



UNIVERSITY
of
GLASGOW

Sbarbaro, D. and Murray-Smith, R. (2005) Self-tuning control of non-linear systems using gaussian process prior models. *Lecture Notes in Computer Science 3355*:pp. 140-157.

<http://eprints.gla.ac.uk/3720/>

Self-tuning Control of Non-linear Systems Using Gaussian Process Prior Models

Daniel Sbarbaro¹ and Roderick Murray-Smith^{2,3}

¹ Departamento de Ingeniería Eléctrica, Universidad de Concepción, Chile
dsbarbar@die.udec.cl

² Department of Computing Science, University of Glasgow, Scotland
rod@dcs.gla.ac.uk

³ Hamilton Institute, NUI Maynooth, Co. Kildare, Ireland

Abstract. Gaussian Process prior models, as used in Bayesian non-parametric statistical models methodology are applied 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) in areas of high uncertainty. As a consequence, the controller has dual features, since it both tracks a reference signal and learns a model of the system from observed responses. The general method and its unique features are illustrated on simulation examples.

1 Introduction

Linear control algorithms have been successfully applied to control nonlinear systems, since they can adapt their parameters to cope the nonlinear characteristics of real systems. However, their performance degrades as the system undergoes rapid and larger changes in its operating point. Several authors have proposed the use of non-linear models as a base to build nonlinear adaptive controllers. Agarwal and Seborg [1], for instance, have proposed the use of known nonlinearities, capturing the main characteristic of the process, to design a Generalized Minimum Variance type of self-tuning controller. In many applications, however, these nonlinearities are not known, and nonlinear parameterisation must be used instead. A popular choice has been the use of Artificial Neural Networks for estimating the nonlinearities of the system [2, 3, 4, 5]. All these works have adopted the certainty equivalence principle for designing the controllers, 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 nonlinear models, the accuracy of the model predictions should also be taken into account. A common approach to consider the uncertainty in the parameters, is to add an extra term in the cost function of a Minimum Variance controller, which penalizes the uncertainty in the parameters of the nonlinear approximation [6]. Another similar approach based on the minimization of two separate cost functions, has been proposed in [7], the first one is used to improve the parameter estimation and the second one to drive the system output to follow a given reference signal. This approach is called bicriterial, and it has also been extended to deal with nonlinear systems [8].

Most of these 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 of the parameters. 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.

Non-parametric models retain the available data and perform inference conditional on the current state and local data (called ‘smoothing’ in some frameworks). As the data are used directly in prediction, unlike the parametric methods more commonly used in control contexts, non-parametric methods have advantages for off-equilibrium regions, since normally in these regions the amount of data available for identification is much smaller than that available in steady state. The uncertainty of model predictions can be made dependent on local data density, and the model complexity automatically related to the amount and distribution of 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 non-parametric model, the bias is typically less than in parametric models.

Non-parametric models are also well-suited to initial data analysis and exploration, as they are powerful models of the data, with robust behaviour despite few prior structural assumptions. This paper describes an approach based on Gaussian process priors, as an example of a non-parametric model with particularly nice analytic properties. This 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, depending on model accuracy. This paper expands on previous work [9] by making the cost function more flexible, introducing priors and investigating modelling and control performance for nonlinear systems affine in control inputs.

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 with parameters of models with fixed structure [10, 11].

The paper is organised as follows: section 2 describes the characteristics of non-parametric models. Section 3 introduces Gaussian Process priors. Section 4 illustrates how to design controllers based on the above representation. In section 5, we illustrate the control behaviour via simulation. Finally, some conclusions and future directions are outlined.

2 Controller Design

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

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

where $\mathbf{x}(t)$ is the state vector at a discrete time t , which in this paper will be defined as $\mathbf{x}(t) = [y(t), \dots, y(t-n), u(t-1), \dots, u(t-m)]$, $y(t+1)$ the output, $u(t)$ the current control vector, f and g are unknown smooth nonlinear functions. We also assume that g is bounded away from zero. In addition, it is also assumed that the system is minimum phase, as defined in [4]. For notational simplicity we consider single control input systems, but extending the presentation to vector $u(t)$ is trivial. The noise term $\varepsilon(t)$ is assumed zero mean Gaussian, but with unknown variance σ_n^2 . The control strategy consists in choosing a control variable $u(t)$ so as to minimize the following cost function:

$$J = E\{(y_d(t+1) - y(t+1))^2\} + (R(q^{-1})u(t))^2, \quad (2)$$

where $y_d(t)$ is a bounded reference signal, the polynomial $R(q^{-1})$ is defined as:

$$R(q^{-1}) = r_0 + r_1q^{-1} + \dots + r_{n_r}q^{-n_r} \quad (3)$$

where q^{-1} is a unit backward shift operator. The polynomial coefficients can be used as tuning parameters.

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

$$J = (y_d(t+1) - E\{y(t+1)\})^2 + \text{Var}\{y(t+1)\} + (R(q^{-1})u(t))^2. \quad (4)$$

Note that we have not ‘added’ the model uncertainty term, $\text{Var}\{y(t+1)\}$, to the classical quadratic cost function – most conventional work has ‘ignored’ it, or have added extra terms to the cost function [10, 11].

Since f and g are unknown, it will be necessary to use a model to predict the output of the system.

3 Non-parametric Models: Gaussian Process Priors

In a Bayesian framework the model must be based on a prior distribution over the infinite-dimensional space of functions. As illustrated in [12], such priors can be defined as Gaussian processes. These models have attracted a great deal of interest recently, in for example reviews such as [13]. Rasmussen [14] showed empirically that Gaussian processes were extremely competitive with leading nonlinear identification methods on a range of benchmark examples. The further advantage that they provide analytic predictions of model uncertainty makes them very interesting for control applications. Use of GPs in a control systems context is discussed in [15, 16]. A variation which can include ARMA noise models is described in [17]. k -step ahead prediction with GP’s is described in [18] and integration of prior information in the form of state or control linearisations is presented in [19].

Let’s assume a model $y(i) = h(\phi(i)) + \varepsilon(i)$, where $\phi(i) \in \mathbb{R}^p$ is the input vector, $\varepsilon(i)$ is a noise term, and $y(i) \in \mathbb{R}$ is the corresponding output. Instead of parameterizing $h(\phi(i))$ as a parametric model, we obtain an inference of function $h(\phi(i))$ by computing the distribution $P(h(\phi(i)) | \mathcal{D}, \phi(i))$ of the scalar output $h(\phi(i))$, given the input vector

$\phi(i)$ and a set of N training data points $\mathcal{D} = \{(\phi(i), y(i)) \mid i = 1, 2, \dots, N\}$. The given N data pairs used for identification are stacked in matrices Φ_N and \mathbf{y}_N . The vector with the stacked values of the function is defined as \mathbf{h}_N . A Gaussian process represents the simplest form of prior over functions and introduces a set of N stochastic variables $\mathbf{H}(1) \dots \mathbf{H}(N)$, for modelling the function at the corresponding inputs $\phi(1) \dots \phi(N)$ [14]. Then, a multivariable prior distribution with zero mean⁴ and covariance function \mathbf{K}_N is assumed for these variables:

$$P(h(1) \dots h(N) | \Phi_N) \propto \exp\left[-\frac{1}{2}(\mathbf{h}_N^T \mathbf{K}_N^{-1} \mathbf{h}_N)\right], \quad (5)$$

this prior specifies the joint distribution of the function values given the inputs. On the other hand, the likelihood relates the underlying function which is modeled by the function $h(\phi(i))$ to the observed outputs $y(i)$, $i = 1, \dots, N$. If the noise is assumed to be Gaussian with some unknown variance σ_n^2 , then by combining the prior and the likelihood the distribution of the observed data will simply be:

$$P(\mathbf{y}_N | \Phi_N) \propto \exp\left[-\frac{1}{2}(\mathbf{y}_N^T \mathbf{C}_N^{-1} \mathbf{y}_N)\right], \quad (6)$$

where $\mathbf{C}_N = \mathbf{K}_N + \sigma_n^2 \mathbf{I}$. The prediction $y(N+1)$ given the data \mathcal{D} and a new input vector $\phi(N+1)$ can be calculated by obtaining the following conditioned Gaussian distribution :

$$\begin{aligned} P(y(N+1) | \mathcal{D}, \phi(N+1)) &= \frac{P(\mathbf{y}_{N+1} | \Phi_N, \phi(N+1))}{P(\mathbf{y}_N | \Phi_N)} \\ &\propto \exp\left[-\frac{1}{2}(\mathbf{y}_{N+1}^T \mathbf{C}_{N+1} \mathbf{y}_{N+1} - \mathbf{y}_N^T \mathbf{C}_N \mathbf{y}_N)\right] \end{aligned} \quad (7)$$

where \mathbf{C}_{N+1} can be partitioned as:

$$\mathbf{C}_{N+1} = \begin{bmatrix} \mathbf{C}_N & \mathbf{k} \\ \mathbf{k}^T & \kappa \end{bmatrix}. \quad (8)$$

The partitioned form (8) can be used, as it is illustrated in [21], to obtain the parameters of the conditioned Gaussian distribution:

$$P(y(N+1) | \mathcal{D}, \phi(N+1)) = \frac{1}{(2\pi\hat{\sigma}_y^2)^{\frac{1}{2}}} \exp\left[-\frac{(y(t+1) - \hat{\mu}_y)^2}{2\hat{\sigma}_y^2}\right],$$

where the mean and variance are:

$$\hat{\mu}_y = \mathbf{k}^T \mathbf{C}_N^{-1} \mathbf{y}_N, \quad (9)$$

$$\text{Var}\{y\} = \hat{\sigma}_y^2 = \kappa - \mathbf{k}^T \mathbf{C}_N^{-1} \mathbf{k}. \quad (10)$$

We can use $\hat{\mu}_y(\phi(N+1))$ as the expected model output, with a variance of $\hat{\sigma}(\phi(N+1))^2$. Thus the dynamical system (1) can be modelled under this framework by considering the input vector as $\phi(i) = [\mathbf{x}(t) \ u(t)]$ and the corresponding output $y(i) = y(t+1)$.

⁴ Note, as explained in [20] the zero mean assumption does not mean that we expect the regression function to be spread equally on either side of zero. If a covariance function had a large constant term the actual function could be always positive or always negative over the range of interest. The zero mean reflects or ignorance as to what that sign will be. There are good numerical computational reasons for transforming data to be zero mean.

3.1 The Covariance Function

The Normal assumption may seem strangely restrictive initially, but represents a powerful tool since the model's prior expectations can be adapted to a given application by altering the covariance function. The choice of this function is only constrained in that it must always generate a non-negative definite covariance matrix for any inputs Φ , so we can represent a spectrum of systems from very local 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 K and one due to measurement noise C_n . The entries ij of this matrix are then:

$$\mathbf{C}_{N_{ij}} = K(\Phi_{N_i}, \Phi_{N_j}; \Theta) + C_n(\Phi_{N_i}, \Phi_{N_j}; \Theta_n) \quad (11)$$

where Θ denotes a set of parameters, which in the GP framework are also called hyperparameters. As pointed out in [14] it is convenient to specify priors in terms of the hyperparameters, which then can be adapted as the model is fit to the identification data. The covariance associated to the noise $C_n()$ could be $\delta_{ij}\mathcal{N}(\Phi_N; \Theta_n)$, which would be adding a noise model \mathcal{N} to the diagonal entries of \mathbf{C}_N . This framework allows the use of different noise models, as discussed in [17], where ARMA noise models were used.

Since the output is an affine function of the control input, it is reasonable to propose a covariance function with a contribution from the control inputs as an affine function as well:

$$K(\phi(i), \phi(j); \Theta) = C_x(\mathbf{x}(i), \mathbf{x}(j); \Theta_x) + u(i)C_u(\mathbf{x}(i), \mathbf{x}(j); \Theta_u)u(j) \quad (12)$$

where the first term represents the contribution of the state vector and the second one the contribution of the input signal. The covariance function C_u can be parameterised in any suitable way. Here, we use the same structure as in C_x above, but with different set of hyperparameters, Θ_u , to those used in C_x .

The covariance function for C_x represents a straightforward covariance function proposed by [14], which has demonstrated to work well in practice:

$$C_x(\mathbf{x}(i), \mathbf{x}(j); \Theta) = v_0\rho(|\mathbf{x}(i) - \mathbf{x}(j)|, \alpha) + \sum_{k=1}^p a_k x_k(i)x_k(j) + a_0, \quad (13)$$

so that the parameter vector $\Theta = \log[v_0, \alpha_1, \dots, \alpha_p, a_0]^T$ (the log is applied elementwise) and p is the dimension of vector \mathbf{x} . The parameters are defined to be the log of the variable in equation (13) since these are positive scale-parameters. The function $\rho(d)$ is a function of a distance measure d , which should be one at $d = 0$ and which should be a monotonically decreasing function of d . The one used here was

$$\rho(|\mathbf{x}(i) - \mathbf{x}(j)|, \alpha) = e^{-\frac{1}{2} \sum_{k=1}^p \alpha_k (x_k(i) - x_k(j))^2}. \quad (14)$$

The α_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 [22], which helps weight the importance of different input dimensions. Other bases which included a nonlinear transformation of

\mathbf{x} , like the RBF neural networks used in [6], could be put into this framework. The prior associated with this covariance function states that outputs associated with ϕ 's closer together should have higher covariance than points further apart.

The Gaussian Process approach to regression is simple and elegant, and can model nonlinear problems in a probabilistic framework. There tend also to be far fewer parameters to identify in the Gaussian Process approach than for competing approaches (such as e.g. artificial neural networks). The disadvantage is its computational complexity, as estimating the mean $\hat{\mu}_y$ requires a matrix inversion of the $N \times N$ covariance matrix, which becomes problematic for identification data where $N > 1000$. In transient regimes, however, we have very few data points and we wish to make robust estimates of model behaviour, which are now possible. This suggests that a multiple-model style partitioning of the state-space could make GPs more feasible in many applications [23].

Adapting the Covariance Function Parameters The hyperparameter vector Θ provides flexibility to define a family of covariance functions which provide suitable prior distributions over functions. In most cases we will only have uncertain knowledge of Θ . Given unknown hyperparameters we can use numerical methods such as Markov-Chain Monte Carlo (MCMC) to integrate over hyperparameters, or use maximum likelihood methods, with standard gradient-based optimisation tools to optimise hyperparameters. The log-likelihood l of the training data can be calculated analytically as [13]:

$$l = -\frac{1}{2} \log \det \mathbf{C}_N - \frac{1}{2} \mathbf{y}_N^T \mathbf{C}_N^{-1} \mathbf{y}_N - \frac{n}{2} \log 2\pi. \quad (15)$$

The partial derivative of the log likelihood with respect to the hyperparameters is:

$$\frac{\partial l}{\partial \theta_i} = -\frac{1}{2} \text{tr} \left[\mathbf{C}_N^{-1} \frac{\partial \mathbf{C}_N}{\partial \theta_i} \right] + \frac{1}{2} \mathbf{y}_N^T \mathbf{C}_N^{-1} \frac{\partial \mathbf{C}_N}{\partial \theta_i} \mathbf{C}_N^{-1} \mathbf{y}_N. \quad (16)$$

Given l and its derivative with respect to θ_i it is straightforward to use an efficient optimization program in order to obtain a local maximum of the likelihood.

In parametric models the parameters must be updated each sampling time, but in the nonparametric framework this is not necessary, as it will be illustrated in section 5, since the model also relies on the data contained in the identification data set.

Hierarchical Priors The hyperparameters of the covariance function will rarely be known exactly in advance, so they are usually given a vague prior distribution, such as a gamma prior [20].

$$p(\phi) = \frac{(a/2\omega)^{a/2}}{\Gamma(a/2)} \psi^{((a/2)-1)} \exp\left(-\frac{\psi a}{2\omega}\right) \quad (17)$$

where $\psi = \theta^{-2}$ for a hyperparameter θ . a is a positive shape parameter and ω is the mean of ψ . Large values of a produce priors for θ concentrated near ω^{-2} and small values lead to vague priors. Each hyperparameter of the covariance function can be given an independent prior distribution. If prior distributions on the hyperparameters, such

as equation (17) are used then obviously these are included in the likelihood equations and the derivative terms. The use of gamma priors does not add significant complexity to the optimisation, and if used appropriately makes the model behaviour more robust with small numbers of training data.

4 Derivation of Control Law

Given the cost function (4), and observations to time t , if we wish to find the optimal $u(t)$, we need the derivative of J ,

$$\frac{\partial J}{\partial u(t)} = -2(y_d(t+1) - \mu_y) \frac{\partial \mu_y}{\partial u(t)} + \frac{\partial \text{Var}\{y(t+1)\}}{\partial u(t)} + 2r_0 R(q^{-1})u(t). \quad (18)$$

With most models, estimation of $\text{Var}\{y\}$, or $\frac{\partial \text{Var}\{y\}}{\partial u(t)}$ would be difficult, but with the Gaussian process prior (assuming smooth, differentiable covariance functions – see [24]) the following straightforward analytic solutions can be obtained:

$$\frac{\partial \mu_y}{\partial u(t)} = \frac{\partial \mathbf{k}^T}{\partial u(t)} \mathbf{C}_N^{-1} \mathbf{y}_N \quad (19)$$

$$\frac{\partial \text{Var}\{y\}}{\partial u(t)} = \frac{\partial \kappa}{\partial u(t)} - 2\mathbf{k}^T \mathbf{C}_N^{-1} \frac{\partial \mathbf{k}}{\partial u(t)}. \quad (20)$$

The covariance matrix \mathbf{k} and κ can be expressed in terms of the independent control variable $u(t)$ as follows:

$$\mathbf{k} = \Omega_1 + u(t)\Omega_2 \quad (21)$$

$$\kappa = \Omega_3 + \Omega_4 u(t)^2, \quad (22)$$

where $\Omega_1 = C_x(\mathbf{x}(t), \Phi_N, \Theta_x)$, $\Omega_2 = C_x(\mathbf{x}(t), \Phi_N, \Theta_u) * U_N$, where U_N is a vector with all the values of $u(t)$ contained in the identification data set, and $*$ indicates elementwise multiplication of two matrices. $\Omega_3 = C_x(\mathbf{x}(t), \mathbf{x}(t), \Theta_x)$, and $\Omega_4 = C_x(\mathbf{x}(t), \mathbf{x}(t), \Theta_u)$. The final expressions for μ_y and $\text{Var}\{y\}$ are:

$$\mu_y = \Omega_1 \mathbf{C}_N^{-1} \mathbf{y}_N + \Omega_2 \mathbf{C}_N^{-1} \mathbf{y}_N u(t), \quad (23)$$

$$\text{Var}\{y\} = \Omega_3 + \Omega_4 u(t)^2 - (\Omega_1 + u(t)\Omega_2) \mathbf{C}_N^{-1} (\Omega_1 + u(t)\Omega_2)^T. \quad (24)$$

Taking the partial derivatives of the variance and the mean expressions and replacing their values in (18), it follows:

$$\begin{aligned} \frac{\partial J}{\partial u(t)} = & -2(y_d(t+1) - \Omega_1 \mathbf{C}_N^{-1} \mathbf{y}_N - \Omega_2 \mathbf{C}_N^{-1} \mathbf{y}_N u(t)) (\Omega_2 \mathbf{C}_N^{-1} \mathbf{y}_N)^T \\ & + 2\Omega_4 u(t) - 2\Omega_1 \mathbf{C}_N^{-1} \Omega_2^T - 2u(t)\Omega_2 \mathbf{C}_N^{-1} \Omega_2^T + 2r_0 R(q^{-1})u(t). \end{aligned} \quad (25)$$

At $\frac{\partial J}{\partial u(t)} = 0$, the optimal control signal is obtained as:

$$u(t) = \frac{(y_d(t+1) - \Omega_1 \mathbf{C}_N^{-1} \mathbf{y}_N) (\Omega_2 \mathbf{C}_N^{-1} \mathbf{y}_N)^T + \Omega_1 \mathbf{C}_N^{-1} \Omega_2^T - r_0 (R(q^{-1}) - r_0) u(t)}{r_0^2 + \Omega_4 - \Omega_2 \mathbf{C}_N^{-1} \Omega_2^T + \Omega_2 \mathbf{C}_N^{-1} \mathbf{y}_N (\Omega_2 \mathbf{C}_N^{-1} \mathbf{y}_N)^T}. \quad (26)$$

Note that equation (26) can also be presented as

$$u(t) = \frac{(y_d(t+1) - \Omega_1 \mathbf{C}_N^{-1} \mathbf{y}_N)(\Omega_2 \mathbf{C}_N^{-1} \mathbf{y}_N)^T + \alpha(t) - r_0(R(q^{-1}) - r_0)u(t)}{r_0^2 + \beta(t) + \Omega_2 \mathbf{C}_N^{-1} \mathbf{y}_N (\Omega_2 \mathbf{C}_N^{-1} \mathbf{y}_N)^T}. \quad (27)$$

where $\alpha(t) = \Omega_1 \mathbf{C}_N^{-1} \Omega_2^T$, and $\beta(t) = \Omega_4 - \Omega_2 \mathbf{C}_N^{-1} \Omega_2^T$. If we had not included the variance term in cost function (2), or if we were in a region of the state-space where the variance was zero, the optimal control law would be equation (27) with $\alpha = \beta = 0$. We can therefore see that analysing the values of α and β is a promising approach to gaining insight into the behaviour of the new form of controller. These terms make a control effort penalty constant, or regulariser unnecessary in many applications.

4.1 Adapting Control Behaviour with New Data

After $u(t)$ has been calculated, applied, and the output observed, we add the information $\mathbf{x}(t)$, $u(t)$, $y(t+1)$ to the training set, and the new \mathbf{C}_N increases in size to $N+1 \times N+1$. Obviously, given the expense of inverting \mathbf{C}_N for large N , this naive approach will only work for relatively small data sets. For a more general solution, we can potentially incorporate elements of Relevance Vector Machines [25], or use heuristics for selection of data for use in an active training set, as in e.g. [26].

We can then choose to optimise the hyperparameters of the covariance function to further refine the model, or keep the covariance function fixed, and just use the extra data points to improve model performance. In the next section, will be illustrated the performance obtained with both strategies.

5 Simulation Results

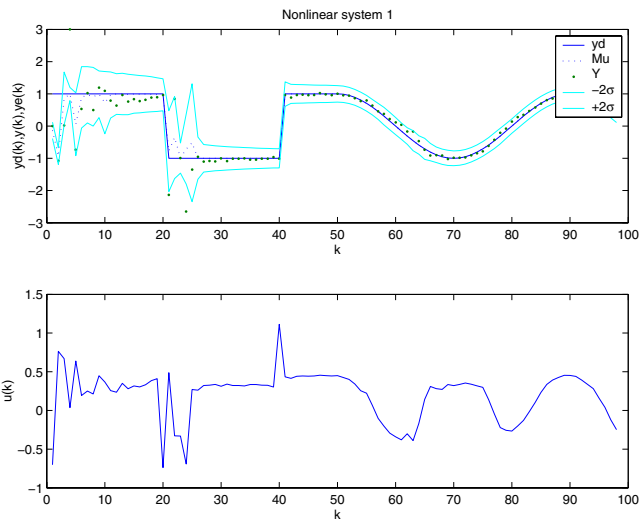
To illustrate the feasibility of the approach we used it to control several target plants based on noisy observed responses. We start off with only *two* training points, and add subsequent data to the model during operation. The model has had no prior adaptation to the system before the experiment. A gamma distribution was used for all hyperparameters, with ω set equal to the initial condition for each variable and shape parameter $a = 3$, indicating vague knowledge about the variable. The noise term σ_n , was given a tighter distribution, with $a = 5$. The covariance functions chosen are the same for all the experiments.

5.1 Non-linear System 1

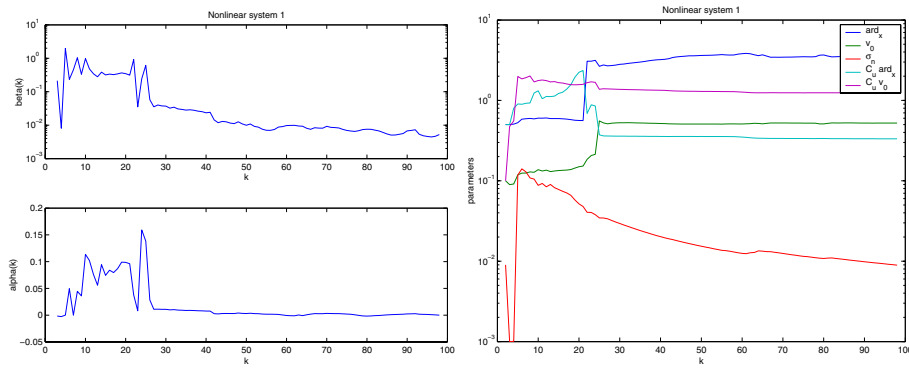
Let non-linear system 1 be:

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

where $\mathbf{x}(t) = y(t)$, $f(\mathbf{x}(t)) = \sin(y(t)) + \cos(3y(t))$ and $g(\mathbf{x}(t)) = 2 + \cos(y(t))$, subject to noise with variance $\sigma_n^2 = 0.001$ [6]. Model hyperparameters are adapted after each iteration using conjugate gradient descent optimisation algorithm.



(a) Simulation of nonlinear GP-based controller

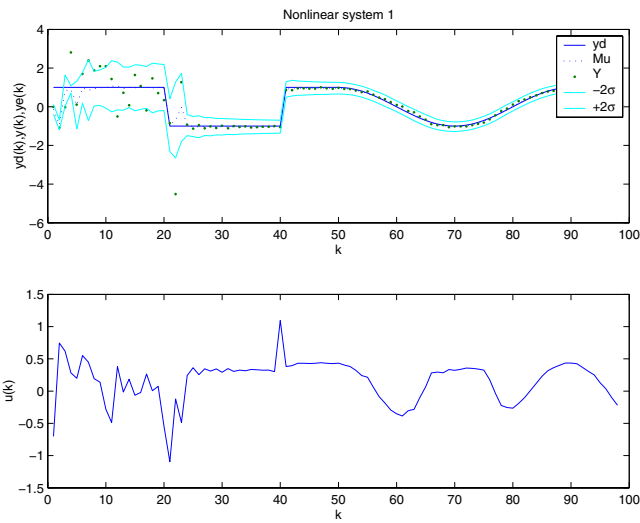


(b) α and β (regularisation term)

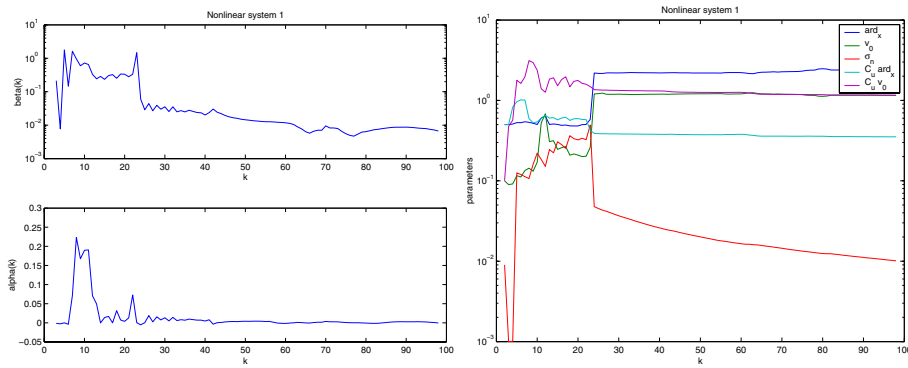
(c) Covariance function hyperparameters

Fig. 1. Simulation results for nonlinear system 1, showing modelling accuracy, control signals, tracking behaviour and levels of α and β at each stage.

Note how in Figure 1 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 uncertainty, but reduces caution as the model accuracy increases. In terms of the hyperparameters, most hyperparameters have converged by about 30 data points. The noise parameter σ_n decreases with increasing levels of training data. After this point the control signal u is also fairly smooth, despite the noisy nature of the data.



(a) Simulation of nonlinear GP-based controller



(b) α and β (regularisation term)

(c) Covariance function hyperparameters

Fig. 2. Simulation results on nonlinear system 1, *without including the α and β terms linked to the model variance in the control law.* Data shows modelling accuracy, control signals and tracking behaviour.

α can be seen to be larger in higher variance regions, essentially adding an excitatory component which decreases with the decrease in model uncertainty, and in this example plays almost no role after about iteration 30.

Figure 2 shows control performance on the same system where the variance part of the cost function is ignored (i.e. α and β are removed from the control law. In order

to achieve any reasonable control behaviour, we set $r_0 = 0.5$. As can be seen in the figure, the system still tracks the trajectory, and after iteration 30 there is little visible difference between the two control laws, but ignoring the variance does lead to the use of greater control effort, with larger model uncertainty in the early stages of learning. The hyperparameter estimates also fluctuate much more in the early stages of learning, when variance is not considered, although both systems converge on similar values after the initial stages. The constant nature of r_0 as a regularising term, as opposed to the dynamically changing $\alpha(t), \beta(t)$ makes controller design more difficult, as we can see that in early stages of learning it tends to be too small, reducing robustness, while later it is larger than $\alpha(t), \beta(t)$ damaging performance.

We now plot the nonlinear mappings involved in non-linear system 1, to give the reader a clearer impression of the adaptation of the system. The surfaces in figure 4 show the development in the mean mapping from $x(t), u(t)$ to $y(t+1)$ as the system acquires data, taken from the simulation shown in figure 1 at $t = 3, 20, 99$. For comparison, figure 5.1 shows the true mapping. Examining the surfaces in figure 4 we can see how the nonlinear mapping adapts gradually given increasing numbers of data points, but we also see that the standard deviation of the mapping also evolves in an appropriate manner, indicating clearly at each stage of adaptation the model uncertainty over the state-space. In the final plot, Figure 4(f) we can see a uniformly low uncertainty in the areas covered by data, but a rapid increase in uncertainty as we move beyond that in the x -axis. Note that the uncertainty grows much more slowly in the u -axis because of the affine assumption inherent in the covariance function, which constrains the freedom of the model.

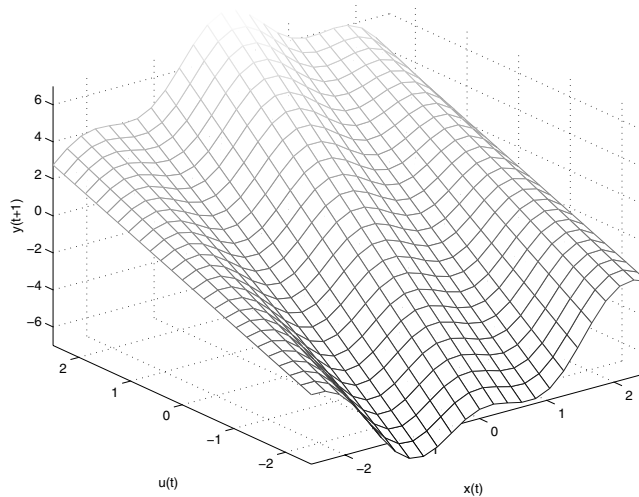


Fig. 3. True surface (mesh) of nonlinear system 1, $y(t+1) = \sin(y(t)) + \cos(3y(t)) + (2 + \cos(y(t)))u(t)$, over the space $y \times u$.

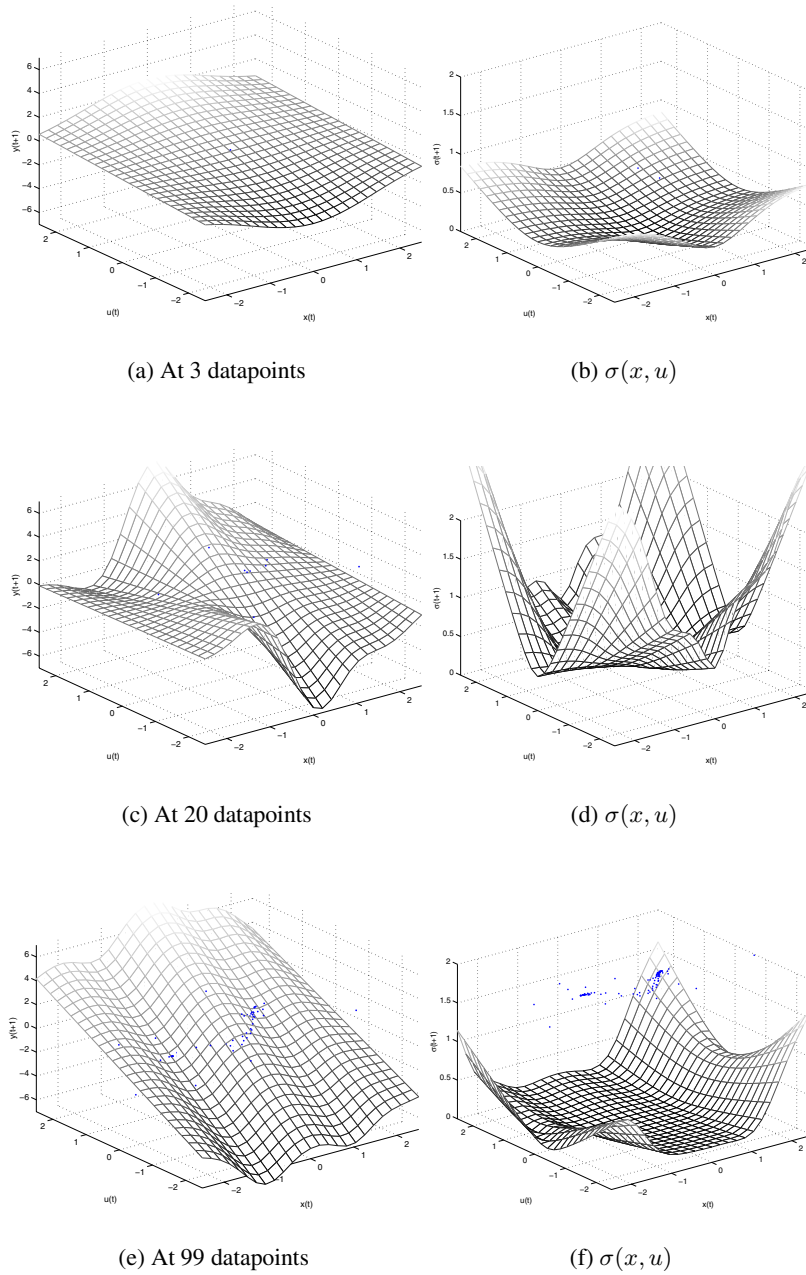


Fig. 4. Left-hand figures show mean surface (mesh) $y(t + 1) = f(x(t), u(t))$ of non-linear system 1 over the space $x \times u$ during the learning process. These can be compared to the true mapping in Figure 5.1. Right-hand figures show condition standard deviation $\sigma(x(t), u(t))$ surfaces. Each figure also shows the available data at that point in the learning process.

5.2 Non-linear System 2

The second nonlinear example considers the following non-linear functions:

$$f(\mathbf{x}(t)) = \frac{y(t)y(t-1)y(t-2)u(t-1)(y(t-2)-1)}{1+y(t-1)^2+y(t-2)^2}$$

$$g(\mathbf{x}(t)) = \frac{1}{1+y(t-1)^2+y(t-2)^2},$$

where $\mathbf{x} = [y(t) \ y(t-1) \ y(t-2) \ u(t-1)]^T$ [2]. The system noise has a variance $\sigma_n^2 = 0.001$, and we had 6 initial data points. The results are shown in Figure 5. Again, the trend of decreasing α and β can be seen, although they do increase in magnitude following changes in the system state towards higher uncertainty regions, showing that the control signal will be appropriately damped when the system moves to a less well-modelled area of the state-space. The hyperparameters in Figure 5(c) make few rapid changes, seeming well-behaved during learning.

Figure 6 illustrates the effect of keeping constant the initial set of hyperparameters. As it can be seen in the figure, even with this set of parameters, which have values very far away from the optimal ones, the system is capable of controlling the system.

The next example illustrates the use of the polynomial $R(q^{-1})$ for shaping the closed loop response. In this example, it was selected as $R(q^{-1}) = r_o - r_o q^{-1}$, so as to weight the control signal deviations for tuning the speed of response without introducing steady state errors, the associated cost function is:

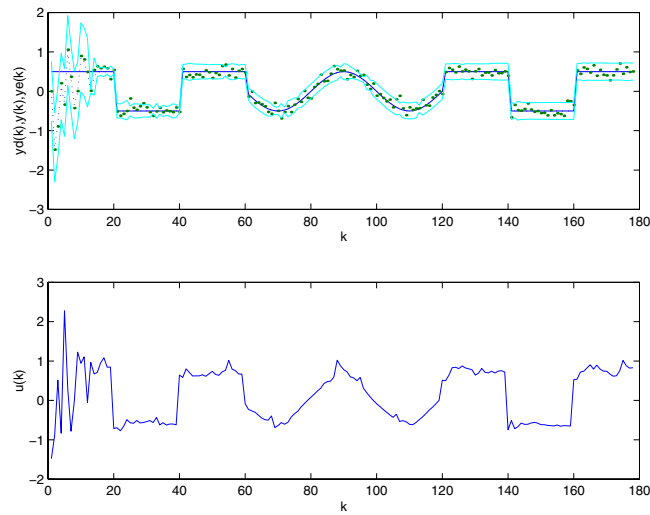
$$J = E\{(y_d(t+1) - y(t+1))^2\} + (r_o(u(t) - u(t-1)))^2. \quad (28)$$

The response obtained for a $r_o = 0.6$ is illustrated in Figure 7, where as we expected the speed of response is much slower than the case of having no control weighting.

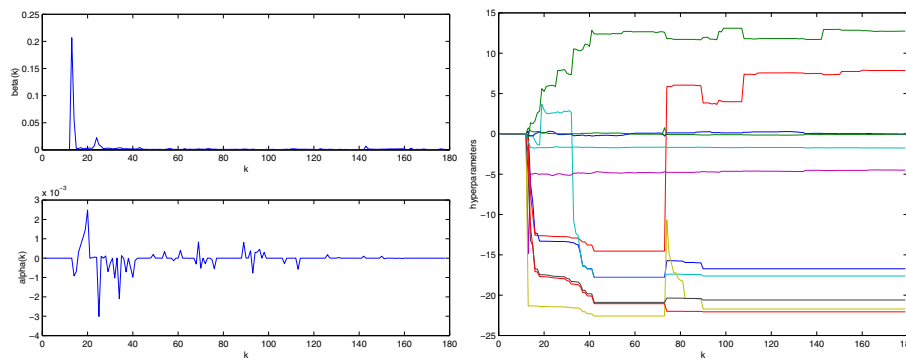
6 Conclusions

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, based on analytical estimates of model uncertainty. This leads to a robust control action during adaptation, and when extended to multi-step ahead prediction, forms the basis of full dual control with implicit excitatory components. Simulation results, considering linear and non-linear systems, demonstrate the interesting characteristics of this type of adaptive control algorithm.

The GP models are capable of high performance, with or without priors being placed on their hyperparameters. Use of gamma prior distributions led to increased robustness and higher performance in the early stages of adaptation with very few data points, but the relative advantage decreases with the amount of initial data available, as would be expected. Since the predictions are not only rely on the hyperparameters,



(a) Simulation of nonlinear GP-based controller

(b) α and β (regularisation term)

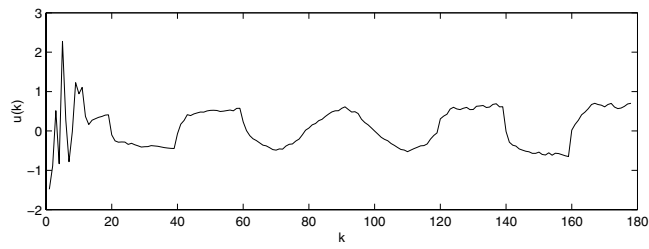
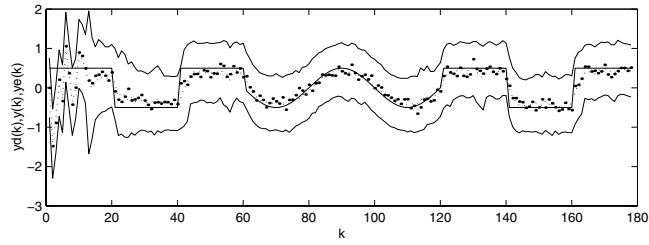
(c) Covariance function hyperparameters

Fig. 5. Simulation results for nonlinear system 2, showing modelling accuracy, control signals, tracking behaviour and levels of α and β at each stage.

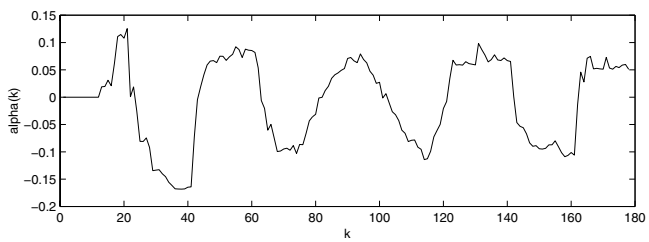
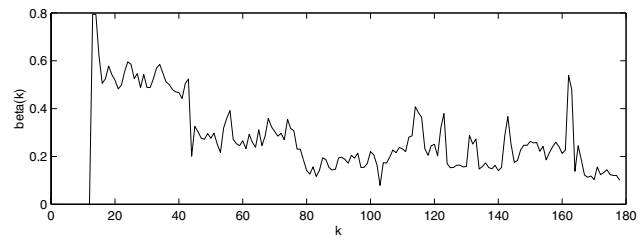
but also on the training data set, their on-line adaptation can be carried out at a sampling interval much bigger than the one used for controlling the system.

The additional polynomial term in the cost function can be used to shape the closed loop response without introducing steady state error.

GP's have been successfully adopted from their statistics origins by the neural network community [13]. This paper is intended to bring the GP approach to the attention

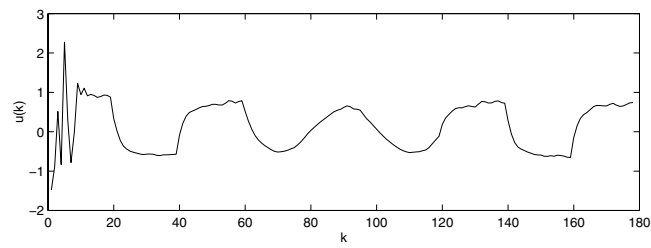
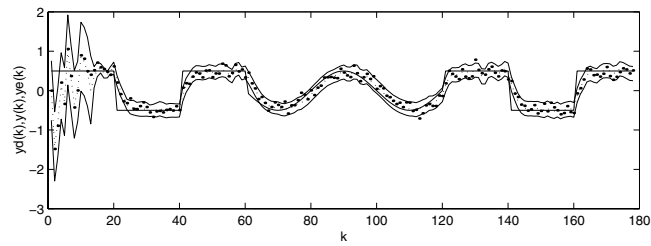


(a) Simulation of nonlinear GP-based controller



(b) α and β (regularisation term)

Fig. 6. Simulation results for nonlinear system 2 without adapting the hyperparameters, showing modelling accuracy, control signals, tracking behaviour and levels of α and β at each stage.



(a) Simulation of nonlinear GP-based controller

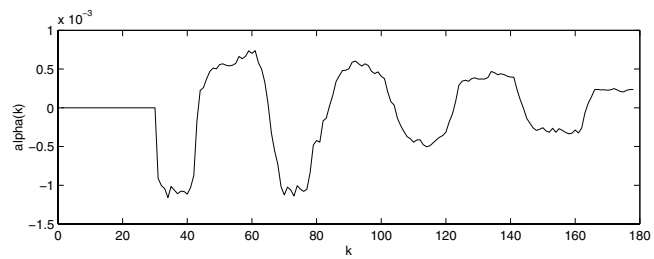
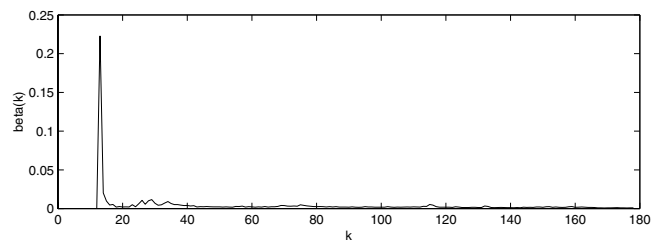
(b) α and β (regularisation term)

Fig. 7. Simulation results for nonlinear system 2 with $r_o = 0.6$, $r_1 = -0.6$, showing modelling accuracy, control signals, tracking behaviour and levels of α and β at each stage.

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.

Further work is underway to address the control of multivariable systems, non-minimum-phase systems and implementation efficiency issues. The robust inference of the GP approach in sparsely populated spaces makes it particularly promising in multivariable and high-order systems.

Acknowledgements

Both authors are grateful for support from FONDECYT Project 700397 and the Hamilton Institute. 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.

References

- [1] Agarwal, M., Seborg, D.E.: Self-tuning controllers for nonlinear systems. *Automatica* (1987) 209–214
- [2] Narendra, K., Parthasarathy, P.: Identification and control of dynamical systems using neural networks. *IEEE. Trans. Neural Networks* **1** (1990) 4–27
- [3] Liang, F., ElMargahy, H.: Self-tuning neurocontrol of nonlinear systems using localized polynomial networks with CLI cells. In: *Proceedings of the American Control Conference, Baltimore, Maryland* (1994) 2148–2152
- [4] Chen, F., Khalil, H.: Adaptive control of a class of nonlinear discrete-time systems. *IEEE. Trans. Automatic Control* **40** (1995) 791–801
- [5] Bittanti, S., Piroddi, L.: Neural implementation of GMV control schemes based on affine input/output models. *Proc. IEE Control Theory Appl.* (1997) 521–530
- [6] Fabri, S., Kadirkamanathan, V.: Dual adaptive control of stochastic systems using neural networks. *Automatica* **14** (1998) 245–253
- [7] Filatov, N., Unbehauen, H., Keuchel, U.: Dual pole placement controller with direct adaptation. *Automatica* **33** (1997) 113–117
- [8] Sbarbaro, D., Filatov, N., Unbehauen, H.: Adaptive dual controller for a class of nonlinear systems. In: *Proceedings of the IFAC Workshop on Adaptive systems in Control and Signal Processing, Glasgow, U.K.* (1998) 28–33
- [9] Murray-Smith, R., Sbarbaro, D.: Nonlinear adaptive control using non-parametric gaussian process prior models. In: *Proceedings of the 15th IFAC world congress, Barcelona, Spain* (2002)
- [10] Wittenmark, R.: Adaptive dual control methods: An overview. In: *Proceedings of the 5th IFAC Symposium on Adaptive systems in Control and Signal Processing, Budapest* (1995) 67–92
- [11] Filatov, N., Unbehauen, H.: Survey of adaptive dual control methods. *Proc. IEE Control Theory Appl.* (2000) 119–128
- [12] O'Hagan, A.: On curve fitting and optimal design for regression (with discussion). *Journal of the Royal Statistical Society B* (1978) 1–42

- [13] Williams, C.: Prediction with Gaussian process: From linear regression to linear prediction and beyond. In Jordan, M., ed.: *Learning and Inference in Graphical Models*, Kluwer (1998) 599–621
- [14] Rasmussen, C.: *Evaluation of Gaussian Process and other Methods for non-linear regression*. PhD thesis, Department of Computer Science, University of Toronto (1996)
- [15] Murray-Smith, R., Johansen, T., Shorten, R.: On transient dynamics, off-equilibrium behaviour and identification in blended multiple model structures. In: *Proceedings of the European Control Conference, Karlsruhe, Germany (1999)* BA–14
- [16] Leith, D., Murray-smith, R., Leithhead, W.: Nonlinear structure identification: A gaussian process prior/velocity-based approach. In: *Proceedings of Control 2000, Cambridge, U.K. (2000)*
- [17] Murray-Smith, R., Girard, A.: Gaussian process priors with ARMA noise models. In: *Proceedings of the Irish Sgnals and Systems Conference, Maynooth, Ireland (2001)* 147–152
- [18] Girard, A., Rasmussen, C.E., Quinonero-Candela, J., Murray-Smith, R.: Gaussian process priors with uncertain inputs – application to multiple-step ahead time series forecasting. In Becker, S., Thrun, S., Obermayer, K., eds.: *Advances in Neural Information processing Systems 15*. (2003)
- [19] Solak, E., Murray-Smith, R., Leithhead, W.E., Leith, D.J., Rasmussen, C.E.: Derivative observations in Gaussian process models of dynamic systems. In Becker, S., Thrun, S., Obermayer, K., eds.: *Advances in Neural Information processing Systems 15*. (2003)
- [20] Neal, R.: Monte Carlo implementation of Gaussian process models for Bayesian regression and classification. Technical Report 9702, Department of Statistics, University of Toronto (1997)
- [21] Gibbs, M.: *Bayesian Gaussian processes for regression and classification*. PhD thesis, Cavendish Laboratory, University of Cambridge (1998)
- [22] MacKay, D.: Gaussian processes: A replacement for supervised neural networks? In: *Lectures notes for the NIPS 1997, Denver, Colorado (1997)*
- [23] Shi, J., Murray-Smith, R., Titterington, D.M.: Bayesian regression and classification using mixtures of multiple Gaussian processes. *International Journal of Adaptive Control and Signal Processing* **17** (2003) 149–161
- [24] O’Hagan, A.: Some Bayesian numerical analysis. In Bernardo, J., Berger, J., Dawid, A., Smith, F., eds.: *Bayesian Statistics 4*, Oxford University Press (2001) 345–363
- [25] Tipping, M.E.: Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research* **1** (2001) 211–244
- [26] Seeger, M., Williams, C., Lawrence, D.: Fast forward selection to speed up sparse Gaussian process regression. In: *Proceedings of the Ninth International Workshop on AI and Statistics*. (2003)