Appl. Math. and Comp. Sci., 1998, Vol.8, No.3, 585–598

MODEL-BASED PREDICTIVE CONTROL OF LARGE-SCALE SYSTEMS USING A NEURAL ESTIMATOR

SPYROS TZAFESTAS*, EFTHIMIOS KYRIANNAKIS*
YIANNIS ANTHOPOULOS*

An approach to the design of discrete-time decentralized control systems based on model-based predictive control (MBPC) and neural estimation is proposed. The class of interconnected large-scale systems (LSS) is considered, and a model is used at each control station to predict the corresponding subsystem output over a long time period. In the case of subsystems with m-step delay information patterns the non-locally available interaction trajectories are estimated by a multi-layer neural network trained on-line with a modified backpropagation-type algortithm. Representative computer simulation results are provided and compared for a set of illustrative examples. The proposed control scheme shows better performance than the other schemes, and also covers the important case where the subsystems' interactions are nonlinear.

1. Introduction

The decentralized control of interconnected dynamical systems is still generating an increasing interest among theorists and practitioners (Bahnasawi *et al.*, 1990; Corfmat and Morse, 1976; Jamshidi, 1996). One of the benefits of decentralized control is that large-scale systems can be decomposed into many subsystems (Linnemann, 1984; Singh and Titli, 1978), and the control design and implementation of each of them can be performed independently. This simplifies the overall control problem. Moreover, the computational burden can be shared by all the control stations involved. The main difficulty in designing decentralized control systems is the limited information available for the control (Ho and Chu, 1974; Sandell and Athans, 1974; Yoshikawa and Kobayashi, 1978). The set of the control stations is constrained to have access only to local information, i.e. only to measurements of local outputs and states with no communication allowable among them (Kurtaram and Sivan, 1974). This is exactly what characterizes a non-classical information pattern. Finally, the local control stations cannot take into account the interactions with other subsystems, and therefore techniques to deal with this problem must be developed. Many optimal decentralized control algorithms have been proposed for dealing with such problems.

* Intelligent Robotics and Automation Laboratory, Department of Electrical and Computer Engineering, National Technical University of Athens, Zographou 15773, Athens, Greece, e-mail: tzafesta@softlab.ece.ntua.gr.

But it is well-known that optimal control is not very suitable for complex industrial systems or general large-scale systems, since it needs an accurate model of the system being controlled and it is sensitive to parameter variations and to the existence of stochastic disturbances.

A new control appproach which does not have the drawbacks of standard optimal-regulator control and is suitable for complex large-scale systems (LSS's) is the so-called Model-Based Predictive Control (MBPC) approach (Clarke, 1990; De Keyser, 1990; Richalet, 1990). This approach was developed in the mid-seventies and allows for model uncertainties, updates the output of the model by closed-loop corrections, and optimizes the control law on a moving horizon. On the other hand, neural networks (NN) are parallel systems consisting of a large number of relatively simple processing units, highly interconnected via unidirectional signal channels called connections. Each unit performs a (usually) nonlinear transformation on the weighted sum of its inputs in order to produce the output signal which is then fed to the other units connected with it. Neural networks process information in a non-parametric way and are able to learn by iteratively adjusting their weights (Hecht-Nielsen, 1990). In this paper the decentralized control approach is combined with model-based predictive control (Tzafestas *et al.*, 1995) and a neural estimator. The resulting overall MBPC-NN control scheme was verified by simulation to be a very good candidate for application to industrial and other LSS's of real-life systems (Singh and Titli, 1977; Tzafestas, 1989).

2. Problem Formulation

Consider a discrete-time, linear, possibly time-varying large-scale system which consists of N-interconnected subsystems (see Fig. 1), each of which has the following state-space description:

$$\boldsymbol{x}_i(t+1) = \boldsymbol{A}_i\boldsymbol{x}_i(t) + \boldsymbol{B}_i\boldsymbol{u}_i(t) + \boldsymbol{E}_i\boldsymbol{z}_i(t) \tag{1a}$$

$$\boldsymbol{x}_i(0) = \boldsymbol{x}_{i_0} \tag{1b}$$

$$\boldsymbol{y}_{m_i}(t) = \boldsymbol{C}_i\boldsymbol{x}_i(t) \tag{1c}$$

for $i = 1, 2, \dots, N$, where $\boldsymbol{x}_i(t)$ is the n_i-dimensional state vector of the i-th subsystem at time t, $\boldsymbol{u}_i(t)$ is the r_i-dimensional control vector of the i-th subsystem at time t, $\boldsymbol{y}_{m_i}(t)$ is the p_i-dimensional output vector of the model of the i-th subsystem at time t, $\boldsymbol{z}_i(t)$ is the q_i-dimensional interconnection vector which describes the influence of all other subsystems upon the i-th one.

The vector $\boldsymbol{z}_i(t)$ is considered to be a linear combination of the states of all other subsystems, i.e.[1]

$$\boldsymbol{z}_i(t) = \sum_{j=1}^{N} \boldsymbol{L}_{ij}\boldsymbol{x}_j(t), \quad i = 1, 2, \dots, N, \ \text{with } i \neq j \tag{2}$$

[1] This assumption can be relaxed due to the nonlinear estimation capability of neural networks.

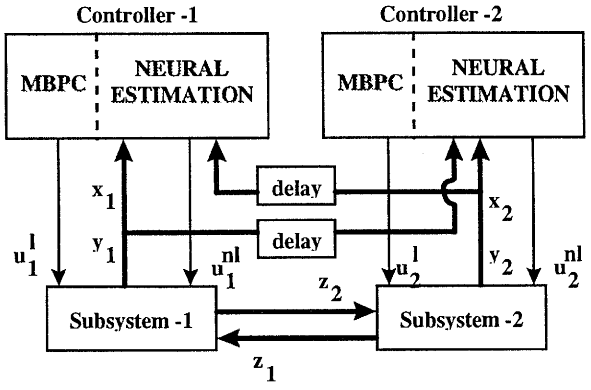

Fig. 1. The decentralized control system with mixed-type control stations.

where L_{ij} are matrices of appropriate dimensions assumed to be known to every control station. Let us note that the output $y_{m_i}(t)$ of the model may generally differ a little from the real output $y_i(t)$ because of modelling errors or noise which affect the whole system or its parts. Finally, the matrices A_i, B_i, E_i, C_i are of suitable dimensions. The problem is *to find at every time t the best control $u_i(t)$ for the i-th subsystem, which leads the current value of the output $y_i(t)$ to its set-point $w_i(t)$.* The control must be "the best" in the sense of minimizing a cost function which will be described later.

Moreover, the control laws must be specified in a decentralized way. In the following, predictive control techniques and neural estimation will be used to satisfy the problem requirements. It will be shown that the control laws are of the form

$$u_i(t) = u_i^l(t) + u_i^{nl}(t), \quad i = 1, 2, \dots, N \tag{3}$$

where $u_i^l(t)$ is the "local" part of the control, i.e. the part which depends on information available to the i-th control station, and $u_i^{nl}(t)$ is the "non-local" part of the control law, which depends on information not available to the i-th control station.

3. MBPC and Computation of the Local Part of the Control

At each time t, the output $y_i(t + k)$ is predicted over a future period of time $k = 1, 2, \dots, L_y$ where L_y is the prediction horizon. The predictions are denoted by $y_{p_i}(t + k/t)$ and determined by means of a model, e.g. a state-space model (1). The predictions $y_{p_i}(t + k/t)$, $k = 1, 2, \dots, L_y$ depend on the future control values $u_i(t + k/t)$, $k = 0, 1, \dots, L_u$, where L_u is the control horizon ($L_u \leq L_y$). In the control horizon we have

$$u_i(t + L_u + k/t) = u_i(t + L_u - 1), \quad k \geq 0$$

The output predictions for the i-th subsystem can be calculated as

$$y_{p_i}(t + k/t) = y_{m_i}(t + k/t) + q_i(t + k/t) \tag{4}$$

where, by (1a) and (1b),

$$\begin{aligned} \boldsymbol{y}_{m_i}(t+k/t) &= \boldsymbol{C}_i\boldsymbol{x}_i(t+k/t) \\ &= \boldsymbol{C}_i\Big[\boldsymbol{A}_i^k\boldsymbol{x}_i(t) + \sum_{j=1}^{k}\boldsymbol{A}_i^{j-1}\boldsymbol{B}_i\boldsymbol{u}_i(t+k-j/t)\Big] \\ &= \sum_{j=1}^{k}\boldsymbol{A}_i^{j-1}\boldsymbol{E}_i\boldsymbol{z}_i(t+k-j/t), \quad k=1,2,\ldots,L_y \end{aligned} \tag{5}$$

and $\boldsymbol{q}_i(t+k/t)$ is the cloded-loop correction vector based on the information set available at time t. A recommended form for $\boldsymbol{q}_i(t+k/t)$ is

$$\boldsymbol{q}_i(t+k/t) = \boldsymbol{y}_i(t) - \boldsymbol{y}_{m_i}(t) \tag{6}$$

where $\boldsymbol{y}_i(t)$ is the measured value of the output vector at time t. A reference trajectory $\boldsymbol{r}_i(t+k/t)$, $k=1,2,\ldots,L_y$ is defined over the prediction horizon, which describes how one wants to guide the output vector $\boldsymbol{y}_i(t)$ to its set-point $\boldsymbol{w}_i(t)$, i.e.

$$\boldsymbol{r}_i(t+k/t) = \boldsymbol{w}_i(t_k/t) - \boldsymbol{v}_i(t+k/t) \tag{7}$$

where $\boldsymbol{v}_i(t+k)$ is a correction vector based on the previous error information set $\{\boldsymbol{w}_i(t)-\boldsymbol{y}_i(t),\ldots,\boldsymbol{w}_i(1)-\boldsymbol{y}_i(1)\}$. A simple form which gives good results is the following:

$$\boldsymbol{v}_i(t+k/t) = a^k\Big[\boldsymbol{w}_i(t) - \boldsymbol{y}_i(t)\Big] \tag{8}$$

where $0 < a \leq 1$ is a tuning parameter which specifies the desired closed-loop dynamics ($a \to 0$ for fast control, $a \to 1$ for slow control). The reference trajectory is initiated at the current measured output, i.e. $\boldsymbol{r}_i(t/t) = \boldsymbol{y}_i(t)$. Note that if the future set-point values $\boldsymbol{w}_i(t+k/t)$, $k=1,2,\ldots,L_y$ are unknown at time t, one can assume

$$\boldsymbol{w}_i(t+k/t) = \boldsymbol{w}_i(t), \quad k=1,2,\ldots,L_y$$

All the above issues are illustrated in Fig. 2.

The cost function of the i-th control station has the form

$$J_i(t) = \frac{1}{2}\sum_{k=L_0}^{L_y}\left\|\boldsymbol{r}_i(t+k/t) - \boldsymbol{y}_{p_i}(t+k/t)\right\|^2_{\boldsymbol{Q}_i(k)} + \frac{1}{2}\sum_{k=0}^{L_u-1}\left\|\boldsymbol{u}_i(t+k/t)\right\|^2_{\boldsymbol{R}_i(k)} \tag{9}$$

where $\boldsymbol{Q}_i(k) \geq 0$, $k=L_0,\ldots,L_y$ and $\boldsymbol{R}_i(k) \geq 0$, $k=0,\ldots,L_u-1$. Since $J_i(t)$ varies with time t and has a moving optimization horizon, only the first term in the optimal solution is implemented to control the i-th subsystem. The optimization parameter L_0 determines, together with L_y, the "coincidence horizon", i.e. one wants the predicted output to follow the reference trajectory over the time interval $[t+L_0,\ldots,t+L_y]$. Minimizing the cost function (9), we get

$$\boldsymbol{u}_i(t) = \boldsymbol{u}_i^l(t) + \boldsymbol{u}_i^{nl}(t) = \boldsymbol{D}_i(t)\gamma_i(\boldsymbol{x}_i,\boldsymbol{y}_i) + \boldsymbol{D}_i(t)\delta_i(\boldsymbol{z}_i) \tag{10}$$

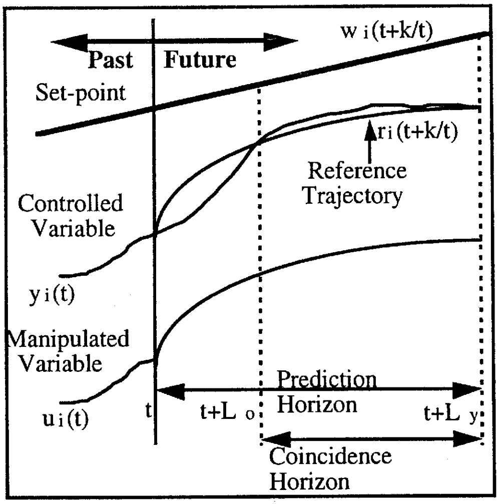

Fig. 2. The reference trajectory, the set-point trajectory, the prediction horizon and the coincidence horizon.

where $\boldsymbol{D}_i$ and γ_i, δ_i are the computable matrix and functions, respectively (Tzafestas *et al.*, 1995). The second term is not locally computable because it depends on the non-available $z_i(t)$ and its predictions in the prediction horizon $(z_i(t+k/t),\ k = 0, 1, \dots, L_y - 1)$. The next section deals with that problem.

4. Neural Estimation for the Non-Local Part of the Control

Multilayer Perceptrons (MLP's) are networks whose processing units (neurons) are arranged in layers. The nodes of each layer take as the input the outputs of the nodes of the previous layer and perform a non-linear (usually sigmoidal) transformation on them in order to produce their outputs. The units belonging to one layer are connected only to the nodes of the previous and the next layer and not to one another (Fig. 3). The network learns by adjusting its weights according to the backpropagation algorithm which computes the error between the actual and the desired output for a number of training patterns (supervised learning) and "backpropagates" it to the units which adjust their associated weight values so that the actual response of the network moves closer to the desired response. To express the above in a formal mathematical way, the output of the i-th node is assumed to be

$$o_i = \sigma\left(\sum_j w_{ij} + \vartheta_I\right), \qquad \sigma(x) = 1/(1-e^{-x}) \tag{11}$$

where w_{ij} is the weight corresponding to the connection from node j to node i, and ϑ_i a fixed value called the 'threshold.' The algorithm minimizes the value of the

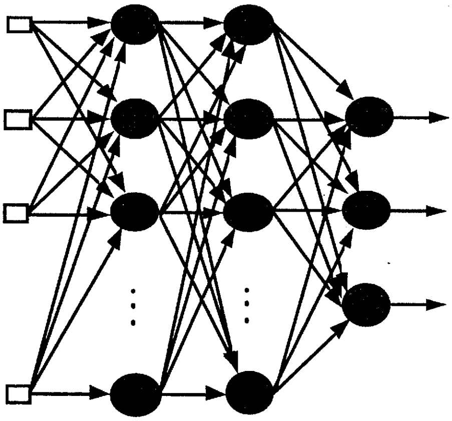

Fig. 3. A multi-layer perceptron with two hidden layers.

"energy" function

$$E(w) = \sum_p (t_p - o_p)^2 \tag{12}$$

where t_p and o_p stand respectively for the desired and the actual output when the network receives as the input the pattern p. To do so, it uses the delta rule, thus

$$w_{ij}(n+1) = w_{ij}(n) - n\nabla E(w) \tag{13}$$

MLP's can be trained either off-line when all of the training patterns are known beforehand or on-line in the opposite case (Rumelhart *et al.*, 1989). It can be proved that an MLP trained with the back-propagation algorithm can approximate arbitrarily well every function belonging to the L_2 class. Another very important characteristic of the MLP networks is their ability to generalize, i.e. to give a valid (meaningful) output for the input patterns to which they have not been trained (since after their training they implement a continuous mapping function).

In this paper, a multi-layer perceptron is used at each control station to predict the interaction trajectories over the prediction horizon in the case where the information pattern of the problem is an m-step delay sharing one. A decentralized control problem is said to have an *m-step delay sharing pattern* when it permits the spreading of its information through the subsystems but with delay of m time steps. Clearly, each control station obtains *instantaneously* all the information about its associated subsystem, and after the delay of m time steps all the information available to *all* the control stations. For our problem this means that at time t in the i-th control station the vectors $\boldsymbol{x}_j(t-m)$, $j \neq i$ and all the past values $\boldsymbol{x}_j(t-m-k)$, $k > 0$ are known. Then one can calculate $z_i(t-m)$ using (2) as

$$z_i(t-m) = \sum_{j=1}^{N} \boldsymbol{L}_{ij}\boldsymbol{x}_j(t-m), \quad i = 1, 2, \ldots, N, \text{ with } i \neq j \tag{14}$$

i.e. $z_i(t-m)$ is well-known to the i-th control station at time t. An estimate of $z_i(t+k/t)$, $k=0,1,\dots,L_y-1$ can be obtained by an appropriate model of $z_i(t-m)$ as a function of $z_i(t-m-j)$, $j=1,2,\dots,p$, where p is a design parameter. For that purpose the vector $z_i(t-m)$ is computed as the output of an MLP with $z_i(t-m-j)$ values as its inputs, $j=1,2,\dots,p$. In this case, of course, the training is done on-line since each estimated value is used for the estimation of the next one. The training algorithm is a variant of the standard back-propagation algorithm, i.e. the energy function to be minimized is as follows:

$$E(\boldsymbol{w})=\sum_p (t_p-o_p)^2+\frac{n_w\sum_i\sum_j w_{ij}^2}{\sum_i\sum_j w_{ij}^2+n_w K_2} \tag{15}$$

where w_{ij} is the weight of the connection from node j to node i, and K_2, n_w are appropriate constants. The extra term increases the stability of the algorithm and the generalization capability of the network by reducing the number of active weights (i.e. the weights with values not practically equal to zero). The weights are again adapted according to the delta rule where the gradient of the new energy function is computed. The initial weight values are random but small, so that the derivatives of the network output attain their maximum values. During the first m time-steps before the real value of the predicted variable is communicated to the subsystem, the network output is random. From that time on, the network is trained on-line so as to minimize the energy funtion (15), where p increases as more real values are communicated to the subsystem. The training stops when the value of $E(\boldsymbol{w})$ becomes smaller than a preset limit depending on the specifications of each problem.

5. Simulation Results

Extensive simulation studies have been carried out to demonstrate the effectiveness of the proposed approach. Zero initial conditions (start-up of the operation) were assumed and the constrained MBPC version was used in order to take into account physical constraints on the control variables of the problem. The on-line training of the NN with the extra term in the energy function was proved to upgrade the steady-state response performance (Tzafestas *et al.*, 1996), but with the respective increase in the duration of the transient state due to the NN training process (also depending on the delay of the sharing information pattern). Moreover, due to the generality which the NN offers to the estimation problem, the proposed technique can treat processes with non-linear interaction models.

System (i): The present system consists of three subsystems with state-space matrices

$$\boldsymbol{A}_1=\begin{bmatrix}0 & 1\\ -0.3 & 0.2\end{bmatrix},\quad \boldsymbol{A}_2=\begin{bmatrix}0 & 1\\ 0.1 & -0.4\end{bmatrix},\quad \boldsymbol{A}_3=\begin{bmatrix}0 & 1\\ -0.2 & -0.5\end{bmatrix}$$

$$\boldsymbol{B}_1=\begin{bmatrix}0.3 & 0.1\\ -0.1 & 0.3\end{bmatrix},\quad \boldsymbol{B}_2=\begin{bmatrix}0.2 & 0\\ 0.1 & 0.2\end{bmatrix},\quad \boldsymbol{B}_3=\begin{bmatrix}0.4 & 0.1\\ 0.1 & 0.4\end{bmatrix}$$

$$\begin{aligned}
\boldsymbol{C}_1 &= \begin{bmatrix} 2 & 1 \end{bmatrix}, & \boldsymbol{C}_2 &= \begin{bmatrix} -3 & 1.5 \end{bmatrix}, & \boldsymbol{C}_3 &= \begin{bmatrix} 1 & 4 \end{bmatrix} \\
\boldsymbol{L}_{12} &= \begin{bmatrix} 2 & 1 \end{bmatrix}, & \boldsymbol{L}_{21} &= \begin{bmatrix} -1 & 1 \end{bmatrix}, & \boldsymbol{L}_{31} &= \begin{bmatrix} 0 & 2 \end{bmatrix} \\
\boldsymbol{L}_{13} &= \begin{bmatrix} -1 & 0 \end{bmatrix}, & \boldsymbol{L}_{23} &= \begin{bmatrix} 0 & -2 \end{bmatrix}, & \boldsymbol{L}_{32} &= \begin{bmatrix} -2 & 1 \end{bmatrix}
\end{aligned}$$

and a two-step delay sharing pattern. The predictions for the interconnections are produced from an NN since the system allows for the spreading of its information. For the controller the parameters are $L_y = 10$, $L_u = 4$, $a = 0.1$, initial $p = 3$, $\boldsymbol{Q}_i = \boldsymbol{I}_i$ and $\boldsymbol{R}_i = 10^{-3}\boldsymbol{I}_i$ ($\boldsymbol{I}_i$ is the identity matrix of appropriate dimensions).

The MLP has one output layer with one node and a hidden layer with three nodes. One can easily observe (see Figs. 4–6) good tracking in the steady-state response, whereas some oscillations in the transient response are due to the delay of the information flow.

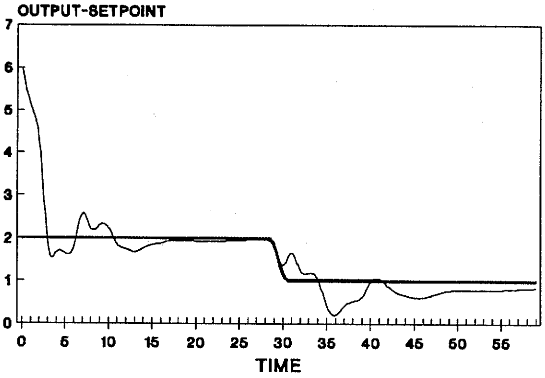

Fig. 4. Subsystem 1: output-setpoint trajectories.

System (ii): This system consists of two subsystems with state-space matrices

$$\begin{aligned}
\boldsymbol{A}_1 &= \begin{bmatrix} 0 & 1 \\ 0.2 & -0.4 \end{bmatrix}, & \boldsymbol{A}_2 &= \begin{bmatrix} 0 & 1 \\ -0.5 & -0.4 \end{bmatrix} \\
\boldsymbol{B}_1 &= \begin{bmatrix} 0.2 & -0.1 \\ 0.1 & 0.4 \end{bmatrix}, & \boldsymbol{B}_2 &= \begin{bmatrix} 0.2 & 0 \\ 0 & 0.2 \end{bmatrix} \\
\boldsymbol{C}_1 &= \begin{bmatrix} 2 & 1 \end{bmatrix}, & \boldsymbol{C}_2 &= \begin{bmatrix} 3 & -2 \end{bmatrix} \\
\boldsymbol{L}_{12} &= \begin{bmatrix} -1 & 2 \end{bmatrix}, & \boldsymbol{L}_{21} &= \begin{bmatrix} -1 & 2 \end{bmatrix}
\end{aligned}$$

and an m-step delay sharing pattern.We apply the suggested controller in the following cases:

a) a controller with $m = 1$ (see Figs. 7 and 8), and
b) a controller with $m = 4$ (see Figs. 9 and 10).

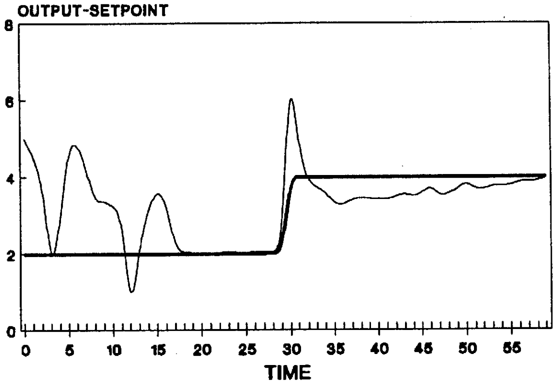

Fig. 5. Subsystem 2: output-setpoint trajectories.

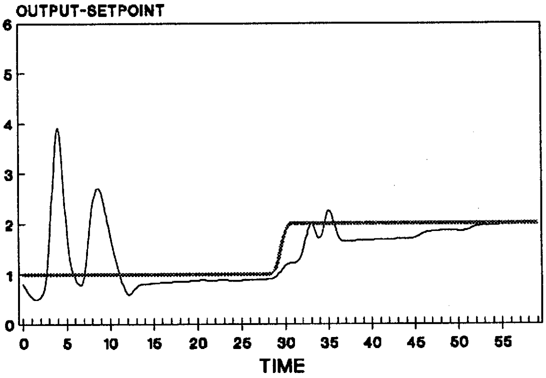

Fig. 6. Subsystem 3: output-setpoint trajectories.

The parameter values $L_y = 10$, $L_u = 3$, $L_0 = 1$, $a = 0.1$, initial $p = 3$, $\boldsymbol{Q}_i = \boldsymbol{I}_i$ and $\boldsymbol{R}_i = 10^{-2}\boldsymbol{I}_i$ were used in both the cases. The MLP used has one output layer with one node and a hidden layer with two nodes. It is easy to see a delay which exists in the results of Case (b) in both the transient and the steady state response in comparison with Case (a).

System (iii): The system in this example consists of the two subsystems of the previous example (with $m = 4$) but with the following non-linear interconnection model:

$$z_1 = \sin\big(x_2(t)\big), \quad z_2 = x_1^2(t) - 2\cos\big(x_1(t)\big)$$

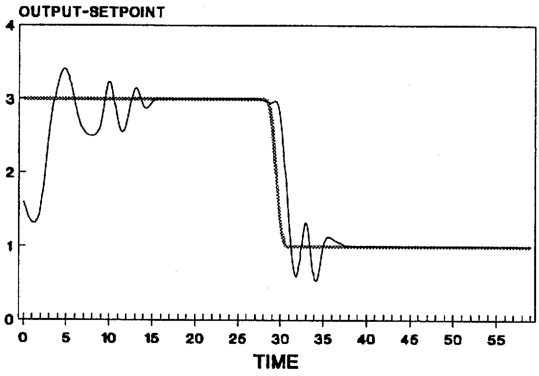

Fig. 7. Subsystem 1: output-setpoint trajectories $m = 1$.

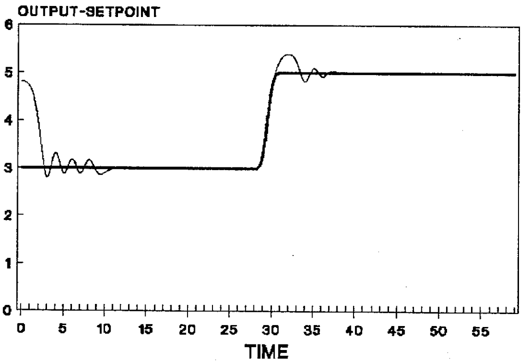

Fig. 8. Subsystem 2: output-setpoint trajectories $m = 1$.

For the controller, the parameter values $L_y = 10$, $L_u = 3$, $L_0 = 1$, $a = 0.1$, initial $p = 3$, $\boldsymbol{Q}_i = \boldsymbol{I}_i$ and $\boldsymbol{R}_i = 10^{-2}\boldsymbol{I}_i$ were used in both the cases. One can easily observe (see Figs. 11 and 12) good tracking in the steady-state response and a success of the NN in estimating the interaction trajectories governed by the non-linear model. The oscillatory behaviour in the transient response is again caused by the delay of the information flow.

In almost all cases, a small wave effect is observed every time a peak is achieved, probably due to the inherited conflicts between the subsystems when a steady state is brought about. A way suggested to improve the transient response is the known technique of the parallel introduction and the use of a standard decentralized PID controller when there is a serious deviation of the outputs from their setpoints and the learning period becomes a critical factor.

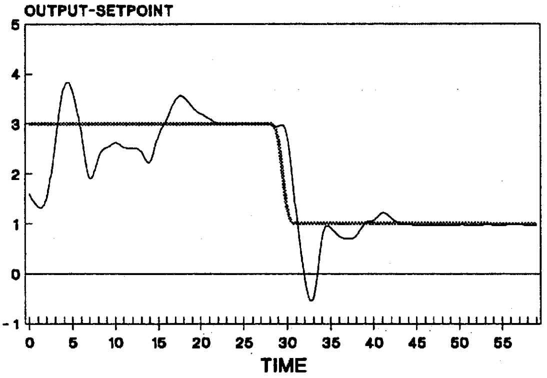

Fig. 9. Subsystem 1: output-setpoint trajectories $m = 4$.

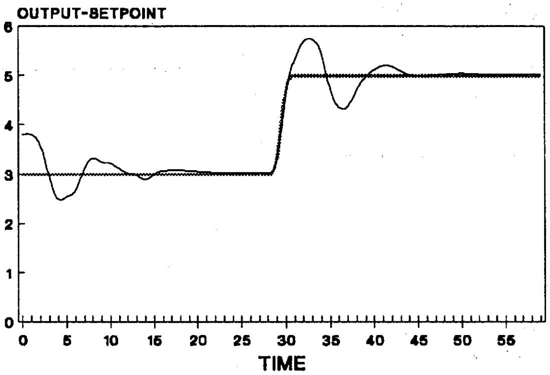

Fig. 10. Subsystem 2: output-setpoint trajectories $m = 4$.

6. Conclusion

The "optimal" control $u_i(t) = u_i^l(t)+u_i^{nl}(t)$ is optimal in the sense of decentralization, but it remains suboptimal in comparison with the solution which would be attainable if the whole information was available to every control station. The proposed decentralized controller depends weakly on the initial conditions and strongly on the approximate model used for the interconnections. The last dependence seems to be reduced by the on-line neural estimation for the predictions of the interconnections based on their approximate model.

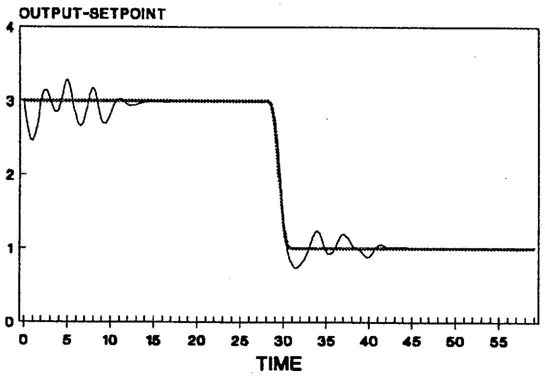

Fig. 11. Subsystem 1: output-setpoint trajectories (nonlinear interactions).

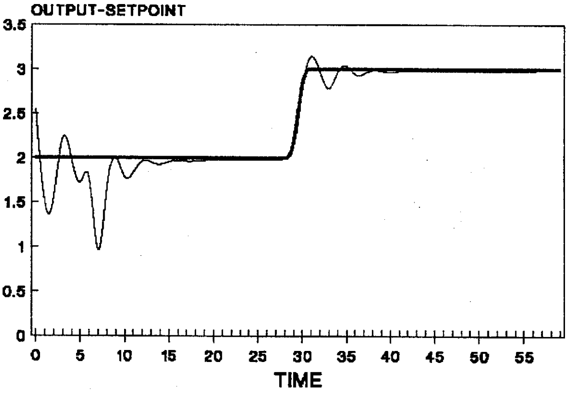

Fig. 12. Subsystem 2: output-setpoint trajectories (nonlinear interactions).

The neural network used is an MLP trained with a modified version of the backpropagation algorithm. The term added to the standard energy function to be minimized increases the plasticity and the generalization capability of the network by reducing the variance of the network output. This reduces the oscillatory behaviour during the transient response, but does not completely eliminate it because of the inherited delay in the information flow of the system. Moreover, the system reaches more quickly its steady state in comparison with the method presented in (Tzafestas *et al.*, 1995) where an ARMA linear model for the prediction of the interconnections was used and the parameter values were updated via a least-squares algorithm.

The approach presented in the paper covers all the cases of large-scale systems with linear subsystems and linear or non-linear interactions processes. These systems are encountered in many engineering applications (Singh and Titli, 1977; Tzafestas, 1989; Tzafestas and Hassan, 1986).

References

Bahnasawi A.A., Al-Fuhaid A.S. and Mahmoud M.S. (1990): *Decentralized and hierarchical control of interconnected systems.* — Proc. IEE, Vol.D-137, pp.311–21.

Clarke D.W. (1990): *Generalized predictive control and its application.* — Proc. CIM-Europe Workshop *Computer Integrated Design of Controlled Industrial Systems*, Brussels, April 26–27, pp.57–75.

Corfmat J.P. and Morse A.S. (1976): *Decentralized control of multivariable systems.* — Automatica, Vol.12, pp.479–95.

De Keyser R.M.C. (1990): *Model based predictive control toolbox* . — Proc. CIM-Europe Workshop *Computer Integrated Design of Controlled Industrial Systems*, Brussels, April 26–27, pp.35–56.

Hecht-Nielsen R. (1990): *Neurocomputing.* — Addison-Wesley.

Ho Y.C. and Chu K.C. (1974): *Information structure in many-person optimization problems.* — Automatica, Vol.10, pp.49–60.

Jamshidi M. (1996): *Large Scale Systems: Modeling, Control and Fuzzy Logic.* — Prentice Hall.

Kurtaram B. and Sivan R. (1974): *LQG control with one-step-delay sharing pattern.* — IEEE Trans. Automat. Contr., Vol.AC-19, pp.571–574.

Linnemann A. (1984): *Decentralized control of dynamically interconnected systems.* — IEEE Trans. Autom. Cotr., Vol.AC-29, pp.1052–1054.

Richalet J. (1990): *Model based predictive control in the context of integrated design.* — Proc. CIM-Europe Workshop *Computer Integrated Design of Controlled Industrial Systems*, Brussels, April 26–27, pp.3–34.

Rumelhart D.E., Hinton G.E. and Williams R.J. (1989): *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol.1.* — Cambridge, MA: MIT Press.

Sandell N.R. and Athans M. (1974): *Solution of some non-classical LQG stochastic decision problems.* — IEEE Trans. Automat. Control, Vol.AC-19, pp.108–116.

Singh M.G and Titli A. (Eds.) (1977): *Handbook of Large Scale Systems Engineering Applications.* — Amsterdam: North Holland.

Singh M.G. and Titli A. (1978): *Systems Decomposition, Optimization and Control.* — Oxford: Pergamon.

Tzafestas S.G. (1989): *Impact of distributed-parameter systems theory on the computer control of modern life systems*, In: Distributed Parameter Systems: Modelling and Simulation (T. Futagami, S.G. Tzafestas and Y. Sunahara, Eds.). — Amsterdam: North Holland, pp.3–15.

Tzafestas S.G., Dalianis P. and Anthopoulos G. (1996): *On the overtraining phenomenon of back-propagation neural networks.* — IMACS Trans. Mathematics and Computers in Simulation, Vol.40, No.5–6, pp.507–521.

Tzafestas S.G and Hassan M.F. (1986): *Complex large scale systems methodologies in conjunction with modern computer technology,* In: Control: Theory and Advanced Technology (Eds.). — Vol.2, pp.105–130.

Tzafestas S., Kyriannakis E. and Kapsiotis G. (1995): *Decentralized model based predictive control of large scale systems.* — Proc. 3rd IEEE Mediterranean Symp. *New Directions in Control and Automation*, Cyprus, Vol.II, pp.275–284.

Yoshikawa T. and Kobayashi H. (1978): *Separation of estimation and control for decentralized stochastic control systems.* — Automatica, Vol.14, pp.623–628.

Received: 18 November 1996
Revised: 4 July 1997