Multivariable Generalized Minimum Variance Control Based on Artificial Neural Networks and Gaussian Process Models

Daniel Sbarbaro[1]*, Roderick Murray-Smith[2], and Arturo Valdes[1]

[1] Department of Electrical Engineering, Universidad de Concepcion, Chile
{dsbarbar, avald}@die.udec.cl
[2] Department of Computer Science, University of Glasgow, Glasgow, U.K. &
Hamilton Institute, NUI Maynooth, Ireland.
rod@dcs.gla.ac.uk

Abstract. The control of an unknown multivariable nonlinear process represents a challenging problem. Model based approaches, like Generalized Minimum Variance, provide a flexible framework for addressing the main issues arising in the control of complex nonlinear systems. However, the final performance will depend heavily on the models representing the system. This work presents a comparative analysis of two modelling approaches for nonlinear systems, namely Artificial Neural Network (ANN) and Gaussian processes. Their advantages and disadvantages as building blocks of a GMV controller are illustrated by simulation.

1 Introduction

The control of a nonlinear multivariable system is a difficult task due to the number of variables involved and the complex coupling among inputs and outputs. This problem is even worse if the model of the system is unknown.

The use of a flexible modelling technique, such as artificial neural networks (ANN), for controlling Multivariable Nonlinear systems has been addressed from a theoretical point of view and illustrated by simulation in [4]. A design methodology based on the idea of a model reference and ANN is also developed. Generalized Minimum Variance control approach is a versatile and very popular formalism for designing linear self-tuning controllers [1], Gawthrop has also shown that model reference and self-tuning controllers are closely related. The GMV controllers can be enhanced to deal with nonlinear systems if nonlinear models are considered. ANN models, in this context, were first proposed by [2]. In [3] an extension to a multivariable system is presented, but in this case a linear model extended by a ANN was considered. A typical assumption used in these approaches is the "Certainty Equivalence principle" under which the controller is designed in terms of the identified model as if the model were the real process.

There has been an increase of interest in the use of Gaussian Process priors as alternative to ANN [9,5,8]. Mackay describes in his seminal paper the similarity

* This work was supported by Fondecyt project 1040486

F. Yin, J. Wang, and C. Guo (Eds.): ISNN 2004, LNCS 3174, pp. 52–58, 2004.

and differences of both approaches from a statistical perspective [5]. In [6] a MV adaptive controller for single-input single-output system based on GP is presented. One of the main advantages of this modelling approach is that provides a direct estimation of the output variance, which is normally ignored by other methods. This is a relevant information for control design, since it allows the design of self-tuning controllers without resorting to the Certainty Equivalence principle.

Without losing generality, this work considers the MIMO systems as multiloop systems, consisting of several SISO systems where coupling effects are considered as disturbances.

This paper is organized as follows: section 2 describes the GMV approach for control design. Section 3 presents briefly models used to approximate the system and the resulting expression for GMV controllers. Section 4 illustrates by performing simulations, the main characteristics of both modelling approaches. Some final remarks are given in section 5.

2 Multivariable Generalized Minimum Variance Controller

Let us consider the Multivariable nonlinear system described by the following affine in the control structure:

$$\mathbf{Y}(k+1) = F(\mathbf{X}(k)) + G(\mathbf{X}(k))\mathbf{U}(k) + \Xi(k) \tag{1}$$

where Ξ is a vector of Gaussian zero mean random noise, $\mathbf{Y}(k) = [y_1(k) \dots y_n(k)]^T$, $\mathbf{U}(k) = [u_1(k) \dots u_m(k)]^T$, n the number of outputs and m the number of inputs. The vector $\mathbf{X}$ represents a vector with all the past information as described by:

$$\mathbf{X}^T(k) = [\mathbf{Y}(k)^T \mathbf{Y}(k-1)^T \dots \mathbf{Y}(k-r)^T \mathbf{U}(k-1)^T \mathbf{U}(k-2)^T \dots \mathbf{U}(k-s)^T] \tag{2}$$

The integers r and s are the number of delayed inputs and outputs respectively necessary for describing the system. As pointed out in [2] the affinity property represents more an advantage for designing the controller, rather than a modelling constraint. In the generalized minimum variance control approach, the performance index is:

$$J = E\{\mathbf{E}(k+1)^T \mathbf{Q}\mathbf{E}(k+1)\} + \mathbf{U}(k)^T \mathbf{R}(q^{-1})\mathbf{U}(k) \tag{3}$$

where $\mathbf{E}(k+1) = \mathbf{Y}_d(k+1) - \mathbf{Y}(k+1)$, the matrix $\mathbf{Q}$ is definite positive matrix and $\mathbf{R}$ is polynomial matrix in the backward shift operator q^{-1} . The expected value of the quadratic error term can be expanded in terms of the variance and the mean values as follows:

$$\begin{aligned} E\{\mathbf{E}(k+1)^T \mathbf{Q}\mathbf{E}(k+1)\} = \\ (\mathbf{Y}_d(k+1) - \mu_Y(k+1))^T \mathbf{Q}(\mathbf{Y}_d(k+1) - \mu_Y(k+1)) \\ + \sum_{j=1}^{n} q_{jj} var\{y_j(k)\} + \sum_{i=1}^{n}\sum_{j=1}^{n} q_{ij} covar\{y_i(k) y_j(k)\} \end{aligned} \tag{4}$$

where μ_Y represents a vector having the mean value of each output; i.e. $\mu_Y = [\mu_{y_1} \dots \mu_{y_n}]^T$.

Without losing generality we will consider $\mathbf{Q} = diag[q_1 \dots q_n]$ and $\mathbf{R}(q^{-1}) = diag[R_1(q^{-1}) \dots R_n(q^{-1})]$. The necessary condition for $u_i(k)$ being a minimum of the cost function 3 is:

$$\mathbf{E}(k+1)^T \mathbf{Q} \frac{\partial \mathbf{E}(k+1)}{\partial u_i(k)} + R_i(q^{-1}) u_i(k) + \sum_{j=1}^{n} q_{jj} \frac{\partial var\{y_j(k)\}}{\partial u_i(k)} = 0. \tag{5}$$

3 Modelling Approaches

3.1 ANN Models

Every output of the nonlinear system is represented by a set of ANNs structured as follows:

$$y_j(k+1) = NN_{j,0}(k) + \sum_{l=1}^{m} NN_{j,l}(k) u_l(k) \tag{6}$$

where

$$NN_{j,l}(k) = \mathbf{W}_{j,l} \sigma(\mathbf{V}_{j,l} \mathbf{X}(k)). \tag{7}$$

The uncertainty associated with each output can be estimated by several standard confidence bound estimation algorithms [7], [10] which normally gave satisfactory results. However, the size of the estimated confidence interval could be biased [11]. A more important issue that hamper the application of these estimates in a control algorithm is the fact that they depends on a Jacobian matrix in a very complex form, making impossible their use in the control algorithm. In addition, they are 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. Thus, we will consider the outputs of the ANN as if they were the real output; i.e. we will apply the "certainty equivalence principle". In this way, equation (5):

$$\sum_{j=1}^{n} q_{jj} \left[y_j^d(k+1) - NN_{j,0}(k) + \sum_{l=1}^{m} NN_{j,l}(k) u_l(k) \right] NN_{j,i} + R_i(q^{-1}) u_i(k) = 0 \tag{8}$$

From this set of equations we can obtain the control as follows:

$$\mathbf{U}(k) = \mathbf{M}(k)^{-1} \begin{bmatrix} \sum_{j=1}^{n} q_{jj} \left[y_j^d(k+1) - NN_{j,0}(k) \right] NN_{j,i}(k) - \bar{R}_1(q^{-1}) u_1(k) \\ \vdots \\ \sum_{j=1}^{n} q_{jj} \left[y_j^d(k+1) - NN_{j,0}(k) \right] NN_{j,m}(k) - \bar{R}_m(q^{-1}) u_m(k) \end{bmatrix} \tag{9}$$

$$\mathbf{M}(k) = \begin{bmatrix} r_{10} + \sum_{j=1}^{m} q_{jj} NN_{j,1}(k)^2 & \dots & \sum_{j=1}^{m} q_{jj} NN_{j,m}(k) NN_{j,1}(k) \\ & \vdots & \\ \sum_{j=1}^{m} q_{jj} NN_{j,1}(k) NN_{j,m}(k) & \dots & r_{m0} + \sum_{j=1}^{m} q_{jj} NN_{j,m}(k)^2 \end{bmatrix} \tag{10}$$

where $\bar{R}_i(q^{-1}) = r_{i0} + r_{i1}q^{-1}...$ is a polynomial of order p.

3.2 GP Models

Let us consider the jth output of the system be described as:

$$y_j(k+1) = F_j(\phi(k)) + \epsilon_i(k) \tag{11}$$

where $\phi(k)^T = [\mathbf{X}^T(k) \;\; \mathbf{U}^T(k)]^T$. Instead of parameterising the system (1) as a parametric model, we can place a prior directly on the space of functions where F_j 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. Let the input and output sequence be stacked in the following matrix Φ_N and vector $\mathbf{y}_N$, respectively. We will assume zero mean, so for the case with partitioned data $\mathbf{y}_N$ and $y(k+1)$ we will have the multivariate Normal distribution,

$$\begin{bmatrix} \mathbf{y}_N \\ y(k+1) \end{bmatrix} \sim \mathcal{N}\left(0, \mathbf{C}_{N+1}\right), \quad \mathbf{C}_{N+1} = \begin{bmatrix} \mathbf{C}_N & \mathbf{K} \\ \mathbf{K}^T & \kappa \end{bmatrix}. \tag{12}$$

where $\mathbf{C}_{N+1}$ is the full covariance matrix. Like the Gaussian distribution, the Gaussian Process is fully specified by a mean and its covariance function, so we denote the distribution $GP(\mu, C)$. The covariance function $C(\phi(k), \phi(l))$ expresses the expected covariance between $y(k+1)$ and $y(l+1)$. We can therefore, infer $y(l+1)$'s from the N data pairs Φ_N and $\mathbf{y}_N$'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,

$$P(\mathbf{y}_N|\Phi_N) = (2\pi)^{-\frac{N}{2}} |\mathbf{C}_N|^{-\frac{1}{2}} e^{\left(-\frac{1}{2}\mathbf{y}_N^T \mathbf{C}_N^{-1} \mathbf{y}_N\right)}. \tag{13}$$

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 $\Phi_N, \mathbf{y}_N$ and the test points $\phi(k)$,

$$P(y(k+1)|\Phi_N, \mathbf{y}_N, \phi(k)) = \frac{P(y(k+1), \mathbf{y}_N)}{P(\mathbf{y}_N)} = \frac{1}{(2\pi\hat{\sigma}_y^2)^{\frac{1}{2}}} e^{-\frac{(y(k+1)-\hat{\mu}_y)^2}{2\hat{\sigma}_y^2}},$$

where, as in the straightforward multinormal case described earlier,

$$\hat{\mu}_{y_i} = \mathbf{K}^T \mathbf{C}_N^{-1} \mathbf{y}_N \qquad var\{y_i\} = \hat{\sigma}_y^2 = \kappa - \mathbf{K}^T \mathbf{C}_N^{-1} \mathbf{K}, \tag{14}$$

so we can use $\hat{\mu}(\phi(k))$ as the expected model output, with a variance of $\hat{\sigma}(\phi(k))^2$. These expressions can be simplified [6] to obtain:

$$\hat{\mu}_{y_j} = \omega_{j,0} + \sum_{l=1}^{m} \omega_{j,l} u_l(k) = \omega_{j,0} + \Omega_j^T \mathbf{U}(k) \tag{15}$$

$$var\{y_j\} = \sigma_{j,0} + \sum_{l=1}^{m} \gamma_{j,l} u(l) + \sum_{i=1}^{m}\sum_{l=1}^{m} \sigma_{j,il} u_l(k) u_i(k) \tag{16}$$

$$= \sigma_{j,0} + \mathbf{U}(k)^T \Gamma_j + \mathbf{U}(k)^T \Sigma_j \mathbf{U}(k). \tag{17}$$

where $\sigma_{j,0}, \omega_{j,0}, \Gamma_j$, and Σ_j are all functions of time depending on the data gathered up to time k and the set of parameters associated to the covariance functions. Thus, the control signal is obtained as:

$$\mathbf{U}(k) = \mathbf{M}(k)^{-1} \begin{bmatrix} \sum_{j=1}^{n} q_{jj} \left[y_j^d(k+1) - \omega_{j,0}(k) \right] \omega_{j,1}(k) + \gamma_{j,1} - \bar{R}_1(q^{-1}) u_1(k) \\ \vdots \\ \sum_{j=1}^{n} q_{jj} \left[y_j^d(k+1) - \omega_{j,0}(k) \right] \omega_{j,m}(k) + \gamma_{j,m} - \bar{R}_m(q^{-1}) u_m(k) \end{bmatrix} \tag{18}$$

$$\mathbf{M}(k) = \begin{bmatrix} r_{10} + \sum_{j=1}^{m} q_{jj} \omega_{j,1}(k)^2 + \sigma_{j,11} & \ldots & \sum_{j=1}^{m} q_{jj} \omega_{j,m}(k) \omega_{j,1}(k) + \sigma_{j,1m} \\ \vdots & & \\ \sum_{j=1}^{m} q_{jj} \omega_{j,1}(k) \omega_{j,m}(k) + \sigma_{j,m1} & \ldots & r_{m0} + \sum_{j=1}^{m} q_{jj} \omega_{j,m}(k)^2 + \sigma_{j,mm} \end{bmatrix} \tag{19}$$

Notice that equations (18) and (19) have additional terms compared to (9) and (10). As pointed out in [6] these terms provide extra robustness in the face of uncertainty in the output estimation.

4 Simulations

In order to illustrate the main characteristics of both models, the following multivariable nonlinear system was considered:

$$y_1(k) = \frac{0.7 y_1(k-1) y_1(k-2)}{1 + y_1(k-1)^2 + y_2(k-2)^2} + \frac{10^{-4} u_2(k-1)}{1 + 3 y_1(k-2)^2 + y_2(k-1)^2} + u_1(k-1) + .25 u_1(k-2) + 0.5 u_2(k-2) \tag{20}$$

$$y_2(k) = \frac{0.5 y_2(k-1) sin(y_2(k-2))}{1 + y_2(k-1)^2 + y_1(k-2)^2} + 0.5 u_2(k-2) + .3 u_1(k-2) + +u_2(k-1)(0.1 u_2(k-2) - 1.5) \tag{21}$$

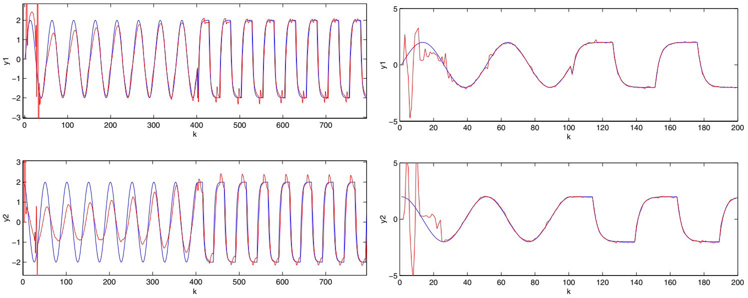
The nonlinear functions were approximated by four ANNs with 8 units in the hidden layer and the following structure :

$$y_i(k) = NN_{i,0}(k-1) + NN_{i,1}(k-1) u_i(k-1) \; i = 1, 2 \tag{22}$$

where

$$NN_{j,l}(k) = \mathbf{W}_{j,l} \tanh(\mathbf{V}_{j,l}\mathbf{X}(k)) \qquad l = 0, 1 \qquad j = 1, 2 \tag{23}$$

and $\mathbf{X} = [y_1(k)\ y_1(k-1)\ y_2(k)\ y_2(k-1)\ u_2(k-1)\ u_1(k-1)]^T$. The initial parameters were all set to small positive random numbers. The algorithm for updating the parameters was a standard backpropagation algorithm. The weighting polynomial associated with the control signal was selected as: $R(q^{-1}) = \lambda(1 - q^{-1})$; i.e. weights the control deviations, the parameter was $\lambda = .05$. This model structure is suitable for on-line learning, as shown in the simulations. The polynomial $R(q^{-1})$ introduces a certain degree of regularization on the control signal, avoiding extremely large excursions of the system output. Nevertheless, we can still see in figure 1(a) some large values. The convergence is quite slow and after 800 sampling steps the controller does not quite compensate for the coupling between the systems.



(a) ANN based controller

(b) GP-based controller

Fig. 1. Simulation results for both models

The GP model considered two covariance function with the following structure for each output:

$$\begin{aligned}\mathbf{C}_1(\phi_1(i), \phi_1(j)) &= C(\mathbf{X}(i), \mathbf{X}(j); \Theta_{1,0}) + u_1(i)C(\mathbf{X}(i), \mathbf{X}(j); \Theta_{1,1})u_1(j) \quad (24)\\ \mathbf{C}_2(\phi_2(i), \phi_2(j)) &= C(\mathbf{X}(i), \mathbf{X}(j); \Theta_{2,0}) + u_2(i)C(\mathbf{X}(i), \mathbf{X}(j); \Theta_{2,1})u_2(j)\end{aligned}$$

where $\phi_1(k) = [y_1(k)\ y_1(k-1)\ y_2(k)\ y_2(k-1)\ u_2(k-1)\ u_1(k-1)\ u_1(k)]^T$, $\phi_2(k) = [y_1(k)\ y_1(k-1)\ y_2(k)\ y_2(k-1)\ u_2(k-1)\ u_1(k-1)\ u_2(k)]^T$, and

$$C(\mathbf{X}(i), \mathbf{X}(j); \Theta_{lz}) = v_{lz} e^{-\frac{1}{2}\sum_{k=1}^{p} \alpha_{k,lz}(x_k(i) - x_k(j))^2} + a_{lz}. \tag{25}$$

The parameters of the covariance function were adjusted by maximizing a likelihood function two times during the simulation, at time $k = 15$ and $k = 25$. In addition, the first ten samples were used to do an off-line identification and after that the control loop was closed. Figure 1(b) shows these initial steps. It is worth pointing out that, in this case, with very few observations, the controller is able to handle the system quite well. It can also be noticed that there are no abrupt changes when the controller enters in operation; this is due to the extra terms that provide a regularization of the control signal. As seen in figure 1(b) the long run behaviour of the controller is also quite acceptable.

5 Conclusions

The comparative study carried out in this paper shows that GP models are suitable models to be used as building blocks of a GMV controller. The use of ANN, in this context, is limited since it is not possible to have a simple expression of the uncertainty bounds associated with the output. The use of the weighting polynomial plays an important role in smoothing the control action in both approaches.

References

1. Clarke, D.W., Gawthrop, P.G.: Self-tuning Controller. IEE Proc. **122**(9) (1975) 929-934
2. Bittanti, S., Piroddi, L.: Neural Implementation of GMV Control Schemes Based on Affine Input/output Models. IEE Proc.-Control Theory Appl. **144**(6) (1997) 521–530
3. Zhu, Q., Ma, Z., Warwick, K.: Neural Network Enhanced Generalised Minimum Variance Self-tuning Controller for Nonlinear Discrete-Time Systems. IEE Proc.-Control Theory Appl. **146**(4) (1999) 319–326
4. Narendra, K., Mukhopadhyay, S.: Adaptive Control of Nonlinear Multivariable Systems Using Neural Networks. Neural Networks **7** (1994) 737–752
5. MacKay, D.: Gaussian Processes - A Replacement for Supervised Neural Networks?. Lecture Notes for a Tutorial at NIPS (1997)
6. Murray-Smith, R., Sbarbaro, D.: Nonlinear Adaptive Control Using Non-Parametric Gaussian Process Prior Models. Proceedings of the 15th IFAC World Congress (2003)
7. Chryssolouris, G., Lee, M., Ramsey, A.: Confidence Interval Prediction for Neural Network Models. IEEE Transactions Neural Networks **7**(1)(1996) 229-232
8. Williams, C. K. I.: Prediction with Gaussian Processes: From Linear Regression to Linear Prediction and Beyond. In: Learning and Inference in Graphical Models, Ed. M. I. Jordan, Kluwer. (1988) 599–621
9. Williams, C. K. I., Rasmussen, C. E.: Gaussian Processes for Regression. Neural Information Processing Systems, Vol. 8. MIT Press (1996) 514-520
10. De Veaux, R. D., Schumi, J., Schweinsberg, J., Ungar, L. H.: Prediction Intervals for Neural Networks via Nonlinear Regression. Technometrics **40**(4)(1998) 273-282
11. Yang, L., Kavli, T., Carlin, M., Clausen, S., de Groot P. F. M.: An Evaluation of Confidence Bound Estimation Methods for Neural Networks. In (H.-J. Zimmermann et. al. ed.) Advances in Computational Intelligence and Learning: Methods and Applications. Kluwer Academic Publishers (2002) 71-84