

# Gaussian Process Priors with ARMA Noise Models

*R. Murray-Smith and A. Girard*  
*Department of Computing Science*  
*University of Glasgow*  
*Glasgow G12 8QQ*  
{rod, agathe}@dcs.gla.ac.uk

## Abstract

We extend the standard covariance function used in the Gaussian Process prior nonparametric modelling approach to include correlated (ARMA) noise models. The improvement in performance is illustrated on some simulation examples of data generated by nonlinear static functions corrupted with additive ARMA noise.

## 1 Gaussian Process priors

In recent years many flexible parametric and semi-parametric approaches to empirical identification of nonlinear systems have been used. In this paper we use *nonparametric* models which retain the available data and perform inference conditional on the current state and local data (called ‘smoothing’ in some frameworks). This direct use of the data has potential advantages in many control contexts. The uncertainty of model predictions can be made dependent on local data density, and the model complexity is automatically related to the amount of available data (more complex models need more evidence to make them likely).

The nonparametric model used in this paper is a *Gaussian Process prior*, as developed by O’Hagan [1] and reviewed in [2, 3]. An application to modelling a system within a control context is described in [4], and further developments relating to their use in gain scheduling are described in [5]. Most previous published work has focused on regression tasks with independent identically distributed noise characteristics. Input-dependent noise is described in [6], but we are not aware of previous work with coloured noise covariance functions in Gaussian Process priors.

This paper shows how knowledge about correlation structure of additive unmeasured noise or disturbances can be incorporated into the model<sup>1</sup>. This improves the performance of the model in finding optimal parameters for describing the deterministic aspects of the system, and can be used to make online prediction more accurately. We expect this will make the use of Gaussian Process priors more attractive for use in control and signal processing contexts.

## 2 Modelling with GPs

We assume that we are modelling an unknown nonlinear system  $f(x)$ , with known inputs  $x$ , using observed outputs  $y$ . These have been corrupted by an additive discrete-time process  $\epsilon(t)$ . Here we assume that  $f(x^i)$  and  $\epsilon^i$  are independent. Let  $\mathbf{y} = [y^1, \dots, y^N]^T$ , a set of observed data or *targets* be such that

$$y^i = f(x^i) + \epsilon^i, \quad i = 1, \dots, n \quad (1)$$

### 2.1 The Gaussian Process prior approach

A prior is placed directly on the space of functions for modelling the above system. We assume that the values of the function  $f(x)$  at inputs  $x^1, \dots, x^n$ , outputs  $y^1, \dots, y^n$ , constitute a set of random variables which we assume will have a joint  $n$ -dimensional multivariate Normal distribution. The Gaussian Process is then fully specified by its mean<sup>2</sup> and covariance function  $C(x^i, x^j)$ . We note

$$(y^1, \dots, y^n)^T \sim \mathcal{N}(0, \Sigma), \quad (2)$$

where  $\Sigma$  is the covariance matrix whose entries  $\Sigma^{ij}$  are given by  $C(x^i, x^j)$ . We now have a prior distribution for the target values which is a multivariate Normal:

$$p(\mathbf{y}|\mathbf{x}) = (2\pi)^{-\frac{n}{2}} |\Sigma|^{-\frac{1}{2}} \exp \left[ -\frac{1}{2} \mathbf{y}^T \Sigma^{-1} \mathbf{y} \right], \quad (3)$$

---

<sup>1</sup>See a standard text such as [7] for a discussion of disturbance models in the linear system identification context.

<sup>2</sup>In what follows, we assume a zero mean process.

**Useful notations** What follows is more straightforward if we partition the full set of points into a training and a test part. Let  $\mathbf{y}_1 = [y^1, \dots, y^N]^T$ ,  $\boldsymbol{\epsilon}_1 = [\epsilon^1, \dots, \epsilon^N]^T$  and  $\mathbf{x}_1 = [x^1, \dots, x^N]^T$  the sets of training outputs, disturbances and inputs respectively.  $\mathbf{y}_2, \mathbf{x}_2$  are the corresponding terms for the test data. In this paper we will consider single point prediction, and use the notation  $y^{N+1} = \mathbf{y}_2$ , to indicate that we are predicting the  $N+1$ th point, given the  $N$  training data, and the new input  $x^{N+1}$ . We then have

$$\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} \sim \mathcal{N}(0, \Sigma), \text{ with covariance matrix } \Sigma \text{ partitioned into } \Sigma = \begin{bmatrix} \Sigma_1 & \Sigma_{12} \\ \Sigma_{21} & \Sigma_2 \end{bmatrix}, \quad (4)$$

where  $\Sigma_{21} = \Sigma_{12}^T$ , and

- $\Sigma_1^{ij} = C_f(x_1^i, x_1^j) + C_n(\epsilon_1^i, \epsilon_1^j)$  are the covariances between the training data (matrix  $N \times N$ ),
- $\Sigma_{12}^j = \Sigma_{21}^j = C_f(x^{N+1}, x_1^j) + C_n(\epsilon^{N+1}, \epsilon_1^j)$  is the vector ( $N \times 1$ ) of covariances between the test and the training targets and
- $\Sigma_2 = C_f(x^{N+1}, x^{N+1}) + C_n(\epsilon^{N+1}, \epsilon^{N+1})$  is the variance of the test point.

We can view the joint probability as the combination of a marginal multinormal distribution and a conditional multinormal distribution. The marginal term gives us the likelihood of the training data

$$p(\mathbf{y}_1 | \mathbf{x}_1) = (2\pi)^{-\frac{N}{2}} |\Sigma|^{-\frac{1}{2}} \exp \left[ -\frac{1}{2} \mathbf{y}_1^T \Sigma^{-1} \mathbf{y}_1 \right], \quad (5)$$

while the conditional term gives us the output posterior density conditional on the training data at the test points  $\mathbf{x}_2$

$$p(y^{N+1} | x^{N+1}, \mathbf{x}_1, \mathbf{y}_1) = (2\pi)^{-\frac{N_2}{2}} |\Sigma_{2.1}|^{-\frac{1}{2}} \exp \left[ -\frac{1}{2} (y^{N+1} - \mu)^T \Sigma_{2.1}^{-1} (y^{N+1} - \mu) \right], \quad (6)$$

where

$$\mu = \Sigma_{12}^T \Sigma_1^{-1} \mathbf{y}_1 \quad (7)$$

$$\Sigma_{2.1} = \Sigma_2 - \Sigma_{12}^T \Sigma_1^{-1} \Sigma_{21}. \quad (8)$$

We use  $\mu$  as a mean estimate  $\hat{y}^{N+1}$  for the test point, with a variance of  $\Sigma_{2.1}$ . Note that the inversion of an  $N \times N$  matrix is computationally nontrivial for  $N > 1000$ , so the GP approach is currently suited to small and medium-sized data sets.<sup>3</sup>

## 3 The covariance function

The covariance function has a central role in the GP modelling framework, expressing the expected covariance between two outputs  $y_i$  and  $y_j$ . The covariance function is constrained to lead to a positive definite covariance matrix for any inputs  $x$ . We view the total covariance as being composed of covariance functions  $C_f()$  due to the underlying system model  $f(x)$  and  $C_n()$  due to the noise process  $\epsilon$ .

$$C(x^i, x^j) = C_f(x^i, x^j) + C_n(i, j) \quad (9)$$

### 3.1 The ‘model’ covariance function $C_f()$

The covariance function associated with the ‘model’ of the system,  $C_f$ , is a function of the inputs  $x$  only. We choose this covariance to be such that inputs ‘close’ together will have outputs that are highly correlated and thus are likely to be quite similar. It corresponds to a prior giving higher probability to functions  $f(x)$  which are smoothly varying in  $x$ .

A commonly-used covariance function which assumes smoothness is

$$C_f(x^i, x^j) = v_0 \exp \left[ -\frac{1}{2} \sum_{d=1}^D w_d (x_d^i - x_d^j)^2 \right] + a_0, \quad (10)$$

in which

- $v_0$  controls the vertical scale of variation of a typical function,
- $a_0$  allows the function to be offset away from 0 and
- $w_d$  allow different distance measure for each input dimension.

In practice, a small ‘jitter’,  $J\delta_{ij}$  is added to  $C_f()$ , for numerical reasons (this adds a small diagonal term to  $\Sigma$  improving the condition of the matrix). The hyperparameters  $\Theta_f = (w, v_0, a_0)^T$  can be provided as prior knowledge, or identified from training data.

<sup>3</sup>See [8] for discussion of ways to speed up the inversion.

## 3.2 The ‘noise’ covariance function $C_n()$

### 3.2.1 Uncorrelated white noise

The simplest choice of noise model is to assume a Gaussian white noise:  $\epsilon \sim \mathcal{N}(0, \Sigma_N)$ . In that case, the covariance function is simply

$$C_n(\epsilon^i, \epsilon^j) = \sigma_\epsilon^2 \delta^{ij}. \quad (11)$$

so the noise covariance matrix when predicting a single future point, from  $n$  training data is  $\Sigma_N = \sigma_\epsilon^2 \mathbf{I}$  with  $\mathbf{I}$  the  $N+1 \times N+1$  identity matrix.

### 3.2.2 Parametric correlated-noise models

We assume that correlations between the disturbances on individual outputs exist,  $\epsilon(t)$  is a coloured noise/disturbance process.  $\epsilon^i$ , the disturbance associated with the point  $y^i$  occurs at time  $t_i$ , and depends on previous values of  $\epsilon$ . We consider two dynamic linear models: the auto regressive (AR) and moving average (MA) models separately, and their combination into an ARMA disturbance model. For such models, the partitioned components of the noise covariance matrix  $\Sigma_N$  are

$$\begin{cases} \Sigma_{N1} &= C_n(\epsilon(t), \epsilon(t+\tau)) \text{ for } t=1, \dots, N \text{ and } \tau=0, \dots, N \\ \Sigma_{N12} &= C_n(\epsilon(t^{N+1}), \epsilon(t+\tau)) \text{ for } t=1, \dots, N \text{ and } \tau=0, \dots, N \\ \Sigma_{N2} &= C_n(\epsilon(t^{N+1}), \epsilon(t^{N+1})) \end{cases} \quad (12)$$

where  $\tau$  is the lag. Example plots of such noise covariance matrices are shown in Figure 1.

#### AR noise model

The auto regressive model of order  $n_a$ ,  $\text{AR}(n_a)$ , can be written

$$\epsilon(t) = e(t) - a_1 \epsilon(t-1) - \dots - a_{n_a} \epsilon(t-n_a), \quad (13)$$

where  $e$  is a Gaussian white noise with variance  $\sigma_e^2$ . In transfer operator notation, where  $q^{-1}$  is the delay function,

$$\epsilon(t) = \frac{1}{A(q)} e(t), \quad (14)$$

where  $A(q) = 1 + a_1 q^{-1} + \dots + a_{n_a} q^{-n_a}$ . The covariance function is

$$C_n(\tau) = \begin{cases} a_1 C_n(\tau-1) + a_2 C_n(\tau-2) + \dots + a_p C_n(\tau-n_a) & \text{for } \tau > 0 \\ a_1 C_n(1) + a_2 C_n(2) + \dots + a_{n_a} C_n(n_a) + \sigma_e^2 & \text{for } \tau = 0 \end{cases} \quad (15)$$

The first  $n_a$  elements of the covariance function  $C_n(1..n_a)$  are estimated by solving the Yule-Walker equations. The rest by applying the AR process iteratively.

If identifying the parameters of an AR model of order  $n_a$ , there are then  $(n_a + 1)$  parameters to estimate. Note that for the process to be stationary, the roots of  $A(q) = 0$  must lie outside the unit circle, so the optimization process must be constrained.

#### MA noise model

The  $\text{MA}(n_b)$  model

$$\epsilon(t) = e(t) + b_1 e(t-1) + \dots + b_{n_b} e(t-n_b) = B(q)e(t), \quad (16)$$

has the following covariance function

$$C_n(\tau) = \sigma_e^2 \begin{cases} 1 + b_1^2 + b_2^2 + \dots + b_{n_b}^2 & \text{for } \tau = 0 \\ b_{|\tau|} + b_{1+|\tau|} b_1 + b_{2+|\tau|} b_2 + \dots + b_{n_b-|\tau|} b_{n_b} & \text{for } 1 \leq |\tau| \leq n_b \end{cases} \quad (17)$$

In that case, we see the covariance is zero beyond the order of the model (this leads to a band-diagonal noise covariance matrix).

#### ARMA noise model

The ARMA noise model is a combination of the above

$$\epsilon(t) = e(t) + b_1 e(t-1) + \dots + b_{n_b} e(t-n_b) - a_1 \epsilon(t-1) - \dots - a_{n_a} \epsilon(t-n_a) = \frac{B(q)}{A(q)} e(t), \quad (18)$$

### 3.3 Training: learning the hyperparameters

We have a parametric form for the covariance functions, depending on a set of hyperparameters  $\theta$ . If we take a maximum a posteriori approach, for a new modelling task we will need to learn these hyperparameters from the training data. This is done by maximising the likelihood of the training data (equation (5)). We used a standard conjugate gradient algorithm.<sup>4</sup>

In this paper we optimise the hyperparameters related to the model, but assume prior knowledge of the noise covariance function hyperparameters.

## 4 Experiments

We wish to test experimentally whether the extended model improves performance by testing its behaviour on simulated identification data. We can measure how well we predict the underlying nonlinear function  $f(x)$ , and also how well we are able to simulate the complete process  $f(x) + \epsilon$ , and make one-step-ahead predictions based on recent observations of the system output. We also show how the same model can make  $k$ -step-ahead predictions.

We illustrate the use of Gaussian process priors with correlated noise models using the simple one-dimensional non linear function  $y = \tanh(x)$ ,  $x \in [-5, 5]$ . This is a one-dimensional input space, but the approach is straightforward to use for multiple inputs. We consider the same  $x$ -range for both the training and the test sets. In the training set, the *states*  $x$  have been randomly chosen from a uniform distribution over  $[-5, 5]$ . We train our model using a small set of  $n = 20$  points to highlight the advantages of improved prior knowledge of noise characteristics. We simulate the trained model on 101 points.

The coloured noise is created by filtering a vector of points sampled from a normal distribution with variance  $\sigma_e^2 = 0.05$ . Figure 1 shows the noise covariance matrices associated with the following noise models:

- AR(2):  $\epsilon(t) = e(t) - 0.7\epsilon(t-1) - 0.1\epsilon(t-2)$
- MA(2):  $\epsilon(t) = e(t) + \frac{1}{2}e(t-1) + \frac{1}{3}e(t-2)$ .
- ARMA(2):  $\epsilon(t) = e(t) - 0.7\epsilon(t-1) - 0.1\epsilon(t-2) + \frac{1}{2}e(t-1) + \frac{1}{3}e(t-2)$ .

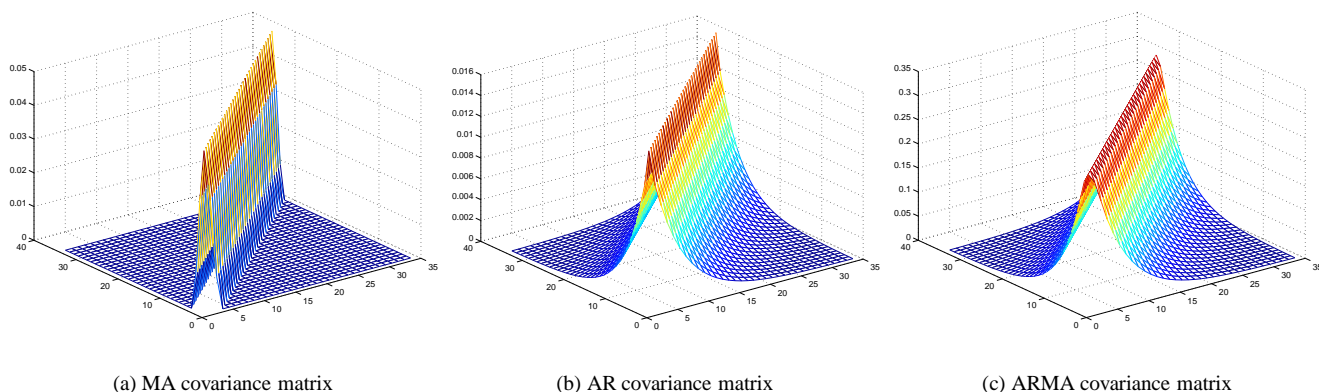


Figure 1: Covariance structures for the noise component. Axes show ordered, evenly sampled time steps.

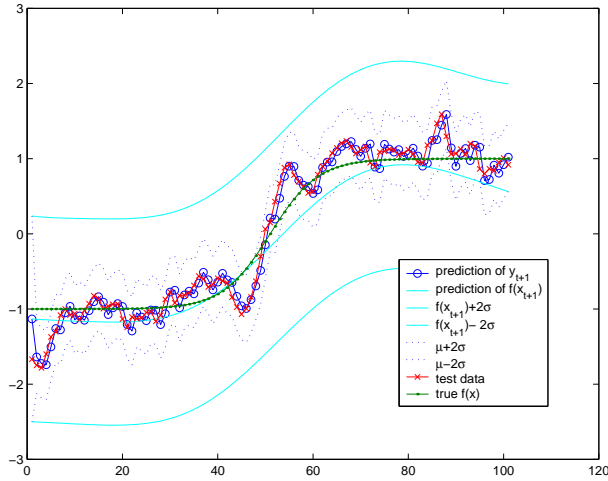
### 4.1 Results

Figure 2 shows the experimental results. In terms of fit to 'true' model  $f(x)$ , the GP with ARMA noise has a r.m.s.e. of 0.1955, compared to the r.m.s.e. of 0.3005 for the GP with white noise. The comparison is plotted in Figure 2(c). We can see that the more complete model of the covariance between data due to the noise process in the GP with ARMA has improved our fit to the underlying nonlinear system, compared to the white noise version.

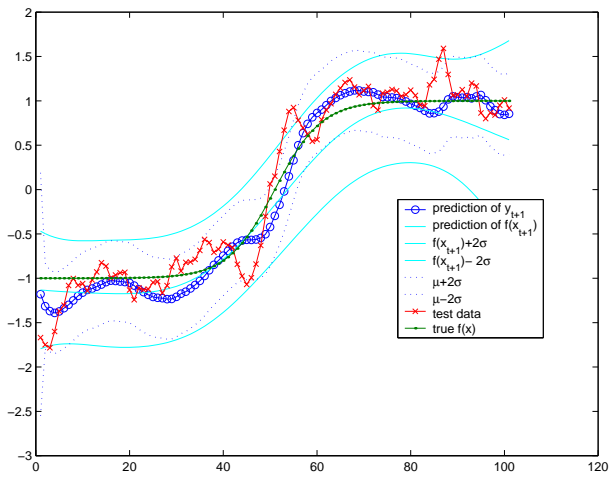
The r.m.s.e. in terms of fit to the test data in a one-step-ahead prediction is 0.1299 for the ARMA model and 0.2606 for the white noise model. This shows that the recent data can help predictions significantly, if the noise model is used.

The  $k$ -step-ahead prediction is a prediction of individual outputs at time  $t$ , where the information about the output of the system after some time  $t_{end}$  is not known. Inputs  $x$  are assumed known at all times. Figure 2(d) shows how the predicted  $y$  values gradually return to the mean prediction, as the information about the current state of the system becomes more dated (recent  $y$ 's are used for prediction until  $t_{end} = 65$ ). Note also the gradual increase in prediction uncertainty over prediction horizon, until it reaches the uncertainty of predictions with no knowledge of recent  $y$ .

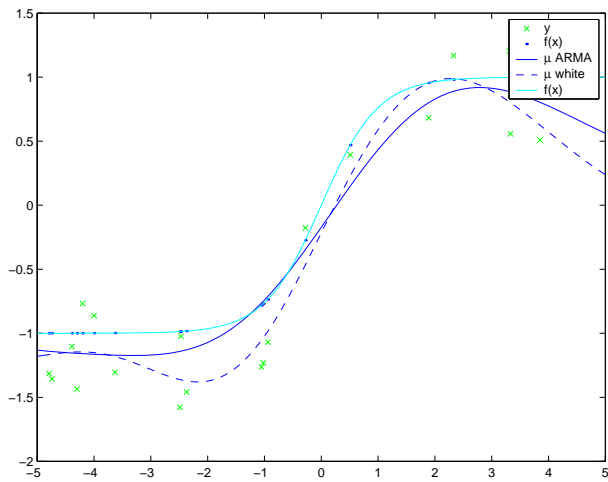
<sup>4</sup>Ideally, for a full Bayesian treatment we should give prior distributions to the hyperparameters and base predictions on a sample of values from their posterior distribution, as an approximation to integration. Here, we do not use *hyperpriors* but initialize these hyperparameters.



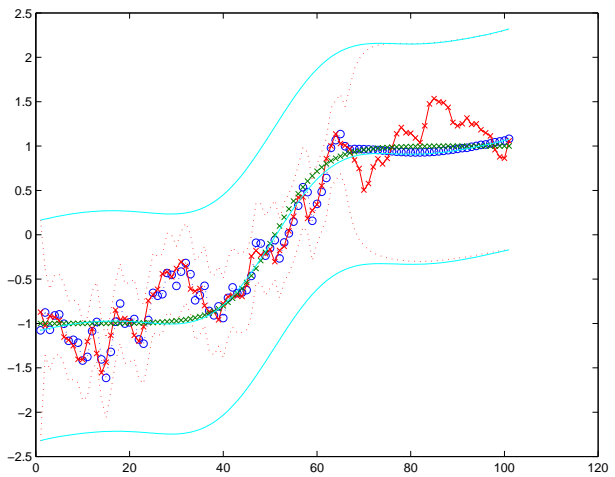
(a) GP with ARMA noise one-step-ahead prediction results. Noise model allows better prediction of test data.



(b) GP with white noise prediction results. Here, without being able to correlate recent errors to predict future ones, the accuracy of prediction is degraded. Note also the unrealistically tight confidence bands.



(c) Comparison between GP with ARMA noise and white noise for mean prediction of  $f(x)$ . Use of noise model improves accuracy of mean fit to underlying system  $f(x)$ . Note small number of training points ( $n = 20$ ).



(d) GP with ARMA noise  $k$ -step-ahead prediction results from  $t = 65$ . Note how predictions regress to mean, as information about  $y$  becomes dated. Also note the gradual increase in uncertainty to the limit of no recent information.

Figure 2: Results for a training set of 20 points for a  $\tanh()$  nonlinearity and an additive ARMA process, with covariance shown in Figure 1

## 5 Conclusions

We have illustrated the use of ARMA coloured noise models for improving the accuracy of modelling nonlinear systems of the form  $y(t) = f(x(t)) + \epsilon(t)$  using Gaussian Process priors. The examples used in this paper assumed full prior knowledge of the parameters of the noise process for clarity of presentation. Given such prior knowledge, the expected improvements were found in simulated examples.

It is possible to optimise both model parameters and noise parameters simultaneously, and there is scope for much interesting future work in this area, and the best combination of prior knowledge of model order, and parameters from physical insight into the system with parameter optimisation will depend on the particular application.

Further extensions would be to go for a more Bayesian approach, place priors on the hyperparameters, and make use of

numerical integration techniques such as Markov-Chain Monte Carlo (MCMC) to integrate over the parameters, rather than maximising the likelihood.

## 6 Acknowledgements

The authors gratefully acknowledge the support of the *Multi-Agent Control* Research Training Network – EC TMR grant HPRN-CT-1999-00107<sup>5</sup>, and EPSRC grant *Modern statistical approaches to off-equilibrium modelling for nonlinear system control* GR/M76379/01. This work leads on from research done while at the Department of Mathematical Modelling, Technical University of Denmark, supported by Marie Curie TMR grant FMBICT961369. We would also like to thank Carl Rasmussen for providing stimulating input on the topics discussed in this paper.

## References

- [1] A. O’Hagan, “On curve fitting and optimal design for regression,” *Journal of the Royal Statistical Society B*, vol. 40, pp. 1–42, 1978.
- [2] C. K. I. Williams, “Prediction with Gaussian processes: From linear regression to linear prediction and beyond,” in *Learning and Inference in Graphical Models* (M. I. Jordan, ed.), Kluwer, 1998.
- [3] C. E. Rasmussen, *Evaluation of Gaussian Processes and other Methods for Non-Linear Regression*. PhD thesis, Graduate department of Computer Science, University of Toronto, 1996.
- [4] R. Murray-Smith, T. A. Johansen, and R. Shorten, “On transient dynamics, off-equilibrium behaviour and identification in blended multiple model structures,” in *European Control Conference, Karlsruhe, 1999*, pp. BA–14, 1999.
- [5] D. J. Leith, R. Murray-Smith, and W. Leithead, “Nonlinear structure identification: A Gaussian Process prior/Velocity-based approach,” in *Control 2000, Cambridge*, 2000.
- [6] P. W. Goldberg, C. K. I. Williams, and C. M. Bishop, “Regression with input-dependent noise,” in *Advances in Neural Information Processing Systems 10* (M. J. K. M. I. Jordan and S. A. Solla, eds.), Lawrence Erlbaum, 1998.
- [7] L. Ljung, *System Identification — Theory for the User*. Englewood Cliffs, New Jersey, USA: Prentice Hall, 1987.
- [8] M. N. Gibbs, *Bayesian Gaussian Processes for Regression and Classification*. PhD thesis, University of Cambridge, 1997.

---

<sup>5</sup>This work is the sole responsibility of the authors, and does not reflect the European Community’s opinion. The Community is not responsible for any use that might be made of data appearing in this publication.