ETAI|2011|ETAI

In honour of
Professor Georgi M. Dimirovski

SPECIAL INTERNATIONAL CONFERENCE ON COMPLEX SYSTEMS: SYNERGY OF CONTROL, COMMUNICATIONS AND COMPUTING

Proceedings of COSY 2011 papers

Edited by
Tatjana Kolemishevska-Gugulovska and Mile J. Stankovski

OHRID, REPUBLIC OF MACEDONIA, SEPTEMBER 16-20, 2011

Society for Electronics, Telecommunications, Automation and Informatics – Macedonian NMO of the IFAC

IEEE Republic of Macedonia Section

In honour of
Professor Georgi M. Dimirovski

SPECIAL INTERNATIONAL CONFERENCE ON COMPLEX SYSTEMS: SYNERGY OF CONTROL, COMMUNICATIONS AND COMPUTING

PROCEEDINGS OF COSY 2011 PAPERS

Edited by
Tatjana Kolemishevska-Gugulovska and Mile J. Stankovski

Published by
ETAI – The Society for Electronics, Telecommunications, Automation, and Informatics of the Republic of Macedonia

Chairman of COSY 2011
Mogens Blanke

Organised by

Society for Electronics, Telecommunications, Automation, and Informatics (ETAI) – Macedonian NMO to the IFAC – in co-operation with the CAS/CIS/CSS IEEE Chapter of the IEEE Republic of Macedonia Section.

Faculty of Electrical Engineering and Information Technologies (FEIT) of the SSC&MU of Skopje

Sponsored by

IEEE Macedonian Section
SS Cyril and Methodius University of Skopje
St. Clement Ohridski University of Bitola
Faculty of Electrical Engineering and information Technologies – Skopje
Faculty of Technical Sciences – Bitola
Ministry of education and science
Engineers' Institution of the Republic of Macedonia

ISBN 978-9989-2175-8-6 National Library of R. Macedonia, Skopje

Skopje, September 2011

CIP – Катаголизација во публикација
Национална и универзитетска библиотека “Св. Климент Охридски”, Скопје

621.39(062)
004:681.5(062)
621.38(062)

SPECIAL international conference on complex systems (2011 ; Ohrid)
COSY 2011: proceedings of COSY 2011 papers / Special international conference on complex systems: synergy of control, communications and computing, Ohrid, September 16–20.2011 ; edited by Tatjana Kolemishevska-Gugulovska and Mile J. Stankovski. – Skopje : Society for electronics, telecommunications, automation, and informatics – Macedonian NMO of the IFAC, 2011. – 391 стр. : граф. прикази ; 30 см

Библиографија кон главите

ISBN 978-9989-2175-8-6
1. Kolemishevska-Gugulovska, Tatjana [уредник] 2. Stankovski, Mile J. [уредник]. – I. Kolemishevska-Gugulovska, Tatjana види Kolemiševska-Gugulovska, Tatjana
а) Телекомуникации – Собири б) Информатика – Автоматски контролни системи – Собири в) Електроника – Собири
COBISS.MK-ID 89183498

Proceedings Volume from the Special International Conference on Complex Systems:
Synergy of Control, Communications and Computing - COSY 2011
Hotel Metropol Resort, Ohrid, Republic of Macedonia, September, 16–20 2011

Control Algorithms Based on Gaussian Process Models: A State-of-the-Art Survey

Juš Kocijan *

* *Jozef Stefan Institute, Jamova 39, SI-1000 Ljubljana, Slovenia*
&
University of Nova Gorica, Nova Gorica, Slovenia

Abstract: Gaussian-process models provide a probabilistic, nonparametric modelling approach for black-box identification of nonlinear dynamic systems. They 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. Gaussian-process models contain noticeably less coefficients to be optimised than commonly used parametric models. This paper provides the state-of-the-art survey of control algorithms for dynamic systems described in publications where Gaussian-process models have been used for control design or as a part of controller. These methods, ranging from direct inverse control to advanced forms of adaptive control, may take into account the variance information provided by the Gaussian-process model. Beside the survey of control algorithms also trends, challenges and future opportunities are discussed in the contribution.

Keywords: Nonlinear control systems, nonlinear models, identification, probabilistic models, Gaussian processes.

1. INTRODUCTION

This paper provides the state-of-the-art survey of control algorithms for dynamic systems described in publications where Gaussian-process (GP) models have been used for control design or as a part of controller.

Dynamic systems control design utilises various kinds of black-box models. Gaussian-process models provide a probabilistic, nonparametric modelling approach for black-box identification of nonlinear dynamic systems. They 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. Gaussian-process models contain noticeably less coefficients to be optimised than commonly used parametric models. This approach to modelling is not considered as a replacement to any existing systems identification method, but rather as a complementary approach to modelling. The drawback of Gaussian process models is their considerable computational burden. This burden may be perceived as an obstacle for Gaussian process model usage in control applications.

The methods that are described in the following sections are mainly, but not exclusively, meant for the control of nonlinear and uncertain dynamic systems for which GP models are most suitable. Only general description of methods and application of GP models in a context of systems control is given in the paper. The nonlinear systems analysis that goes hand-in-hand with control design is addressed in many references, e.g., Khalil (2002), Slotine and Li (1991), and many others, but the formal analysis of probabilistic nonparametric models or models containing probabilistic nonparametric parts that would be applicable in the context of GP models is rare. The stability analysis as the most important issue in the closed-loop control design is not addressed formally in this text. Nevertheless, the stability of closed-loop systems containing GP models is in various publications shown with computer simulation, which is nowadays a common tool in engineering design.

The method used for control design always depends on the model of system to be controlled. Different sorts of models also mean different sorts of control methods as it will be seen in the following sections. The paper is devoted to control methods that have already been presented in literature. Reader is referred to literature where details of these methods can be found.

The structure of the paper is as follows. Principles of Gaussian process modelling are briefly described in the next section. Survey of control algorithms is given in the third section. The following control principles based on GP models are covered: inverse dynamics control, model-based predictive control, gain-scheduling control and adaptive control. Trends, challenges and research opportunities are given in the fourth section.

2. SYSTEMS MODELLING WITH GAUSSIAN PROCESSES

A Gaussian process (GP) model is a probabilistic, non-parametric model for the prediction of output-variable

⋆ This work was supported by grants P2-0001 and J2-2099 of the Slovenian Research Agency.

distributions. Its use and properties for modelling are thoroughly described in Rasmussen and Williams (2006). Here, only a brief description is given.

A Gaussian process is a collection of random variables which have a joint multivariate Gaussian distribution. Assuming a relationship of the form $y = f(\mathbf{x})$ between input $\mathbf{x}$ and output y, we have $y_1, \ldots, y_n \sim \mathcal{N}(0, \mathbf{\Sigma})$, where elements of the covariance matrix Σ, namely, $\Sigma_{pq} = \mathrm{Cov}(y_p, y_q) = C(\mathbf{x}_p, \mathbf{x}_q)$ give the covariance between output points corresponding to input points $\mathbf{x}_p$ and $\mathbf{x}_q$. Thus, the mean $\mu(\mathbf{x})$ and the covariance function $C(\mathbf{x}_p, \mathbf{x}_q)$ fully specify the Gaussian process.

The value of covariance function $C(\mathbf{x}_p, \mathbf{x}_q)$ expresses the correlation between the individual outputs $f(\mathbf{x}_p)$ and $f(\mathbf{x}_q)$ with respect to inputs $\mathbf{x}_p$ and $\mathbf{x}_q$. Note that the covariance function $C(\cdot, \cdot)$ can be any function that generates a positive semi-definite covariance matrix. It is usually composed of two parts:

$$C(\mathbf{x}_p, \mathbf{x}_q) = C_f(\mathbf{x}_p, \mathbf{x}_q) + C_n(\mathbf{x}_p, \mathbf{x}_q), \tag{1}$$

where C_f represents the functional part and describes the unknown system we are modelling, and C_n represents the noise part and describes the model of the noise.

A frequently chosen covariance function consists of the square exponential covariance function for functional part C_f and the constant covariance function for the noise part C_n, and is of the following form:

$$C(\mathbf{x}_p, \mathbf{x}_q) = v_1 \exp\left[-\frac{1}{2} \sum_{d=1}^{D} w_d (x_{dp} - x_{dq})^2 \right] + \delta_{pq} v_0 \tag{2}$$

where w_d, v_0, v_1 are the 'hyperparameters' of the covariance function, D is the dimension of input space, and $\delta_{pq} = 1$ if $p = q$ and 0 otherwise. The hyperparameters can be written as a vector $\mathbf{\Theta} = [w_1 \ldots w_D \; v_0 \; v_1]^T$. This covariance function is smooth and continuous. It presumes that the process is stationary and that the noise is white. Other forms and combinations of covariance functions suitable for various applications can be found in Rasmussen and Williams (2006). For a given problem, the hyperparameter values are learned using the data at hand.

To accurately reflect the correlations present in the training data, the hyperparameters of the covariance function need to be optimized. Due to the probabilistic nature of the GP models, the common model optimization approach where model parameters and possibly also the model structure are optimized through the minimization of a cost function defined in terms of model error (e.g., mean square error), is not readily applicable. A probabilistic approach to the optimization of the model seems more appropriate. Actually, instead of minimizing the model error, the log of marginal likelihood is maximized.

GP models can be easily utilized for regression calculation. Consider a set of N D-dimensional input vectors $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]$ and a vector of output data $\mathbf{y} = [y_1, y_2, \ldots, y_N]^T$. Based on the data $(\mathbf{X}, \mathbf{y})$, and given a new input vector $\mathbf{x}^*$, we wish to find the predictive distribution of the corresponding output y^*. Based on training set $\mathbf{X}$, a covariance matrix $\mathbf{K}$ of size $N \times N$ is determined. The overall problem of learning unknown parameters from data corresponds to the predictive distribution $p(y^*|\mathbf{y}, \mathbf{X}, \mathbf{x}^*)$ of the new target y, given the training data $(\mathbf{y}, \mathbf{X})$ and a new input $\mathbf{x}^*$. In order to calculate this posterior distribution, a prior distribution over the hyperparameters $p(\mathbf{\Theta}|\mathbf{y}, \mathbf{X})$ can first be defined, followed by the integration of the model over the hyperparameters

$$p(y^*|\mathbf{y}, \mathbf{X}, \mathbf{x}^*) = \int p(y^*|\mathbf{\Theta}, \mathbf{y}, \mathbf{X}, \mathbf{x}^*) p(\mathbf{\Theta}|\mathbf{y}, \mathbf{X}) d\mathbf{\Theta}. \tag{3}$$

The computation of such integrals can be difficult due to the intractable nature of the nonlinear functions. A solution to the problem of intractable integrals is to adopt numerical integration methods such as the Monte-Carlo approach. Unfortunately, significant computational efforts may be required to achieve a sufficiently accurate approximation.

An alternative approach based on the Maximum Likelihood optimization method has been developed and is applied to maximize the marginal likelihood. It can be restated as a cost function that is to be maximized. For numerical scaling purposes the log of the marginal likelihood is taken:

$$L(\mathbf{\Theta}) = -\frac{1}{2} log(|\mathbf{K}|) - \frac{1}{2} \mathbf{y}^T \mathbf{K}^{-1} \mathbf{y} - \frac{N}{2} log(2\pi). \tag{4}$$

A frequently used method for optimizing the cost function is a conjugate gradient method.

The predictive distribution of the GP model output for a new test input $\mathbf{x}^*$ has normal probability distribution with mean and variance

$$\mu(y^*) = \mathbf{k}(\mathbf{x}^*)^T \mathbf{K}^{-1} \mathbf{y}, \tag{5}$$

$$\sigma^2(y^*) = \kappa(\mathbf{x}^*) - \mathbf{k}(\mathbf{x}^*)^T \mathbf{K}^{-1} \mathbf{k}(\mathbf{x}^*), \tag{6}$$

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 $\kappa(x^*) = C(\mathbf{x}^*, \mathbf{x}^*)$ is the covariance between the test input itself.

The above modelling procedure was developed for modelling static nonlinearities, but it can be readily applied for modelling dynamic systems. Consider a dynamic system in the ARX representation, where the output at time step k depends on the delayed outputs y and the exogenous control inputs u:

$$y(k) = f(y(k-1), \ldots, y(k-n), u(k-1), \ldots, u(k-m)) + \epsilon(k) \tag{7}$$

where f denotes a function, $\epsilon(k)$ is white noise and the output $y(k)$ depends on the state vector $\mathbf{x}(k) = [y(k-1), y(k-2), \ldots, y(k-n), u(k-1), u(k-2), \ldots, u(k-m)]^T$ at time step k.

Assuming the signal is known up to k, we wish to predict the output of the system l steps ahead, i.e., we need to find the predictive distribution of $y(k+l)$ corresponding to $\mathbf{x}(k+l)$. Multiple-step-ahead predictions of a system modelled by (7) can be achieved by iteratively making repeated one-step-ahead predictions, up to the desired horizon.

A noticeable drawback of the system identification with GP models is the computation time necessary for the modelling. Regression based on GP models involves several

matrix computations in which the load increases with the third power of the number of input data, such as matrix inversion and the calculation of the log-determinant of the used covariance matrix. To overcome the computational-limitation issues and to also make use of the method for large-scale dataset applications, numerous authors have suggested various sparse approximations as well as on-line modelling (Rasmussen and Williams, 2006).

A survey of control algorithms that make use of Gaussian process models is given in the next section.

3. SURVEY OF CONTROL ALGORITHMS

3.1 Inverse dynamics control

When machine learning methods are preliminary introduced for control, frequently the following scheme appears. An inverse model of the process is developed to be connected in series with the process and therefore an open-loop control system is formed. This kind of approach is usually not meant as effective control solution, but mainly as a demonstration of particular machine learning method.

The basic principle in brief is as follows. If the system to be controlled can be described by input-output model

$$y(k+1) = f(y(k),\ldots,y(k-n+1),\ldots,u(k),\ldots,u(k-m)) \quad (8)$$

then the corresponding inverse model is

$$\tilde{u}(k) = \hat{f}^{-1}(y(k+1),y(k),\ldots,y(k-n+1),\ldots,u(k-1),\ldots,u(k-m)) \quad (9)$$

where the notation $\hat{x}$ denotes the estimator of x.

Assuming that this inverse system has been obtained it can be used to generate control input that approaches desired process output, when the reference input is given to the inverse model. This means that samples of y in equation (9) are replaced by reference values r.

The principle is illustrated in Fig. 1.

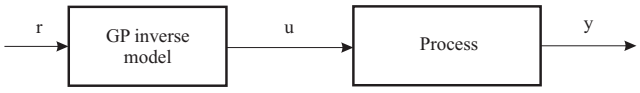

Fig. 1. General block scheme of the direct inverse control

There is a list of assumptions and constraints that need to be satisfied for a such system to be practical implemented. The assumptions necessary for the open-loop control to be operational are: no disturbances in the system, no uncertainties and changes in the process and open-loop controller that is the perfect inverse of the process in the region of operation. Since these assumptions are frequently not fulfilled in the real world, the inverse system is usually realised as the adaptive system, where the controller matches any changes in the process on-line.

Training of the inverse model requires that the process and inverse model are input-output stable. This is because signals are always constrained in magnitude, which disables the open-loop control of unstable systems. Even in the case of computer simulation, inputs and outputs can not be infinitely large.

When the mentioned assumptions are satisfied, the inverse model can be modelled from appropriately selected outputs and inputs of the process following equation (9). In the case of open-loop controller realisation with GP model, only the mean value of controller output prediction is the input into the process to be controlled.

The reinforcement learning of the described open-loop controller actions based on GP model is introduced in Engel et al. (2006). The entire system is implemented only as a computer simulation and meant to demonstrate reinforcement learning, rather then practically applicable control system principle.

Different approach to open-loop control is given in Ko et al. (2007) where GP model in the role of open-loop controller is taught with reinforcement learning to mimic the outputs of an optimal and closed-loop conventional controller.

Another method that uses inverse model for cancelling nonlinearities of the process to be controlled is Inverse Dynamics Control (Nguyen-Tuong et al., 2008b) illustrated in Fig. 2.

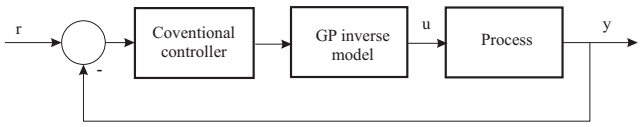

Fig. 2. General block scheme of the inverse dynamics control

This is the closed-loop method that contains conventional controller to deal with miss-matches between nonlinearity compensator and process as well as with the process disturbances. Such scheme with different sorts of inverse models is commonly used for dynamics control in robotics. The inverse model can be identified off-line or on-line. The application with the GP model of inverse process dynamics that is identified off-line is given in Nguyen-Tuong et al. (2008b) for a robot control investigation.

Feedforward that eliminates the process nonlinearities is another control method that is used mainly in robotics. Its principle is depicted in Fig. 3.

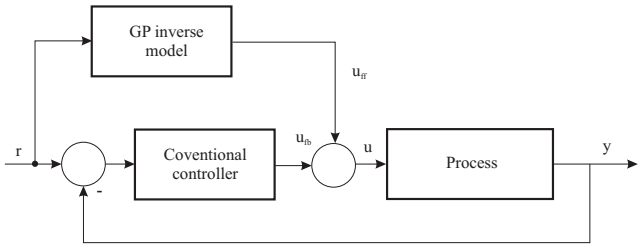

Fig. 3. General block scheme of existing control system with GP inverse model as feedforward for improvement of closed-loop performance

The control signal consists of feedforward and feedback component $u = u_{ff} + u_{fb}$. The feedback loop with a conventional, frequently linear, controller is required to maintain stability and disturbance rejection for this control system purposed for set-point tracking. The feedforward that is a stable inverse model is meant to compensate for process nonlinearities. The inverse model has to be as precise as possible in the region of operation to enable the required performance. The closed-loop performance is deteriorated in the case of unmodelled nonlinearities. The feedforward is generally considered as a function of the desired set-point, in the case of robotic control that would mean desired robot trajectories.

The concept has some practical advantages (Norgaard et al., 2000). First, the data necessary for the inverse model can be collected from beforehand assembled closed-loop system without the feedforward component. Second, the feedforward signal can be introduced gradually during control system implementation from the reason of cautiousness. Third, in the case that the inverse dynamics is to be avoided, only static feedforward can be used with feedback controller compensating for erroneous feedforward signal.

The inverse model can be identified off-line or on-line. The case when inverse GP model is identified off line and used in such control set-up is described in Nguyen-Tuong et al. (2008a) and Nguyen-Tuong et al. (2008b). The adaptive cases are mentioned in Section 3.4.

None of these applications of inverse GP models uses entire information from prediction distribution rather they focus on the mean value of the prediction. In such a way the full potential of GP models for this kind of control is not utilised entirely, e.g., information of variances could be used for maintaining or indicating the region of the nominal closed-loop performance.

3.2 Model-based predictive control

Model Predictive Control (MPC)(Maciejowski, 2002) is a common name for 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 optimizes 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 period.

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 optimization method.

In our case we are interested in the applications of Nonlinear Model Predictive Control (NMPC) principle with a Gaussian process model. Stochastic NMPC problems are formulated in the applications where the system to be controlled is described by a stochastic model such as the GP model. Stochastic problems like state estimation are studied for long time, but, in our case, we explore only stochastic NMPC problem. Nevertheless, most known stochastic MPC approaches are based on parametric probabilistic models. Alternatively, the stochastic systems can be modeled with nonparametric models which can offer a significant advantage compared to the *parametric* models. This is related to the fact that the *nonparametric* probabilistic models, like Gaussian process models, provide information about prediction uncertainties which are difficult to evaluate appropriately with the parametric models.

The nonlinear model predictive control as it is applied with the Gaussian process model can be in general described with a block diagram, as depicted in Fig. 4. The model is fixed, identified off-line, which means that the resulting control algorithm is not an adaptive one. The structure of the entire control loop is therefore less complex as in the case of adaptive control.

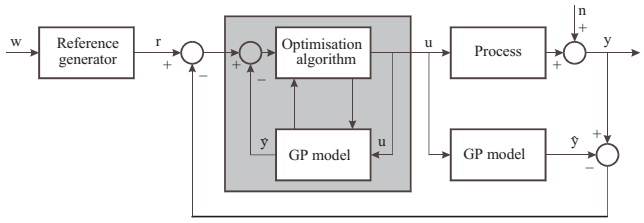

Fig. 4. Block diagram of model predictive control system

The control objective is to be achieved by minimization of the cost function. The cost function penalizes deviations of the predicted controlled outputs $\hat{y}(k+j|k)$ from a reference trajectory $r(k+j|k)$. This reference trajectory may depend on measurements made up to time instant k. Its initial point may be the output measurement $y(k)$, but can also be a fixed set-point, or some predetermined trajectory. The minimization of cost function, in which future control signal $(u(k))$ is calculated, can be subject to various constraints (e.g., input, state, rates, etc).

MPC solves a constrained control problem. The single-input, single-output case is elaborated here, but it can be generalised to multiple-input, multiple-output case. A stochastic nonlinear discrete-time system can be described in the input-output form (7) or in the very common state-space form:

$$\begin{aligned} \mathbf{x}(k+1) &= F\left(\mathbf{x}(k), u(k)\right) + \epsilon_1(k) \\ y(k) &= G\left(\mathbf{x}(k), u(k)\right) + \epsilon_2(k) \end{aligned} \tag{10}$$

where $\mathbf{x}$, u and y are the state, input and output variables respectively, $\epsilon_i(k);\ i = 1, 2$ are Gaussian disturbances, and F, G are nonlinear continuous functions.

Corresponding input and state constraints of the general form are:

$$u(k) \in \mathcal{U}; \tag{11}$$

$$x(k) \in \mathcal{X} \tag{12}$$

where $\mathcal{U}$ and $\mathcal{X}$ are sets of possible inputs and feasible states respectively and the optimisation problem is

$$J_{\text{opt}}(k) = \min_{\mathbf{u}} J(\mathbf{u}, \mathbf{x}(k), r(k), u(k-1)) \tag{13}$$

where the cost function of a general form is

$$J(\mathbf{u}, \mathbf{x}(k), r(k), u(k-1)) = \sum_{j=0}^{N-1} L\left(\hat{\mathbf{x}}(k+j|k), \hat{u}(k+j-1|k)\right) \quad (14)$$

where L is a nonlinear continuous function and it is assumed that the cost falls to zero once the state has entered the set of optimal states $\mathcal{X}_0$, namely $L(\mathbf{x}, u) = 0$ if $x \in \mathcal{X}_0$. The following terminal constraint is imposed:

$$\hat{\mathbf{x}}(k+N|k) \in \mathcal{X}_0. \quad (15)$$

This is a general form and MPC formulations vary with various models, cost functions and parameters.

Frequently used cost function in MPC literature is:

$$J(k) = ||\mathrm{E}\{y(k+N|k)\} - r(k)||^2_{\mathbf{P}} + \sum_{j=0}^{N-1} \left[||\mathrm{E}\{y(k+j|k)\} - r(k)||^2_{\mathbf{Q}} + ||\Delta u_{k+j}||^2_{\mathbf{R}}\right] \quad (16)$$

where N is a finite horizon and $\|\mathbf{x}\|_{\mathbf{A}} = \sqrt{\mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x}}$; $\mathbf{A} = \mathbf{P}, \mathbf{Q}, \mathbf{R}$ are positive definite matrices and the notation $\mathrm{E}\{\cdot\}$ denotes an 'expectation' conditional upon data available up to and including current time instant k.

There are many alternative ways of how NMPC can be realised with Gaussian process models:

Cost function. The cost function (14) is a general one and various special cost functions can be derived out of it. It is well known that the selection of the cost function has a major impact on the amount of computation.

Optimization problem for Δu instead of u. This is not just a change of the formalism, but also enables forms of MPC containing integral action.

Process model. The process model can be determined off-line and fixed for the time of operation or determined on-line during the operation of controller. The on-line model identification is described in Section 3.4.

Soft constraints. Using constraint optimization algorithms is very demanding for computation and soft constrains. In other words weights on constrained variables in cost function, can be used to decrease the amount of computation. More on this topic can be found in Maciejowski (2002).

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

Various predictive control methods can be applied with GP models depending on designers choice and imposed constraints. Using GP models does not impose any particular constraint on cost function, optimisation method or any other element of choice for predictive control design.

An application of model predictive control with the GP model using the general cost function described with equation (16) can be found in Grancharova and Kocijan (2007). Investigations of three special forms of MPC are more frequent in the literature. These three algorithms are: Internal Model Control (IMC), Predictive Functional Control (PFC), and the approximate explicit control, which are described in subsequent subsections. A principle different from the listed, where control is based on the estimation and the multiple-step prediction of system output in combination with fuzzy models is given in (Palm, 2007).

Internal Model Control

In this strategy the controller is chosen to be an inverse of the plant model. Internal model control is one of the most commonly used model–based techniques for the control of nonlinear systems. It can be considered also as the simplest form of MPC with prediction and control horizon equal to one step. IMC with the GP model is elaborated in Gregorčič and Lightbody (2002), Gregorčič and Lightbody (2003a) and Gregorčič and Lightbody (2005). The description of IMC with the GP model hereafter is adopted from these references.

The main difference between the various internal model control approaches is in the choice of the internal model and its inverse. It was shown in Gregorčič and Lightbody (2003b) that the GP model based on the squared-exponential covariance function is not analytically invertible. Instead of calculating the exact inverse, a numerical approach such as successive approximation or Newton-Raphson optimisation method, can be used to find the control effort to solve the following equation:

$$H(\mathbf{u}(k), \mathbf{y}(k)) - q(k) = 0, \quad (17)$$

where $\mathbf{u}(k) = [u(k) \dots u(k-m)]^T, \mathbf{y}(k) = [y(k) ... y(k-n)]^T, H(\mathbf{u}(k), \mathbf{y}(k)) = \hat{y}(k+1)$ and $q(k)$ is the controller input.

The GP model is trained as a one-step ahead prediction model. The IMC strategy requires the use of the parallel model of equation $\hat{y}(k+1) = \hat{H}(\mathbf{u}(k), \hat{\mathbf{y}}(k))$. This GP model is then included in the IMC structure and the numerical inverse of the equation (17) is found at each sample. The IMC works well when the control input and the output of the system are in the region where the model was trained. As soon as the system moves away from the well modelled region this can cause sluggish and, in certain cases, also unstable closed-loop system behaviour.

Since poor closed-loop performance is the result of the the model being driven outside its trained region, the naive approach would be to constrain the control input. When the system is driven in this untrained portion of the operating space, the increase of the predicted variance will indicate a reduced confidence in the prediction. This increase of variance can be used as a constraint in the optimisation algorithm utilised to solve equation (17). This concept is shown in Fig. 5.

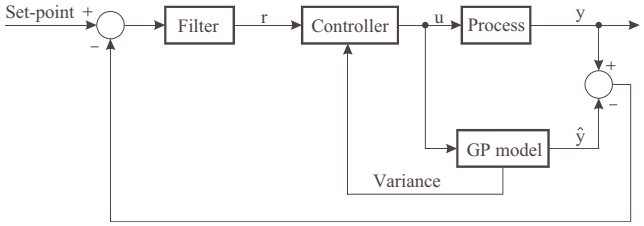

Fig. 5. Variance-constrained Internal Model Control Structure

The basic idea of the algorithm is to optimise the control effort so that the variance does not increase above its

predefined limit. Since the GP model is not analytically invertible and numerical approaches have to be utilised to find the inverse of the model at each sample time, the associated computation load rises rapidly with the number of training data points. This is the main drawback of the GP modelling approach for IMC.

Predictive Functional Control

Predictive functional control is a form of the predictive control that in principle is no different to a general predictive control. Its distinct features are relatively low number of so called coincidence points, the use of a reference trajectory, which is distinct from the set-point trajectory and the assumption that the future input is a linear combination of a few simple basis functions. More details can be found in, e.g., Maciejowski (2002).

In the following description the PFC with one coincidence point and constant output within the control horizon is used. Variants of this kind of predictive control with the GP model are described in Kocijan et al. (2003), Kocijan and Leith (2004), Kocijan et al. (2004), Kocijan and Murray-Smith (2005), Likar and Kocijan (2007) and Ažman and Kocijan (2008). The predictive control based on the GP model was for the first time introduced in published reference in Kocijan et al. (2003).

A moving-horizon minimisation problem of the form (Maciejowski, 2002)

$$J_{\text{opt}}(k) = \min_{\mathbf{u}(k)} \; [r(k+P) - \mathrm{E}\{\hat{y}(k+P)\}]^2 \tag{18}$$

subject to constraints:

$$\begin{aligned} \text{var } \hat{y}(k+P) &\leq k_v \\ \mid \mathbf{u}(k) \mid &\leq k_{ih} \\ \mid \Delta\mathbf{u}(k) \mid &\leq k_{ir} \\ \mid \mathbf{x}(k) \mid &\leq k_{sh} \\ \mid \Delta\mathbf{x}(k) \mid &\leq k_{sr} \end{aligned} \tag{19}$$

is applied as the first presented choice, where $\mathbf{u}(k) = [u(k) \dots u(k+P)]$ is the input signal, P is the coincidence point, i.e., the point where a match between output and the reference value is expected, and inequalities (19) represent constraint on the output variance k_v, the input hard constraint k_{ih}, the input rate constraint k_{ir}, the state hard constraint k_{sh} and the state rate constraint k_{sr} respectively. These constraints are in general functions of some scheduling variable in the general form, but are many times set to be constant values. The process model is a GP model.

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.

A possible alternative selection of the cost function that avoids constrained optimisation and is therefore computationally less demanding would be

$$J(k) = \mathrm{E}\{[r(k+P) - \hat{y}(k+P)]^2\}. \tag{20}$$

Using the fact that $\mathrm{var}\{\hat{y}\} = \mathrm{E}\{\hat{y}^2\} - \mathrm{E}\{\hat{y}\}^2$, the cost function can be written as

$$J(k) = [r(k+P) - \mathrm{E}\{\hat{y}(k+P)\}]^2 + \mathrm{var}\{\hat{y}(k+P)\}. \tag{21}$$

The control strategy with cost function (21) is 'to avoid' going into regions with higher variance. The term 'higher variance' does not specify any specific value. In the case that controller does not seem to be 'cautious' enough, a 'quick-and-dirty' option is that the variance term can be weighted with a constant λ_{var} to enable shaping of the closed-loop response according to variance information:

$$J(k) = [r(k+P) - \mathrm{E}\{\hat{y}(k+P)\}]^2 + \lambda_{\text{var}}\mathrm{var}\{\hat{y}(k+P)\} \tag{22}$$

Beside the difference in the optimisation algorithm the presented options give also a design choice on how 'safe' the control algorithm is. In the case when it is very undesirable to go into 'unknown' regions the constrained version may be better option.

Approximate explicit stochastic nonlinear predictive control

Bemporad et al. Bemporad et al. (2000) have proposed an approach to implement the MPC, where the computation effort is moved off-line. The MPC formulation described up-to-now provides the control action $u(k)$ as a function of states $\mathbf{x}(k)$ defined implicitly by the cost function and constraints. By treating $\mathbf{x}(k)$ as a vector of parameters, the goal of the proposed method is to solve MPC problem off-line with the respect to all values of $\mathbf{x}(k)$ of interest and make this dependence explicit. Solving this problem means solving a multi-parametric Quadratic Programme (mp-QP). The solution of the mp-QP is a continuous and piecewise affine function of $\mathbf{x}$. When the problem involves the nonlinear model of the system, then solving a multi-parametric Quadratic Programme becomes solving a multi-parametric NonLinear Programme (mp-NLP).

The benefits of an *explicit* solution, in addition to the efficient on-line computations, include also the verifiability of implementation, which is an essential issue in safety-critical applications. For the nonlinear and stochastic MPC the benefits of *explicit* solutions are even higher than for linear MPC, since the computational efficiency and verifiability are even more important. Grancharova et al. (2007b) propose an approach for off-line computation of *explicit* suboptimal stochastic NMPC controller for constrained nonlinear systems based on a GP model. The approach is based on the multi-parametric Nonlinear Programming (mp-NLP) ideas and represents an extension of the approximate methods in Grancharova et al. (2007a) and Johansen (2004). Approximate explicit nonlinear predictive control based on the GP model has been elaborated in Grancharova et al. (2007b) and Grancharova et al. (2008).

3.3 Gain-scheduling control

The gain-scheduling method is probably the most wide spread nonlinear control design method. It has been successfully applied in fields ranging from process control to aerospace engineering. The basic idea behind the approach is called divide-and-conquer method where a nonlinear system is divided into local subsystems that are modelled as linear dynamic systems. A linear control problem is then solved for each of these subsystems. The global control solution – called gain-scheduling control – is afterwards put together from partial local solutions. Overviews of the

gain-scheduling method and its applications can be found in Rugh (1991), Rugh and Shamma (2000) and Leith and Leithead (2000).

The traditional gain-scheduled controller, which is adjusted with reference to an externally measured vector of variables, $\varrho(\mathbf{x}(t)\mathbf{u}(t))$, has the form

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}_c(\varrho(t))\mathbf{x} + \mathbf{B}_c(\varrho(t))\mathbf{u}\\ \mathbf{y} &= \mathbf{C}_c(\varrho(t))\mathbf{x} + \mathbf{D}_c(\varrho(t))\mathbf{u}\end{aligned} \tag{23}$$

where $\mathbf{A}_c(\varrho(t)), \mathbf{B}_c(\varrho(t)), \mathbf{C}_c(\varrho(t)), \mathbf{D}_c(\varrho(t))$ are matrix functions. Dynamic properties change with so called scheduling vector $\varrho(\mathbf{x}(t)\mathbf{u}(t))$ but, provided that the rate of change is not too rapid, then the dynamic properties of the time-varying controller are similar to those of the linear controllers obtained by 'freezing' the value of $\varrho(t)$; that is, the nonlinear controller inherits the dynamic properties of the family of linear controllers (Leith and Leithead, 1998).

A block scheme showing a general principle of gain-scheduling control is given in Fig. 6.

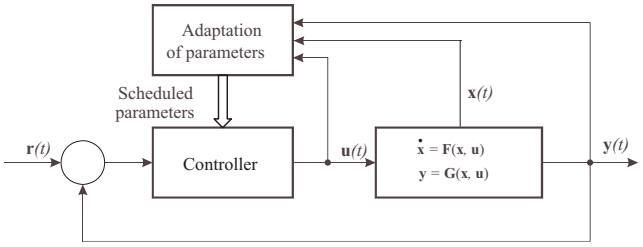

Fig. 6. General block scheme of the closed-loop system with a gain-scheduling controller

Number of nonlinear identification methods including conventional nonparametric GP model identification methods provide models that can be used only with model based predictive control. A Fixed-Structure Gaussian-Process (FSGP) model is introduced in Ažman and Kocijan (2006). The FSGP model is a model with predetermined linear structure with varying and probabilistic parameters represented by Gaussian-process models.

FSGP can be used for wider range of control design methods and not only those which are based on black-box GP models. One possible control design approach is the gain-scheduling control design. In this case local controllers are designed for selected local model of the process. Gain-scheduling control based on GP model, namely on FSGP model, is described in Ažman and Kocijan (2006) and Ažman and Kocijan (2009).

Selection of local process models for control design depends on region where closed-loop dynamics is expected and is in general not the same as the set of local models used for process modelling. It is sensible to keep the system where its model is good, i.e., where variances of local models' parameters are small. Variances of GP models contained in the FSGP model provide this information. Parameters of the local controllers depend on the same scheduling variables as the associated process local model parameters.

3.4 Adaptive control

Adaptive controller is the controller that continuously adapts to some changing process. Adaptive controllers emerged in early sixties of the previous century. At the beginning these controllers were mainly adapting themselves based on linear models with changing parameters. Since then several authors have proposed the use of non-linear models as a base to build nonlinear adaptive controllers. These are meant for the control of time-varying nonlinear systems or of time-invariant nonlinear systems that are modelled as parameter-varying simplified nonlinear models.

Various divisions of adaptive control structures are possible. One possible division (Isermann et al., 1992) is into open-loop and closed-loop adaptive systems.

Open-loop adaptive systems are gain-scheduling or parameter-scheduling controllers already described in Section 3.3. Closed-loop adaptive systems can be further divided to dual and non-dual adaptive systems.

Dual adaptive systems (Filatov and Unbehauen, 2000; Wittenmark, 2002) are those where the optimisation of the information collection and the control action are pursued at the same time. The control signal should ensure that the system output cautiously tracks the desired reference value and at the same time excites the plant sufficiently to accelerate the identification process. The solution to the dual control problem is based on dynamic programming and the resulting functional equation is often the Bellman equation. Not a large number of such controllers have been developed.

The difficulties to find the optimal solution for dual adaptive control lead to suboptimal adaptive dual controllers (Filatov and Unbehauen, 2000; Wittenmark, 2002) obtained by either various approximations or by reformulating the problem. Such a reformulated adaptive dual control problem is when a special cost function is considered, which consists of two added parts: control losses and an uncertainty measure. This is appealing for application with the Gaussian process model that provides measures of uncertainty.

Many adaptive controllers in general are based on the separation principle (Wittenmark, 2002) that implies separate estimation of system model, i.e., system parameters, and the application of this model for control design. When the identified model used for control design and adaptation is presumed to be the same as the true system then the adaptive controller of this kind is said to be based on certainty equivalence principle and such adaptive control is named non-dual adaptive control. The control actions in non-dual adaptive control do not take any active actions that will influence the uncertainty.

When using the GP model for the adaptive control, different from gain-scheduling control described in Section 3.3, the GP model is identified on-line and this model is used in the control algorithm. The block scheme showing the general principle of adaptive control with the GP model identification is given in Fig. 7. It is sensible that advantages of GP models are considered in the control design, which relates the GP model-based adaptive control

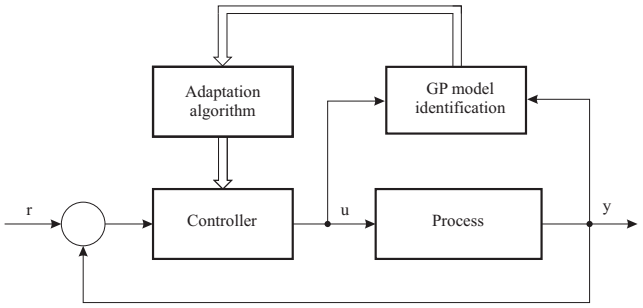

Fig. 7. General block scheme of the closed-loop system with adaptive controller

at least to suboptimal dual adaptive control principles. The uncertainty of model predictions obtained with the GP model prediction are dependent, among others, on local learning-data density, and the model complexity is automatically related to the amount and the distribution of the available data – more complex models need more evidence to make them likely. Both aspects are very useful in sparsely-populated transient regimes. Moreover, since weaker prior assumptions are typically applied in a nonparametric model, the bias is typically lower than in parametric models.

The above ideas are indeed related to the work done on the dual adaptive control, where the main effort has been concentrated on the analysis and design of adaptive controllers based on the use of the uncertainty associated with parameters of models with a fixed structure (Filatov and Unbehauen, 2000; Sbarbaro and Murray-Smith, 2005).

The major differences in up-to-now published adaptive systems based on GP models are in the way how the on-line model identification is pursued.

Increasing the size of the covariance matrix, i.e., 'blow-up model', with the in-streaming data and repeating model optimisation is used in papers (Murray-Smith and Sbarbaro, 2002), (Murray-Smith et al., 2003), (Sbarbaro et al., 2004), (Sbarbaro and Murray-smith, 2005) and (Sbarbaro and Murray-Smith, 2005), where more attention is devoted to control algorithms and their benefits based on information gained from the GP model and not on the model identification itself.

Another adaptive control algorithm implementation is control with feedback for cancelling nonlinearities, described already in Section 3.1 with the on-line learning of the inverse model. This sort of adaptive control with the increasing covariance matrix with the in-streaming data is described in Nguyen-Tuong and Peters (2008). Two sorts of on-line learning for the mentioned feedforward contained control is described in Nguyen-Tuong et al. (2010). The first sort is with moving window strategy, where the old data are dropped from the on-line learned model, while the new data is accommodated, the second one accommodates only new data with sufficient information gain. These applications of referenced inverse GP models do not use entire information from the prediction distribution, but like those non-adaptive based on the same principle from Section 3.1 they focus on the mean value of prediction.

Alternatively to listed adaptive controllers, the adaptive control system principle described by Petelin and Kocijan (2011) is based on the evolving GP model. The basic idea of the control based on the evolving system model is that the system GP model evolves with the in-streaming data and the information about system from the model is then used for its control. One option is that the information can be in the form of the GP model prediction for one or several steps ahead which is then used to calculate the optimal control input in the controlled system. Different possibilities exist for the evolving GP model depending on the level of changes we accommodate in the evolving system model. On the other hand, various control algorithms can be used depending on the GP model or closed-loop requirements.

A lot of GP model-based adaptive-control algorithms from the referenced publications are based on the Minimum Variance controller. One of the reasons is that the Minimum Variance controller explores the variance that is readily available with the GP model prediction.

The Minimum Variance controller in general (Isermann et al., 1992) looks for a control signal $u(k)$ in time instant k, that will minimize the following cost function:

$$J_{\mathrm{MV}} = \mathrm{E}\{(r(k) - y(k+m))^2\} \tag{24}$$

Taking the expected value of a variable squared gives the variance of that variable. In this case, J_{MV} therefore refers to the variance of the error between set-point $r(k)$ and the controlled output m-time steps in the future, $y(k+m)$. The desired controller is thus the one that minimizes this variance, hence the name Minimum Variance control. The optimal control signal u_{opt} can be obtained by minimising selected cost function. The minimisation can be done analytically, but also numerically, using any appropriate optimisation method.

The cost function (24) can be expanded with a penalty term λ_u on the control effort:

$$J_{\mathrm{MV}_2} = \mathrm{E}\{(r(k) - y(k+m))^2\} + \lambda_u u^2(k) \tag{25}$$

The term λ_u can be used for 'tuning' of the closed-loop system performance. As mentioned by Murray-Smith and Sbarbaro (2002), this cost function can be written as:

$$J_{\mathrm{MV}_2} = (r(k) - \mathrm{E}\{y(k+m)\})^2 + \mathrm{var}\{y(k+m)\} + \lambda_u u^2(k) \tag{26}$$

where the second therm represents the model uncertainty which is available from the GP model prediction and can be used in the optimal control signal minimization. Note that the most of conventional work has ignored it, or have added extra terms to the cost function, or has pursued other sub-optimal solutions. The simplest form of the cost function (25) is when $m = 1$ and therefore only the one-step-ahead prediction is used. The case when $m > 1$ is elaborated in Section 3.2 and the adaptive-control application, though with increasing dimensions of the covariance matrix with in-streaming data, is given in Murray-Smith et al. (2003) and can be considered as an adaptive model predictive control.

The cost function (25) can be further expanded with other forms of penalty leading to Generalized Minimum Variance control. Possible alternative to cost function (26) is

$$J_{\mathrm{GMV}} = Q(q^{-1})(r(k) - \mathrm{E}\{y(k+m)\})^2 + \mathrm{var}\{y(k+m)\} + R(q^{-1})u^2(k) \quad (27)$$

where polynomials $Q(q^{-1})$ and $R(q^{-1})$ are defined as:

$$Q(q^{-1}) = Q_0 + Q_1 q^{-1} + \ldots + Q_{n_q} q^{n_q} \quad (28)$$

$$R(q^{-1}) = R_0 + R_1 q^{-1} + \ldots + R_{n_r} q^{n_r} \quad (29)$$

where q^{-1} is a unit backward shift operator. The polynomial coefficients can be used as tuning parameters. Similar cost function is used in Sbarbaro and Murray-Smith (2005).

Yet another possibility, suggested in Sbarbaro and Murray-smith (2005) in the context of GP models is the enhanced version of Generalized Minimum Variance controller known as Generalized Linearising controller (Goodwin et al., 2001):

$$J_{\mathrm{GLC}} = (1-\nu)\mathrm{E}\{(r(k) - y(k+m))^2\} + \nu(R(q^{-1})u(k) - u_r)^2 \quad (30)$$

where u_r is the input associated to r, the coefficients of $R(q^{-1})$ and ν are tuning parameters. When $\nu = 0$, the cost function corresponds to the minimum variance cost function, and when $\nu = 1$, the minimum of the cost function corresponds to a simple feedforward control. On the other hand if the open-loop system is stable but with an unstable inverse, then $\nu = 1$ provides a stable controller.

Furtherly, papers Murray-Smith and Sbarbaro (2002) and Sbarbaro and Murray-smith (2005) describe the control of an affine nonlinear system of the form

$$y(k+1) = f(x(k)) + g(x(k))u(k) + \xi(k+1), \quad (31)$$

which allows the combination of squared exponential and linear covariance function for the GP model and the combination with Minimum Variance control. This application is generalised to multiple-input multiple-output case in Sbarbaro et al. (2004).

Gaussian process dynamic programming

The method named Gaussian Process Dynamic Programming (GPDP) is a Gaussian-process model-based adaptive control algorithm with the closest proximity to dual adaptive control. The details of the method are described in Deisenroth et al. (2009). The following description is summarised from Deisenroth et al. (2009), Deisenroth (2010) and Deisenroth and Rasmussen (2009a). The evolution of method can be followed through time with publications Rasmussen and Kuss (2004); Deisenroth et al. (2008a,b); Rasmussen and Deisenroth (2008); Deisenroth and Rasmussen (2009a); Deisenroth et al. (2009); Deisenroth and Rasmussen (2009b) and Deisenroth (2010), where the method is named more generally as Probabilistic Inference and Learning for COntrol (PILCO).

The general idea of the method is to learn the system model on-line and control the closed-loop system taking into account the probabilistic model of the process. The algorithm can be divided into three layers: top level for controller adaptation, intermediate layer for approximate inference for long-term predictions and bottom layer for learning the model dynamics.

A discrete-time system described by equation (32) is considered throughout the method description

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)) + \mathbf{w}, \quad (32)$$

where $\mathbf{x}$ is a vector of states, $\mathbf{u}$ is a control vector and $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_w)$ a Gaussian distributed noise random variable, where $\mathbf{\Sigma}_w$ is diagonal. The model dynamics or transition function $\mathbf{f}$ mapping a pair $\mathbf{x}(k), \mathbf{u}(k)$ to a successor state $\mathbf{x}(k+1)$ is assumed to evolve smoothly over time and is time-invariant (Deisenroth et al., 2009). The dynamics function is modelled with a Gaussian process model. For a stochastic system, the noise term $\mathbf{w}$ in the system equation (32) is the process noise. The obtained dynamics GP model of the underlying stochastic dynamics function $\mathbf{f}$ contains two sources of uncertainty: First is the uncertainty about the underlying system function itself, and second the uncertainty induced by the process noise. With the increasing amount of absorbed information the first source of uncertainty tends to zero, whereas stochasticity due to the process noise $\mathbf{w}$ is always present. Therefore, only the uncertainty about the model vanishes with the time of operation.

The learning of the model dynamics is implemented as an on-line learning of the model. Active learning is the strategy for optimal data selection to make learning more efficient. In the case of GPDP method, training data are selected according to a utility function. The utility function that rates the quality of candidate data in the context of GPDP is

$$U = \rho \mathrm{E}\{J_{\mathrm{opt}}(k)\} + \frac{\beta}{2} \log \mathrm{var}\{J_{\mathrm{opt}}(k)\}, \quad (33)$$

where J_{opt} is the cost function, modelled by another GP model that satisfied the Bellman equation (see equation (35)) for all states and with weighting factors ρ, and β.

To find an optimal control law, i.e., a policy, guiding the system from an initial state to the goal state, Bayesian active learning is incorporated into GPDP such that only a relevant part of the state space is explored. The GP model of the dynamics is built on-line. This on-line algorithm largely exploits information which is already computed within GPDP. The combination of active learning and GPDP is called ALGPDP in the sequel. Instead of globally, sufficiently accurate value function model, ALGPDP aims to find a locally appropriate value function model $J_{\mathrm{opt}}(k)$ in the vicinity of the most promising trajectories from the initial states to the goal state.

ALGPDP starts from a small initial set of state vectors $\mathcal{X}(N)$, where N is the length of the optimisation horizon. Using Bayesian active learning, new state vectors are added to the current set $\mathcal{X}(k)$ at any time step k. The sets $\mathcal{X}(k)$ serve as training inputs for both the GP model describing system and the GP model describing the cost function. At each time step, GP models are updated to incorporate the most recent information.

The controller of the adaptive control system optimises some long-term performance measure. The control law is a mapping from a state vector into a control vector that assigns a value of control signal to each state, i.e., a nonlinear state controller. For an initial state $\mathbf{x}(0)$ and the selected controller, the expected cumulative cost of a finite N-step optimisation horizon is

$$J(\mathbf{x}(0)) = \mathrm{E}\left\{\phi(\mathbf{x}(N)) + \sum_{k=0}^{N-1} L(\mathbf{x}(k), \mathbf{u}(k))\right\}. \quad (34)$$

The function $\phi(N)$ is a control-independent terminal cost that incurs at the last time step N. The immediate cost is denoted as $L(\mathbf{x}(k), \mathbf{u}(k))$. An optimal control sequence $\mathbf{u}$ for the N-step problem minimises equation (34) for any initial state $\mathbf{x}(0)$. The associated state-value function J_{opt} satisfies the Bellman equation:

$$J_{\mathrm{opt}}(\mathbf{x}(k)) = \min_{\mathbf{u}}(L(\mathbf{x}(k), \mathbf{u}(k)) + \gamma \mathrm{E}\left\{J_{opt}(\mathbf{x}(k+1)|(\mathbf{x}(k), \mathbf{u}(k))\right\} \quad (35)$$

for all states $\mathbf{x}(k)$. The successor state for a given state-action pair $(\mathbf{x}(k), \mathbf{u}(k))$ is denoted by $\mathbf{x}(k+1)$. Assuming the time-additive cost and Markovian transitions, the minimal expected cumulative cost can be calculated by dynamic programming.

Reader is referred to Deisenroth et al. (2009) for details and demonstration of the method. Unfortunately, according to the method's authors (Deisenroth et al., 2009), ALGPDP cannot be directly applied to a dynamic system, because it is often not possible to experience arbitrary state transitions. Authors suggest a combination with ideas from Rasmussen and Deisenroth (2008) to make a possible adaptation to real-word problems so that algorithm experience most promising trajectories following the current policy.

4. TRENDS, CHALLENGES AND RESEARCH OPPORTUNITIES

Several research topics remain not enough explored before Gaussian-process models will become mature technology ready to use in the control-engineering practice. The relative immaturity for engineering practice is evident from a small number of real-life control applications, i.e., (Likar and Kocijan, 2007; Ko et al., 2007) and (Deisenroth and Rasmussen, 2009b), reported in literature until present.

Research opportunities can be roughly divided as follows: firstly to issues concerning dynamic systems modelling with Gaussian process models related to the control design, secondly issues concerning the control design itself and thirdly some general issues related to the control design and applications. The given list of issues is subjective and heavily based on the available information about ongoing research activities throughout the world.

Modelling trends and issues. The computational burden that increases with increasing number of the data contained in the model, caused mainly by calculation of the inverse covariance matrix, directs researchers to find more efficient methods for the inverse covariance matrix calculation or the input data selection. The issue of automatically polishing data and finding informative portions is reported as one of key issues in dynamic systems identification in general (Ljung, 2008) and remains one of current challenges also in the GP modelling research.

The issue of recursive model identification is the issue that is closely linked to adaptive control methods. In the machine learning community this kind of identification is known as on-line learning, which is not limited only to sequentially in-streaming data. The efficient method for recursive identification of the GP model still remains an unanswered challenge.

Further, methods for developing GP state-space models (Turner et al., 2010) still offer a lot of unsolved research problems.

Control design trends and issues. Most of the reported control methods have not addressed disturbance rejection, which is crucial in control systems, but have been more focused on the set-point tracking. The complete assessment of control methods requires also the disturbance rejection analysis, which in many cases still remain an unexplored issue.

The current research on control methods deals with adaptive and predictive control. First results that have a potential for control applications is also modelling of switching systems, e.g., (Saatçi et al., 2010).

If control methods are meant to be used in engineering practice more results on robust control design methodologies are necessary. Gaussian process models offer a lot of potential for the robust control design and offer a lot of research and application opportunities.

Some general issues that need to be brought-up are benchmarking of control methods with purpose to assess different control methods properly and fairly. A giant step to bring research results closer to the engineering practice is the integration of knowledge and software upgrade from pieces scattered around, mainly on the internet, into the user-friendly integrated software.

Research opportunities lay also in developing methods and procedures for various kinds analysis of system models and closed-loop systems. Model-simulations stability and closed-loop stability are only two, very different, but important and challenging problems among many to be addressed.

5. CONCLUSIONS

The control design faces the challenge of more and more complex systems to be controlled. On the other hand the development of control technology has given the ability to design control for increasingly uncertain systems, especially away from typical engineering fields, e.g., biosystems. The control based on Gaussian process models is an approach that may be promising for treating such systems (Ažman and Kocijan, 2007).

This paper provides the state-of-the-art survey of control algorithms for dynamic systems described in publications where Gaussian-process models have been used for the control design or as the part of controller. These methods, ranging from direct inverse control to advanced forms of the adaptive control, take or not into account the variance information provided by the Gaussian-process model.

The survey touched also possible trends, challenges and research opportunities. It was shown that a number of challenges and research opportunities that will bring the method to the level of maturity appropriate for engineering practice still lie ahead.

REFERENCES

Ažman, K. and Kocijan, J. (2008). Non-linear model predictive control for models with local information and uncertainties. *Transactions of the Institute of Measurement and Control*, 30(5).

Ažman, K. and Kocijan, J. (2006). An application of Gaussian process models for control design. In *International Control Conference ICC 2006*, 4 pages. Glasgow.

Ažman, K. and Kocijan, J. (2007). Application of Gaussian processes for black-box modelling of biosystems. *ISA Transactions*, 46, 443–457.

Ažman, K. and Kocijan, J. (2009). Fixed-structure Gaussian process model. *International Journal of Systems Science*, 40(12), 1253–1262.

Bemporad, A., Morari, M., Dua, V., and Pistikopoulos, E.N. (2000). The explicit solution of model predictive conrol via multiparametric quadratic programming. In *Proceedings of the American Control Conference*, 872–876. Chicago, Illinois.

Deisenroth, M.P. (2010). *Efficient Reinforcement Learning using Gaussian Processes*. Ph.D. thesis, Karlsruhe Institute of Technology, Karlsruhe.

Deisenroth, M.P., Rasmussen, C.E., and Peters, J. (2009). Gaussian process dynamic programming. *Neurocomputing*, 72(7–9), 1508–1524.

Deisenroth, M., Peters, J., and Rasmussen, C. (2008a). Approximate dynamic programming with Gaussian processes. In *Proceedings of American Control Conference (ACC)*, 4480–4485. Seattle, WA.

Deisenroth, M. and Rasmussen, C. (2009a). Bayesian Inference for Efficient Learning in Control. In *Proceedings of Multidisciplinary Symposium on Reinforcement Learning (MSRL)*. Montreal, Canada.

Deisenroth, M. and Rasmussen, C. (2009b). Efficient reinforcement learning for motor control. In *Proceedings of the 10th International PhD Workshop on Systems and Control, a Young Generation Viewpoint*. Hluboka nad Vltavou, Czech Republic.

Deisenroth, M., Rasmussen, C., and Peters, J. (2008b). Model-based reinforcement learning with continuous states and actions. In *Proceedings of the European Symposium on Artificial Neural Networks (ESANN)*, 19–24. Bruges, Belgium.

Engel, Y., Szabo, P., and Volkinshtein, D. (2006). Learning to Control an Octopus Arm with Gaussian Process Temporal Difference Methods. In Y. Weiss, B. Schoelkopf, and J. Platt (eds.), *Advances in Neural Information Processing Systems*, volume 18, 347–354. MIT Press.

Filatov, N. and Unbehauen, H. (2000). Survey of adaptive dual control methods. *IEE Proceedings - Control theory and applications*, 147(1), 119–128.

Goodwin, G.C., Rojas, O., and Takata, H. (2001). Nonlinear control via generalized feedback linearization using neural networks. *Asian Journal of Control*, 3(2), 79–88.

Grancharova, A., Johansen, T.A., and Toendel, P. (2007a). Computational aspects of approximate explicit nonlinear model predictive control. In R. Findeisen, F. Allgoewer, and L.T. Biegler (eds.), *Assessment and future directions of nonlinear model predictive control*, volume 358 of *Lecture notes in control and information sciences*, 181–190. Springer, Berlin, Heidelberg.

Grancharova, A. and Kocijan, J. (2007). Stochastic predictive control of a thermoelectric power plant. In *Proceedings of the International Conference Automatics and Informatics07*, I–13–I–16. Sofia.

Grancharova, A., Kocijan, J., and Johansen, T.A. (2007b). Explicit stochastic nonlinear predictive control based on Gaussian process models. In *Proceedings of European control conference (ECC)*, 2340–2347. Kos, Greece.

Grancharova, A., Kocijan, J., and Johansen, T.A. (2008). Explicit stochastic predictive control of combustion plants based on Gaussian process models. *Automatica*, 44(4), 1621–1631.

Gregorčič, G. and Lightbody, G. (2002). Gaussian processes for internal model control. In A. Rakar (ed.), *Proceedings of 3rd International PhD Workshop on Advances in Supervision and Control Systems, A Young Generation Viewpoint*, 39–46. Strunjan, Slovenia.

Gregorčič, G. and Lightbody, G. (2003a). From multiple model networks to the Gaussian processes prior model. In *Proceedings of IFAC ICONS conference*, 149–154. Faro.

Gregorčič, G. and Lightbody, G. (2003b). Internal model control based on Gaussian process prior model. In *Proceedings of the 2003 American Control Conference, ACC 2003*, 4981–4986. Denver, CO.

Gregorčič, G. and Lightbody, G. (2005). Gaussian process approaches to nonlinear modelling for control. In *Intelligent Control Systems using Computational Intelligence Techniques*, volume 70 of *IEE Control Series*, 177–217. IEE, London.

Isermann, R., Lachman, K.H., and Matko, D. (1992). *Adaptive Control Systems*. Systems and Control Engineering. Prentice Hall International.

Johansen, T.A. (2004). Approximate explicit receding horizon control of constrained nonlinear systems. *Automatica*, 40, 293–300.

Khalil, H.K. (2002). *Nonlinear systems*. Prentice-Hall, Upper Saddle River, NJ, 3rd edition.

Ko, J., Klein, D.J., Fox, D., and Haehnel, D. (2007). Gaussian processes and reinforcement learning for identification and control of an autonomous blimp. In *Proceedings of the International Conference on Robotics and Automation*, 742–747. Rome.

Kocijan, J. and Murray-Smith, R. (2005). Nonlinear predictive control with Gaussian process model. In R. Murray-Smith and R. Shorten (eds.), *Switching and Learning in Feedback Systems*, volume 3355 of *Lecture Notes in Computer Science*, 185–200. Springer.

Kocijan, J., Murray-Smith, R., Rasmussen, C., and Girard, A. (2004). Gaussian process model based predictive control. In *Proceedings of 4th American Control Conference (ACC 2004)*, 2214–2218. Boston, MA.

Kocijan, J. and Leith, D.J. (2004). Derivative observations used in predictive control. In *Proceedings of IEEE Melecon conference*, volume 1, 379–382. Dubrovnik.

Kocijan, J., Murray-Smith, R., Rasmussen, C.E., and Likar, B. (2003). Predictive control with Gaussian process models. In *Proceedings of IEEE Region 8 EUROCON 2003: computer as a tool*, volume A, 352–356. Ljubljana.

Leith, D.J. and Leithead, W.E. (1998). Appropriate realisation of MIMO gain-scheduled controllers. *International Journal of Control*, 70, 13–50.

Leith, D.J. and Leithead, W.E. (2000). Survey of gain-scheduling analysis and design. *International Journal*

of Control, 73, 1001–1025.

Likar, B. and Kocijan, J. (2007). Predictive control of a gas-liquid separation plant based on a Gaussian process model. *Computers and chemical engineering*, 31(3), 142–152.

Ljung, L. (2008). Perspectives on system identification. In *Proceedings of IFAC 17th World Congress 2008*, 6 pages. Seoul.

Maciejowski, J.M. (2002). *Predictive control with constraints*. Pearson Education Limited, Harlow.

Murray-Smith, R., Sbarbaro, D., Rasmussen, C., and Girard, A. (2003). Adaptive, cautious, predictive control with Gaussian process priors. In *Proceedings of 13th IFAC Symposium on System Identification*. Rotterdam, Netherlands.

Murray-Smith, R. and Sbarbaro, D. (2002). Nonlinear adaptive control using nonparametric Gaussian process prior models. In *Proceedings of IFAC 15th World Congress*. Barcelona.

Nguyen-Tuong, D. and Peters, J. (2008). Learning robot dynamics for computed torque control using local Gaussian processes regression. In *Symposium on Learning and Adaptive Behaviors for Robotic Systems*, 59–64.

Nguyen-Tuong, D., Peters, J., Seeger, M., and Schoelkopf, B. (2008a). Learning inverse dynamics: a comparison. In *Proceedings of the European Symposium on Artificial Neural Networks (ESANN)*, 13–18. Bruges, Belgium.

Nguyen-Tuong, D., Seeger, M., and Peters, J. (2010). *Real-time local GP model learning*, volume 264, chapter From Motor Learning to Interaction Learning in Robots, 193–207. Springer.

Nguyen-Tuong, D., Seeger, M., and Peters, J. (2008b). Computed torque control with nonparametric regression models. In *Proceedings of the 2008 American Control Conference, ACC 2008*, 6 pages. Seattle, Washington.

Norgaard, M., Ravn, O., Poulsen, N.K., and Hansen, L.K. (2000). *Neural Networks for Modelling and Control of Dynamic Systems, A practitioner's Handbook*. Advanced Textbooks in Control and Signal Processing. Springer.

Palm, R. (2007). Multiple-step-ahead prediction in control systems with Gaussian process models and TS-fuzzy models. *Engineering Applications of Artificial Intelligence*, 20(8), 1023–1035.

Petelin, D. and Kocijan, J. (2011). Control system with evolving Gaussian process model. In *Proceedings of IEEE Symposium Series on Computational Intelligence, SSCI 2011*. IEEE, Paris.

Rasmussen, C.E. and Deisenroth, M.P. (2008). Probabilistic inference for fast learning in control. In *Recent Advances in Reinforcement Learning, Lecture Notes on Computer Science*, volume 5323, 229–242. Springer.

Rasmussen, C.E. and Kuss, M. (2004). Gaussian processes in reinforcement learning. In S. Thurn, L. Saul, and B. Schoelkopf (eds.), *Advances in Neural Information Processing Systems conference*, volume 16, 751–759. MIT Press.

Rasmussen, C.E. and Williams, C.K.I. (2006). *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA.

Rugh, W.J. (1991). Analytic framework for gain-scheduling. *IEEE Control Systems magazine*, 11, 79–84.

Rugh, W.J. and Shamma, J.S. (2000). Research on gain scheduling. *Automatica*, 36, 1401–1425.

Saatçi, Y., Turner, R., and Rasmussen, C.E. (2010). Gaussian process change point models. In *Proceedings of the 27th Annual International Conference on Machine Learning*, 927–934.

Sbarbaro, D. and Murray-smith, R. (2005). An adaptive nonparametric controller for a class of nonminimum phase non-linear system. In *Proceedings of IFAC 16th World Congress*. Prague, Czech Republic.

Sbarbaro, D., Murray-Smith, R., and Valdes, A. (2004). Multivariable generalized minimum variance control based on artificial neural networks and Gaussian process models. In *International Symposium on Neural Networks*. Springer.

Sbarbaro, D. and Murray-Smith, R. (2005). Self-tuning control of nonlinear systems using Gaussian process prior models. In R. Murray-Smith and R. Shorten (eds.), *Switching and Learning in Feedback Systems*, volume 3355 of *Lecture Notes in Computer Science*, 140–157. Springer, Heidelberg.

Slotine, J.J.E. and Li, W. (1991). *Applied nonlinear control*. Prentice Hall, Englewood Cliffs, NJ.

Turner, R., Deisenroth, M., and Rasmussen, C. (2010). State-space inference and learning with Gaussian processes. In *Proceedings of 13th International Conference on Artificial Intelligence and Statistics*, volume 9, 868–875. Sardinia, Italy.

Wittenmark, B. (2002). Adaptive dual control. In *Control Systems, Robotics and Automation, Encyclopedia of Life Support Systems (EOLSS), Developed under the auspices of the UNESCO*. Eolss Publishers, Oxford, UK.