NONLINEAR ADAPTIVE CONTROL USING NONPARAMETRIC GAUSSIAN PROCESS PRIOR MODELS

Roderick Murray-Smith * **Daniel Sbarbaro** **

* *Department of Computing Science, University of Glasgow, Glasgow*
G12 8QQ, Scotland, UK. & Hamilton Institute, NUI Maynooth, Ireland
E-mail: rod@dcs.gla.ac.uk
** *Departamento de Ingenieŕia Eléctrica, Universidad de Concepción,*
Chile. E-mail: dsbarbar@die.udec.cl

Abstract: Nonparametric Gaussian Process prior models, taken from Bayesian statistics methodology are used to implement a nonlinear adaptive control law. The expected value of a quadratic cost function is minimised, without ignoring the variance of the model predictions. This leads to implicit regularisation of the control signal (caution), and excitation of the system. The controller has dual features, since it is both tracking a reference signal and learning a model of the system from observed responses. The general method and its main features are illustrated on a simulation example. Copyright ©2002 IFAC.

Keywords: Gaussian process priors, nonparametric models, dual control, nonlinear model-based predictive control.

1. INTRODUCTION

Many authors have proposed the use of non-linear models as a base to build nonlinear adaptive controllers. In many applications, however, these nonlinearities are not known, and non-linear parameterision must be used instead. A popular choice has been the use of artificial neural networks for estimating the nonlinearities of the system (Narendra and Parthasarathy, 1990; Liang and ElMargahy, 1994; Chen and Khalil, 1995; Bittanti and Piroddi, 1997). All these works have adopted for controller design, the *certainty equivalence principle*, where the model is used in the control law, as if it were the true system. In order to improve the performance of nonlinear adaptive controllers based on a nonlinear models, the accuracy of the model predictions should also be taken into account.

Most engineering applications are still based on parametric models, where the functional form is fully described by a finite number of parameters, often a linear function. Even in the cases where flexible parametric models are used, such as neural networks, spline-based models, multiple models etc, the uncertainty is usually expressed as uncertainty of parameters (even though the parameters often have no physical interpretation), and do not take into account uncertainty about model structure, or distance of current prediction point from training data used to estimate parameters. This paper describes an approach based on Gaussian process priors, as an example of a non-parametric model with useful analytic properties. These allow us to analytically obtain a control law which perfectly minimises the expected value of a quadratic cost function, which does not disregard the variance of the model prediction as an element to be minimised. This leads naturally, and automatically to a suitable combination of regularising *caution* in control behaviour in following the reference trajectory, and excitation of control effort, depending on model accuracy. The above ideas are closely related to the work done on 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 to parameters of models with fixed structure (Wittenmark, 1995; Filatov and Unbehauen, 2000).

2. NON-PARAMETRIC MODELS AND UNCERTAINTY PREDICTION

Parametric models commonly used in control contexts assume that the system's functional form can be described by a finite set of parameters. Non-parametric models (called 'smoothing' in some frameworks) do not constrain the model to a pre-specified functional form, and do not project the observed data down to a finite parameterisation. The weaker prior assumptions and flexible form typically applied in a non-parametric model make the approach well-suited for initial data analysis and exploration.

A significant advantage when modelling nonlinear dynamic systems is that nonparametric approaches retain the available data and perform inference conditional on the current state and local data. This is advantageous in off-equilibrium regions, since normally in these regions the amount of data available for identification is much smaller than that available close to equilibrium conditions. The uncertainty of model predictions can be made dependent on local data density, and the local model complexity automatically related to the amount of available data (more complex models need more evidence to make them likely). Both aspects are very useful when modelling sparsely-populated transient regimes in dynamic systems, and will have significant effects on control behaviour.

3. GAUSSIAN PROCESS PRIORS

An example of the use of a flexible non-parametric model with uncertainty predictions is a *Gaussian Process prior*, introduced in (O'Hagan, 1978) and recently reviewed in (Williams, 1998). Use of GPs in a control systems context is discussed in (Murray-Smith *et al.*, 1999; Leith *et al.*, 2000). A variation which can include ARMA noise models is described in (Murray-Smith and Girard, 2001).

In the following, the full matrix of state and control input vectors is denoted by $\mathbf{\Phi}$, and the vector of output points is $\mathbf{y}$. The discrete data of the regression model are $\phi_k = [\mathbf{x}(t), \mathbf{u}(t)]$ and $\mathbf{y}_k = y(t)$. In the example used in this paper, $\mathbf{x}(t) = [\tilde{y}(t-1), \ldots, \tilde{y}(t-n)]$, and $\mathbf{u}(t) = [u(t-1), \ldots, u(t-m)]$. The given N_1 data pairs used for identification are stacked in matrices $\mathbf{\Phi}_1, \mathbf{y}_1$ and the N_2 data pairs used for prediction are $\mathbf{\Phi}_2, \mathbf{y}_2$. Instead of parameterising $y(t) = f(\phi(t))$ as a parametric model, we can place a prior directly on the space of functions where f is assumed to belong. A Gaussian process represents the simplest form of prior over functions – we assume that any p points have a p-dimensional multivariate Normal distribution. We will assume zero mean, so for the case with partitioned data $\mathbf{y}_1$ and $\mathbf{y}_2$ we will have the multivariate Normal distribution (we will assume zero mean),

$$\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} \sim \mathcal{N}(0, \mathbf{\Sigma}), \quad \mathbf{\Sigma} = \begin{bmatrix} \mathbf{\Sigma}_1 & \mathbf{\Sigma}_{12} \\ \mathbf{\Sigma}_{21} & \mathbf{\Sigma}_2 \end{bmatrix}. \quad (1)$$

where $\mathbf{\Sigma}$ is the full covariance matrix for the training inputs $\mathbf{\Phi}_1$, and where $\Sigma_{21} = \Sigma_{12}^T$ for the cross-covariance between $\mathbf{\Phi}_1$ and $\mathbf{\Phi}_2$. Like the Gaussian distribution, the Gaussian Process is fully specified by a mean μ and its covariance function. The covariance function $C(\phi_i, \phi_j)$ expresses the expected covariance between y_i and y_j – we can therefore infer $\mathbf{y}_2$'s from constant given $\mathbf{\Phi}_1, \mathbf{y}_1$'s rather than building explicit parametric models.

As in the multinormal case, we can divide the joint probability into a marginal Gaussian process and a conditional Gaussian process. The marginal term gives us the likelihood of the training data,

$$P(\mathbf{y}_1|\mathbf{\Phi}_1) = (2\pi)^{-\frac{N_1}{2}} |\Sigma_1|^{-\frac{1}{2}} e^{\left(-\frac{1}{2}\mathbf{y}_1^T \Sigma_1^{-1} \mathbf{y}_1\right)}. \quad (2)$$

The conditional part of the model, which best relates to a traditional regression model is therefore the Gaussian process, which gives us the output posterior density function conditional on the training data $\mathbf{\Phi}_1, \mathbf{y}_1$ and the test points $\mathbf{\Phi}_2$,

$$\begin{aligned} P(\mathbf{y}_2|\mathbf{\Phi}_1, \mathbf{y}_1, \mathbf{\Phi}_2) &= \frac{P(\mathbf{y}_2, \mathbf{y}_1)}{P(\mathbf{y}_1)} \\ &= \frac{e^{-\frac{1}{2}(\mathbf{y}_2 - \mu_{21})^T \Sigma_{21}^{-1} (\mathbf{y}_2 - \mu_{21})}}{(2\pi)^{\frac{N_2}{2}} |\Sigma_{21}|^{\frac{1}{2}}}, \end{aligned}$$

where, as in the straightforward multinormal case described earlier,

$$\mu_{2.1} = \Sigma_{12}^T \Sigma_1^{-1} \mathbf{y}_1 \quad (3)$$

$$\Sigma_{2.1} = \Sigma_2 - \Sigma_{12}^T \Sigma_1^{-1} \Sigma_{21}, \quad (4)$$

so we can use $y(\mathbf{\Phi}_2) = \mu_{21}$ as the expected model output, with a variance of $\sigma(\mathbf{\Phi}_2) = \Sigma_{21}$.

3.1 *The covariance function*

The multivariate normal assumption may seem restrictive, but we can actually model a wide range of functions, depending on the choice of covariance function. The model's prior over possible functions can be adapted to a given application by altering the structure or parameters of the covariance function. The choice of function is only constrained in that it must always generate a non-negative definite covariance matrix for any inputs $\mathbf{\Phi}$, so we can represent a spectrum of systems from very nonlinear models, to standard linear models using the same framework. The covariance function will also often be viewed as being the combination of a covariance function due to the underlying model C_m and one due to measurement noise C_n. The entries of this matrix are then:

$$\Sigma_{1_{ij}} = C_m(\mathbf{\Phi}_{1_i}, \mathbf{\Phi}_{1_j}; \Theta) + C_n(\mathbf{\Phi}_{1_i}, \mathbf{\Phi}_{1_j}; \Theta) (5)$$

where $C_n()$ could be $\delta_{ij}\mathcal{N}(\mathbf{\Phi}_1; \Theta)$, which would be adding a noise model $\mathcal{N}$ to the diagonal entries of Σ_1.

We discuss covariance functions for correlated noise models in (Murray-Smith and Girard, 2001).

In this paper, we use a straightforward covariance function,

$$C(x_i, x_j; \Theta) = v_0 \rho(|x_i - x_j|, l) + \sum_{k=1}^{p} a_k x_{ik} x_{jk} + a_0,$$

so that the parameter vector $\Theta = [v_0, l_{1,..p}, a_0]^T$. $\rho(d, l)$ is a distance measure, which should be one at $d = 0$ and which should be a monotonically decreasing function of d. The one used here was

$$\rho(|x_i - x_j|, l) = e^{-\frac{1}{2} \sum_{k=1}^{p} l_k (x_{ik} - x_{jk})^2}. \quad (6)$$

The l_k's determine how quickly the function varies in dimension k. This estimates the relative smoothness of different input dimensions, and can therefore be viewed as an automatic relevance detection (ARD) tool, which helps weight the importance of different input dimensions. Other bases which include a nonlinear transformation of $\mathbf{x}$, like the RBF neural networks used in (Fabri and Kadirkamanathan, 1998), could be put into this framework. The parameter vector $\Theta = [v_0, l_{1,..p}, a_{0,...p}]^T$ can be adapted to maximise the likelihood using standard gradient-based optimisation algorithms. The prior associated with this covariance function states that outputs associated with ϕ's closer together should have higher covariance than points further apart.

3.2 *Nonlinear regression example*

As an illustrative example, we use a simple one-dimensional nonlinear function

$$y = \sin(10\pi x \sin(x - 0.5)^3) \quad (7)$$

corrupted with zero mean Gaussian noise, with standard deviation of 0.05. To show how well the system copes with small amounts of data only 30 points are used, which are sampled uniformly in $x \in [0, 1]$. In this range the function has significantly varying derivatives. An interesting aspect of the Gaussian process is its ability to produce good estimates of prediction variance. Figure 1 shows the mean and 2σ contours for y given x. Note how the 2σ contours change over the input-space depending on the proximity of training data.

4. DERIVATION OF CONTROL LAW

The objective of this paper is to control a multi-input, single-output, affine nonlinear system of the form,

$$y(t+1) = f(\mathbf{x}(t)) + g(\mathbf{x}(t))u(t) + e(t+1) \quad (8)$$

where $\mathbf{x}(t)$ is the state vector at time t, which in this paper will be defined as $\mathbf{x}(t) = [y(t), \ldots, y(t-n), u(t-1), \ldots, u(t-m)]$, $y(t+1)$ the output, $u(t)$ the current control vector, f and g are smooth nonlinear functions, and g is bounded away from zero. In addition, it is also assumed that the system is minimum phase, as defined in (Chen and Khalil, 1995). The noise term $e(t)$ is assumed zero mean Gaussian, but with unknown variance σ^2.

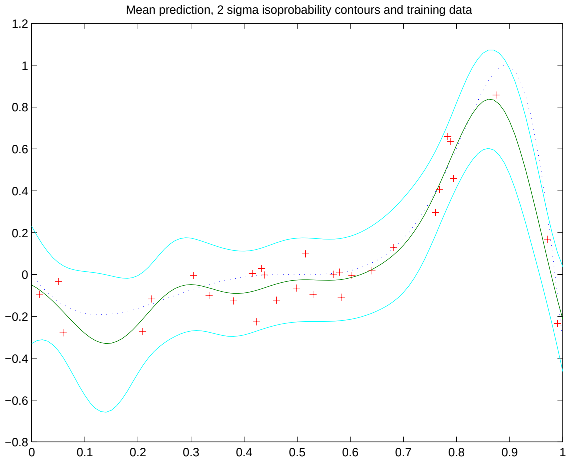


Fig. 1. Modelling a nonlinear function using a Gaussian Process prior. Model mean and 2σ contours, with target function and training data

The cost function proposed is the mean squared error from a bounded reference signal $y_d(t)$, with a penalty term λ on the control effort:

$$J = E\{(y_d(t+1) - y(t+1))^2\} + \lambda u(t)^2. \quad (9)$$

Using the fact that $\mathrm{Var}\{y\} = E\{y^2\} - \mu_y^2$, where $\mu_y = E\{y\}$, the cost function can be written as:

$$J = (y_d(t+1) - E\{y(t+1)\})^2 + \\ \mathrm{Var}\{y(t+1)\} + \lambda u(t)^2.$$

Note that we have not 'added' the model uncertainty term, $\mathrm{Var}\{y(t+1)\}$, to the classical quadratic cost function – most conventional work has ignored it, or have added extra terms to the cost function,or have pursued other sub-optimal solutions (Wittenmark, 1995; Filatov and Unbehauen, 2000).

If we now wish to find the optimal $u(t)$, we need the derivative of J,

$$\frac{\partial J}{\partial u(t)} = -2\left(y_d(t+1) - \mu_y\right)\frac{\partial \mu_y}{\partial u(t)} + \\ \frac{\partial \mathrm{Var}\{y(t+1)\}}{\partial u(t)} + 2\lambda u(t). \quad (10)$$

With most models, estimation of $\mathrm{Var}\{y(t+1)\}$, or $\frac{\partial \mathrm{Var}\{y(t+1)\}}{\partial u(t)}$ would be difficult, but with the Gaussian process prior (assuming smooth, differentiable covariance functions – see (O'Hagan, 1992)) the following straightforward analytic solutions can be obtained:

$$\begin{aligned}\frac{\partial \mu_y}{\partial u(t)} &= \frac{\partial \Sigma_{21}}{\partial u(t)}\Sigma_1^{-1}\mathbf{y}_1\\ \text{Var}\{y(t+1)\} &= \Sigma_2 - \Sigma_{21}\Sigma_1^{-1}\Sigma_{21}^T,\\ \frac{\partial \text{Var}\{y(t+1)\}}{\partial u(t)} &= \frac{\partial \Sigma_2}{\partial u(t)} -\\ &\qquad \Sigma_{21}\Sigma_1^{-1}\frac{\partial \Sigma_{21}^T}{\partial u(t)} - \frac{\partial \Sigma_{21}}{\partial u(t)}\Sigma_1^{-1}\Sigma_{21}^T.\end{aligned}$$

4.1 *Example: linear in parameters GP*

As an introductory example, if we have a GP to represent a linear model, for a target system of the form

$$\begin{aligned}y(t+1) &= w_1 x(t) + w_2 u(t) + e(t+1)\\ &= [w_1 w_2]\phi^T(t) + e(t+1)\end{aligned}$$

where $\Sigma_1 = \sigma_n I + \mathbf{\Phi}_1\Sigma_w\mathbf{\Phi}_1^T$, and $\Sigma_{21} = \phi\Sigma_w\mathbf{\Phi}_1$, where in this case we use the simple linear basis $\phi(t) = \left[\, x(t) \;\; u(t) \,\right]$, and Σ_w is a diagonal matrix with the vector w^2 as elements

$$\begin{aligned}\mu_y &= \phi\Sigma_w\mathbf{\Phi}_1\Sigma_1^{-1}\mathbf{y} = \phi^T\mathbf{b} = \left[\, x(t) \;\; u(t) \,\right]\begin{bmatrix} b_1 \\ b_2 \end{bmatrix}\\ \text{Var}\{y\} &= \sigma_n + \phi\Sigma_w\phi^T - \phi\Sigma_w\mathbf{\Phi}_1^T\Sigma_1^{-1}\mathbf{\Phi}_1\Sigma_w\phi^T,\\ &= \phi\mathbf{V}\phi^T \qquad (11)\end{aligned}$$

where,

$$\begin{aligned}\mathbf{b} &= \Sigma_w\mathbf{\Phi}^T\Sigma_1^{-1}\mathbf{y}\\ \mathbf{V} &= \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{bmatrix} = \Sigma_w - \Sigma_w\mathbf{\Phi}_1^T\Sigma_1^{-1}\mathbf{\Phi}_1\Sigma_w\end{aligned}$$

for notational convenience. $v_{21} = v_{12}$ (SISO).

$$\begin{aligned}\frac{\partial \text{Var}\{y(t+1)\}}{\partial u(t)} &= x(t)v_{12} + u(t)v_{22}\\ \frac{\partial J}{\partial u(t)} &= -2(\big(y_d(t+1) - \left[\, x(t) \;\; u(t) \,\right]\mathbf{b}\big)\, b_2^T\\ &\qquad + x(t)v_{12} + u(t)v_{22} + \lambda u(t))\end{aligned}$$

At $\frac{\partial J}{\partial u(t)} = 0$,

$$\begin{aligned}u(t) &= ((y_d(t+1) - x(t)b_1)b_2^T + x(t)v_{12})\\ &\qquad (b_2 b_2^T + v_{22} + \lambda)^{-1} \qquad (12)\end{aligned}$$

To understand the effect of the variance term in the cost function on the behaviour of the control law we will create two variables $\alpha = x(t)v_{12}$, and $\beta = v_{22}$. β is independent of the current $\phi(t), y(t), y_d(t+1)$. Note that the β term makes the use of a regularising constant λ redundant in many applications. Only when there is no model uncertainty will $\beta = 0$. In the experiments in this paper $\lambda = 0$.

$$u = ((y_d(t+1) - x(t)b_1)b_2^T + \alpha)(b_2 b_2^T + \beta + \lambda)^{-1}$$

After $u(t)$ has been calculated, applied, and the output observed, we add the information $x(t), u(t), y(t+1)$ to the training set, and the new Σ_1 increases in size to $(N_1 + 1) \times (N_1 + 1)$. Obviously, this naive approach will only work for relatively small data sets, and for a more general solution, expected utility based selection of data for use in an active training set will be used.

4.2 *Gaussian Process, nonlinear in x*

If to model systems of the form,

$$y(t+1) = f(x(t)) + wu(t) + e(t+1)$$

we use a covariance function which is the sum of the linear one used above, and one nonlinear in the state $x(t)$, but not in control input $u(t)$. I.e. $C(\phi_i, \phi_j) = C_{nl}(x_i, x_j) + C_{lin}(u_i, u_j)$, so $\Sigma_1 = N_1 + L_1$, where $L_1 = U_1\Sigma_w U_1^T$ is the covariance matrix of C_{lin} for all u in the training set, and N_1 is the covariance matrix of C_{nl} for all x in the training set. The variance of the output is:

$$\begin{aligned}\text{Var}\{y(t+1)\} &= N_2 + L_2 - (N_{21} + L_{21})\Sigma_1^{-1}(N_{12} + L_{12})\\ &= u(t)\mathbf{V}u(t)^T - 2N_{21}\Sigma_1^{-1}U_1\Sigma_w u(t)^T\\ &\quad + N_2 - N_{21}\Sigma_1^{-1}N_{12}\\ &= u(t)\mathbf{V}u(t)^T - 2\alpha u(t) + N_2 - N_{21}\Sigma_1^{-1}N_{12}\\ \frac{\partial J}{\partial u(t)} &= 2\left(y_d(t+1) - N_{21}\Sigma_1^{-1}\mathbf{y}_1 - u(t)\mathbf{b}\right)\mathbf{b}^T\\ &\quad -\alpha + u(t)\mathbf{V} + \lambda u(t)),\end{aligned}$$

where $\alpha = N_{21}\Sigma_1^{-1}U_1\Sigma_w$ and $\mathbf{V} = \Sigma_w - \Sigma_w U_1^T\Sigma_1^{-1}U_1\Sigma_w$. Taking the partial derivative:

$$\begin{aligned}\frac{\partial J}{\partial u(t)} &= 2\left(y_d(t+1) - N_{21}\Sigma_1^{-1}\mathbf{y}_1 - u(t)\mathbf{b}\right)\mathbf{b}^T,\\ &\quad -\alpha + u(t)\beta + \lambda u(t)),\end{aligned}$$

the constant β and the vector $\mathbf{b}$ are defined as $\beta = \mathbf{V}$, and $\mathbf{b} = \frac{\partial L_{21}}{\partial u(t)}\Sigma_1^{-1}\mathbf{y} = \Sigma_w U_1^T\Sigma_1^{-1}\mathbf{y}$. At $\frac{\partial J}{\partial u(t)} = 0$,

$$\begin{aligned}u(t) &= ((y_d(t+1) - N_{21}\Sigma_1^{-1}\mathbf{y}_1)\mathbf{b}^T - \alpha)\\ &\qquad (\mathbf{b}\mathbf{b}^T + \beta + \lambda)^{-1}.\end{aligned}$$

4.3 *Affine in u, for given x*

The above was for a linear weighting of the control $y(t+1) = f(x(t)) + wu(t)$ with the control input affecting the output through the covariance function $C_{lin}(u_i, u_j)$ which is then $U\Sigma_w U^T$. In the case of a weighting dependent on x, $y(t+1) = f(x(t)) + g(x(t))u(t)$, we can make the covariance contribution from the control inputs an affine function where the gradient will wander with state $x(t)$. I.e. $\Sigma_w = C(x_i, x_j)$, so

$$C_{aff}(\phi_i, \phi_j) = u_i C_u(x_i, x_j) u_j \qquad (13)$$

The covariance function C_u can be parameterised in any suitable way. Here we use the same structure as in C_{nl} above but we use different values of the hyperparameters to those used in C_{nl}.

5. SIMULATION RESULTS

To illustrate the feasibility of the approach we used it to control a simple system based on noisy observed responses. We start off with only *two* training points, and add subsequent data to the model during operation. The model has had no prior adaptation to the system before the experiment, and the covariance functions chosen are very general. Model hyperparameters are adapted after each iteration using a conjugate descent algorithm. The priority here is to show that the GP models are capable of adaptive nonlinear control, even when no priors have been placed on their hyperparameters. A more complete Bayesian approach would be to use the full prior structure in O'Hagan (1978), which would also lead to increased robustness and higher performance in the early stages of adaptation. Let the non-linear system be:

$$y(t+1) = f(\mathbf{x}(t)) + g(\mathbf{x}(t))u(t) + e(t+1)$$

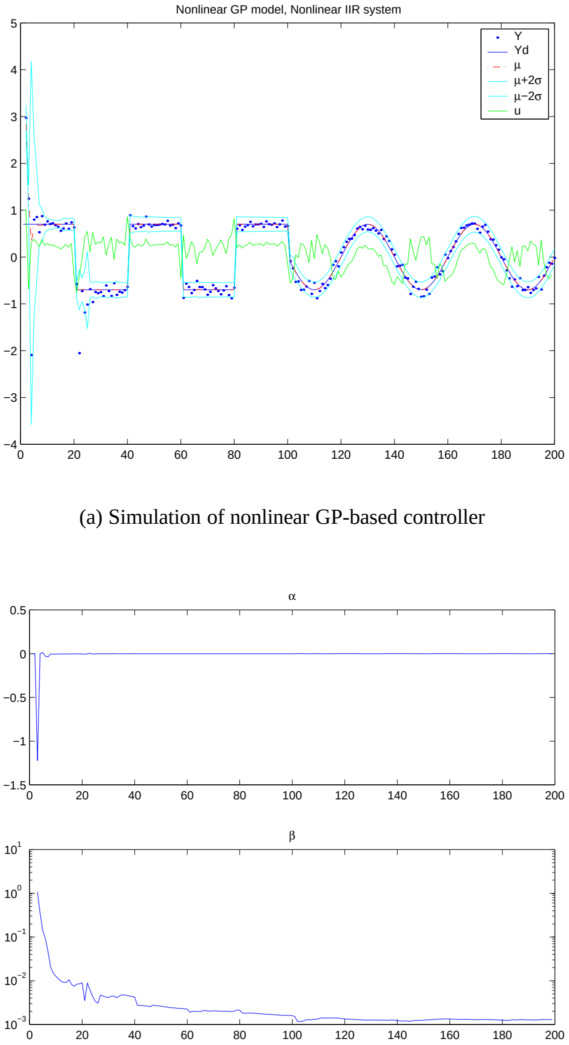
where $\mathbf{x} = y(t)$, $f(\mathbf{x}) = \sin(y(t)) + \cos(3y(t))$ and $g(\mathbf{x}) = 2$, subject to noise with variance $\sigma^2 = 0.001$ (Fabri and Kadirkamanathan, 1998).

Note how in Figure 2 β is large in the early stages of learning, but decreasing with the decrease in variance, showing how the regularising effect enforces caution in the face of model uncertainty, but reduces caution as the model accuracy increases. α is also larger in the early stages of learning, essentially adding an excitatory component. We expect α will play a more significant role when the covariance of the control signal depends on the state. Figure 3 shows the development in the mean mapping from $x(t), u(t)$ to $y(t+1)$ as the system acquires data.

6. CONCLUSIONS

This work has presented a novel adaptive controller based on non-parametric models. The control design is based on the expected value of a quadratic cost function, leading to a controller that not only will minimise the squared difference between the reference signal and the expected value of the output, but will also try to minimise the variance of the output. This leads to a robust control action during adaptation. Simulation results, considering linear and non-linear systems, demonstrate the interesting characteristics of this type of adaptive control algorithm.

GP's have been successfully adopted from their statistics origins by the neural network community (Williams, 1998). This paper is intended to bring the GP approach to the attention of the control community, and to show that the basic approach is a competitive approach for modelling and control of nonlinear dynamic systems, even when little attempt has been made to analyse the designer's prior knowledge of the system – there is much more that can be taken from the Bayesian approach to use in the dual control and nonlinear control areas, e.g. use of the framework suggested in (O'Hagan, 1978) for optimal experiment design with Gaussian process priors could improve exploration.

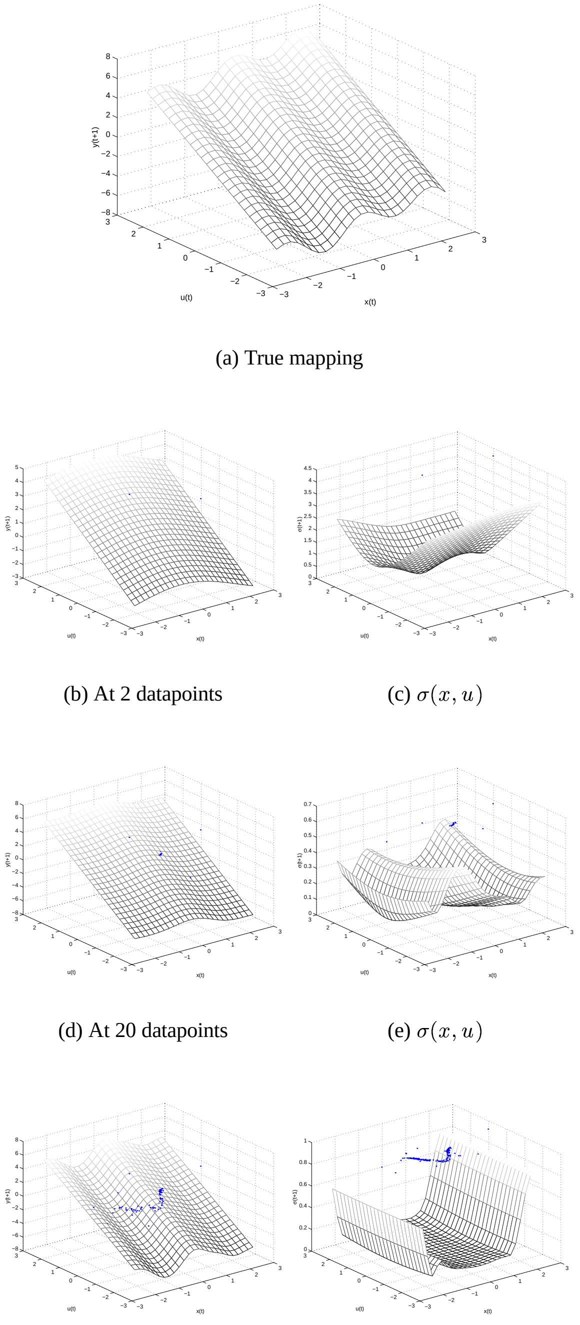


(a) Simulation of nonlinear GP-based controller

(b) α and β (regularisation term)

Fig. 2. Simulation results showing modelling accuracy, control signals, tracking behaviour and levels of α and β at each stage.

The Gaussian Process approach to regression is simple and elegant, and can model nonlinear problems in a probabilistic framework. The disadvantage is its computational complexity, as estimating the mean $\mu_{21} = \hat{y}(\phi)$ requires a matrix inversion of the $N_1 \times N_1$ covariance matrix, which becomes problematic for identification data where $N_1 > 1000$. In transient regimes, however, we have very few data points and we wish to make robust estimates of model behaviour, which are now possible. The robust inference of the GP approach in sparsely populated spaces makes it partic-



(a) True mapping

(b) At 2 datapoints

(c) $\sigma(x, u)$

(d) At 20 datapoints

(e) $\sigma(x, u)$

(f) At 100 datapoints

(g) $\sigma(x, u)$

Fig. 3. Mean surface (mesh) $y(t+1) = f(x(t), u(t))$ over the space $x \times u$, with $\sigma(x(t), u(t))$ uncertainty surfaces.

ularly promising in multivariable and high-order systems. Further work is underway to address the control of multivariable systems, nonminimum-phase systems and implementation efficiency issues.

7. ACKNOWLEDGEMENTS

Both authors are grateful for support from FONDECYT Project 700397. RM-S gratefully acknowledges the support of the *Multi-Agent Control* Research Training Network supported by EC TMR grant HPRN-CT-1999-00107, and the EPSRC grant *Modern statistical approaches to off-equilibrium modelling for nonlinear system control* GR/M76379/01.

8. REFERENCES

Bittanti, S. and L. Piroddi (1997). Neural implementation of GMV control shemes based on affine input/output models. *Proc. IEE Control Theory Appl.* (6), 521–530.

Chen, F.C. and H. Khalil (1995). Adaptive control of a class of nonlinear discrete-time systems. *IEEE. Trans. Automatic Control* **40**(5), 791–801.

Fabri, S. and V. Kadirkamanathan (1998). Dual adaptive control of stochastic systems using neural networks. *Automatica* **14**(2), 245–253.

Filatov, N.M. and H. Unbehauen (2000). Survey of adaptive dual control methods. *Proc. IEE Control Theory Appl.* (1), 119–128.

Leith, D. J., R. Murray-Smith and W. E. Leithead (2000). Nonlinear structure identification: A Gaussian Process prior/Velocity-based approach. In: *Control 2000, Cambridge*.

Liang, F. and H.A. ElMargahy (1994). Self-tuning neurocontrol of nonlinear systems using localized polynomial networks with CLI cells. In: *Proceedings of the American Control Conference*. Baltimore, Maryland. pp. 2148–2152.

Murray-Smith, R. and A. Girard (2001). Gaussian Process priors with ARMA noise models. In: *Irish Signals and Systems Conference, Maynooth*. pp. 147–152.

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, 1999*. pp. BA–14.

Narendra, K.S. and P. Parthasarathy (1990). Identification and control of dynamical systems using neural networks. *IEEE. Trans. Neural Networks* **1**(1), 4–27.

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

O'Hagan, A. (1992). Some Bayesian numerical analysis. In: *Bayesian Statistics 4* (J. M. Bernardo, J. O. Berger, A. P. Dawid and A. F. M. Smith, Eds.). pp. 345–363. Oxford University Press.

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

Wittenmark, R. (1995). Adaptive dual control methods: An overview. In: *Proceedings of the 5th IFAC Symposium on Adaptive systems in Control and Signal Processing*. Budapest. pp. 67–92.