INCORPORATING LINEAR LOCAL MODELS IN GAUSSIAN PROCESS MODEL

Juš Kocijan *,** **Agathe Girard** ***

* *Jozef Stefan Institute, Ljubljana*
** *Nova Gorica Polytechnic, Nova Gorica*
*** *Department of Computing Science, University of Glasgow, Glasgow*

Abstract: Identification of nonlinear dynamic systems from experimental data can be difficult when, as often happens, more data are available around equilibrium points and only sparse data are available far from those points. The probabilistic Gaussian Process model has already proved to model such systems efficiently. The purpose of this paper is to show how one can relatively easily combine measured data and linear local models in this model. Also, using previous results, we show how uncertainty can be propagated through such models when predicting ahead in time in an iterative manner. The approach is illustrated with a simple numerical example. *Copyright © 2005 IFAC*

Keywords: Systems identification, Gaussian process models, nonlinear systems

1. INTRODUCTION

One of the problems frequently met in practice when modelling dynamic systems is the difficulty of constructing a nonlinear model on a reliable and consistent basis from available data. In this paper, we focus on experimental rather than first principles modelling. Owing to operating and safety constraints, the available measured data from which we are required to construct an empirical model is often concentrated mainly around equilibrium points, with only relatively sparse data measured far from equilibrium. A common approach in this situation is to build local models using the data in vicinity of equilibrium points and then blend these models so as to obtain a nonlinear model covering the operating envelope, refer to e.g. (Murray-Smith et al., 1999).

The purpose of this paper is to show how linear local models can be incorporated in Gaussian processes (GPs) models of dynamic systems. The use of Gaussian processes for modelling dynamic systems has recently been studied, e.g. (Kocijan et al., 2003a; Girard et al., 2002; Gregorčič and Lightbody, 2003). A key issue when modelling with this probabilistic model is that, in its simplest form, the computational burden associated with it is cubic in the number of data points used, as it requires the inversion of an $N \times N$ matrix with dimension, where N is the number of data points. Although this computational burden can be reduced by employing approximate inverses, we suggest an alternative approach that summarizes measured data in the vicinity of an equilibrium point by a derivative observation, i.e. a local linear model. Therefore, this approach is not only in accord with engineering practice but it can also directly reduce the computational burden. Also, following (Girard et al., 2002), we show how one can propagate the uncertainty ahead in time,

[1] This work was made possible by EC funded Research Training Network HPRN-CT-1999-00107 and Enterprise Ireland International Collaboration Grant IC/2003/25

when predicting multiple steps ahead with such models.

The paper is organized as follows. Gaussian process models are briefly reviewed and the incorporation of derivative observations is then discussed. The modelling of dynamic systems with such models is described in Section 3. An example in Section 4 illustrates the modelling and forecasting of a simulated dynamic system. Conclusions are summarized at the end of the paper.

2. GAUSSIAN PROCESS MODEL

A detailed presentation of Gaussian processes can be found in (O'Hagan, 1978; Williams, 1998). A Gaussian process is a random function fully characterized by its mean and covariance functions. For simplicity, we assume a zero-mean process. Given $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, the corresponding $f(\mathbf{x}_1), \dots, f(\mathbf{x}_n)$ can be viewed as a collection of random variables which have a joint multivariate Gaussian distribution: $f(\mathbf{x}_1), \dots, f(\mathbf{x}_n) \sim \mathcal{N}(0, \mathbf{\Sigma})$, where Σ_{pq} gives the covariance between $f(\mathbf{x}_p)$ and $f(\mathbf{x}_q)$ and is a function of the corresponding $\mathbf{x}_p$ and $\mathbf{x}_q$: $\Sigma_{pq} = C(\mathbf{x}_p, \mathbf{x}_q)$. The covariance function $C(.,.)$ can be of any kind, provided that it generates a positive definite covariance matrix Σ. The Gaussian Process model fits naturally in the Bayesian modelling framework, as it places a prior directly over functions, instead of parameterizing $f(\mathbf{x})$. In the following, we assume a stationary process, where the stationarity assumption implies that the covariance between two points depends only on the distance between them and is invariant by translation in the input space. A common choice of covariance function is the squared exponential or Gaussian one:

$$\begin{aligned}\mathrm{Cov}[f(\mathbf{x}_p)\ , f(\mathbf{x}_q)] &= C(\mathbf{x}_p, \mathbf{x}_q) = \\ &= v_1 \exp\left[-\frac{1}{2}\sum_{d=1}^{D} w_d (x_p^d - x_q^d)^2\right] \ ,\end{aligned} \tag{1}$$

where x_p^d denotes the d^{th} component of the D-dimensional input vector $\mathbf{x}_p$, and $v_1, w_1, \dots, w_D$ are free parameters. This covariance function is such that points close together in the input space lead to more correlated outputs than points further apart (a smoothness assumption). The parameter v_1 controls the vertical scale of variation and the w_d's are inversely proportional to the horizontal length-scale in dimension d ($\lambda_d = 1/\sqrt{w_d}$).

Let the input/target relationship be $y = f(\mathbf{x}) + \epsilon$. We assume an additive white noise with variance v_0, $\epsilon \sim \mathcal{N}(0, v_0)$, and put a GP prior on $f(.)$, with covariance function (1) and unknown parameters. Within this probabilistic framework, we can write $y_1, \dots, y_n \sim \mathcal{N}(0, \mathbf{K}_n)$, with $K_{npq} = \Sigma_{pq} + v_0\delta_{pq}$, where $\delta_{pq} = 1$ if $p = q$ and 0 otherwise. If we split $y_1, \dots, y_n$ into two parts, $\mathbf{y} = [y_1, \dots, y_N]$ and y^*, we can write

$$\mathbf{y}, y^* \sim \mathcal{N}(0, \mathbf{K}_n) \tag{2}$$

with

$$\mathbf{K}_n = \begin{bmatrix} \begin{bmatrix} & & \\ & \mathbf{K} & \\ & & \end{bmatrix} & \begin{bmatrix} \\ \mathbf{k}(\mathbf{x}^*) \\ \\ \end{bmatrix} \\ \\ \begin{bmatrix} \mathbf{k}(\mathbf{x}^*)^T \end{bmatrix} & \begin{bmatrix} \kappa(\mathbf{x}^*) \end{bmatrix} \end{bmatrix} \ , \tag{3}$$

where $\mathbf{K}$ is an $N \times N$ matrix giving the covariances between y_p and y_q, for $p, q = 1 \dots N$, $\mathbf{k}(\mathbf{x}^*)$ is an $N \times 1$ vector giving the covariances between y^* and y_p ($k_p(\mathbf{x}^*) = C(\mathbf{x}_p, \mathbf{x}^*)$, for $p = 1 \dots N$), and $\kappa(\mathbf{x}^*) = C(\mathbf{x}^*, \mathbf{x}^*)$ is the covariance between the test output and itself.

For our modelling purposes, we can then divide this joint probability into a marginal and a conditional part. Given a set of N training data pairs, $\{\mathbf{x}_p, y_p\}_{p=1}^N$, the marginal term gives us the likelihood of the observed data: $\mathbf{y}|\mathbf{X} \sim \mathcal{N}(0, \mathbf{K})$, where $\mathbf{y}$ is the $N \times 1$ vector of training targets and $\mathbf{X}$ the $N \times D$ matrix of the corresponding training inputs. We can then estimate the unknown parameters of the covariance function, as well as the noise variance v_0, via maximization of the log-likelihood. The conditional part of (2) provides us with the predictive distribution of y^* corresponding to a new given input $\mathbf{x}^*$. We only need to condition the joint distribution on the training data and the new input $\mathbf{x}^*$, $p(y^*|\mathbf{y}, \mathbf{X}, \mathbf{x}^*) = \frac{p(\mathbf{y}, y^*)}{p(\mathbf{y}|\mathbf{X})}$. It can be shown that this distribution is Gaussian with mean and variance

$$\begin{aligned} \mu(\mathbf{x}^*) &= \mathbf{k}(\mathbf{x}^*)^T\, \mathbf{K}^{-1}\, \mathbf{y} & (4) \\ \sigma^2(\mathbf{x}^*) &= \kappa(\mathbf{x}^*) \ -\ \mathbf{k}(\mathbf{x}^*)^T\, \mathbf{K}^{-1}\, \mathbf{k}(\mathbf{x}^*) + v_0 \ . & (5) \end{aligned}$$

This way, we can use the predictive mean $\mu(\mathbf{x}^*)$ as an estimate for y^* and the predictive variance, or standard deviation $\sigma(\mathbf{x}^*)$, as the uncertainty attached to it.

2.1 Incorporating derivative observations

The Gaussian process modelling framework is readily extended to include situations where derivatives of the function are observed, as well as (or instead of) the values of the function itself. More on this topic can be found in (Leith et al., 2002; Solak et al., 2002; Kocijan et al., 2003d). Since differentiation is a linear operation, the derivative of a GP remains a GP. Assuming a zero-mean GP for $y = f(\mathbf{x})$, with Gaussian covariance function, the mean and covariance functions of the derivative process (in a given dimension) are readily obtained. The output (target) vector $\mathbf{y}$, which before consisted solely of output measurements, now also contains derivative observations. Similarly, the corresponding inputs are the values of the regressor associated with each function and derivative observation, and the covariance matrix is changed accordingly. In the case of the Gaussian covariance function (1) using relation $y_p = f(\mathbf{x}_p)$, the covariance between two functional observations is

$$\mathrm{Cov}[y_p, y_q] = v_1 \exp\left[-\frac{1}{2}\sum_{d=1}^{D} w_d (x_p^d - x_p^d)^2\right] ,$$

that between two different derivative observations is

$$\begin{aligned}\mathrm{Cov}[\frac{\partial y_p}{\partial x_p^d}, \frac{\partial y_q}{\partial x_q^e}] = {} & v_1 w_e (\delta_{d,e} - w_d (x_p^d - x_q^d) \\ & (x_p^e - x_q^e) \\ & \exp\left[-\frac{1}{2}\sum_{d=1}^{D} w_d (x_p^d - x_q^d)^2\right] ,\end{aligned} \tag{6}$$

where $\frac{\partial y_p}{\partial x_p^d}$ denotes the first derivative of y_p in direction of d^{th} component of the D-dimensional input vector $\mathbf{x}_p$.

The covariance between a derivative and functional observation is

$$\begin{aligned}\mathrm{Cov}[\frac{\partial y_p}{\partial x_p^d}, y_q] = {} & -v_1 w_d (x_p^d - x_q^d) \\ & \exp\left[-\frac{1}{2}\sum_{d=1}^{D} w_d (x_p^d - x_q^d)^2\right] .\end{aligned} \tag{7}$$

The GP model acts to integrate and smooth the noisy derivative observations. Derivative observations around an equilibrium point can be interpreted as observations of a local linear model about this equilibrium point. This means that the derivative observations can be synthesized using standard linear regression. Such synthetic derivative observations can then be used to summarize training points in the vicinity of equilibrium points, thereby effectively reducing the number of data points in the model for computational purposes. It is important to note that a local linear input-output model such as a transfer function model only specifies a derivative observation up to a co-ordinate transformation. For simplicity, in this paper, we use lagged input signal samples and lagged output signal samples as our state coordinates, although other choices are possible.

The data may or may not contain information about the noise. For function observations, assuming a white noise, the noise information is added to the diagonal elements of the covariance matrix, corresponding to these points. If no information is available, the noise variance v_0 is learned as in Section 2. For the derivative observations, noise information for each local model is also obtained when standard identification methods are used. The covariance matrices of each linear local model obtained at identification are added to the overall covariance matrix for the corresponding derivative component (see (Solak et al., 2002)).

Given that derivative and function observations are available, the predictive distribution of a function output corresponding to a new $\mathbf{x}$ has mean and variance given by equations (4) and (5), with the matrix $\mathbf{K}$ and the vector $\mathbf{k}^*$ changed adequately. In fact, these moments can be written so as to reflect the mixed nature of the training data (see (Kocijan et al., 2003d) for details).

3. GAUSSIAN PROCESS MODELLING OF DYNAMIC SYSTEMS

The above modelling procedure can be readily applied to dynamic systems, within an autoregressive (AR) representation of the system (Girard et al., 2002; Kocijan et al., 2003a).

Consider the following ARX model, where the current output depends on delayed outputs and exogeneous control inputs:

$$\begin{aligned}y(k) = {} & f(y(k-1), y(k-2), \ldots, y(k-L), \\ & u(k-1), u(k-2), \ldots, u(k-L)) + \epsilon ,\end{aligned} \tag{8}$$

where ϵ is a white noise and k denotes consecutive number of data sample. Let $\mathbf{x}(k)$ be the state vector at k, composed of the previous outputs y and inputs u, up to a given lag L ($\mathbf{x}(k) = [y(k-1), y(k-2), \ldots, y(k-L), u(k-1), u(k-2), \ldots, u(k-L)]^T$) and $y(k)$ the corresponding output. We can then model this dynamic system using a Gaussian Process.

3.1 Multiple-step-ahead predictions

Assuming the time-series is known up to, say, k, we wish to predict n steps ahead: That is to say, to find the predictive distribution of $y(k+n)$ corresponding to $\mathbf{x}(k+n) = [y(k+n-1), \ldots, y(k+n-L)]^T$. Multiple-step-ahead predictions of a system modelled by (8) can be achieved by iteratively making repeated one-step-ahead predictions, up to the desired horizon.

A naive way of doing so is, at each time-step, to feed back the predictive mean (estimate of the output): by considering $\mathbf{x}(k +$

$n) = [\hat{y}(k+n-1), \dots, \hat{y}(k+n-L)]^T$, where $\hat{y}(k+n-i)$ is the point estimate of $y(k+n-i)$. Although this approach is approximate (as the variance of the lagged outputs on the right-hand side of equation (8) is neglected), it has been used when modelling dynamic systems with neural networks or fuzzy models. However, it has been shown to lead to unrealistically small variances for the multiple-step-ahead predictions (Girard et al., 2002).

In (Girard et al., 2002), iterative multiple-step-ahead prediction is done by feeding back the predictive mean as well as the predictive variance at each time-step, thus taking the uncertainty attached to each intermediate prediction into account. This means that the input at which we wish to predict becomes a normally distributed random variable. The illustration of such dynamical model simulation is given in Figure 1 and we are now going to show how this approach can be applied to a GP with derivative observations.

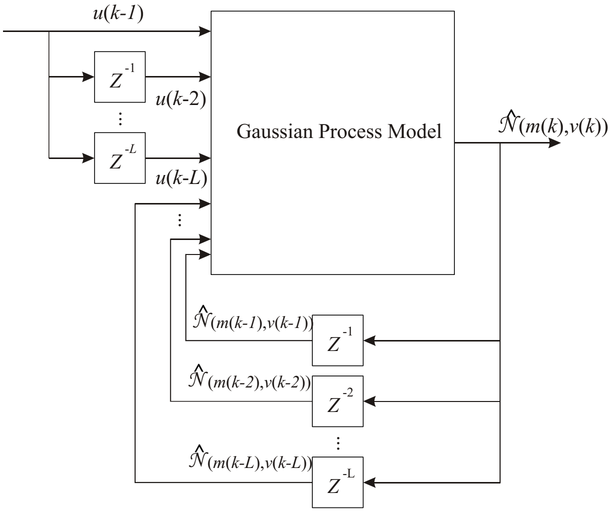

Fig. 1. Block scheme of dynamical system simulation with iterative method where variance is propagated through the system

3.2 Accounting for derivative observations

We consider the case where derivative observations are used in combination with function observations. If we wish to make multiple-step-ahead predictions with such data, and also account for the uncertainty induced by each successive prediction as we predict ahead in time, we have to account for both sorts of data when computing the predictive mean and variance of the output.

In case of function observations only, the predictive mean and variance of the output corresponding to a noisy input $\mathbf{x}$ are obtained by solving ((Girard et al., 2002))

$$m(\boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\Sigma}_x) = E_{\mathbf{x}}[\mu(\mathbf{x})] \tag{9}$$

$$\begin{aligned} v(\boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\Sigma}_x) &= E_{\mathbf{x}}[\sigma^2(\mathbf{x})] + E_{\mathbf{x}}[\mu(\mathbf{x})^2] \\ &\quad - m(\boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\Sigma}_x)^2 \ . \end{aligned} \tag{10}$$

Similarly, we can show that when derivative observations are also present, we need to compute

$$m(\boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\Sigma}_x) = E_{\mathbf{x}}[\mu_1(\mathbf{x})] + E_{\mathbf{x}}[\mu_d(\mathbf{x})] \ , \tag{11}$$

for the mean, and

$$\begin{aligned} v(\boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\Sigma}_x) &= v + E_{\mathbf{x}}[\sigma_1^2(\mathbf{x})] + 2E_{\mathbf{x}}[\sigma_{1d}^2(\mathbf{x})] \\ &\quad + E_{\mathbf{x}}[\sigma_d^2(\mathbf{x})] + E_{\mathbf{x}}[\mu_1(\mathbf{x})^2] \\ &\quad + 2E_{\mathbf{x}}[\mu_{1d}(\mathbf{x})^2] + E_{\mathbf{x}}[\mu_d(\mathbf{x})^2] \\ &\quad - m(\boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\Sigma}_x)^2, \end{aligned} \tag{12}$$

for the variance, where $\mu_1(\mathbf{x})$ and $\sigma_1^2(\mathbf{x})$ correspond to the predictive mean and variance when only function observations are available, $\mu_d(\mathbf{x})$ and $\sigma_d^2(\mathbf{x})$ the predictive mean and variance corresponding to derivative observations in direction d, and $\sigma_{1d}^2(\mathbf{x})$ is the component reflecting the mixed nature of the data. Due to space restriction, we refer to (Kocijan et al., 2003d) for the detailed derivations and the final expressions.

We can now apply these results to the multiple-step-ahead prediction task of a dynamic system, when derivative observations are available. For simplicity, we consider the following AR model

$$\begin{aligned} y(k) &= f(\mathbf{x}(k)) + \epsilon \quad \text{with} \\ \mathbf{x}(k) &= [y(k-1), \dots, y(k-L)]^T \end{aligned} \tag{13}$$

where ϵ is a white noise with variance v_0. Note that when predicting ahead in time, and propagating the uncertainty, the exogenous input u are assumed to be known and consequently do not need to be dealt with.

As done in ((Girard et al., 2002)) in the case of function observations only, we can predict n-steps ahead and propagate the uncertainty of the successive predictions by considering each $y(k+n-i)$ as a Gaussian random variable, and therefore, an $L \times 1$ random state $\mathbf{x}(k+n) = [y(k+n-1), \dots, y(k+n-L)]^T \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\Sigma}_x)$ at each time-step, with mean

$$\boldsymbol{\mu}_{\mathbf{x}} = \begin{bmatrix} m(\mathbf{x}(k+n-1)) \\ \vdots \\ m(\mathbf{x}(k+n-L) \end{bmatrix} \tag{14}$$

and covariance matrix

$$\mathbf{\Sigma}_x = \begin{bmatrix} v(\mathbf{x}(k+n-1))+v_0 & \cdots & \mathrm{Cov}(y(k+n-L),y(k+n-1)) \\ \vdots & \vdots & \vdots \\ \mathrm{Cov}(y(k+n-1),y(k+n-L)) & \cdots & v(\mathbf{x}(k+n-L))+v_0 \end{bmatrix}\Bigg) , \tag{15}$$

where $m(.)$ and $v(.)$ are computed using equations (11) and (12) and this is how derivative observations are taken into account (see (Kocijan et al., 2003d) for details). It is the computation of $m(.)$ and $v(.)$ that is different in the case of using derivative observations in comparison with using functional observations only.

In general, at time sample $k+l$, we have the random input vector $x(k+l) = [y(k+l-1), \ldots, y(k+l-L)]^T \sim \mathcal{N}(\boldsymbol{\mu}_\mathbf{x}, \mathbf{\Sigma}_x)$ with mean $\boldsymbol{\mu}_\mathbf{x}$ formed by the predictive mean of the lagged outputs $y(k+l-\tau)$, $\tau = 1, \ldots, L$, given by (4), or by (11), depending on l, and the diagonal elements of the $L \times L$ input covariance matrix $\mathbf{\Sigma}_x$ contain the corresponding predictive variances. The cross-covariance terms $\mathrm{Cov}[y(k+l-i), y(k+l-j)]$, for $i,j = 1 \ldots L$ with $i \neq j$, are obtained by computing $\mathrm{Cov}[y(k+l), \mathbf{x}(k+l)]$, disregarding the last (*oldest*) element of $\mathbf{x}(k+l)$. We have

$$\mathrm{Cov}[y(k+l), \mathbf{x}(k+l)) = E[y(k+l)\mathbf{x}(k+l)] \\ -E[y(k+l)]E[\mathbf{x}(k+l)] \ . \tag{16}$$

Again, we refer to (Girard et al., 2002) and (Kocijan et al., 2003d) for more details.

4. EXAMPLE

The following example is intended to explore the potential of achieving an accurate model of a dynamic system, when using derivative observations at equilibrium points and a small number of function observations at off-equilibrium points.

Consider the nonlinear dynamic system described by

$$y(k) = 0.5y(k-1) + \tanh(y(k-1) + u^3(k-1)) \ , \tag{17}$$

where the sampling time is 0.5 seconds. We select ten equilibrium points, uniformly spanning the operating region of interest. At each equilibrium point, we apply a small-scale pseudo-random binary signal with mean 0 and magnitude 0.03 and the corresponding output signal is contaminated with normally distributed measurement noise in the range [-0.001,0.001]. A linear, first order, approximation to the local dynamics at the equilibrium point is identified using the Matlab algorithm IV4. In addition to this equilibrium information, a small sparse set of off-equilibrium input-output data, consisting of only 6 points, is selected (larger numbers of off-equilibrium observations were also studied but 6 represents a good compromise between predictive accuracy and number of data points used). A GP model with zero-mean and Gaussian covariance function is then trained using

- 10 input-output values at equilibrium, spanning the operating region of interest;
- The set of coefficients of the identified first order linear models representing the partial derivatives of the output - 10 times 2 coefficients ;
- The 6 input-output values that were sampled out of equilibrium points.

Figure 2 shows a plot of the output predictive variance. A region with low variance indicates a region where the model is confident about its prediction.

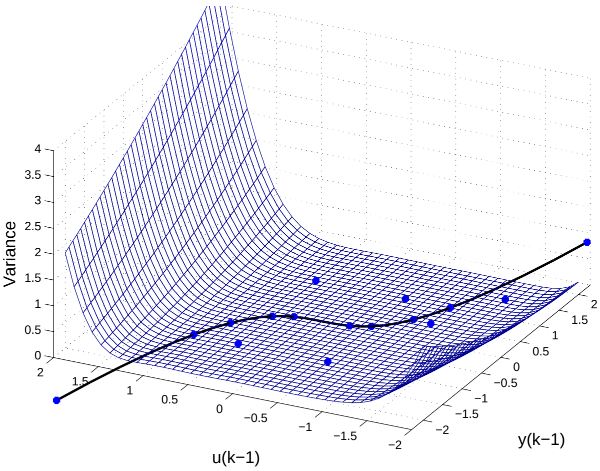

Fig. 2. Variance of the GP model on the same figure together with equilibrium locus represented with solid line and training points represented with dots

The results of the simulation of the system (that is the n-step-ahead prediction, where n is the length of the validation signal) is shown in Figure 3. It can be seen that propagating the uncertainty causes the standard deviations to become larger in some areas, compared to the naive approach. Also, the means are affected.

5. CONCLUSIONS

This paper describes how linear local models can be incorporated in the Gaussian Process model. Also, we show how one can propagate the uncer-

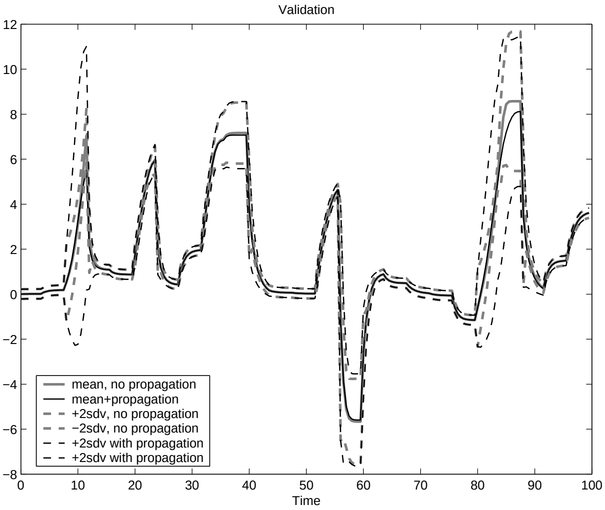

Fig. 3. Response on validation data: GP model response without propagation of uncertainty - grey lines, GP model response with propagation of uncertainty - black lines

tainty when making iterative multiple-step-ahead predictions with such a model.

Accounting for derivative observations, obtained as coefficients of local linear models in equilibrium points with regular linear regression method, means joining local linear models and GP models. Whereas local model networks have problems retaining local information when optimized to fit the process globally, GP models prove difficult when a lot of data is used for identification. Joining these two approaches results in global models containing global and local information, of acceptable dimensions and suited to the kind of data usually available in practice when carrying out experimental modelling (a lot of data in vicinity of equilibrium points and few data far from equilibria).

The main conclusions are as follows:

- The data used to obtain the grey-box model is well suited to the kind of data usually available in practice when carrying out experimental modelling.
- The model obtained is relatively small in comparison with a GP model that does not make use of derivative observations, while the model quality is comparable. This makes it very suitable for applications.

Our simulated example is encouraging and these results offer new possibilities for dynamic system analysis and control, whenever uncertainty information is necessary.

REFERENCES

Girard A., C.E. Rasmussen, R. Murray-Smith (2002). Multi-step ahead prediction for non linear dynamic sytems - A Gaussian Process treatment with propagation of the uncertainty, In:*Advances in Neural Information processing Systems*, (S. Becker and S. Thrun and K. Obermayer, (Eds.)), Vol. 15, pp. 545-552, MIT Press, Cambridge, MA.

Gregorčič G. and G. Lightbody (2003). Internal model control based on a Gaussian process prior model, In: *Proceedings of ACC'2003*, Denver, pp. 4981-4986.

Kocijan J., A. Girard, B. Banko and R. Murray-Smith (2003a). Dynamic Systems Identification with Gaussian Processes, In: *Proceedings of 4th Mathmod*, Vienna, 776-784, expanded version accepted for publication in journal *Mathematical and Computer Modelling of Dynamic Systems.*

Kocijan, J. A. Girard and D. J. Leith (2003b). Incorporating linear local models in Gaussian process models, IJS report DP-8895, Jozef Stefan Institute, Ljubljana.

Leith, D. J., W. E. Leithead, E. Solak and R. Murray-Smith (2002). Divide & conquer identification: Using Gaussian process priors to combine derivative and non-derivative observations in a consistent manner, In: *Conference on Decision and Control 2002*, Las Vegas, pp. 624-629.

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

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

Solak, E., R. Murray-Smith, W. E. Leithead, D. J. Leith and C. E. Rasmussen (2002). Derivative observations in Gaussian Process models of dynamic systems, In: *Advances in Neural Information processing Systems*, (S. Becker and S. Thrun and K. Obermayer, (Eds.)), Vol. 15, pp. 529-536, MIT Press, Cambridge, MA.

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