Journal of Theoretical and Applied Information Technology
15th July 2021. Vol.99. No 13

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A MULTIMODEL INTERNAL MODEL CONTROL APPROACH BASED ON NEURAL NETWORK

AMIRA SLIMANI[1], AYACHI ERRACHDI[1], MOHAMED BENREJEB[1]

[1]Tunis El Manar University, National Engineering School of Tunis, Department of Electrical Engineering

Le Belvédère B.P. 37, Tunis, 1002, Tunisia

E-mail: [1] amira.slimani@enit.utm.tn

ABSTRACT

This paper proposes a multimodel internal model control approach based on neural network, using a variable learning rate, for a nonlinear discrete system. The multimodel controller consists of two blocks : an inverse model and an internal model. Each block is based on neural network can be got directly, which simplifies the control law design and analyses greatly. Meanwhile, the way of model switch is developed based on neural network decision. This approach avoids the complex calculation when adjusting the controller parameter and overcomes the switch operation. By applying the proposed approach to a nonlinear system, simulation results demonstrate that the strategy has advantage of multimodel internal model control and could achieve better system performance than the classical one.

Keywords: *Multimodel Control, Internal Control, Neural Network, Nonlinear System, Learning Rate*

1. INTRODUCTION

The classical internal model control (IMC), as one of the advanced control arithmetic, has drowned more attention since it was used because it gives a simple yet effective framework for the analysis of control system [1]. This method is easy to be designed and regulated on-line, and is also adapt to eliminate unmeasured disturbance. The main key of this control strategy is to use two blocks; the inverse model and the internal model as a controller. But, sometimes, it is difficult to find the ideal inverse model so a little bit of time-varying parameters of nonlinear process or such added disturbances, the system performance would be deteriorated.

In recent times, neural network (NN) had been applied in nonlinear IMC design due to its good ability of approximate nonlinear functions [2][3]. However, for some industrial fields, there are many different unknown environments, for example, external disturbances, changes in subsystem dynamics, strong nonlinearity, variation of parameters, sudden change, the process characteristic changes drastically and falls outside training region. In this case, even though the NN model is available, it is difficult to design the NN inverse model unless the model is open-loop stable [4] and consequently the controller based on a fixed model will be unreliable and thus the system performance is affected seriously.

One way to solve this problem is to use multimodel internal model control based on neural network, in this paper. The main idea of this method is representing a dynamical system by a set of locally valid sub-models across the operating range.

Many researchers used the multimodel control in their applications. For example, reference [5] adopts multimodel adaptive control strategy in pH CSTR and gets satisfactory performance, but it is too difficult to get inverse model directly and adjust on-line.

Reference [6] describes a hybrid learning approach for networks constructed from ARX local models and normalized Gaussian basis functions. Using the method, multimodel control on complicated process is realized. But this arithmetic involves large calculation of parameters, and the traditional NN IMC needs to adjust process model and inverse model online. So the system is hard to meet the real-time requirement in most industrial system.

Reference [7] presents the design and the evaluation of a system that implements a multimodel, multi-level system using the artificial neural network architecture for eczema detection. In [7], multimodel system is defined as an architecture with different models depending on the input characteristic. The outputs of these models are integrated by a decision layer, thus multi-level, which computes the probability of an eczema case.

Reference [8] presents an important problems in the multimodel structure concerns the validity calculation which is a fundamental point especially when the process is corrupted with noise and/or its parameters are of high variations.

Reference [9] presents a methodology for identification and control of complex nonlinear plants using multimodel approach. The proposed methodology is based on fuzzy decomposition of the steady state map.

Reference [10] presents a multimodel controller based on the neural network sets for the time-variant nonlinear systems with uncertainty and time-delay.

Reference [11] presents a comparison between the methods of designing and implementation for multiple models and inverse model command.

In [12], a tracking control using multimodel switching is employed for uncertain nonlinear systems with time-varying delay. The paper [13] develops a fuzzy-weighted control scheme for control of multimodel MIMO systems.

In our paper, a multimodel internal model control method based on neural network, using a variable learning rate, is proposed. Indeed, the neural network is applied to construct the nonlinear process local-model, and the local-inverse model could be obtained from the process. These local-internal model and the local-inverse model represent a local-controller. A switch algorithm between these local-controllers is proposed. Because each local sub-model has nonlinear property, the error caused by local-model could be reduced. In this paper we propose an approach to solve the control algorithms switching.

This paper is organized as follows. Section 2 briefly introduces the statement problem. In section 3, the multimodel neural network approach is detailed. In section 4, the proposed multimodel neural network controller is presented. In section 5, an example of a nonlinear system is presented to illustrate the proposed efficiency of the methods. At last, concluding remarks are in Section 6.

2. STATEMENT PROBLEM

Consider the above discrete nonlinear single input single output (SISO) system with n inputs and n outputs expressed in terms of its difference equation in the following form.

$$y(k)=f(y(k-1),\ldots,y(k-n_y),u(k-1),\ldots,u(k-n_u)) \quad (1)$$

where $y(k)$ and $u(k)$ ($y(k)\in \mathbf{R}^n$, $u(k)\in \mathbf{R}^n$), are respectively the input and output vectors, n_y and n_u are the number of past system output and input respectively, $f(.)$ is the nonlinear function mapping specified by the model and k is the discrete time index.

The problem is to find a suitable controller to ensure that the system output $y(k)$ tracks as possible the desired reference $r(k)$.

To overcome this control problem, we propose a multimodel internal model control approach based on neural network, using a variable learning rate, as presented in figure 1.

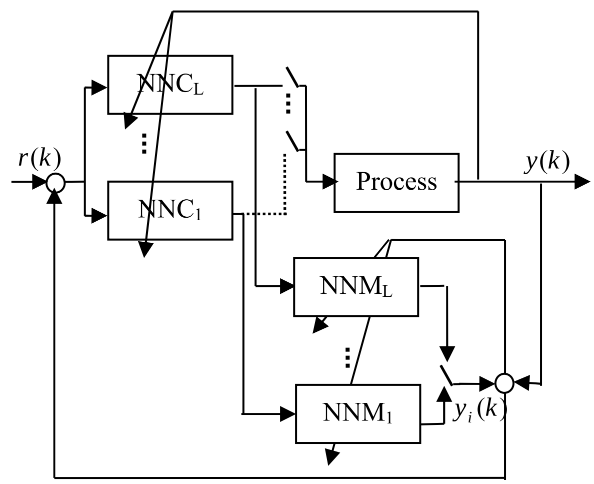

Figure 1: Multimodel internal model control approach based on neural network

Figure 1 presents the structure of multimodel internal model control approach based on neural network. The ith local neural network model, NNM$_i$, $i = 1,2,\ldots,M$, is the ith sub-model of the process and the local-neural network controller, NNC$_i$, is the ith corresponding controller.

For the ith controller, the output is given by:

$$u_{ri}(k)=f\left(\sum zf\sum wx\right) \quad (2)$$

where z and w are the synaptic weights of the neural network controller and $e_c(k) = r(k)-y(k)$ is the tracking error vector. Therefore, the adaptive mechanism is used for self-adjustment of the neural network controller to achieve the best tracking performance.

3. MULIMODEL NEURAL NETWORK APPROACH

The multimodel structure consists of M sub-model. Each sub-model can approximate the process function. However, the identification can be in online or offline operation. This methodology aims to replace the search for a single model that is often difficult to obtain by searching for a family of sub-models $g_i(k)$ and basic functions $\mu_i(.)$ as given by the following expression.

$$g(k)=\sum_{i=1}^{L}\mu_i(.)g_i(\varphi(k),\theta_i) \tag{3}$$

the whole characterizing the overall behavior of the system is

$$y(k)=g(\varphi(k),\theta) \tag{4}$$

A judicious choice of the structure of the sub-models $g_i(k)$ and basic functions $\mu_i(.)$ allows in theory to approach with an imposed precision any nonlinear behavior in a wide field of operation.

The multimodel approach consists, more precisely, of reducing the complexity of the system by breaking down its operating space into a finite number of operating zones.

Let consider $D_i \subset \mathbf{R}^n$ as a domain D_i resulting from the partition of the operating space of the system such as $D=\cup_i D_i$. The system exhibits a relatively homogeneous dynamic behavior in each operating zone. Whereas, $\xi \in \mathbf{R}$, a known vector variable, characteristic of the system and accessible by measurement in real time. It can be, for example, a measurable state variable and / or an input signal of the system.

In this paper, we have used the weighting functions : $\mu_i(\xi(k))$: $\mathbf{R}\rightarrow\mathbf{R}$ depending on the decision variables. These functions are associated with the different areas of operation and thus serve to gradually quantify the membership of the current operating point of the system to a given operating area.

The weighting functions are chosen to verify the following convex sum properties:

$$\sum_{i=1}^{L}\mu_i(\xi(k))=1 \quad ; \\ 0\le\mu_i(\xi(k))\le 1, \qquad \forall i=1,\dots,L \ \ \forall k \tag{5}$$

They can be constructed either from discontinuous derivative functions or from continuous derivative functions. We opt, in this paper, for weighting functions $\mu_i(.)$ constructed from Gaussian functions. This choice makes it possible to obtain, on the one hand, functions that are easy to compute in the multivariable case and, on the other hand, functions that are continuously differentiable.

They are obtained, in the monovariable case, from the function $w_i(.)$: $\mathbf{R}\rightarrow\mathbf{R}$

$$w_i(\xi(k))=\exp\left(-\frac{(\xi(k)-c_i)^2}{\sigma_i^2}\right) \tag{6}$$

whose parameters are centers c_i and dispersions σ_i.

The weighting functions are finally obtained by normalizing the functions $w_i(\xi(k))$ as follows:

$$\mu_i(\xi(k))=\frac{w_i(\xi(k))}{\sum_{j=1}^{L}w_j(\xi(k))} \tag{7}$$

It should be noted that a common dispersion $\sigma=\sigma_i$ ($\sigma_i=\sigma_{i,j}$ in the multivariable case) to all the weighting functions avoids so-called reactivation phenomena in which the same weighting function is significantly different from zero in two distinct operating zones [2] .

The output obtained by the multimodel approach is as follows:

$$y_{MM}(k)=\sum_{i=1}^{L}\mu_i(\xi(k))y_i(k) \tag{8}$$

where L is the number of sub-models, y_{MM} the output of the multimodel, $y_i(k)$ the output of the ith sub-model, $\xi(k)$ the decision variable and $\mu_i(\xi(k))$ the weighting function associated with the ith sub-model.

Figure 2 presents the principle of the fusion of the multimodel approach.

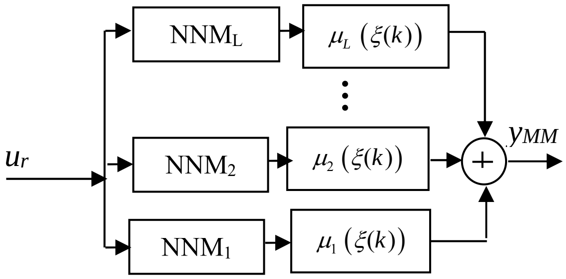

Figure 2: Principle of the fusion of the multimodel approach

The weighting functions make it possible to determine the relative contribution of each sub-model according to the zone where the system is evolving.

Obtaining such multimodel representation depends on the choice of the index variable ξ, the decomposition of the operating space into a number L of operating zones and the determination of the structure and parameters of each sub-model.

4. MULTIMODEL NEURAL NETWORK CONTROLLER

The multimodel neural network controller contains two blocks: the first is the inverse neural network model and the second is the internal neural network model.

4.1 Design of the ith Inverse Neural Network Model

In this section, we focus on to build the ith inverse neural network as a local-controller. The expression of the inverse model based on input-output measurement $(u_i(k), y_i(k))$ is given by :

$$u_{ri}(k) = f^{-1}(y_i(k+1), y_i(k), \ldots,) \tag{9}$$

where f^{-1} is an inverse nonlinear function of f. The ith inverse neural network model can be obtained in on-line identification as given by the following expression:

$$\nabla_w J(a_k) = \frac{\partial J(a_k, i=k)}{\partial a_k} = \frac{\partial(\frac{1}{2}\sum(e_{ci}(k))^2)}{\partial a_k} \tag{10}$$

or in offline identification as given by the following expression:

$$\nabla_w J(a_k) = \frac{\partial J(a_k, i=1,\ldots,N)}{\partial a_k} = \frac{\partial(\frac{1}{2}\sum(e_{ci}(k))^2)}{\partial a_k} \tag{11}$$

where a_k is z^1 or w^1.

The ith inverse neural network model is given by the samplest structure and presented by the following figure :

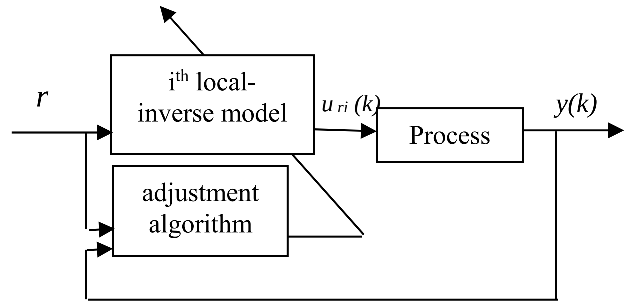

Figure 3: The proposed inverse model structure

The output of the ith inverse neural network model is given by the following expression:

$$\begin{aligned} u_{ri}(k) &= \lambda_c f(\sum_{j=1}^{n_4} z_j^1 f(\sum_{i=1}^{n_3} w_{ji}^1 x_i^1)) \\ &= \lambda_c f\left[z^{1T} F(Wx^1)\right] \end{aligned} \tag{12}$$

where $x^1 = \left[x_i^1\right]^T, i = 1,\ldots,n_3$ is the input vector

$$x_i^1(k) = \left[r_i(k) \quad \ldots \quad u_{ri}(k-1) \quad \ldots\right]^T \tag{13}$$

and

$W = \left[w_{ji}^1\right], i = 1,\ldots,n_1, j = 1,\ldots,n_4$,

$F(Wx^1) = \left[f(\sum_{i=1}^{n_1} w_{ji}^1 x_i^1)\right]^T, j = 1,\ldots,n_4$,

$z^1 = \left[z_j^1\right]^T, j = 1,\ldots,n_4$.

The estimation of synaptic weights is performed by minimizing a cost function J_c given by the following expression:

$$J_c = \frac{1}{2}(e_c(k))^2 = \frac{1}{2}(r(k) - y(k))^2 \quad (14)$$

After minimizing the cost function, the incremental changes of the output layer and the hidden layer are given as follows:

$$\Delta z_j^1(k) = -\eta_c \lambda_c \frac{\partial J_c(k)}{\partial z_j^1(k)} \quad (15)$$

$$\Delta w_{ij}^1(k) = -\eta_c \lambda_c \frac{\partial J_c(k)}{\partial w_{ij}^1(k)} \quad (16)$$

The update of the synaptic weights is as follows :

$$w_{ij}^1(k+1) = w_{ij}^1(k) + \eta_c \Delta w_{ij}^1(k) \quad (17)$$

$$z_j^1(k+1) = z_j^1(k) + \eta_c \Delta z_j^1(k) \quad (18)$$

with η_c, $0 \le \eta_c \le 1$, a variable learning rate given by the following expression:

$$\eta_c(k) = \begin{pmatrix} \lambda^2 f'(\sum_{i=1}^{n_1} w_{ji}^1 x_i^1)(F^T(Wx^1)F(Wx^1)) \\ +z^{1T} F'(Wx^1) z^1 x^{1T} x^1) \end{pmatrix}^{-1}$$

The ith inverse neural network model algorithm is given by some steps:

First step :

1- Find the necessary observation set $(u_i(k), y_i(k))$

2- Find the inverse neural network model parameters: layer number, neuron number by each hidden layer, activation function,

3- Compute the ith obtained neural network model using a reduced observation number:

$$u_{ri}(k) = \lambda f(\sum_{j=1}^{n_4} z_j^1 f(\sum_{i=1}^{n_3} w_{ji}^1 x_i^1)) \quad (19)$$

Second step :

4- At time instant ($k+1$), we have a new data $r_i(k+1)$, using the obtained input vector x^1, if the condition $e_{ci}(k+1) \le \varepsilon_1$, where $\varepsilon_1 > 0$ is a given small constant, is satisfied then the ith neural network inverse model, given by the equation (12), approaches sufficiently the behavior of the system,

5- Calculate the ith neural output at (k+1)

$$u_{ri}(k) = \lambda f(\sum_{j=1}^{n_4} z_j^1 f(\sum_{i=1}^{n_3} w_{ji}^1 x_i^1)) \quad (20)$$

6- Evaluate the gap $|e_{ci}(k+1)|$

7- If the condition $e_{ci}(k+1) \le \varepsilon_1$, is satisfied then the neural network inverse model provides sufficiently the control law $u_{ri}(k+1)$,

8- If $e_{ci}(k+1) \ge \varepsilon_1$, the synaptic weights of the neural network model is necessary, using the equation (17) and (18).

Using the obtained inverse model as a controller becomes a difficult task because the fast switching operation between M inverse model can cause dangerous behaviors whereas the slow swithching operation involves only the system's performances' alteration.

The switching operation is the main problems to be solved when designing the algorithms' switching block. Because this block may contain all algorithms' implementation and the algorithms' coefficients.

To overcome this problem, an internal model is proposed and the used controller becomes based on two blocks: the inverse local-model and the internal multimodel.

4.2 The design of the multimodel internal model

In this section, we focus on building the neural network controller which consists of the inverse model, which deltailed in the previous section, and a multimodel internal model.

The general structure of the ith internal neural model is given by the following relationship

$$y_i(k) = \lambda s(\sum_{j=1}^{n_2} z_j s(\sum_{i=1}^{n_1} w_{ji} x_{1i})) = \lambda s\left[z^T S(Wx)\right] \quad (21)$$

where $x = \left[x_i\right]^T, i = 1, \ldots, n_1$ is the input vector

$$x_i(k) = \left[y_i(k-1) \quad \ldots \quad u_{ri}(k-1) \quad \ldots\right]^T \quad (22)$$

and $W = \left[w_{ji}\right], i = 1, \ldots, n_1, j = 1, \ldots, n_2$,

$$S(Wx) = \left[s(\sum_{i=1}^{n_1} w_{ji} x_i))\right]^T, j = 1, \ldots, n_2,$$

$$z = \left[z_j\right]^T, j = 1, \ldots, n_2.$$

The estimation of synaptic weights is performed by minimizing a cost function J given by the following expression:

$$J_i = \frac{1}{2}(e_i(k))^2 = \frac{1}{2}(y(k) - y_i(k))^2 \quad (23)$$

After minimizing the cost function, the incremental changes of the output layer and the hidden layer are given as follows:

$$\Delta z_j(k) = \eta \lambda e_i(k) s'(\sum_{i=1}^{n_1} w_{ji} x_i)) S(Wx) \quad (24)$$

$$\Delta w_{ji}(k) = \eta \lambda s'(\sum_{i=1}^{n_1} w_{ji} x_i)) S'(Wx) z_j x^T e_i(k) \quad (25)$$

The update of the synaptic weights is as follows :

$$w_{ji}(k+1) = w_{ji}(k) + \eta \Delta w_{ji}(k) \quad (26)$$

$$z_j(k+1) = z_j(k) + \eta \Delta z_j(k) \quad (27)$$

with η , $0 \leq \eta \leq 1$, a variable learning rate is given by the following expression:

$$\eta(k) = \left(\begin{array}{l} \lambda^2 s'^2 (\sum_{j=1}^{t} w_{ij}^{(1)} x_j)(S^T(Wx)S(Wx) + \\ z_j^T S'(Wx) S'(Wx) z_j^T x^T x) e_1(k) \end{array} \right)^{-1}$$

The output obtained by the multimodel approach is as follows:

$$y_{MM}(k) = \sum_{i=1}^{L} \mu_i(\xi(k)) y_i(k) \quad (28)$$

where L is the number of neural network sub-models.

4.3 The proposed switch algorithm

In this section, the adaptive switch controller is proposed for the nonlinear system to achieve closed-loop stability and tracking. In fact, in this structure, we design the switching controller consisting of multi-controllers: an inverse neural network model and a multimodel neural network internal model, as shown in Figure 4.

It is proposed, in our paper, a method that provides good results for nonlinear system. It is supposed that just one single internal neural network controller (INNC), is to be maintained active, the best one, however all the other $(L-1)$ internal neural network controller rest inactive, during the current functioning of multimodel control systems with L model-algorithm pairs. The best neural network controller is found when the corresponding control error is the smallest. In the switching operation, we remove to use another neural network controller in the contrary case. Figure 4 presents this solution.

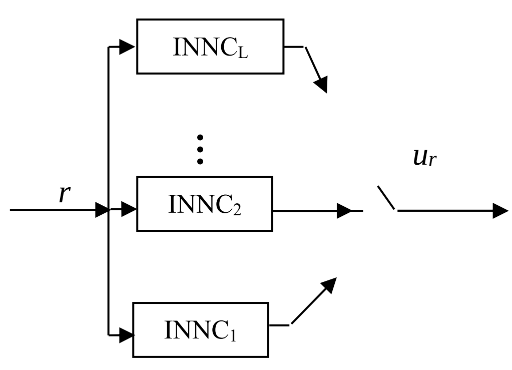

Figure 4: Switching algorithm

A summary of the proposed switch algorithm of the adaptive multimodel internal control for nonlinear system is presented.

Offline phase:

Initialization of parameters w^1, z^1, w and z of the inverse neural network model and the multimodel internal neural network model, respectively, is based on expressions (17), (18), (26) and (27) using a reduced number of observations M, ($M << N$), where N is the observation number.

Online phase:

(i) At time instant ($k+1$), we have a new data $r_i(k+1)$, using the obtained input vector x^1.

(ii) If the condition $e(k+1) \leq \varepsilon_2$, where $\varepsilon_2 > 0$ is a given small constant, is satisfied, then the internal multimodel neural network, given by the equation (28), approaches sufficiently the behavior of the nonlinear system.

(iii) If the condition $e_{c_i}(k+1) \leq \varepsilon_3$, $i = 1,..,L$, where $\varepsilon_3 > 0$ is a given small constant, is satisfied then the internal neural network controller provides sufficiently the control law $u_{ri}(k+1)$, using expression (12).

(iv) If the condition $e_i(k+1) \leq \varepsilon_4$ and $\varepsilon_4 \leq \varepsilon_3$, where ε_4 is a given small constant, is satisfied, the update of the neural network controller provides sufficiently the control law $u_r(k+1)$, using expression (12), $i = 1,..,L$.

(v) In this case, $\varepsilon_4 \leq \varepsilon_3$, the controller error is the smaller so we should use this internal neural network controller.

(vi) If the condition $e_i(k+1) \leq \varepsilon_4$ or $e_{c_i}(k+1) \leq \varepsilon_3$ are not satisfied, then the update of synaptic weights of the internal neural network controller is necessary, using expressions (17) and (18).

(vii) If $e_{c_i}(k+1) \leq \varepsilon_1$ is not satisfied, then the update of the synaptic weights of the internal multimodel neural network model is necessary, using equations (26) and (27)

(viii) End.

5. SIMULATION RESULTS

In order to evaluate the performance of the proposed method, simulations of a nonlinear system [16] are carried out.

$$y(k) = \frac{y(k-1)y(k-2)y(k-3)u(k-2)(y(k-3)-1)}{1+y^2(k-2)+y^2(k-3)} + \frac{u(k-1)}{1+y^2(k-2)+y^2(k-3)} \quad (29)$$

with $y(1) = y(2) = y(3) = 0$.

$u(k)$ and $y(k)$ are, respectively, the inputs and the outputs of the system.

In this section, we examine the effectiveness of the proposed multimodel internal model control approach based on neural network, using a variable learning rate, for a nonlinear discrete system given by the equation (29).

5.1 Limits of classical neural networks internal model control

The nonlinear discrete system given by the equation (29) is of third order and the input vector of the artificial neural network model is five where the used initial synaptic weights of the artificial neural model were random.

The fixed learning rate $\eta_m = 0.42$ for neural network model and $\eta_c = 0.45$ for neural network controller.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

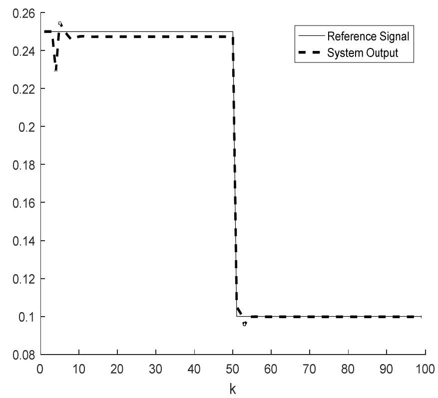

Figure 5: The evolution of the desired and the system output (Classical IMC)

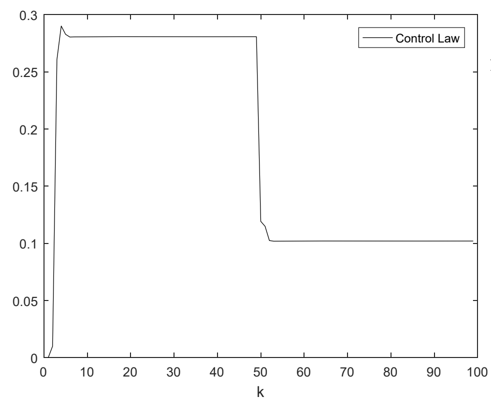

Figure 6: Control law signal

Figures 5 and 6 present, respectively, the evolutions of the desired reference trajectory and the system output in the case of the classical internal model control based on neural network and the control law. It may be noticed that the closed loop performances, in the case of no added disturbances, are good.

However, in the case of a random disturbances are added to the system output, the artificial neural network model may have a respective incapacity to provide good performance in closed loop.

Indeed, the choice of a classical artificial neural network model created by a one hidden layer and sigmoid as an activation function, thanks to its simple derivative, cannot be always sufficient in presence of strong nonlinearities.

Furthermore, by examining equation (29), it is obvious that accuracy of the modeling is closely linked to the initialization and the convergence of the weights of artificial neural network model. Poor initialization, a slow convergence of the weights which caused by a bad choice of fixed learning rate and the presence of disturbances can lead to poor performance of the control system.

To illustrate this problem, we consider the same system described by the equation (23), using the same learning rate for both blocks networks, the same initialization of the controller parameters. The performance in regulation and tracking are considerable deteriorated because initial weights of the mapping network were random, random disturbances were added and bad initialization of weights of the mapping network were used.

Figures 7 and 8 show respectively the results of system output and the control signal. In these figures a disturbance was added in the output of system. It may be noticed that the closed loop performances, in this condition, are deteriorated.

From these figures we noticed and proved the used classical neural network internal model control is limited. To overcome this problem a multimodel neural network should be used.

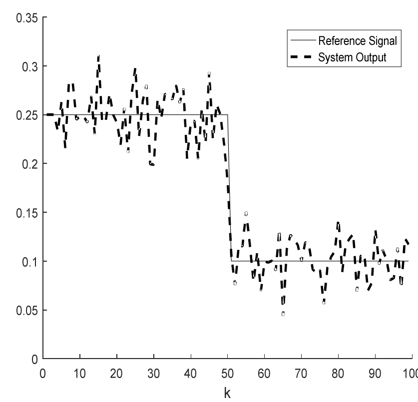

Figure 7: The evolution of the desired and the system output with disturbances (Classical IMC)

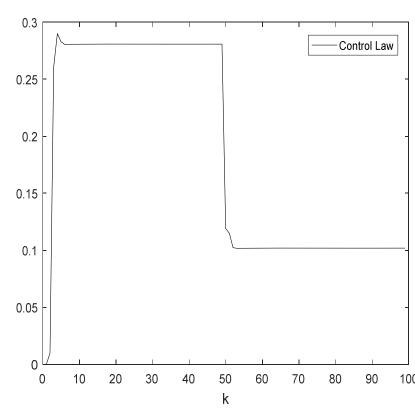

Figure 8: Control law signal

5.2 A multimodel internal neural network for system identification

In this section, we examine the effectiveness of the proposed multimodel internal model control system for the nonlinear system.

A performance index, based on the Mean Square Error (MSE), is used to evaluate the quality of the model.

$$MSE = \frac{1}{N}\sum_{k=1}^{N}\left(y_{MM}(k) - y(k)\right)^2 \qquad (29)$$

where N is the number of observations, $y_{MM}(k)$ is the output of the multi-model and $y(k)$ is the output of the system.

In this section, the parametric estimation based on the multimodel approach is presented. The used output of the multimodel approach is given as follows:

$$y_{MM}(k) = \sum_{i=1}^{L}\mu_i(\xi(k))y_i(k) \qquad (30)$$

where $y_i(k)$ is the output of the ith sub-neural network model which is given by

$$y_i(k) = \lambda s(\sum_{j=1}^{n_2} z_j s(\sum_{i=1}^{n_1} w_{ji}x_{1i})) \qquad (31)$$

$\mu_i\left(\xi(k)\right)$ is the weighting function is given by

$$\mu_i(\xi(k)) = \frac{w_i(\xi(k))}{\sum_{j=1}^{L} w_j(\xi(k))} \qquad (32)$$

$w_i(\xi(k))$ is the Gaussian function is given by

$$w_i(\xi(k)) = \exp\left(-\frac{\left(\xi(k) - c_i\right)^2}{\sigma_i^2}\right) \qquad (33)$$

Homogeneous partitioning of the four-zone system operating space is performed using the weighting functions centered on $c_1 = -1$, $c_2 = -0.33$, $c_3 = 0.33$ and $c_4 = 1$ and dispersion $\sigma = 0.4$ are given by figure 9 and the weighting functions are shown in figure 10.

The parametric estimation is based on an iterative minimization procedure of quadratic global criterion:

$$J = \frac{1}{2}(e(k))^2 = \frac{1}{2}(y(k) - y_{MM}(k))^2 \qquad (34)$$

We use 800 couples of input-output for identification. The input $u(k)$ of the system consists of the concatenation of amplitude slots (belonging to [-1,1]) and variable durations.

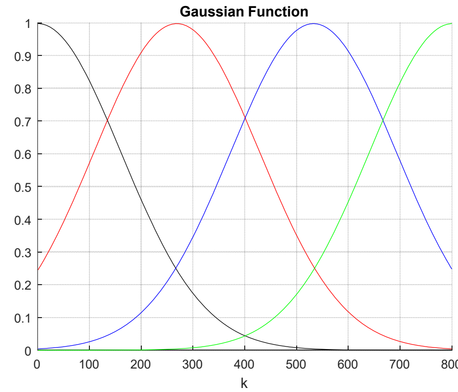

Figure 9: Gaussian functions

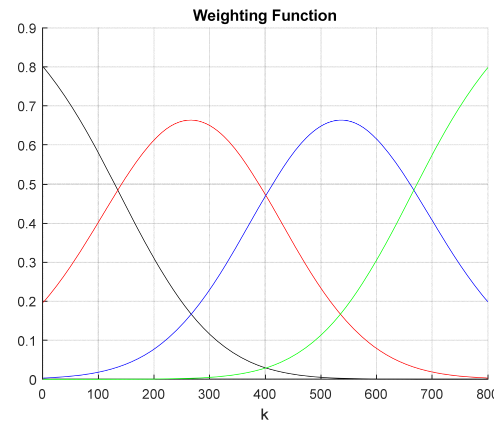

Figure 10: Weithing functions

The validation of the model is carried out considering the following input :

$$\begin{cases} u(k) = \sin\left(\dfrac{2\pi}{250}k\right) & 1 \le k \le 500 \\ u(k) = 0.8\sin\left(\dfrac{2\pi}{250}k\right) + 0.2\sin\left(\dfrac{2\pi}{25}k\right) & 1 > 500 \end{cases} \qquad (35)$$

In this section, we used the equation (23) to optimize each multilayered neural networks models $y_i(k)$, $i = 1,\ldots,4$, given by the equation (31). The found models are used in the equation of the proposed multimodel $y_{MM}(k)$, according to equation (30).

In figure 11, the nonlinear system responses and the optimized multilayered model by gradient descent method are illustrated. The good concordance between the system output and that of the multimodel is clearly highlighted. The performance index obtained is MSE=2.5842e-04. In figure 12, the error between the multimodel $y_{MM}(k)$ and $y(k)$ is illustrated.

Four multilayered neural networks models are used in the multimodel approach. The first neural networks multilayered model is described by 12 hidden sigmoid function, 18 neurons in the input layer with an input vector x_1, with :

$$x_1 = [y(k), y(k-1), y(k-1), u(k-1), u(k-2)]^T .$$

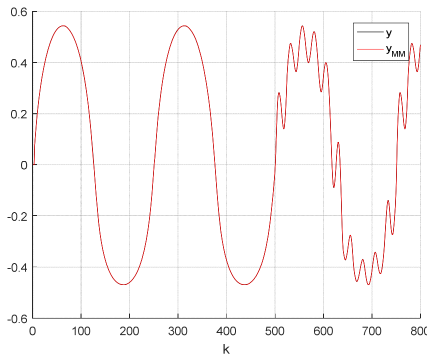

Figure 11: The multimodel output and the system output

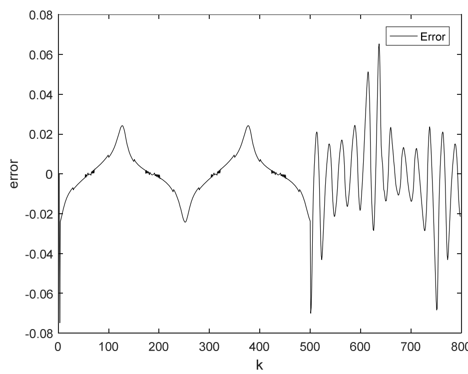

Figure 12: The error between the multimodel output and the system output

The second neural networks multilayered model is described by 21 sigmoid function, 14 neurons in the input layer with an input vector x_2, with

$$x_2 = [y(k), y(k-1), y(k-2), u(k)y(k-1), u(k-1)]^T$$

The third neural networks multilayered model is described by 19 sigmoid function, 18 neurons in the input layer with an input vector x_3, with :

$$x_3 = [y(k), y(k-1), y(k-2), u(k-1), u(k-2)]^T .$$

The fourth neural networks multilayered model is described by 22 sigmoid function, 10 neurons in the input layer with an input vector x_4, with :

$$x_4 = [y(k), y(k-1), y(k-2), u(k), u(k-1)u(k-2)]^T$$

These four artificial neural networks models are used as local-models of the non linear system.

5.3 Neural network controller using internal multimodel for system control

In this section, the resulting local neural network models is used now to emulate the dynamics of the considered nonlinear system.

Figures 13 and 14 present, respectively, the evolutions of the desired reference trajectory and the system output using the proposed multimodel internal model control based on neural model using a variable learning rate.

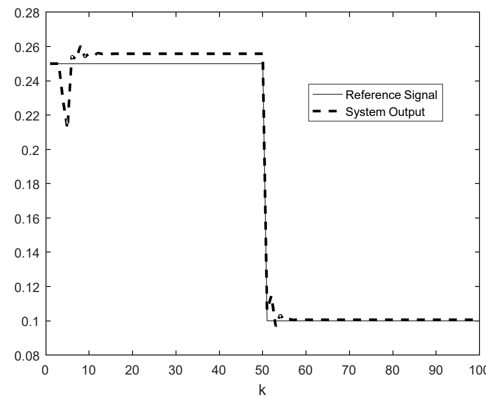

Figure 13: The evolution of the desired and the system output (Proposed Multimodel IMC)

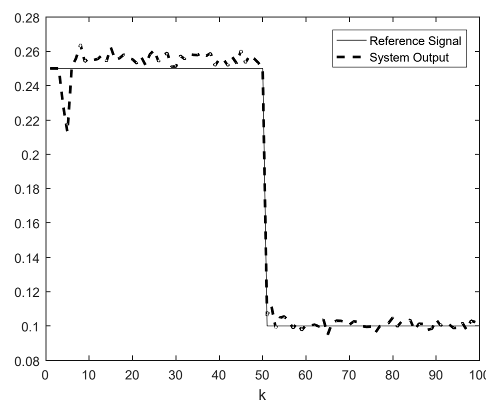

Figure 14: The evolution of the desired and the system output with disturbances (Proposed Multimodel IMC)

These figures show that the results recorded in the case of the strategy advanced in this paper is more better compared to the classical internal model control (figures 5 to 8).

Figures 15 and 16 present, respectively, the control signal obtained by classical and proposed approach.

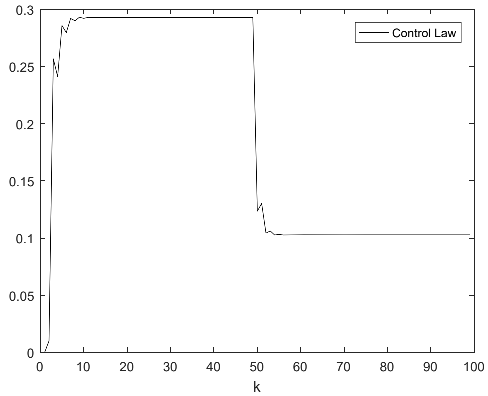

Figure 15: The evolution of the control law

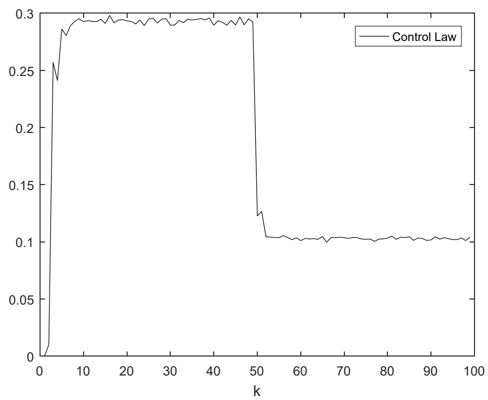

Figure 16: The control law the system in the case of disturbances

In the following table the performance of the proposed approach is summarized and compared to the classical IMC. From this table, it is noticed that the proposed approach gives the smallest controller error so and the best internal neural network controller.

Table 1: Summarized performance of the proposed approach..

Approach		Learning rate	NMSE
Classical IMC	Without disturbances	0.42	0.54
	With disturbances	0.38	0.63
Proposed multimodel IMC	Without disturbances	variable	0.42
	With disturbances	variable	0.49

6. CONCLUSIONS

In this paper, multimodel internal model control approach based on neural network is proposed. The neural network is used both to create the local model and the local controller. The used neural network is based on a variable learning rate. This avoids the error brought by the local modeling, enhances the robustness of the controller, reduces the difficulty of the building of the nonlinear model, decreases the problem of the design of the local-controllers, avoids the oscillation brought by model switching and realizes smooth-switch. Compared to the classical ICM method, the proposed multimodel IMC gives a smaller MSE and extensive simulation results demonstrate the validity of this method.

REFRENCES:

[1] Garcia C.E, Morari M., Internal model control. 1. a unifying review and some new results. Ind. Eng. Chem Process Des.&.Dev., Vol.21, No.2, pp. 308-323, 1982.

[2] Isabelle R., Nonlinear internal model control using neural networks application to processes with delay and design issues, IEEE Transaction on Neural Networks, Vol.11, No.1, pp. 80-90, 2000.

[3] Hunt K.J., Neural networks for nonlinear internal model control, IEE. Pro-D, Vol.138, No.5, pp. 431-438, 1991.

[4] Boukezzoula R., Galichet S., Foulloy L., Nonlinear internal model control: application of inverse model based fuzzy control, IEEE Trans. Fuzzy System, Vol.11, No.6, pp. 814-829, 2003.

[5] Xiaohong C., Multi model direct adaptive control based on RBFN and fuzzy decision. Control theory and application, Vol.16, No.1, pp. 100-107, 1999.

[6] Brown M.D., Lightbody G., Nonlinear internal model control using local model networks, IEE Proc-Control Theory and Applications, Vol.144, No.6, pp. 505-514, 1997.

[7] Launcelot C.D-G., Ryan Paolo C.M., Vianca May B.T, Simon Philippe A.Z., Macario O.C., Design and evaluation of a multi-model, multi-level artificial neural network for eczema skin lesion detection, 2015 3rd International Conference on Artificial Intelligence, Modelling and Simulation, pp. 42-47, 2015. © 2015 IEEE. DOI 10.1109/AIMS.2015.17.

[8] Bennasr H., M'Sahli F., Multimodel representation of complex nonlinear systems : a multifaceted approach for real-time application. Mathematical Problems in Engineering Volume 2018, Article ID 1829396, 15 pages https://doi.org/10.1155/2018/1829396.

[9] Aswin N.V., Vijaysai P., Ravindra D.G., Identification of complex nonlinear processes based on fuzzy decomposition of the steady state space. Journal of Process Control, Vol.13, Issue 6, pp. 473-488, 2003.

[10] Zhang S., Huang Y., Gao Z., Design of multi-model controller based on neural network sets, 2009 International Workshop on Intelligent Systems and Applications, Wuhan, 2009, pp. 1-4, doi: 10.1109/IWISA.2009.5072821.

[11] Lupu C., Petrescu C., Dimon C., Multi-model and inverse model control methods for nonlinear systems, The 4th International Federation of Automatic Control Conference on Management and Control of Production and Logistics, September 27-30, Sibiu-Romania, 2007.

[12] Chengzhong H., Chengshan Q., Zhaohua Y., Yanqing W., Tracking control for a class of uncertain nonlinear systems with time-varying delay using multi-model switching, Journal of Applied Science, Vol.26, No.2, pp. 209-215, 2008.

[13] Daoying P., Youxian S., A fuzzy-weighted control scheme for multi-model MIMO systems, ACTA Automatic Sinica, Vol.24, No.3, pp. 387-390, 1998.

[14] Zhi Y., An adaptive method for nonlinear and time delay systems, Information and Control, Vol. 30, No.2, pp. 143-148, 2001.

[15] Hanxiong L., An approximate internal model-based neural control for unknown nonlinear discrete processes. IEEE transactions on neural networks, Vol.17, No.3, pp. 659-670, 2006.

[16] Narendra K.S., Parthasarathy K., Identification and control of dynamical systems using neural networks, IEEE Transactions on Neural Networks, Vol.1, No.1, pp. 4-27, 1990.

[17] Slama, S., Errachdi, A., Benrejeb, M., Online MRAC method using neural networks based on variable learning rate for non-linear systems International Journal of Computer Applications in Technology, 2019, 60(3), pp. 215–224