# Incorporating knowledge about model structure in the identification of Gaussian-process models

JUŠ KOCIJAN
Jozef Stefan Institute & University of Nova Gorica
SLOVENIA
jus.kocijan@ijs.si

*Abstract:* Dynamic system identification with Gaussian-process prior model is a probabilistic, nonparametric modelling method for identification. Gaussian-process models provide, besides the prediction, also the information about prediction uncertainty based on the availability or uncertainty of the data used for the modelling. An advantage of this kind of model is a small number of training parameters, a facilitated structure determination and the possibility to include various sorts of prior knowledge into the model. One of possibilities is to include block-structure knowledge like Hammerstein model structure. The identification procedure of Gaussian-process model with Hammerstein model structure will be presented and illustrated with an example.

*Key–Words:* System identification, Gaussian process models, dynamic systems, Hammerstein model.

## 1 Introduction

Gaussian-process (GP) models [1] form a new, emerging, complementary method in the field of computational intelligence that can be used for dynamic system identification. The GP model is a probabilistic, nonparametric, black-box model that has generated interest in the machine-learning community in the past decade. Because of its properties, for example, the ability of the models to provide a measure of confidence for their prediction, these models are also interesting for solving engineering problems.

The modelling of dynamic systems from data or dynamic systems identification is a widespread engineering tool. It frequently provides, but not always, models of the input-output behaviour that are used for various purposes. Computational intelligence methods have been shown to be very efficient for the modelling of nonlinear systems.

Most of the computational intelligence methods for regression modelling are meant for the modelling of the mapping between input and output data. This is the mapping of a static function between the given input data set and the output or target data set. Dynamics can be introduced into these models if the lagged samples of the input and output signals are fed back and used as regressors. In general, it is the same case with GP models. This method of dynamic systems, black-box identification with GP models is described in, e.g., [2], [3]. The way that GP models handle noisy, uncertain and outlier data as well as model uncertainties [2] is attractive from the engineering point of view.

The aim of this paper is to propose a possible framework for the modelling of a particular type of nonlinear dynamic systems such as GP models, i.e., the Hammerstein models [4].

GP models, even though nonparametric, are well suited for the incorporation of prior knowledge about the modelled system [1]. Prior knowledge in the form of a known structure such as the Wiener and Hammerstein models are attractive for the engineering community. Regardless of the well-established methods relating to this topic, new research and application results emerge continuously (e.g., [5]), because of the facilitated analysis and control design of the nonlinear systems frequently found in practice.

This paper is organized as follows. In the next section the fundamentals of GP modelling are presented. Section 3 presents the GP modelling of Hammerstein models, which is demonstrated in Section 4. The last section summarizes the main results and concludes the paper.

## 2 Gaussian Process Modelling

A GP model is a probabilistic, non-parametric model for the prediction of output-variable distributions. Its use and properties for modelling are thoroughly described in [1]. Here, only a brief description, necessary for the paper's understanding, is given.

The GP is a Gaussian random function, fully described by its mean and variance. GPs can be viewed as a collection of random variables $f(\mathbf{x}_i)$ with a joint multivariate Gaussian distribution: $f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n) \sim \mathcal{N}(0, \mathbf{\Sigma})$. The elements $\mathbf{\Sigma}_{ij}$

ISBN: 978-1-61804-169-2

of the covariance matrix $\mathbf{\Sigma}$ are covariances between the values of the functions $f(\mathbf{x}_i)$ and $f(\mathbf{x}_j)$, and are functions of the corresponding arguments $\mathbf{x}_i$ and $\mathbf{x}_j$: $\Sigma_{ij} = \mathrm{cov}(f(\mathbf{x}_i), f(\mathbf{x}_j)) = C(\mathbf{x}_i, \mathbf{x}_j)$. Any function $C(\mathbf{x}_i, \mathbf{x}_j)$ can be a covariance function, providing it generates a positive, semi-definite, covariance matrix $\mathbf{\Sigma}$.

The covariance function $C(\mathbf{x}_i, \mathbf{x}_j)$ can be interpreted as a measure of the distance between the input points $\mathbf{x}_i$ and $\mathbf{x}_j$. For systems modelling it is usually composed of two main parts:

$$C(\mathbf{x}_i, \mathbf{x}_j) = C_f(\mathbf{x}_i, \mathbf{x}_j) + C_n(\mathbf{x}_i, \mathbf{x}_j) \qquad (1)$$

where $C_f$ represents the functional part and describes the unknown system we are modelling and $C_n$ represents the noise part. The covariance function for the modelling of noise is usually a constant one representing white noise.

Some possible choices for $C_f$ are:

- the *squared exponential or Gaussian covariance function*, which is most frequently used in the functional part

$$\begin{aligned} C_f(\mathbf{x}_i, \mathbf{x}_j) &= v \exp\left[ -\frac{1}{2} \sum_{d=1}^{D} w_d (x_{id} - x_{jd})^2 \right] \\ &= v \exp\left[ -\frac{1}{2} (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{W}^{-1} (\mathbf{x}_i - \mathbf{x}_j) \right], \end{aligned} \qquad (2)$$

  the squared exponential covariance function is used when a smooth and stationary functional part is assumed,

- *linear covariance function*

$$\begin{aligned} C_f(\mathbf{x}_i, \mathbf{x}_j) &= \sum_{d=1}^{D} w_d x_{id} x_{jd} \\ &= \mathbf{x}_i^T \mathbf{W} \mathbf{x}_j, \end{aligned} \qquad (3)$$

  the linear covariance function is used when a linear functional part is assumed.

$\mathbf{\Theta} = [w_1 \dots w_D \; v \; v_0]^T$ are the 'hyperparameters' of the covariance functions, $x_{id}$ and $x_{jd}$ are the $d^{\text{th}}$ components of the input vectors $\mathbf{x}_i, \mathbf{x}_j$ and $D$ is the input dimension.

A regression problem is that from some noisy measurements of a dependent variable, at certain values of the independent variable (or several variables) one tries to find what is the best estimate of the dependent variable at a new value of independent variable (or several variables). In the context of GP regression this means that based on the data $(\mathbf{X}, \mathbf{y})$, and given a new input vector $\mathbf{x}^*$, we wish to find the predictive distribution of the corresponding output $y^*$. The mapping function $f$ between independent variable and dependent variable is sought. Most of the effort in the design of GP model consists in *tuning* the parameters of the covariance function. The hyperparameters are learned, i.e., identified, using the data at hand. This is done by maximisation of the log-likelihood

$$\begin{aligned} &\mathcal{L}(\mathbf{\Theta}) = \log(p(\mathbf{y}|\mathbf{X})) \\ &= -\frac{1}{2} \log(\mid \mathbf{K} \mid) - \frac{1}{2} \mathbf{y}^T \mathbf{K}^{-1} \mathbf{y} - \frac{N}{2} \log(2\pi) \end{aligned} \qquad (4)$$

where $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]$ is a set of $N$ $D$-dimensional input vectors, $\mathbf{y} = [y_1, y_2, \dots, y_N]^T$ is a vector of output data, $\mathbf{\Theta}$ is the vector of hyperparameters and $\mathbf{K}$ is the $N \times N$ training covariance matrix. The calculation of the log-likelihood and its derivatives due to the optimisation algorithm involves the computation of the inverse of the $N \times N$ covariance matrix $\mathbf{K}$ at every iteration, which can become computationally demanding for large $N$. Nevertheless, the number of parameters to be optimised is small, which means that optimisation convergence might be faster and that the 'curse of dimensionality' known in other data-based modelling methods is circumvented or at least decreased.

After the hyperparameters have been estimated, we can obtain a prediction of the GP model at the input $\mathbf{x}_{N+1}$ that is Gaussian distribution with a mean and variance:

$$\mu(\mathbf{x}_{N+1}) = \mathbf{k}^T \mathbf{K}^{-1} \mathbf{y} \qquad (5)$$

$$\sigma^2(\mathbf{x}_{N+1}) = \kappa(\mathbf{x}_{N+1}) - \mathbf{k}^T \mathbf{K}^{-1} \mathbf{k} \qquad (6)$$

where $\mathbf{k} = \mathbf{k}(\mathbf{x}_{N+1}) = [C(\mathbf{x}_1, \mathbf{x}_{N+1}), \dots, C(\mathbf{x}_N, \mathbf{x}_{N+1})]^T$ is the $N \times 1$ vector of covariances between the training inputs and the test input, and $\kappa(\mathbf{x}_{N+1}) = C(\mathbf{x}_{N+1}, \mathbf{x}_{N+1})$ is the autocovariance of the test input.

The vector $\mathbf{k}^T(\mathbf{x}_{N+1})\ \mathbf{K}^{-1}$ in (5) can be interpreted as a vector of smoothing terms that weights the training outputs $\mathbf{y}$ to make a prediction at the test point $\mathbf{x}_{N+1}$. If the new input is far away from the data points, the term $\mathbf{k}^T(\mathbf{x}_{N+1})\ \mathbf{K}^{-1}\ \mathbf{k}(\mathbf{x}_{N+1})$ in (6) will be small, so that the predicted variance $\sigma^2(\mathbf{x}_{N+1})$ will be large. The regions of the input space where there are few data or where the data have high complexity, or are corrupted with noise, are in this way indicated through a higher variance.

## 2.1 Modelling and simulation of dynamic systems

The above modelling procedure was developed for modelling static nonlinearities, but it can be readily

applied for modelling dynamic systems, as shown in [6],[3]. It is important to stress that the model prediction in the form of GP is just an approximation when the Gaussian assumption is not fulfilled, which is in line with common engineering practice. Consider a dynamic system in the ARX representation, where the output at time step $k$ depends on the delayed outputs $y$ and the exogenous control inputs $u$:

$$\begin{aligned} y(k) &= f(y(k-1),\ldots,y(k-L), \\ &\quad u(k-1),\ldots,u(k-L)) + \epsilon(k) \end{aligned} \quad (7)$$

where $\epsilon(k)$ is white noise and the output $y(k)$ depends on the state vector $\mathbf{x}(k) = [y(k-1), y(k-2),\ldots,y(k-L),u(k-1),u(k-2),\ldots,u(k-L)]^T$ at time step $k$. For the discussion on advantages and disadvantages of GP models for the identification of dynamic systems see, e.g., [7].

Assuming the signal is known up to $k$, we wish to predict the output of the system $n$ steps ahead, i.e., we need to find the predictive distribution of $y(k+n)$ corresponding to $\mathbf{x}(k+n)$. Multiple-step-ahead predictions of a system modelled by (7) can be achieved by iteratively making repeated one-step-ahead predictions, up to the desired horizon. A naive way of doing so is, at each time-step, to feed back the mean of the predictive distribution (the estimate of the output) by considering $\mathbf{x}(k+n) = [\hat{y}(k+n-1),\ldots,\hat{y}(k+n-L),u(k+n-1),\ldots,u(k+n-L)]^T$, where $\hat{y}(k+n-i)$ is the point estimate of $y(k+n-i)$. This way of generating multiple-step-ahead predictions is commonly referred to as "output error" or "parallel model" in the identification literature.

In [6] the iterative, multiple-step-ahead prediction is done by feeding back the mean of the predictive distribution as well as the variance of the predictive distribution at each time-step, thus taking the uncertainty attached to each intermediate prediction into account. In this way, each input for which we wish to predict becomes a normally distributed random variable. However, this is still an approximation, as will be explained in the following section.

# 3 GP Modelling Used in the Hammerstein Model

The Hammerstein structure consists of a nonlinear static block followed by a linear dynamic block, as is depicted in Fig. 1. It is a frequently applied, nonlinear dynamic systems modelling approach. This kind of model can be used where the actuator dominates the system behaviour with its nonlinear static characteristics.

The structure of the Hammerstein model can be linearly parameterized, which can be reflected in the choice of regressors when modelling with the GP model. The idea behind this approach is to represent a static nonlinearity with a polynomial approximation and in that case the overall input-output relationship is linear in the parameters. In the case of a GP model identification with a linear covariance function this approach requires the manual or automated selection of polynomial regressors, which are at the same time also regressors of the complete GP Hammerstein model. The identification procedure is mainly composed of regressors' selection, which might be a lengthy operation, and the input-output identification, which are both tightly interconnected.

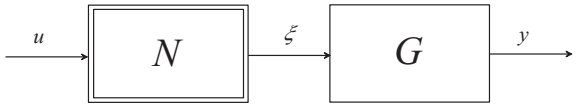


Figure 1: Principial scheme of the Hammerstein model

In the case that the nonlinearity requires a complicated polynomial representation, a two-stage procedure might be an alternative choice. Therefore, we are focusing on the concept of a two-stage procedure with the identification of the static nonlinearity first and the identification of the dynamic part in the next step.

## 3.1 Modelling of the Static Part

There are at least three possible ways to acquire the nonlinear static part of the system to be modelled. The most common in practice is measuring the static characteristics, which means that the equilibrium points of the system are acquired systematically with measurements of the constant-input and constant-output signals in the steady state.

The second approach is that the curve of equilibria is obtained with the training of the GP model for a static mapping from the system input to the system output, but based on samples from variable input and output signals. Most commonly, the majority of the measured samples of input and output signals can be found in the vicinity of the equilibria. The GP model, because of its smoothing abilities [1],[3], will smooth all the excursions away from the equilibria and, provided that the output response is symmetric in the excursions around the equilibrium points, the mean-value predictions will approximate the curve of the equilibria. The mean value of the GP model predictions can therefore be considered as an approximations of the system's nonlinear static part.

In the case that the distributions of measured samples of the input and output signals are spread very

widely over the nonlinear region away from the equilibria, then these samples can be brought closely to equilibria with filtering, or only the samples lying in a selected vicinity of the equilibria may be used.

The nonlinearity is identified with a GP model that contains a covariance function that reflects prior knowledge about the static nonlinearity. If there is no particular prior knowledge, then an assumption about the smoothness and the stationarity might be considered and consequently a squared exponential covariance function (2) selected.

### 3.2 Modelling of the Dynamic Part

When the GP model of static nonlinearity is obtained, the intermediate signal $\hat{\xi}$ can be inferred from the input signal with this model. The output of the GP model is the predictive distribution that will form the input for the linear dynamic part of the Hammerstein model. These predictive distributions can be considered as an uncertain input signal for the linear dynamic part. Therefore, we need to consider the learning of the dynamic GP model with a linear covariance function with inputs that have a random distribution. The assumption is made that the inputs are independent and normally distributed. The derivation for the general covariance function is given firstly, adopted from [8].

In the situation where the inputs are modelled as normally distributed random inputs it can be written

$$y_i \;=\; f(\mathbf{x}_i) \;\text{ with }\; \mathbf{x}_i \sim \mathcal{N}(\mathbf{u}_i, \mathbf{\Sigma}_{\mathbf{x}_i}) \qquad (8)$$

Although the process in the general case is not Gaussian anymore, the mean and covariance function of the random process can still be determined.

The covariances between the outputs can be written as [8]

$$\mathrm{cov}[y_i, y_j|\mathbf{u}_i, \mathbf{u}_j] = \int\int C(\mathbf{x}_i, \mathbf{x}_j) p(\mathbf{x}_i, \mathbf{x}_j) d\mathbf{x}_i d\mathbf{x}_j \qquad (9)$$

where a noise variation for each input, i.e.

$$p(\mathbf{x}_i) = \mathcal{N}_{\mathbf{x}_i}(\mathbf{u}_i, \mathbf{\Sigma}_{\mathbf{x}_i}); p(\mathbf{x}_j) = \mathcal{N}_{\mathbf{x}_j}(\mathbf{u}_j, \mathbf{\Sigma}_{\mathbf{x}_j}) \quad (10)$$

is allowed.

Let $C_n(\mathbf{u}_i, \mathbf{u}_j)$ denote the 'random' covariance function giving the covariance between $y_i$ and $y_j$. Assuming the inputs are independent given their characteristics, it can be defined

$$C_n(\mathbf{u}_i, \mathbf{u}_j) = \int\int C(\mathbf{x}_i, \mathbf{x}_j) p(\mathbf{x}_i) p(\mathbf{x}_j) d\mathbf{x}_i d\mathbf{x}_j \qquad (11)$$

Equation (11) with the linear covariance function is solvable and can be written as

$$\begin{aligned} C_n(\mathbf{u}_i, \mathbf{u}_j) &= \int\int \mathbf{x}_i^T \mathbf{W} \mathbf{x}_j p(\mathbf{x}_i) p(\mathbf{x}_j) d\mathbf{x}_i d\mathbf{x}_j \\ &= \mathbf{u}_i^T \mathbf{W} \mathbf{u}_j \end{aligned} \qquad (12)$$

The obtained result is the same as it is for data learning without the input uncertainty, Eq. (3). This means that the same learning procedure can be pursued.

The validation of the identified GP Hammerstein model is made in two stages and off-line. Firstly, with a prediction of the intermediate signal transferring the input signal through the static nonlinearity and, secondly, with a simulation of the dynamic GP model with a propagation of uncertainty.

The theoretical backgrounds for these predictions and simulations are as follows.

**Prediction at a new random input for a GP model** In this part, we are summarising the results from [9] and [8] of extensions to the GP modelling framework for dealing with random inputs. We first look at making a prediction for a new random input $\mathbf{x}$, when the training inputs are not random, a situation that might arise for instance when making a multiple-step-ahead prediction of a signal by propagation of the uncertainty.

It was explained in Section 2, how based on observed data and on a new input $\mathbf{x}_{N+1}$, the predictive distribution of the corresponding $y_{N+1} = f(\mathbf{x}_{N+1})$ was readily obtained. The index $N+1$ is left out for simplicity in the remaining part of the section. We recall from Section 2 that GP model output is Gaussian with a mean and variance

$$\mu(\mathbf{x}) = \mathbf{k}(\mathbf{x})^T \mathbf{b} = \sum_{i=1}^{N} \beta_i C(\mathbf{x}, \mathbf{x}_i) \qquad (13)$$

$$\begin{aligned} \sigma^2(\mathbf{x}) &= C(\mathbf{x}, \mathbf{x}) - \mathbf{k}(\mathbf{x})^T \mathbf{K}^{-1} \mathbf{k}(\mathbf{x}) \\ &= C(\mathbf{x}, \mathbf{x}) - \sum_{i,j=1}^{N} K_{ij}^{-1} C(\mathbf{x}, \mathbf{x}_i) C(\mathbf{x}, \mathbf{x}_j) \end{aligned} \qquad (14)$$

where $\mathbf{b} = \mathbf{K}^{-1}\mathbf{y}$.

If we now wish to make a prediction at $\mathbf{x} \sim \mathcal{N}_{\mathbf{x}}(\mathbf{u}, \mathbf{\Sigma}_x)$, where $\mathbf{u} = E[\mathbf{x}]$ and $\mathbf{\Sigma}_x = \mathrm{var}[\mathbf{x}]$, we need to integrate the predictive distribution over the possible $\mathbf{x}$'s, that is

$$p(f(\mathbf{x})|\mathcal{D}, \mathbf{u}, \mathbf{\Sigma}_x) = \int p(f(\mathbf{x})|\mathcal{D}, \mathbf{x}) p(\mathbf{x}) d\mathbf{x} \qquad (15)$$

where $p(\mathbf{x}) = \mathcal{N}_{\mathbf{x}}(\mathbf{u}, \mathbf{\Sigma}_x)$ and $p(f(\mathbf{x})|\mathcal{D}, \mathbf{x})$ has a mean $\mu(\mathbf{x})$ and a variance $\sigma^2(\mathbf{x})$.

As $p(f(\mathbf{x})|\mathcal{D}, \mathbf{x})$ is a nonlinear function of $\mathbf{x}$, this integral cannot be solved analytically without an approximation.

ISBN: 978-1-61804-169-2

One way of solving this integral is to go for a *numerical approximation*, that is

$$p(f(\mathbf{x})|\mathcal{D},\mathbf{u},\mathbf{\Sigma}_x) \simeq \frac{1}{T}\sum_{t=1}^{T} p(f(\mathbf{x})|\mathcal{D},\mathbf{x}^t) \qquad (16)$$

where $\mathbf{x}^t$ is a sample from $p(\mathbf{x})$. This can be done using MCMC methods.

An alternative is *analytical approximation*, more precisely a Gaussian analytical approximation, that is, in computing the mean and variance of $p(f(\mathbf{x})|\mathcal{D},\mathbf{u},\mathbf{\Sigma}_x)$ only. This is the approximation we are going to use.

The expressions for the mean and variance are:

$$m(\mathbf{u},\mathbf{\Sigma}_x) = E_{\mathbf{x}}[\mu(\mathbf{x})] \qquad (17)$$

where $m(\mathbf{u},\mathbf{\Sigma}_x)$ is the expectation of $y|\mathcal{D},\mathbf{u},\mathbf{\Sigma}_x$ and

$$v(\mathbf{u},\mathbf{\Sigma}_x) = E_{\mathbf{x}}[\sigma^2(\mathbf{x})] + E_{\mathbf{x}}[\mu(\mathbf{x})^2] - (E_{\mathbf{x}}[\mu(\mathbf{x})])^2 \qquad (18)$$

where $v(\mathbf{u},\mathbf{\Sigma}_x)$ is the variance of $f(\mathbf{x})|\mathcal{D},\mathbf{u},\mathbf{\Sigma}_x$.

**The special case of the linear covariance function** This derivation is necessary for the GP model simulation of the dynamic part, which is modelled with the GP model containing a linear covariance function. The exact derivations can be found in [9]. Here we are presenting just the final results.

The linear covariance function is considered given by (3). According to (17), the *new predictive mean* is

$$\begin{aligned} m(\mathbf{u},\mathbf{\Sigma}_x) &= E_{\mathbf{x}}[\mu(\mathbf{x})] \\ &= \mathbf{u}(\mathbf{XW})^T(\mathbf{XWX}^T)^{-1}\mathbf{y} \end{aligned} \qquad (19)$$

According to (18), the variance is given by

$$\begin{aligned} v(\mathbf{u},\mathbf{\Sigma}_x) &= E_{\mathbf{x}}[\sigma^2(\mathbf{x})] + E_{\mathbf{x}}[\mu(\mathbf{x})^2] - (E_{\mathbf{x}}[\mu(\mathbf{x})])^2 \\ &= \mathbf{u}\alpha\mathbf{u}^T + \mathrm{Tr}[\alpha\mathbf{\Sigma}_{\mathbf{x}}] + \mathrm{Tr}[\gamma\mathbf{\Sigma}_{\mathbf{x}}] \end{aligned} \qquad (20)$$

where $\alpha = \mathbf{W} - (\mathbf{XW})^T(\mathbf{XWX}^T)^{-1}(\mathbf{XW})$. $\gamma = (\mathbf{XW})^T\mathbf{b}\mathbf{b}^T(\mathbf{XW})$.

As already stated, the simulation of the dynamic system is pursued as a repeated prediction and the feedback of the Gaussian random variable resulting in an $L \times 1$ input into the model $\mathbf{x}(k+n) = [y(k+n-1),\ldots,y(k+n-L)]^T \sim \mathcal{N}(\mu_{\mathbf{x}},\mathbf{\Sigma}_{\mathbf{x}})$ at each time-step with the mean

$$\mu_{\mathbf{x}} = \begin{bmatrix} m(\mathbf{x}(k+n-1)) \\ \vdots \\ m(\mathbf{x}(k+n-L) \end{bmatrix} \qquad (21)$$

and the covariance matrix

$$\mathbf{\Sigma}_{\mathbf{x}} = \begin{bmatrix} v(\mathbf{x}(k+n-1))+v_0 & .. & \mathrm{cov}(y(k+n-L),y(k+n-1)) \\ \vdots & \vdots & \vdots \\ \mathrm{cov}(y(k+n-1),y(k+n-L)) & .. & v(\mathbf{x}(k+n-L))+v_0 \end{bmatrix}, \qquad (22)$$

where $m(.)$ and $v(.)$ are computed using equations (19) and (20).

In general, at time sample $k+l$, we have the random input vector $\mathbf{x}(k+l) = [y(k+l-1),\ldots,y(k+l-L)]^T \sim \mathcal{N}(\mu_{\mathbf{x}},\mathbf{\Sigma}_{\mathbf{x}})$ with the mean $\mu_{\mathbf{x}}$ formed by the mean of the predictive distribution of the lagged outputs $y(k+l-\tau)$, $\tau = 1,\ldots,L$, given by (19) and the diagonal elements of the $L \times L$ input covariance matrix $\mathbf{\Sigma}_{\mathbf{x}}$ containing the corresponding predictive covariances. The cross-covariance terms $\mathrm{cov}[y(k+l-i),y(k+l-j)]$, for $i,j = 1\ldots L$ with $i \neq j$, are obtained by computing $\mathrm{cov}[y(k+l),\mathbf{x}(k+l)]$, disregarding the last, the oldest element of $\mathbf{x}(k+l)$:

$$\begin{aligned} &\mathrm{cov}[y(k+l),\mathbf{x}(k+l)] = \\ &= E[y(k+l)\mathbf{x}(k+l)] - E[y(k+l)]E[\mathbf{x}(k+l)] \\ &= \mathbf{w}\mathbf{\Sigma}_{\mathbf{x}}. \end{aligned} \qquad (23)$$

For a detailed derivation see [8].

The next section shows an illustrative example where the presented procedures are used for an identification of the GP Hammerstein model.

# 4 Illustrative Example

The nonlinear system that is used in this section for an illustration of the Hammerstein model identification in the framework of the GP model is composed of the static nonlinearity

$$\xi = \frac{u}{\sqrt{0.1+0.9u^2}} \qquad (24)$$

and the consequent linear dynamic part

$$\begin{aligned} y(k+1) &= 1.4138y(k) - 0.6065y(k-1) \\ &+ 0.1044\xi(k) + 0.0883\xi(k-1) \end{aligned} \qquad (25)$$

The sampling time is one time unit. The input signal $u(k)$ is a random signal with a hold time, i.e., the period of time for which the signal stays constant, of 30 time units.

The two-stage identification procedure is divided into the four steps.

**The modelling of the static part**

The nonlinearity is assumed to be continuous and smooth. The samples from the output and input signals were filtered and then identified with a GP model

with a squared exponential covariance function.

**The compensation of the static nonlinearity and the generation of the input signal for the linear system identification**

Input data for the identification of the linear dynamic part, i.e., samples from $\hat{\xi}$, are obtained as predictions of the static GP model with samples from the input signal $u(t)$ at its input.

**The identification of the dynamic linear part**

The obtained input signal and corresponding output signal are utilized for the training of the GP model with a linear covariance function.

**The validation of the identified model with a simulation**

The validation with an input signal different than the one used for the identification is made. A segment of the Hammerstein model response is given in Fig. 2. Fig. 2 shows a satisfactory input-output response

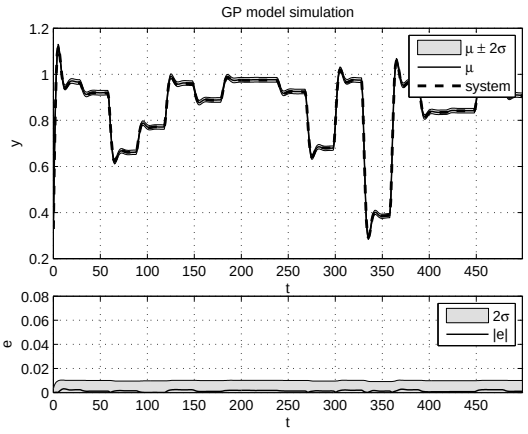


Figure 2: GP model response on a validation signal and a comparison with the system response (top figure) and the absolute value of the model residuals with a 95% confidence band (bottom figure) - the segment of the first 500 samples.

of the GP-based Hammerstein model on a validation signal.

# 5 Conclusion

This paper presents an identification of the Hammerstein models within a GP framework. When modelling the GP Hammerstein models the learning part is pursued with uncertain inputs that are obtained from a nonlinearity compensation, but it does not change the expressions for predicted outputs. The simulation of the GP Hammerstein model has to be pursued with the uncertainty of the inputs and outputs propagated through the dynamic part of the model.

The modelling procedure has been illustrated with a simple example. The obtained model provided output distributions that can be considered as measures of the prediction uncertainty, but also as measures of the confidence in the prediction.

The GP Hammerstein model can be used for the design of robust nonlinear control and other designs where these kinds of nonlinear models with information about the uncertainty can be utilized. The GP framework offers a potentially useful tool for modelling input-output systems. Its potential for the field of engineering, however, remains to be fully explored.

**Acknowledgements:** This work was supported by grant P2-0001 of the Slovenian Research Agency.

*References:*

[1] C. E. Rasmusen and C. K. I. Williams. *Gaussian Processes for Machine Learning*, Cambridge: MIT Press, 2006.

[2] K. Ažman, J. Kocijan, "Application of Gaussian processes for black-box modelling of biosystems", *ISA Transactions*, 46, pp. 443–457, 2007.

[3] J. Kocijan, A. Girard, B. Banko and R. Murray-Smith, "Dynamic Systems Identification with Gaussian Processes", *Mathematical and Computer Modelling of Dynamic Systems*, 11, pp. 411–424, 2005.

[4] R. Haber, H. Unbehauen, "Structure identification of nonlinear dynamic systems: A survey on input/output approaches", *Automatica*, 26(4), 651–677, 1990.

[5] F. Giri, E.-W. Bai, (Eds.) *Block-oriented Nonlinear System Identification*, Springer, 2010.

[6] A. Girard, C. E. Rasmussen, J. Quiñonero Candela, R. Murray-Smith, "Multi-step ahead prediction for non linear dynamic sytems - A Gaussian Process treatment with propagation of the uncertainty", *NIPS* 15, Vancouver, Canada, MIT Press, 2003.

[7] E.-W. Bai, "Prediction eror adjusted Gaussian process for nonlinear non-parametric system Identification", *16th IFAC Symposium on System Identification*, Brussels, Belgium, IFAC, 2012, 101–106.

[8] A. Girard, "Approximate Methods for Propagation of Uncertainty with Gaussian Process Models". PhD thesis, University of Glasgow, Glasgow (2004). URL: www.dcs.gla.ac.uk/ rod/publications/ Gir04.pdf

[9] J. Kocijan, A. Girard, and D. J. Leith. Incorporating linear local models in Gaussian process model. Technical Report DP-8895, Institut Jozef Stefan, Ljubljana, December 2003. URL: dsc.ijs.si/jus.kocijan/GPdyn/2003/ KocGirLei03.pdf