DYNAMIC SYSTEMS IDENTIFICATION WITH GAUSSIAN PROCESSES

J. Kocijan[1,2], A. Girard,[3], B. Banko[1], R. Murray-Smith[3,4]
[1]**Jozef Stefan Institute, Ljubljana, Slovenia**
[2]**Nova Gorica Polytechnic, Nova Gorica, Slovenia**
[3]**University of Glasgow, Glasgow, United Kingdom**
[4]**Hamilton Institute, NUI Maynooth, Ireland**
Corresponding Author: J. Kocijan
Jozef Stefan Institute, Jamova 39
SI-1000 Ljubljana, Slovenia
Phone: +386 1 4773 661, Fax: +386 1 4257 009
email: jus.kocijan@ijs.si

Abstract. In this paper a novel approach for black-box identification of non-linear dynamic systems is described. The Gaussian process prior approach is a statistical model, representative of probabilistic non-parametric modelling approaches. It offers more insight in variance of obtained model response, as well as fewer parameters to determine than other models. The Gaussian processes can highlight areas of the input space where prediction quality is poor, due to the lack of data or its complexity, by indicating the higher variance of the predicted mean. The Gaussian process modelling technique is demonstrated on a simulated example of a non-linear system.

1. Introduction

Most control engineering applications are still based on parametric models, where the functional form is fully described by a finite number of parameters. The information about uncertainty is usually expressed as uncertainty of parameters and does not take into account uncertainty about model structure, or distance of current prediction point from training data used to estimate parameters. This paper describes modelling based on Gaussian processes which is an example of a non-parametric model that gives also the information about prediction uncertainties which are difficult to evaluate appropriately in nonlinear parametric models. Gaussian processe approaches can be applied to many of the problems currently modelled by artificial neural networks.
The use of Gaussian processes to tackle many of the standard problems usually solved by artificial neural networks has been introduced recently e.g.[9]. It was shown that neural networks and Gaussian processes were closely related, in the limit of an infinite number of neurons in hidden layer [7]. Nevertheless, the majority of work on Gaussian processes shown up to now considers modelling of static non-linearities. Fragments on the use of Gaussian processes in modelling dynamic systems have been published recently, e.g. [4,3,2,6,10] and propagation of variance in dynamic systems has just been presented in [1]. The purpose of this paper is to bring aspects from these recent contributions together with an illustrative example, as a brief tutorial example on dynamic systems identification by means of Gaussian process models.
The paper is organised as follows. In the next section, we introduce the use of Gaussian processes for modelling static systems. How this approach can be used for dynamic systems identification is described in Section 3. An illustrative example in Section 4 presents an application of the method. The last section gives some concluding remarks.

2. Modelling with Gaussian process

A Gaussian process [8] is an example of the use of a flexible probabilistic non-parametric model with uncertainty predictions. Its use and properties for modelling are given in [11]. A Gaussian process is a collection of random variables which have a joint multivariate Gaussian distribution: $f(x^1), \ldots, f(x^n) \sim \mathcal{N}(0, \Sigma)$, where Σ_{pq} gives the covariance between points x^p and x^q. Mean $\mu(f(x^p))$, which can be removed ($\mu(f(x^p)) = 0$), and covariance function $\Sigma_{pq} = \mathrm{Cov}(x^p, x^q)$ determine a Gaussian process. Assuming a relationship of the form $y = f(x)$ between the inputs x and outputs y, we have $\mathrm{Cov}(y^p, y^q) = C(x^p, x^q)$, where $C(.,.)$ is some function with the property that it generates a positive definite covariance matrix. This means that the covariance between the variables that represent the outputs for cases number p and

q is a function of the inputs corresponding to the same cases p and q. In general, a stationary (depends only on the distance between points in the input space[1]) Gaussian processes can be effectively used for identification of static nonlinear regression model which is described below.
Consider a set of N D-dimensional vectors containing noisy input data $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_D]$ and a vector of output data $\mathbf{y} = [y(1), y(2), \ldots, y(N)]^T$ representing the static system. The aim is to construct the model, namely function $f(\cdot)$ depending on $\mathbf{X}$ and $\mathbf{y}$, and than at some new input vector $\mathbf{x}^* = [x_1(N+1), x_2(N+1), \ldots, x_d(N+1)]$ find the distribution of the corresponding output $y(N+1)$. The model is determined according to $f(.)$, $\mathbf{X}$ and $\mathbf{y}$ and not on parameter determination within fixed model structure. That is why this is a probabilistic non-parametric approach. The probability of hypothesis $f(\mathbf{x}^*)$ according on data set $\mathbf{X}$ and $\mathbf{y}$ can be written as

$$p(f(\mathbf{x}^*) \mid \mathbf{X}, \mathbf{y}) = \frac{p(\mathbf{y} \mid f(\mathbf{x}^*, \mathbf{X}))p(f(\mathbf{x}^*))}{P(\mathbf{y} \mid \mathbf{X})} \tag{1}$$

$p(\mathbf{y} \mid f(\mathbf{x}^*, \mathbf{X}))$ is the conditional likelihood of model and represents model output in the form of mean and variance. $p(f(\mathbf{x}^*))$ represents prior knowledge contained in the model. Based on the covariance function, the parameters of which (the so called hyperparameters) are determined from training set $\mathbf{X}, \mathbf{y}$, the *a posteriori* value $y(N+1)$ can be determined.
An appropriate covariance function has to be chosen for model identification. Any choice of the covariance function, which will generate a non-negative definite covariance matrix for any set of input points, can be chosen. A common choice is

$$C(x^p, x^q) \;=\; v_1 \exp\left[-\frac{1}{2}\sum_{d=1}^{D} w_d (x_d^p - x_d^q)^2\right] + v_0 \tag{2}$$

where $v_0, v_1, w_d, d = 1, \ldots, D$ are hyperparameters of covariance functions and D is the input dimension. Other forms of covariance functions suitable for different applications can be found in [9], however it is necessary to point out that selection of covariance functions suitable for robust generalisation in typical dynamic systems applications is still an area open for research. Given a set of training cases the hyperparameters of the covariance function $\Theta = [w_1 \ldots w_D \; v_0 \; v_1]^T$ should be learned (identified). There is a hyperparameter correponding to each regressor 'component' so that, after the learning, if a hyperparameter is zero or near zero it means that the corresponding regressor 'component' has little impact and could be removed.
Covariance functions hyperparameters are obtained from training set by maximisation of the likelihood $p(f(\mathbf{x}^*) \mid \mathbf{X}, \mathbf{y})$. Since the analytic solution is very difficult to obtain other approaches are in place. The description of one possible approach follows.
Calculation of model output is straightforward for a given covariance function. It can be seen from equation (1) that posteriori probability depends on hyperparameters through likelihood $p(\mathbf{y} \mid f(\mathbf{x}^*), \mathbf{X})$. Its logarithm can be derived analytically.

$$\mathcal{L}(\boldsymbol{\Theta}) = \log(p(\mathbf{y} \mid f(\mathbf{x}^*, \mathbf{X})) = -\frac{1}{2}\log(\mid \mathbf{K} \mid) - \frac{1}{2}\mathbf{y}^T\mathbf{K}^{-1}\mathbf{y} - \frac{N}{2}\log(2\pi) \tag{3}$$

where $\mathbf{y}$ is the $N \times 1$ vector of training targets and $\mathbf{K}$ is the $N \times N$ training covariance matrix.
The partial derivative of equation (3) for hyperparameters Θ_i is

$$\frac{\partial \mathcal{L}(\boldsymbol{\Theta})}{\partial \Theta_i} = -\frac{1}{2}\mathrm{trace}\left(\mathbf{K}^{-1}\frac{\partial \mathbf{K}}{\partial \Theta_i}\right) + \frac{1}{2}\mathbf{y}^T\mathbf{K}^{-1}\frac{\partial \mathbf{K}}{\partial \Theta_i}\mathbf{K}^{-1}\mathbf{y} \tag{4}$$

The approach where hyperparameters are obtained with minimisation of negative value $\mathcal{L}$ is known as maximum likelihood method. Any optimisation method can be used for the described minimisation. Nevertheless, it has to be kept in mind that the approach is computationally relatively demanding since inverse covariance matrix has to be calculated in every iteration.
MCMC (Markov Chain Monte Carlo) approaches to numerical integration [9] provide an alternative to optimisation.
The described approach can be easily utilised for regression calculation. Based on training set $\mathbf{X}$ a covariance matrix $\mathbf{K}_N$ of order $N \times N$ is determined. As already mentioned before the aim is to find the

[1] Points close together are more correlated than points far apart – a smoothness assumption.

distribution of the corresponding output $y(N+1)$ at some new input vector $\mathbf{x}^* = [x_1(N+1), x_2(N+1), \ldots, x_D(N+1)]^T$. This means that for new input vector $\mathbf{x}^*$, a new covariance matrix $\mathbf{K}_{N+1}$ or order $(N+1) \times (N+1)$ is calculated in form

$$\mathbf{K}_{N+1} = \begin{bmatrix} \begin{bmatrix} & \mathbf{K}_N & \end{bmatrix} & \begin{bmatrix} \mathbf{k}(\mathbf{x}^*) \\ \\ \end{bmatrix} \\ \begin{bmatrix} \mathbf{k}(\mathbf{x}^*)^T \end{bmatrix} & \begin{bmatrix} k(\mathbf{x}^*) \end{bmatrix} \end{bmatrix} \tag{5}$$

where $\mathbf{k}(\mathbf{x}^*) = [C(\mathbf{x}(1), \mathbf{x}^*), \ldots, C(\mathbf{x}(N), \mathbf{x}^*)]^T$ is the $N \times 1$ vector of covariances between the test and training cases and $k(\mathbf{x}^*) = C(\mathbf{x}^*, \mathbf{x}^*)$ is the variance of the new test case.
A prediction at point $y(N+1)$ is also a Gaussian process (Figure 1).

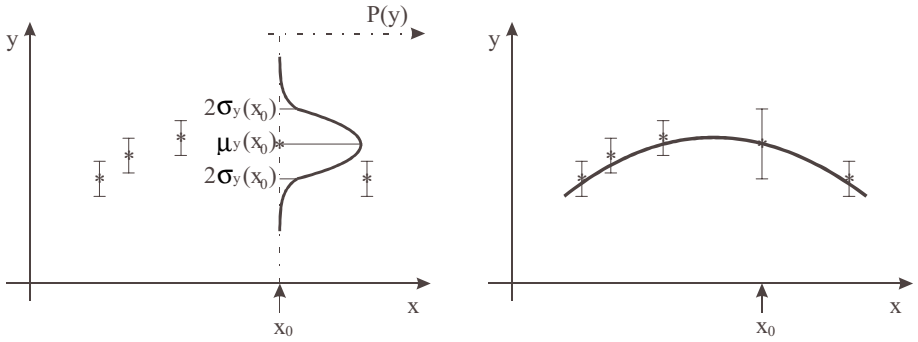

Figure 1: The illustration of a posteriori value determination from Gaussian process model at input value x_0: the output of model is a Gaussian process (left figure) that can be represented by its mean and variance (right figure)

For a new test input $\mathbf{x}^*$, the predictive distribution of the corresponding output is $\hat{y}(N+1)|\mathbf{x}^* \sim \mathcal{N}(\mu(\mathbf{x}^*), \sigma^2(\mathbf{x}^*))$ with

$$\mu(\mathbf{x}^*) = \mathbf{k}(\mathbf{x}^*)^T \, K^{-1} \, \mathbf{y} \tag{6}$$

$$\sigma^2(\mathbf{x}^*) = k(\mathbf{x}^*) - \mathbf{k}(\mathbf{x}^*)^T \, \mathbf{K}^{-1} \, \mathbf{k}(\mathbf{x}^*) + v_0 \tag{7}$$

For k-step ahead prediction we have to take account of the uncertainty of future predictions which provide the 'inputs' for estimating furhter means and uncertainties. We can use a Gaussian approximation to the uncertainty of inputs. The predictive distribution of the corresponding output at the random input x^* is $\mathcal{N}(m(x^*), v(x^*))$ where $m(\mathbf{x}^*)$ and $v(\mathbf{x}^*)$ are approximations of $\mu(\mathbf{x}^*)$ and $\sigma^2(\mathbf{x}^*)$.

$$\begin{aligned} m(\mathbf{x}^*) &= E_{\mathbf{x}^*}[\mu(\mathbf{x}^*)] \\ &\approx \mathbf{k}(\mu(\mathbf{x}^*)^T \mathbf{K}^{-1} \mathbf{y} \end{aligned} \tag{8}$$

$$\begin{aligned} v(\mathbf{x}^*) &= E_{\mathbf{x}^*}[\sigma^2(\mathbf{x}^*)] + \mathrm{var}_{\mathbf{x}^*}(\mu(\mathbf{x}^*)) \\ &\approx \sigma^2(\mu(\mathbf{x}^*)) + \mathrm{trace}\left\{ \Sigma_{\mathbf{x}^*} \left(\frac{1}{2} \frac{\partial^2 \sigma^2(\mathbf{x}^*)}{\partial \mathbf{x}^* \partial \mathbf{x}^{*T}} \Big|_{\mathbf{x}^*=\mu(\mathbf{x}^*)} + \frac{\partial \mu(\mathbf{x}^*)}{\partial \mathbf{x}^*} \Big|_{\mathbf{x}^*=\mu(\mathbf{x}^*)} \frac{\partial \mu(\mathbf{x}^*)}{\partial \mathbf{x}^*} \Big|^T_{\mathbf{x}^*=\mu(\mathbf{x}^*)} \right) \right\} \end{aligned} \tag{9}$$

For more detailed derivation see [1].

3. Dynamic systems identification

Gaussian processes can, like neural networks, be used to model static nonlinearities and can therefore be used for modelling of dynamic systems if delayed input and output signals are fed back and used as

regressors. In such cases an autoregressive model is considered, such that the current output depends on previous outputs, as well as on previous control inputs.

$$\begin{aligned} \mathbf{x}(k) &= [y(k-1), y(k-2), \ldots, y(k-L), u(k-1), u(k-2), \ldots, u(k-L)]^T \\ y(k) &= f(\mathbf{x}(k)) + \epsilon \end{aligned} \tag{10}$$

Where k denotes consecutive number of data sample. Let $\mathbf{x}$ denote the state vector composed of the previous outputs y and inputs u up to a given lag L and ϵ is white noise. We wish to make k-step ahead predictions. Currently, in the framework of Gaussian processes, this has been achieved by either training the model to learn *how* to make k-step ahead predictions (*direct method*) or by simulating the system (repeated one-step ahead predictions up to k - *iterative method*). That is, at each time step, by feeding back the mean prediction (estimate of the output) and its variance as it is illustrated in Figure 2. This corresponds to

$$y(k) = f(\hat{y}(k-1), \hat{y}(k-2), \ldots, \hat{y}(k-L), u(k-1), u(k-2), \ldots, u(k-L)) \tag{11}$$

where $\hat{y}$ denotes the estimate.

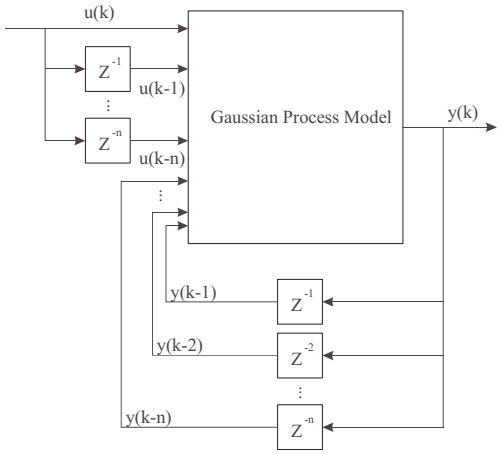

Figure 2: Block scheme of dynamical system simulation with iterative method

The iterative approach is preferred to the direct method because it provides us with predictions for any k-step ahead, unlike the direct method which is only valid for the k-step ahead points.
Using the model (10) and assuming the data is known up to time step i the prediction of y at $k+i$ is computed via

$$\begin{aligned} \mathbf{x}(k+i) &\sim \mathcal{N}\left(\begin{bmatrix} m(\mathbf{x}(k+i-1)) \\ \vdots \\ m(\mathbf{x}(k+i-L)) \end{bmatrix}, \begin{bmatrix} v(\mathbf{x}(k+i-1)) + v_0 & \cdots & \mathrm{cov}(y(k+i-1), u(k+1-L)) \\ \vdots & \vdots & \vdots \\ \mathrm{cov}(u(k+i-L), y(k+1-1)) & \cdots & v(\mathbf{x}(k+i-L)) + v_0 \end{bmatrix} \right) \\ y(k+i) &\sim \mathcal{N}(m(\mathbf{x}(k+i)), v(\mathbf{x}(k+i)) + v_0) \end{aligned} \tag{12}$$

where the point estimates $m(\mathbf{x}(k+i-j)); j = 1, \ldots, L$ are computed using equation (8) and variances $v(\mathbf{x}(k+i-j)); j = 1, \ldots, L$ associated to each $\hat{y}$ are computed using equation (9). It is worthwhile noting that derivatives of mean and variances can be calculated in straightforward manner. For more details see [1].
As can be seen from the presented relations the obtained model does not describe only the dynamic characteristics of non-linear system, but at the same time provides also information about the confidence in these predictions. The Gaussian process can highlight such areas of the input space where prediction quality is poor, due to the lack of data or its complexity, by indicating the higher variance of the predicted mean.

4. Example

The described approach is illustrated with identification of a system that is described by the equation

$$\dot{y} = -\tanh(y + u^3) \tag{13}$$

with output signal y and input signal u. The output signal was disturbed with white noise of variance 0.0025 and zero mean. In our case added noise was white, if noise is correlated then the covariance function (2) can be modified as shown in [5]. Data sampling time, determined according to system dynamics, was selected to be 0.5 units. Input signal was generated by a random number generator with normal distribution and rate of 3 units in the range between -1.3 and 1.3. The number of input signal samples determines dimensions of covariance matrix. To avoid excessive computation time it is sensible to choose number of samples to be no more than a couple of hundred samples. In our case 200 samples have been used for identification.
Input, and output signals and these two signals delayed for one sample were chosen as regressors. The selected model can therefore be written in the form

$$y(k+1) = f(y(k), u(k)) \tag{14}$$

where function $f(\cdot)$ represents the Gaussian process model as a two dimensional regression model. Since the system in equation (13), as well as its discrete equivalent, have order one it is reasonable to expect that the identified model would also be of the system order, because the order of model spans from the order of identified system itself. Some extra identification runs with model structure of higher order were also pursued and results confirmed that choice of the first order structure is the most optimal. The covariance function (2) was used for the model identification and the maximum likelihood framework was used to determine the hyperparameters. The optimization method used for identification of Gaussian process model was in our case a conjugate gradient with line-searches [9] due to its good convergation properties. The following set of hyperparameters was found:

$$\boldsymbol{\Theta} = [w_1, w_2, v_0, v_1] = [0.1312, 0.2948, 6.2618, 0.0045] \tag{15}$$

where hyperparameters w_1 and w_2 allow a weight for each input dimension. The validation signal was also generated by a random number generator with normal distribution and at different rate than for the identification signal. Responses of the system and its Gaussian process model are given in Figures 3 and 4. Gaussian process model responses are shown by means of Gaussian processes mean and double standard deviation (95% confidence interval).

Fitting of the response for validation signal:

- average absolute test error

$$AE = \frac{1}{N} \sum \mid \hat{\mathbf{y}} - \mathbf{y} \mid = 0.028 \tag{16}$$

  where N is the number of used data, $\mathbf{y}$ is the process response (target) in the test set, and $\hat{\mathbf{y}}$ is the simulated value;

- average squared test error

$$SE = \frac{1}{N} \sum (\hat{\mathbf{y}} - \mathbf{y})^2 = 0.0016 \tag{17}$$

- log density error

$$LD = \frac{1}{2N} \sum (\log(2\pi) + \log(\boldsymbol{\sigma}^2) + \frac{(\hat{\mathbf{y}} - \mathbf{y})^2}{\boldsymbol{\sigma}^2}) = -1.6992 \tag{18}$$

  where $\boldsymbol{\sigma}^2$ is a vector of predicted variances.

Results on the validation signal, which was different from the identification one, show that the Gaussian process model successfully models the system based on chosen identification signals. Moreover the information about uncertainty which comes with the Gaussian process model indicate the level to which results are to be trusted.

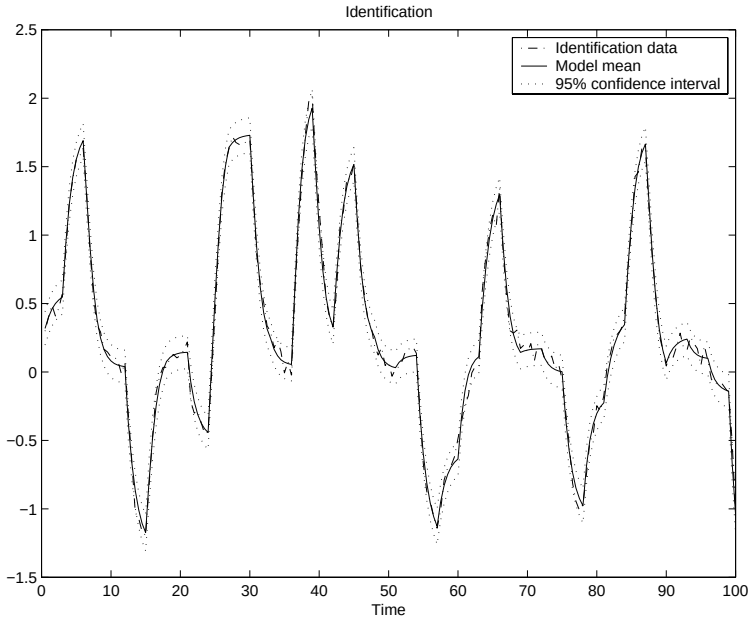

Figure 3: Responses of Gaussian process model (dashed line) and process model (full line) on identification input signal

The noise free discretised version of the true system in equation (13) can be presented as 3-D surface $y(k+1) = f(y(k), u(k))$ as shown in Figure 5. In the same figure the approximation of the surface using Gaussian process model is also given.
Contour and gradient plots of the true system function and its Gaussian process model are depicted in Figure 7. It can be seen from the magnified portions of the plot that the model represents the system well in the region where the model has been trained. Away from that region however the model is no longer a good representation of the true function. This is indicated in the mesh plots of predicted standard deviation, shown in Figure 6, which are low where there was data, but rapidly increase as predictions are made away from the data. Note that the selected region contains a fair portion of nonlinearity.

5. Conclusions

Gaussian process models for the modelling of non-linear systems from input-output data was explained in the paper. This is the approach that is scope of recent work. As with other newly developed approaches number of advantages and disadvantages are yet to be revealed, but some of them are already apparent.

- Modelling with Gaussian process models is probabilistic non-parametric approach to identification, which is relatively new to the control community.
- The approach has some overlap with other more widely used approaches to non-linear systems identification, like the ones with artificial neural networks or fuzzy models. Many similar issues are present like choice of regressors, signals, sampling time, etc. However, there are differences in model structure and obtained information that make the Gaussian process models attractive.
- Only a parsimonious set of hyperparameters needs to be identified. Their number depends on the number of regressors.
- Output of a Gaussian process model to every input data is a Gaussian process determined by its mean value and variance.
- This kind of output data representation contains the level of confidence to obtained output. This is undoubtedly a precious piece of information for every robust control design that comes as a bonus when selecting this modelling approach. Therefore, this method seems to be a very promising approach for control design.

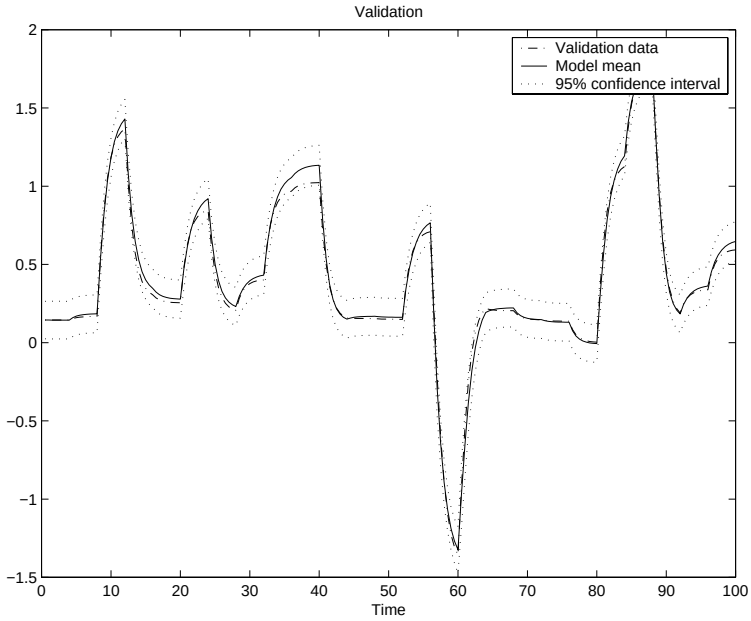

Figure 4: Responses of Gaussian process model (dashed line) and process model (full line) on validation input signal

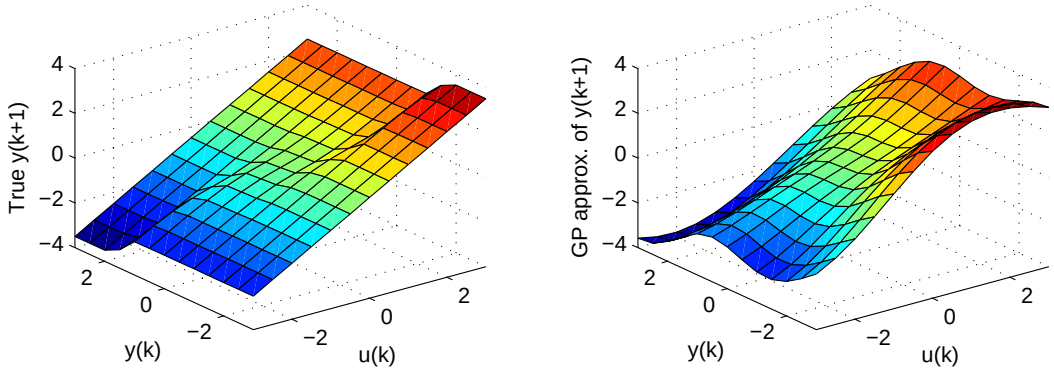

Figure 5: True process surface y(k+1)=f(u(k),y(k)) (left figure) and Gaussian process approximation of the process surface (right figure)

- Derivatives of mean and variance can be relatively easy extracted from Gaussian process models and used in, for example, optimisation.
- A noticable disadvantage is certainly a fact that the method is computationally relatively demanding, especially for more than a few hundred data, but not enough to prevent its use.

Acknowledgement
This work was made possible by EC funded Multi-Agent Control Research Training Network HPRN-CT-1999-00107.

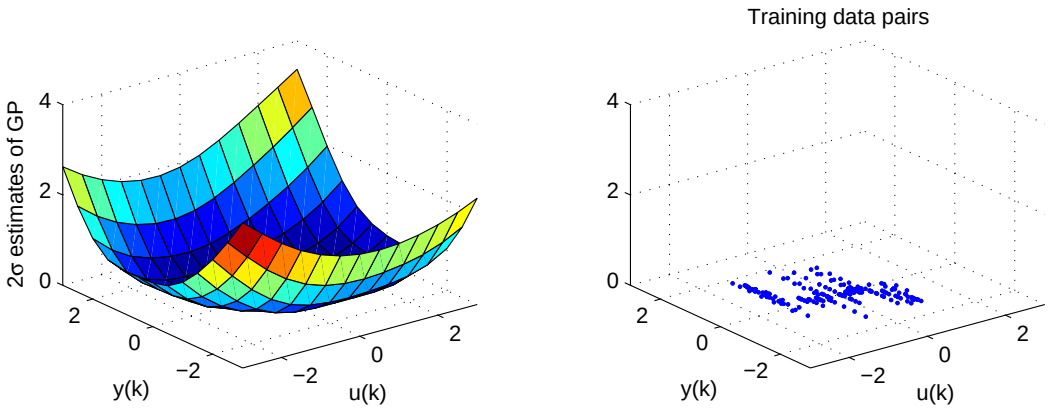

Figure 6: Uncertainty surface (left plot) $\sigma(k+1) = f(u(k), y(k))$ for the GP approximation shown in Figure 5 and location of training data (right plot)

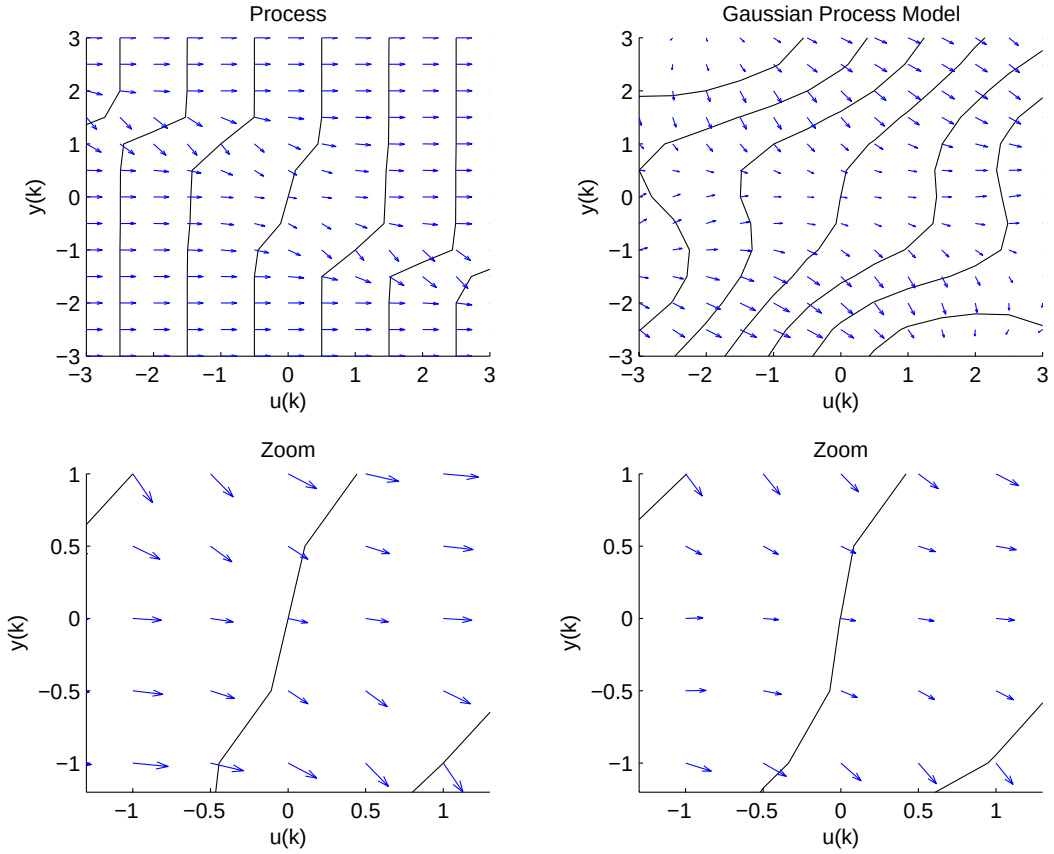

Figure 7: Contour and gradient plot of the true process function (upper left figure) and Gaussian process approximation of the process function (upper right figure) and corresponding magnified portions of the operating space where the model was trained (lower left and lower right figure)

6. References

1. Girard A., Rasmussen C.E., Murray-Smith R., Multi-step ahead prediction for non linear dynamic sytems - A Gaussian Process treatment with propagation of the uncertainty, Advances in Neural Information processing Systems 16, S. Becker and S. Thrun and K. Obermayer, *To appear*, 2002.

2. Gregorčič G. and Lightbody G., Gaussian Processes for Modelling of Dynamic Non–linear Systems, In: Proc. Irish Signals and Systems Conference, 2002, 141-147.

3. Leith D. J., Murray-Smith R.,Leithead W. E., Nonlinear Structure Identification: A Gaussian Process Prior/Velocity-based approach, Control 2000, Cambridge, 2000.

4. R. Murray-Smith, T. A. Johansen, R. Shorten, On transient dynamics, off-equilibrium behaviour and identification in blended multiple model structures, European Control Conference, Karlsruhe, BA-14, 1999.

5. Murray-Smith R. and Girard A., Gaussian Process priors with ARMA noise models, Irish Signals and Systems Conference, Maynooth, 2001, 147-152.

6. Murray-Smith R. and Sbarbaro D., Nonlinear adaptive control using nonparametric Gaussian process prior models, In: Proc. IFAC Congress, Barcelona, 2002.

7. Neal R.M., Bayesian learning for neural networks, Lecture notes in statistics, Springer Verlag, New York, 1996.

8. O'Hagan A., On curve fitting and optimal design for regression (with discussion), Journal of the Royal Statistical Society B, 40, 1978, 1-42.

9. Rasmussen C.E., Evaluation of Gaussian Processes and other Methods for Non-Linear Regression, Ph.D. Disertation, Graduate department of Computer Science, University of Toronto, Toronto, 1996.

10. E. Solak, R. Murray-Smith, W. E. Leithead, D. J. Leith, C. E. Rasmussen, Derivative observations in Gaussian Process models of dynamic systems, Advances in Neural Information processing Systems 16, S. Becker and S. Thrun and K. Obermayer, *To appear*, 2002.

11. Williams C.K.I., Prediction with Gaussian processes: From linear regression to linear prediction and beyond, In: Learning in Graphical Models (Edt.: Jordan, M.I.),Kluwer Academic, Dordrecht, 1998, 599-621.