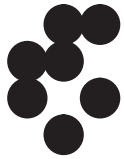

J. Stefan Institute, Ljubljana, Slovenia

Report **DP-8710**

Gaussian Process Model Based Predictive Control

Juš Kocijan

December 2002

Abstract

In this paper a principle of Gaussian process model based predictive control is described. Gaussian process models are recently utilised probabilistic non-parametric modelling approach for black-box identification of non-linear dynamic systems. 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. This property is used in predictive control, where optimisation of control signal takes the variance information into account. The predictive control principle is demonstrated on two simulated examples of nonlinear systems.

Keywords

Model based predictive control, Nonlinear control, Gaussian process models, Constraint optimisation, pH process.

1 Introduction

Model Predictive Control (MPC) is one of the most frequently met control algorithms in industrial practice. These are computer control algorithms that use an explicit process model to predict the future plant response. According to this prediction in the chosen period, also known as prediction horizon, the MPC algorithm optimises manipulated variable to obtain optimal future plant response. The input of chosen length, also known as control horizon, is sent into the plant and then the entire sequence is repeated again in the next time sample. The popularity of MPC is to a great extent owed to ability of MPC algorithms to deal with constraints that are frequently met in control practice and are often not well addressed with other approaches. MPC algorithms can handle hard and rate constraints on inputs and states that are incorporated in the algorithms usually, but not always, via optimisation method.

Linear model based predictive control approaches [12] have started appearing at the very beginning of eighties and are tightly established in control practice (e.g. overview in [15]). Nonlinear model based predictive control (NMPC) approaches [1] start to appear about ten years later and have also found their way into control practice (e.g. [16, 19]).

There were a number of contributions in the field of nonlinear model based predictive control dealing with issues like stability, efficient computation, optimisation, constraints and others. Some recent work in this field can be found in [11].

NMPC algorithms are based on various nonlinear models. Often are these models developed as first principles models, but other approaches, like black-box identification

approaches are also popular. Various predictive control algorithms are based on neural networks model e.g. [14], fuzzy models e.g. [2] or local model networks e.g. [6]. The quality of control depends on quality of model and it is in common to all approaches to try to find a good process model, first principle or black-box, that will successfully predict process behaviour. New developments in NMPC approaches are coming from resolving various issues: from faster optimisation methods to different process model.

The contribution of this paper is to describe a NMPC principle with a Gaussian process model. Gaussian process model is an example of a probabilistic non-parametric model that gives also the information about prediction uncertainties which are difficult to evaluate appropriately in nonlinear parametric models. Gaussian processes approaches which originated in statistics research are in many respects related to artificial neural networks, in terms of their application domain. Majority of work on Gaussian processes shown up to now considers modelling of static non-linearities. The use of Gaussian processes in modelling dynamic systems is a recent development e.g. [3, 4, 9, 10] and some control algorithms based on such are described in [13, 5].

The paper is organised as follows. Dynamic Gaussian process models are described in the next section. Control algorithm principle is described in Section 3 and illustrated with two examples in Section 4. Conclusions are stated at the end of paper.

2 Dynamic Gaussian process models

A Gaussian process 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 [18]. 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 Gaussian process, which means that it depends only on the distance between points in the input space[1]), 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

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

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 - the so called hyperparameters - of which 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 [17], 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

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

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 [17] 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 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. 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 further 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 $\sigma^2(\mathbf{x}^*)$.

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

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

For more detailed derivation see [3].

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, so called *direct method*, or by simulating the system which means repeated one-step ahead predictions up to k - so called *iterative method*. That is, at each time step, by feeding back the mean prediction also called estimate of the output and its variance. 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.

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

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

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 [3].

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.

3 Controller synthesis

Predictive control principle (Figure 1) can be sumarised as follows:

- Prediction of system output signal $y(k+j)$ is calculated for each discrete sample k for a large horizon in future ($j = N_1, \ldots, N_2$). Predictions are denoted as $\hat{y}(k+j|k)$ and represent j-step ahead prediction, while N_1 and N_2 determine lower and upper bound of prediction horizon. Lower and upper bound of output signal prediction horizon determine coincidence horizon, within which a match between output and reference signal is expected. Output signal prediction is calculated from process model. Predictions are dependent also on control scenario in the future $u(k+j|k), j = 0, \ldots, N_u - 1$, which is intended to be applied from a moment k onwards.
- Reference trajectory is determined $r(k+j|k), j = N_1, \ldots, N_2$, which determines reference process response from present value $y(k)$ to the setpoint trajectory $w(k)$.
- Vector of future control signal ($u(k+j|k), j = 0, \ldots, N_u - 1$) is calculated by minimisation of objective function in a way that predicted error between $r(k+j|k)$ and $\hat{y}(k+j|k, j = N_1, \ldots, N_2$. Structuring of future control samples can be used in some approaches.
- Only first element $u(k|k)$ of the optimal control signal vector $u(k+j|k), j = 0, \ldots, N_u - 1$ is applied.

In the next sample a new measured output sample is available and the entire described procedure is repeated. This principle is called receding horizon strategy.

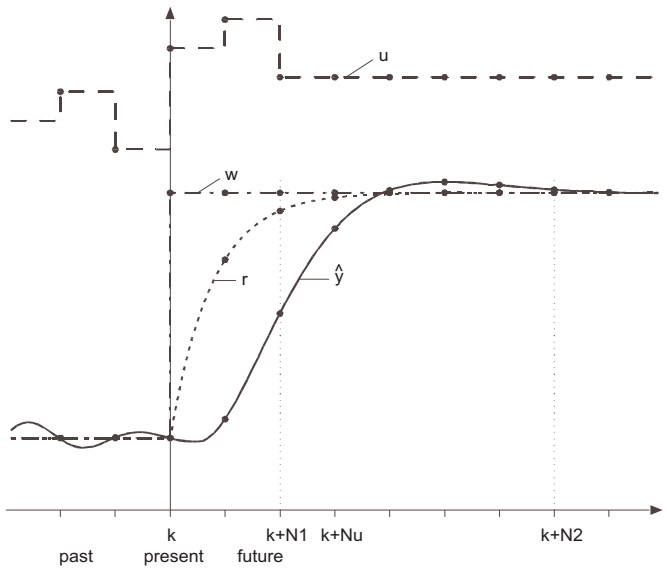

Figure 1: Predictive control principle

Nonlinear model predictive control as it was applied with Gaussian process model can be in general described with block scheme depicted in Figure 2.

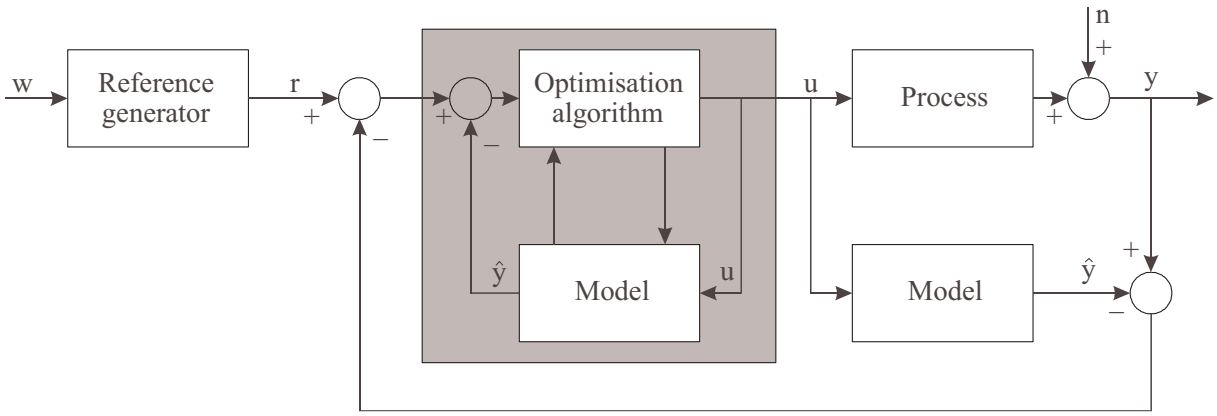

Figure 2: Block diagram of model predictive control system

A moving-horizon minimisation problem of the special form [12]

$$\min_{\mathbf{U}(k)} \; [r(k+P) - \hat{y}(k+P)]^2 \tag{13}$$

subject to:

$$\begin{aligned} \text{var}\ \hat{y}(k+P) &\leq k_v && (14)\\ |\ \mathbf{U}(k)\ | &\leq k_{ih} && (15)\\ |\ \dot{\mathbf{U}}(k)\ | &\leq k_{ir} && (16)\\ |\ \mathbf{x}(k)\ | &\leq k_{sh} && (17)\\ |\ \dot{\mathbf{x}}(k)\ | &\leq k_{sr} && (18) \end{aligned}$$

where $\mathbf{U}(k) = [u(k) \dots u(k+P)]$ is input signal, P is the coincidence point (the point where a match between output and reference value it is expected) and inequalities from (14) to (18) represent constraint on output variance k_v, input hard constraint k_{ih}, input rate constraint k_{ir}, state hard constraint k_{sh} and state rate constraint k_{sr} respectively.

The optimisation algorithm, which is constraint nonlinear programming, is solved at each sample time over a prediction horizon of length P, for a series of moves which equals to control horizon. In our case control horizon was chosen to be one and to demonstrate constraint on variance the rest of constraints was not taken into the account. Nevertheless, all this modifications does not change the generality of solution, but they do affect the numerical solution itself.

The process model is a Gaussian process. Some issues of interest with the applied NMPC are:

Efficient numerical solution Nonlinear programming optimisation algorithm is very demanding for computation. Various approximations and other approaches exist to

decrease computational load, mainly for special cases, like linear process models or special objective functions.

The computational load necessary for optimisation can be decreased easily with the incorporation of prediction derivation (and variance) into optimisation algorithm. When using Gaussian process models the prediction and variance derivation can be calculated easily.

Stability At present no stability conditions have been derived for Gaussian processes as a representative of probabilistic non-parametric models.

Robustness This issue has a major impact on applicability of algorithm in practice. The fact that the process model contains the information about the model confidence enables controller to optimise the manipulative variable to "avoid" regions where the confidence in model is not high enough. This possibility itself makes the controller robust if applied properly.

Alternative ways of how NMPC with Gaussian process models can be realised are as follows.

Different objective function Used objective function (13) is just one of possible ones. It is well know that selection of objective function has a major impact on the amount of computation.

Optimisation problem for $\Delta\mathbf{U}(k)$ instead of $\mathbf{U}(k)$ This is not just change of formalism, but also enables other forms of NPC. One possibility is a DMC controller with nonlinear model, e.g. [2] - a frequently used principle, that together with appropriate objective function enables problem representation as a least squares problem that can be solved in one iteration. This is, as in the case with other special case simplifications, not a general case solution.

Soft constraints Using constraint optimisation algorithms is very demanding for computation and soft constrains, namely weights on constrained variables in objective function, can be used to decrease the amount of computation. More on this topic can be found in [8, 20].

Linear MPC It is worth to remark that even though this is a constrained nonlinear MPC problem it can be used in its specialised form as a robust linear MPC.

4 Illustrative examples

4.1 The first order non-linear process

4.1.1 Process description

The described approach is illustrated with control of system that is described with equation

$$\dot{y} = -\tanh(y + u^3) \tag{19}$$

with output signal y and input signal u. The output signal was disturbed with the white noise of variance 0.0025 and zero mean. Data sampling time, determined according to system dynamics, was selected to be 0.5 units.

4.1.2 Model identification

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, output signal and these two signals delayed for one sample were chosen as regressors. The selected model can therefore be written in the form

$$y(k+1) = f(y(k), u(k)) \tag{20}$$

where function $f(\cdot)$ represents Gaussian process model as a two dimensional regression model. Since the system in equation (19), 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 [17] due to its good convergence properties. The following set of hyperparameters was found:

$$\boldsymbol{\Theta} = [w_1, w_2, v_0, v_1] = [0.1312, 0.2948, 6.2618, 0.0045] \tag{21}$$

where hyperparameters w_1 and w_2 allow a weight for each input dimension.

The process (19) was identified in the region that can be figured out from response on Figure 3. Validation signal was also generated by random number generator with

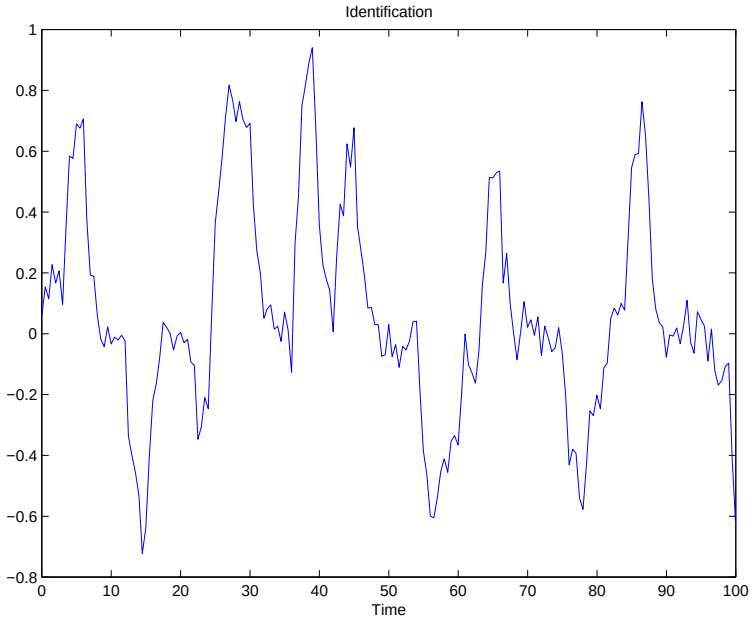

Figure 3: Response of GP model on excitation signal used for identification

normal distribution and at different rate (4) than for the identification signal. Results on validation signal, different from the identification one, show that 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. For more details see [9].

4.1.3 Control

The reference trajectory r is defined so that it approaches the set-point exponentially from the current output value. This means that closed-loop system should behave as the first order system. The closed-loop response of unconstrained control is given in Figure 4 and corresponding standard deviation in Figure 5. Set-point was chosen in a way that it goes from region where model is more trusted towards the region where the model is less trusted. Less reliable is the model, worse is closed-loop response and consequently higher (much higher) is its variance.

To avoid unpleasant responses that are the consequence of difference between model and real process a hard constraint on variance value has been set and constraint predictive control obtained. Closed-loop response can be seen in Figure 6 and corresponding standard deviation in Figure 7.

It can be seen that in the second case controller "avoids" control inputs that lead the process in the regions where variances are big. This means that the obtained nonlinear control algorithm is robust enough to ensure specified performance and most likely stability. It can be said that it is intelligent enough to avoid regions where it can not

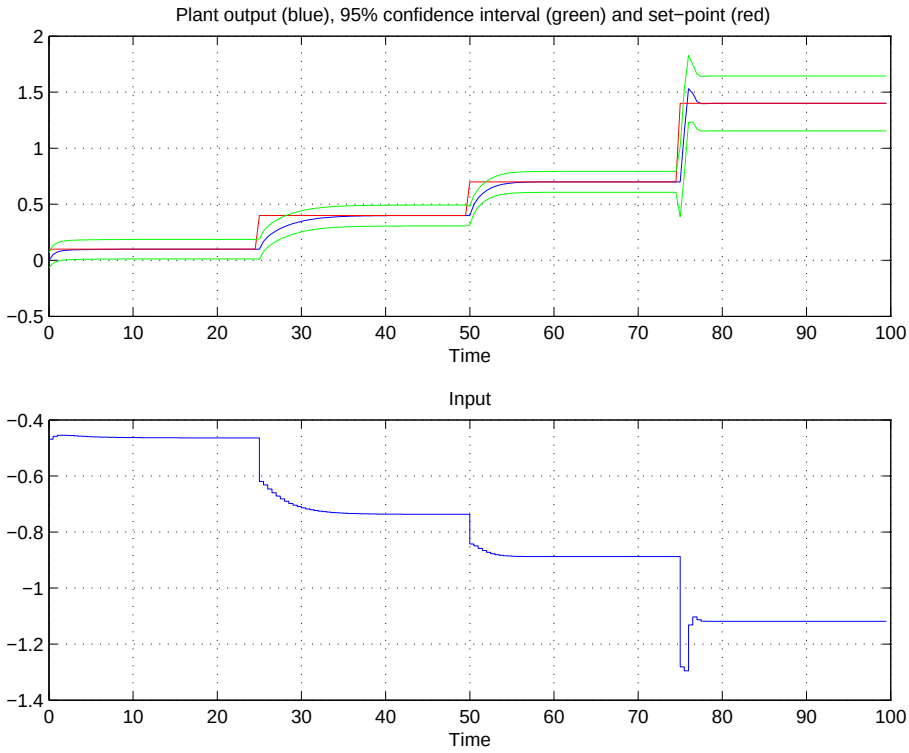

Figure 4: Non-constrained case: response of GP model based control (upper figure) and control signal (bottom figure)

operate within specified parameters. This is a consequence of the information contained in the Gaussian process model.

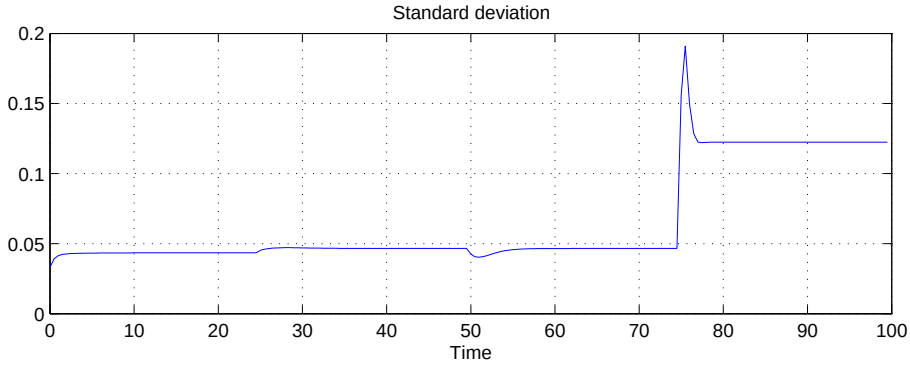

Figure 5: Non-constrained case: standard deviation corresponding to the previous figure

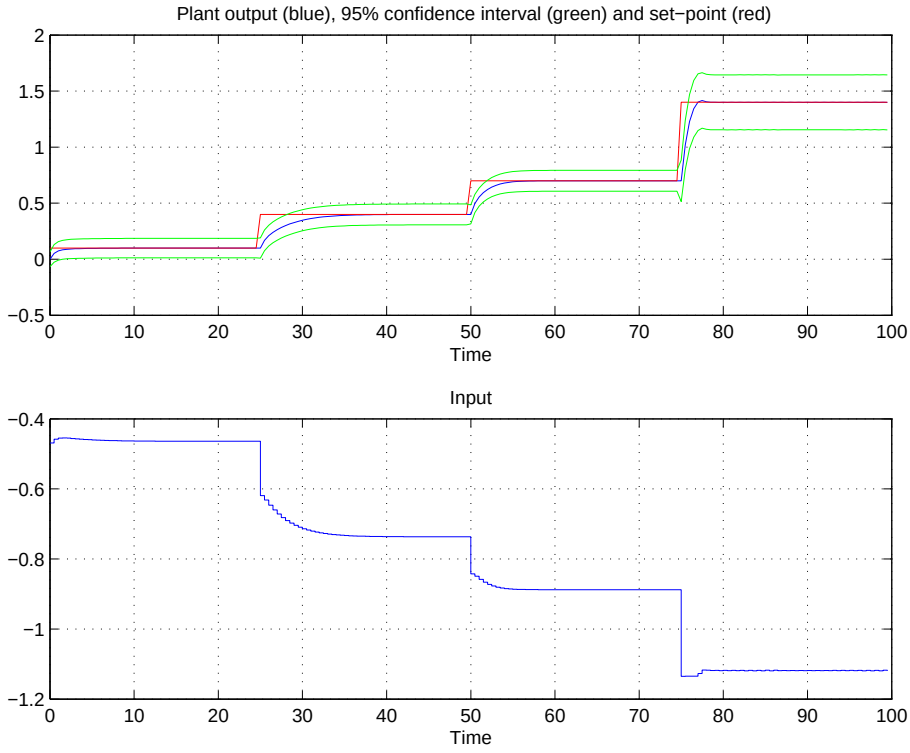

Figure 6: Constrained case ($\sigma_{max} = 0.13$): response of GP model based control (upper figure) and control signal (bottom figure)

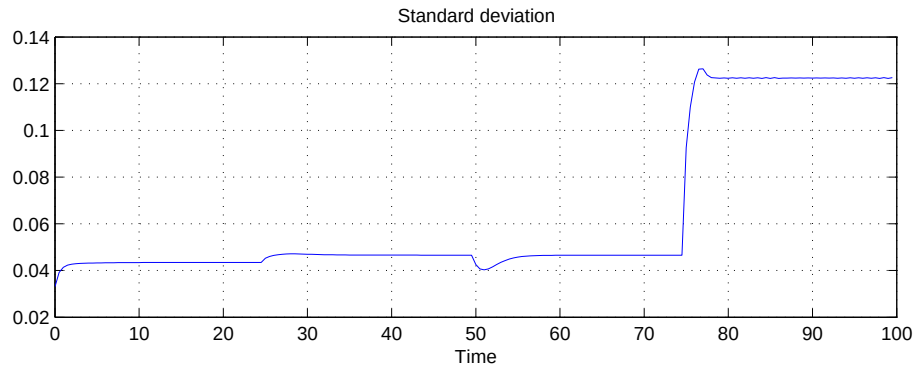

Figure 7: Constrained case ($\sigma_{max} = 0.13$): standard deviation corresponding to the previous figure

4.2 pH process

4.2.1 Process description

A simplified schematic diagram of the pH neutralization process taken from [7] is given in Figure 8. The process consists of an acid stream (Q_1), buffer stream (Q_2) and base stream

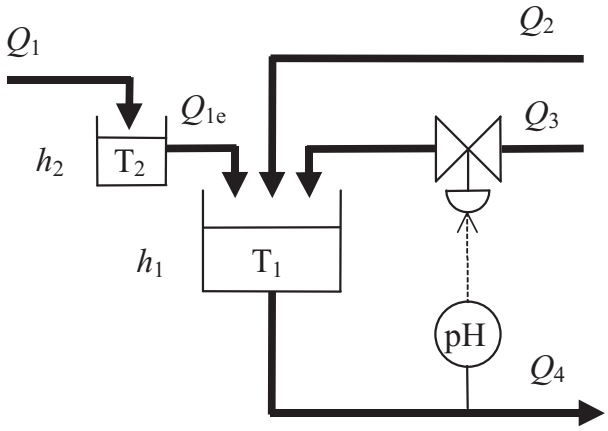

Figure 8: The pH neutralization system scheme

(Q_3) that are mixed in a tank T_1. Prior to mixing, the acid stream enters the tank T_2 which introduces additional flow dynamics. The acid and base flow rates are controlled with flow control valves, while the buffer flow rate is controlled manually with a rotameter. The effluent pH (pH) is the measured variable. Since the pH probe is located downstream from the tank T_1, a time delay (T_d) is introduced in the pH measurement. In this study, the pH is controlled by manipulating the base flow rate. A more detailed description of the process with mathematical model and necessary parameters is presented in [7].

The dynamic model of the pH neutralization system shown in Fig. 8 is derived using the conservation equations and equilibrium relations. The model also includes valve and transmitter dynamics as well as hydraulic relationships for the tank outlet flows. Modelling assumptions include perfect mixing, constant density, and complete solubility of the ions involved. The simulation model of pH process, which was used for necessary data generation contains therefore various non-linear elements as well as implicitly calculated function which is value of highly non-linear titration curve.

4.2.2 Model identification

Based on responses and iterative cut-and-try procedure a sampling time of 25 seconds was selected. The sampling time was so large that the dead-time mentioned in the previous section disappeared.

The chosen identification signal of 400 samples was generated from a uniform random distribution and rate of 50 seconds.

Obtained hyperparameters of the fourth order Gaussian process model were:

$$\begin{aligned} \boldsymbol{\Theta} &= [w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8, v_0, v_1] \\ &= [0.6648, 0.1063, 0.0024, 0.0007, 0.0002, 0.0275, 0.0046, 0, 0.0045, 2.339] \end{aligned} \quad (22)$$

where hyperparameters from w_1 to w_4 denote a weight for each output regressor, from w_5 to w_8 denote a weight for each input regressor, v_0 is estimated noise variance and v_1 is the estimate of the vertical variance.

The region in which the model was obtained can be seen from Figure 9. A very good

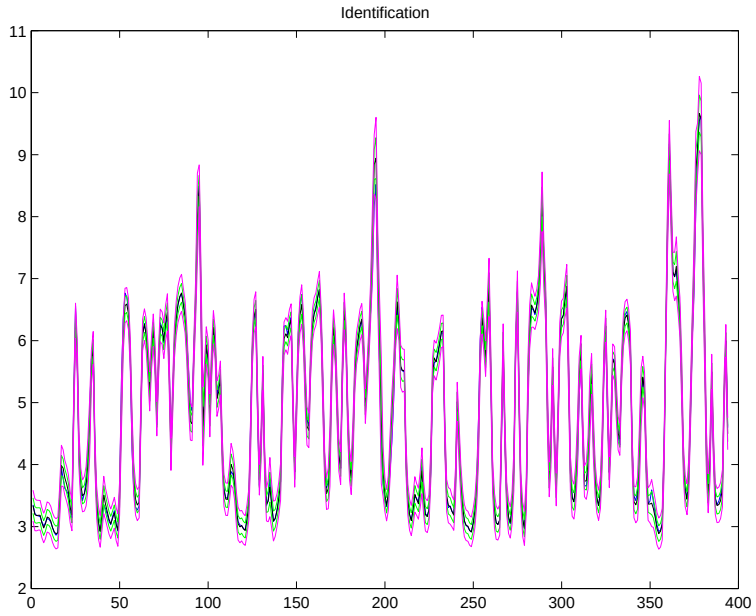

Figure 9: Response of GP model on excitation signal used for identification

fit can be observed for identification input signal which was used for optimization. The validation signal had lower magnitude and frequency components than the identification signal. The rationale behind this is that if identified model was excited with a richer signal, than it has to respond well to the signal with less components. The validation signal was obtained with generator of random noise with uniform distribution and rate of 500 seconds. The response on validation signal is not so good. Despite a poorer fit to the validation data, the model could still be useful for control design. See [10] for more details.

4.2.3 Control

The same control algorithm as in the previous example has been tested for the pH process. Again the reference trajectory r is defined so that it approaches the set-point exponentially

from the current output value. Results of unconstrained control are given in Figures 10 and 11.

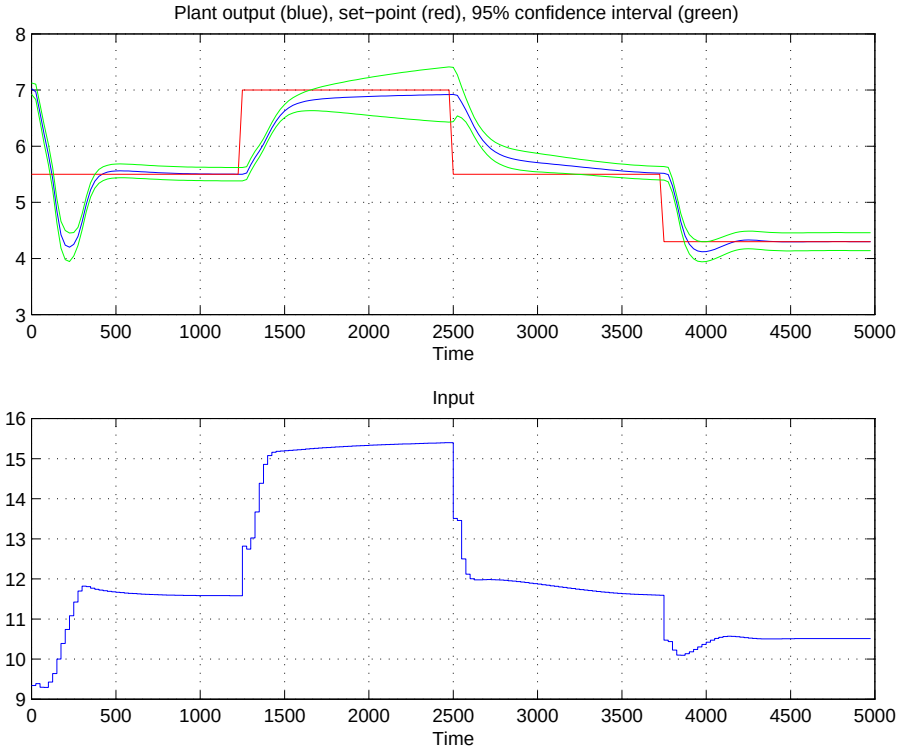

Figure 10: Non-constrained case: response of GP model based control (upper figure) and control signal (bottom figure)

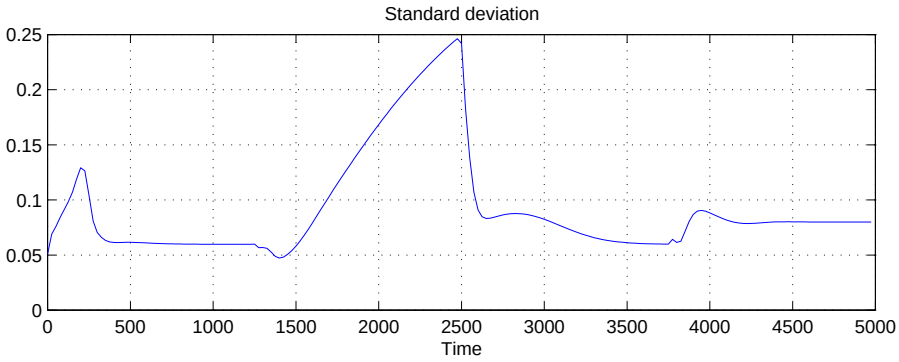

Figure 11: Non-constrained case: standard deviation corresponding to the previous figure

It can be seen from different set-point responses that model differs from the process in different regions. It can also be observed that variances do not contain information about model quality. They provide only information about regions where model is more or less to be trusted depending upon available identification data. Even though variances are relatively small in some regions, performance varies. When variances increase too much,

the response can be optimised with constrained control. Results can be seen in Figures 12 and 13.

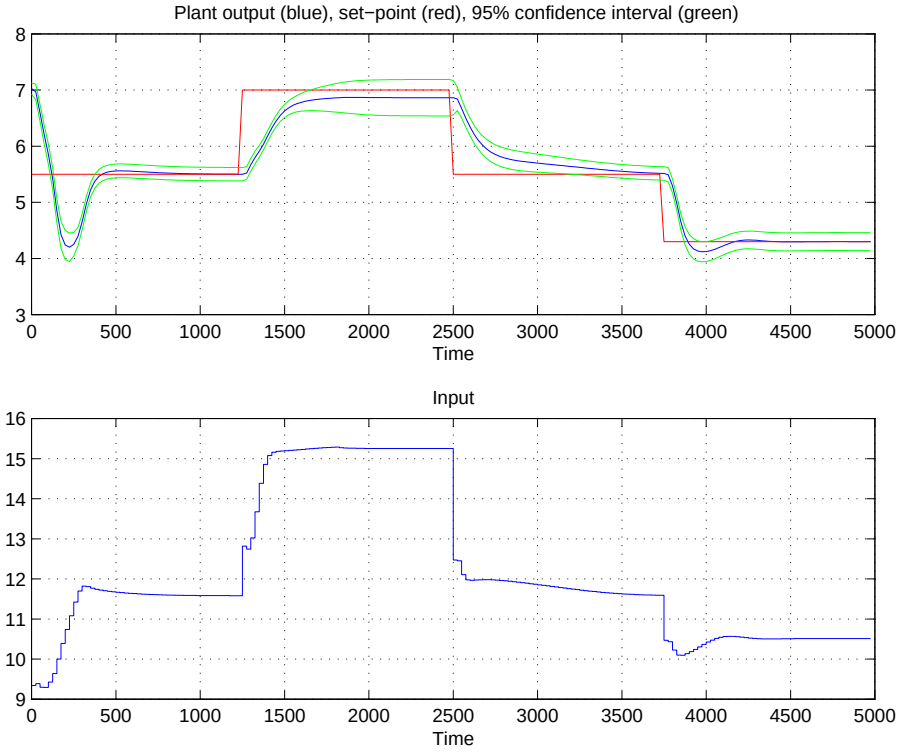

Figure 12: Constrained case ($\sigma_{max} = 0.15$): response of GP model based control (upper figure) and control signal (bottom figure)

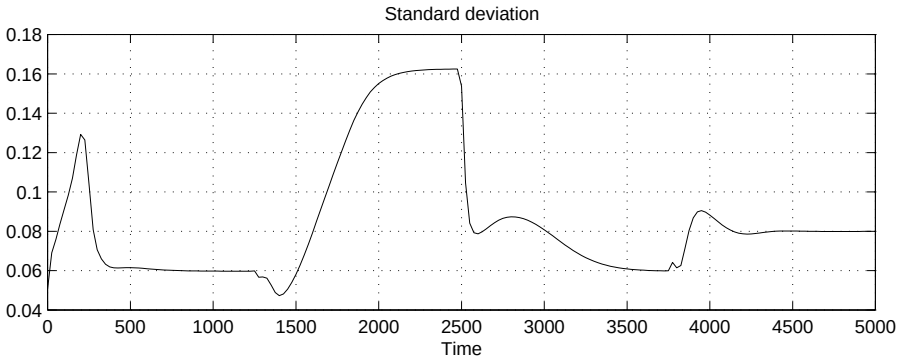

Figure 13: Constrained case ($\sigma_{max} = 0.15$): standard deviation corresponding to the previous figure

5 Conclusions

Model Predictive Control is industrially attractive and frequently applied because it handles hard constraints, usually on input and output signals. The principle of Model Predictive Control based on Gaussian process model was presented in the paper and illustrated with two examples. In the presented examples constraint on model variances was included. This can be complimented also with other constraints when necessary.

Use of Gaussian process models makes possible to include information about the trust into model depending on the region. Incorporating this information enables a design of robust controller that will optimise action according to the validity of model.

However, a distinction has to be made between information contained in Gaussian process about trust into model and model quality that depends on data used for identification.

Nevertheless, it was shown that using Gaussian process models offers an attractive possibility for control design that results in a controller with a higher level of robustness due to information contained in the model.

The principle shown in the paper is quite general and several modifications that accelerate computation can be used and are planned to be derived in the future.

Acknowledgement

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

References

[1] Allgöwer F., Badgwell T.A., Qin S.J., Rawlings J.B., Wright S.J., Nonlinear predictive control and moving horizon estimation - an introductory overview, In: Frank, P.M (Edt.), Advances in control: highlights of ECC'99, Springer, 1999, 391-449.

[2] Fisher M., Schmidt M., Kavšek-Biasizzo K., Nonlinear predictive control based on the extraction of step response models from Takagi-Sugeno fuzzy systems, Proceedings of ACC, Albuquerque, 1997.

[3] 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, NIPS 2002 conference, 2002.

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

[5] Gregorčič G. and Lightbody G., Internal model control based on a Gaussian process prior model, Accepted for ACC'2003, Denver, CO, 2003.

[6] Johansen T.A., Foss B.A., Sorensen A.V., Non-linear predictive control using local models - applied to a batch fermentation process, Control Eng. Practice, 1995, 3(3), 389-396.

[7] Henson M.A. ,Seborg D.E., Adaptive Nonlinear Control of a pH Neutralization Process, IEEE Trans. Control System Technology, Vol. 2, No. 3, 169-183, 1994.

[8] Kerrigan E.C., Maciejowski J.M., Soft constraints and exact penalty functions in model predictive control, Control 2000 Conference, Cambridge, 2000.

[9] Kocijan J., Girard A., Banko B., Murray-Smith R., Dynamic Systems Identification with Gaussian Processes, Accepted for presentation at 4th Mathmod, Vienna, 2003.

[10] Kocijan J., Likar B., Banko B., Girard A., Murray-Smith R., Rasmussen C.E., A case based comparison of identification with neural network and Gaussian process models, Accepted for presentation at IFAC International Conference on Intelligent Control Systems and Signal Processing, Faro, 2003.

[11] Kouvaritakis B., Cannon M. (Eds.), Nonlinear predictive control, Theory and practice, IEE Control Engineering Series 61, IEE, 2001.

[12] Maciejowski J.M., Predictive control with constraints, Pearson Education Limited, Harlow, 2002.

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

[14] Nørgaard M., Ravn O., Poulsen N.K., Hansen L.K., Neural networks for modelling and control of dynamic systems, Springer, London, 2000.

[15] Qin S.J., Badgwell T.A., An overview of industrial model predictive control technology, In: Kantor J.C., Garcia C.E., Carnahan B. (Eds.) Fifthe international conference on Chemical process control, AChE and CACHE, 1997, 232-56.

[16] Qin S.J., Badgwell T.A., An overview of nonlinear model predictive control applications, In: Allgöwer F., Zheng A. (Eds.), Nonlinear model predictive control, Birkhauser Verlag, 2000, 369-392.

[17] 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.

[18] 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.

[19] Young R.E.,Bartusiak R.D., Fontaine, R.W., Evolution of an industrial nonlinear model predictive controller, Preprints on Chemical Process Control - CPC VI, CACHE, Tucson, AZ, 2001, 399-401.

[20] Zheng A., Morari M., Stability of model predictive control with mixed constraints, IEEE Trans. Autom. Control, 1995, 40(19), 1818-1823.