}$$

$$\Sigma_{\mathbf{f}}^{(ii)}(\mathbf{x}_a) = {\alpha^{(i)}}^2 + \boldsymbol{\beta}^{(i)T}\mathbf{Q}^{(ii)}\boldsymbol{\beta}^{(i)} - \text{tr}((\mathbf{\Sigma_y}^{(i)})^{-1}\mathbf{Q}^{(ii)}) - m_f^{(i)}(\mathbf{x}_a)^2 \tag{47}$$

$$\Sigma_{\mathbf{f}}^{(ij)}(\mathbf{x}_a) = \boldsymbol{\beta}^{(i)T}\mathbf{Q}^{(ij)}\boldsymbol{\beta}^{(j)} - m_f^{(i)}(\mathbf{x}_a)m_f^{(j)}(\mathbf{x}_a) \tag{48}$$

where $\boldsymbol{\beta}^{(i)} = (\mathbf{\Sigma_y}^{(i)})^{-1}\mathbf{y}^{(i)}$, $\mathbf{m_f}(\mathbf{x}_a) = [m_f^{(1)}(\mathbf{x}_a), \ldots, m_f^{(n)}(\mathbf{x}_a)]^T$, $q_p^{(i)} = {\alpha^{(i)}}^2|\mathbf{\Sigma_{x_a}}\mathbf{\Lambda}^{(i)} + \mathbf{I}|^{-1/2}\exp(-\frac{1}{2}(\mathbf{m_{x_a}} - \mathbf{x}_{ap})^T(\mathbf{\Sigma_{x_a}} + (\mathbf{\Lambda}^{(i)})^{-1})^{-1}(\mathbf{m_{x_a}} - \mathbf{x}_{ap}))$ and

$$\mathbf{\Sigma_f}(\mathbf{x}_a) = \begin{bmatrix} \Sigma_{\mathbf{f}}^{(11)}(\mathbf{x}_a) & \ldots & \Sigma_{\mathbf{f}}^{(1n)}(\mathbf{x}_a) \\ \vdots & \ddots & \vdots \\ \Sigma_{\mathbf{f}}^{(n1)}(\mathbf{x}_a) & \ldots & \Sigma_{\mathbf{f}}^{(nn)}(\mathbf{x}_a) \end{bmatrix}$$

$$Q_{pq}^{(ij)} = k^{(i)}(\mathbf{x}_{ap}, \mathbf{m_{x_a}})k^{(j)}(\mathbf{x}_{aq}, \mathbf{m_{x_a}})|\mathbf{R}|^{-1/2} \times \exp\left(\frac{1}{2}(\boldsymbol{\nu} - \mathbf{m_{x_a}})^T\mathbf{R}^{-1}\mathbf{\Sigma_{x_a}}(\boldsymbol{\nu} - \mathbf{m_{x_a}})\right) \tag{49}$$

where $\mathbf{R} = \mathbf{\Sigma_{x_a}}(\mathbf{\Lambda}^{(i)} + \mathbf{\Lambda}^{(j)}) + \mathbf{I}$ and $\boldsymbol{\nu} = \mathbf{\Lambda}^{(i)}(\mathbf{x}_{ap} - \mathbf{m_{x_a}}) + \mathbf{\Lambda}^{(j)}(\mathbf{x}_{aq} - \mathbf{m_{x_a}})$. The superscripts i and j refer

to the various quantities with respect to the i^{th} and j^{th} Gaussian process, respectively, of the n independent Gaussian processes trained. The vector $\mathbf{x}_{ap}$ is the p^{th} training input contained in $\mathbf{X}$, *i.e.* the p^{th} column of $\mathbf{X}$.

We are now able to make multivariate, multi-step ahead predictions by recursively applying Equations 45 to 49 together with the following equations to propagate the state through the predicted $\mathbf{\Delta}_{\mathbf{x}}(t)$ (Deisenroth and Rasmussen, 2011):

$$\mathbf{x}(t) \sim \mathcal{N}(\mathbf{m}_{\mathbf{x}}(t), \mathbf{\Sigma}_{\mathbf{x}}(t)) \tag{50}$$

$$\mathbf{m}_{\mathbf{x}}(t) = \mathbf{m}_{\mathbf{x}}(t-1) + \mathbf{C}\mathbf{m}_{\mathbf{f}}(\mathbf{x}_a(t-1)) - \mathbf{d} \tag{51}$$

$$\mathbf{\Sigma}_{\mathbf{x}}(t) = \mathbf{\Sigma}_{\mathbf{x}}(t-1) + \mathbf{C}\mathbf{\Sigma}_{\mathbf{f}}(\mathbf{x}_a(t-1))\mathbf{C}^T + 2\text{cov}(\mathbf{\Delta}_{\mathbf{x}}(t-1), \mathbf{u}(t-1)) \tag{52}$$

$$\mathbf{x}_a(t-1) \sim \mathcal{N}(\mathbf{A}^{-1}([\mathbf{m}_{\mathbf{x}}{}^T(t-1), \mathbf{m}_{\mathbf{u}}{}^T(t-1)]^T + \mathbf{b}), \mathbf{A}^{-1}(\text{diag}(\mathbf{\Sigma}_{\mathbf{x}}(t-1), \mathbf{\Sigma}_{\mathbf{u}}(t-1))\mathbf{A}^{-T}) \tag{53}$$

where $\text{cov}(\mathbf{\Delta}_{\mathbf{x}}(t-1), \mathbf{u}(t-1))$ is 0 and $\mathbf{\Sigma}_{\mathbf{u}} = [0]_{m\times m}$ for deterministic control inputs $\mathbf{u}(t-1)$. For the case when one is interested in a feedback control law, such that the input is given as some function of the current state, $\mathbf{u}(t-1) = \kappa(\mathbf{x}(t-1))$, please refer to Deisenroth (2010). $\mathbf{m}_{\mathbf{x}}(t)$ is the best estimate of the state at k with corresponding covariance $\mathbf{\Sigma}_{\mathbf{x}}(t)$.

Let $\mathbf{y}_{\mathbf{x}}(t)$ be the observations of $\mathbf{x}(t)$, then $\mathbf{y}_{\mathbf{x}}(t)$ has the same mean as $\mathbf{x}(t)$, but a larger covariance:

$$\mathbf{y}_{\mathbf{x}}(t) \sim \mathcal{N}\left(\mathbf{m}_{\mathbf{x}}(t), \mathbf{\Sigma}_{\mathbf{x}}(t) + \mathbf{\Sigma}_{\boldsymbol{\epsilon}}\right) \tag{54}$$

where $\mathbf{y}_{\mathbf{x}}(t)$ has the same distribution as $\mathbf{x}(t)$ with the difference that the measurement noise $\mathbf{\Sigma}_{\boldsymbol{\epsilon}}$ needs to be added to the covariance matrix.

The initial state $\mathbf{x}(0)$ and covariance matrix $\mathbf{\Sigma}_{\mathbf{x}}(0)$ need to be given from which the state can be then propagated to an arbitrary time horizon recursively. The initial state is generally known, while the covariance matrix can be obtained by error propagation. For the lutein case study these were stated in Section 2.3.

4. Artificial neural network

In this section a brief introduction is given to artificial neural networks (ANNs). Currently, ANNs are used as the standard black-box modeling tool to simulate both traditional chemical engineering processes and emerging biological systems. In order to demonstrate the outstanding characteristics of GPs, in this research ANNs are considered as the benchmark to investigate the simulation and prediction capabilities of GPs for bioprocess systems engineering. In particular, a state-of-the-art ANN construction strategy was recently developed to simulate microalgal lutein production in del Rio-Chanona et al. (2017b). Thus, this ANN model will be used to compare against the current constructed GPs.

In general, an ANN is a system of nodes or 'neurons', based on graph theory, organized in layers which are bound together by a series of mono-directional connections and are meant to represent biological learning and computation. These nodes accept inputs and generate outputs which are then either returned or used as inputs to another layer of neurons (García-Camacho et al., 2016; Hosen et al., 2011; Xiong and Zhang, 2004). The source and destination of the connections depend on the structure of the type of network chosen. In specific, the ANN presented in the recent study was built up based upon a type called multi-layer perceptron (del Rio-Chanona et al., 2017b), where connections only point to the next layer, as in the feed-forward case. A schematic of a multilayer ANN is illustrated in Fig. 3. Other architectures have been developed over years, however, they are not within the scope of this study.

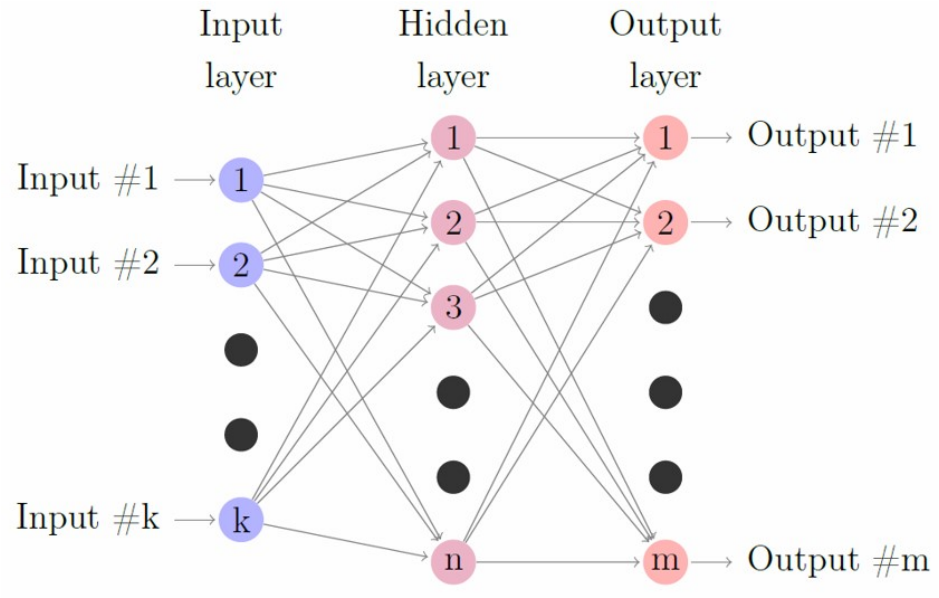

Figure 3: Schematic of an ANN with a single hidden layer, k inputs and m outputs

Within the last decade, there has been a push toward the use of ANNs in chemical engineering (Himmelblau, 2008), where they have found use as estimators and as part of simulation of processes (Pareek et al., 2002; Nelofer et al., 2012; Feng et al., 2013; Mohd Ali et al., 2015). As an example, ANNs have been employed for modeling and prediction in traditional processes such as in distillation, fuel production and in the design of fuel cells (Ochoa-Estopier et al., 2012; Feng et al., 2013; Baroi and Dalai, 2014). They have been also extensively used to simulate difficult processes such as the influence of parameters in catalyst preparation through experimental data (Gunay et al., 2012), or even to extract such rules from existing literature (Odabasi et al., 2014). More importantly, the modeling of different useful processes involving microorganisms have been successfully tackled by ANNs (Vats and Negi, 2013; Nasr et al., 2013). Furthermore, there has been newly reported research focusing on applying ANNs to identify optimal operating conditions for the production of microalgae biorenewables, such as the work published in Mohamed et al. (2013) and del Rio-Chanona et al. (2016b).

Specific to the previous study where ANNs were applied to simulate microalgae biomass growth and lutein production (del Rio-Chanona et al., 2017b), a two hidden layer ANN consisting of 15 nodes in each hidden layer, 5 inputs (biomass concentration, lutein production, light intensity, nitrate concentration, nitrate inflow rate), and 3 outputs (change of biomass concentration, lutein production, nitrate concentration) was constructed, together with another one-layer ANN comprising the same inputs and outputs but 20 nodes in the hidden layer. These optimal network structures were determined through the cutting-edge hyperparameter selection framework, and the ANNs were demonstrated to be of high accuracy and predictive capability. Thus, they are selected in this study for comparison. The hyperparameter selection algorithm is the following:

Algorithm 1 ANN hyperparameter selection

Initialization:
Define the set of possible hidden layers Ω and possible neurons per hidden layer Λ. In this research these sets were set to $\Omega = \{1, 2, 3\}$, $\Lambda = \{3, 5, 10, 15, 20, 25\}$.
Define possible neural network structures as $\Upsilon = \Omega \times \Lambda$, where $\times$ denotes the Cartesian product.
Define the set of time-series Γ. In this study Γ was defined by all the experimental time-series.

1. For ANN_k (neural network structure) in Υ
   (a) For i in Γ: Select this time-series i as the test-set and group the rest as the cross-validation data-set Ψ
   (b) Initialize regularization penalty λ to a small value; in this study $\lambda = 0.001$
      i. For each time-series j in Ψ: Select this time-series j as the cross-validation set and group the rest as the training-set Θ
         A. Train ANN_k on Θ
         B. Compute training and cross-validation errors
         C. Increase regularization penalty λ
         D. If training error has stopped its sharp decrease and cross-validation error increases continue to step (ii). Else, return to (A).
      ii. Use ANN_k to predict i (test-set) and compute the error
   (c) Compile test-set errors for ANN_k
2. Compare test-set errors for all ANN structures, and determine the optimal structure for 1 and 2 hidden layer ANNs

Parameter λ is a penalty imposed to the size of the weights of the ANN to avoid overfitting. In this implementation, the term divides the weight values, hence it starts at a small value and increases gradually throughout the algorithm. Step 1,b,i,D allows to compute the best regularization penalty parameter to reduce overfitting.

5. Results and discussion

5.1. Comparison between Gaussian process and artificial neural network

The performance of the techniques was compared by leaving out the data set for a single experiment and in turn predicting the trajectory of this experiment using the GP methodology outlined in section 2 (multi-step ahead prediction). For the ANN the same cross-validation procedure was implemented for networks with 1 and 2 hidden layers. This approach was applied to all experiments, thus in total there are predictions for all 7 experiments. More importantly, to verify the predictive capability of the GP for complex biological systems, the comparison between GP and ANNs in the current study is executed through an offline framework where only initial operating conditions (and nitrate inflow rate) are provided, and the models have to predict the entire dynamic behavior of the process.

The results are shown in Figures 4-7, on which the predictions from the GP, ANN with 1 hidden layer and 2 Hidden layers were plotted for each of the 7 experiments. In addition, error bars were added to the GP predictions showing the 99% confidence regions of the latent functions given by the GPs. The confidence regions presented are based on the experimental observations as given by Equation 44. This is the correct confidence region to show, since we are trying to see in how far the experimental data perturbed by measurement noise is contained within the confidence region, while the true underlying function is unknown. It is necessary to recall that ANNs are not able to estimate the confidence region of their predicted results.

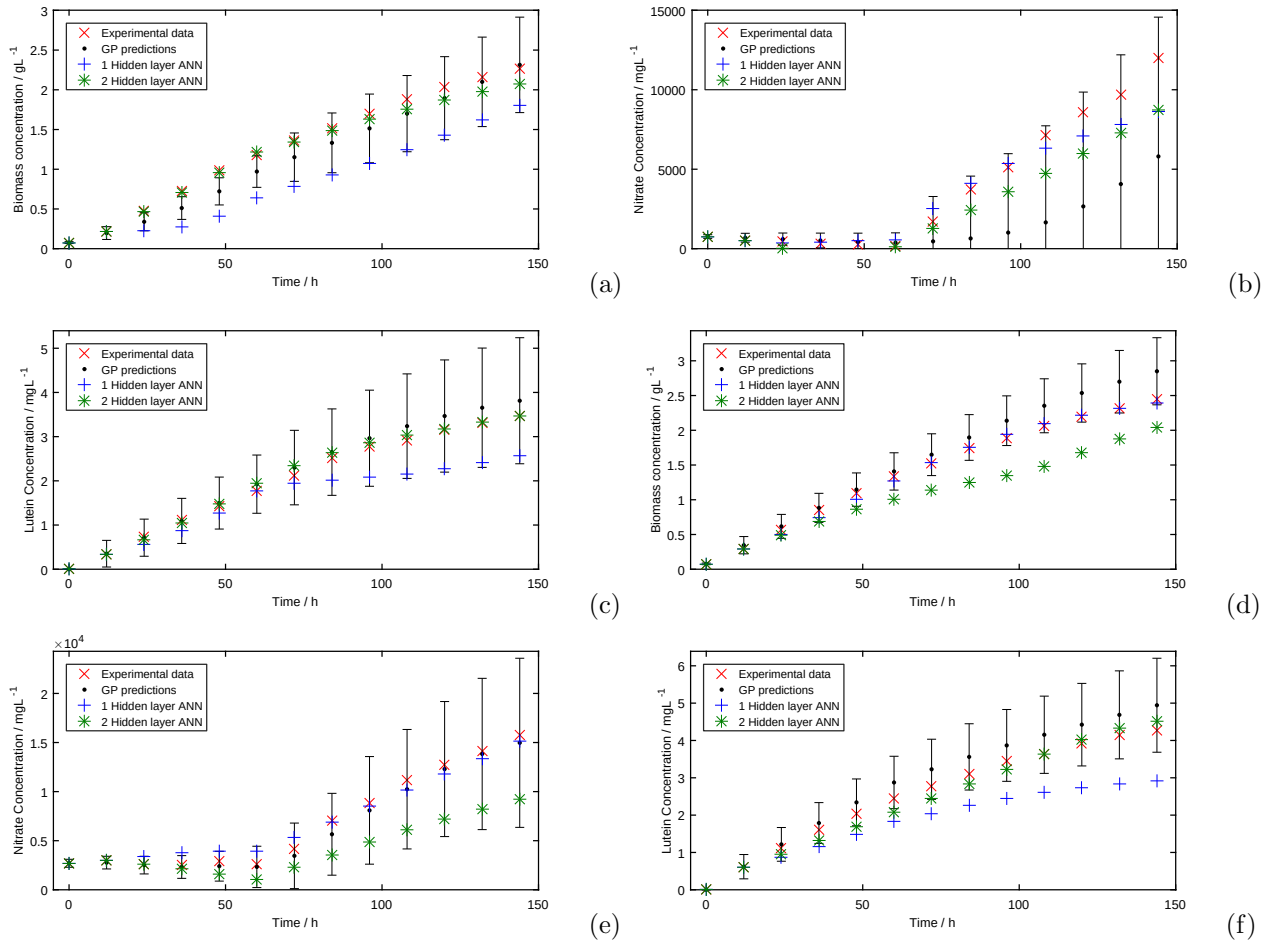

Figure 4: Cross-validation for data set 1, where (a) is the dynamic performance for biomass concentration, (b) for nitrate concentration and (c) for lutein concentration; and for cross-validation for data set 2, (d) is the dynamic performance for biomass concentration, (e) for nitrate concentration and (f) for lutein concentration

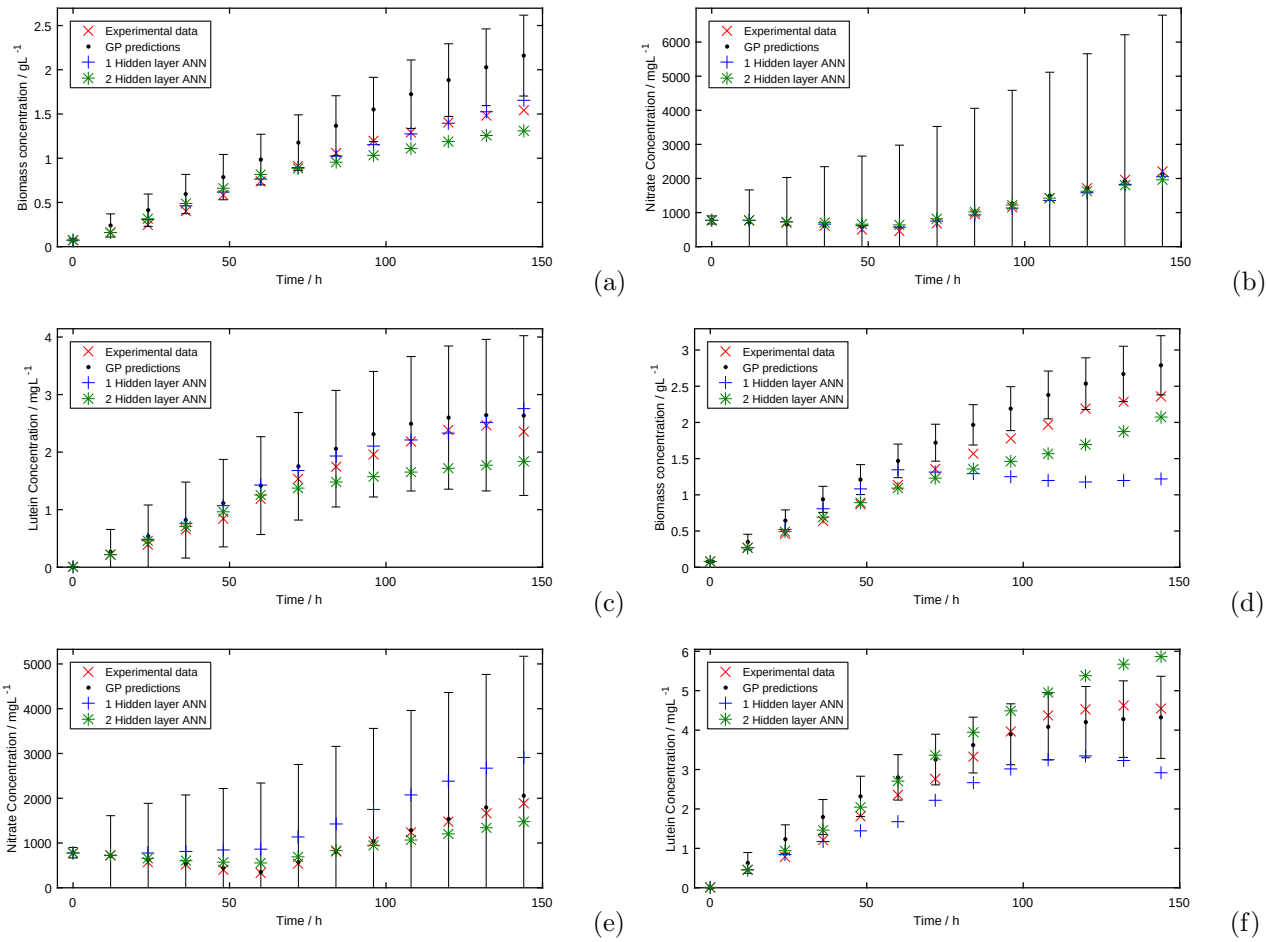

Figure 5: Cross-validation for data set 3, where (a) is the dynamic performance for biomass concentration, (b) for nitrate concentration and (c) for lutein concentration; and for cross-validation for data set 4, (d) is the dynamic performance for biomass concentration, (e) for nitrate concentration and (f) for lutein concentration

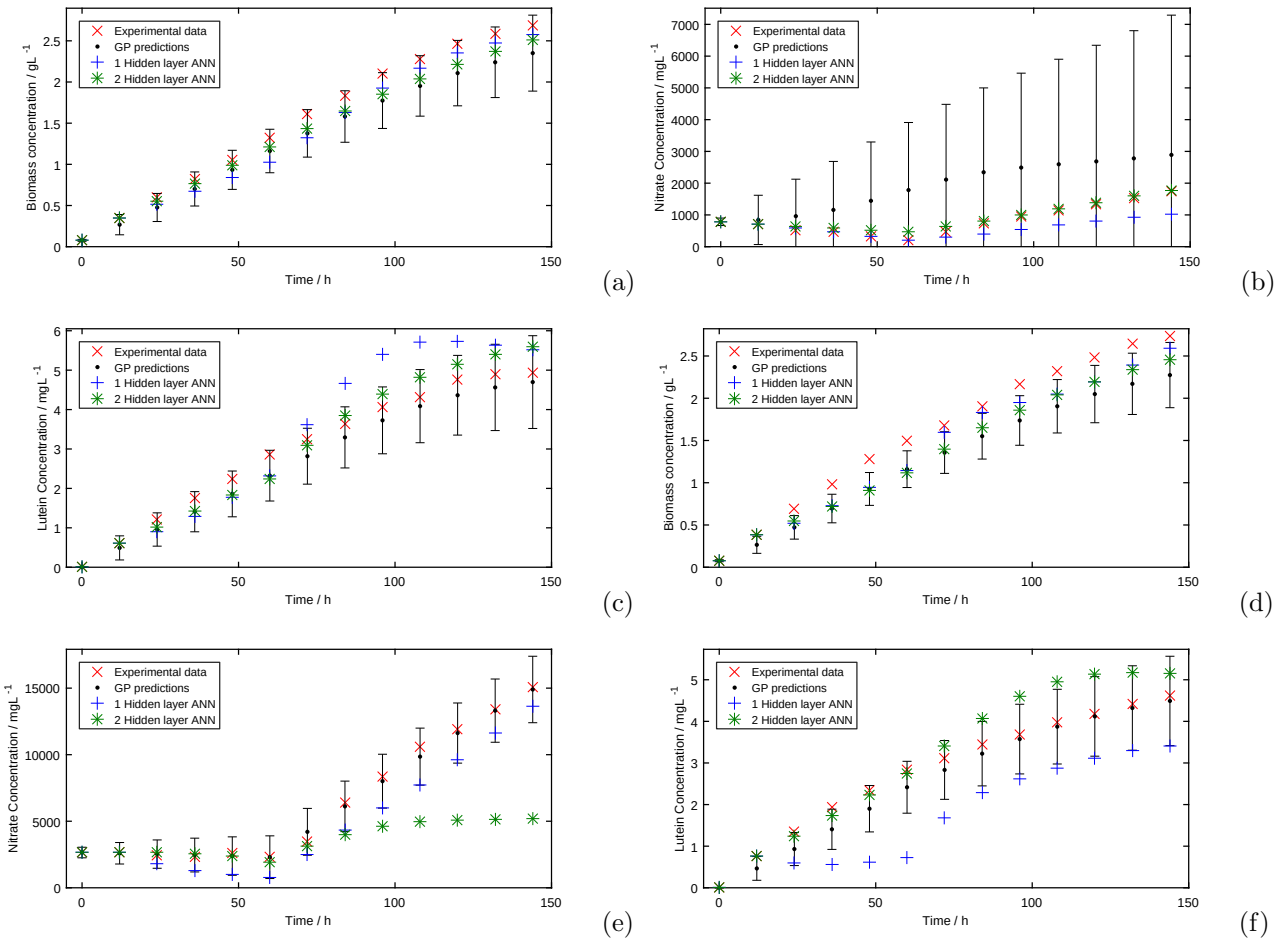

Figure 6: Cross-validation for data set 5, where (a) is the dynamic performance for biomass concentration, (b) for nitrate concentration and (c) for lutein concentration; and for cross-validation for data set 6, (d) is the dynamic performance for biomass concentration, (e) for nitrate concentration and (f) for lutein concentration

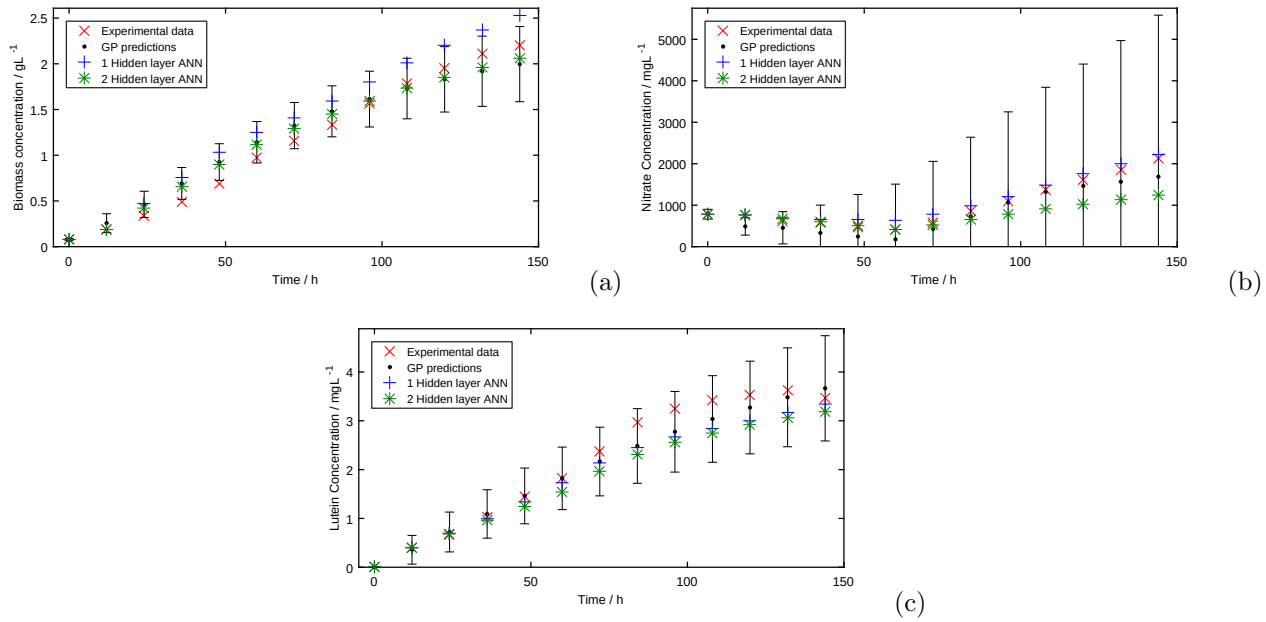

Figure 7: Cross-validation for data set 7, where (a) is the dynamic performance for biomass concentration, (b) for nitrate concentration and (c) for lutein concentration

From Figures 4-7 it can be appreciated that all models show good predictions of the results given the limited data available, although it is difficult to determine the prediction superiority of one method over the other. Each method can be seen to be better at predicting certain datasets, while none is always superior.

From the GP predictions it can be seen that the experimental data is rarely outside the confidence interval provided, this being an encouraging result for future research. However, it can be also seen that the error propagation is relatively large in the case of the nitrate concentration. This is mainly attributed to the fact that the measurement noise is taken to be the same for both high and low nitrate concentrations. For low nitrate concentrations, in fact, percentage-wise the measurement noise is similar to that of the high nitrate concentration. Nonetheless, in an actual implementation, states would be monitored online and real-time datasets would be updated to GPs constantly. As a result, the prediction uncertainty would be contained within a narrower region.

However, in Fig. 5 (d) and Fig. 6 (d) the data points indeed fall slightly outside the confidence regions. This does indicates two limitations of the GP framework proposed in this paper. The first one is that errors may not be entirely reliable, since the error propagation involves approximating the true distribution using only the mean and variance, which may lead to too low noise if the true distribution is either particular skewed or multi-modal. This could be alleviated in future work by using particle-based approaches instead, which in the limit approximates the true distribution exactly (Girard et al., 2003). In addition, the hyperparameters were set using optimization, which ignores the uncertainty of the parameter values themselves. A more accurate, but expensive solution to this problem is to integrate the hyperparameters out instead (Rasmussen and Williams, 2006), which could also be tested in future work.

Finally, through a design of experiment framework, it is possible to improve the accuracy of the GP models. One efficient way to conduct this is to design experiments in areas in which the variance of the GP is high, since this shows regions that have high sparsity of data-points. Furthermore, if we are interested in finding optimal operating conditions, it is sensible to try to learn the model more accurately in areas that are promising. This has been used to great success in the global optimization community by sequentially designing experiments that trade off exploring unknown regions and exploiting regions in which good operating conditions have already been observed (Sacks et al., 1989). This shows again an advantage of GPs over ANNs due to the availability of an uncertainty measure.

In addition to the visual comparison given in Figures 4-7, the mean square error (MSE) over each time series was calculated for each machine learning algorithm, *i.e.* the difference of the prediction from the

measurements was squared and averaged over each time series. The values of the MSE for the biomass concentration, nitrate concentration and lutein concentration can be found in Tables 3, 4 and 5 respectively. The smallest value (best performing algorithm) among the three models is highlighted in bold.

Table 3: Comparison of MSE for GP, 1 Hidden-layer ANN (1HL-ANN), 2 Hidden-layer ANN (2HL-ANN) from Exp 1-7 for **biomass concentration** (g^2L^{-2}) . The best performing algorithm in terms of MSE is highlighted in bold.

Exp	GP	1HL-ANN	2HL-ANN
1	0.028	0.270	**0.010**
2	0.052	**0.011**	**0.011**
3	0.130	**0.002**	0.021
4	0.121	0.381	**0.068**
5	0.086	0.030	**0.028**
6	0.133	**0.057**	0.368
7	0.024	0.062	**0.015**

Table 4: Comparison of MSE for GP, 1 Hidden-layer ANN (1HL-ANN), 2 Hidden-layer ANN (2HL-ANN) from Exp 1-7 for **nitrate concentration** (g^2L^{-2}) . The best performing algorithm in terms of MSE is highlighted in bold.

Exp	GP	1HL-ANN	2HL-ANN
1	13.290	**1.552**	2.812
2	**0.745**	4.919	26.448
3	**0.006**	0.008	0.016
4	**0.002**	0.454	0.420
5	0.125	0.142	**0.015**
6	**0.226**	0.951	10.626
7	0.062	**0.019**	0.167

Table 5: Comparison of MSE for GP, 1 Hidden-layer ANN (1HL-ANN), 2 Hidden-layer ANN (2HL-ANN) from Exp 1-7 for **lutein concentration** (mg^2L^{-2}) . The best performing algorithm in terms of MSE is highlighted in bold.

Exp	GP	1HL-ANN	2HL-ANN
1	0.045	0.324	**0.011**
2	**0.175**	0.967	0.658
3	0.059	**0.031**	0.144
4	**0.145**	0.793	0.432
5	**0.092**	0.631	0.162
6	**0.090**	1.337	0.227
7	**0.064**	0.127	0.203

From Tables 3, 4 and 5, it is concluded that the GP shows a comparable performance to the ANN (either 1 or 2 layers). For example, it can be seen that GP attains the best prediction result on 5 out of the 7 experiments and comes second for the remaining 2 experiments when predicting lutein concentration. Similarly, it possesses the best prediction on 4 out of the 7 experiments and comes second twice when estimating the trajectory of nitrate concentration in the current study. Even though the best prediction for biomass concentration is always an ANN, the current constructed GP still provides a comparable result (second best prediction on 4 experiments) for the majority of the experimental tests. This clearly indicates the great predictive capability of GPs and promising potential for bioprocess systems engineering applications.

Moreover, significant attention should be paid on the fact that the current used ANNs were constructed based on advanced methodologies through which the optimal structure of ANNs were identified and their predictive capability is maximized (del Rio-Chanona et al., 2017b). However, as GPs have never been applied to describe and understand the behavior of complex biological systems, specific strategies capable of identifying the optimal structure of GPs are not available yet. Therefore, the comparable predictive

capability and performance of the current GPs against the optimal ANNs strongly suggests the potential of GPs on bioprocess modeling and optimization.

The most important contribution of the GP is that a confidence region is simultaneously estimated during process prediction, and it is found that experimental measurements in almost all the 7 experiments fall within this region. Such a region is essential for sensitive bioprocess optimization and for the implementation of robust optimization strategies (*e.g.* worst-case scenario optimization), as the safety of a bioprocess is in general given higher priority than the process yield. Furthermore, the GP never gave catastrophically unreliable trajectory predictions. This conclusion further emphasizes that GPs can not only provide an accurate prediction for long-term biosystems, but also contribute a reliable estimation for dynamic bioprocess design, modeling and control.

5.2. Dynamic optimization with stochastic constraints

One of the main advantages of GP regression over more common regression methods, such as ANNs, is that it gives us a measure of prediction uncertainty. In this section, we show how this measure can be used in optimization to ensure that the optimal solution remains in the validity range of the model. The objective of the optimization is to find operating conditions to yield the maximum lutein concentration by the end of the process at the 144$^{\text{th}}$ hour, *i.e.* with a time horizon of length $N = 12$. The operating conditions are given by control actions of light intensity and the nitrate inflow rate chosen at each time stage. In addition, the mean of the initial concentrations of biomass and nitrate was also varied.

The model adopted in this section is determined by using all the data from the 7 experiments given in Table 1 and following the procedure in Section 2. The control actions are taken to be deterministic. The first constraint given in brackets in Equation 55 is the stochastic constraint that limits the variance of lutein at the final stage to be below 0.025 mg L^{-1}, and hence the lutein concentration to lie in a confidence region of 95% with $\pm$0.025 mg L^{-1}. This measure was chosen, as it directly targets the relevant uncertainty of the measure to be optimized, while still considering all other uncertainties since these are iteratively used as noisy input to obtain the final prediction. The overall optimization problem is given below:

Dynamic optimization problem using GP model

$$\max_{\mathbf{U},\, \mathbf{m_x}(0)} \quad \mathbb{E}\left[C_{L_i}(N)\right] = m_x^{(3)}\left(\mathbf{x}_a(N)\right) \tag{55a}$$

subject to:

$$\Sigma_x^{(3,3)}(N) \leq 0.025 \tag{55b}$$

$$\mathbf{\Sigma_x}(0) = \text{diag}\left(\left[\left(0.05 m_x^{(1)}(0)\right)^2, \left(0.05 m_x^{(2)}(0)\right)^2, \left(0.05 m_x^{(3)}(0)\right)^2\right]\right) \tag{55c}$$

$$[0,0]^T \leq \mathbf{u}(k) \leq [600, 1.5]^T \quad \forall k \in \{0, ..., N-1\} \tag{55d}$$

$$[0,0,0]^T \leq \mathbf{m_x}(0) \leq [0.5, 6000, 0]^T \tag{55e}$$

$$(48) - (51) \quad \forall k \in \{1, \ldots, N\} \tag{55f}$$

where $\mathbf{U} = [\mathbf{u}(0), \ldots, \mathbf{u}(N-1)]$ is a matrix of control inputs at each time interval, $\Sigma_x^{(3,3)}(t)$ is the 3$^{\text{rd}}$ diagonal element of the state matrix $\mathbf{\Sigma_x}(t)$ corresponding to the variance of lutein concentration at time stage k and $m_x^{(i)}(t)$ is the i$^{\text{th}}$ dimension of the vector $\mathbf{m_x}(t)$ corresponding to the expected value of the respective concentrations at time stage k.

The optimization in Equation 55 was conducted using the fmincon function in Matlab with 20 multistart initial points chosen according to a *maximin* Latin hypercube. Choosing initial starting points according to a Latin hypercube scaled to the upper and lower bounds of the decision variables has been shown to yield good optimization results in Raue et al. (2013). With stochastic constraints for example the optimization converged to either of six solutions with relatively small variations. The optimal solution yielded a value of lutein of 4.94mg L^{-1}. The optimal solution was observed twice in the optimization procedure. The mean of the solutions obtained was 4.4mg L^{-1} with a standard deviation of 0.33mg L^{-1}. The optimal trajectories are shown in Figures 8 and 9, where two optimization results are shown, one with the stochastic constraints and another without.

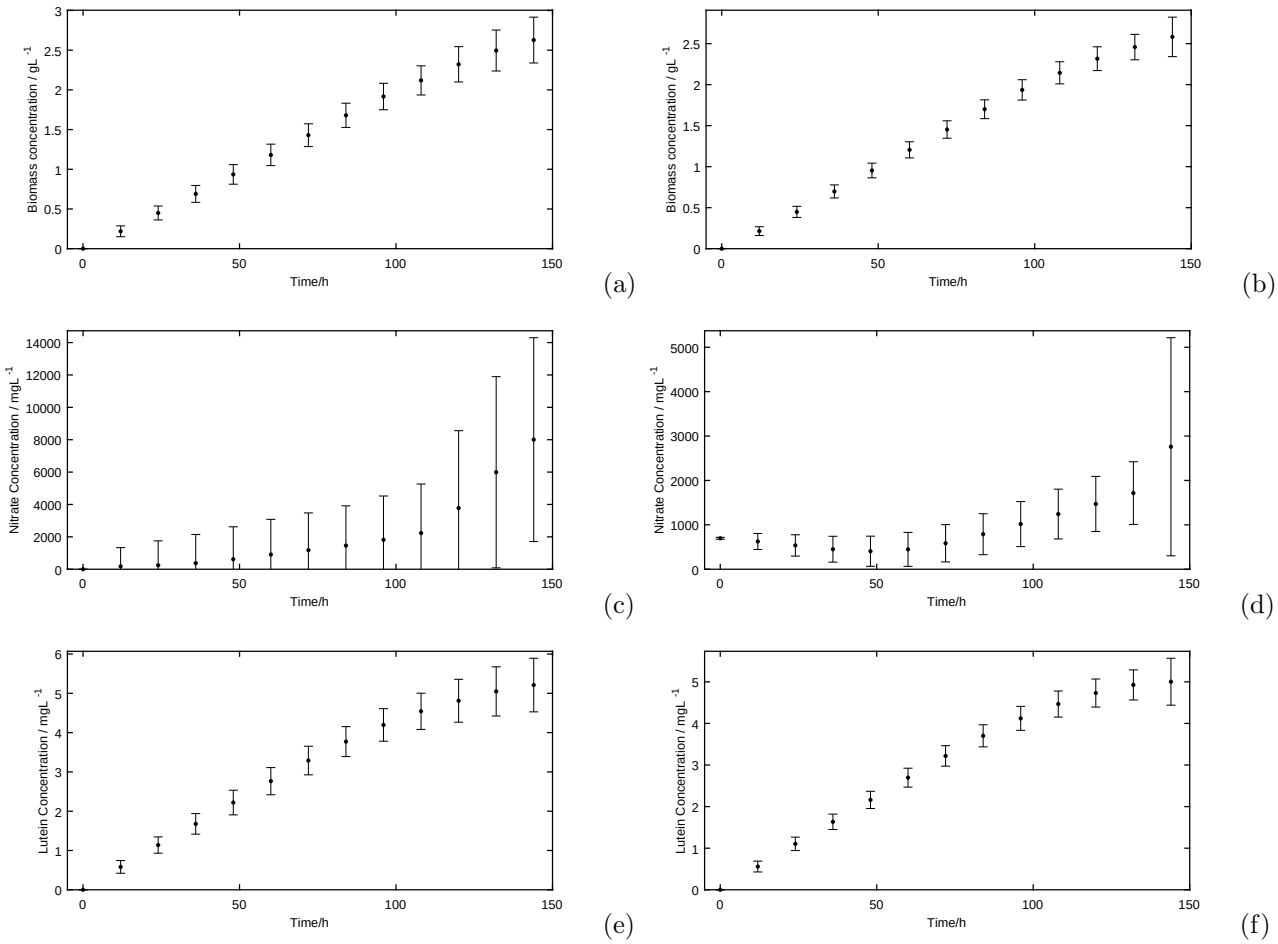

Figure 8: Results of dynamic optimization for lutein production. (a), (c), (e): Optimal trajectory of concentrations of biomass, nitrate, and lutein without stochastic constrain, respectively; (b), (d), (f): Optimal trajectory of concentrations of biomass, nitrate, and lutein with stochastic constraint, respectively.

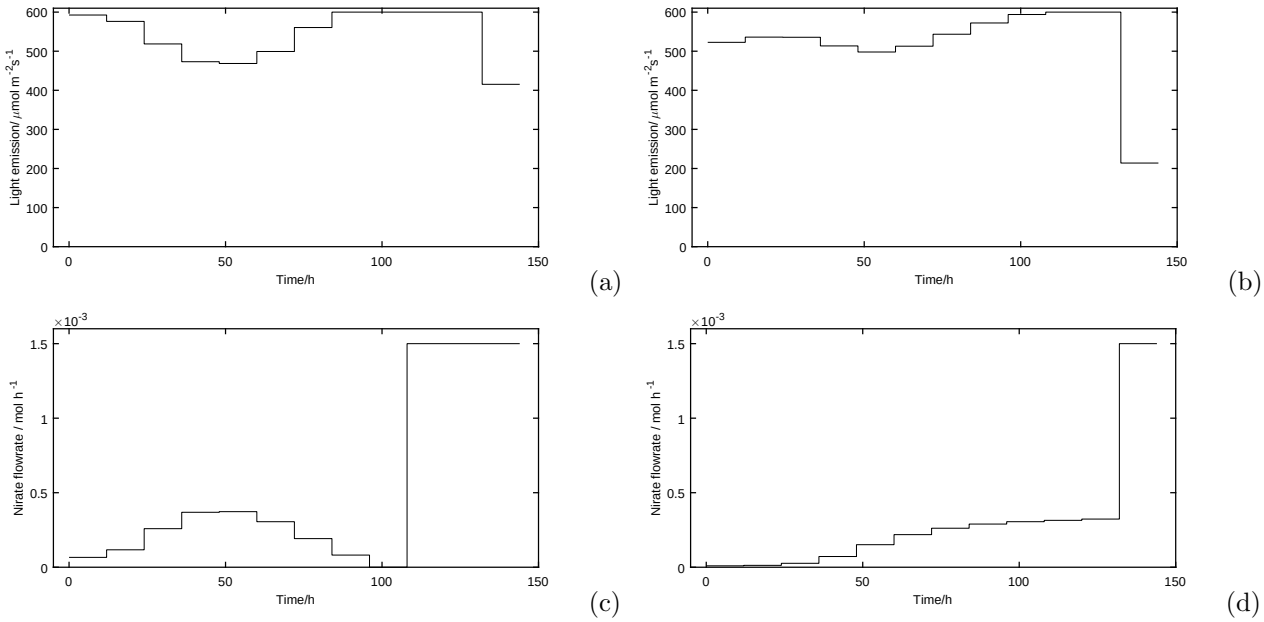

Figure 9: Results of the optimal control scheme for lutein production. (a), (c): Optimal control input of light emission and nitrate inflow rate without stochastic constraint, respectively; (b), (d): Optimal control input of light emission and nitrate inflow rate with stochastic constraint, respectively.

From the figures, it is seen that both optimization scenarios yield similar results since the predicted optimal operating conditions lie on the boundary of the model, *i.e.* the optimum is close to the conditions in the second training experiment (Exp2), given in Table 1. Comparing the two scenarios, although the optimization without stochastic constraints yields a slightly higher final lutein concentration (5.10 mg L^{-1}), the case with stochastic constraints (4.95 mg $^{-1}$), can be seen to show a lower uncertainty on all the state trajectories, in particular when predicting nitrate concentrations. This suggests that in order to reduce the uncertainty of process optimization and guarantee the safety of underlying biosystems, it is necessary to embed stochastic constraints into the optimization framework. It is also worth noting that the optimal result with stochastic constraints is closer to the second training experiment than without it. This was executed to minimize the uncertainty of the model, *i.e.* as shown in Figure 1, where uncertainty is substantially higher away from the measurements. Such a result means that experimental trajectories near optimal solutions can be highly valuable. Furthermore, in general, the model uncertainty is relatively high, suggesting that more data should be used to further enhance the optimization results.

6. Conclusions

Overall, a new methodology was introduced to construct a dynamic model for biorenewable synthesis and process optimization by using Gaussian process regression. By comparing against ANNs, the high predictive capability and simultaneous uncertainty measure of GPs show an outstanding capacity to simulate and optimize complex biological processes, particularly in cases where the lack of experimental data becomes a severe challenge for the construction of kinetic models. Furthermore, a distinctive feature of GPs is the simultaneous estimation of model uncertainty alongside the real-time optimisation framework, which is difficult to be achieved by other techniques.

In particular, the provision of a confidence region from GPs has the potential to significantly facilitate their application in process scale-up and real-time optimal control for both traditional bioprocesses such as fermentation and newly proposed algae based photo-production systems, since the precise prediction and control action decision-making throughout the entire process is indispensable to guarantee the safety and productivity of these systems. An important issue to note, is that, compared to traditional empirical and phenomenological models, data-driven models are more susceptible to the amount, quality and range of the data used, this is of paramount importance and should be carefully considered before building such models.

Acknowledgement

This project has received funding from the EPSRC project (EP/P016650/1,P65332).

References

Adesanya, V.O., Davey, M.P., Scott, S.A., Smith, A.G., 2014. Kinetic modelling of growth and storage molecule production in microalgae under mixotrophic and autotrophic conditions. Bioresour. Technol. 157, 293–304.

Aksoy, S., Haralick, R.M., 2001. Feature normalization and likelihood-based similarity measures for image retrieval. Pattern recognition letters 22, 563–582.

Baroi, C., Dalai, A.K., 2014. Review on Biodiesel Production from Various Feedstocks Using 12-Tungstophosphoric Acid (TPA) as a Solid Acid Catalyst Precursor. Ind. Eng. Chem. Res. 53, 18611–18624.

Bradford, E., Imsland, L., 2018. Stochastic Nonlinear Model Predictive Control Using Gaussian Processes, in: 2018 European Control Conference (ECC), pp. 1027–1034.

Bradford, E., Schweidtmann, A.M., Lapkin, A., 2018. Efficient multiobjective optimization employing Gaussian processes, spectral sampling and a genetic algorithm. Journal of Global Optimization 71, 407–438.

Brahim-Belhouari, S., Bermak, A., 2004. Gaussian process for nonstationary time series prediction. Comput. Stat. Data Anal. 47, 705–712.

Brennan, L., Owende, P., 2010. Biofuels from microalgae-A review of technologies for production, processing, and extractions of biofuels and co-products. Renew. Sustain. Energy Rev. 14, 557–577.

Chu, W.l., 2012. Biotechnological applications of microalgae. Int. e-Journal Sci. Med. Educ. 6, 24–37.

Deisenroth, M., Rasmussen, C.E., 2011. PILCO: A model-based and data-efficient approach to policy search, in: Proc. 28th Int. Conf. Mach. Learn., pp. 465–472.

Deisenroth, M.P., 2010. Efficient Reinforcement Learning using Gaussian Processes. KIT Scientific Publishing 9.

Deisenroth, M.P., Huber, M.F., Hanebeck, U.D., 2009. Analytic moment-based Gaussian process filtering, in: Proc. 26th Annu. Int. Conf. Mach. Learn., ACM. pp. 225–232.

Ebden, M., 2015. Gaussian processes: A quick introduction. arXiv Prepr. arXiv1505.02965 .

Fábregas, J., Otero, A., Maseda, A., Domínguez, A., 2001. Two-stage cultures for the production of Astaxanthin from Haematococcus pluvialis. J. Biotechnol. 89, 65–71.

Feng, Y., Barr, W., Harper, W.F., 2013. Neural network processing of microbial fuel cell signals for the identification of chemicals present in water. J. Environ. Manage. 120, 84–92.

García-Camacho, F., López-Rosales, L., Sánchez-Mirón, A., Belarbi, E., Chisti, Y., Molina-Grima, E., 2016. Artificial neural network modeling for predicting the growth of the microalga Karlodinium veneficum. Algal Res. 14, 58–64.

Girard, A., Rasmussen, C.E., Candela, J.Q., Murray-Smith, R., 2003. Gaussian process priors with uncertain inputs-application to multiple-step ahead time series forecasting. Adv. Neural Inf. Process. Syst. , 545–552.

Gunay, M.E., Akpinar, F., Onsan, Z.I., Yildirim, R., 2012. Investigation of water gas-shift activity of Pt-MOx-CeO2 Al2O3 using modular artificial neural networks. Int. J. Hydrogen Energy 37, 2094–2102.

Himmelblau, D.M., 2008. Accounts of Experiences in the Application of Artificial Neural Networks in Chemical Engineering. Ind. Eng. Chem. Res. 47, 5782–5796.

Ho, S.H., Xie, Y., Chan, M.C., Liu, C.C., Chen, C.Y., Lee, D.J., Huang, C.C., Chang, J.S., 2015. Effects of nitrogen source availability and bioreactor operating strategies on lutein production with Scenedesmus obliquus FSP-3. Bioresour. Technol. 184, 131–138.

Hosen, M.A., Hussain, M.A., Mjalli, F.S., 2011. Control of polystyrene batch reactors using neural network based model predictive control (NNMPC): An experimental investigation. Control Eng. Pract. 19, 454–467.

Jones, D.R., Schonlau, M., Welch, W.J., 1998. Efficient global optimization of expensive black-box functions. Journal of global optimization 13, 455–492.

Jones, E., Oliphant, T., Peterson, P., Others, 2001. SciPy: Open Source Scientific Tools for Python.

Kocijan, J., Girard, A., Banko, B., Murray-Smith, R., 2005. Dynamic systems identification with Gaussian processes. Math. Comput. Model. Dyn. Syst. 11, 411–424.

Kuddus, M., Singh, P., Thomas, G., Al-Hazimi, A., 2013. Recent developments in production and biotechnological applications of C-phycocyanin. Biomed Res. Int. 2013, 742859.

Malek, A., Zullo, L.C., Daoutidis, P., 2016. Modeling and Dynamic Optimization of Microalgae Cultivation in Outdoor Open Ponds. Ind. Eng. Chem. Res. 55, 3327–3337.

Mata, T.M., Martins, A.A., Caetano, N.S., 2010. Microalgae for biodiesel production and other applications: A review. Renew. Sustain. Energy Rev. 14, 217–232.

Mohamed, M.S., Tan, J.S., Mohamad, R., Mokhtar, M.N., Ariff, A.B., 2013. Comparative Analyses of Response Surface Methodology and Artificial Neural Network on Medium Optimization for Tetraselmis sp. FTC209 Grown under Mixotrophic Condition. Sci. World J. 2013, 1–14.

Mohd Ali, J., Hussain, M.A., Tade, M.O., Zhang, J., 2015. Artificial Intelligence techniques applied as estimator in chemical process systems - A literature survey. Expert Syst. Appl. 42, 5915–5931.

Nasr, N., Hafez, H., El Naggar, M.H., Nakhla, G., 2013. Application of artificial neural networks for modeling of biohydrogen production. Int. J. Hydrogen Energy 38, 3189–3195.

Neal, R.M., 2012. Bayesian learning for neural networks. volume 118. Springer Science {&} Business Media.

Nelofer, R., Ramanan, R.N., Rahman, R.N.Z.R.A., Basri, M., Ariff, A.B., 2012. Comparison of the estimation capabilities of response surface methodology and artificial neural network for the optimization of recombinant lipase production by E. coli BL21. J. Ind. Microbiol. Biotechnol. 39, 243–254.

Ochoa-Estopier, L.M., Jobson, M., Smith, R., 2012. Operational optimization of crude oil distillation systems using artificial neural networks. Comput. Aided Chem. Eng. 30, 982–986.

Odabasi, C., Gunay, M.E., Yildirim, R., 2014. Knowledge extraction for water gas shift reaction over noble metal catalysts from publications in the literature between 2002 and 2012. Int. J. Hydrogen Energy 39, 5733–5746.

O'Hagan, A., Kingman, J.F.C., 1978. Curve fitting and optimal design for prediction. J. R. Stat. Soc. Ser. B , 1–42.

Pareek, V.K., Brungs, M.P., a.a Adesina, Sharma, R., 2002. Artificial neural network modeling of a multiphase photodegradation system. J. Photochem. Photobiol. A Chem. 149, 139–146.

Rasmussen, C.E., 1996. Evaluation of Gaussian processes and other methods for non-linear regression. University of Toronto PhD thesis.

Rasmussen, C.E., Williams, C.K., 2006. Gaussian processes for machine learning. The MIT Press .

Raue, A., Schilling, M., Bachmann, J., Matteson, A., Schelke, M., Kaschek, D., Hug, S., Kreutz, C., Harms, B.D., Theis, F.J., et al., 2013. Lessons learned from quantitative dynamical modeling in systems biology. PloS one 8, e74335.

del Rio-Chanona, E.A., rashid Ahmed, N., Zhang, D., Lu, Y., Jing, K., 2017a. Kinetic modeling and process analysis for Desmodesmus sp. lutein photo-production. AIChE J. 63, 2546–2554.

del Rio-Chanona, E.A., Dechatiwongse, P., Zhang, D., Maitland, G.C., Hellgardt, K., Arellano-Garcia, H., Vassiliadis, V.S., 2015. Optimal Operation Strategy for Biohydrogen Production. Ind. Eng. Chem. Res. 54, 6334–6343.

del Rio-Chanona, E.A., Fiorelli, F., Zhang, D., Ahmed, N.R., Jing, K., Shah, N., 2017b. An efficient model construction strategy to simulate microalgal lutein photo-production dynamic process. Biotechnol. Bioeng. 114, 2518–2527.

del Rio-Chanona, E.A., Manirafasha, E., Zhang, D., Yue, Q., Jing, K., 2016a. Dynamic modeling and optimization of cyanobacterial C-phycocyanin production process by artificial neural network. Algal Res. 13, 7–15.

del Rio-Chanona, E.A., Zhang, D., Vassiliadis, V.S., 2016b. Model-based real-time optimisation of a fed-batch cyanobacterial hydrogen production process using economic model predictive control strategy. Chem. Eng. Sci. 142, 289–298.

Sacks, J., Welch, W.J., Mitchell, T.J., Wynn, H.P., 1989. Design and analysis of computer experiments. Statistical science , 409–423.

Shahriari, B., Swersky, K., Wang, Z., Adams, R.P., de Freitas, N., 2016. Taking the human out of the loop: A review of bayesian optimization. Proceedings of the IEEE 104, 148–175.

Sun, Z., Li, T., Zhou, Z.g., Jiang, Y., 2015. Microalgae as a Source of Lutein: Chemistry, Biosynthesis, and Carotenogenesis, in: Adv. Biochem. Eng.. volume 153, pp. 37–58.

Sundararajan, S., Keerthi, S.S., 2001. Predictive approaches for choosing hyperparameters in Gaussian processes. Neural Comput. 13, 1103–1118.

Tamburic, B., Zemichael, F.W., Maitland, G.C., Hellgardt, K., 2011. Parameters affecting the growth and hydrogen production of the green alga Chlamydomonas reinhardtii. Int. J. Hydrogen Energy 36, 7872–7876.

Urtasun, R., Fleet, D.J., Fua, P., 2006. 3D people tracking with Gaussian process dynamical models, in: Comput. Vis. Pattern Recognition, 2006 IEEE Comput. Soc. Conf., IEEE. pp. 238–245.

Vats, S., Negi, S., 2013. Use of artificial neural network (ANN) for the development of bioprocess using Pinus roxburghii fallen foliages for the release of polyphenols and reducing sugars. Bioresour. Technol. 140, 392–398.

Wang, J.M., Fleet, D.J., Hertzmann, A., 2005. Gaussian process dynamical models, in: NIPS, p. 3.

Xie, Y., Ho, S.H., Chen, C.N.N., Chen, C.Y., Ng, I.S., Jing, K.J., Chang, J.S., Lu, Y., 2013. Phototrophic cultivation of a thermo-tolerant Desmodesmus sp. for lutein production: effects of nitrate concentration, light intensity and fed-batch operation. Bioresour. Technol. 144, 435–44.

Xiong, Z., Zhang, J., 2004. Modelling and optimal control of fed-batch processes using a novel control affine feedforward neural network. Neurocomputing 61, 317–337.

Yen, H.W., Sun, C.H., Ma, T.W., 2011. The Comparison of Lutein Production by Scenesdesmus sp. in the Autotrophic and the Mixotrophic Cultivation. Appl. Biochem. Biotechnol. 164, 353–361.

Zhang, D., Dechatiwongse, P., Del-Rio-Chanona, E.A., Hellgardt, K., Maitland, G.C., Vassiliadis, V.S., 2015a. Analysis of the cyanobacterial hydrogen photoproduction process via model identification and process simulation. Chem. Eng. Sci. 128, 130–146.

Zhang, D., Xiao, N., Mahbubani, K., del Rio-Chanona, E., Slater, N., Vassiliadis, V., 2015b. Bioprocess modelling of biohydrogen production by Rhodopseudomonas palustris: Model development and effects of operating conditions on hydrogen yield and glycerol conversion efficiency. Chem. Eng. Sci. 130, 68–78.