Internal Model Control Based on a Gaussian Process Prior Model

Gregor Gregorčič and Gordon Lightbody
Department of Electrical Engineering, University College Cork, Ireland
gregorg@rennes.ucc.ie

Abstract

To improve transparency and reduce the curse of dimensionality of non–linear black–box models, the local modelling approach was proposed. Poor transient response of Local Model Networks led to the use of non–parametrical probabilistic models such as the Gaussian Process prior approach. Recently, Gaussian Process models were applied for Minimum Variance Control. This paper introduces the use of the Gaussian Process model for non–linear Internal Model control. The invertibility of the Gaussian Process model is discussed and the use of predicted variance is illustrated on a simulated example.

1 Introduction

In the past years, many approaches to the modelling of non–linear systems using Neural Network [1] and fuzzy models [2] have been proposed. The difficulties associated with black–box modelling approaches are mainly related to the curse of dimensionality and lack of transparency of the global model. The local modelling approach [3] has been proposed to increase transparency as well as reduce the curse of dimensionality. Difficulties related to partitioning of the operating space, structure determination, local model identification and off–equilibrium dynamics are the main drawbacks of the local modelling techniques. To improve the off-equilibrium behavior, the use of non–parametrical probabilistic models, such as Gaussian Processes priors was proposed for dynamic non–linear modelling in [4]. The ability to make a robust estimation in the transient region, where only a limited number of data points is available, is the main advantage of the Gaussian Process in comparison to Local Models Network. The Gaussian Process prior approach was introduced in [5] and revised in [6]. Recently it was adopted by the Neural Network and machine learning community. Gaussian Process models provide prediction as well as variance of the predicted output. This variance can be interpreted as a level of confidence of the prediction, which is a main advantage of this approach in comparison to Neural Network or fuzzy models. Gaussian Processes have been introduced to control applications in [7], where the controller scheduling variable was estimated from the Input/Output measured data. In [8], Minimum Variance Control using the Gaussian Process model has been proposed.

This paper introduces the use of Gaussian Process models for non–linear Internal Model Control. Internal Model Control is one of the common used model–based techniques for the control of the non–linear systems. In this strategy the controller is chosen to be an inverse of the model. Many different approaches to Internal Model Control have been proposed. The main difference in approaches is in the choice of type of internal model and its inverse. An analytical inverse of the non–linear model, based on the physical understanding of the system, was proposed in [9]. Also an analytical inverse of the local model network was shown in [10] and a numerical inverse, based on the Neural Network was utilised in [11].

The invertibility of the Gaussian Process model is discussed in this paper. Use of the predicted variance is proposed to extend the non–linear Internal Model Control structure. A simulated example was used to illustrate the benefits of the proposed technique.

2 Gaussian Processes

Any finite set of random variables, which have a joint Gaussian distribution is a Gaussian Process. Consider some noisy input/output set of data $\mathcal{D}$. The full matrix Φ_N of N d–dimensional input vectors is constructed as follows:

$$\Phi_N = \begin{bmatrix} u_1(1) & u_2(1) & \cdots & u_d(1) \\ u_1(2) & u_2(2) & \cdots & u_d(2) \\ \vdots & \vdots & \vdots & \vdots \\ u_1(k) & u_2(k) & \cdots & u_d(k) \\ \vdots & \vdots & \vdots & \vdots \\ u_1(N) & u_2(N) & \cdots & u_d(N) \end{bmatrix} \quad (1)$$

Scalar outputs are arranged in the output vector $\underline{y}_N$:

$$\underline{y}_N = [y(1), y(2), \ldots, y(k), \ldots, y(N)]^T \quad (2)$$

The aim is to construct the model, and then at some new input vector:

$$\underline{u}^T(N+1) = [u_1(N+1), u_2(N+1), \ldots, u_d(N+1)] \notin \mathcal{D} \quad (3)$$

find the distribution of the corresponding output $y(N+1)$. A general model for the set of data can be written as:

$$y(k) = y_{nf}(\underline{u}(k)) + \eta(k) \quad (4)$$

where $y(k)$ is a noisy output, y_{nf} is the modelling function which produces noise free output from the input vector $\underline{u}(k)$ and $\eta(k)$ is additive noise. The prior over the space of possible functions to model the data can be defined as $P(y_{nf}|\underline{\alpha})$, where $\underline{\alpha}$ is some set of hyperparameters. Also a prior over the noise $P\left(\underline{\eta}\middle|\underline{\beta}\right)$ can be defined, where $\underline{\eta}$ is the vector

0-7803-7896-2/03/$17.00 ©2003 IEEE

Proceedings of the American Control Conference
Denver, Colorado June 4-6, 2003

of noise values $\underline{\eta} = [\eta(1), \eta(2), \ldots, \eta(k), \ldots, \eta(N)]$ and $\underline{\beta}$ is the set of hyperparameters. The probability of the data given hyperparameters $\underline{\alpha}$ and $\underline{\beta}$ can be written as:

$$P\left(\underline{y}_N \middle| \Phi_N, \underline{\alpha}, \underline{\beta}\right) = \int P\left(\underline{y}_N \middle| \Phi_N, y_{nf}, \underline{\eta}\right) P(y_{nf}|\underline{\alpha}) P\left(\underline{\eta}\middle|\underline{\beta}\right) dy_{nf} d\underline{\eta} \quad (5)$$

Define the matrix: $\Phi_{N+1} = \left[\Phi_N^T, \underline{u}(N+1)\right]^T$ and vector: $\underline{y}_{N+1} = \left[\underline{y}_N^T, y(N+1)\right]^T$. The conditional distribution of $\underline{y}_{N+1}$ can then be written as:

$$P\left(\underline{y}_{N+1} \middle| \mathcal{D}, \underline{\alpha}, \underline{\beta}, \Phi_{N+1}\right) = \frac{P\left(\underline{y}_{N+1} \middle| \underline{\alpha}, \underline{\beta}, \Phi_{N+1}\right)}{P\left(\underline{y}_N \middle| \underline{\alpha}, \underline{\beta}, \Phi_N\right)} \quad (6)$$

This conditional distribution can be used to make a prediction about $y(N+1)$. The integral in the equation (5) is complicated. Some of the standard approaches to solving such problems are reviewed in [12]. Assuming that additive noise is Gaussian, the approach based on Gaussian Process priors gives an exact analytic form of equation (5), and facilitates matrix manipulations.

The Gaussian process is fully represented by its mean and covariance function $C(\cdot)$ which produces the covariance matrix $\mathbf{C}$. In this paper, a zero–mean distribution is assumed:

$$\underline{y}_N = [y(1), y(2), \ldots, y(k), \ldots, y(N)]^T \sim \mathcal{N}(0, \mathbf{C}) \quad (7)$$

Obviously not all data can be modelled as a zero–mean process. If the data is properly scaled and detrended, then this assumption is correct. A prediction at point $y(N+1)$ can be made using the conditional distribution [12], which for a Gaussian process is also Gaussian [6].

$$\begin{aligned} &\hat{y}(N+1) \sim \mathcal{N}\left(\mu_{\hat{y}(N+1)}, \sigma^2_{\hat{y}(N+1)}\right), \text{ where :} \\ &\mu_{\hat{y}(N+1)} = \underline{v}_{N+1}^T \mathbf{C}_N^{-1} \underline{y}_N \\ &\sigma^2_{\hat{y}(N+1)} = \nu - \underline{v}_{N+1}^T \mathbf{C}_N^{-1} \underline{v}_{N+1} \end{aligned} \quad (8)$$

$\mathbf{C}_N$ of size $N \times N$, $\underline{v}_{N+1}$, of size $1 \times N$ are the covariance matrices and ν is a scalar, constructed as follows:

$$\begin{aligned} \mathbf{C}_N &= \begin{bmatrix} C_{1,1} & \cdots & C_{1,N} \\ \vdots & C_{m,n} & \vdots \\ C_{N,1} & \cdots & C_{N,N} \end{bmatrix} \\ C_{m,n} &= C(\underline{u}(m), \underline{u}(n)) \\ \underline{v}_{N+1} &= [C(\underline{u}(1), \underline{u}(N+1)), \ldots, C(\underline{u}(N), \underline{u}(N+1))]^T \\ \nu &= C(\underline{u}(N+1), \underline{u}(N+1)) \end{aligned} \quad (9)$$

$C(\cdot)$ is the covariance function of the inputs only. Any choice of the covariance function, which will generate a non–negative definite covariance matrix for any set of input points, can be chosen [13], which offers the ability to include the prior knowledge in the model. The covariance function (10) used in this paper was derived from a fixed Radial Basis Functions Network, where the centers of the basis function were placed on each data point.

$$C(\underline{u}(m), \underline{u}(n)) = w_0 e^{-\frac{1}{2}\sum_{l=1}^{d} w_l (\underline{u}_l(m) - \underline{u}_l(n))^2} + w_\eta \delta(m,n) \quad (10)$$

$\underline{\theta} = [w_0, w_1, \ldots, w_d, w_\eta]^T$ is the vector of hyperparameters and d is the dimension of the input space. The effect of the hyperparameters on the covariance function is discussed in section 2.2. To be able to make a prediction, using equation (8), the hyperparameters have to be provided either as prior knowledge, or trained from the training data. One of the possible techniques is maximizing of the likelihood of the training data by the given covariance function [12, 13].

2.1 Linear Regression vs. Gaussian Process Models

Consider a single input, single output system, represented by equation (4). Let the function $y_{nf}(k) = Ku(k)$, where K is an unknown gain. For each pair of data $\{u(k), y(k)\}$ for $k = 1 \ldots N$, the parameter K can be calculated. Since $y(k)$ data is corrupted with noise, every calculation gives a different value value of K. Parameter $\hat{K}$ can be estimated using the Least–Squares algorithm. This approach is equivalent to fitting the best line through the set of data $\mathcal{D}$. The prediction at point $u(N+1)$ can be written as:

$$\hat{y}(N+1) = \hat{K} \cdot u(N+1) \quad (11)$$

A different approach can be taken for the prediction of $y(N+1)$ at the point $u(N+1)$ using the same set of data. Assuming that outputs $y(k)$ have a joint normal (Gaussian) distribution[1], the Gaussian process prior was placed directly on the space of functions (in this case on the space of slopes), that can model the data and noise.

The difference between equation (11) and the equation (8) is obvious. Given $\hat{K}$ at the new input $u(N+1)$ equation (11) gives prediction $\hat{y}(N+1)$. However the prediction based on Gaussian processes is more powerful. Given the $\mathbf{C}_N$, $\underline{v}_{N+1}$ and ν (which are functions of all inputs and hyperparameters) and the vector of noisy outputs $\underline{y}_N$, the prediction at the new input $u(N+1)$ has a Gaussian distribution. That means that at the point $u(N+1)$ the output $y(N+1)$ is most likely going to be the mean prediction $\mu_{\hat{y}(N+1)}$. The level of confidence for this prediction is given by the variance $\sigma^2_{\hat{y}(N+1)}$ about the mean. The variance can be seen as "error bars" for this prediction (Fig. 1b).

2.2 Introduction to Gaussian Process Modelling of Dynamic Processes

Consider a first order discrete dynamic system described by the non–linear function:

$$y(k+1) = 0.95 \cdot \tanh(y(k)) + \sin(u(k)) + \eta(k) \quad (12)$$

where $u(k)$ and $y(k)$ are the input and the output of the system, $y(k+1)$ is the one–step ahead predicted output and $\eta(k)$ is white noise. To model the system using a Gaussian

[1] The set of outputs $y(1), \ldots, y(N)$ can be seen as the set of random variables.

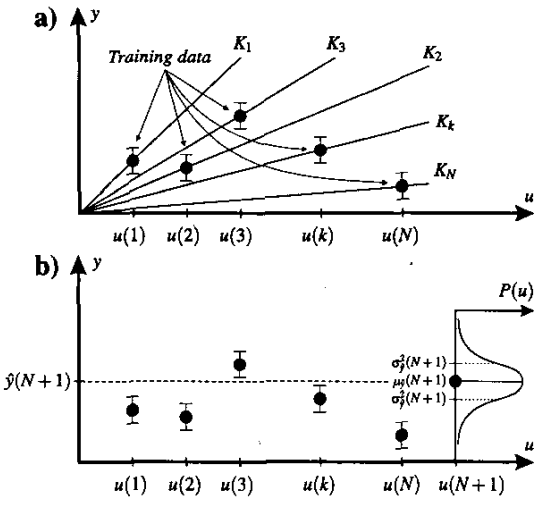

Figure 1: a) Training set of data, **b)** Mean $\mu_{\hat{y}(N+1)}$ of the prediction distribution is the predicted value of $\hat{y}(N+1)$ at $u(N+1)$. Variance $\sigma^2_{\hat{y}(N+1)}$ gives error bars $\hat{y}(N+1) \pm \sigma^2_{\hat{y}(N+1)}$ on prediction.

Process model, the dimension of the input space should be selected. The problem is similar to the structure selection of an ARX model [14]. Since the system presented in equation (12) has order one, the input space is going to be two dimensional. One present input and output are required to make a one–step ahead prediction of the first order system. The selection of the input space is discussed in [15].

The system was simulated to generate representative data. A Gaussian process model was trained as a one–step ahead prediction model. The training data points were arranged as follows:

$$\Phi_N = \begin{bmatrix} u(1) & u(2) & \cdots & u(k) & \cdots & u(N) \\ y(1) & y(2) & \cdots & y(k) & \cdots & y(N) \end{bmatrix}^T \qquad (13)$$
$$\underline{y}_N = [y(2), y(3), \ldots, y(k+1), \ldots, y(N)]^T$$

The noise variance was 0.01 and the sampling time was 0.1 sec. The training data is shown in Fig. 2. The Maximum Likelihood framework was used to determine the hyperparameters. A conjugate gradient optimisation algorithm with random restarts was used to search for their optimal values. The following set of hyperparameters was found: $\underline{\theta} = [w_1, w_2, w_0, w_\eta]^T = [0.1971, 0.1796, 1.6931, 0.0104]^T$.

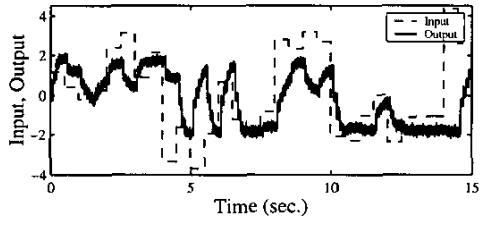

Figure 2: Training data set.

Hyperparameters w_1 and w_2 allow a weight for each input dimension. In this case w_1 and w_2 are close together, which means that input $u(k)$ and $y(k)$ have equal weight in the prediction. The hyperparameter w_0 gives the overall scale of the local correlation. w_η is the estimated variance of the noise. It was assumed, that the added noise $\eta(k)$ was white. If noise is correlated then the covariance function (10) can be modified as shown in [16]. A more detailed description of the role of the hyperparameters can be found in [12] and [13].

This Gaussian process model was then tested on a validation data set. Fig. 3 shows the input $u(k)$, true noise free output and the prediction $\hat{y}(k+1)$. In the range of the operating space, where the model was trained (white region), the predicted output fits well to the true output. As soon as the input moves away from the well modelled region, the prediction does not fit to the true output. In the time interval between 8 and 11 seconds, the system was driven well into an untrained region of the operating space. The shaded band around the the prediction presents the confidence level of the the predicted mean. The wider confidence bar in this region indicate that the model is less certain about its prediction.

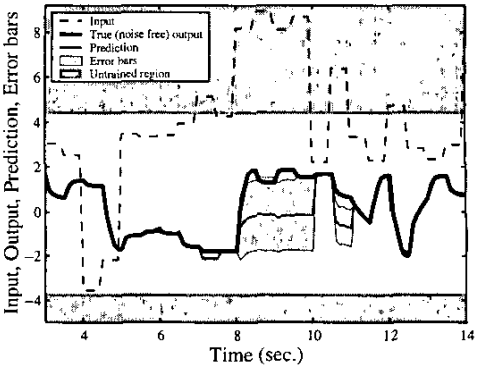

Figure 3: Validation of the model. In the range away from operating space spanned by the training data, the prediction does not fit well to true noise free output. The less certain prediction in this region is confirmed by the wider error bars.

The noise free version of the system of equation (12) can be presented as a 3–D surface $y(k+1) = f(u(k), y(k))$ as shown in Fig. 4a. Fig. 4b shows the approximation of the surface using the Gaussian Process model.

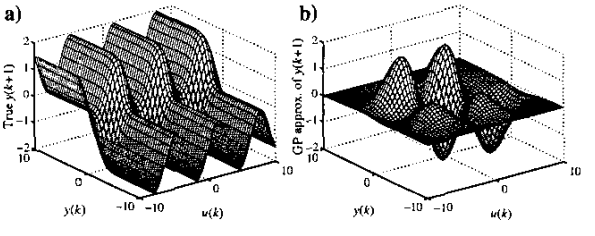

Figure 4: a) True process surface $y(k+1) = f(u(k), y(k))$, **b)** Gaussian Process approximation of the process surface.

Contour and gradient plots of the true process function and its Gaussian Processes model can be seen on Fig. 5. The dots represent the training set of data. It can be seen on the magnified portion of the plot, that the model represents the true function well, on the region where the model has been trained. Away from that region however the model is no longer a good representation of the true function.

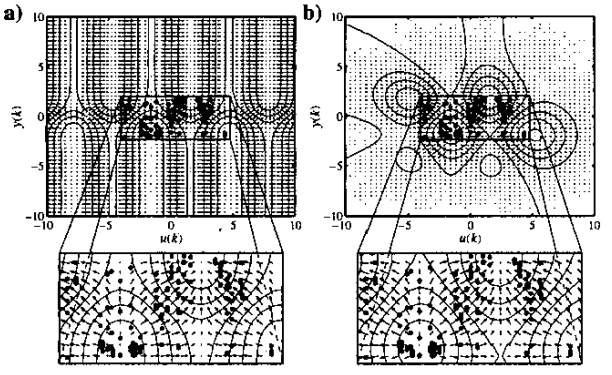

Figure 5: **a)** Contour and gradient plot of the true process function, **b)** Gaussian Process approximation of the process function. The magnified portion of the plot shows the close–up of the operating space where the model was trained.

The Gaussian Process approach for modelling of non–linear systems from data is straightforward, with the main drawback being the computation cost. Since the prediction of the model output requires inversion of the covariance matrix $\mathbf{C}_N$, size $N \times N$, the use of the Gaussian processes model might became problematic with large amounts of data.

3 Gaussian Process Model for the Internal Model Control

When the model is available, the Internal Model Control strategy is one of the widely used approaches for control of the non–linear systems. A general representation is shown in Fig. 6. It was shown, that with the perfect model, stability of closed–loop system is assured, if the plant and the controller are stable. If the controller is chosen as the inverse of the model, perfect control can be achieved [9]. The filter in the loop defines desired closed–loop behavior of the closed–loop system and balances robustness to the model mismatch.

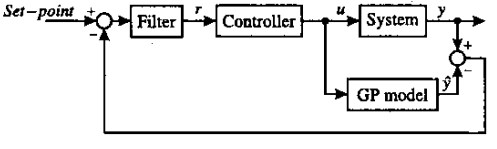

Figure 6: General IMC structure.

3.1 Inverse of the Gaussian Process Model

To investigate invertibility of the Gaussian Process model, consider a linear model of the form,

$$\begin{aligned} y(k+1) &= bu(k) + ay(k) + \eta(k) \\ &= \begin{bmatrix} u(k) & y(k) \end{bmatrix} \begin{bmatrix} b \\ a \end{bmatrix} + \eta(k) \\ &= \underline{\phi}(k)^T \underline{\Theta} + \eta(k), \end{aligned} \tag{14}$$

where added noise is assumed to be white zero–mean Gaussian of variance σ_η^2. The system can be modelled using the linear covariance function[2]:

$$\begin{aligned} C_{m,n} &= E\left\{ f(u(m), y(m)) f(u(n), y(n)) \right\} \\ &= b^2 u(m)u(n) + a^2 y(m)y(n) + \sigma_\eta^2 \\ &= \underline{\phi}^T(m) \mathbf{C}_\Theta \underline{\phi}(n) + \sigma_\eta^2 \end{aligned} \tag{15}$$

where:

$$\mathbf{C}_\Theta = \begin{bmatrix} b^2 & 0 \\ 0 & a^2 \end{bmatrix} \tag{16}$$

The covariance matrix $\mathbf{C}_N$ and vector $\underline{\nu}_{N+1}^T$ can be written as follows:

$$\begin{aligned} \mathbf{C}_N &= \Phi_N^T \mathbf{C}_\Theta \Phi_N + \mathbf{I}\sigma_\eta^2 \\ \underline{\nu}_{N+1}^T &= \underline{\phi}(\kappa)^T \mathbf{C}_\Theta \Phi_N \end{aligned} \tag{17}$$

The prediction equation (8) can be written as:

$$\begin{aligned} \mu_{\hat{y}(N+1)} &= \underline{\phi}(\kappa)^T \mathbf{C}_\Theta \Phi_N \mathbf{C}_N^{-1} \underline{y}_N \\ &= \underline{\phi}(\kappa)^T \begin{bmatrix} \chi_1 \\ \chi_2 \end{bmatrix} = \underline{\phi}(\kappa)^T \underline{\chi} \end{aligned} \tag{18}$$

where $\underline{\phi}(\kappa)^T$ is new point, consisting Input/Output sample. $\underline{\chi}$ is a constant vector, which is a function of the training data. Given the predicted and present output sample, the inverse can be written as:

$$u(\kappa+1) = \frac{1}{\chi_1} \left(\mu_{\hat{y}(N+1)} - \chi_2 y(\kappa) \right) \tag{19}$$

It is obvious that the existence of the analytical inverse depends of the form of the covariance function, which is directly related to the structure of the modelled system. In the example above the model is affine in u for any y. For the non–linear systems of form:

$$y(k+1) = bu(k) + f(\underline{x}(k)) + \eta(k); \tag{20}$$

where $\underline{x}(k)$ is the state vector. It was shown in [8], that the covariance function can be decomposed to obtain an affine structure. As shown in equation (19), the inverse is then straightforward.

When the structure of the modelled non–linear system is not known, the system is then usually modelled using the covariance function (10). This covariance function results in a model, which is not analytically invertible. Instead of

[2] It is assumed in following derivation, that parameters a and b are not random variables $E\{a^2\} = a^2, E\{b^2\} = b^2$ and the noise variance σ_η^2 is known.

calculating the exact inverse, a numerical approach such as successive approximation or Newton–Raphson [11], can be used to find the control effort to solve the following equation:

$$f(u(k),y(k)) - r(k) = 0; \quad \text{where } f(u(k),y(k)) = \hat{y}(k+1) \tag{21}$$

3.2 Unconstrained Internal Model Control

As an example, consider the first order non–linear system:

$$y(t) = \frac{K(t)}{1+p\tau(t)}u(t); p \triangleq \frac{d}{dt}; \begin{cases} K(t) = f(u(t),y(t)) \\ \tau(t) = f(u(t),y(t)) \end{cases} \tag{22}$$

The gain and time constant of the system change with the operating point. Fig. 7 shows the open–loop system response at different operating points. Simulation was used

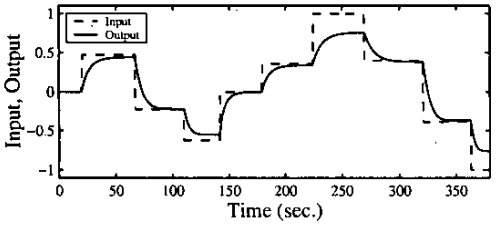

Figure 7: Gain and time constant are non–linear functions of the operating point.

to generate 582 Input/Output training data points spanning the operating space. The sampling time was 0.5 seconds and the noise variance was set to 10^{-4}. The Gaussian Process model was trained as a one–step ahead prediction model. Since the Internal Model Control strategy requires a parallel model, the trained one–step ahead prediction model was then utilised as seen in Fig. 8. This Gaussian Process model

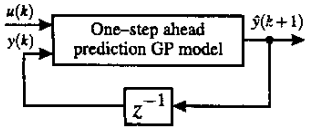

Figure 8: Realisation of the parallel Gaussian Process model.

was then included in the Internal Model Control structure and the numerical inverse of the equation (21) was found every sample time.

As seen in Fig. 9, the Internal Model Control works well when the control input and 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, equation (21) does not have a solution which means that control input u in the left side of the equation (21) can not be found to force the right side of equation (21) to zero. The value which drives equation (21) closest to zero is applied instead. This incorrect control input might drive the output of the model even further away from the well modelled operating space. This behavior which can drive the closed–loop system to the point of instability, can be seen in Fig. 9. In the region after 60 seconds, where the model output does not follow the system output. Variance of the prediction, in this case, increases significantly.

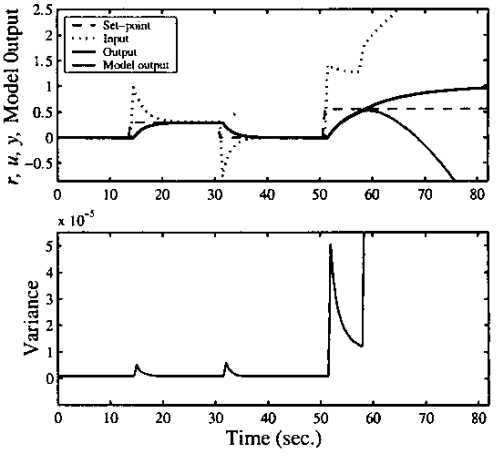

Figure 9: Closed–loop response using Internal Model Control strategy. Desired closed–loop behavior was chosen as first order response, with unity gain and time constant of 2 seconds.

3.3 Variance–constrained Inverse of Gaussian Process Model

Since poor closed–loop performance is a result of the the model being driven outside its trained region, which is governed by control input, the naive approach would be to constrain the control input. It could happen however that the operating space is not fully represented within the training data set. When the system is driven in this untrained portion of the operating space, the model will no longer represent the system well and the initial constraints of the control signal might not be relevant. However, the increase of the predicted variance will indicate, that prediction is not confident any more. This increase of variance, can be used as a constraint in the optimisation algorithm utilised to solve equation (21). The concept is shown in Fig. 10. The aim is to optimise the control effort, so that the variance does not increase above its predefined limit. A variance constraint should be defined by the designer and in general it can be a function of some scheduling variable. In this paper, the variance constraint is set to be a constant value.

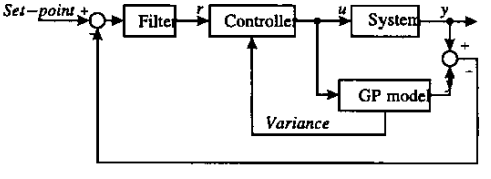

Figure 10: Variance–constrained IMC structure.

Constrained nonlinear programming with random restarts was used to optimise the control effort. In this example the variance was constrained to 4×10^{-6}. As seen in Fig. 11, the closed loop–response has been improved. Ob-

viously, instead of constraining the instantaneous variance[3], the optimisation over the predicted horizon would give the smoother response. To achieve this goal, the propagation of the uncertainty for the Gaussian Process model has to be utilised. This is one of the main research topics in the machine learning community [17]. Uncertainty propagation will offer the ability to improve and extend the ideas presented in this paper to Model Predictive Control [18] and Minimum Variance Control based on Gaussian Process models.

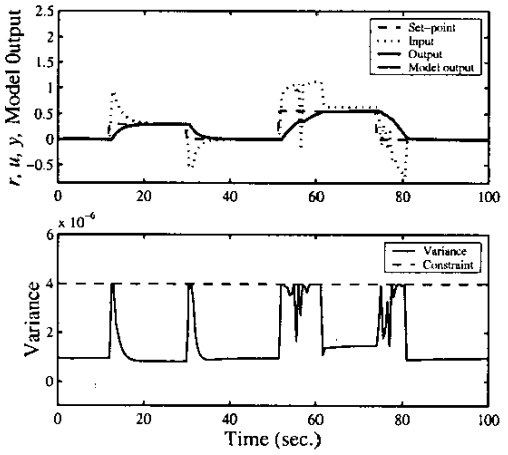

Figure 11: Improved closed–loop response using variance–constrained Internal Model Control.

4 Conclusions

The formulation of the Gaussian process model for the modelling of non–linear systems from data was explained. The modelling procedure is elegant and simple. The user has to specify the dimension of the input space and process the data through the Gaussian Processes machinery. Since a relatively small amount of data is required and a parsimonious set of hyperparameters have to be trained, the Gaussian Process prior approach has proved to be a robust regression technique. It was shown on a simulated example, that information about the predicted variance can be seen as a level of confidence in the prediction. This information can then be used to detect that the operating point has strayed from the well modelled region. The main disadvantage of the Gaussian Processes modelling approach is its high computation cost.

A novel approach for the use of the prediction variance in Internal Model Control, based in the Gaussian Process prior model was proposed in this paper. The control effort was optimised, so that the variance does not increase above a predefined limit.

The invertibility of a Gaussian Process model was discussed. It was shown, that an analytic inverse of the model depends of the choice of covariance function. When the most common covariance function for non–linear system modelling is applied, the analytical inverse can not be found and a numerical inverse must be used instead. Part of the future work will focus on investigation of the form of covariance function that will produce an analytically invertible model. If achieved, this will dramatically reduce the computation cost.

Concepts presented in this paper will be extended to Model Predictive Control and Minimum Variance Control.

[3] Variance in this case can be seen as one–step ahead predicted variance, where the present information is not feed back to the input of the model and does not effect the variance prediction in the next step.

References

[1] K. S. Narendra and K. Parthasarathy. Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, 1(1):4–27, Mar. 1990.

[2] T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man and Cybernetics*, SMC–15(1):116–132, Jan.–Feb. 1985.

[3] R. Murray-Smith and T. A. Johansen (Eds.). *Multiple Model Approaches to Modelling and Control*. Taylor and Francis, London, 1997.

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

[5] A. O'Hagan. Curve fitting and optimal design for prediction (with discussion). *Journal of the Royal Statistical Society B*, 40(1):1–42, 1978.

[6] C.K.I. Williams. Prediction with gaussian processes: From linear regression to linear prediction and beyond. *Learning and Inference in Graphical Models*, 1998.

[7] D.J. Leith, W.E. Leithead, and R. Murray-Smith. Nonlinear structure identification: A gaussian process/velocity–based approach. 2001.

[8] R. Murray-Smith and D. Sbarbaro. Nonlinear adaptive control using nonparametric gaussian process models. In *International Federation of Automatic Control, 15th IFAC Triennial World Congress*, 2002.

[9] C. G. Economou, M. Morari, and B. O. Palsson. Internal model control: Extension to non–linear systems. *Ind. Eng. Chem. Process Des. Dev.*, 25:403–411, 1986.

[10] M. D. Brown, G. Lightbody, and G. W. Irwin. Nonlinear internal model control using local model networks. *IEE Proceedings: Control Theory and Applications*, 144(6):505–514, Nov. 1997.

[11] E. P. Nahas, M. A. Henson, and D.E. Seborg. Non–linear internal model control strategy for neural network models. *Comp. Chem. Eng.*, 16:1039–1057, 1992.

[12] M. N. Gibbs. *Bayesian Gaussian Processes for Regresion and Classification*. PhD thesis, Univercity of Cambridge, 1997.

[13] C. K. I. Williams and C. E. Rasmussen. Gaussian processes for regression. In M. E. Hasselmo Touretzky, M. C. Mozer, editor, *Advances in Neural Information Processing Systems 8*, pages 514–520. MIT Press, 1996.

[14] L. Ljung. *System Identification: Theory for the User*. Prentice Hall PTR, 2nd edition, Jan. 1999.

[15] G. Gregorčič and G. Lightbody. Gaussian processes for modelling of dynamic non–linear systems. In *Proceedings of the Irish Signals and Systems Conference, Cork*, pages 141–147, Jun. 2002.

[16] R. Murray-Smith and A. Girard. Gaussian process priors with arma noise models. In *Proceedings of The Irish Signals and Systems Conference, Maynooth*, pages 147–152, Jun. 2001.

[17] A. Girard, C. E. Rasmussen, and R. Murray-Smith. Multiple–step ahead predictions of time series models with gaussian processes by propagating uncertainty. In *Proceedings of The International Conference on Current Advances and Trends in Nonparametric Statistics*, Jun. 2002.

[18] J. Kocijan. Gaussian process model based predictive control. Technical Report DP–8710, Jožef Stefan Institute, Dec. 2002.