Pergamon PII: S0016-0032(96)00059-2

J. Franklin Inst. Vol. 334B, No. 1, pp. 135–153, 1997

Published by Elsevier Science Ltd
Printed in Great Britain
0016-0032/96 $17.00+0.00

Nonlinear Identification and Control of a Heat Exchanger: A Neural Network Approach

by S. BITTANTI *and* L. PIRODDI

Dipartimento di Elettronica e Informazione, Facoltà di Ingegneria, Politecnico di Milano, Piazza Leonardo da Vinci, 32-20133 Milano, Italy

(Received 11 March 1996; accepted 25 April 1996)

ABSTRACT: *In this paper the potentials of neural networks-based control techniques are explored by applying a nonlinear generalized minimum variance control methodology to a simulated application example. In particular, reference is made to the control problem of regulating the output temperature of a liquid-satured steam heat exchanger by acting on the liquid flow-rate. Due to the non-minimum phase characteristic of the dynamics of the process, a simple inverting minimum variance controller is unsuitable. On the other hand, an effective solution is provided by a detuned model reference approach, which introduces a penalization factor in the control variable. A steady-state off-set error problem, caused by the neural network approximations, is tackled by means of an hybrid control structure, which combines a nonlinear integral action block with a neural controller. A comparison analysis is made to show the effectiveness of the proposed neural control schemes with respect to classical linear controllers.* Copyright © 1996 Published by Elsevier Science Ltd

I. Introduction

With respect to nonlinear model-based control, it can be seen that design techniques based on mathematical models with fixed nonlinearities are mostly found in the literature (**1–7**). In addition, the structure of the model must be known in advance to the designer. In the last few years, however, advances in nonlinear identification (**8**) have offered new tools for the application of black box methods in the identification of nonlinear systems. A major role in this process has been played by the developments in the field of neural networks. Correspondingly, a variety of nonlinear control techniques based on neural models have been proposed in the literature (**9–11**). These techniques are usually based on a two-step procedure: a forward model of the process is usually identified and, subsequently, a suitable controller, mostly based on the process inverse dynamics, is obtained through an additional identification procedure or an optimization task.

More recently, minimum variance (*MV*) and generalized minimum variance (*GMV*) control techniques have been extended to the nonlinear case in a neural network framework (**12, 13**). The proposed methods result in the design of nonlinear controllers composed of linear and nonlinear blocks: the latter are realized through feedforward

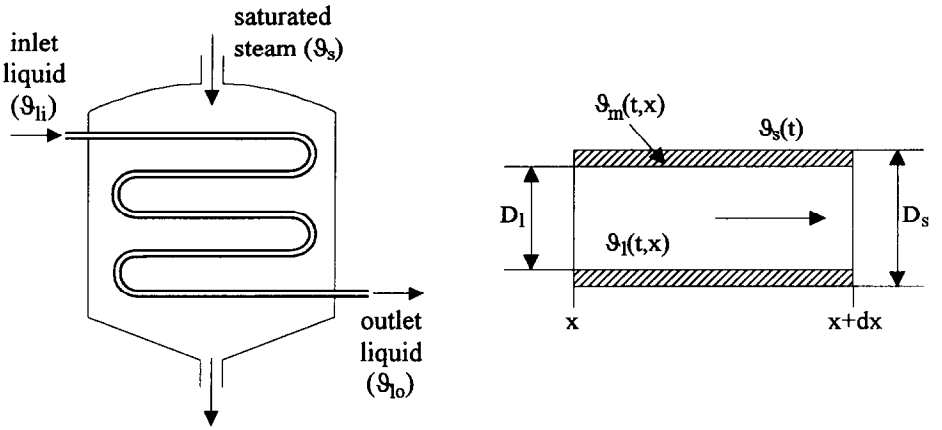

FIG. 1. The structure of the heat exchanger.

neural networks, trained so as to approximate suitable nonlinear recursive models. The design of the nonlinear parts of the controller is performed off-line, i.e. the weights of the neural networks are estimated on the basis of input–output data collected using suitable open loop experiments. On the other hand, the linear blocks can be easily turned on-line, to tailor the control system performance as needed.

The purpose of this paper is to test the applicability of these control methodologies, especially of *GMV* techniques, in a practical problem: that of regulating the output temperature of a liquid-satured steam heat exchanger by acting on the liquid flow-rate. The results in Refs (**14, 15**) provide full reference for the heat exchanger model. Moreover, a number of linear controllers are studied in Ref. (**16**). Herein, we discuss the design of neural network-based nonlinear predictive controllers of the *MV*/*GMV* type. The performances of these controllers are then compared with those of conventional control techniques.

The main motivation for the choice of the heat exchanger as a significant benckmark is that this plant is characterized by a non-minimum phase behaviour which makes the design of suitable controllers particulary challenging even in a linear design context. In the neural control framework this aspect is crucial, since inverting type controllers are usually employed and no assessed methodologies exist to deal with non-minimum phase plants. In what follows, it will be shown that the proposed nonlinear *GMV* methodology based on neural networks provides an effective tool to overcome this problem.

II. Process Description

Consider a heat exchanger of the type shown in Fig. 1, where water is heated by pressurized saturated steam through a copper tube. The controlled variable is the outlet liquid temperature, $\vartheta_{lo}(t)$. Among the input variables, the liquid flow rate, $q(t)$, is selected as the control variable, whereas the steam temperature, $\vartheta_s(t)$, and the inlet liquid temperature, $\vartheta_{li}(t)$ are disturbances.

The tube is described by a linear coordinate x, which measures the distance of a

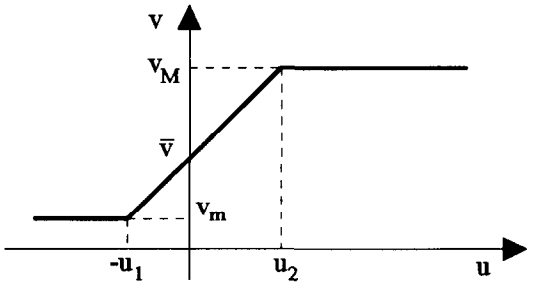

FIG. 2. The actuator characteristic.

generic section from the inlet. We will assume that the liquid and metal temperatures (ϑ_l and ϑ_m) are functions only of time and the axial coordinate x, whereas the saturated steam temperature $\vartheta_s(t)$ is uniform and independent of the shape of the tube. D_l and D_s are the internal and the external diameter of the tube, respectively. The liquid speed, $v(t) = q(t)/\mu_l$, where μ_l is the liquid density, is assumed to be uniform in the tube.

The process equations (**14**) are given by:

$$\begin{cases} \tau_l \dfrac{\partial \vartheta_l(t,x)}{\partial t} + \tau_l v(t) \dfrac{\partial \vartheta_l(t,x)}{\partial x} + \vartheta_l(t,x) - \vartheta_m(t,x) = 0 \\ T_m \dfrac{\partial \vartheta_m(t,x)}{\partial t} + \beta[\vartheta_m(t,x) - \vartheta_l(t,x)] + \vartheta_m(t,x) - \vartheta_v(t,x) = 0 \end{cases} \tag{1}$$

In the previous equations the coefficients τ_l, T_m and β are computed as follows:

$$\tau_l = \frac{\mu_l c_l}{\alpha_l \pi D_l}, \quad \tau_m = \frac{\mu_m c_m}{\alpha_s \pi D_s}, \quad \beta = \frac{\alpha_l D_l}{\alpha_s D_s},$$

where μ_l and μ_m are linear densities (mass per length unit), c_l and c_m are specific heats, α_l is the liquid/metal heat transfer coefficient and α_m is the steam/metal heat transfer coefficient. Assuming that the specific heats c_l and c_m are constant, coefficient T_m is also constant, whereas τ_l and β depend on the tube speed $v(t)$ through coefficient α_l according to the Dittus Boelter relation (**17**):

$$\alpha_l = Kv^\eta, \quad \eta \cong 0.8.$$

For reference sake, the specific heat exchanger considered in this paper is quantitatively defined as follows: the tube is 2.44 m long, with diameters $D_l = 0.0547$ m and $D_s = 0.0613$ m; the nominal heat transfer coefficients are given by $\bar{\alpha}_l = 754$ J m^{-2} s^{-1} °C^{-1} and $\bar{\alpha}_s = 3510$ J m^{-2} s^{-1} °C^{-1}; the specific heats are $c_l = 1$ kcal kg^{-1} °C^{-1} and $c_m = 0.094$ kcal kg^{-1} °C^{-1}, respectively; the linear densities are given by $\mu_l = 0.223$ kg m^{-1} and $\mu_m = 0.532$ kg m^{-1}.

Since the problem is to control the fluid outlet temperature by adjusting the fluid flow rate, realistic models of the fluid flow rate actuator and of the fluid outlet temperature transducer have to be introduced. The actuator is a valve, with infinitely rapid dynamics with respect to the heat exchanger. Therefore, we will assume that it is characterized by the nonlinear saturating characteristic of Fig. 2, where $u(t)$ represents the position

of the valve and $v(t)$ is the liquid speed. The extremal values $v_m = 0.1$ m s^{-1} and $v_M = 0.7$ m s^{-1} are assumed. The $u = 0$ position corresponds to the nominal value of the speed $\bar{v} = 0.3$ m s^{-1}. The transducer is described by the following first order linear model:

$$\dot{\tilde{\vartheta}}_{lo}(t) = -\frac{1}{\tau}\tilde{\vartheta}_{lo}(t) + \mu_T \vartheta_{lo}(t)$$

where $\tilde{\vartheta}_{lo}(t)$ is the transducer output. In the sequel, the values $\mu_T = 1$, $\tau = 1$ s are assumed for the coefficients.

Refer to Ref. (**18**) for the details regarding the integration method adopted for the simulation of the process.

III. Nonlinear Modelling of the Heat Exchanger

3.1. *Definition of the neural network structure*

First of all, we concisely summarize the results of Ref. (**19**), where the objective was the development of an *ARMAX* model of the type:

$$A(z)\hat{y}(t+1/t) = B(z)u(t) + C(z)e(t+1)$$

to describe the heat exchanger dynamics near a specific equilibrium point. In particular, the steady-state corresponding to $\bar{v} = 0.3$ m s^{-1}, $\bar{\vartheta}_{li} = 65°C$, $\bar{\vartheta}_{lu} = 98.765°C$, $\bar{\vartheta}_v = 120°C$ is taken as the nominal point. As an input signal $u(k)$ a Gaussian white noise was taken, with variance chosen to be small enough to preserve linear behaviour near the nominal point. The disturbances ϑ_{li} and ϑ_s are kept constant and equal to their nominal values.

The *ARMAX* modelling is a linear discrete-time methodology, suited for digital control purposes. Therefore, a sampling of the input–output data is necessary: a 1 s sampling time was selected on the basis of experimental observation of the settling time values near the nominal point. Four thousand input–output data were collected and processed by the identification algorithm. Among the various linear models identified on this set of data, an *ARMAX*(4, 10, 7) has been selected (**19**).

Turning to the nonlinear realm, we will consider a nonlinear *ARX* model of the type

$$\hat{y}(t+1/t) = f(y(t), \quad y(t-1), \ldots, y(t-3); \quad u(t), \quad u(t-1), \ldots, u(t-9)) \qquad (2)$$

where the orders 3 for the "*AR* part" and 9 for the "*X* part" are taken to be coincident with the orders of the same parts in the above *ARMAX* modelling. The nonlinear function $f(\cdot)$ has been selected in a black box manner by resorting to a neural network of the multilayer perception type with two hidden layers. This choice has been based on a set of experimental simulation trials in which the performance of a number of feedforward neural networks has been compared. The number of hidden neurons has also been optimized by trial and error on the basis of many identification runs carried out on neural networks with increasing complexity. A reasonable dimension for the neural network has been defined to be 14 neurons in the first hidden layer and 7 in the second.

Remark. In model (2) a fixed choice of the input–output structure was made: this

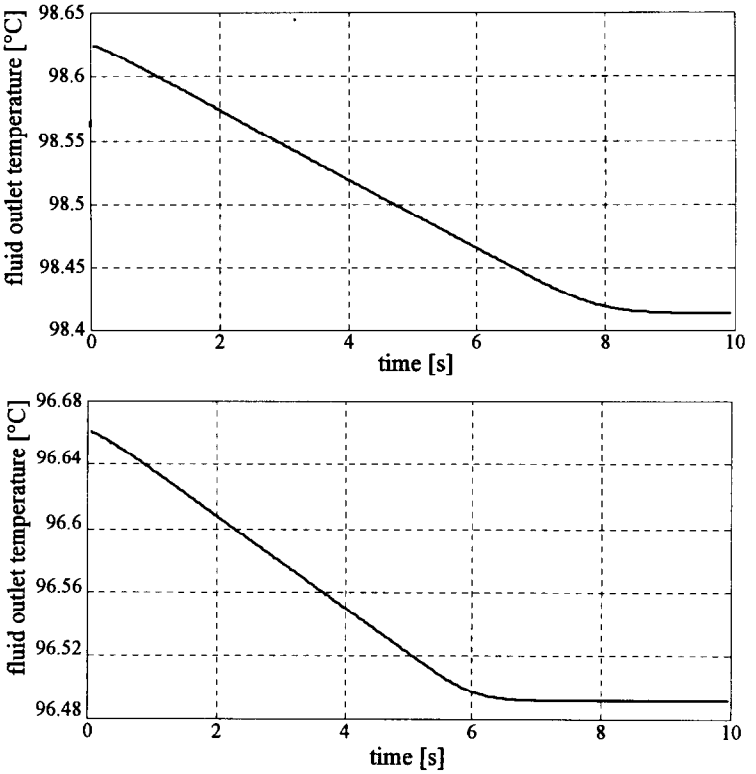

FIG. 3. Variability of the settling time with the initial conditions. (a) Transient generated by $\Delta v = +0.01$ starting from a steady-state corresponding to $\bar{v} = 0.3$. (b) Transient generated by $\Delta v = +0.01$ starting from a steady-state corresponding to $\bar{v} = 0.4$.

corresponds to the assumption that the same model structure applies globally to the system dynamics. In fact, it can be shown that the transient length varies considerably with the steady-state point. For example, an increment of 0.01 m s^{-1} of the liquid speed determines a much different transient length if the initial conditions correspond to the steady-state associated with $\bar{v} = 0.4$ m s^{-1}, or to the nominal point ($\bar{v} = 0.3$ m s^{-1}). In the first case, the transient ends in ~ 6 s, whereas in the nominal case the settling time is more than 8 s (Fig. 3). For equal input variations, the settling time is greater for lower steady-state speeds, so that more past input terms should be included in the model (**20**).

3.2. *The training set*

The quality of the model obtained is intimately related to the choice of the set of data employed in the learning phase. More precisely, the set of input–output data used in the training phase should ideally contain as much information as possible on the dynamics of the process. It is also important that the training set be homogeneous and balanced, in order to obtain a comparable prediction accuracy in the whole operation region. Therefore, it is necessary that not only the inputs, but also the outputs of the process scan the entire domain of interest. Summarizing, the identification experiment

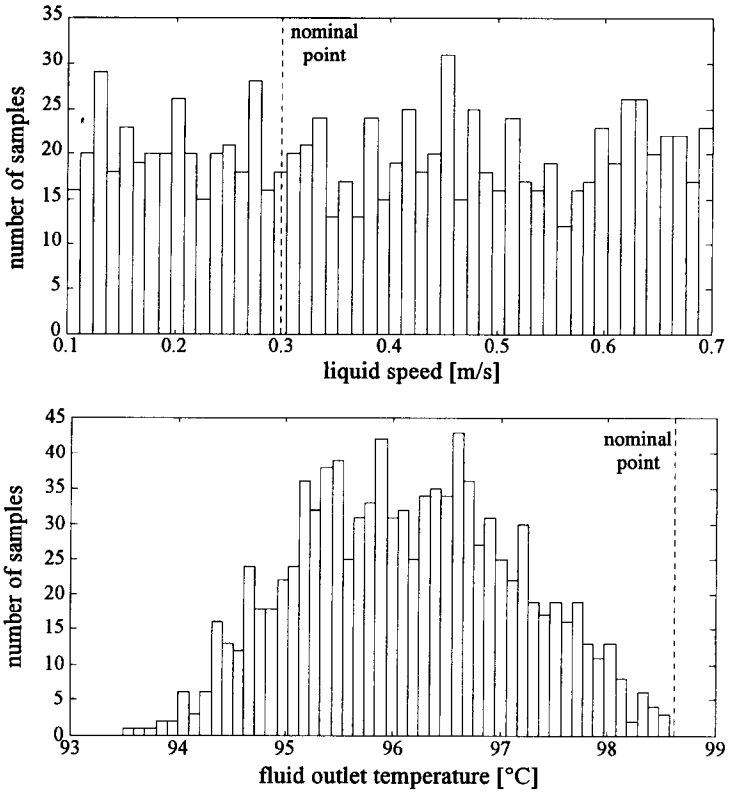

FIG. 4. Distribution of the input–output data (1000 samples).

which yields the data for the training set must be designed so that the input sequence excites the process dynamics as completely and uniformly as possible. This task is far from trivial due to the nonlinearity of the system which filters the input data. In other words, it is impossible to fix arbitrarily the joint probability of the network input data.

Various sets of training data have been derived using input signals with different probabilistic distribution. First of all, the case of uniform distribution of the input signal has been considered. The input and output data are represented in Fig. 4. The resulting distribution of the output values is concentrated on low temperatures and the number of samples of the output near the nominal point is insufficient. If this set of data is used in the learning phase, a model which is inaccurate right near the nominal point of the process is obtained.

To obtain a fair distribution of the output samples one has instead to resort to a composite input excitation, for example as follows. Use first an input signal made up of 100 steady-state samples in the nominal point, followed by 100 Gaussian distributed samples centred on the nominal steady-state speed, 600 beta-distributed samples with $a = 2.142, b = 1.415$, weighted towards low speed values, 1200 beta-distributed samples with $a = 1.621$, $b = 3.829$, which privilege high speeds, and finally 2000 uniformly distributed samples (for details on the beta distribution, see the Appendix). The distri-

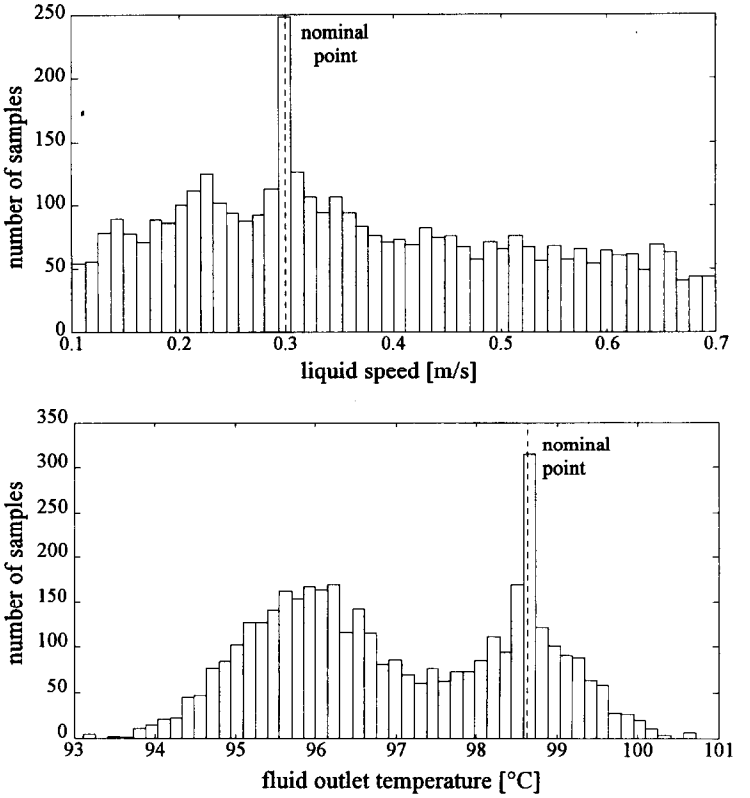

FIG. 5. Distribution of the input–output data (4000 samples).

bution of the input–output data of the training set obtained in this way is shown in Fig. 5.

In this case, the region near the nominal point is well represented and the distribution of the output samples is globally more uniform. Notice, however, the almost complete lack of low ($T_{lu} < 94°C$) and high ($T_{lu} > 100°C$) temperature samples.

3.3. *Identification of a neural forward model*

The identification of the neural model has been achieved through extensive training of the neural network on the data set of Fig. 5, by means of the classical Back-propagation method (**21**). The accuracy of the obtained neural forward model can be appreciated in the following figures, where the one-step prediction performance of the best model is reported, with respect to a uniformly distributed random input (Fig. 6), and the gain (Fig. 7) and settling time (Fig. 8) of the best neural model and the best *ARMAX* model are compared.

IV. Predictive Control with Neural Networks

In order to control the fluid outlet temperature, a digital controller must be designed. In general, such a controller will implement a control law of the type:

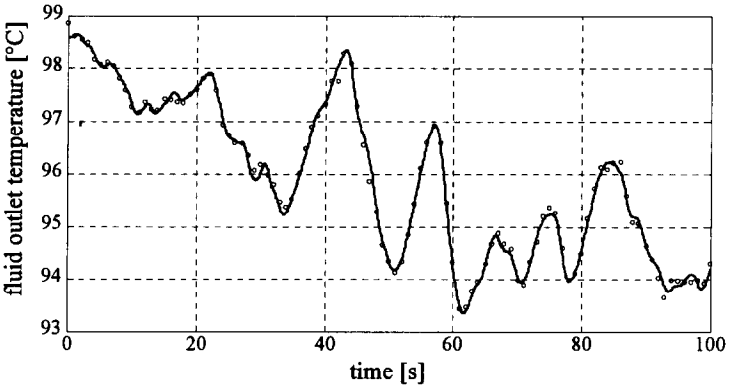

FIG. 6. Outlet temperature of the simulated process (continuous line) and the neural prediction form model (circles).

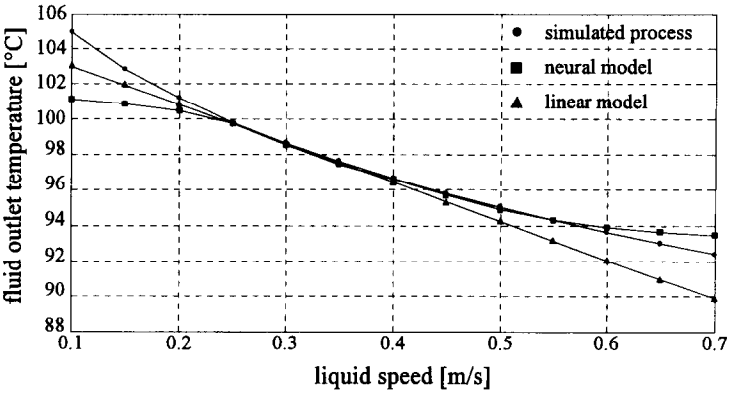

FIG. 7. Steady-state outlet temperature vs liquid speed for the simulated process, the neural prediction form model and the linear *ARMAX* model.

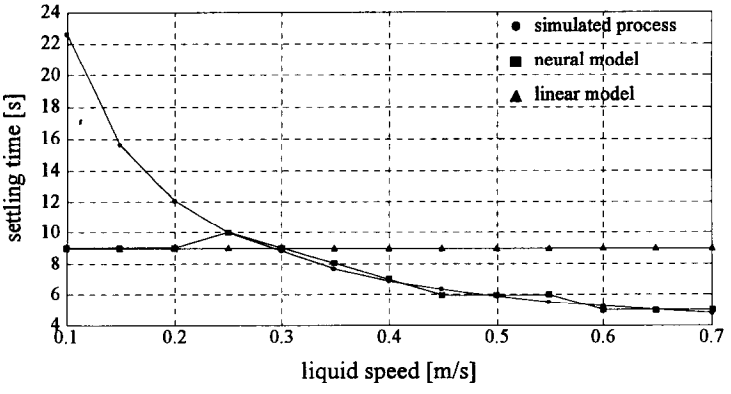

FIG. 8. Settling time vs liquid speed for the simulated process, the neural prediction form model and the linear *ARMAX* model.

$$u(t) = g(u(t-1),\ u(t-2),\ldots,\ u(t-9);\ y(t),\ y(t-1),\ldots,y(t-3);\ y^{\circ}(t)), \tag{3}$$

where $y^{\circ}(t)$ denotes the set-point and $g(\cdot)$ is a nonlinear function. The choice of the specific nonlinearity $g(\cdot)$ is a designer task: two possibilities are explored in the subsequent subsections.

4.1. *Nonlinear MV control with neural networks*

The possibility to employ an *MV* inverting controller for the control of the heat exchanger has been first analyzed. The nonlinear *MV* control strategy amounts to choosing the controller of Eq. (3) so as to minimize the following performance index:

$$J = \|y(t+d) - y^{\circ}(t)\|^2. \tag{4}$$

In the stochastic formulation index Eq. (4) represents the output variance, hence the denomination of the design methodology. The control law computation requires that condition $y(t+d|t) = y^{\circ}(t)$ is imposed, where $y(t+d|t)$ denotes a d-step prediction form model. The latter is unknown and has to be estimated, so that a sub-optimal controller is in fact obtained by solving equation $\hat{y}(t+d|t) = y^{\circ}(t)$ with respect to $u(t)$. Since

$$J = \|y(t+d) - y^{\circ}(t)\|^2 = \|(\hat{y}(t+d|t) - y^{\circ}(t)) + (y(t+d) - \hat{y}(t+d|t))\|^2$$
$$= \|y(t+d) - \hat{y}(t+d|t)\|^2$$

the control system performance increases as the estimated prediction form model gets closer to the actual system. Notice that this method only applies if the inversion operation required to solve the implicit control law is admissible. Furthermore, if the *MV* control law $\hat{y}(t+d|t) = y^{\circ}(t)$ can only be solved approximately, an additional error will affect the control system performance.

In our case, the 1-step prediction form model (2) has been estimated, which implies the following implicit *MV* control law:

$$y^{\circ}(t) = f(y(t),\ldots,y(t-3);\ u(t),\ldots,u(t-9)).$$

If the inverse function $g(\cdot)$ of $F(\cdot)$ exists and is available, this equation can be solved with respect to $u(t)$ to given an explicit law of type (3). The resulting *MV* control system is shown in Fig. 9.

In the specific case of the heat exchanger the dynamics are invertible, but the inverse model turns out to be at the limits of stability. In fact, an inspection of the linearized

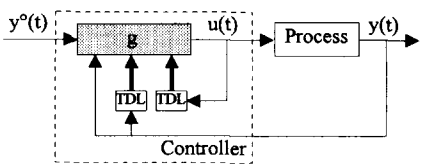

FIG. 9. The *MV* control system; tapped delay times are denoted with the acronym TDL.

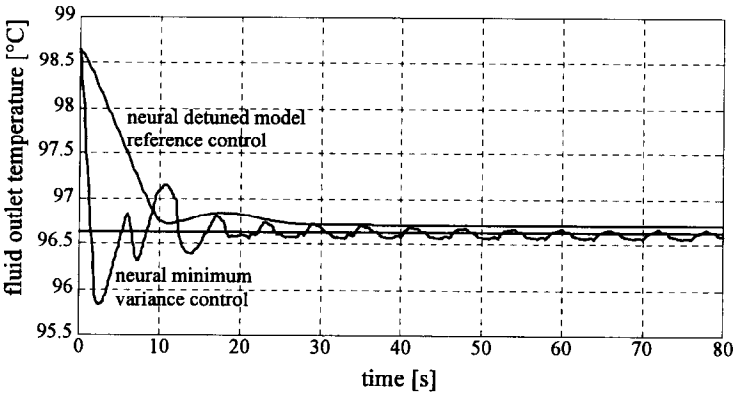

FIG. 10. Response of the neural minimum variance controller and the neural detuned model reference controller: fluid outlet temperature for a 2°C decrease in the reference signal.

model readily shows that an infinite number of zeros are located on the imaginary axis, no matter what nominal point is selected for the linearization. Consequently, the exact inverse model has poles on the imaginary axis (**14**). However, since the training of the neural network tends to operate a low frequency approximation of the dynamics to be identified, the inversion effect might not involve these imaginary axis poles, so providing a stable control system despite the critical dynamics of the plant.

The training of the inverse neural network employs the same data set used in the forward modelling phase. The network inputs are $u(t-1), u(t-2), \ldots, u(t-9), y(t+1), y(t), \ldots, y(t-3)$ whereas the output is the estimate of $u(t)$. After the training phase the neural network is used as a *causal* controller by substituting the input $y(t+1)$ with the reference signal $y^{\circ}(t)$. Observe that a good selection of the training set is even more important for the correct identification of the process inverse dynamics than it is for the estimation of the forward model.

The closed loop performance of the so-designed nonlinear minimum variance controller is characterized by an extremely short transient, but also by a non-zero steady-state off-set error and oscillations, though these effects are contained. In Fig. 10 an example of the closed loop response of the nonlinear control system with a neural minimum variance controller is shown: a small amplitude step signal ($\bar{T}_{\text{lu}} - 2°\text{C}$) on the set-point variable is imposed to the control system.

In conclusion, an inverting type of controller, like that obtained by means of a minimum variance design, inevitably causes steady-state permanent oscillations, due to the non-minimum phase characteristics of the process. The problem is remarkable in the linear case (**18**), but becomes much less important when a low frequency approximation of the process inverse dynamics is operated, as implicitly happens during the neural network learning procedure. In the linear case the usual remedy would be to introduce a control penalization factor in the minimum variance performance index: this has the effect that the closed-loop poles are moved inside the unit circle, so that the global stability properties of the closed-loop system are improved. The same method proves effective also in the nonlinear case, as will appear from the simulations reported in the following section.

4.2. *Rejection of the steady-state permanent oscillations: nonlinear GMV control with neural networks*

The design of a general nonlinear *GMV* controller follows the outline given in Ref. (**13**). The following performance index is minimized:

$$J = \|P(z)y(t+d) - R(z)y^{\circ}(t) + Q(z)u(t)\|^2 \tag{5}$$

where $P(z)$, $Q(z)$ and $R(z)$ are suitable user defined biproper and stable rational functions. For our purposes, it will suffice to consider a detuned model reference control [$P(z) = R(z) = 1$]. The implicit form of the control law is derived as follows:

$$\hat{y}(t+1|t) - y^{\circ}(t) + Q(z)u(t) = 0$$

or

$$Q_{\mathrm{D}}(z)\hat{y}(t+1|t) - Q_{\mathrm{D}}(z)y^{\circ}(t) + Q_{\mathrm{N}}(z)u(t) = 0$$

where $Q(z) = Q_{\mathrm{N}}(z)/Q_{\mathrm{D}}(z)$ is a rational transfer function which weights the control action. Without loss of generality we can assume that $Q_{\mathrm{D}}(z)$ is a monic polynomial. To obtain a causal form of the implicit control law, the prediction form model equation (2) is employed:

$$f(y(t), \quad y(t-1), \ldots, y(t-3);$$

$$u(t), \quad u(t-1), \ldots, u(t-9)) + Q_{\mathrm{N}}(z)u(t) = Q_{\mathrm{D}}(z)y^{\circ}(t) + (1 - Q_{\mathrm{D}}(z))y(t).$$

The preceding expression is conveniently rearranged as:

$$\bar{f}(y(t), \quad y(t-1), \ldots, y(t-3); \quad u(t), \quad u(t-1), \ldots, u(t-9)) = \tilde{y}^{\circ}(t)$$

where the following definitions are made:

$$\bar{f}(y(t), \ldots, y(t-3); \quad u(t), \ldots, u(t-9)) = f(y(t), \ldots, y(t-3);$$

$$u(t), \ldots, u(t-9)) + q_{\mathrm{N0}}u(t),$$

$$\tilde{y}^{\circ}(t) = Q_{\mathrm{D}}(z)y^{\circ}(t) + (1 - Q_{\mathrm{D}}(z))y(t) + (q_{\mathrm{N0}} - Q_{\mathrm{N}}(z))u(t)$$

and q_{N0}, q_{D0} are the leading coefficients of polynomials $Q_{\mathrm{N}}(z)$ and $Q_{\mathrm{D}}(z)$, respectively. Notice that $q_{\mathrm{D0}} = 1$ since $Q_{\mathrm{D}}(z)$ is monic.

If function $\bar{f}(\cdot)$ is invertible with respect to $u(t)$, a closed-loop controller of the type of Eq. (3):

$$u(t) = g(u(t-1), \ldots, u(t-9); \quad y(t), \ldots, y(t-3); \quad \tilde{y}^{\circ}(t)),$$

can be employed where function $g(\cdot)$ realizes the inverse of $\bar{f}(\cdot)$.

In fact, the task of inverting $f(\cdot)$ can also be left to a neural network, so avoiding cumbersome and complicated numerical procedures of inversion. To this purpose, during the off-line learning phase the neural network will be fed with the auxiliary signal $y(t+1) + q_{\mathrm{N0}}u(t)$ at the input, instead of $\tilde{y}^{\circ}(t)$, which is not available. Thus, it is possible to perform the network learning by means of the same data set used in the previous section.

In the following simulations the control action has been penalized by a weighting function $Q(z)$ of the form:

$$Q(z) = \beta \frac{1-z^{-1}}{1-\alpha z^{-1}}, \quad \alpha < 1,$$

where the numerator is structured so as to penalize the variations (and hence the oscillations) of the control variable, whereas the role of the denominator is to forget past signals, with a rate determined by coefficient α. In order to perform a fair comparison with respect to the corresponding linear detuned model reference controller described in Ref. (**16**), the same $Q(z)$ function, with $\alpha = 0.7$ and $\beta = 4$, has been adopted in both cases. On the whole, the so-designed neural controller features a quick response, with smooth transients and negligible oscillations. A slight increase in the steady-state off-set error is detected with respect to the minimum variance neural controller. Compare in Fig. 10 the step response transients of the neural detuned model reference control and the neural minimum variance control schemes. Comparisons of these neural controllers with their linear counterparts are provided in Section 4.4.

4.3. *Off-set error rejection*

The presence of steady-state off-set errors has been detected for both the designed neural controllers: this is not surprising, since the neural networks are just dynamic approximators and their steady-state errors reflect into a control off-set. Different control architectures with an integrator block can be used to reject these errors: this block is conceived to cooperate with the previously designed neural controller so that no further neural network training is required.

In particular, we will investigate the effect of introducing a nonlinear discrete time integrator in parallel with the neural controller as in Fig. 11. The additional block is composed of a linear discrete time integrator whose gain is modulated through an algebraic saturating function. This is required to avoid affecting the control action of the neural controller with the integral effect during the initial part of the transient, when the error is necessarily large. The algebraic function is designed so that when the output error is large the integral action is almost absent and it gets active when the error decreases below a certain threshold.

In fact, the designed neural controllers exhibit a most satisfying behaviour during transients, reaching quickly and smoothly an almost correct equilibrium value. Thanks to the algebraic saturating block, these dynamics are preserved in the transient, and the integrator effect prevails only in a second phase to correct the residual error. The

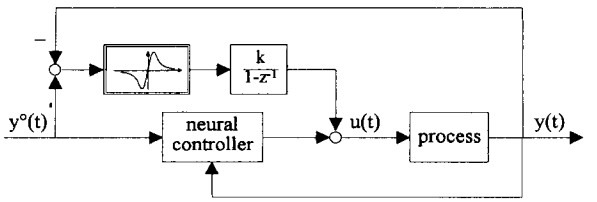

FIG. 11. Neural controller with nonlinear integrator block in parallel.

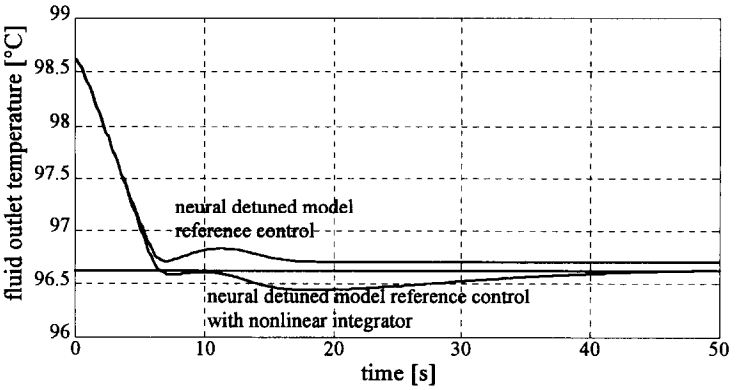

FIG. 12. Response of the neural detuned model reference controller with and without the nonlinear integrator block in parallel: fluid outlet temperature for a 2°C decrease in the reference signal.

same transient of Fig. 10 has been used as a test for the neural detuned model reference controller with the nonlinear integrator block in parallel: the output response is shown in Fig. 12.

4.4. *A comparison with traditional control techniques*

Various simulations have been carried out to validate the designed neural controllers and to compare their performances with those of the linear controllers developed for the same application in Ref. (**18**). The characteristics of the controllers taken into consideration are composed in Table I. On the whole, the proposed neural controllers yield medium to high performance and the control system behaviour is uniform in the operation region. The nonlinear structure of these controllers determines sudden changes in the dynamics of the response, when during the transient the output variable commutes from one operation region to another. The control action is generally modest and gradual, without abrupt peaks or oscillations. This is partly due to the particular training set employed for the neural network learning, which does not contain many samples corresponding to the extremal values of the control variable. Finally, the addition of the integral block eliminates the steady-state off-set error.

The superiority of the neural controllers with respect to the entire family of linear controllers is apparent when the system is induced to operate far from the nominal point: in the case of the linear controllers small displacements of the nominal temperature set-point can result in badly deteriorated dynamics. In addition, when linear controllers without integral action are employed, the steady-state off-set error increases rapidly away from the nominal point. On the other hand, for the neural controllers, the steady-state off-set error is independent of the particular nominal equilibrium point considered, and is determined by the efficiency of the learning phase. Finally, notice that the characteristics of the neural controllers, in terms of response speed, are comparable to those of the linear controllers.

In the following, the closed-loop system response with respect to different step signals on the set-point is shown, in order to compare the performance of the different types

TABLE I
Characteristics of the controllers taken into consideration

Type	Response speed	Steady-state off-set error	Control action	Robustness w.r.t. disturbances
PID (Ziegler–Nichols)	very good	zero	intense, abrupt oscillations	yes
Pole Assignment + integrator	good	zero	medium	yes
MV	—	—	intense, abrupt oscillations	no
GMV (detuned model reference control)	good	non zero, depends on the nominal point	modest	no
Pole Zero Assignment (discrete time)	good	non zero, depends on the nominal point	modest	no
Neuro-*MV*	excellent	non zero, oscillations	modest	no
Neuro-*GMV* (detuned model reference control)	good	non zero, independent of the nominal point	modest	no
Neuro-*GMV* (detuned model reference control) + integrator in parallel	good	zero	modest	yes

of controllers. In these simulations the disturbances are kept constant to their nominal value.

In Fig. 13 the responses of the neural minimum variance controller and of a linear minimum variance controller with respect to a reference signal step of variation of $-1\,^{\circ}\mathrm{C}$ are compared. The linear controller is capable of stabilizing the system, due to the non-minimum phase nature of the controlled system. On the contrary, thanks to the filtering effect of the learning phase, the neural minimum variance controller achieves an acceptable performance: the dynamics of the controlled system show the presence of low damped oscillations with small amplitude.

The limits of the linear controllers can be fully appreciated in the simulation of Fig. 14, where the neural minimum variable controller is compared to a classical PID controller, tuned according to the Ziegler–Nichols rules. Both control systems are fed with a low amplitude square wave (the set-point varies from $T_{\mathrm{lu}}-2^{\circ}\mathrm{C}$ to $T_{\mathrm{lu}}-2^{\circ}\mathrm{C}$). Though modest step variations are exerted on the reference signal, the PID control

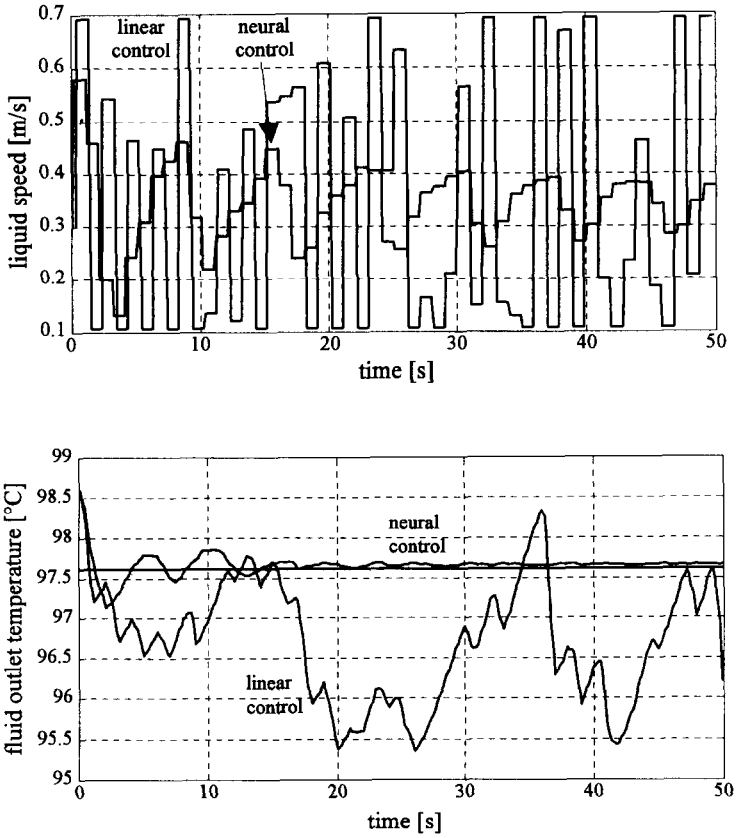

FIG. 13. Responses of the neural minimum variance controller and of a linear minimum variance controller: fluid outlet temperature for a 1°C decrease in the reference signal.

system performance appears to be extremely unsatisfactory at the higher temperature equilibrium point: this is probably due to the excessive control action which results in a severe overshoot ($\sim 50\%$). The neural minimum variance controller displays the limit of stability behaviour discussed above: persistent oscillations are present both in the control and the controlled variable. On the other hand, the response speed is good and the amplitude of the oscillations is very small. In addition, the maximum overshoot amounts to $\sim 30\%$ at the lower temperature equilibrium point.

The neural detuned model reference controller (without integral action) is compared with its linear counterpart in Fig. 15. In both cases a step variation of $-5°C$ on the set-point is applied. As expected, the linear control system performance degrades away from the nominal point: a severe steady-state off-set error ($\sim 0.4°C$) is observed and the initial part of the transient features oscillatory behaviour and an $\sim 10\%$ overshoot. Though slower, the neural controller displays a better performance on the whole, approaching the steady-state value with very smooth dynamics. What is more, the control action is very modest at the beginning of the transient and reaches gradually its steady-state value in a second phase.

We conclude this comparison section by presenting the performances of two integral

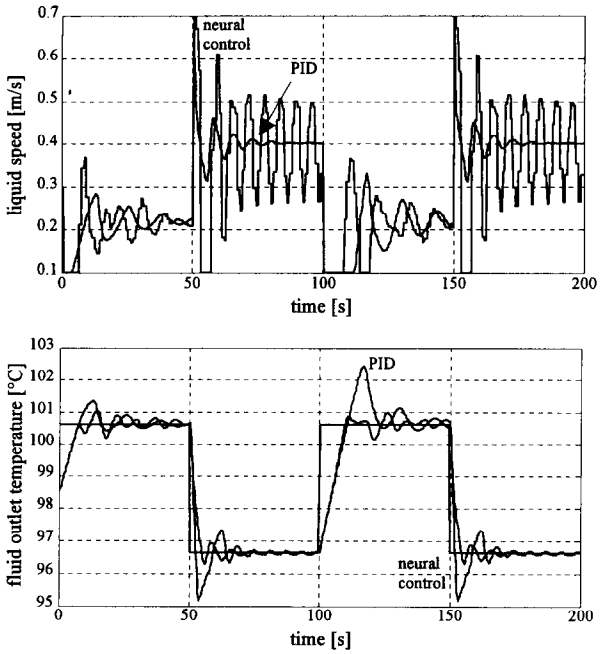

FIG. 14. Responses of the neural minimum variance controller and of a PID controller: fluid outlet temperature for a low amplitude square wave reference signal.

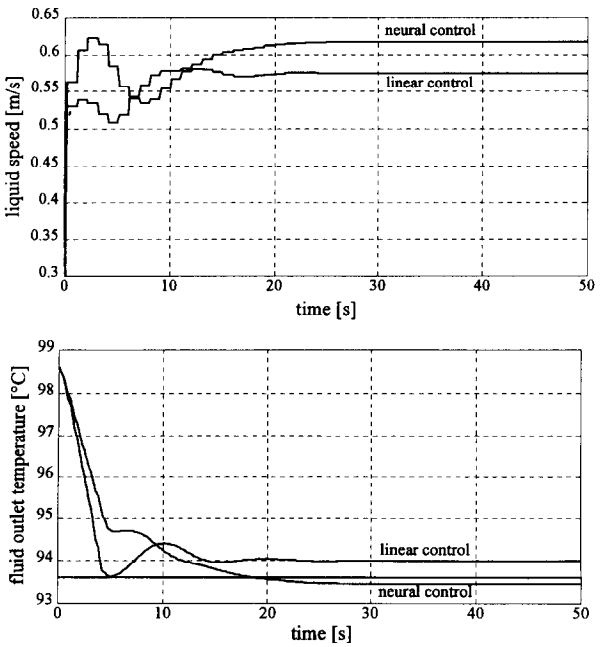

FIG. 15. Responses of the neural detuned model reference controller and of a linear detuned model reference controller: fluid outlet temperature for a $-5°C$ step on the reference signal.

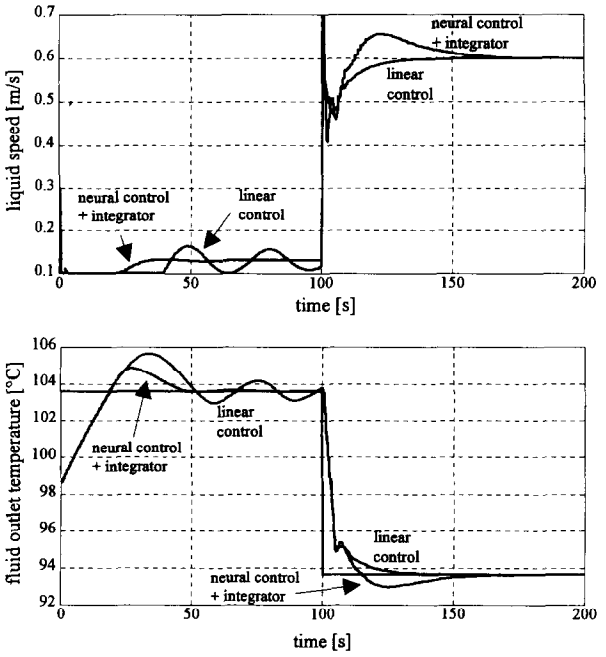

FIG. 16. Responses of the neural detuned model reference controller with integral action and of a pole assignment controller with integral action: fluid outlet temperature for a high amplitude square wave reference signal.

type controllers (Fig. 16). A linear controller with integral action designed by a pole assignment technique is compared with the neural detuned model reference controller with nonlinear integral action. The reference signal is a high amplitude square wave: the set-point varies from $T_{lu}-5°C$ to $T_{lu}+5°C$. The linear controller, designed with all the poles in -1, is not capable of yielding a comparable performance in the entire operation region scanned by the reference signal. In particular, a low performance is achieved at the higher temperature equilibrium point. The neural controller, instead, displays smooth and uniform dynamics and no steady-state off-set error, thanks to the integral action. Notice that the integral action does not heavily affect the control system dynamics: only a minor overshoot is induced. Observe also that the two control systems display comparable response speed.

Acknowledgements

This paper has been supported by the European HCM project "Nonlinear and adaptive control", the Italian MURST project "Model identification, system control and signal processing", the CNR special project "Neural Networks for Control Systems" and CNR—Centro di Teoria dei Sistemi, Milano.

References

(1) K. Anbumani, L. M. Patnaik and I. G. Sarma, "Self-tuning minimum-variance control of nonlinear systems of the Hammerstein model", *IEEE Trans. Autom. Control*, Vol. 26, No. 4, pp. 959–961, 1981.

(2) S. Svoronos, G. Stephanopoulos and R. Aris, "On bilinear estimation and control", *Int. J. Control*, Vol. 34, pp. 651–684, 1981.
(3) D. Dochain and G. Bastin, "Adaptive identification and control algorithms for nonlinear bacterial growth systems", *Automatica*, Vol. 20, pp. 612–634, 1984.
(4) M. Kung and B. F. Womack, "Discrete time adaptive control of linear systems with preload nonlinearity", *Automatica*, Vol. 20, No. 4, pp. 447–479, 1984.
(5) M. Agarwal and D. E. Seborg, "Self-tuning controllers for nonlinear systems", *Automatica*, Vol. 23, No. 2, pp. 209–214, 1987.
(6) J. Zhang and S. Lang, "Indirect adaptive suboptimal control for linear dynamic systems having polynomial nonlinearities", *IEEE Trans. Autom. Control*. Vol. 33, No. 4, pp. 389–392, 1988.
(7) J. Zhang and S. Lang, "Explicit self-tuning control for a class of non-linear systems", *Automatica*, Vol. 25, No. 4, pp. 593–596, 1989.
(8) S. Bittanti and L. Picci, "Identification, Adaption, Learning—the Science of Learning Models from Data", Springer-Verlag, Berlin, 1996.
(9) W. T. Miller, R. S. Sutton and P. J. Werbos, "Neural Networks for Control", MIT Press, Cambridge, MA, 1990.
(10) K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks", *IEEE Trans. Neural Networks*, Vol. 1, No. 1, pp. 4–27, 1990.
(11) K. J. Hunt, D. Sbarbaro, R. Zbikowski and P. J. Gawthrop, "Neural networks for control systems—a survey", *Automatica*, Vol. 28, No. 6, pp. 1083–1112, 1992.
(12) S. Bittanti and L. Piroddi, "GMV technique for nonlinear control with neural networks", *IEEE Proc. D, Control Theory & Appl.* Vol. 141, pp. 57–69, 1994.
(13) L. Piroddi, "Neural Networks for Nonlinear Predictive Control", Ph.D. Thesis, Politecnico di Milano, (in Italian), 1995.
(14) S. Bittanti and R. Scattolini, "Optimal stochastic control of a liquid-saturated steam heat exchanger", in "Time Series Analysis: Theory and Practice" (Edited by O. D. Anderson), North-Holland, Amsterdam, 1982.
(15) S. Bittanti, D. W. Clarke, F. Romeo and R. Scattolini, "Self-tuning control of a liquid-saturated steam heat exchanger", in "Proc. of the 7th IFAC/IFIP Symposium of Real Time Digital Control Applications", pp. 631–638, Guadalajara, Mexico. 1983.
(16) S. Bittanti, F. Romeo and R. Scattolini, "Stochastic identification and digital control of a heat exchanger: a simulation test case", *J. Franklin Inst.* Vol. 318, pp. 29–57, 1984.
(17) W. M. Rohsenow and J. D. Hartnett, "Handbook of Heat Transfer", McGraw-Hill, New York, 1973.
(18) S. Bittanti, "Simulation, Identification and Control of a Heat Exchanger", Pitagora edn., Bologna, Italy, (In Italian), 1985.
(19) S. Bittanti, A. Cividini and R. Scattolini, "Identification of a liquid-saturated steam heat exchanger", in "Proc. of the 7th IFAC Symposium in Identification and System Parameter Estimation", pp. 168–173, Washington, 1982.
(20) J. Sjöberg, H. Hjalmarsson and L. Ljung, "Neural networks in system identification", in "Preprints of the 10th IFAC Symposium on Identification and System Parameter Estimation", Copenhagen, 1994.
(21) D. E. Rumelhart, G. E. Hinton and R. J. Williams, "Learning internal representations by error propagation", in "Parallel Distributed Processing: Explorations in the Microstructures of Cognition", Vol. 1 (Edited by D. E. Rumelhart and J. L. McClelland), MIT Press, Cambridge, MA, 1986.
(22) A. Mood, F. Graybill and D. Boes, "Introduzione alla Statistica", McGraw-Hill, New York, 1988.

Appendix: The Beta Probability Density

A beta-distributed input can be used to slide the centre of the output distribution.

The beta probability density (**22**) is defined by

$$f(x) = \frac{1}{B(a,b)} x^{a-1}(1-x)^{b-1}, \quad x \in [0,1], \quad a,b > 0$$

where

$$B(a,b) = \int_0^1 x^{a-1}(1-x)^{b-1}\,dx.$$

Parameters a and b determine the eccentricity of the function with respect to the maximum point, as shown in Fig. A1.

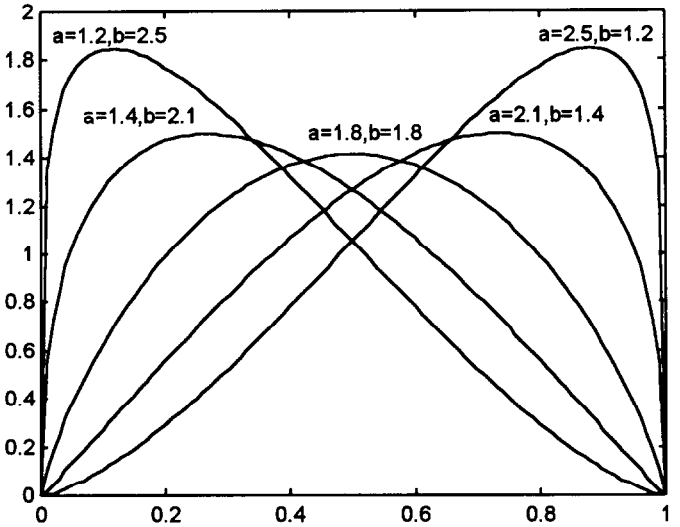

FIG. A1. $\beta(1.415, 2.122)$ probability density.