

Multiple neural control and stabilization

Fathi Fourati¹

Received: 22 January 2016 / Accepted: 24 October 2016 / Published online: 10 November 2016
© The Author(s) 2016. This article is published with open access at Springerlink.com

Abstract In this paper, we present a multiple neural control and stabilization strategy for nonlinear and unstable systems. This control strategy method is efficient especially when the system presents different behaviors or different equilibrium points and when we hope to drive the whole process to a desired state ensuring stabilization. The considered control strategy has been applied on a nonlinear unstable system possessing two equilibrium points. It has been shown that the use of the multiple neural control and stabilization strategy increases further the stability domain of the system further than when we use a single neural control strategy.

Keywords Behaviors · State variables · Multiple neural controllers · Stabilization

1 Introduction

Multiple models characterizing different plant operation modes are used to predict the system behaviors. They are the one that best describes the plant and used to initialize new adaptation and/or generate new control input. From a practical point of view, the need to use multiple models in control is often necessary specially when sudden changes in the plant occur, in order to give better performance such as more accurate tracking and larger operation domain. In addition to the abilities of neural networks to imitate nonlinear plant characteristics, both multiple models and

neural networks tools have attracted researchers to investigate in the domain of control of complex and nonlinear system especially the field of multiple neural control strategies.

In the 1960s and 1970s, most of works on the multiple model control were based on optimal control. Specifically, problem solving was based on the use of Kalman filters and linear control minimizing a quadratic loss function [1].

In the context of identification, there is no new means used in the multiple model approach, but in the control context, the switching problem was raised and the first proposals were published by Martensson [2]. Following, two types of switching will begin to appear in the literature. The first is known as direct switching, where the choice of the next controller is predetermined and depends on the outputs of the system. The second is known as indirect switching, where local models are used at each moment in which controller will be used [3]. The latter kind of switch is also called supervised control.

There has been a major research activity to extend the multiple model approach in the control field. Narendra et al. [4] presented a general methodology to design a multiple model adaptive control of uncertain systems. This methodology makes systems to operate effectively in an environment with a high degree of uncertainty. As applications, they considered a system described by various behaviors; each behavior is represented by a model including the dynamic relating to the considered environment.

In their book, Murray-Smith and Johansen [5] have made a collection of a number of articles on multiple model approaches. This book considers the various aspects and applications of multiple model approach for modeling, identification and control of nonlinear systems, and it summarizes the theory and application of multiple model

✉ Fathi Fourati
Fethi.Fourati@ipeis.rnu.tn

¹ Control and Energy Management Laboratory (CEM-Lab),
University of Sfax, Sfax, Tunisia

adaptive control. In this book, the authors also open up horizons of research on the topic of adaptive multiple model control, such as the determination of the local models number of the complex system to be controlled, the choice of the types of controllers and the validity of the models (weighting functions).

Since 2000, research on multiple model approach has geared toward stability analysis and robust control of systems described by multiple models [6].

This paper is organized such that Sect. 2 presents a general description of a multiple neural control. Section 3 presents the structure of the multiple neural control and stabilization strategy. In Sect. 4, we show a single neural control stabilization method, and in Sect. 5, simulation results are carried out using the multiple neural control and stabilization on an unstable nonlinear system. Finally, a conclusion and prospects are given in Sect. 6.

2 Multiple neural control

The application of multiple neural control strategy is based on neural models, which incorporates a set of pair model/controller. Combination and switching between

models are the only characteristics of the multiple neural control [4, 5, 7, 8].

The general outline of this control strategy is shown in Fig. 1.

The weighting function $f_i(x)$ ($i \in \{1, \dots, n\}$) represents the validity of the model number i (and/or the corresponding controller). In the case when we select a single controller at a given instant (e.g., the i^{eme} controller), the value of the function $f_i(x)$ is equal to 1 and 0 for all others $f_j(x)$ ($j \in \{1, \dots, n\}$ and $j \neq i$). The value of the function $f_i(x)$ is belonging to the range $[0, 1]$ in the case when we combine all models and controllers.

The banks of neural models and neural controllers are made after learning steps from sub-databases representing different behaviors of the controlled system.

3 Multiple neural control and stabilization strategy

The multiple neural controller and stabilization strategy is used in order to increase the system's stabilization domain. It consists on the build of neural controllers of which the

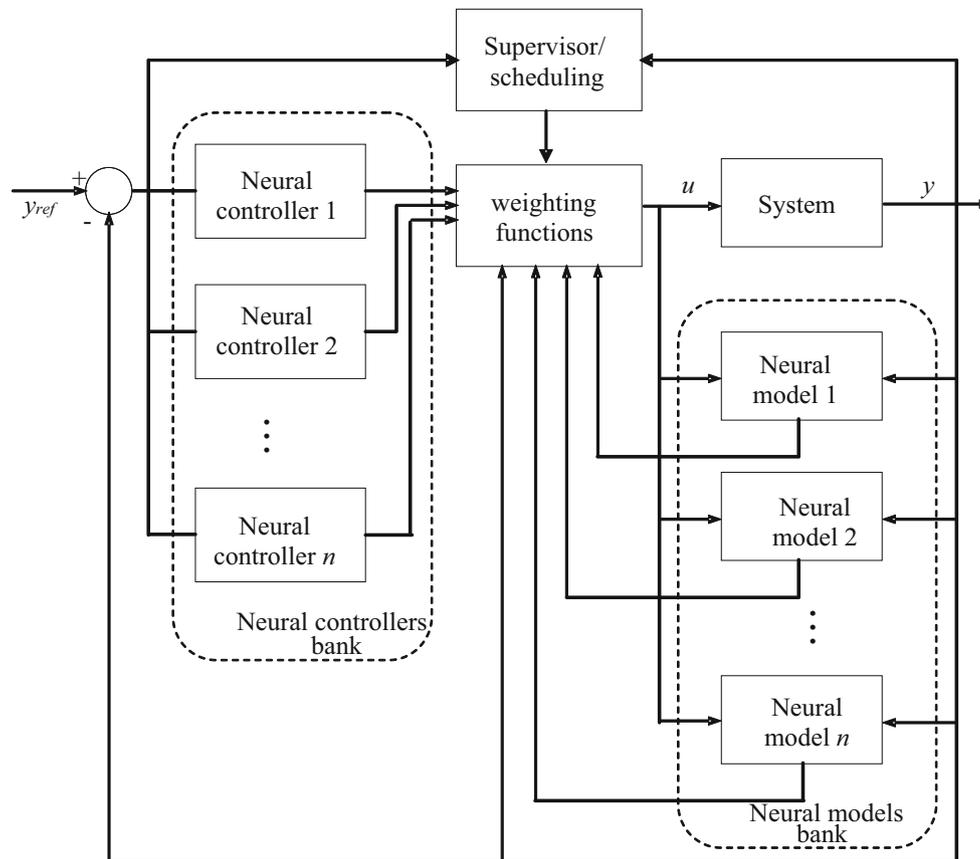


Fig. 1 Multiple neural control strategy

learning step is carried out through sub-databases representing different regions (behaviors) of the system.

3.1 Principle

The principle idea in this control strategy is to decompose the operating domain into local regions where the system presents different behaviors (compartments in the case of equilibrium points) in order to solve modeling problems, control and stabilization [6, 9]. An example of behaviors partition in an overall operating domain is shown in Fig. 2.

The multiple neural control and stabilization strategy is based on the accomplishment of neural controllers direct neural models related to each behavior. The achievement of the learning for controllers and direct models is made from sub-databases identified around the center of each behavior. Each region may be represented in the space of a

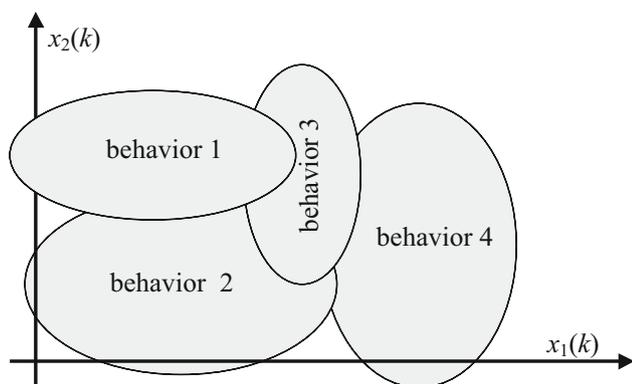


Fig. 2 Behaviours partition in operating domain

sphere or an ellipsoid whose center is the desired stabilization point.

3.2 Structure of the multiple control and stabilization

In our case, the centers of the operating modes (behaviors) are the equilibrium system points. The used controllers' selection method is a binary one, so at each simple time only one neural controller (NC) from the bank of controllers is active. The selection criterion is based on the computation of algebraic distance between the current states of the direct neural local model (DNLM) and the desired state. The chosen controller is the one in which the corresponding direct neural local model gives the minimal distance. The diagram of the control strategy is illustrated in Fig. 3.

The selection criterion C (1) is based on the computation of the minimum distance between the current and the desired states, so the selected neural controller i is such that:

$$C = \min(d_i) \quad i \in \{1, \dots, n\} \tag{1}$$

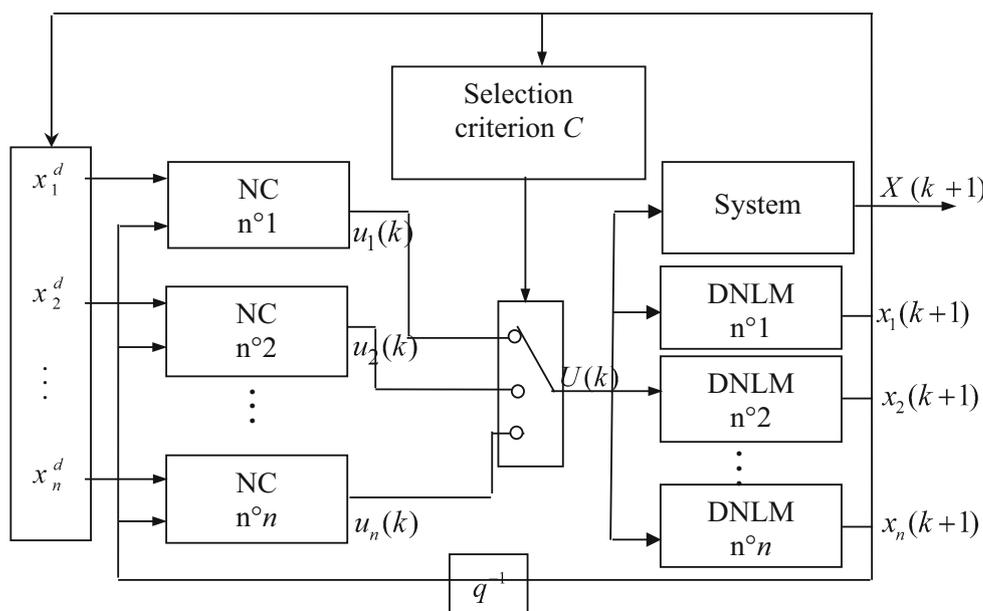
where d_i is the distance between the current state $x_i(k)$ of the direct neural local model i and the desired state x_j^d .

3.3 Controller principle

Consider a nonlinear system described by the following state-space model [9]:

$$x(k + 1) = f[x(k), u(k)] \tag{2}$$

Fig. 3 Multiple neural control and stabilization strategy



where $x(k) \in R^n$ is the vector of state variables at time k $u(k) \in R^m$ is the control vector and $f [\cdot]$ is the vector of nonlinear functions.

It is assumed that the state variables are accessible and measurable. We can write (2) such that:

$$x(k + 2) = f[x(k + 1), u(k + 1)] \tag{3}$$

which can be written:

$$x(k + 2) = f[f[x(k), u(k)], u(k + 1)] \tag{4}$$

This implies that the states at time $k + 2$ are determined from the states at time k and the control values between times k and $k + 2$.

By repeating this reasoning, we can write:

$$x(k + N) = f[\dots f[f[x(k), u(k)], u(k + 1)], \dots, u(k + N - 1)] \tag{5}$$

which can be rewritten in the following compact form:

$$x(k + N) = F[x(k), U(k)] \tag{6}$$

where

$$U(k)^T = [u(k), u(k + 1), \dots, u(k + N - 1)] \tag{7}$$

In conclusion, the states at time $k + N$ are determined by the state vector at time k and controls values between times k and $k + N - 1$.

If Eq. (6) is invertible [10], then $U(k)$ can be solved according to $x(k + N)$ and $x(k)$. In fact, the authors proved that in this case, the condition of invertibility is interpreted as a local condition freely achievable. In addition, in Ref. [11], it is shown that if the linearized model of the system is controllable and observable, then the local inverse model of the system exists.

$$U(k) = G[x(k), x(k + N)] \tag{8}$$

where G is a nonlinear function and Eq. (8) is a fundamental relationship representing the inverse dynamic of the system [10]. The nonlinear function G can be approximated by a neural network. This last will be exploited as a neural controller providing the control actions to stabilize the system around an equilibrium point or a desired state.

3.4 Structure of the neural controller

A large class of nonlinear dynamic systems is presented with the state-space models (2), so to realize neural control strategy it is necessary to build neural controllers based on input and states data.

Generally, a multilayer feed-forward neural network with one hidden layer is used to model the direct and inverse dynamic nonlinear systems [10, 12, 13]. The same structure of neural network will be used to generate the

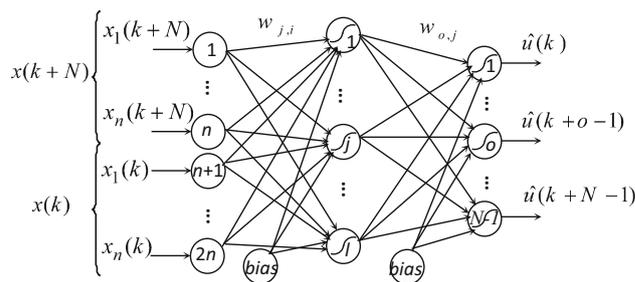


Fig. 4 Structure of the neural controller

system control law, given the current state and the future state (desired state) [14]. Figure 4 shows the structure of the used neural controller.

The connection weight $w_{j,i}$ network and $w_{o,j}$ are adjusted in order to minimize a quadratic error criterion (12) between network outputs $\hat{U}(k)$ and the desired outputs $U(k)$ [10–13].

The activation function used is sigmoidal one given by (13).

The neural controller output layer contains $M = N \cdot m$ nodes if the dimension of the control vector $u(k)$ is equal to m .

N is the necessary number of iterations to evolve the system from the actual state $x(k)$ to the future state $x(k + N)$.

This neural network structure provides a function $\hat{G}[x(k), x(k + N), W]$ that models the inverse dynamics of the system. It is trained to provide the control action law $\hat{U}(k)$ to the system.

$$\hat{U}(k) = \hat{G}[x(k), x(k + N), W] \tag{9}$$

W is the vector connection weights of the neural network.

The number of nodes in the input layer is defined according to the number of current and future states.

The number of nodes of the output layer is equal to the components number of the sequence of control actions applied to the system to reach the future state from the current state.

The number of nodes in the hidden layer is chosen after learning experiences. It is fixed by the structure which gives the lowest value of the error criterion (12).

3.5 Learning procedure

The learning procedure of the neural controller is referred to the inverse neural modeling of the system [10–15] and [16]. The process that accomplished this step is shown in Fig. 5.

Here, sequences of synthesized signal $U(k)$ are applied as inputs to the system. The corresponding states of these last are used as inputs to the neural network of which the outputs are compared with the training signal (the input of

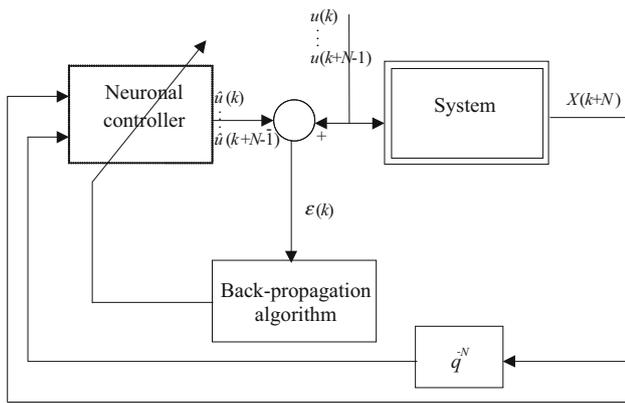


Fig. 5 Learning procedure of the neural controller

the system). The resulting error is used to adjust the neural network connection weights. This procedure tends to force the neural network to emulate the inverse dynamics of the system.

This learning structure is a classic method of supervised learning, where the teacher (i.e., the synthesis signal) provides directly target values to the output of the learner (i.e., the network model).

The used algorithm for learning is the back-propagation one ([17, 18]). For each input vector, the network calculates the output vector and adjusts the connections weights as described by Eqs. (16) and (20). The purpose of learning is to minimize the error $\varepsilon(k)$ obtained for each learning pattern.

$$\varepsilon(k) = \sum_{q=1}^N \sum_{o=1}^m [u_{od}(k - q + 1) - u_o(k - q + 1)] \quad (10)$$

3.6 Weights connection adaptation

The values of the input nodes are distributed to the hidden nodes through the weights connection w_{ji} . The input value of the j th node in the hidden layer is computed such that:

$$\text{net}_j^h = \sum_{i=1}^{2n} w_{ji} x_i + \theta_j^h \quad (11)$$

where w_{ji} is the weight connection between j th node and the i th node of the input layer and θ_j^h is the bias value of the node j .

The output of the node j is given by:

$$e_j = g(\text{net}_j^h) \quad (12)$$

where g is a sigmoidal function (13)

$$g(x) = \frac{1}{1 + e^{-x}} \quad (13)$$

The input value of the o th node in the output layer is computed such that

$$\text{net}_o^s = \sum_{j=1}^L w_{oj} e_j + \theta_o^s \quad (14)$$

the output is:

$$s_o = g(\text{net}_o^s) \quad (15)$$

The adaptation of the connection weights w_{oj} between the hidden layer and the output layer is performed as follows:

$$w_{oj}[t + 1] = w_{oj}[t] - \eta \frac{\partial \varepsilon}{\partial w_{oj}} \quad (16)$$

$$\frac{\partial \varepsilon}{\partial w_{oj}} = \frac{\partial \varepsilon}{\partial u_o(k)} \frac{\partial u_o(k)}{\partial w_{oj}} \quad (17)$$

$$\frac{\partial \varepsilon}{\partial u_o(k)} = -(u_{od}(k) - u_o(k)) \quad (18)$$

and

$$\frac{\partial u_o(k)}{\partial w_{oj}} = s'_o \cdot e_j = \frac{e^{\text{net}_o^s}}{(1 + e^{\text{net}_o^s})^2} \frac{1}{(1 + e^{-\text{net}_o^s})} \quad (19)$$

The adaptation of the connection weights w_{ji} between the input layer and the hidden layer is such that:

$$w_{ji}[t + 1] = w_{ji}[t] - \eta \frac{\partial \varepsilon}{\partial w_{ji}} \quad (20)$$

$$\frac{\partial \varepsilon}{\partial w_{ji}} = \frac{\partial \varepsilon}{\partial u_o(k)} \frac{\partial u_o(k)}{\partial w_{ji}} \quad (21)$$

and

$$\frac{\partial u_o(k)}{\partial w_{ji}} = \sum_{o=1}^{N \cdot m} s'_o \cdot w_{oj} \cdot e'_j \cdot x_i \quad (22)$$

where

$$s'_o(x) = \frac{\partial s_o(x)}{\partial x} = \frac{e^{\text{net}_o^s}}{(1 + e^{\text{net}_o^s})^2} \quad (23)$$

$$e'_j(x) = \frac{\partial e_j(x)}{\partial x} = \frac{e^{\text{net}_j^h}}{(1 + e^{\text{net}_j^h})^2} \quad (24)$$

where η is the learning rate.

4 Single neural stabilization strategy

After a learning step, the neural network emulating the inverse dynamic of the system can be operated in a closed-loop control providing the control law. In this case, the neural network is placed in a cascade with the system. Both of them (neural network and system) established a neuronal stabilizing feedback control states [14]. Figure 6 shows this control structure.

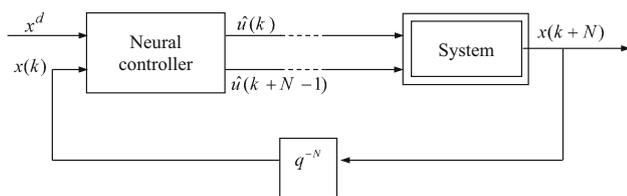


Fig. 6 Single neural control and stabilization strategy

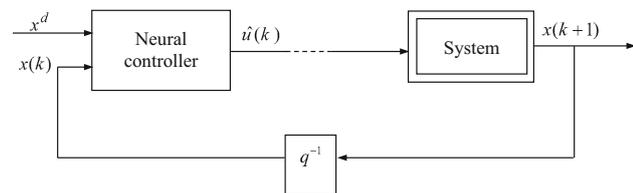


Fig. 7 Modified single neural control and stabilization strategy

The controller provides the system an N control values in each interval of N sampling periods.

The parameter N must be at least equal to n (order system) for local controllability [11].

This structure of stabilization and control will be used in the case of multiple neural control and stabilization.

In order to compute the value of the selection criterion (1) at each sample time in the case of multiple neural control and stabilization, only the first component of the input vector $\hat{u}(k)$ will be applied to the system. This is illustrated in Fig. 7.

5 Simulation results

In this section, we present the simulation results after the application of multiple neuronal and stabilization strategy for a nonlinear unstable system extract from Ref. [19].

5.1 Presentation of the system

The considered nonlinear system of the third order (three states) defined by the following equations:

$$\begin{cases} x_1(k + 1) = x_2(k) - x_3(k) + u^2(k) \\ x_2(k + 1) = 2x_1(k) - [1 + 0.5x_2(k)] \cdot u(k) \\ x_3(k + 1) = x_1(k) [x_2(k) - x_3(k)] + u(k) \end{cases} \quad (25)$$

This system presents two equilibrium points $x_{e1}^T = (0 \ 0 \ 0)$ and $x_{e2}^T = (1 \ 2 \ 1)$.

In order to apply the multiple neural control and stabilization strategy, we have made two sub-databases around each equilibrium point (behavior). These sub-databases will be used to learn neural controllers and direct neural local models useful to compute the selection criterion C and then to select the appropriate controller.

5.2 Sub-databases

The considered system (25) is highly unstable and divergent. The sub-databases are built by varying the system input at a random manner. The inputs should be bounded with low values, so that the system doesn't diverge from the first iterations. For both sub-databases, we adopt an input signal $u(k) \in [-0, 5; 0, 5]$ and different initial states $x(k)$ chosen near a considered equilibrium point. The future states values $x(k + N)$ are the computed, and all values from time k to time $k + N$ (in our case $N = 3$) are then recorded in text file establishing a sub-database which will be used to learn neural controller and direct neural local model.

For the first sub-database related to the first equilibrium point $x_{e1}^T = (0 \ 0 \ 0)$, the initial states $x(k) = (x_1(k) \ x_2(k) \ x_3(k))^T$ are chosen at random values belonging to the interval $[-1, 1]$ and the second sub-database on the second equilibrium point $x_{e2}^T = (1 \ 2 \ 1)$ where the initial states are chosen at random values belonging, respectively, to the intervals $[0, 2]$, $[1, 3]$, $[0, 2]$.

5.3 System response with multiple neural control and stabilization

In this part, we will apply the multiple neural controller and stabilization strategy in order to stabilize the system around its two equilibrium states which are defined as desired states. The controller's selection is based on the computation result of the criterion (1) (here $n = 2$), minimal distance between the current and the desired states.

$$x_1^d = x_{e1}^T = (0 \ 0 \ 0) \text{ and } x_2^d = x_{e2}^T = (1 \ 2 \ 1) \quad (26)$$

To show the benefits of the considered control and stabilization strategy, we have considered initial states in the neighborhood of the first equilibrium point and others near the second equilibrium point, and then, we have applied this strategy of control and stabilization. With a first initial state $x_i(1) = (-0.5 \ 0.3 \ 0.7)^T$, the states evolutions through time with the multiple neural control and stabilization strategy are shown in Fig. 8.

Here, the first controller is selected and the system stabilizes around the desired state (first equilibrium point) x_1^d . In this case, the distance between initial and desired states is $d_1 = 0.911$. With the second controller where the distance between initial and second desired states is $d_2 = 2.287$, the system diverges.

With the same initial state and using single neural control stabilization, the states diverge. Figure 9 shows the evolution of the states.

Note that with the multiple neural control strategy, the system is stabilized around the first equilibrium point

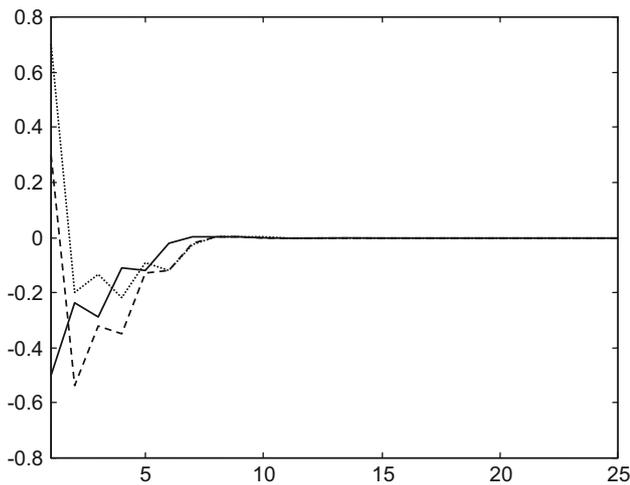


Fig. 8 Evolutions of the states in the case of the multiple neural control and stabilization strategy with the initial state $x_i(1)$

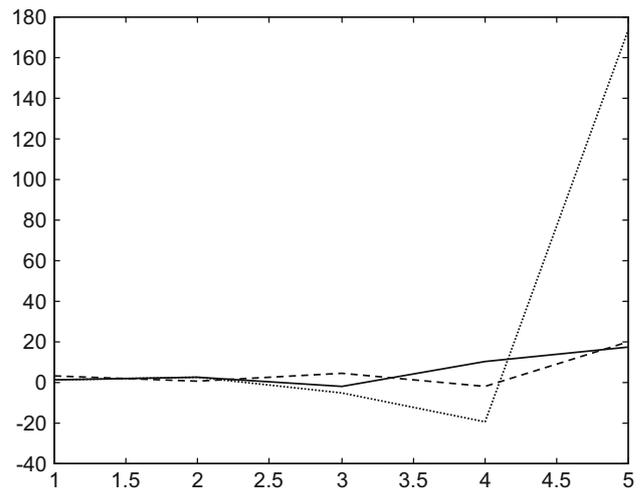


Fig. 10 Evolutions of the states using single neural control stabilization strategy with the initial state $x_i(2)$

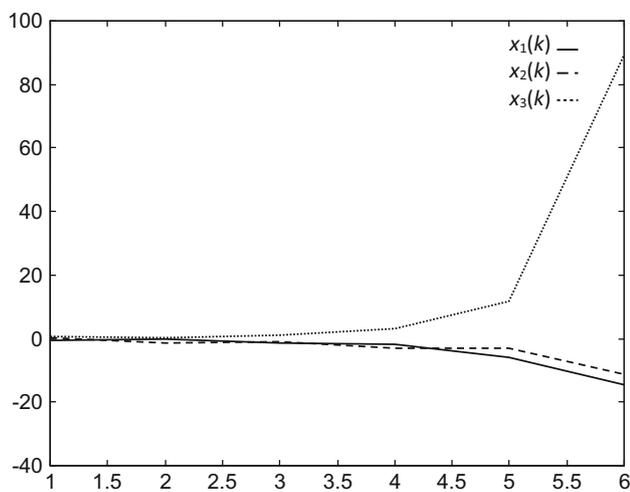


Fig. 9 Evolutions of the states using single neural control stabilization strategy with the initial state $x_i(1)$

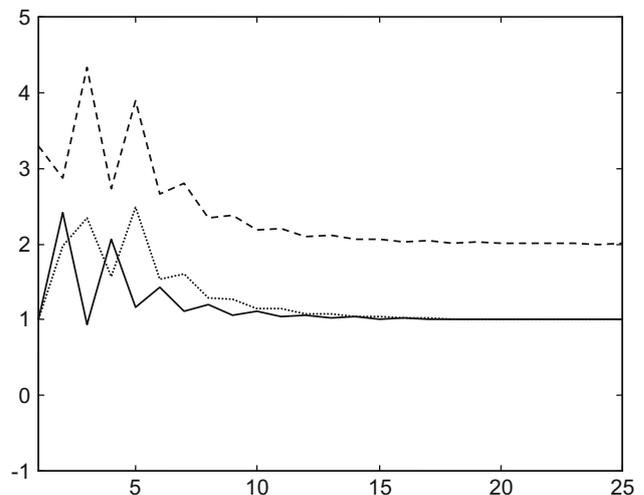


Fig. 11 Evolutions of the states in the case of the multiple neural control and stabilization strategy with the initial state $x_i(2)$

(desired state), whereas with a single neural control strategy the system diverges.

Then with a second initial state is $x_i(2) = (1 \ 3.3 \ 1)^T$, the evolutions of the states through time with a single neural control strategy are given in Fig. 10.

We remark that with the single control strategy the states diverge.

With the same initial state and using multiple neural control strategy, the states stabilize around the second equilibrium point. Figure 11 shows the evolution of the states.

Here, the selected controller is the second one, the distance between initial and desired states is $d_2 = 1.3$, and the system stabilizes around the desired state (second equilibrium point) x_2^d . While with the first controller where the

distance between initial and desired states is $d_1 = 3.59$, the system diverges.

These results confirm the efficient of the multiple neural control and stabilization strategy.

6 Conclusion

We have presented a multiple neuronal control and stabilization strategy which can be applied on systems characterized by different behaviors in different regions of the functional domain. In the simulation results, we have considered an unstable system possessing two equilibrium points. So two neural controllers and two direct neural local models have been built around each point; then, the multiple neural control and stabilization strategy has been

applied using a distance criterion between the current and the desired states to select the appropriate controller. The use of the latter strategy increases the system stabilization domain further. As future work, we hope to use an adaptive multiple neural control and stabilization strategy with weighting functions representing the validity of models.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Athans MD, Castanou D, Dunn KP, Greene CS, Lee WH, Sandell NR, Willsky AS (1977) The stochastic control of the F-8C aircraft using a multiple model adaptive control (MMAC) method—Part 1: equilibrium flight. *IEEE Trans Autom Control* 22(5):768–780
- Martensson B (1986) Adaptive stabilization. Ph.D. Dissertation. Lund Institute of Technology, Lund
- Middleton RH, Goodwin GC, Hill DJ, Mayne DQ (1988) Design issues in adaptive control. *IEEE Trans Autom Control* 33:50–58
- Narendra KS, Balakrishnan J, Ciliz MK (1995) Adaptation and learning using multiple models, switching, and tuning. *IEEE Control Syst Mag* 15:37–51
- Murray-Smith R, Johansen TA (1997) Multiple model approaches to modelling and control. Taylor & Francis, Oxford
- Kalkkuhl J, Hunt KJ, Zbikowski R, Dzieliński A (1997) Applications of neural adaptive control technology. Series in Robotics and intelligent systems, vol 17. World Scientific, Singapore
- Zhai JY, Shu MF, Xiao HM (2008) Multiple models switching control based on recurrent neural networks. *Neural Comput Appl* 17:365–371
- Fourati F (2014) Multiple neural control of a greenhouse. *Neurocomputing* 139:138–144
- Li X, Wang S, Wang W (2002) Improved multiple model adaptive control and its stability analysis, 15th Triennial World Congress. S.S.T-Mo-A03, A.C.3b, Barcelona
- Li W, Slotine JJE (1989) Neural network control of unknown nonlinear systems. In: American control conference, vol 2. Pittsburg, pp. 1136–1141
- Levin AU, Narendra KS (1996) Control of nonlinear dynamical systems using neural networks—Part II: observability, identification and control. *IEEE Trans Neural Netw* 7:30–42
- Psaltis D, Sideris A, Ymamura AA (1988) A multilayered neural network controller. *IEEE Control Syst Mag* 8:17–21
- Nguyen DH, Widrow B (1991) Neural networks for self-learning control systems. *IEEE Control Syst Mag* 10:18–23
- Fourati F, Chtourou M, Kamoun M (2007) Stabilization of unknown nonlinear systems using neural networks. *Appl Soft Comput* 8:1121–1130
- Hunt KJ, Sbarbaro D, Zbikowski R, Gawthrop PJ (1992) Neural networks for control systems—a survey. *Automatica* 28:1083–1112
- Xu C, Maoxin L, Qiming Z (2015) On the mean square exponential stability for a stochastic fuzzy cellular neural network with distributed delays and time-varying delays. *Int J Innovative Comput Inf Control* 11:247–256
- Rumelhart DE, Hinton GE, Williams RJ (1986) Learning internal representations by error propagation. In: Rumelhart DE, McClelland JL, PDP Research Group Corporate (eds) *Parallel distributed processing: explorations in the microstructure of cognition*, vol 1. MIT Press, Cambridge, pp 318–362
- Yabuta T, Yamada T (1992) Learning control using neural networks. *J Syst Eng* 2:180–191
- Levin AU, Narendra KS (1993) Control of nonlinear dynamical systems using neural networks: controllability and stabilization. *IEEE Trans Neural Netw* 4:192–206