INTERNATIONAL JOURNAL OF ADAPTIVE CONTROL AND SIGNAL PROCESSING
Int. J. Adapt. Control Signal Process. 2003; **17**: 417–430 (DOI: 10.1002/acs.759)

Intelligent control using multiple neural networks

Lingji Chen[1,*] and Kumpati S. Narendra[2]

[1] *Scientific Systems Company Inc., 500 West Cummings Park, Suite 3000, Woburn, MA 01801, USA*
[2] *Center for Systems Science, Faculty of Engineering, Yale University, P.O. Box 208267, New Haven, CT 06520-8267, USA*

SUMMARY

In this paper a framework for intelligent control is established to adaptively control a class of non-linear discrete-time dynamical systems while assuring boundedness of signals. A linear robust adaptive controller and multiple non-linear neural network based adaptive controllers are used, and a switching law is suitably defined to switch between them, based upon their performances in predicting the plant output. Boundedness of signals is established with minimum requirements on the parameter adjustment mechanisms of the neural network controllers, and thus the latter can be used in novel ways to better detect changes in the system being controlled, and to initiate fast adaptation. Simulation studies show the effectiveness of the proposed approach. Copyright © 2003 John Wiley & Sons, Ltd.

KEY WORDS: intelligent control; non-linear control; adaptive control; neural networks; multiple models; switching systems; stability; robustness; performance

1. INTRODUCTION

Terms like ''intelligent control'' are generally imprecise, and the ideas that they are intended to convey cannot be compressed into simple statements without vital loss of content. Yet, such terms are useful in that they help us to communicate, and numerous articles have appeared in recent years, in the adaptive control literature, with these terms in their titles [1–6]. In this paper, we shall imply by *intelligent control systems* those that can operate successfully in environments with greater uncertainty and more rapid time-variations than those encountered in conventional adaptive systems.

Since environments can change rapidly, the objective is to achieve fast and sufficiently accurate adaptation. Multiple Models, Switching, and Tuning (MMST), a methodology proposed in Reference [7], provides a natural framework for the design of such systems. Many models of the plant are used in parallel as predictors of the plant output, and the one that performs the best at any instant, in a well defined sense, is chosen at that instant to generate the

*Correspondence to: Dr. Lingi Chen, Scientific Systems Company Inc., 500 West Cummings Park, Suite 3000, Woburn, MA 02148, USA.

Contract/grant sponsor: NSF; contract/grant number: ECS-0113239

 Accepted 29 May 2003

control input. The rationale used to justify the scheme is that if the models are so located that one of them is close to the plant at every instant, switching and tuning would result in rapid and accurate adaptation. However, a critical issue in using this approach is stability, and the design of such systems can be undertaken with confidence only when their stability properties are well understood.

The dynamical plant to be controlled can operate in continuous or discrete time, can be deterministic or stochastic, and can be linear or non-linear. In this paper our interest is in deterministic nonlinear systems operating in discrete time, which are controlled by neural networks. Recent years have witnessed much progress in the stability analysis of neural network based control systems [8–11]. Structural knowledge of the underlying system is usually exploited in these papers to derive suitable network tuning algorithms. At the same time, the stability of multiple model based adaptive control has been extended systematically to linear deterministic [7], linear stochastic [12], and special classes of nonlinear systems [13]. Our objective here is to extend the above results to the control of systems in which multiple neural networks are used. The principal question addressed is how neural networks can be introduced and tuned online to improve performance without causing instability.

From a practical point of view, neural networks are often used in conjunction with an existing linear controller, to improve performance. Recently, an approach was proposed in Reference [14], in which a controller made up of a robust linear controller and a neural network based controller was shown to be stable. This concept is extended in this paper to the case where multiple neural networks are present. The stability issues that arise, and the manner in which the certainty equivalence control input is computed form the main contributions. How the control scheme fits into the general methodology described in Reference [7], and how it extends its scope, are discussed towards the end of the paper.

The paper is organized as follows. A statement of the problem is given in Section 2. The stability result is presented in Section 3. The issue of generating certainty equivalence control input is re-examined in Section 4, which leads to some relaxed conditions on the switching system. A variety of ways of using multiple models is briefly discussed in Section 5. A simulation study is presented in Section 6, and conclusions are drawn in Section 7. Some mathematical tools used throughout the paper are included in Appendix A for easy reference.

2. STATEMENT OF THE PROBLEM

Let a single-input single-output discrete-time non-linear dynamical system be described by the state equations

$$\Sigma : x(k+1) = F(x(k), u(k))$$
$$y(k) = H(x(k)) \tag{1}$$

where $u(k), y(k) \in \mathbb{R}$ are the input and output respectively, $x(k) \in \mathbb{R}^n$ the state, and F and H analytic functions that are unknown. It is further assumed that $F(0,0) = 0$ and $H(0) = 0$, so that the origin is an equilibrium state. Qualitatively, the adaptive control problem is to determine the input control law so that all signals in the system remain bounded, while the output $y(k)$ behaves in some desired fashion. This problem has been considered by many authors in the neurocontrol literature, and a precise formulation of the theoretical questions that arise, and methods for addressing them, can be found in Reference [15]. The results that were obtained are local in nature; they are based on the properties of the linearized system $\sum_L$ around the equilibrium state, and are derived using the Implicit Function Theorem. It was consequently proposed in References [16, 17] that the results can be made transparent by expressing Σ in the form

$$\Sigma : x(k+1) = Ax(k) + bu(k) + F_1(x(k), u(k)) \\ y(k) = cx(k) + H_1(x(k)) \tag{2}$$

where A, b, c are, respectively $\partial f/\partial x|_{(0,0)}$, $\partial f/\partial u|_{(0,0)}$, and $\partial h/\partial x|_{(0)}$, and hence matrices and vectors of appropriate dimensions. $F_1(\cdot)$ and $H_1(\cdot)$ (defined as "higher order functions") contain higher order terms in $x(k)$ and $u(k)$, and the roles they play in establishing system theoretic properties of Σ as well as in solving various control problems, have been investigated in References [16, 17]. For the case when only the input and output of Σ are accessible, our interest is in the input–output representation Σ_{IO} of Σ. It has been shown that Equation (2) can also be expressed (locally) as

$$\Sigma_{IO} : y(k+d) = a_0 y(k) + \ldots + a_{n-1} y(k-n+1) + b_0 u(k) + \ldots + b_{n-1} u(k-n+1) \\ + f(y(k), \ldots, y(k-n+1), u(k), \ldots, u(k-n+1)) \tag{3}$$

$$\triangleq \theta^{\mathrm{T}} \omega(k) + f(\omega(k)) \tag{4}$$

where d is the relative degree (delay) of the system, $\theta^{\mathrm{T}} = [a_0, \ldots, a_{n-1}, b_0, \ldots, b_{n-1}]$ is the (linear) parameter vector, and $\omega(k) = [y(k), \ldots, y(k-n+1), u(k), \ldots, u(k-n+1)]$ is the regression vector. From the previous assumptions, $f(\cdot)$ can be shown to be a smooth nonlinear function consisting only of higher order terms in $\omega(k)$. It is assumed that the order n of the system, and its delay d are known while the parameter vector θ and the function $f(\cdot)$ are unknown.

Asymptotic tracking: Given a bounded signal $y^*(k)$ which represents the desired output, the objective of control is to generate a bounded control input $u(k)$ such that $\lim_{k\to\infty} |y(k) - y^*(k)| = 0$.

Adaptive control: Since $f(\cdot)$ and θ are unknown, the problem is one of adaptive control. Assuming that $y^*(k+d)$ is known at time k, the problem is to estimate θ and $f(\cdot)$ and use the estimates in turn to determine the control input $u(k)$.

In Reference [15] it was shown that if (i) Σ_L is observable, (ii) the equilibrium state of Σ is asymptotically stable, and (iii) Σ has a well defined relative degree and zero dynamics that is exponentially stable, then a controller can in theory be synthesized to achieve asymptotic tracking.

Robust adaptive control: It is well known that in any physical system, external and internal disturbances are invariably present. If adaptive control is used in such systems, the control laws have to be suitably modified so that all signals in the system remain bounded and the overall system is robust.

MMST using neural networks: Our objective in this paper is to establish a framework for the design of neural network based adaptive controllers for non-linear systems. As stated in the introduction, an MMST methodology has proved very successful in the adaptive control of linear systems, and an attempt is made here to extend the same methodology to nonlinear systems. As in the linear case, multiple estimation models are used to estimate θ and $f(\cdot)$ in Equation (4), and their parameters are adaptively adjusted online. At every instant, when the best performing model is chosen according to some criterion, it gives rise to a controller that computes the control input $u(k)$ by treating the model as if it were the unknown plant.

Stability and robustness are the fundamental considerations when designing adaptive controllers. While assuring stability for linear plants is a difficult task, it becomes substantially more so in the non-linear case. In the following sections we propose a method by which linear and nonlinear controllers are judiciously combined to assure robust adaptive control with the possibility of improved performance. Needless to say, assumptions have to be made concerning the system Σ and the non-linear function $f(\cdot)$ to assure these properties. They are listed below:

(i) Its input-output representation Σ_{IO} is global.
(ii) The nonlinearity $f(\cdot)$ is globally bounded, i.e. $|f(\cdot)| \le L$, and the bound L is known.
(iii) The system Σ_{IO} has asymptotically stable zero dynamics and the input sequence cannot grow faster than the output sequence.†
(iv) Σ_{L} the linearized system is controllable and observable, and the parameter θ belongs to a known compact set.

The approach: Since the nonlinearity f is bounded, system Σ_{IO} can be considered as a linear system with a bounded disturbance. A linear robust adaptive controller with a dead-zone in the adaptive law can be designed so that the error $|y(k) - y^*(k) \triangleq |e(k)|$ remains bounded.

In the second stage, to improve performance, neural networks are used to estimate and compensate for the nonlinearity. Multiple neural networks are used to assure rapid adaptation. Assuring boundedness of the signals and discussing the conditions under which the overall performance can be improved form the main contributions of the paper.

3. THE SWITCHING SYSTEM

As stated in the previous section, the first step in the design procedure is the synthesis of a control input, assuming a linear model, which assures a bounded error. To highlight the essence of the proof, we analyse the following system when the delay is unity

$$y(k+1) = \theta^{\mathrm{T}}\omega(k) + f(\omega(k)) \triangleq b_0 u(k) + \theta^{\mathrm{T}}\bar{\omega}(k) \tag{5}$$

where $|f(\cdot) \le L$, and the component b_0 in θ satisfies $b_0 \ge b_m > 0$ where b_m is a known constant. The regression vector and the parameter without the $u(k)$ term and its coefficient respectively are denoted by a bar.

3.1. The estimation models and the corresponding controllers

3.1.1. The linear estimation model

To estimate θ in Equation (5), a model described by the equation

$$\hat{y}_1(k+1) = \hat{\theta}_1^{\mathrm{T}}(k)\omega(k) \tag{6}$$

† For a characterization of "minimum phase" for non-linear systems, readers are referred to Reference [18].

is used, where $\hat{\theta}_1(k)$ is the estimate of θ, and $\hat{y}_1(k)$ the corresponding estimate of $y(k)$. The parameter $\hat{\theta}_1(k)$ is updated using the estimation error $e_1(k) \triangleq \hat{y}_1(k) - y(k)$ as

$$\hat{\theta}_1(k) = \hat{\theta}_1(k-1) - \frac{\alpha(k)e_1(k)\omega(k-1)}{1 + |\omega(k-1)|^2} \tag{7}$$

where

$$\alpha(k) = \begin{cases} 1 & \text{if} |e_1(k)| > 2L \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

and $|\cdot|$ denotes Euclidean norm. The parameter $\hat{b}_{10}(k)$ as a component of $\hat{\theta}_1(k)$ is always constrained to be greater than or equal to $b_m > 0$. The above model has been investigated in detail and the reader is referred to Reference [19]. It can be shown that (i) $\hat{\theta}(k)$ is bounded, and (ii) $\lim_{k\to\infty} a(k)(e^2(k) - 4L^2)/(2(1 + |\omega(k-1)|^2)) \to 0$. It then follows that $e(k)$ grows slower than $\omega(k-1)$ if the latter is unbounded. The formal definition of growth rates of signals and their properties are given in Appendix A; for more details readers are referred to Reference [20].

The control input $u(k)$ at instant k is computed as

$$u_1(k) = \frac{1}{\hat{b}_{10}(k)}\left(y^*(k+1) - \hat{\bar{\theta}}_1(k)^{\mathrm{T}}\bar{\omega}(k)\right)$$

Comment: If all the signals in the system can be shown to be bounded, it would then follow that the estimation error $e_1(k)$ of the linear model would satisfy the inequality $\lim_{k\to\infty}|e_1(k)| \leq 2L$. From Equation (8) it follows that $\lim_{k\to\infty}\alpha(k) = 0$, so that adaptation would stop for this model.

3.1.2. The neural network estimation models

The performance of a controller based on a linear model may not be satisfactory if L is large. To improve performance, non-linear models will be needed which will approximate the plant Σ more accurately. This would result in a smaller estimation error and consequently (as justified later) a smaller control error. With this in view, $(m-1)$ nonlinear models are set up in parallel with the linear model as shown in Figure 1.

The model M_j is described by the equation

$$M_j : \hat{y}_j(k+1) = \hat{\theta}_j^{\mathrm{T}}(k)\omega(k) + \hat{f}_j(\bar{\omega}(k), W_j(k)), \quad j = 2, \ldots, m \tag{9}$$

where $\hat{f}_j(\cdot)$ is a *bounded* continuous non-linear function parameterized by a vector $W_j(k)$ (i.e. a neural network with a weights vector W_j). At this stage no restriction is placed on the manner in which the parameter vector $\hat{\theta}_j(k)$ or $\hat{W}_j(k)$ are to be updated, except that they belong to a predefined compact set $\mathcal{S}_j$:

$$\hat{\theta}_j(k), W_j(k) \in \mathcal{S}_j \tag{10}$$

Here $\mathcal{S}_j$ is chosen in such a way that the coefficient b_{j0} associated with $u(k)$ is always greater than b_m.

At every instant k, if model M_j is chosen by the switching system, a control input $u_j(k)$ is generated so that the output of the model, $\hat{y}_j(k+1)$, is made equal to $y^*(k+1)$ at the next instant. This is referred to as the "certainty equivalence control" of the model M_j. While the existence of $u_j(k)$ is assured by the assumptions made [14], the computation of $u_j(k)$ involves the

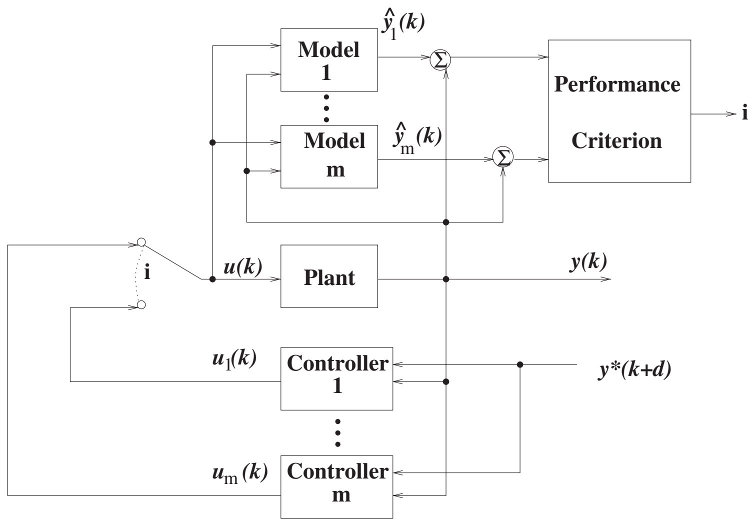

Figure 1. Multiple models: switching between a linear controller and a neural network controller.

solution of a non-linear algebraic equation and is non-trivial. Consequently, methods for obtaining approximate solutions have to be resorted to, and these are presented in Section 4.

3.1.3. The switching criterion

At every instant k, the m models described earlier update their parameter and function estimates and yield m different estimation errors. The next objective is to determine a switching criterion to switch between the models/controllers so that tracking performance is improved while robustness is maintained.

Towards this end, a performance criterion $J_i(k)$ for model M_i is defined as

$$J_i(k) = \sum_{l=1}^{k} \frac{a_i(l)(e_i^2(l) - 4L^2)}{2(1 + |\omega(l-1)|^2)} + c \sum_{l=k-N+1}^{k} (1 - a_i(l))e_i^2(l), \quad i = 1, 2, \ldots, m \tag{11}$$

The performance indices consist of two parts. The first part deals with estimation errors whose absolute values are larger than $2L$. It is mainly used to distinguish between signals with different growth rates, so that boundedness of the signals can be established. The second part is a measure of the prediction error over a finite window and is included to improve performance.

At every instant k, if model M_i yields the smallest[‡] performance criterion $J_i(k)$, then its associated certainty equivalence control input $u_i(k)$ will be used to control the system, i.e. we set $u(k) = u_i(k)$. A block diagram of the switching system is shown in Figure 1.

Comment: Unlike earlier papers on MMST, we have in the present case a linear controller which can control the plant in a robust fashion. The objective is therefore to improve system performance using an appropriate non-linear controller. The problem to be addressed is whether introducing non-linear controllers at random instants of time, based on the switching criterion, can de-stabilize the overall system.

[‡] If a tie occurs, any tie-breaking rule suffices.

3.2. The boundedness result

Theorem 1

For the switching system described above, all signals in the plant (5), the models (6) and (9), and the parameter adaptations (7) and (10), are bounded[§].

Proof

We will prove the theorem by contradiction. Assume that $y(k)$ is unbounded.

(i) By certainty equivalence, $u(k)$, at any instant k , is chosen such that for some model M_j,
$$\hat{y}_j(k+1) = y^*(k+1)$$
and therefore the control error is always equal to the identification error of that model, i.e.
$$e_c(k) \triangleq y^*(k) - y(k) = \hat{y}_j(k) - y(k) \triangleq e_j(k)$$
If corresponding to every instant k, the model M_j that is used to compute the control input is known, a virtual identification error sequence $\{e_v(k)\}$ can be defined such that
$$e_v(k) = e_j(k) = e_c(k) \tag{12}$$
at any instant. Our objective is to examine the growth rates of the virtual identification error and the control error and arrive at a contradiction, as in the single model case.

(ii) By the properties of the linear robust model, $J_1(k)$ is always bounded. (Note that the second term in (11) is always bounded for all models.) By the switching rule, there exists a finite time k_0 such that for $k > k_0$, switching occurs only between the models with bounded performance indices $J_i(k)$. This means that the virtual identification error $e_v(k)$ has the property
$$\lim_{k\to\infty} \frac{a(k)(e_v^2(k) - 4L^2)}{2(1 + |\omega(k-1)|^2)} \to 0 \tag{13}$$
Since $y(k)$ is assumed to be unbounded, so is the regression vector $\omega(k)$, and Equation (13) implies that $e_v(k)$ grows slower than $\omega(k-1)$.

(iii) By the assumptions that the zero dynamics of Σ_{IO} is asymptotically stable, and that the input sequence $u(k-1)$ cannot grow faster than the output sequence $y(k)$, it follows that the regression vector
$$\omega(k-1) = [y(k-1), \ldots, y(k-n), u(k-1), \ldots, u(k-n)]^T$$
cannot grow faster than $y(k)$. Therefore, from Equation (13), we conclude that $e_v(k)$ grows slower than $y(k)$.

(iv) Since $y^*(k)$ is bounded, $y(k)$ is equivalent to the control error $e_c(k)$, and hence is equivalent to the virtual identification error $e_v(k)$. But on the other hand, we have just established that the virtual identification error $e_v(k)$ grows slower than $y(k)$. This cannot happen if $y(k)$ is unbounded. Thus $y(k)$ is bounded, and the boundedness of other signals follows. □

[§]The only signals that can become unbounded are the performance indices $J_j(k)$, $j = 2, \ldots, m$. It is possible to introduce forgetting factors to deal with this problem.

Comment: In the argument given above, no assumption is made explicitly about the stability of the non-linear identification models. The robustness of the linear model, and the fact that at every instant the control input is chosen to make the output of the best performing model to be equal to the desired output, form the core of the boundedness proof.

The same arguments can be carried over to the case where the delay d is greater than unity. Interested readers are referred to Reference [14] for details.

4. MODIFICATIONS ON CERTAINTY EQUIVALENCE CONTROL

In the proof given in Section 3, we have assumed that a certainty equivalence control input $u_j(k)$ associated with some model M_j can always be computed so that

$$\hat{y}_j(k+1) = y^*(k+1) \tag{14}$$

This is straightforward for the case when $u(k)$ appears linearly in the model, but is non-trivial for other cases. Since the existence of such an input is guaranteed, we can use a search procedure, which iterates on the model (but not on the plant), to find the desired value of $u_j(k)$. However, given a finite time period and finite computing power, an exact solution may not be found at all.

This difficulty is not fundamental, if we examine the role of the equality (12) more carefully. The proof of boundedness relies on comparison of growth rates of unbounded signals, and therefore it remains essentially the same if we add bounded terms to these signals. Thus we have the following two corollaries.

Corollary 1

Let $\varepsilon > 0$ be an arbitrarily pre-defined constant. Theorem 1 still holds if every certainty equivalence control input $u_i(k)$ is such that

$$\hat{y}_i(k) = y^*(k) + \varsigma_i(k), \quad |\varsigma_i(k)| < \varepsilon, \quad \forall k$$

According to this corollary, the search for the certainty equivalence control input $u(k)$ can terminate after a certain level of accuracy has been achieved. This affects the performance improvement that can be gained, but not the boundedness of the signals.

If the pre-determined accuracy cannot be achieved within the time allocated for the computation, we can always revert to the linear robust adaptive controller. This is formally stated as the next corollary.

Corollary 2

Theorem 1 still holds if the switching rule is changed to

$$u(k) = u_i(k), \quad \text{where either } J_i(k) \text{ is the smallest, or } i = 1$$

According to this corollary, at each instant of time, either the best model or simply the linear model can be chosen to generate a control input. This affects tracking performance, but it does not affect boundedness of signals.

The implications of the above two corollaries for controller design are as follows. At any instant k, the output of the system $y(k)$ is first measured, and then an input $u(k)$ to the system is computed. Although the same index k is used for the two, there is invariably a delay between $y(k)$ and $u(k)$. This is true even for a linear system where the computation of $u(k)$ is straightforward. Assume that this computational time is bounded above by T_L. Also assume that, by conducting robustness analysis, we determine that the system can tolerate a delay T_d in which computation of $u(k)$ can be carried out. Our strategy is to utilize a time interval T_d–T_L for computing a non-linear certainty equivalence control input $u(k)$. If such a value is successfully found within certain tolerance, it can be used to control the system. If not, we can then compute the linear control input $u(k)$ in the remaining time interval T_L, and use that to control the system. More specifically, we adopt the following procedure:

1. Choose a constant $\varepsilon > 0$.
2. At each time instant, after $y(k)$ becomes available, start computing the certainty equivalence control input $u_j(k)$, by using a non-linear optimization procedure for non-linear models.
3. Stop the computation after a time of length T_d–T_L has elapsed. Examine the smallest error $|\hat{y}_j(k+1) - y^*(k+1)|$ achieved in the optimization procedure:
   - If it is smaller than ε, use the corresponding $u_j(k)$ as a control input to the system.
   - Otherwise, compute and use the linear control input $u_1(k)$.
4. Go to 2.

5. CHOICE OF NON-LINEAR MODELS

The fact that very few assumptions were made concerning the non-linear models in Section 3 provides us with adequate freedom to structure these suitably, depending upon the prior information available concerning the plant. For example, they can be merely

(i) linear models, or
(ii) non-linear auto-regressive moving average (NARMA) models which take the form

$$y(k+d) = F(y(k), y(k-1), \ldots, y(k-n+1), u(k), u(k-1), \ldots, u(k-n+1)), \quad \text{or}$$

(iii) approximate models given by

$$\begin{aligned} y(k+d) = {} & F_1(y(k), \ldots, y(k-n+1), u(k-1), \ldots, u(k-n+1) \\ & + G_1(y(k), \ldots, y(k-n+1), u(k-1), \ldots, u(k-n+1))u(k) \end{aligned}$$

Comment: Even though NARMA models are exact (local) representations of nonlinear systems, it is generally difficult to obtain the certainty equivalence control input $u(k)$ associated with these models, since F is a non-linear function in $u(k)$. The approximate models avoids this difficulty. Detailed discussions about models of this type can be found in Reference [21].

If we categorize the models according to their parameter adaptation mechanisms, we can have

(i) fixed models whose parameters are determined off-line and are fixed on-line,
(ii) adaptive models whose parameters are adaptive as in conventional adaptive control, or
(iii) re-initialized adaptive models whose parameters are adaptive, and re-initialized each time when some fixed model is chosen by the switching rule.

For more detailed discussion of these models, readers are referred to Reference [7].

6. SIMULATIONS

To illustrate the effectiveness of the proposed framework for intelligent control using multiple neural networks, a simulation study is described in this section. The objective is to control a plant so that it follows the reference trajectory given by

$$r(k) = 1.5\left(\sin\left(\frac{2\pi k}{10}\right) + \sin\left(\frac{2\pi k}{25}\right)\right)$$

The simulation is carried out from $k = 1$ to $k = 400$. At $k = 200$, an abrupt change is introduced to the plant so that it switches from P_1 to P_2 where P_1 is described by

$$\begin{aligned} P_1 : y(k+1) &= 2.6y(k) - 1.2y(k-1) + u(k) + 0.5u(k-1) \\ &\quad + \sin((u(k-1) + y(k) + y(k-1)) - \frac{u(k-1) + y(k) + y(k-1)}{1 + u^2(k-1) + y^2(k) + y^2(k-1)} \end{aligned}$$

and P_2 is described by

$$\begin{aligned} P_2 : y(k+1) &= y(k) - 2y(k-1) + u(k) + 0.5u(k-1) \\ &\quad + \sin(0.9u(k-1) + y(k) + y(k-1)) - \frac{0.9u(k-1) + y(k) + y(k-1)}{1 + 0.81u^2(k-1) + y^2(k) + y^2(k-1)} \end{aligned}$$

Since the nonlinear terms are bounded, a linear robust adaptive controller is first used to control the plant. A snapshot of the performance is shown in Figure 2, with Figure 2(a) showing the system output (solid) and the desired output (dashed), while the control input is depicted in Figure 2(b). The evolution of the 4 linear parameters of the adaptive controller is shown in Figure 2(c). It can be seen that the changes in parameters have become small near $k = 200$, at which point the plant switches from P_1 to P_2, and the linear controller re-adapts itself to the new plant. The input and the output of the plant are always bounded, however, due to the presence of the non-linearities, the performance is poor. After input–output data for the plant is collected while the linear robust adaptive controller is running, the simulation proceeds as follows. First the data is separated into two sets that correspond to $k < 200$ and $k \geq 200$, respectively. For each set, a linear model and a non-linear model are extracted. The former is of the form

$$\hat{y}(k+1) = a_0 y(k) + a_1 y(k-1) + b_0 u(k) + b_1 u(k-1)$$

where the parameters are obtained simply by solving a matrix equation. The latter is of the form

$$\hat{y}(k+1) = a_0 y(k) + a_1 y(k-1) + b_0 u(k) + b_1 u(k-1) + f_{NN}(y(k), y(k), y(k-1), u(k-1))$$

where the linear parameters are the same as the former, and the neural network parameters are obtained by training. The multilayer perceptron neural network used had two hidden layers with 10 and 8 nodes, respectively. (The whole neural network structure is sometimes denoted by $\mathcal{N}_{3,10,8,1}$). After this, the following models are constructed and used in the control

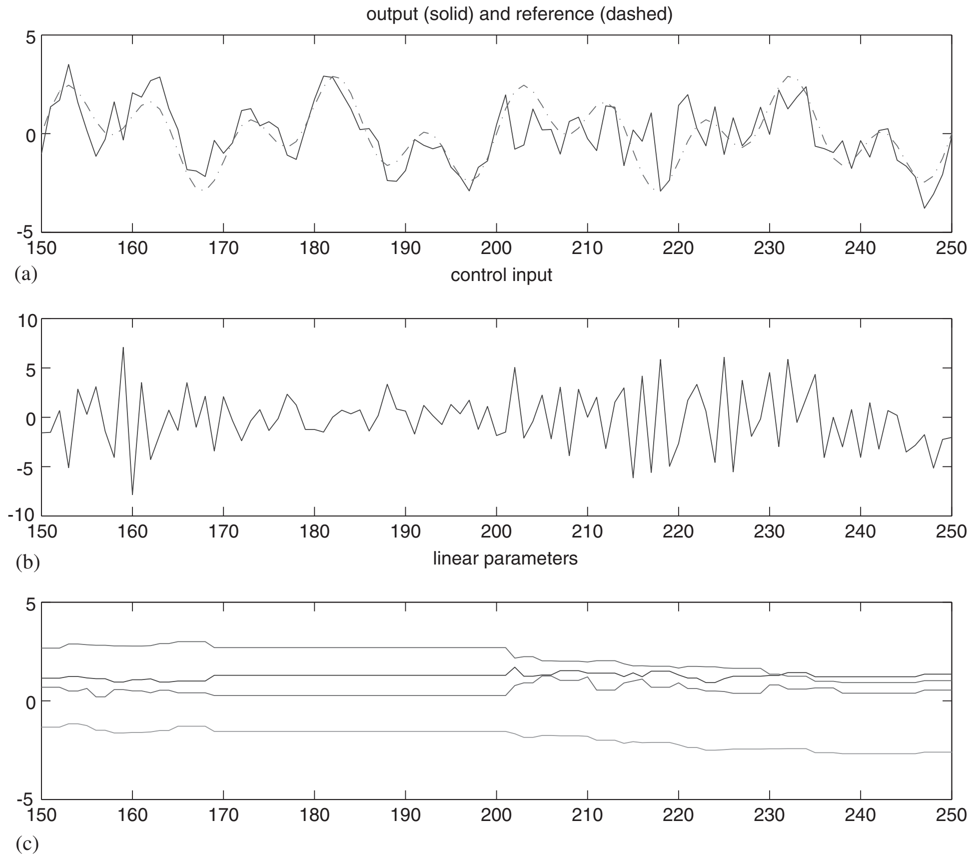

Figure 2. Performance of the linear robust adaptive controller.

of the plant:

(i) A free running linear robust adaptive controller, denoted by C_1.
(ii) Two linear controllers with fixed parameters, denoted by C_2 and C_3 respectively.
(iii) Two non-linear controllers with fixed parameters, denoted by C_4 and C_5 respectively.
(iv) Two non-linear adaptive controllers whose initial parameters are the same as in C_4 and C_5, respectively. They are referred to as C_6 and C_7.

Simulations have been conducted using (i) C_1 only, (ii) C_1, C_2 and C_3, (iii) C_1, C_2, C_3, C_4 and C_5, and (iv) C_1, C_2, C_3, C_4, C_5, C_6 and C_7. The addition of a model usually improves the performance to some extent. When all 7 models are used, the performance is shown in Figure 3. The window size N in the switching criterion (11) was chosen as 4 in the simulations. It can be seen that, due to the success of extensive off-line training, switching occurs mostly between the fixed non-linear model C_4 and the adaptive non-linear model C_6 when the plant is P_1, and between C_5 and C_7 when the plant is P_2, and the tracking error is very small, even when the plant undergoes an abrupt change.

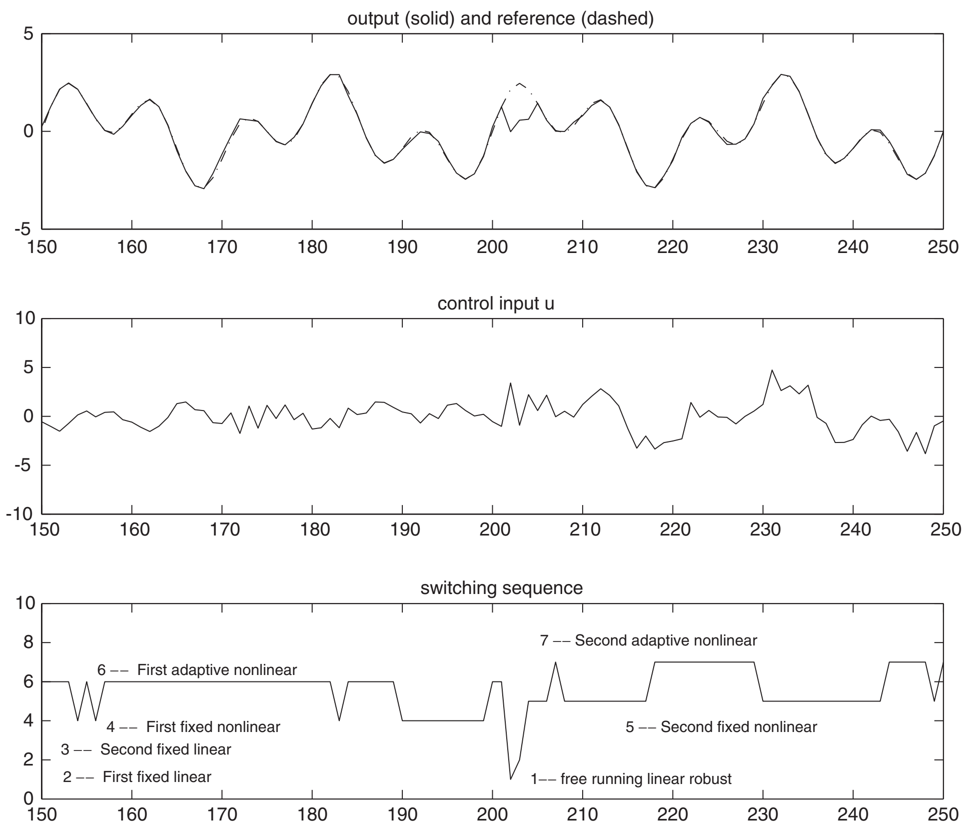

Figure 3. Performance when using all 7 models.

7. CONCLUSIONS

In this paper a framework for intelligent control is established to adaptively control a class of non-linear discrete-time dynamical systems while assuring boundedness of the signals. One linear robust adaptive controller and $(m-1)$ nonlinear neural network based adaptive controllers are used, and a switching law is suitably defined to switch between them, based upon their performance in predicting the plant output. Boundedness of the signals is established regardless of the parameter adjustment mechanism of the neural network controllers, and thus neural network models can be used in a variety of ways to better detect changes in the system and initiate adaptation. Simulation studies show the effectiveness of the proposed approach, which appears to have considerable potential for the intelligent control of time-varying systems.

APPENDIX A: GROWTH RATES OF SIGNALS

One of the principal tools used in proving boundedness of signals in an adaptive system is the relationship between different signals in the system and their rates of growth. In this section a review of growth rates of signals [20] is presented. The properties listed here were used implicitly throughout the paper.

Let $x(k)$ and $y(k)$ (scalar or vector) be two discrete time signals defined for all $k \in \mathbb{N}^+$ where $\mathbb{N}^+$ is the set of all non-negative integers. Let $|\cdot|$ denote a norm.

Definition 1 (large order)

We denote $y(k) = O[x(k)]$ if there exist positive constants M_1, M_2 and such that $|y(k)| \leqslant M_1 \max_{r \leq k} x(\tau)| + M_2, \quad \forall k \geq k_0$

Definition 2 (small order)

We denote $y(k) = o[x(k)]$ if there exists a discrete-time function $\beta(k)$ with the property that $\lim_{k \to \infty} \beta(k) = 0$, and a constant k_0 such that $|y(k)| \leq \beta(k) \max_{r \leqslant k} |x(\tau)|, \quad \forall k \geq k_0$

Definition 3 (equivalence)

If $x(k) = O[y(k)]$ and $y(k) = O[x(k)]$, we refer to $x(k)$ and $y(k)$ as being equivalent and denote it as $x(k) \sim y(k)$. If follows directly that this equivalence relation is reflexive, symmetric and transitive, so that the symbol $\sim$ represents an equivalence class.

It is straightforward to verify the following properties about growth rates of signals. (To simplify the notations, time k is dropped.)

(i) If z is bounded and x is unbounded, then $y = O[x] \Rightarrow y = O[x + z]$.
(ii) If z is bounded, then $y = O[x] \Rightarrow y + z = O[x]$.
(iii) If z_1, z_2 are bounded and x and y are unbounded, then $y \sim x \Leftrightarrow y + z_1 \sim x + z_2$.
(iv) If x is unbounded and y is bounded, then $y = o[x]$.
(v) If z is unbounded, then $y = o[x], x = O[z] \Rightarrow y = o[z]$.
(vi) If y is unbounded, then $z = O[y], y = o[x] \Rightarrow z = o[x]$.
(vii) $y_1 = o[x], y_2 = o[x] \Rightarrow y_1 + y_2 = o[x]$.
(viii) If y consists of two subsequences y_1 and y_2, and the corresponding subsequences of x is x_1 and x_2, then $y_1 = o[x_1], y_2 = o[x_2] \Rightarrow y = o[x]$.
(ix) $y = o[y] \Rightarrow \lim_{k \to \infty} y(k) = 0$.

REFERENCES

1. Astrom KJ. Where is the intelligence in intelligent control? *IEEE Control and System Magazine* 1991; **11**(1):37–39.
2. Narendra KS, Mukhopadhyay S. Intelligent control using neural networks. *IEEE Control and System Magazine* 1992; **12**(2):11–18.
3. Polycarpou MM. Fault accommodation of a class of multivariable nonlinear dynamical systems using a learning approach. *IEEE Transactions on Automatic Control* 2001; **46**(5):736–742.
4. Parisini T, Sacone S. Hybrid control based on discrete-event automata and receding-horizon neural controllers. In *Proceedings of the 2000 IEEE International Symposium on Intelligent Control*, Rio Patras, Greece, 2000; 303–308.
5. Meystel A, Messina E. The challenge of intelligent systems. In *Proceedings of the 2000 IEEE International Symposium on Intelligent Control*, Rio Patras, Greece, 2000; 211–216.
6. Diao Y, Passino KM. Intelligent fault tolerant control using adaptive schemes and multiple models. In *Proceedings of 2001 American Control Conference*, vol. 4, Artlington, VA, USA, 2001; 2854–2859.
7. Narendra KS, Balakrishnan J. Adaptive control using multiple models. *IEEE Transactions on Automatic Control* 1997; **42**(2):171–187.
8. Lemmon MD, Michel AN. Guest editorial: special section on neural networks in control, identification, and decision making. *IEEE Transactions on Automatic Control* 1999; **44**(11):1993–1994.
9. Narendra KS, Lewis FL. Introduction to the special issue on neural network feedback control. *Automatica* 2001; **37**(8):1147–1148.
10. Kim YH, Lewis FL. *High-level feedback control with neural networks*. World Scientific: Singapore, 1999.
11. Ge SS, Hang CC, Lee TH, Zhang T. *Stable adaptive neural network control*. Kluwer Academic: Norwell, USA, 2001.
12. Narendra KS, Xiang C. Adaptive control of discrete-time systems using multiple models. *IEEE Transactions on Automatic Control* 2000; **45**(9):1669–1686.

13. Narendra KS, George K. Adaptive control of simple nonlinear systems using multiple models. In *Proceedings of the 2002 American Control Conference,* vol 3, Anchorage, Alaska, USA, 2002.
14. Chen L, Narendra KS. Nonlinear adaptive control using neural networks and multiple models. *Automatica* 2001; **37**(8):1245–1255.
15. Cabrera JBD, Narendra KS. Issues in the application of neural networks for tracking based on inverse control. *IEEE Transactions on Automatic Control* 1999; **44**(11):2007–2027.
16. Narendra KS, Chen L. Identification and control of a nonlinear dynamical system Σ based on its linearization Σ_L: Part I. *Proceedings of the 37th IEEE Conference on Decision and Control*, vol. 3, Tampa, Florida, USA, 1998; 2977–2982.
17. L Chen, Narendra KS. Identification and control of a nonlinear dynamical system based on its linearization. *Proceedings of the 2002 American Control Conference*, Anchorge, Alaska, USA, 2002; 382–387.
18. D Liberzon, Morse AS, Sontag ED. Output-input stability and minimum-phase nonlinear systems. *IEEE Transactions on Automatic Control* 2002; **47**(3):422–436.
19. Goodwin GC, Sin KS. *Adaptive Filtering, Prediction and Control*. Prentice Hall, Englewood Cliffs: New Jersey 1984.
20. Narendra KS, Annaswamy AM. *Stable Adaptive Systems*. Prentice Hall, Englewood Cliffs; New Jersey, 1989.
21. Narendra KS, Mukhopadhyay S. Adaptive control using neural networks and approximate models. *IEEE Transactions on Neural Networks* 1997; **8**(3):475–485.

