Predictive control with Gaussian process models

Juš Kocijan[1,2], Roderick Murray-Smith[3,4], Carl Edward Rasmussen[5], Bojan Likar[1]
[1]Jozef Stefan Institute, Jamova 39, SI-1000 Ljubljana, Slovenia
E-mail: jus.kocijan@ijs.si
[2]Nova Gorica Polytechnic, Nova Gorica
[3] Dept. Computing Science, University of Glasgow, Glasgow
[4] Hamilton Institute, National University of Ireland, Maynooth
[5] Max Planck Institute for Biological Cybernetics, Tübingen

Abstract—**This paper describes model-based predictive control based on Gaussian processes. Gaussian process models provide a 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 around 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 a simulated example of nonlinear system.**

Keywords—**Model based predictive control, Nonlinear control, Gaussian process models, Constraint optimisation.**

I. 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 the prediction horizon, the MPC algorithm optimises the manipulated variable to obtain an 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 the 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 state and rate constraints on inputs and states that are usually, but not always incorporated in the algorithms via an optimisation method. Linear model based predictive control approaches [11] started appearing in the early eighties and are well-established in control practice (e.g. overview in [17]). 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. [18], [23]). 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 [10]. NMPC algorithms are based on various nonlinear models. Often these models are 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. [16], fuzzy models e.g. [6], [21] or local model networks e.g. [4]. The quality of control depends on quality of model. 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. The Gaussian process model is an example of a probabilistic non-parametric model that also provides information about prediction uncertainties which are difficult to evaluate appropriately in nonlinear parametric models. The 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. [13], [12], [2], [20], [8], [9] and some control algorithms based on such are described in [14], [3].

The paper is organised as follows. Dynamic Gaussian process models are described in the next section. The control algorithm principle is described in Section III and illustrated with an example in Section IV. Conclusions are stated at the end of the paper.

II. 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 reviewed in [22]. A Gaussian process is a collection of random variables which have a joint multivariate Gaussian distribution: $y^1, \ldots, y^n \sim \mathcal{N}(0, \Sigma)$, where Σ_{pq} gives the covariance between output points corresponding to input points x^p and x^q. Mean $\mu(y^p)$, which is usually assumed to be zero ($\mu(y^p) = 0$), 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.

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 $f(\cdot)$, $\mathbf{X}$ and $\mathbf{y}$ and not on parameter determination within a fixed model structure. That is why this is a probabilistic non-parametric approach. Based on the covariance function, the parameters – the so called hyperparameters – of which can be determined from a training set $\mathbf{X}, \mathbf{y}$, the *a posteriori* value

0-7803-7763-X/03/$17.00 ©2003 IEEE

$y(N_+ + 1)$ 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

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

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 [19]. 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 corresponding 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 $p(f(\cdot) \mid \mathbf{X}, \mathbf{y})$. Since the analytic solution is very difficult to obtain other approaches are in place. The description of one possible approach follows.

Calculation of the model output is straightforward for a given covariance function. The posteriori probability depends on hyperparameters through the likelihood $p(\mathbf{y} \mid \mathbf{f}(\cdot), \mathbf{X})$.

The maximimum likelihood approach obtains the hyperparameters by minimising negative $\mathcal{L}$. Any appropriate optimisation method can be used for this minimisation. Nevertheless, it has to be kept in mind that the approach is computationally relatively demanding since the inverse covariance matrix has to be calculated at every iteration.

The described approach can be easily utilised for regression calculation. Based on training set $\mathbf{X}$ a covariance matrix $\mathbf{K_N}$ of size $N \times N$ 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 $\mathbf{x}^* = [x_1(N+1), x_2(N+1), \ldots, x_D(N+1)]^T$.

A prediction at point $y(N+1)$ is also a Gaussian distribution. 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

$$\mu(\mathbf{x}^*) = \mathbf{k}(\mathbf{x}^*)^{\mathrm{T}} \mathbf{K}^{-1} \mathbf{y} \quad (2)$$

$$\sigma^2(\mathbf{x}^*) = k(\mathbf{x}^*) - \mathbf{k}(\mathbf{x}^*)^{\mathbf{T}} \mathbf{K}^{-1} \mathbf{k}(\mathbf{x}^*) + \mathbf{v_0} \quad (3)$$

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. Based on recent work by Girard *et. al.* [2], we can use a Gaussian approximation to the uncertainty of inputs. The predictive distribution of the corresponding output at the random input $\mathbf{x}^*$ is $\mathcal{N}(m(\mathbf{x}^*), v(\mathbf{x}^*))$ where $m(\mathbf{x}^*)$ and $v(\mathbf{x}^*)$ are approximations of $\mu(\mathbf{x}^*)$ and $\sigma^2(\mathbf{x}^*)$.

$$\begin{aligned} m(\mathbf{x}^*) &= E_{\mathbf{x}^*}[\mu(\mathbf{x}^*)] \\ &\approx \mathbf{k}(\mu(\mathbf{x}^*)^{\mathbf{T}} \mathbf{K}^{-1} \mathbf{y} \end{aligned} \quad (4)$$

$$\begin{aligned} v(\mathbf{x}^*) &= E_{\mathbf{x}^*}[\sigma^2(\mathbf{x}^*)] + \mathrm{var}_{\mathbf{x}^*}(\mu(\mathbf{x}^*)) \\ &\approx \sigma^2(\mu(\mathbf{x}^*)) + \mathrm{trace}\left\{\Sigma_{\mathbf{x}^*}\left(\frac{1}{2}\frac{\partial^2 \sigma^2(\mathbf{x}^*)}{\partial \mathbf{x}^* \partial \mathbf{x}^{*\mathbf{T}}}\Big|_{\mathbf{x}^*=\mu(\mathbf{x}^*)}\right.\right. \\ &+ \left.\left.\frac{\partial \mu(\mathbf{x}^*)}{\partial \mathbf{x}^*}\Big|_{\mathbf{x}^*=\mu(\mathbf{x}^*)} \frac{\partial \mu(\mathbf{x}^*)}{\partial \mathbf{x}^*}\Big|^T_{\mathbf{x}^*=\mu(\mathbf{x}^*)}\right)\right\} \end{aligned} \quad (5)$$

For a more detailed derivation see [2].

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.

$$\begin{aligned} \mathbf{x}(k) &= [\hat{y}(k-1), \hat{y}(k-2), \ldots, \hat{y}(k-L), u(k-1), \\ &\quad u(k-2), \ldots, u(k-L)]^T \\ \hat{y}(k) &= f(\mathbf{x}(k)) + \epsilon \end{aligned} \quad (6)$$

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.

It is worthwhile noting that the derivatives of means and variances can be calculated in straightforward manner. For more details see [20] or [2].

As can be seen from the presented relations the obtained model describes both the dynamic characteristics of non-linear system, and at the same time provides information about the confidence in these predictions. The Gaussian process 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 around the predicted mean.

III. Controller synthesis

The predictive control principle 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 the 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.
- The 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 $w(k)$.
- The vector of future control signals ($u(k+j|k), j = 0, \ldots, N_u - 1$) is calculated by minimisation of objective function such 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 the 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.

Nonlinear model predictive control as it was applied with the Gaussian process model can be in general described with a block diagram, as depicted in Figure 1. The model used is fixed, identified off-line, which means that used control algorithm is not an adaptive one.

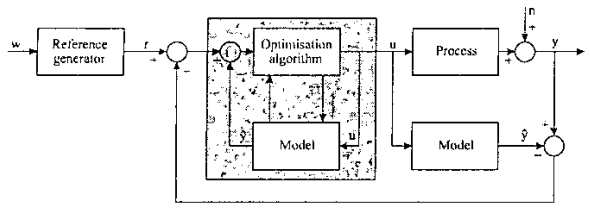

Fig. 1. Block diagram of model predictive control system

A moving-horizon minimisation problem of the special form [11]

$$\min_{\mathbf{U}(k)} [r(k+P) - \hat{y}(k+P)]^2 \tag{7}$$

subject to:

$$\begin{aligned} \text{var}\, \hat{y}(k+P) &\leq k_v && (8)\\ |\, \mathbf{U}(k)\,| &\leq k_{ih} && (9)\\ |\, \dot{\mathbf{U}}(k)\,| &\leq k_{ir} && (10)\\ |\, \mathbf{x}(k)\,| &\leq k_{sh} && (11)\\ |\, \dot{\mathbf{x}}(k)\,| &\leq k_{sr} && (12) \end{aligned}$$

where $\mathbf{U}(k) = [u(k) \ldots u(k+P)]$ is input signal, P is the coincidence point (the point where a match between output and reference value is expected) and inequalities from (8) to (12) 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 constrained 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 do 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 for applied NMPC are:

Efficient numerical solution Nonlinear programming optimisation algorithm is very demanding for computation. Various approximations and other approaches (e.g. approximation of explicit solution) exist to decrease computational load, mainly for special cases, like linear process models or special objective functions.

One possibility to decrease the computational load necessary for optimisation is with the incorporation of prediction derivation (and variance) into optimisation algorithm. When using Gaussian process models the prediction and variance derivation can be calculated in a straightforward manner.

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 the applicability of the 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. MPC robustness in the case of other algorithms is usually not some specially built feature of the MPC algorithms, but was more an issue of assessment for particular MPC algorithms.

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

Different objective function The objective function used (7) is just one of many possible ones. It is well known that selection of the objective function has a major impact on the amount of computation.

Optimisation problem for $\Delta\mathbf{U}(\mathbf{k})$ *instead of* $\mathbf{U}(\mathbf{k})$ This is not just a change of formalism, but also enables other forms of NPC. One possibility is a DMC controller with nonlinear model, e.g. [6] - 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 in which an explicit solution is found. 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 [7], [24].

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.

IV. Example

A. The first order non-linear process

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

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

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.

B. Model identification

The input signal was generated by a random number generator with normal distribution and rate of 3 units in the magnitude range between -1.3 and 1.3 The number of input signal samples determines the dimensions of the covariance matrix. To avoid excessive computation time it is sensible to choose number of samples to be no more than a few 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

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

where function $f(\cdot)$ represents 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 appropriate. The covariance function (1) 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 [19] due to its good convergence properties. The following set of hyperparameters was found:

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

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

The validation signal was also generated by random number generator with 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 the results are to be trusted. For more details see [8].

C. Control

The reference trajectory r is defined so that it approaches the set-point exponentially from the current output value. This means that the closed-loop system should behave close to the first order system when the process model is a good description of the process itself. The coincidence point for the chosen MPC was selected as $P = 8$ and the control horizon $N_u = 1$ as it has already been mentioned in the previous section. The set-point for the closed-loop system was chosen in a way that it goes from region where model is more trusted towards the region where the model is less trusted.

The closed-loop response of unconstrained control is given in Figure 2. It can be seen from Figure 2 that the model is less reliable in certain region of response, and its closed-loop response deteriorates similarly, and its variance is also higher.

To avoid unpleasant responses that are the consequence of difference between the model and the real process a hard constraint on variance value ($\sigma_{max} = 0.13$) has been set and constrained predictive control obtained. Closed-loop response can be seen in Figure 3.

It can be seen that in the second case the controller "avoids" control inputs that lead the process in the regions where variances are big. This means that the obtained non-linear 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 operate within specified parameters[1]. This is a consequence of the information contained in the Gaussian process model.

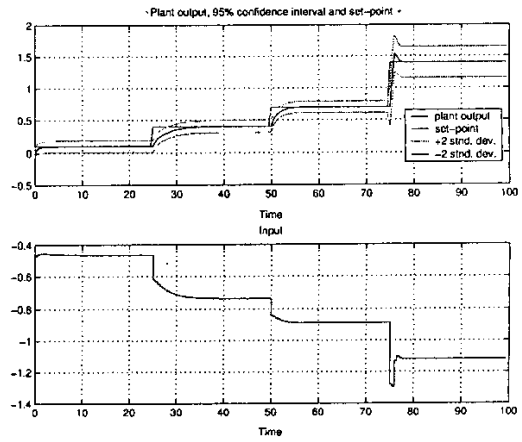

Fig. 2. Non-constrained case: response of GP model based control (upper figure) and control signal (bottom figure)

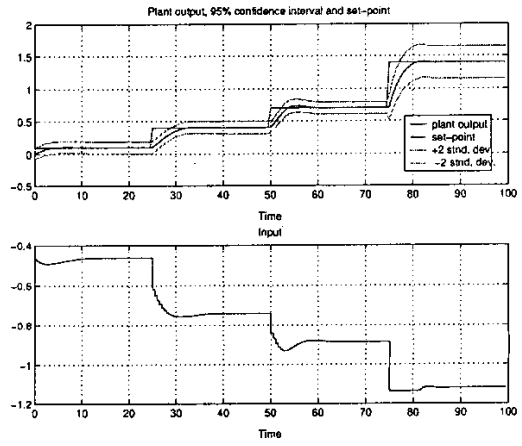

Fig. 3. Constrained case ($\sigma_{max} = 0.13$): response of GP model based control (upper figure) and control signal (bottom figure)

V. 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 a Gaussian process model was presented in the paper and illustrated with an example. In the presented example constraint on model variance was included. This can be complimented also with other constraints when necessary. The use of Gaussian process models makes it possible to include information about the trust in the model depending on the re-

[1] A 'soft' version of this, which leads to different control properties, is described in [15]

gion. 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.

ACKNOWLEDGMENTS

This work was made possible by EC funded Multi-Agent Control Research Training Network HPRN-CT-1999-00107, and RM-S by EPSRC project GR/M76379/01 , and Science Foundation Ireland grant 00/PI.1/C067. C. E. Rassmussen was supported by the German Research Council (DFG) through grant RA 1030/1. J. Kocijan acknowledges the support of Slovene Ministry of education, science and sport. The authors are grateful for comments and useful suggestions from Daniel Sbarbaro and Gregor Gregorčič.

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, 391-449, 1999.

[2] Girard A., Rasmussen C.E., Quinonero Candela, J. and Murray-Smith R., Gaussian Process Priors With Uncertain Inputs & Application to Multiple-Step Ahead Time Series Forecasting, NIPS 15, Vancouver, Canada, MIT Press, 2003.

[3] Gregorčič G., Lightbody G., Internal model control based on a Gaussian process prior model, Proceedings of ACC'2003, Denver, CO, 4981-4986, 2003.

[4] 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*, 3(3), 389-396, 1995.

[5] 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.

[6] Kavšek-Biasizzo K., Škrjanc I., Matko D., Fuzzy predictive control of highly nonlinear pH process, *Computers & chemical engineering*, Vol. 21, Supp. 1997, S613-S618, 1997.

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

[8] Kocijan J., Girard A., Banko B., Murray-Smith R., Dynamic Systems Identification with Gaussian Processes, Proceedings of 4th Mathmod, Vienna, 776-784, 2003.

[9] 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, Preprints of IFAC ICONS Conference, Faro, 137-142, 2003.

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

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

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

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

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

[15] Murray-Smith, R, Sbarbaro, D., Rasmussen, C.E., Girard, A., Adaptive, Cautious, Predictive control with Gaussian Process priors, 13th IFAC Symposium on System Identification, Rotterdam, 2003

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

[17] 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, 232-56, 1997.

[18] 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, 369-392, 2000.

[19] 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.

[20] Solak, E., Murray-Smith R., Leithead, W.E., Leith, D.J., and Rasmussen, C.E., Derivative observations in Gaussian Process models of dynamic systems, NIPS 15, Vancouver, Canada, MIT Press, 2003.

[21] Škrjanc I., Matko D., Predictive functional control based on fuzzy model for heat-exchanger pilot plant, *IEEE Trans. Fuzzy Syst.*, Vol. 8, No. 6, 705-712, 2000.

[22] 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, 599-621, 1998.

[23] 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, 399-401, 2001.

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