NONLINEAR INTERNAL MODEL CONTROL USING NEURAL NETWORKS: APPLICATION TO PROCESSES WITH DELAY AND DESIGN ISSUES

ISABELLE RIVALS and LéON PERSONNAZ

École Supérieure de Physique et de Chimie Industrielles, Laboratoire d'Électronique,
10 rue Vauquelin, 75231 Paris Cedex 05, France.
E-mail: Isabelle.Rivals@espci.fr, Leon.Personnaz@espci.fr

Abstract – We propose a design procedure of neural internal model control systems for stable processes with delay. We show that the design of such non adaptive indirect control systems necessitates only the training of the inverse of the model deprived from its delay, and that the presence of the delay thus does not increase the order of the inverse. The controller is then obtained by cascading this inverse with a rallying model which imposes the regulation dynamic behavior and ensures the robustness of the stability. A change in the desired regulation dynamic behavior, or an improvement of the stability, can be obtained by simply tuning the rallying model, without retraining the whole model reference controller. The robustness properties of internal model control systems being obtained when the inverse is perfect, we detail the precautions which must be taken for the training of the inverse so that it is accurate in the whole space visited during operation with the process. In the same spirit, we make an emphasis on neural models affine in the control input, whose perfect inverse is derived without training. The control of simulated processes illustrates the proposed design procedure and the properties of the neural internal model control system for processes without and with delay.

Index Terms – Affine models, internal model control, inverse model, neural networks, nonlinear systems, reference model, robustness, systems with delay, tracking.

1 INTRODUCTION

There are several ways to cope with nonlinear control design for plants subject to uncertainty and disturbances: among them are robust and adaptive control methods. While neural adaptive control is being intensively developed [1] [2] [3] [4] [5], an approach of neural internal model control and of its robustness properties is presented here. In order to precise the scope of this paper, let us first recall basic notions concerning internal model control systems, their design and the properties which make them attractive.

A *control system* consists of the process to be controlled, and of *a control device* chosen by the designer. The control device computes the control input so as to convey the desired behavior to the control system. The control device generally consists of a *controller* and possibly other elements

1

(filters, observer, internal model…). In this paper, the control device imposes the desired dynamic behavior with the help of a model reference controller, described in section 3. We distinguish between two types of control systems:

– Simple feedback control systems: their control device is made of the controller only (Fig. 1a). They are the most classic ones in the field of adaptive control (see [1] [2] [3] [5]). However, in the case of non adaptive neural control, since the quality of their performance relies heavily on the fidelity of the model used for the design of the controller, they can only be used if the model is very accurate and if the effect of disturbances is small (see for example [6] [7]).

– Internal model control systems: they are characterized by a control device consisting of the controller and of a predictive model of the process, the internal model. The internal model loop uses the difference between the outputs of the process and of the internal model, as shown in Fig. 1b; this difference represents the effect of disturbances and of a mismatch of the model. Note that in Fig. 1a and 1b, the control device is represented on a gray background.
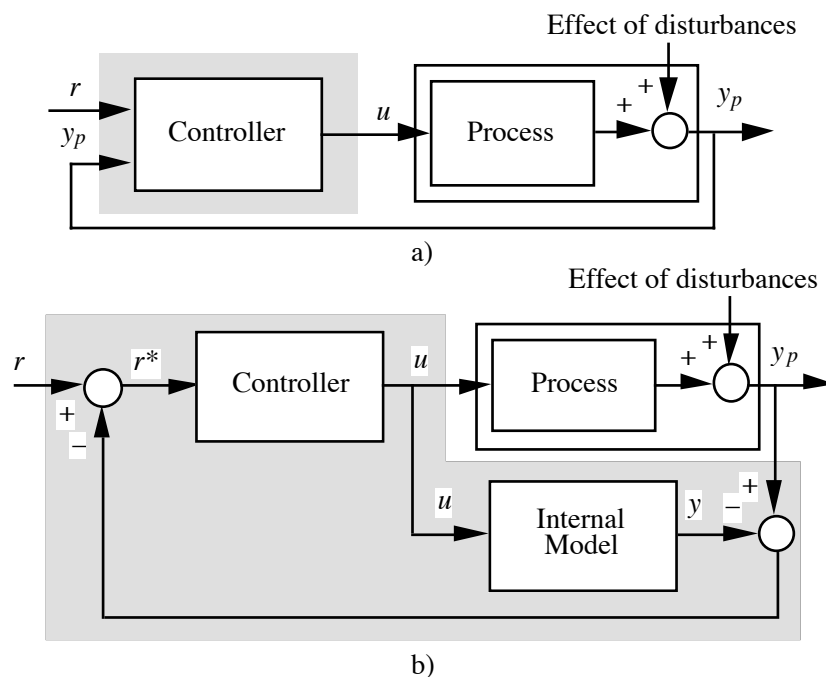

Fig. 1. Two types of control systems: a) simple feedback control system; b) internal model control system.

Internal model control devices have been extensively studied in the case of a linear modeling of the process, and have been shown to have good robustness properties against disturbances and a model mismatch [8]. Developments of internal model control in the case of nonlinear models have been proposed, mainly for continuous-time models [9] [10], but also for particular classes of

discrete-time models [11]; neural discrete-time internal model control systems are discussed in [12] [13] [14] [15]. Discrete-time internal model control possesses the following properties:

(a) If the process and the controller are (input-output) stable, and if the internal model is perfect, then the control system is stable.

(b) If the process and the controller are stable, if the internal model is perfect, if the controller is the inverse[1] of the internal model, and if there is no disturbance, then perfect control is achieved.

(c) If the controller static gain is equal to the inverse of the internal model static gain, and if the control system is stable with this controller, then offset-free control is obtained for constant setpoints and output disturbances.

A frequent design strategy consists in cascading the inverse of the internal model with a unit gain low-pass filter; if the control system is stable, offset-free control is guaranteed for constant inputs (i.e. setpoint and disturbances), and robustness against a possible mismatch of the internal model at high frequencies is obtained.

We propose here a general design procedure of neural internal model control systems. We first show that the design of a model reference controller for internal model control necessitates only the training of the internal model and of the inverse of the model deprived from its delay, and that the presence of the delay does not increase the order of the inverse, a fact of first importance for the training. The controller is then obtained by cascading this inverse with a rallying model which imposes the regulation dynamic behavior and ensures the robustness of the stability of the control system. A change in the desired regulation dynamic behavior, or an improvement of the stability, can be obtained by simply tuning the rallying model, without training the whole model reference controller anew. The advantages of neural networks are exploited throughout the paper, in particular, the ability of neural networks for nonlinear black-box modeling and for inverse modeling. Nevertheless, we also focus on the specific constraints of using neural networks: the robustness properties of internal model control systems being obtained when the inverse is perfect, we detail the precautions which must be taken for the training of the inverse so that it is accurate in the whole space visited during operation with the process. In the same spirit, we make an emphasis

---

[1] This term is defined in section 3 for the class of discrete-time models we deal with.

on neural models affine in the control input, whose perfect inverse is derived without training. The paper is organized as follows:

In section 2, we present the class of models we consider, models that will be used both to train a neural controller and as internal models. In section 3, we define the model reference controller associated to a given model, and show that its design for internal model control necessitates only the training of the inverse of the model deprived from its delay. The training system is then described in section 4. In section 5, the properties of the neural internal model control system we propose are analyzed, and compared to other systems proposed in the past. The design procedure and the properties of the neural internal model control system for processes without and with delay are illustrated on simulated processes in section 6.

2 NONLINEAR MODELS FOR CONTROL SYSTEMS

We are interested in stable single input / single output processes. We consider the process to be described by the following discrete-time deterministic nonlinear input-output model ("true" model):

$$\begin{cases} x_p(k) = h\left(x_p\{_{k-n}^{k-1}, u\{_{k-d-m+1}^{k-d}\right) \\ y_p(k) = x_p(k) + p(k) \end{cases} \tag{2.1}$$

where $x_p\{_{k-n}^{k-1}$ denotes the state $\{x_p(k-1), \ldots, x_p(k-n)\}$ and similarly $u\{_{k-d-m+1}^{k-d}$ denotes the set of m past control inputs $\{u(k-d), \ldots, u(k-d-m+1)\}$; $d \geq 1$ is the delay; h is an unknown nonlinear function; y_p is the output, the effect of disturbances being modeled by p.

A neural *black-box* (BB) predictive model associated to the true model (2.1) is of the form:

$$y(k) = \phi\left(y\{_{k-n}^{k-1}, u\{_{k-d-m+1}^{k-d}; \theta\right) \tag{2.2}$$

where ϕ is the nonlinear function implemented by a feedforward network with weights θ. It was proved that any continuous mapping over a compact set can be approximated as accurately as desired by a feedforward neural network with one hidden layer [16] [17]. Thus, provided that m, n and d are truly determined and that $p=0$, if the number of neurons of the feedforward part of the model is sufficient, if the size and distribution of the training sequences are appropriately chosen, and if the network is trained with an efficient algorithm, then the function ϕ implemented by the neural network will be arbitrarily close to the function h in the domain delimited by the training sequences. The modeling procedure has been extensively discussed in [18] [19], and will not be dealt with here. Efficient training algorithms for recurrent neural networks are described in [20] [21] [22] for example. Unfortunately, the fact that ideally ($p=0$) the function ϕ can approach the

function *h* defining the stable process with an arbitrary accuracy does not imply that the neural model is also stable; nevertheless, sufficient conditions for the stability of particular classes of neural networks are given in [23] [24]. We term *perfect* the model such that $\phi = h$.

After its training, it is another form of the model (2.2) (whose parameters θ are then fixed) that will be needed for internal model control, i.e. to train a neural controller and to implement the internal model. Let us split model (2.2) into the following model, based on the same function ϕ as model (2.2):

$$z(k) = \phi\left(z\{_{k-n}^{k-1}, \, u\{_{k-m}^{k-1}\right) \tag{2.3}$$

and a delay *d*-1: $y(k) = z(k-d+1)$. We name model (2.3) the *delay-deprived model* associated to the model (2.2); it is a model with unit delay.

Remark Concerning Affine Black-Box Models

Less general models than (2.2), such as black-box neural models affine in the control input (ABB), are of a form lending itself to the design method used in this paper. The advantage of using these models is that their perfect inverse can be derived, without training, and thus allow a direct design of a model reference controller (as shown in section 4). An ABB model associated to the true model (2.1) is:

$$y(k) = \xi\left(y\{_{k-n}^{k-1}, \, u\{_{k-m-d+1}^{k-d-1}\right) + \psi\left(y\{_{k-n}^{k-1}, \, u\{_{k-m-d+1}^{k-d-1}\right) u(k\text{-}d) \tag{2.4}$$

where ξ and ψ are nonlinear functions implemented by two feedforward networks obtained after a preliminary modeling procedure. Though not as general as BB models, ABB models are good representations of many physical systems (robot manipulators, mobile robots…), and are often considered in the neural network literature [1] [4] [5], especially in direct adaptive approaches.

3 MODEL REFERENCE CONTROL

We consider the problem of *tracking* a setpoint sequence $\{r(k)\}$, possibly in the presence of deterministic disturbances which might occur randomly in time, and whose effect must be cancelled (*regulation task*). The desired dynamic behavior of the control system is chosen to be given by a stable reference model. In this paper, the reference model is chosen linear, but according to known characteristics of the process and possibly to the saturations of the controller, a nonlinear reference

model can be chosen as well (see for example the training of a minimum-time heading controller

for a 4WD vehicle in [7]). A suitable linear reference model for model (2.2) is given by:

$$E(q^{-1}) y_r(k) = q^{-d} H(q^{-1}) r(k) \tag{3.1}$$

where q^{-1} is the backward shift operator, r denotes the setpoint and y_r the output of the reference

model, with:

$$E(q^{-1}) = 1 + e_1 q^{-1} + \ldots + e_{n_E} q^{-n_E}$$

$$H(q^{-1}) = h_0 + h_1 q^{-1} + \ldots + h_{n_H} q^{-n_H} \qquad h_0 \neq 0 \tag{3.2}$$

$$1 + \sum_{i=1}^{n_E} e_i = \sum_{i=0}^{n_H} h_i$$

(the last relation meaning that the reference model has unit static gain). In order to achieve a

"smooth" control , one should choose $n_E \geq n$; for simplicity, we take $n_E = n_H = n$. The final

objective is thus to impose that the measured response $\{y_p(k)\}$ to the setpoint sequence $\{r(k)\}$ be

given by:

$$E(q^{-1}) y_p(k) = q^{-d} H(q^{-1}) r(k) \tag{3.3}$$

This can be achieved through the use of a so called model reference controller, whose definition and

existence conditions follow, as well as that of the strongly related inverse model.

3.1 The Model Reference Controller

Since we deal with non adaptive *indirect* control, the model reference controller and the inverse

model will be defined with respect to the *model* (2.2). We will distinguish between, on the one

hand, the *theoretical* model reference controller and the *theoretical* inverse model, and, on the other

hand, their neural implementations which will be termed *empirical*.

The theoretical model reference controller associated to a given model (2.2) with delay d is

defined, if it exists, as the state-feedback controller which, when cascaded with the model in a

simple feedback system, imposes the reference dynamic behavior (3.1) to this system[2]:

---

[2] In the scheme proposed here, the model reference controller imposes the same dynamic behavior for tracking and for regulation. For $r(k) = r = cte$, if a disturbance brings the output away from the setpoint r, the dynamic behavior of the control error after the disturbance has disappeared is given by the polynomial $E(q^{-1})$:

$$E(q^{-1}) e(k) = E(q^{-1}) \left(y_r(k) - y_p(k) \right) = 0$$

$$E(q^{-1})\, y(k) = q^{-d} H(q^{-1})\, r(k) \tag{3.1.1}$$

More precisely, whatever the state $y(k)$, …, $y(k-n+1)$ of the model at time k, the theoretical model reference control input $u(k)$ is such that the behavior of the simple feedback system {controller+model} is identical to the behavior of the reference model (3.1) at all time $k' \geq k+d$. We consider the class of models whose associated theoretical model reference controller exists and leads to bounded controls in the state domain of interest, i.e. models which are feedback linearizable and whose zero dynamics are stable. Conditions for the existence of such a model reference control law can be found in [3] [6] [25] [26]. The model reference controller is also well known in the linear literature [27]: the only condition for its existence is that the model be minimum phase (the minimum phase requirement for linear systems is equivalent to the requirement that the control law should result in a bounded control). Whereas it is easy to check the minimum phase condition for a linear system, it is impossible at present to verify analytically that the conditions required for the feedback linearizability and zero dynamics stability of a BB neural model are satisfied: one has to assume that they are satisfied and to apply the training procedure described in section 4.

The model reference controller corresponding to $E(q^{-1}) = H(q^{-1}) = 1$ (i.e. the reference model is the delay operator q^{-d}) is nothing but the inverse model: *the theoretical inverse model* associated to a given model (2.2) with delay d is defined, if it exists, as the controller which, when cascaded with the model in a simple feedback system, imposes the behavior of a pure delay d to this system:

$$y(k+d) = r(k) = y_r(k+d) \tag{3.1.2}$$

It is known in the linear literature as the " one-step-ahead controller " [27].

3.2 The Model Reference Controller Equivalence

By definition, the model reference controller imposes the desired dynamic behavior (3.1.1), which can be rewritten as:

Thus, if a different dynamic behavior is desired for tracking (i.e. an additional degree of freedom), the setpoint sequence should be appropriately filtered. Let $E'(q^{-1})$ and $H'(q^{-1})$ denote the polynomials associated to the desired tracking behavior; the setpoint sequence $r(k)$ should be replaced by the filtered sequence $r'(k)$:

$$r'(k) = \frac{H'(q^{-1})}{E'(q^{-1})} \frac{E(q^{-1})}{H(q^{-1})} r(k)$$

For simplicity, we consider the input of the system to be the sequence $r(k)$.

$$y(k+d) = H(q^{-1})\, r(k) + \left(1 - E(q^{-1})\right) y(k+d) \tag{3.2.1}$$

Using the model expression (2.2), it follows that:

$$\phi\left(y\{^{k+d-1}_{k-n+d},\, u\{^{k}_{k-m+1}\right) = H(q^{-1})\, r(k) + \left(1 - E(q^{-1})\right) y(k+d) \tag{3.2.2}$$

If the feedback linearizability and bounded control assumptions hold, there exists a stable control law $u(k)$ which can be expressed as the function g of $H(q^{-1})\, r(k) + \left(1 - E(q^{-1})\right) y(k+d)$, of the model outputs $y\{^{k+d-1}_{k-n+d}$, and of the inputs $u\{^{k-1}_{k-m+1}$ such that (3.2.2) is verified in the domain of interest. The expression of the theoretical model reference controller is thus:

$$u(k) = g\left(H(q^{-1})\, r(k) + \left(1 - E(q^{-1})\right)\ y(k+d),\, y\{^{k+d-1}_{k+d-n},\, u\{^{k-1}_{k-m+1}\right) \tag{3.2.3}$$

The model reference controller is recursive, of order m-1.

Using the same derivation with relation (3.1.2), the theoretical inverse model is given by:

$$u(k) = g\left(y_r(k+d),\, y\{^{k+d-1}_{k+d-n},\, u\{^{k-1}_{k-m+1}\right) \tag{3.2.4}$$

This shows that the theoretical model reference controller is obtained from the inverse model by replacing $y_r(k+d)$ in (3.2.4) by $H(q^{-1})\, r(k) + \left(1 - E(q^{-1})\right) y(k+d)$. This value can be considered as the output of a transversal filter that we call the *rallying model*[3] associated to the reference model (3.1) for the model (2.2):

$$y_{ral}(k+d) = H(q^{-1})\, r(k) + \left(1 - E(q^{-1})\right) y(k+d) \tag{3.2.5}$$

(as a matter of fact, the rallying model output sequence $\{y_{ral}(k)\}$ "rallies" the model output sequence $\{y(k)\}$ to the setpoint sequence $\{r(k)\}$).

Since, as mentioned in section 2, the model can be split into the delay-deprived model with output z and a delay d-1, the inverse model (3.2.4) is also the inverse of the delay-deprived model:

$$u(k) = g\left(z_{ral}(k+1),\, z\{^{k}_{k-n+1},\, u\{^{k-1}_{k-m+1}\right) \tag{3.2.6}$$

driven by the output of a rallying model associated to the reference model, for the delay-deprived model (2.3):

$$z_{ral}(k+1) = H(q^{-1})\, r(k) + \left(1 - E(q^{-1})\right) z(k+1) \tag{3.2.7}$$

This equivalence is illustrated in Fig. 2.

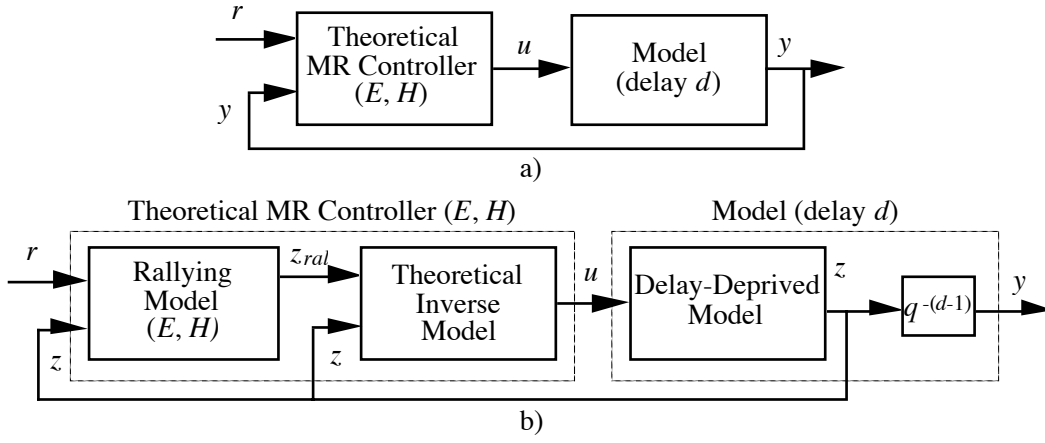[3] The name *rallying model* (in french "modèle de ralliement") is borrowed from [28].

Fig. 2. The model reference controller equivalence.

The inverse model (3.2.4) or (3.2.6) has been called "forward time-shift right inverse" in [29], since if $d>1$ $y\{_{k+d-n}^{k+d-1}$ and $\left(1 - E(q^{-1})\right) y(k+d)$, or equivalently $z\{_{k-n+1}^{k}$ and $\left(1 - E(q^{-1})\right) z(k+1)$, contain not only outputs up to time k, but also the outputs $y(k+1)$, …, $y(k+d-1)$. Thus, if $d>1$, the model reference controller expressed as in (3.2.4) and (3.2.5), or equivalently (3.2.6) and (3.2.7), cannot be used in a *simple feedback control system* where the model reference controller is cascaded with the process (i.e. the output of the model y in (3.2.4) and (3.2.5) is replaced by the measured output of the process y_p), since the outputs $y_p(k+1)$, …, $y_p(k+d-1)$ are not available at time k. It can only be used in a control system containing the model of the process, as in an *internal model control system*. In this case, whatever the chosen reference model, only the function g is to be estimated using a training procedure which is presented in the next section.

Remark on Simple Feedback Control Systems.

If $d > 1$ and if the model reference controller were to be used in a simple feedback control system, the predictions of the future process outputs being unavailable, the theoretical model reference controller (3.2.3) should be reformulated as a function of the past model outputs $y(k)$, …, $y(k-n+1)$ and past inputs $u(k)$, …, $u(k-m+1)$ only. It is possible since the future outputs $y(k+1)$, …, $y(k+d-1)$ are functions of the past inputs and outputs, i.e. there exists a function $\widetilde{\widetilde{g}}_E$ of these arguments such that:

$$\widetilde{\widetilde{g}}_E\left(H(q^{-1}) r(k), y\{_{k-n+1}^{k}, u\{_{k-m-d+1}^{k-1}\right) = g\left(H(q^{-1}) r(k) + \left(1 - E(q^{-1})\right) y(k+d), y\{_{k-n+d}^{k+d-1}, u\{_{k-m+1}^{k-1}\right)\tag{3.2.8}$$

The expression of the model reference control law as a function of past outputs only is thus:

$$u(k) = \widetilde{\widetilde{g}}_E\left(H(q^{-1}) r(k), y\{_{k-n+1}^{k}, u\{_{k-m-d+1}^{k-1}\right)\tag{3.2.9}$$

which is given in [30] for example; the order of this controller is m-1+d, i.e. it is greater than the order of the controller expressed as in (3.2.3). Moreover, $\widetilde{g_E}$ depends on the polynomial $E(q^{-1})$, and the controller must be retrained if the reference model is modified.

4 DESIGN OF A NEURAL MODEL REFERENCE CONTROLLER

For internal model control (or simple feedback control when d=1), whatever the chosen reference model, and once the model (2.2) has been trained, the only problem that remains to be solved is the estimation of the inverse model function g. The *training system* of the inverse of a BB neural model, shown on Fig. 3, is a simple feedback system consisting of:

– the neural inverse model, to be trained:

$$u(k) = \gamma \left(z_r(k+1),\ z\{^{k}_{k-n+1},\ u\{^{k-1}_{k-m+1};\ \theta \right) \tag{4.1}$$

where γ is the nonlinear function implemented by a feedforward network with weights θ, and $\{z_r(k)\}$ is an appropriately chosen reference sequence;

– the delay-deprived model (2.3), which is fixed:

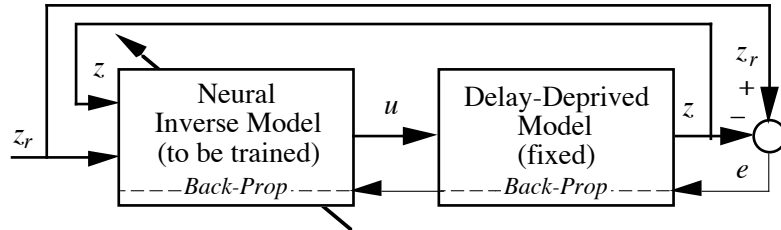$$z(k) = \phi \left(z\{^{k-1}_{k-n},\ u\{^{k-1}_{k-m} \right)$$

Fig. 3. The training system of the inverse model.

The cost function to be minimized is:

$$J(\theta) = \frac{1}{2} \sum_{k=1}^{N} \left(z_r(k) - z(k) \right)^2 \tag{4.2}$$

The size N of the training sequences and their distribution are chosen according to the future use of the controller. If the training algorithm is efficient (the reader is referred anew to the references given in section 2 for the modeling procedure) and if the network architecture is flexible enough, then the function γ implemented by the neural network will be arbitrarily close to the function g in the domain delimited by the training sequences. Again, criteria for the stability of the neural inverse obtained can be found in [23] [24]. We term *perfect* the inverse such that $\gamma = g$: this device is the

perfect realization of the theoretical inverse. The model reference controller is then obtained by cascading the inverse model with the rallying model associated to the reference model.

Remark Concerning Affine Black-Box Models

The expression of the inverse of the ABB model deprived from its delay is directly derived (without training):

$$u(k) = \frac{z_r(k+1) - \xi\left(z\{_{k-n+1}^k, u\{_{k-m+1}^{k-1}\right)}{\psi\left(z\{_{k-n+1}^k, u\{_{k-m+1}^{k-1}\right)} \tag{4.3}$$

The advantage of using an ABB model is naturally that the inverse (4.3) is perfect.

To summarize:

Under the assumptions that the neural model (2.2) of the process is both stable and inverse stable, we have shown that the design of a model reference controller for an internal model control system involves a training procedure which is independent of the chosen reference dynamics and of the delay of the process, since only the inverse of the delay-deprived model needs to be trained. We introduced the rallying model with which the inverse is then to be cascaded with, in order to impose the desired dynamic behavior.

5 THE INTERNAL MODEL CONTROL SYSTEM

Having trained or directly derived a model reference controller, the design of a simple feedback control system is straightforward: it consists of the model reference controller in a loop with the process only. It is shown in Fig. 4 in the case *d*=1.
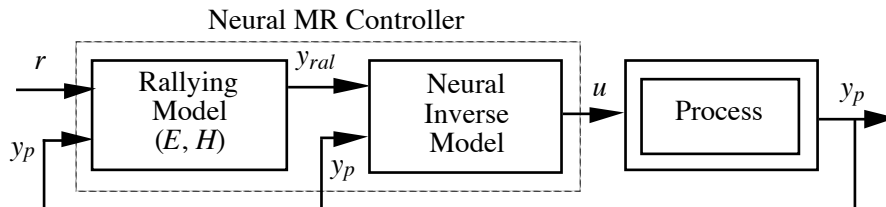

Fig. 4. The simple feedback control system of a process with unit delay.

Since the performance of such a control system is highly dependent on the accuracy of the neural model used for the design of the model reference controller, it might be unsatisfactory

(inappropriate tracking and regulation dynamics, too large a steady-state error…) if the model mismatch is too important, and not cancel the effect of disturbances. Then, if the process is stable, one can advantageously resort to an internal model control system.

5.1 The Internal Model Control System

The control device consists here of the model reference controller and of the internal model, and is shown in Fig. 5. As already mentioned in section 2, the internal model (2.2) is split into:

– the delay-deprived model (2.3) associated to (2.2):

$$z(k) = \phi\left(z\{_{k-n}^{k-1}, u\{_{k-m}^{k-1}\right)$$

– a delay $(d–1)$: $y(k) = z(k-d+1)$.

The model reference controller is built of:

– the neural inverse model;

– the rallying model for the delay-deprived model. Its inputs are the internal model corrected setpoint $r*$ and the past outputs of the delay-deprived model.
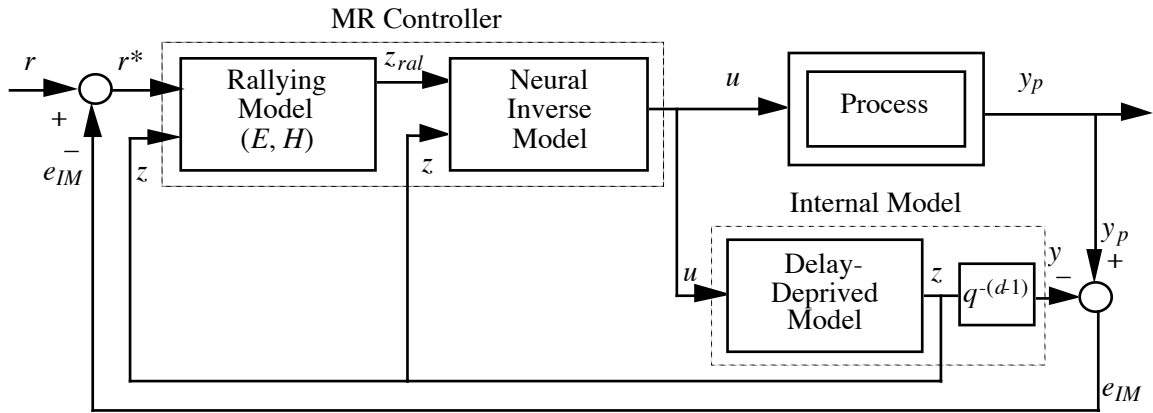

Fig. 5. The proposed internal model control system.

Let us now consider the computations performed by the control device at time k, which is defined as the instant when the measured output $y_p(k)$ becomes available:

a) the difference between the outputs of the process and of the internal model:

$$e_{IM}(k) = y_p(k) - y(k) = y_p(k) - z(k-d+1)$$

b) the corrected setpoint for the internal model: $r*(k) = r(k) - e_{IM}(k)$

c) the output of the rallying model:

$$z_{ral}(k+1) = H(q^{-1})\, r*(k) + \left(1 - E(q^{-1})\right) z(k+1) \tag{5.1.1}$$

d) the control value, output of the inverse model, which is applied to the process:

$$u(k) = \gamma \left(z_{ral}(k+1),\ z\{^{k}_{k-n+1},\ u\{^{k-1}_{k-m+1} \right) \tag{5.1.2}$$

e) the output of the delay-deprived model $z(k+1) = y(k+d)$, which will be needed at the next time

step:

$$z(k+1) = \phi \left(z\{^{k}_{k-n+1},\ u\{^{k}_{k-m+1} \right) \tag{5.1.3}$$

In section 1, we have briefly recalled the well established properties of internal model control systems. Our aim is now to justify the proposed control system of Fig. 5, and to specify the design precautions that must be taken in order to enhance its robustness properties.

5.2 Properties of the Proposed Internal Model Control System

The process, the internal model and the inverse model are assumed to be stable.

a) We first assume that the inverse is perfect (i.e. $\gamma = g$, assumption which is true in the case of an ABB model): the output of the delay-deprived model is then equal to the output of the rallying model: $z(k+1) = z_{ral}(k+1)$. The control system is then equivalent to that shown in Fig. 6.
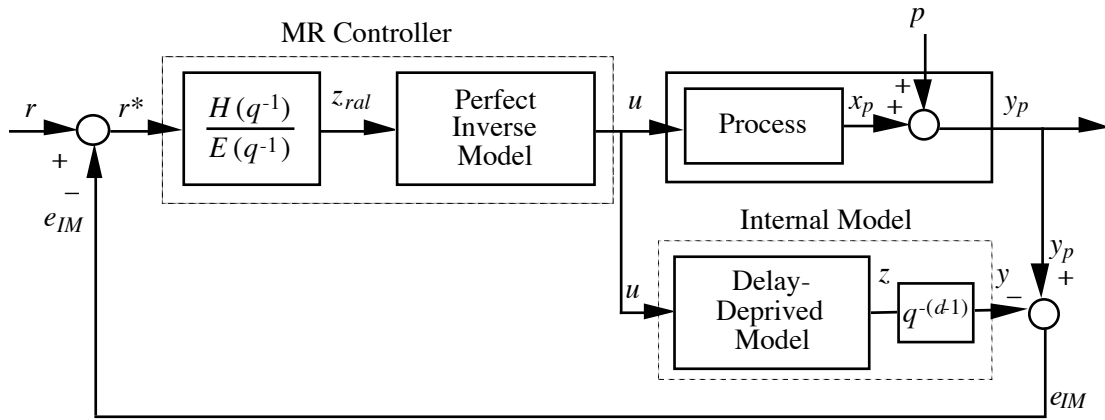

Fig. 6. The internal model control system equivalent to the control system of Fig. 5 if the inverse model is perfect; note that the feedback sequence is $\{e_{IM}(k)\}$.

– If the internal model is perfect ($\phi = h$), the control system is equivalent to the transfer function shown in Fig. 7. It is stable, and if $p = 0$, perfect control ($E(q^{-1})\ y_p(k) = q^{-d}\ H(q^{-1})\ r(k)$) is achieved; the effect of constant disturbances is cancelled in the steady-state.
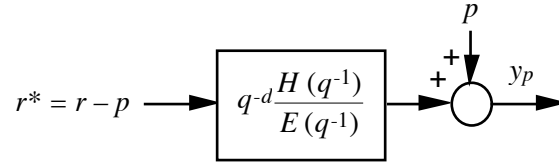
Fig. 7. The transfer function equivalent to the control system of Fig. 5 if the inverse model and the internal model are perfect.

– If the internal model is not perfect, let us study the robustness of the stability and of the performances, as compared to the ideal situation above.

* Robustness of the stability: Let us denote by P the nonlinear operator describing the process, by M the nonlinear operator describing the internal model, by I the nonlinear operator describing the inverse model, by F the linear operator such that $F(q^{-1}) = \dfrac{H(q^{-1})}{E(q^{-1})}$, and by Id the identity operator; in operator representation, the output of the process is given by:

$$y_p = P \circ [Id + I \circ F \circ (P - M)]^{-1} I \circ F (r - p) + p \qquad (5.2.1)$$

Since F is a low-pass filter, the effect on the stability of unmodeled dynamics at high frequency contained in the term P–M can be thus attenuated by diminishing the bandwidth of the filter. Thus, the rallying model plays the role of the classic internal model control filter [9] which ensures the robustness of the stability with respect to a model mismatch. Thanks to the decomposition of the model reference controller into inverse and rallying model we have made, the rallying model can be tuned for stability purposes without having to modify the rest of the control device.

* Robustness of the performance: since $y(k) = z(k\text{-}d\text{+}1) = z_{ral}(k\text{-}d\text{+}1)$, the internal model control system of Fig. 6 is equivalent to the simple feedback control system shown in Fig. 8.
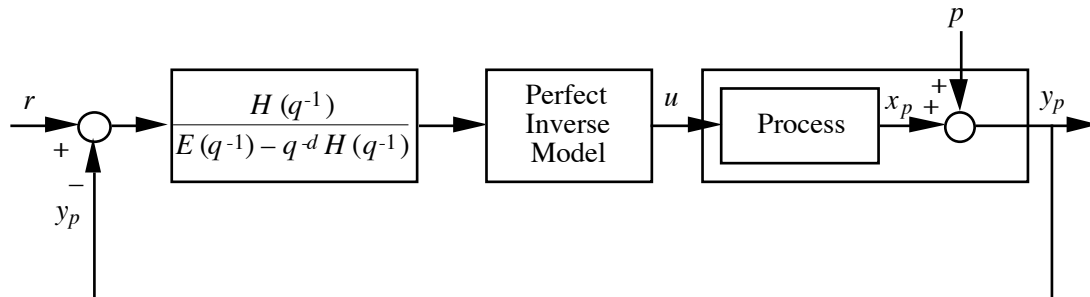

Fig. 8. The simple feedback control system equivalent to the control system of Fig. 5 if the inverse model is perfect; note that the feedback sequence is $\{y_p(k)\}$.

Since we imposed that $\dfrac{H(q^{-1})}{E(q^{-1})}$ has unit gain, $\dfrac{H(q^{-1})}{E(q^{-1}) - q^{-d} H(q^{-1})}$ contains an integrator: thus, the effect of constant disturbances and of a model mismatch are cancelled in the steady-state.

b) If the inverse is not perfect (controller/model mismatch), the general case when the model is not an ABB model, much less can be stated about the robustness of the system. We can only try to demonstrate that the proposed neural internal model control system is the best possible. As a matter of fact, neural internal model control system are often simplified in two ways that are not justified anymore if the inverse is not perfect:

– Instead of the rallying model, which is a transversal filter, a recursive filter with input $r*$ and output $z*$ is used to drive the inverse model, as in [13] [14] for example. In the linear case, we showed in [31] that the stability of an internal model control system with a rallying model is more robust towards a controller/model mismatch than with a recursive filter.

– The internal model is removed and the rallying model is replaced by a recursive filter, as in [12] for example, see Fig. 9.
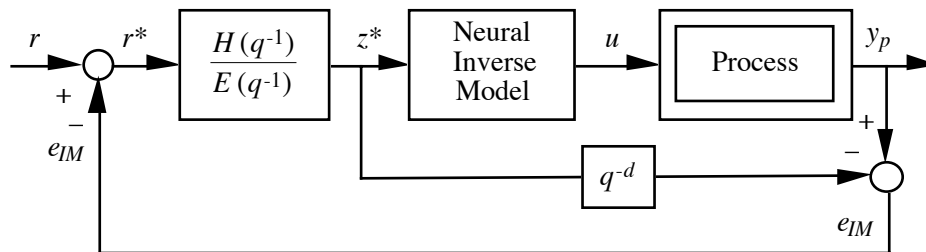

Fig. 9. An "internal model" control system to avoid.

Since the process is usually equipped with an actuator whose output is limited in amplitude, the existence of an inverse is then restricted to these limits, and so is the legitimacy of the internal model removal. As a consequence, the control system of Fig. 9 is not immune to wind-up, because the loop with the delay d creates an integrator that is never switched off, as opposed to the control system that we propose.

5.3 Design Precautions for Internal Model Control Systems Based on Black-Box Models

Let us now specify the design precautions that must be taken in the case of a BB neural network based internal model control system, in order to maximize its robustness properties. One should keep in mind that the validity domain of the neural inverse model is restricted to the domain of the state space it was trained in, i.e. the domain explored when driven by the training reference sequence $\{z_r(k)\}$. During the operating phase, the neural inverse model controls the internal model, i.e. its inputs are the outputs $\{z(k)\}$ of the internal model and the output $\{z_{ral}(k)\}$ of the rallying

model, which might evolve in a larger region of the state space than the process, due to a model mismatch and to disturbances. Thus, the training reference sequence $\{z_r(k)\}$ must be representative of $\{z_{ral}(k)\}$. Practically, it must be computed from a setpoint sequence exploring a larger domain (depending on the model mismatch and the expected disturbance) than that wished for the outputs $\{y_p(k)\}$ of the process, with a faster model than the reference model (see [31] for an illustrative example: the piloting of a 4WD vehicle). This constraint on BB models, added to the difficulty to assess the robustness of the system, makes ABB models again really advantageous for internal model control.

6 ILLUSTRATIVE EXAMPLES

This section presents the modeling and control of simulated processes. The internal model control system presented in the previous section, as well as a simple feedback control system, are designed using successively a BB and an ABB model of the process, and their performances are compared on representative setpoint sequences, in the presence of disturbances. A process with delay is then considered, and the use of a delay-deprived model for IMC illustrates the simplicity of our design procedure for processes with delay (see also [32] for such an illustration).

6.1 The Process to Be Controlled

The process to be controlled is simulated by the following discrete-time equations:

$$\begin{cases} x_p(k) = h\left(\ x_p(k\text{-}1),\ x_p(k\text{-}2),\ u(k\text{-}1)\right) \\ y_p(k) = x_p(k) + p(k) \end{cases} \tag{6.1.1}$$

where the function h is given by:

$$h\left(\ z_1, z_2, z_3\right) = \frac{24 + z_1}{30}\ z_1 - 0.8\ \frac{z_3^2}{1 + z_3^2}\ z_2 + 0.5\ z_3 \tag{6.1.2}$$

For large input amplitudes, the process is a stable nonlinear oscillatory low-pass filter (see Fig. 10). More specifically, around operating points $\{u_0, y_0\}$, its behavior is the following: (i) around $\{0, 0\}$, it behaves like a damped first-order filter with time constant $3.6\ T$ (T is the sampling period); (ii) around operating points corresponding to $0.1 < |u_0| < 0.5$, it behaves like an oscillatory second-order filter with pseudo-period $5\ T$; (iii) for operating points corresponding to larger u_0, the behavior becomes more complex. The amplitude of the control u is limited to [-5; 5] by a saturation of the

actuator output; for a constant amplitude 5 of the control input, the steady-state output is equal to 2.7, and for an amplitude of -5, to -2.3.

The reference model is obtained by a discretization of the continuous-time damped linear second-order model with time constants T and 1.2 T, and with unit static gain:

$$y_r(k+1) + e_1 \, y_r(k) + e_2 \, y_r(k-1) = h_0 \, r(k) + h_1 \, r(k-1) \tag{6.1.3}$$

where: $e_1 = -0.803$; $e_2 = 0.160$; $h_0 = 0.232$; $h_1 = 0.126$.

6.2 Modeling the Process for Control

The control input sequence for the training of the internal model consists of pulses of random amplitude in the range [-5; 5] with a duration of 10 sampling periods; the total length of the sequence is 1000. A control input sequence used for the performance estimation is generated in the same manner; it is shown in Fig. 10. The process is not disturbed ($p=0$) during the recording of the training and performance estimation sequences.
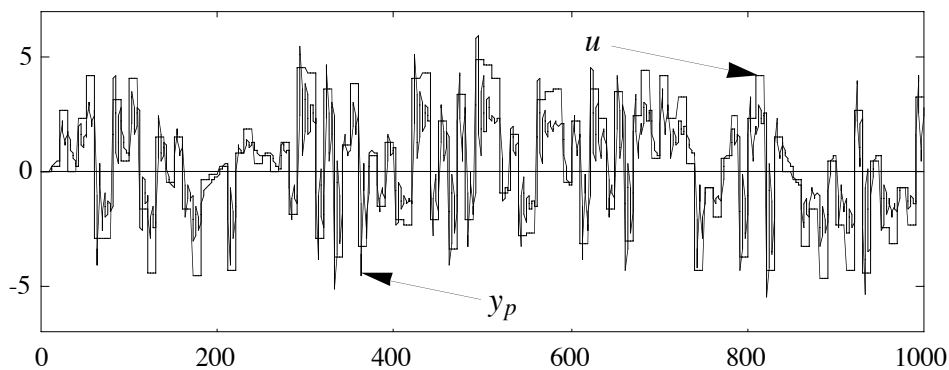

Fig. 10. The performance estimation sequences for the modeling of the simulated process.

In order to design control systems based on the two types of models, ABB and BB candidate models were trained. We retained the two following candidates:

– a BB neural model ($n=2$, $m=1$, $d=1$):

$$y(k) = \phi \left(y(k-1), \, y(k-2), \, u(k-1) \right) \tag{6.2.1}$$

where ϕ is a function implemented by a fully connected network[4] with 5 sigmoidal hidden neurons and a linear output neuron. Using a quasi-Newton algorithm, we obtained a mean square

---

[4] A fully connected network with Ne inputs $\{x_1, \ldots, x_{Ne}\}$, and Nn ordered neurons, is such that the output x_i of neuron i is given by:

$$x_i = f_i \left(\sum_{j=1}^{i-1} \theta_{ij} \, x_j \right) \qquad i = Ne+1 \; to \; Ne+Nn$$

17

error on the training sequences (TMSE) of $1.2 \ 10^{-5}$ ($\sqrt{\text{TMSE}} = 3.5 \ 10^{-3}$), and a mean square error on the sequences for performance estimation (PMSE) of $1.4 \ 10^{-5}$ ($\sqrt{\text{PMSE}} = 3.7 \ 10^{-3}$).

– an ABB neural model ($n=2$, $m=2$, $d=1$):

$$y(k) = \xi\left(y(k\text{-}1), \ y(k\text{-}2), \ u(k\text{-}2)\right) + \psi\left(y(k\text{-}1), \ y(k\text{-}2), \ u(k\text{-}2)\right) u(k\text{-}1) \qquad (6.2.2)$$

where ξ and ϕ are implemented by two fully connected subnetworks with 5 sigmoidal hidden neurons and linear output neurons. Its TMSE is equal to $1.7 \ 10^{-2}$ ($\sqrt{\text{TMSE}} = 1.3 \ 10^{-1}$) and its PMSE to $1.9 \ 10^{-2}$ ($\sqrt{\text{TMSE}} = 1.4 \ 10^{-1}$).

As the simulated process itself is not an ABB system, it is not necessarily possible to get a very accurate ABB model; the above performance is the best we could obtain with an order of 2 and two delayed control inputs (i.e. one more than for the BB model). Nevertheless, we are going to show that control systems based on this ABB model have satisfactory performances, due to the possibility of deriving the perfect inverse of the ABB model.

6.3 Design of the Inverse Models

The process is to be controlled for setpoint pulses of amplitude in the range [-2; 1.75] (this range is comprised in the output range in the steady-state, as described in section 6.1).

6.3.1 *Training the Inverse of the Black-Box Neural Model*

The inverse model is trained using the training system described in section 4 and Fig. 3 for a model with $d=1$ (in that case $z(k) = y(k)$), which consists of:

– the neural inverse model, to be trained:

$$u(k) = \gamma\left(y_r(k\text{+}1), \ y(k), \ y(k\text{-}1); \ \theta\right) \qquad (6.3.1.1)$$

where γ is a function implemented by a fully connected network with sigmoidal hidden neurons, and where, according to the actuator limits, the activation function of the output neuron is the saturation function $sat(v, 5.0)$, i.e.:

$$sat\left(v, a\right) = v \ if \ |v| < a \ ; \ sat\left(v, a\right) = a \ sign(v) \ if \ |v| \geq a. \qquad (6.3.1.2)$$

– the BB neural model (6.2.1), whose weights are now fixed:

$$y(k) = \phi\left(y(k\text{-}1), \ y(k\text{-}2), \ u(k\text{-}1)\right)$$

---

where f_i is the activation function of neuron i. Neuron $Ne+Nn$ is the output neuron.

The training reference sequence $\{y_r(k)\}$ is computed as the output of the reference model for a setpoint sequence consisting of pulses of random amplitude in the range [-3; 3], and with a duration of 20 sampling periods; the total length of the sequence is $N=1000$. The maximum amplitudes are chosen so as to fully explore the desired output amplitude range and to reach the saturations. With 5 hidden neurons, we obtain a mean square error on the training sequences $\frac{1}{N} \sum_{k=1}^{N} (y_r(k) - y(k))^2 = 5.6 \cdot 10^{-4}$ (i.e. a root mean square error of $2.4 \cdot 10^{-2}$). Since most errors are due to the saturation, increasing the number of neurons does not diminish the mean square error significantly.

6.3.2 *Deriving the Inverse of the Affine Black-Box Neural Model*

The inverse of the ABB model (6.2.2) is directly derived:

$$u(k) = \frac{y_r(k+1) - \xi\,(y(k),\ y(k\text{-}1),\ u(k\text{-}1))}{\psi\,(y(k),\ y(k\text{-}1),\ u(k\text{-}1))} \qquad (6.3.2.1)$$

its weights being now fixed.

6.4 Implementation of the Control System

The simple feedback and internal model control systems based on the ABB and BB models are now tested with the simulated process. The setpoint sequence consists of pulses with amplitudes in the range [-2; 1.75]; the total length of the sequence is $N=330$. The process is affected by an output additive disturbance denoted by p, which consists of randomly occurring pulses of amplitude -1.

6.4.1 *Simple Feedback Control*

The behavior of the simple feedback control system based on the BB neural model is shown in Fig. 11a. The performance of the simple feedback control system based on the ABB model is similar, and thus not shown here: the mean square error $\frac{1}{N} \sum_{k=1}^{N} (y_r(k) - y_p(k))^2$ equals 0.81 with the BB and 0.82 with the ABB system (that is the root mean square errors equal roughly 0.9). A disadvantage of simple feedback control systems appears clearly: constant disturbances lead to important steady-state errors.
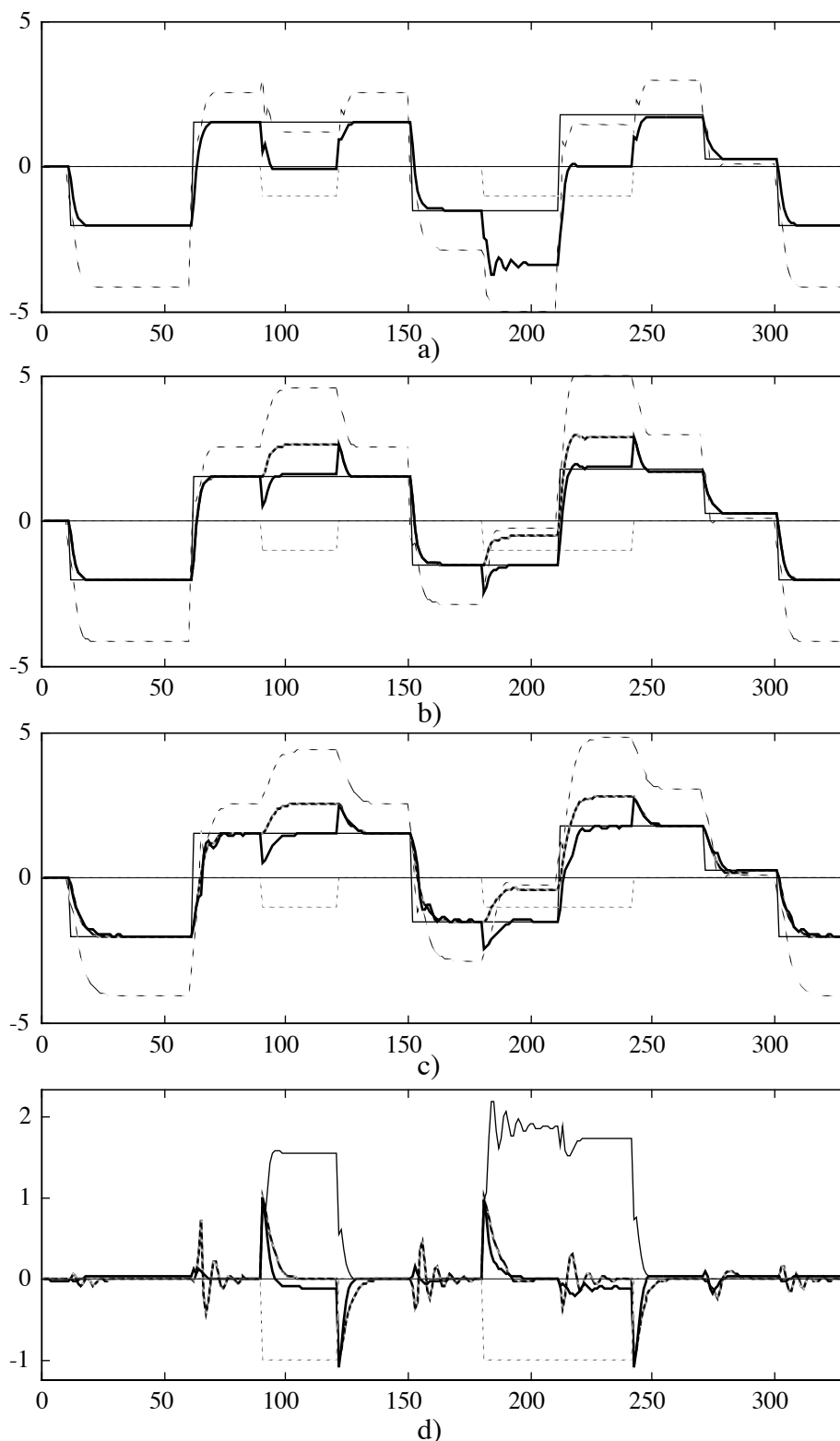
Fig. 11. The control systems for the simulated process (6.1.1) with unit delay: a) simple feedback control system based on the BB model (pulses in thin continuous line: r; negative pulses in thin dotted line: p; thin dashed line: u; thick continuous line: y_p); b) internal model control system based on the BB model (thin continuous line: r; thin dotted line: p; thin dashed line: u; thick continuous line: y_p; thick dashed line: y); c) internal model control system based on the ABB model (thin continuous line: r; thin dotted line: p; thin dashed line: u; thick continuous line: y_p; thick dashed line: y); d) control errors (thin continuous line: simple feedback control system based on the BB model; thick continuous line: internal model control system based on the BB model; thick dashed line: internal model control system based on the ABB model; thin dashed line: p).

20

Note that the poor performance of the SFC system could be improved by using a recursive reference model instead of the rallying model: the steady-state error would then be reduced. But on the other hand, this would lead to large variations at the control input when the disturbance occurs.

6.4.2 Internal Model Control

The performances of the two internal model control systems are quantitatively still very similar (the value of their mean square error is equal to 0.03, that is a root mean square error of $1.7 \ 10^{-1}$), but their behavior shows important qualitative differences. The behavior of the internal model control system based on the BB model is shown in Fig. 11b. y denotes the output of the internal model. Despite the internal model control structure, when the disturbance occurs, its effect is not fully cancelled (it appears clearly in the control error[5] shown in Fig. 11d). This is due to the fact that the model output value is close to or reaches 3, that is the limit of the domain the inverse model was trained in: the neural inverse model is not accurate in this state region. Retraining the controller in a larger domain leads to zero error, but our goal is to emphasize the importance of the design precautions sketched in section 5. The behavior of the internal model control system based on the ABB model is shown in Fig. 11c, and its control error in Fig. 11d. As opposed to the inverse of the BB model, the inverse (6.3.2.1) of the ABB model is perfect; as a consequence (property described in section 5.2.a), the effect of constant additive disturbances is asymptotically cancelled (see the zero steady-state error on Fig. 11d). Nevertheless, the mismatch between the process and the ABB model induces small oscillations after the transitions.

6.5. Application to a process with delay

In order to show that our procedure is not more complex in the case of a process with delay, a second process is chosen to be characterized by the same input-output relationship as process (6.1.1) but with a delay $d=10$. This process is thus simulated by the following discrete-time system:

---

[5] The control error is computed as the difference between the output of the reference model (not shown on the graphs for the sake of clarity), and the output of the process, that is $y_r - y_p$.

$$\begin{cases} x_p(k) = h\,(\ x_p(k\text{-}1),\ x_p(k\text{-}2),\ u(k\text{-}10)) \\ y_p(k) = x_p(k) + p(k) \end{cases} \tag{6.5.1}$$

The delay-deprived model of (6.5.1) is thus given by the same function ϕ of the BB neural model (6.2.1) (alternately, one could use the function of the ABB model):

$$z(k+1) = \phi\,(z(k),\ z(k\text{-}1),\ u(k)) \tag{6.5.2}$$

The neural inverse (6.3.1.1) can thus be used in an IMC system for process (6.5.1). Its behavior is shown in Fig. 12a.
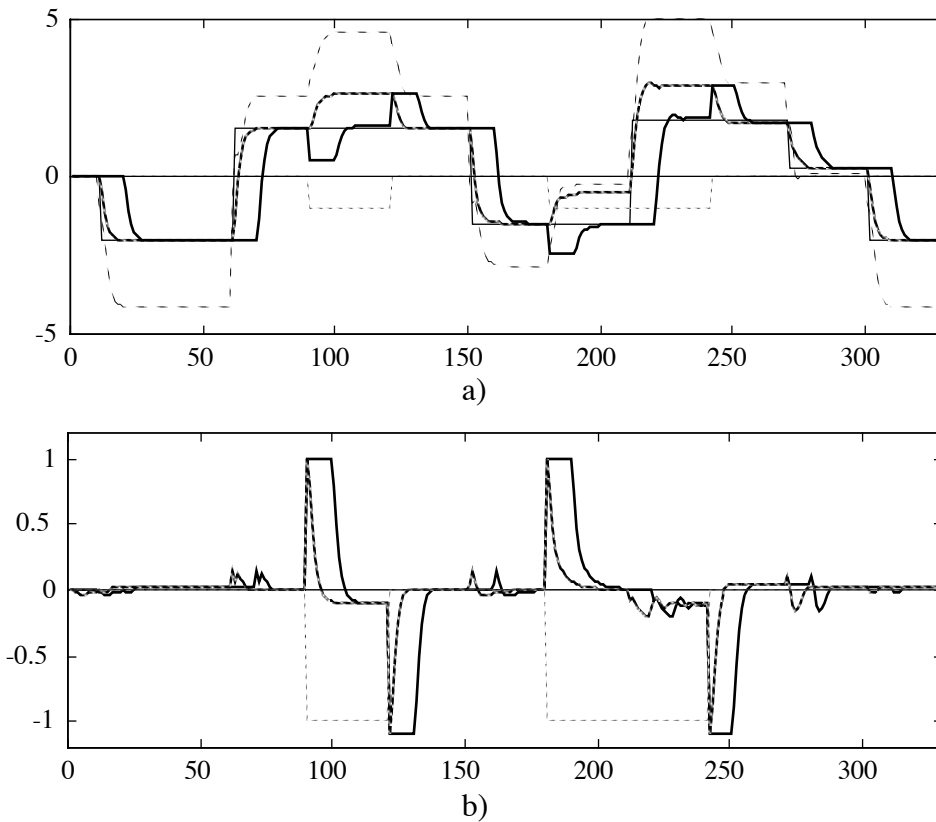

Fig. 12. The control system for the simulated process (6.5.1) with delay $d=10$: a) internal model control system based on the BB model (thin continuous line: r; thin dotted line: p; thin dashed line: u; thick continuous line: y_p; thick dashed line: z); b) control errors (thick continuous line: internal model control system of process (6.5.1) with delay $d=10$; thick dashed line: internal model control system of process (6.1.1) with delay $d=1$; thin dotted line: p).

The errors of the IMC control systems for process (6.1.1) and process (6.5.1) with delay $d = 10$ are shown in Fig. 12b. If we except the uncancellable errors due to the pure delay, the performance of the IMC system of process (6.5.1) is identical to that of process (6.1.1). The same conclusions can be drawn from the behavior of the IMC systems using the ABB model and its perfect inverse.

7 CONCLUSION

We have presented a design procedure of neural internal model control systems based on a model reference controller. We have shown that it can be built from the inverse of the delay-deprived model (which needs to be trained) and a rallying model (not trained) imposing the desired dynamic behavior; if necessary, the model reference controller can thus be modified by simply tuning the rallying model. The rallying model has then been shown to have the properties required to play the role of the classic internal model control filter. The advantage of internal model control systems is their robustness with respect to a model mismatch and to disturbances; these properties being obtained if the model reference controller is perfect, we have shown the advantage of using affine black-box neural models, which allow the derivation of their perfect inverse model. We have also stressed the design precautions that must be taken when the inverse needs to be trained, i.e. in the case of a fully black-box model. This design procedure of neural internal model control systems was illustrated here on simulated processes; it was also successfully applied to the longitudinal piloting of a 4WD Mercedes vehicle [31] [33].

REFERENCES

[1] K. S. Narendra, "Adaptive control of dynamical systems using neural networks", in *Handbook of intelligent control: neural, fuzzy and adaptive approaches*, D. A. White and D. A. Sofge eds, Van Nostrand Reinhold, New York, pp. 141-183, 1992.

[2] O. Nerrand, L. Personnaz and G. Dreyfus, "Nonlinear recursive identification and control by neural networks: a general framework", *Proceedings of the 2nd European Control Conference*, Groningen, Netherlands, pp. 93-98, 1993.

[3]   K. S. Narendra and S. Mukhopadhyay, "Adaptive control of nonlinear multivariable systems using neural networks", *Neural Networks* **Vol. 7 N°5**, pp. 737-752, 1994.

[4]   R. Sanner and J.-J. Slotine, "Stable adaptive control of robot manipulators using 'neural' networks", *Neural Computation* **Vol. 7 N°4**, pp. 753-790, 1995.

[5]   F.-C. Chen and H. K. Khalil, "Adaptive control of a class of nonlinear discrete-time systems using neural networks", *IEEE Trans. on Automatic Control* **Vol. 40 N°5**, pp. 791-801, 1995.

[6]   A. U. Levin and K. S. Narendra, "Control of nonlinear dynamical systems using neural networks: controllability and stabilization", *IEEE Trans. on Neural Networks* **Vol. 4 N°2**, pp. 192-206, 1993.

[7]   I. Rivals, L. Personnaz, G. Dreyfus and D. Canas, "Real-time control of an autonomous vehicle: a neural network approach to the path following problem", *5th International Conference on Neural Networks and their Applications*, Nîmes, pp. 219-229, 1993.

[8]   M. Morari and E. Zafiriou, *Robust process control*, Prentice-Hall International Editions, 1989.

[9]   G. C. Economou, M. Morari and B. O. Palsson, "Internal Model control. 5. Extension to nonlinear systems", *Ind. Eng. Chem. Process Des. Dev.* **Vol. 25 N°2**, pp. 403-411, 1986.

[10]   J. Calvet and Y. Arkun, "Feedforward and feedback linearization of nonlinear systems and its implementation using Internal Model control", *Ind. Eng. Chem. Res.* **Vol. 27 N°10**, pp. 1822-1831, 1988.

[11]   J. Alvarez, "An internal-model controller for nonlinear systems", *Proceedings of the 3rd European Control Conference*, Rome, pp. 301-306, 1995.

[12]   D. C. Psichogios and L. H. Ungar, "Direct and indirect model based control using artificial networks", *Ind. Eng. Chem. Res.* **Vol. 30 N°12**, pp. 2564-2573, 1991.

[13]   K. J. Hunt and D. Sbarbaro, "Neural networks for nonlinear Internal Model control", *IEE Proc. -D* **Vol. 138 N°5**, pp. 431-438, 1991.

[14]   K. J . Hunt and D. Sbarbaro, "Studies in neural network based control", in *Neural networks for control and systems,* K. Warwick, G. W. Irwin and K. J. Hunt eds, Peter Peregrinus Ltd., London, pp. 94-122, 1992.

[15]   J. C. Kalkkuhl and K. J. Hunt, "Discrete-time neural model structures for continuous nonlinear systems: fundamental properties and control aspects", in *Neural Adaptive Control Technology I*, R. Zbikowski and K. J. Hunt eds, World Scientific, pp. 3-40, 1996.

[16] G. Cybenko, "Approximation by superpositions of a sigmoidal function", *Mathematics of Control, Signal and Systems* **Vol. 2 N°4**, pp. 303-314, 1989.

[17] K. Hornik, M. Stinchcombe and H. White, "Multilayer feedforward networks are universal approximators", *Neural Networks* **Vol. 2**, pp. 359-366, 1989.

[18] O. Nerrand, P. Roussel-Ragot, D. Urbani, L. Personnaz and G. Dreyfus, "Training recurrent neural networks: why and how? An Illustration in Process Modeling", *IEEE Trans. on Neural Networks* **Vol. 5** , pp. 178-184, 1994.

[19] I. Rivals and L. Personnaz, "Black-box modeling with state-space neural networks", in *Neural Adaptive Control Technology I*, R. Zbikowski and K. J. Hunt eds, World Scientific, pp. 237-264, 1996.

[20] M. I. Jordan, *The learning of representations for sequential performance*, Doctoral Dissertation, University of California, San Diego, 1985.

[21] D. E. Rumelhart, G. E. Hinton and R. J. Williams, "Learning internal representations by error back-propagation", in *Parallel Distributed Processing: explorations in the microstructure of cognition. Vol.1 : Foundations*, D. E. Rumelhart, J. L. McLelland and the PDP Research Group eds, MIT Press, Cambridge MA, pp. 318-362, 1986.

[22] O. Nerrand, P. Roussel-Ragot, L. Personnaz and G. Dreyfus, "Neural networks and nonlinear adaptive filtering: unifying concepts and new algorithms", *Neural Computation* **Vol. 5 N°2**, pp. 165-199, 1993.

[23] L. Jin, P. N. Nikiforuk and M. M. Gupta, "Diagonal lyapunov functions for global stability of discrete-time neural networks", *Neural Network World* **1/95**, pp. 71-79, 1995.

[24] K. Tanaka, "An approach to stability criteria of neural-network control systems", *IEEE Trans. on Neural Networks* **Vol. 7 N°3**, pp. 629-642, 1996.

[25] S. Monaco and D. Normand-Cyrot, "Minimum-phase nonlinear discrete-time systems and feedback stabilisation", *Proceedings of the 26th Conference on Decision and Control*, Los Angeles, pp. 979-986, 1987.

[26] H. Nijmeijer and A. J. Van Der Schaft, *Nonlinear dynamical control systems*, Springer Verlag, Berlin, 1990.

[27] G. C. Goodwin and K. S. Sin, *Adaptive filtering prediction and control*, Prentice-Hall, New Jersey, 1984.

[28] S. Abu el Ata and A. Coic, "Commande prédictive par inversion: application aux systèmes non-linéaires", DRET contract n°87/1226, ADERSA, 1988.

[29] Ü. Kotta, *Inversion method in the discrete-time nonlinear control systems synthesis problems*, Springer Verlag, London, 1995.

[30] J.-M. Renders, M. Saerens and H. Bersini, "Adaptive neurocontrol of a certain class of MIMO discrete-time processes based on stability theory", in *Neural Network Engineering in Dynamic Control Systems*, K. J. Hunt, G. R. Irwin and K. Warwick eds, Springer, pp. 43-60, 1995.

[31] I. Rivals, *Modélisation et commande de processus par réseaux de neurones ; application au pilotage d'un véhicule autonome*, Thèse de Doctorat de l'Université Paris 6, 1995 (available at http://www.neurones.espci.fr/~rivals/).

[32] I. Rivals and L. Personnaz, "Internal Model control using neural networks", *Proceedings of the IEEE International Symposium on Industrial Electronics*, Warsaw, pp. 109-114, 1996.

[33] I. Rivals, D. Canas, L. Personnaz and G. Dreyfus, "Modeling and control of mobile robots and intelligent vehicles by neural networks", *IEEE Conference on Intelligent Vehicles*, Paris, pp. 137-142, 1994.