Local Model Networks and Self-Tuning Predictive Control
Report: CSC-96001

Peter J. Gawthrop & Eric Ronco
Centre for Systems and Control & Department of Mechanical Engineering,
University of Glasgow,
Glasgow G12 8QQ,
Scotland.
Email: P.Gawthrop@eng.gla.ac.uk
Tel: +44 141 339 8855 x4960
FAX: +44 141 330 4343
WWW: http://www.eng.gla.ac.uk/~peterg/

April 24, 1996

Abstract

Methods drawn from control theory – Generalised Predictive Control and self-tuning control – are brought together with methods drawn from Neural Networks – Local Model Networks and Modular Neural Networks – to give a multiple-model adaptive algorithm for the control of non-linear systems whose dynamics depend on an *unknown* external signal. The method overcomes the "stability-plasticity dilemma" noted by Carpenter and Grossberg.

Keywords Local-model networks, self-tuning, predictive control.

1 Introduction

In a paper summarising their research into "Adaptive Resonance Theory", Carpenter and Grossberg (1988) emphasise "the basic design problem" for learning machines is the "stability-plasticity dilemma". In other words, this basic problem is to *learn without forgetting*. In the context of self-tuning control, this has lead to debate about choice of "exponential forgetting factors" which control the rate of learning at the expense of forgetting. Because conventional self-tuning control only involves *one* estimated model of the controlled system, it is obvious that learning a new model inevitably means forgetting an old one; thus if the controlled process reverts to a previous condition, knowledge of this has been discarded and must be relearned. This conflict between stability and plasticity inevitably arises with a single model; for this reason, a multiple model approach is used here.

Naeandra, Balakrishnan, and Ciliz (1995) give a review of some recent work using the multiple model approach in the context of adaptive control and set it in a historical context including the seminal work of Lainiotis (1976a, Lainiotis (1976b). A key aspect of their work is the dual use of models: to provide a set of controllers, each appropriate to a certain operating condition of the controlled system; and to provide a set of errors upon which to base the choice of controller. It is this dual use of multiple models which distinguishes the work from another main use of multiple models – gain scheduling – and provides the basis for the work reported here. Despite the common ground of the multiple model approach the work reported here differs from that of Naeandra, Balakrishnan, and Ciliz (1995) in a number of ways:

- implicit models (Equation 9) are used
- a generalised predictive control formulation is used
- the models are explicitly used to overcome the stability-plasticity dilemma.

As discussed by Żbikowski, Hunt, Dzieliński, Murray-Smith, and Gawthrop (1994) and by Gawthrop (1996), the Local Model Network of Johansen and Foss(1993, 1992) provides a conceptually powerful combination of general Neural Network ideas with conventional linear control techniques to provide an approach to the control of nonlinear systems. The Local Model Network may also be regarded as a multiple-model approach. This approach may be contrasted with the NARMAX-based approach of Leontaritis and Billings (1985), much used in ANN-based control (Chen, Billings, and Grant 1990; Chen, Billings, Cowan, and Grant 1990; Narendra 1990; Narendra and Parthasarathy 1990; Narendra and Parthasarathy 1991) where the system model is of the form:

$$y(t) = F(y(t-1), ..., y(t-n_y), u(t-k), ..., u(t-k-n_u)) \tag{1}$$

Indeed, in the introduction to his book (p.8), Harris (1994) comments that the NARMAX approach is "commonly adopted by the majority of the book's contributors".

This approach may be viewed as the natural extension of the conventional black-box ARMA approach to adaptive control pioneered by Åström and Wittenmark (1973) and Clarke and Gawthrop (1975). However, as discussed by Kallkuhl and Hunt (1996) the use of Equation 1 has somewhat vague theoretical foundations when regarded as an approximation to as an underlying nonlinear system of the form

$$\dot{x} = f(x, u) \tag{2}$$
$$y = g(x) \tag{3}$$

One way of contrasting this approach with ours is to realise that the standard approach (Chen, Billings, and Grant 1990; Chen, Billings, Cowan, and Grant 1990; Narendra 1990; Narendra and Parthasarathy 1990; Narendra and Parthasarathy 1991; K. J. Hunt and Gawthrop 1992) to combining linear control and ANN to give non-linear control is to embed the ANN within an otherwise linear control structure. In contrast, the approach taken here is to *embed a number of linear controllers within a network* (Gawthrop 1995; Gawthrop 1996). To distinguish between the two approaches, we make the contrast between:

- the *external net* approach comprising a network containing (local adaptive predictive) controllers
- and the *internal net* approach comprising an (adaptive predictive) controller containing a network.

We advocate the former approach.

The Generalised Predictive Control (GPC) of Clarke, Mohtadi and Tuffs (1987a, 1987b, 1989) has proved to be a popular approach to control, particularly in a self-tuning context. Although originally developed in a discrete-time context, a continuous-time version is available in both transfer function form (Demircioglu and Gawthrop 1991; Demircioglu and Gawthrop 1992) and state-space form it is the latter version that is used here. Tan and De Keyser (1994) and Saint-Donat, Bhat, and McAvoy (1994) combine GPC and ANN in a form suitable for adaptive control of nonlinear systems. However both use the internal net approach.

As discussed by Ronco, Gollee, and Gawthrop (1996), we believe that *modularity* is the key to future developments in ANN in general and their application to control in particular. For this reason, this paper presents an approach to self-tuning predictive control of nonlinear systems based on a clear and transparent decomposition of the controller within a continuous-time LMN framework using an *external* net. Our approach to ANN based adaptive predictive control may be contrasted with that of Tan and De Keyser (1994) and Saint-Donat, Bhat, and McAvoy (1994) in a number of ways in that our approach

1. uses an external net, as opposed to an internal net, framework;
2. is set in continuous time rather than discrete time;

3. contains a number of locally valid *physically transparent* models as opposed to a single black-box model with no clear physical significance

The idea of using multiple models in the context of predictive control has been explored by Chow, Kuznetsov, and Clarke (1995). They use a single controller whose coefficients are based on interpolating the poles and zeros of the GPCs corresponding to linearised models corresponding to each operating point. Our approach to LMN based predictive control may be contrasted with the multiple model approach of Chow, Kuznetsov, and Clarke (1995) in a number of ways in that our approach

1. is set in continuous time rather than discrete time;
2. contains a number of local controllers whose outputs are interpolated as opposed to a single controller whose coefficients are interpolated.
3. no scheduling signal is explicitly used – the choice of model is autonomous.

The outline of the paper is as follows. Sections 2 and 3 provide a summary of (continuous-time) Local Model Networks and Generalised Predictive Control respectively. Section 4 contains the Local Model Network based Generalised Predictive Control and forms the core of the paper and Section 5 gives a self-tuning version. Section 6 illustrates and evaluates the algorithm using a nonlinear process engineering example. Section 7 concludes the paper and point to future research directions.

2 Continuous-time LMN

This paper considers the control of *nonlinear time varying* single-input single output (SISO) systems of the form:

$$\dot{x} = f(x, u, v) \tag{4}$$

$$y = g(x) \tag{5}$$

where y,u and x are the system output, control input and state respectively. The (scalar) signal v is a disturbance signal which, typically, changes the dynamics of the system relating y and u. Unlike Gawthrop (1996), the signal v is *unknown* precluding gain-scheduling-like approaches to the problem.

As discussed in some detail by Gawthrop (1996), such systems may be linearised about m equilibrium points to give a set of m *linear* systems which, in Laplace transform terms, are of the form:

$$a(s)Y = b(s)U + d(s)V \tag{6}$$

where $a(s)$, $b(s)$ and $d(s)$ are polynomials in s of the form:

$$a(s) = \sum_{k=0}^{n} a_i s^{n-k} \tag{7}$$

and Y, U and V are the Laplace transforms of y, u and v respectively. As discussed by Gawthrop (1996), careful consideration of the linearisation process leads to the conclusion that the polynomials have order one greater than the dimension of the state x and, moreover, that

$$a(0) = a_n = 0;\ b(0)b_n = 0 \tag{8}$$

In this paper,as v is unknown, the term $d(s)V$ is not explicitly used, but rather regarded as contributing to an error term. Dividing Equation 6 by the polynomial $c(s)$ and rearranging gives:

$$E_0 = \frac{a(s)}{c(s)}Y - \frac{b(s)}{c(s)}U \tag{9}$$

where E_0 is an error term including the effect of V, linearisation error V, and other disturbances.

Defining the parameter vector θ and data vector X as

$$\theta = \begin{pmatrix} a_0 \\ a_1 \\ \dots \\ a_{n-1} \\ b_0 \\ b_1 \\ \dots \\ b_{n-1} \end{pmatrix};\ X = \frac{1}{c(s)} \begin{pmatrix} s^n Y \\ s^{n-1} Y \\ \dots \\ sY \\ s^n U \\ s^{n-1} U \\ \dots \\ sU \end{pmatrix} \tag{10}$$

(To be more precise, X is generated from the state-space system given in Laplace transform terms by Equation 10) Equation 9 becomes:

$$e_0 = \theta^T X \tag{11}$$

Following the discussion by Gawthrop (1996) an array of m models parameterised by $\hat{\theta}_i$ can be defined with corresponding errors e_i as:

$$e_i(t) = \hat{\theta}_i^T X(t) \tag{12}$$

Defining $\hat{\Theta}$ to be the matrix containing the m vectors $\hat{\theta}_i$ and e the column vector containing the m corresponding errors gives:

$$e(t) = \hat{\Theta}^T X(t) \tag{13}$$

Equation 13 is used in the sequel for selecting an appropriate model.

3 Continuous-time GPC

The Generalised Predictive Control (GPC) of Clarke, Mohtadi and Tuffs (1987a, 1987b, 1989) has proved to be a popular approach to control, particularly in a self-tuning context. Although originally developed in a discrete-time context, a continuous-time version is available in both transfer function form (Demircioglu and Gawthrop 1991; Demircioglu and Gawthrop 1992) and state-space form (Gawthrop and Siller-Alcala 1995); it is the latter version that is used here. GPC is particularly relevant for adaptive control as it is insensitive to system structural parameters such as: system order, system relative degree and number of nonminimum phase zeros. The details of GPC appear in the cited works; only the bare essentials are presented here.

Like any other emulator-based controller, GPC can be written in emulator form as:

$$R = \gamma U + \frac{g(s)}{c(s)}U + \frac{f(s)}{c(s)}Y \tag{14}$$

where R is the Laplace transform of the reference signal r and $c(s)$ is a design polynomial (Gawthrop and Siller-Alcala 1995). Using the same approach as Gawthrop (1996), and making use of Equations 8, it follows that this can be rewritten in a form exposing integral action as:

$$U = \bar{U} + \frac{1}{\gamma}(R - \bar{Y}) - \frac{s}{\gamma}\left[\frac{g'(s)}{c(s)}U - \frac{f'(s)}{c(s)}Y\right] \tag{15}$$

where

$$\bar{U} = \frac{U}{c(s)} \qquad \bar{Y} = \frac{Y}{c(s)} \tag{16}$$

$$g'(s) = g(s) - g_n \qquad f'(s) = f(s) - f_n \tag{17}$$

The first, second and third terms of Equations 15 may be regarded as generalised integral, proportional and derivative action respectively.

Defining the *control* parameter vector θ_c and *control* data vector X_c as

$$\theta_c = \frac{1}{\gamma}\begin{pmatrix} 0 \\ -f_1 \\ \cdots\cdots \\ -f_{n-1} \\ -c_n \\ -g_1 \\ \cdots\cdots \\ -g_{n-1} \\ \gamma c_n \end{pmatrix}; \; X_c = \frac{1}{c(s)}\begin{pmatrix} s^n Y \\ s^{n-1}Y \\ \cdots\cdots \\ sY \\ Y \\ s^n U \\ s^{n-1}U \\ \cdots\cdots \\ U \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ \cdot\,\cdot \\ 0 \\ W \\ 0 \\ 0 \\ \cdot\,\cdot \\ 0 \end{pmatrix} \tag{18}$$

Equation 15 becomes:

$$u = \theta_c^T X_c \tag{19}$$

Note that X_c is the same as X (Equation 10) but with the addition of two more elements. In a similar way to that of Section 2, an array of m controllers parameterised by $\hat{\theta}_{ci}$ can be defined with corresponding controls u_i as:

$$u_i = \hat{\theta}_{ci}^T X_c \tag{20}$$

Defining $\hat{\Theta}_c$ to be the matrix containing the m vectors $\hat{\theta}_{ci}$ and u the column vector containing the m corresponding control signals gives:

$$u = \hat{\Theta}_c^T X \tag{21}$$

4 LMNGPC

As a preliminary to the self-tuning algorithm of Section 5, a local-model network based Generalised Predictive Control (LMNGPC) is presented. Although there is no explicit parameter estimation, this algorithm does have an adaptive aspect in that a local-model network is used to choose an appropriate weighting for the m control signals. Unlike gain scheduling, and some local controller networks (Żbikowski, Hunt, Dzieliński, Murray-Smith, and Gawthrop 1994), a measurement of the perturbing variable v is *not* required.

Following Naeandra, Balakrishnan, and Ciliz (1995), the key idea used here is to use the error vector $e(t)$ (Equation 13) to determine the 'best' model, and corresponding controller, to use at each time t. To provide robustness to sudden disturbances, a filtered squared version $\bar{e}(t)$ of e is used give by:

$$\tau \frac{d\bar{e}}{dt}(t) = e^2(t) - \bar{e}(t) \tag{22}$$

The filter time constant τ is chosen to give a trade off between smoothing extraneous noise and detecting genuine changes in the system.

It is this m dimensional vector $\bar{e}(t)$ that is used to determine the LMN weighting vector ρ which selects from the vector of possible control signals u (Equation 21). This weighing vector corresponds to the 'gating function' of Jacobs and Jordan (1993) and the elements ρ_i must satisfy:

$$0 \leq \rho_i \leq 1 \text{ for all } i \tag{23}$$

$$\sum_{i=1}^{m} \rho_i = 1 \tag{24}$$

There are many possible ways of generating ρ; one such possibility is the SOFTMAX function (Haykin 1994) with variable positive gain λ

$$\rho_i = \frac{\exp \lambda \bar{e}_i}{\sum_{j=1}^{m} \exp \lambda \bar{e}_j} \tag{25}$$

Another possibility is a 'winner takes all' operation where one element of ρ is one and the rest zero; this is approximated by choosing a large λ in Equation 25.

The algorithm can now be summarised as:

Algorithm 1

- *Initialisation*

  1. *Linearise Equation 2 about m equilibria.*
  2. *Create the* system *parameter matrix $\hat{\Theta}$.*
  3. *Create the* controller *parameter matrix $\hat{\Theta}_c$.*

- *On-line*

  1. *Generate the controller data vector X_c using state-variable filters (X is a subvector of X_c).*
  2. *Generate the error vector $e = \hat{\Theta}^T X$ (Equation 13)*
  3. *Generate the weight vector ρ (Equation 25)*
  4. *Generate the control vector $u = \hat{\Theta}_c^T X_c$ (Equation 21)*
  5. *Generate the control signal $u = \rho^T u$ (Equation 21)*

5 Self-tuning LMNGPC

Equation 12 gives the ith model error in terms of the data vector $X(t)$ and the ith model parameter vector $\hat{\theta}_i$. The same equation may also be used to choose $\hat{\theta}_i$ to minimise the error over a time-period T_e. In particular the mean squared error from time $T - T_e$ to time T is given by:

$$\int_{T-T_e}^{T} e_i^2 \, dt = \hat{\theta}_i^T S(T) \hat{\theta}_i \tag{26}$$

where

$$S(T) = \int_{T-T_e}^{T} X(t) X^T(t) \, dt \tag{27}$$

To avoid the trivial solution $\hat{\theta}_i = 0$, $\hat{\theta}_i$ is constrained by $\|\hat{\theta}_i\| = 1$. The minimisation of Equation 26 is then simply accomplished by performing a Singular Value Decomposition (SVD) of $S(T)$ and choosing the singular vector corresponding to the smallest singular value.

This leads to the self-tuning version of Algorithm 1.

Algorithm 2 (Self-tuning)

- *Initialisation*

  1. *Choose a set m of arbitrary models (including prior information if available).*
  2. *Create the* system *parameter matrix* $\hat{\Theta}$.
  3. *Create the* controller *parameter matrix* $\hat{\Theta}_c$.
  4. *Choose an* identification time period t_e.
  5. *Set* $i = m + 1$ *and* $S_i = 0$

- *On-line*

  1. *As algorithm 1*
  2. *Update the information matrix* S_i
  3. *Every* T_e*:*
     (a) *Compute a new parameter vector* $\hat{\theta}_i$ *from the SVD of* S
     (b) *Add* $\hat{\theta}_i$ *as a new column of* $\hat{\Theta}$
     (c) *Reinitialise* S*:* $S_{i+1} = \beta S_i$
     (d) *Increment* i

Remarks

- The parameter β allows remembering of information from previous time periods; typically $\beta = 0$ in the multi-model version.

- A single model version of this algorithm arises when $m = 1$ and i is always set to 1; thus each new model overwrites the previous model. With small T_e, this is essentially standard self-tuning control with forgetting factor β.

- The ever growing data requirements are discussed further in Section 7.

6 An example

A detailed model of an experimental heated tank system is given by Costello and Gawthrop (1995). The system is third order and has a single output y = temperature (with respect to ambient), single control input u = Heater power and a single disturbance $v = inflow$. This system was simulated, together with the algorithms of this paper, using Simulink/Matlab for a total time of 200. The reference output temperature $r(t)$ was a square-wave of period 20 and unit amplitude. (The use of a periodic reference is for illustration; it is not required to be periodic). The disturbance v is sinusoidal as illustrated in Figure 1.

Parameter	Meaning	Value
$p(s)$	Reference model	$(1+0.1s)(1+0.5s)^2$
$c(s)$	State-variable filter	$(1+0.5s)^4$
t_1	Min. time horizon	0
t_2	Max time horizon	0.1
N_ϕ	Predictor order	10
N_u	Control order	0

Table 1: GPC parameters

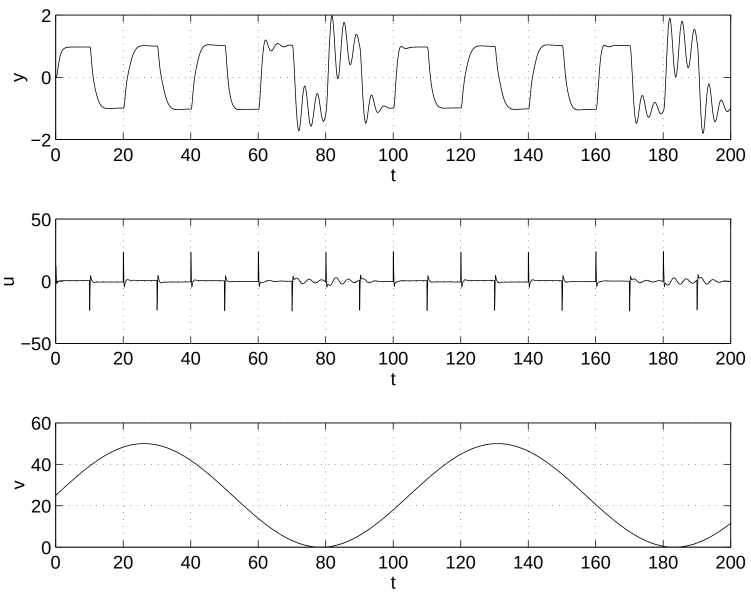

Figure 1: Non-adaptive control based on a single model linearised about $v = 50$. The three plots show the output y, control input u and disturbance d.

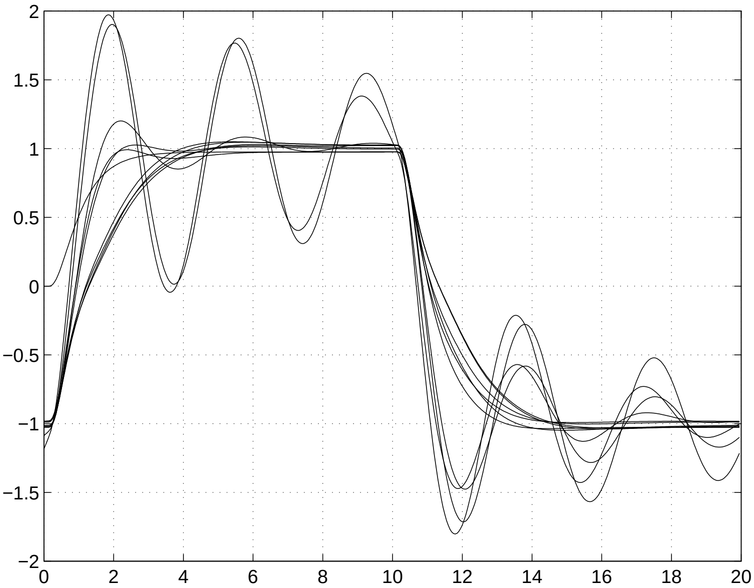

Figure 2: Non-adaptive control based on a single model linearised about $v = 50$. The output y from Figure 1 is superimposed. Notice the variation in response due to changing system dynamics due to v.

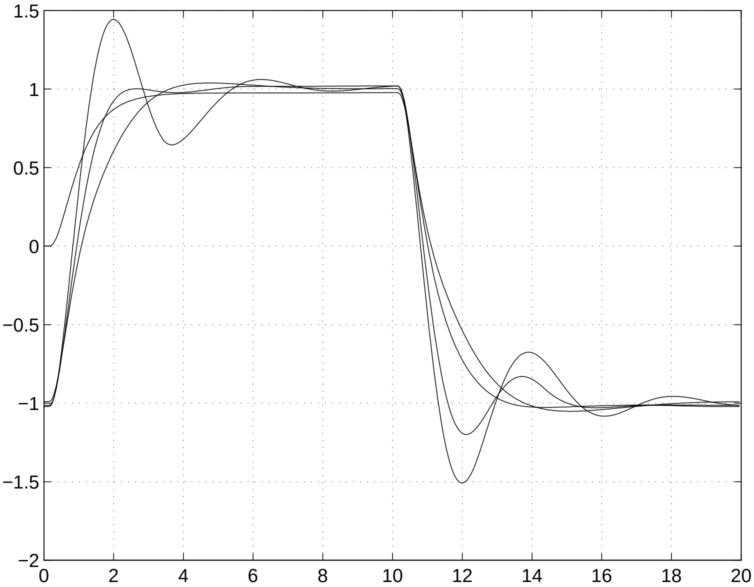

Figure 3: Adaptive control based on a single model ($0 \leq t \leq 100$).

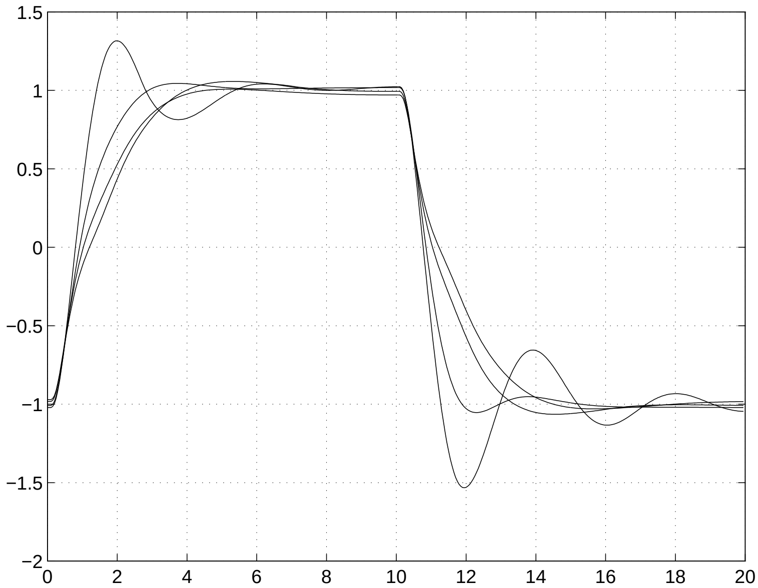

Figure 4: Adaptive control based on a single model($100 \leq t \leq 200$). Notice that the performance is no better than that of Figure 3 – the earlier learning has been forgotten

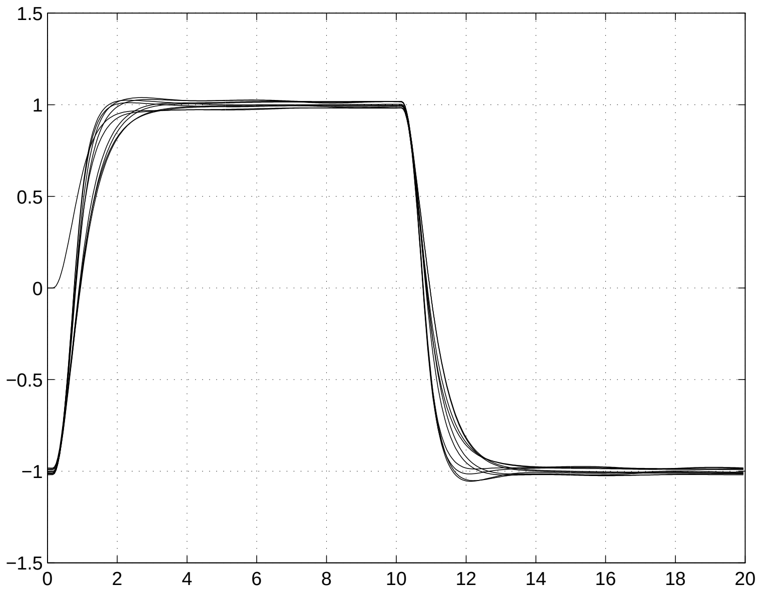

Figure 5: Non-adaptive control based on 26 models linearised about values of v throughout it's range. Notice the improvement relative to Figure 2

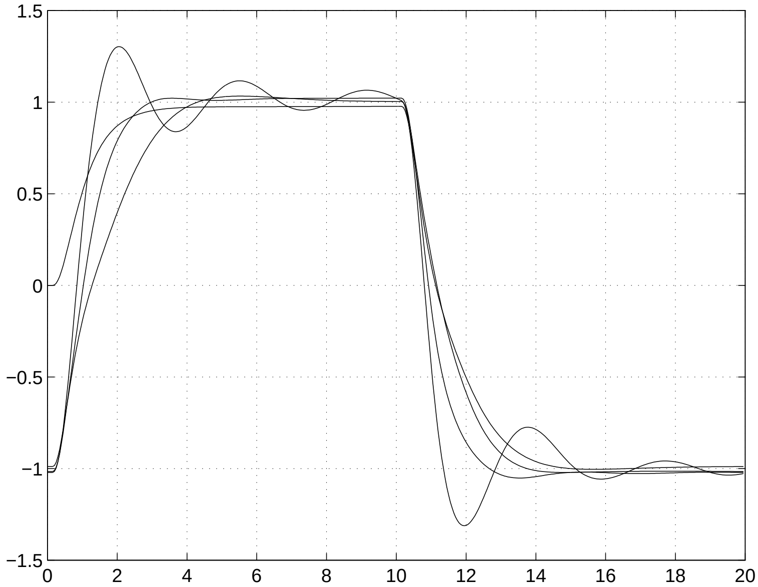

Figure 6: Adaptive control based on 26 models ($0 \leq t \leq 100$).

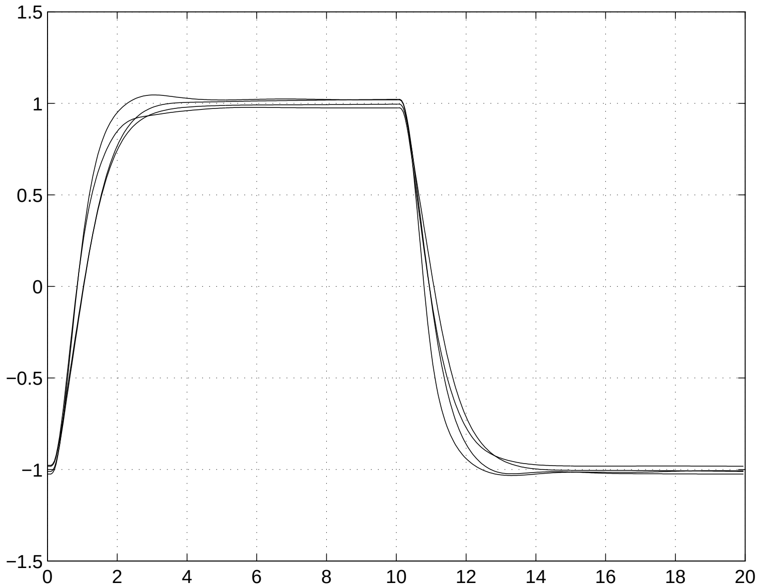

Figure 7: Adaptive control based on 26 models ($100 \leq t \leq 200$). Notice the improvement in performance relative to Figure 6 – this is because no information has been forgotten

In all cases, the GPC design parameters (Gawthrop and Siller-Alcala 1995) were as given in Table 1. Figures 2 – 7 show y, from each period of the reference r, superimposed. This gives a clearer view of the change of performance with time. The single-model versions used $\beta = 0.5$ and the multi-model versions used $\beta = 0$. In each case $T_e = 9$.

7 Conclusions and further work

We have presented an algorithm which overcomes the "stability-plasticity dilemma" of Carpenter and Grossberg (1988) by combining methods associated with self-tuning control and methods associated with artificial neural networks.

The algorithm presented here has an ever-growing memory requirement as models are never discarded; even though some of the models may be similar. We believe that the way forward here is to *cluster* similar models. There are two issues here: to determine which models to combine and how to combine them. There are a number of possible approaches to the former problem. Clustering could be based on the (unit norm) parameter vectors $\hat{\theta}_i$ using standard self-organising methods (see Haykin (1994) for a discussion), adaptive resonance theory Carpenter and Grossberg (1988) or progressive learning (Ronco, Gollee, and Gawthrop 1996). The latter problem is essentially that of deducing a parameter vector describing a number of *non-contiguous* sets of data; this problem has been addressed by Gawthrop (1984).

Acknowledgements

This work was partially supported by ESPRIT grant 8039 (NACT).

References

Åström, K. J. and B. Wittenmark (1973). On self-tuning regulators. *Automatica 9*, 185–199.

Carpenter, G. A. and S. Grossberg (1988, March). The art of adaptive pattern recognition by a self-organising neural network. *IEEE Computer 21*(3), 77–88.

Chen, S., S. A. Billings, C. F. Cowan, and P. M. Grant (1990). Practical identification of NARMAX models using radial basis functions. *Int. J. Control 52*, 1327–1350.

Chen, S., S. A. Billings, and P. M. Grant (1990). Non-linear system identification using neural networks. *Int. J. Control 51*, 1191–1214.

Chow, C., A. Kuznetsov, and D. Clarke (1995). Using multiple models in predictive control. In A. Isidori (Ed.), *Proceedings of the 3rd European Control Conference, Rome, Italy*, pp. 1732–1737.

Clarke, D. W. and P. J. Gawthrop (1975). Self-tuning controller. *Proceedings IEE 122*(9), 929–934.

Clarke, D. W. and C. Mohtadi (1989). Properties of generalised predictive control. *Automatica 25*(6), 859–875.

Clarke, D. W., C. Mohtadi, and P. S. Tuffs (1987a). Generalised Predictive Control—Part I. The Basic Algorithm. *Automatica 23*(2), 137–148.

Clarke, D. W., C. Mohtadi, and P. S. Tuffs (1987b). Generalised Predictive Control—Part II. Extensions and Interpretations. *Automatica 23*(2), 149–160.

Costello, D. J. and P. J. Gawthrop (1995). Physical model-based control: Experiments with a stirred-tank heater. Technical Report CSC-95003, Glasgow University Centre for Systems and Control.

Demircioglu, H. and P. J. Gawthrop (1991, January). Continuous-time generalised predictive control. *Automatica 27*(1), 55–74.

Demircioglu, H. and P. J. Gawthrop (1992). Multivariable continuous-time generalised predictive control. *Automatica 28*(4), 697–713.

Gawthrop, P. (1995). Continuous-time local state local model networks. In *Proceedings of 1995 IEEE conference on Systems Man and Cybernetics*, Vancouver, British Columbia, pp. 852–857.

Gawthrop, P. J. (1984). Parameter identification from non-contiguous data. *Proceedings IEE 131 Pt.D*(6), 261–265.

Gawthrop, P. J. (1996). Continuous-time local model networks. In R. Żbikowski and K. J. Hunt (Eds.), *Neural Adaptive Control Technology*, World Scientific Series in Robotics and Intelligent Systems, Vol. 15, pp. 41–70. Singapore: World Scientific.

Gawthrop, P. J. and I. I. Siller-Alcala (1995). Non-linear generalised predictive control. Technical Report CSC-95031, Glasgow University Centre for Systems and Control.

Harris, C. (1994). *Advances in Intelligent Control.* Taylor and Francis.

Haykin, S. (1994). *Neural Networks: A Comprehensive Foundation.* IEEE press.

Jacobs, R. and M. Jordan (1993). Learning piecewise control strategies in a modular neural network architecture. *IEEE Transaction on Systems, Man, and Cybernetics 23*(2), 337–345.

Johansen, T. A. and B. A. Foss (1992). A NARMAX model representation for adaptive control based on local models. *Modeling, Identification, and Control 13*(1), 25–39.

Johansen, T. A. and B. A. Foss (1993). Constructing NARMAX models using ARMAX models. *Int. J. Control 58*, 1125–1153.

K. J. Hunt, R. Żbikowski, D. S. and P. J. Gawthrop (1992). Neural networks for control systems—a survey. *Automatica 28*(6), 1083–1112.

Kallkuhl, J. C. and K. J. Hunt (1996). Discrete-time neural model structures for continuous nonlinear systems: Fundamental properties and control aspects. In R. Żbikowski and K. J. Hunt (Eds.), *Neural Adaptive Control Technology*, World Scientific Series in Robotics and Intelligent Systems, Vol. 15, pp. 3–40. Singapore: World Scientific.

Lainiotis, D. (1976a, August). Partitioning: A unifying framework for adaptive systems. I. estimation. *Proc. IEEE 64*, 1126–1143.

Lainiotis, D. (1976b, August). Partitioning: A unifying framework for adaptive systems. II. control. *Proc. IEEE 64*, 1182–1197.

Leontaritis, I. J. and S. A. Billings (1985). Input-output parametric models for non-linear stytems. Part I: deterministic non-linear systems. *Int. J. Control 41*, 303–328.

Naeandra, K., J. Balakrishnan, and M. Ciliz (1995, June). Adaptation and learning using multiple models switching and tuning. *IEEE control systems magazine 15*(3), 37–51.

Narendra, K. and K. Parthasarathy (1991). Gradient Methods for the Optimization of Dynamical Systems Containing Neural Networks. *IEEE Trans. on Neural Networks 2*, 252–262.

Narendra, K. S. (1990). *Neural Networks for Control*, Chapter 5, pp. 115–142. MIT Press.

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

Ronco, E., H. Gollee, and P. J. Gawthrop (1996). Modular neural network and self-decomposition. *Connection Science (special issue: COMBINING NEURAL NETS)*. (To appear).

Saint-Donat, J., N. Bhat, and T. McAvoy (1994). Neural net based predictive control. In C. Harris (Ed.), *Advances in Intelligent Control*, pp. 183–198. Taylor and Francis.

Tan, Y. and R. De Keyser (1994). Neural network based adaptive predictive control. In D. Clarke (Ed.), *Advances in Model-based Predictive Control*, pp. 358–369. Oxford University Press.

Żbikowski, R., K. J. Hunt, A. Dzieliński, R. Murray-Smith, and P. J. Gawthrop (1994). A review of advances in neural adaptive control systems. Technical Report of the ESPRIT NACT Project TP-1, Glasgow

University and Daimler-Benz Research. (Available from FTP server ftp.mech.gla.ac.uk as PostScript file /nact/nact_tp1.ps).