Kybernetika

VOLUME *45* (2009), NUMBER 3

The Journal of the Czech Society for Cybernetics and Information Sciences

Published by:

Institute of Information Theory and Automation of the AS CR

Editorial Office:

Pod Vodárenskou věží 4, 182 08 Praha 8

Editor-in-Chief:

Milan Mareš

Managing Editors:

Lucie Fajfrová
Karel Sladký

Kybernetika is a bi-monthly international journal dedicated for rapid publication of high-quality, peer-reviewed research articles in fields covered by its title.

Kybernetika traditionally publishes research results in the fields of Control Sciences, Information Sciences, System Sciences, Statistical Decision Making, Applied Probability Theory, Random Processes, Fuzziness and Uncertainty Theories, Operations Research and Theoretical Computer Science, as well as in the topics closely related to the above fields.

The Journal has been monitored in the Science Citation Index since 1977 and it is abstracted/indexed in databases of Mathematical Reviews, Zentralblatt für Mathematik, Current Mathematical Publications, Current Contents ISI Engineering and Computing Technology.

Kybernetika. Volume *45* (2009) ISSN 0023-5954, MK ČR E 4902.

Published bimonthly by the Institute of Information Theory and Automation of the Academy of Sciences of the Czech Republic, Pod Vodárenskou věží 4, 182 08 Praha 8. — Address of the Editor: P. O. Box 18, 182 08 Prague 8, e-mail: kybernetika@utia.cas.cz. — Printed by PV Press, Pod vrstevnicí 5, 140 00 Prague 4. — Orders and subscriptions should be placed with: MYRIS TRADE Ltd., P. O. Box 2, V Štíhlách 1311, 142 01 Prague 4, Czech Republic, e-mail: myris@myris.cz. — Sole agent for all "western" countries: Kubon & Sagner, P. O. Box 34 01 08, D-8 000 München 34, F.R.G.

Published in June 2009.

KYBERNETIKA — VOLUME *45* (2009), NUMBER 3, PAGES 349 – 386

DIRECT ADAPTIVE CONTROL OF UNKNOWN NONLINEAR SYSTEMS USING A NEW NEURO–FUZZY METHOD TOGETHER WITH A NOVEL APPROACH OF PARAMETER HOPPING

Dimitris Theodoridis, Yiannis Boutalis and Manolis Christodoulou

The direct adaptive regulation for affine in the control nonlinear dynamical systems possessing unknown nonlinearities, is considered in this paper. The method is based on a new Neuro-Fuzzy Dynamical System definition, which uses the concept of Fuzzy Dynamical Systems (FDS) operating in conjunction with High Order Neural Network Functions (F-HONNFs). Since the plant is considered unknown, we first propose its approximation by a special form of a fuzzy dynamical system (FDS) and in the sequel the fuzzy rules are approximated by appropriate HONNFs. The fuzzy-recurrent high order neural networks (F-RHONN) are used as models of the unknown plant, practically transforming the original unknown system into a F-RHONN model which is of known structure, but contains a number of unknown constant value parameters. The proposed scheme does not require a-priori experts' information on the number and type of input variable membership functions making it less vulnerable to initial design assumptions, is extremely fast and, hence, can be applied in several difficult and very demanding real-time engineering applications. When the F-RHONN model matches the unknown plant, we provide a comprehensive and rigorous analysis of the stability properties of the closed loop system. Convergence of the state to zero plus boundedness of all other signals in the closed loop is guaranteed without the need of parameter (weights) convergence, which is assured only if a sufficiency-of-excitation condition is satisfied. The existence of the control signal is always assured by introducing a novel method of parameter hopping and incorporating it in weight updating law. Simulations illustrate the approximation superiority of the proposed scheme in comparison to other well established approaches. The applicability of the method is also tested on well known simulated nonlinear plants where it is shown that by following the proposed procedure one can obtain asymptotic regulation. Comparison is also made to simple RHONN controllers, showing that our approach is superior to the case of simple RHONN's.

Keywords: Neuro-Fuzzy systems, direct adaptive control, parameter hopping

AMS Subject Classification: 68T05, 93C40, 93C42, 93D05

1. INTRODUCTION

Nonlinear dynamical systems can be represented by general nonlinear dynamical equations of the form

$$\dot{x} = f(x, u). \tag{1}$$

The mathematical description of the system is required, so that we are able to control it. Unfortunately, the exact mathematical model of the plant, especially when this is highly nonlinear and complex, is rarely known and thus appropriate identification schemes have to be applied which will provide us with an approximate model of the plant.

It has been established that neural networks and fuzzy inference systems are universal approximators [12, 23, 52], i. e., they can approximate any nonlinear function to any prescribed accuracy provided that sufficient hidden neurons and training data or fuzzy rules are available. Recently, the combination of these two different technologies has given rise to fuzzy neural or neuro fuzzy approaches, that are intended to capture the advantages of both fuzzy logic and neural networks. Numerous works have shown the viability of this approach for system modeling ([4, 8, 20, 21, 29, 31, 36, 40, 54]). The neural and fuzzy approaches are most of the time equivalent, differing between each other mainly in the structure of the approximator chosen. Regarding the approximator structure, linear in the parameters approximators are used in [3, 31, 45] and nonlinear in [26, 38, 42, 50].

Adaptive control theory has been an active area of research over the past years ([14, 15, 16, 18, 19, 34, 35, 37, 47, 48, 49]). As concerning linear systems, there have been some researches on stability analysis of adaptive control systems, and adaptive control of plants [37, 47]. Also, many researchers focus on robust adaptive control that guarantees signal boundness in the presence of modelling errors and bounded disturbances ([14] – [16]). Regarding nonlinear systems, some adaptive control schemes via feedback linearization have been reported ([18, 19, 34, 48, 49]). The fundamental idea of feedback linearization is to transform a nonlinear system into a linear one. Then, linear control techniques are employed to acquire desired performance.

In the neuro or neuro fuzzy control two main approaches are followed. In the indirect adaptive control schemes [3, 8, 9, 19, 25, 26, 32, 38, 42, 50]) first the dynamics of the system are identified and then a control input is generated according to the certainty equivalence principle. In the direct adaptive control schemes ([2, 7, 10, 28, 39, 41, 43, 44, 53]) the controller is directly estimated and the control input is generated to guarantee stability without knowledge of the system dynamics. In [26] both approaches are presented, while in [13] a combined direct and indirect control scheme is used.

In a fuzzy or neuro-fuzzy scheme, the underlying fuzzy modeling scheme may be of Mamdani or TSK type and the identification phase usually consists of two categories: structure identification and parameter identification. Structure identification involves finding the main input variables out of all possible, specifying the membership functions, the partition of the input space and determining the number of fuzzy rules which is often based on a substantial amount of heuristic observation to express proper strategy's knowledge. Most of structure identification methods are based on data clustering, such as fuzzy C-means clustering [6], mountain clustering [21], and subtractive clustering [33]. These approaches require that all input-output data are ready before we start to identify the plant. So these structure identification approaches are off-line.

Recently [5, 24], high order neural network function approximators (HONNFs) have been proposed for the identification of nonlinear dynamical systems of the form (1), approximated by a Fuzzy Dynamical System. This approximation depends on the fact that fuzzy rules could be identified with the help of HONNFs. The same rationale has been employed in [51], where a neuro-fuzzy approach for the indirect control of unknown systems has been introduced.

In this paper HONNFs are also used for the neuro fuzzy direct control of nonlinear dynamical systems. In the proposed approach the underlying fuzzy model is of Mamdnani type. The structure identification is also made off-line based either on human expertise or on gathered data. However [56], the required a-priori information obtained by linguistic information or data is very limited. The only required information is an estimate of the centers of the output fuzzy membership functions. Information on the input variable membership functions and on the underlying fuzzy rules is not necessary because this is automatically estimated by the HONNFs. This way the proposed method is less vulnerable to initial design assumptions. The parameter identification is then easily addressed by HONNFs, based on the linguistic information regarding the structural identification of the output part and from the numerical data obtained from the actual system to be modeled.

We consider that the nonlinear system is affine in the control and could be approximated with the help of two independent fuzzy subsystems. Every fuzzy subsystem is approximated from a family of HONNFs, each one being related with a group of fuzzy rules. Weight updating laws are given and we prove that when the structural identification is appropriate then the error reaches zero very fast. Also, an appropriate state feedback control law is constructed to achieve asymptotic regulation of the output, while keeping bounded all signals in the closed loop. The existence of the control signal is always assured by introducing a method of parameter hopping and incorporating it in the weight updating law.

The paper is organized as follows. Section 2 presents notation and preliminaries related to the concept of fuzzy systems (FS) and the terminology used in the remaining paper. Moreover, it shows off the drawbacks of traditional fuzzy adaptive functional representations. Section 3 reports on the ability of HONNFs to act as fuzzy rule approximators. The direct neuro fuzzy regulation of affine in the control dynamical systems is presented in Section 4, where the method of parameter hopping is explained and the associated weight adaptation laws are given. Simulation results are given in Section 5 illustrating the approximation superiority of the proposed scheme in comparison to other well established approaches. The applicability of the method is also tested on well known simulated nonlinear plants where it is shown that by following the proposed procedure one can obtain asymptotic regulation. Comparisons are also presented showing that our approach is superior to the case of simple RHONN controllers.

2. NOTATION AND PRELIMINARIES

2.1. Notation

The following notations will extensively be used throughout the paper. $|\cdot|$ denotes the usual Euclidean norm of a vector. In case y is a scalar $|y|$ denotes its absolute value.

If A is a matrix, then $\|A\|$ denotes the Frobenious matrix norm defined as $\|A\|^2 = \sum_{ij} |a_{ij}|^2 = \mathrm{tr}\left\{A^T A\right\}$ where $\mathrm{tr}\{\cdot\}$ denotes the trace of a matrix.

An important property that will be used in deriving weight updating laws is the following: For a $m \times n$ matrix $\tilde{X}$ and matrices A, B, satisfying the following equality

$$\mathrm{tr}\left\{\dot{\tilde{X}}^T \tilde{X}\right\} = A\tilde{X}B$$

then

$$\dot{\tilde{X}}^T = BA \Rightarrow \dot{\tilde{X}} = A^T B^T. \tag{2}$$

2.2. Preliminaries

In this section we briefly present the notion of adaptive fuzzy systems and their conventional representation. We are also introducing the representation of fuzzy systems using the fuzzy rule indicator functions, which is used for the development of the proposed method.

2.2.1. Adaptive Fuzzy Systems

The performance, complexity, and adaptive law of an adaptive fuzzy system representation can be quite different depending upon the type of the fuzzy system (Mamdani or Takagi–Sugeno). It also depends upon whether the representations is linear or nonlinear in its adjustable parameters. Adaptive fuzzy controllers depend also on the type of the adaptive fuzzy subsystems they use. Suppose that the adaptive fuzzy system is intended to approximate the nonlinear function $f(x)$. In the mamdani type, linear in the parameters form, the following fuzzy logic representation is used [40, 52]:

$$f(x) = \sum_{l=1}^{M} \theta_l \xi_l(x) = \theta^T \xi(x) \tag{3}$$

where M is the number of fuzzy rules, $\theta = (\theta_1, \ldots, \theta_M)^T$, $\xi(x) = (\xi_1(x), \ldots, \xi_M(x))^T$ and $\xi_l(x)$ is the fuzzy basis function defined by

$$\xi_l(x) = \frac{\prod_{i=1}^{n} \mu_{F_i^l}(x_i)}{\sum_{l=1}^{M} \prod_{i=1}^{n} \mu_{F_i^l}(x_i)} \tag{4}$$

θ_l are adjustable parameters, and $\mu_{F_i^l}$ are given membership functions of the input variables (can be Gaussian, triangular, or any other type of membership functions).

In Tagaki–Sugeno formulation $f(x)$ is given by

$$f(x) = \sum_{l=1}^{M} g_l(x)\xi_l(x) \tag{5}$$

where $g_l(x) = a_{l,0} + a_{l,1}x_1 + \ldots + a_{l,n}x_n$, with $x_i, i = 1 \ldots n$ being the elements of vector x and $\xi_l(x)$ being defined in (40). According to [40], (5) can also be written in the linear to the parameters form, where the adjustable parameters are all $a_{l,i}, l = 1 \ldots M, i = 1 \ldots n$.

From the above definitions it is apparent in both, Mamdani and Tagaki–Sugeno forms that the success of the adaptive fuzzy system representations in approximating the nonlinear function $f(x)$ depends on the careful selection of the fuzzy partitions of input and output variables. Also, the selected type of the membership functions and the proper number of fuzzy rules contribute to the success of the adaptive fuzzy system. This way, any adaptive fuzzy or neuro-fuzzy approach, following a linear in the adjustable parameters formulation becomes vulnerable to initial design assumptions related to the fuzzy partitions and the membership functions chosen. In this paper this drawback is largely overcome by using the concept of rule indicator functions, which are in the sequel approximated by High order Neural Network function approximators (HONNFs). This way there is not any need for initial design assumptions related to the membership values and the fuzzy partitions of the if part.

2.2.2. Fuzzy system description using rule indicator functions

In this paper, we are briefly introducing the representation of fuzzy systems using the fuzzy rule indicator functions, which is used for the development of the proposed method.

Let us consider the system with input space $u \subset \mathbb{R}^m$ and state-space $x \subset \mathbb{R}^n$, with its i/o relation being governed by the following equation

$$z(k) = f(x(k), u(k)) \tag{6}$$

where $f(\cdot)$ is a continuous function and the k denotes the temporal variable. In case the system is dynamic the above equation could be replaced by the following difference equation

$$x(k+1) = f(x(k), u(k)) \tag{7}$$

where $k = 1, 2, \ldots$.

By setting $y = [x, u]$ and omitting k, Eq. (6) may be rewritten as follows

$$z = f(y). \tag{8}$$

In many practical situations, we are unable to measure accurately the states and inputs of a system of the form in (6); in most cases, we are provided with cheap sensors, expert's opinions, e.t.c which provide us with imprecise estimations of the state and input vectors. Thus, instead of vectors x and u we are provided with some linguistic variables $\tilde{x}_i$ and $\tilde{u}_i$, respectively.

Let now $\tilde{y} := (\tilde{x}, \tilde{u})$ and suppose that each linguistic variable $\tilde{y}_i$ belongs to a finite set L_i with cardinality k_i, i. e. $\tilde{y}_i$ takes one of k_i variables. Let also $\tilde{y}_{ij}$ denotes the jth element of the set L_i. Then we may define a function $\tilde{h}_i : \mathbb{R} \to L_i$ to be the output function of the system in Eq. (8) in the sense that

$$\tilde{y}_i = \tilde{h}_i(y_i). \tag{9}$$

Note that $\tilde{h}_i(\cdot)$ maps the real axis into a set of linguistic variables L_i, and thus $\tilde{h}_i(\cdot)$ is not defined in the usual way. In order to overcome such a problem we define the function $\tilde{h}_i : \mathbb{R} \to \{1, 2, \ldots, k_i\}$ as follows

$$\tilde{h}_i(y_i) = \tilde{y}_{ij} \Longleftrightarrow h_i(y_i) = j. \tag{10}$$

Since $h_i(\cdot)$ is very similar to $\tilde{h}_i(\cdot)$, we will call the function $h_i(\cdot)$ the *ith* output of the system in Eq. (8). Also, $\tilde{h}_i(\cdot)$ and consequently $h_i(\cdot)$ is related with the structural identification part mentioned in the introduction and arrive after using an automatic procedure based on system operation data or after consulting human experts advising on how to partition the system variables.

Following the standard approach in fuzzy systems theory we associate with each $\tilde{y}_{ij}$ a membership function $\tilde{\mu}_{ij}(y_i) \in [0, 1]$ which satisfies

$$\tilde{\mu}_{ij}(y_i) = \max_l \tilde{\mu}_{il}(y_i) \Longleftrightarrow h_i(y_i) = j. \tag{11}$$

From the definition of the functions $\tilde{h}_i(\cdot)$ [or $h_i(\cdot)$] we have that the space $\mathcal{Y} = \mathcal{X} \times \mathcal{U}$ is partitioned in the following way: let $\mathcal{Y}_{ij}$ be defined as follows

$$\mathcal{Y}_{ij} = \{y_i \in \mathbb{R} : h_i(y_i) = j\} \tag{12}$$

i. e. $\mathcal{Y}_{ij}$ denotes the set of all the variables y_i that output the same linguistic variable $\tilde{y}_{ij}$. Thus y is partitioned into disjoint subsets $\mathcal{Y}_{j_1, j_2, \ldots, j_{n+m}}$ defined as follows

$$\mathcal{Y}_{j_1, j_2, \ldots, j_{n+m}} := \mathcal{Y}_{1j_1} \times \cdots \times \mathcal{Y}_{(n+m)j_{n+m}}, \quad j_i \in \{1, 2, \ldots, k_i\}. \tag{13}$$

In a similar way we may define the sets $\mathcal{X}_{ij}$, $\mathcal{U}_{ij}$, $\mathcal{Z}_{ij}$ and the sets $\mathcal{X}_{j_1, j_2, \ldots, j_n}$, $\mathcal{U}_{j_1, j_2, \ldots, j_n}$ and $\mathcal{Z}_{j_1, j_2, \ldots, j_n}$. Note now the following fact: for two vectors $(x^{(1)}, u^{(1)}) \in \mathcal{Y}_{j_1, j_2, \ldots, j_{n+m}}$ and $(x^{(2)}, u^{(2)}) \in \mathcal{Y}_{j_1, j_2, \ldots, j_{n+m}}$ there maybe

$$h_i(f_i(x^{(1)}, u^{(1)})) \neq h_i(f_i(x^{(2)}, u^{(2)})) \tag{14}$$

for some $i \in \{1, 2, \ldots, n\}$, i. e. two input vectors belonging to the same subset $\mathcal{Y}_{j_1, j_2, \ldots, j_{n+m}}$ may point – through the vector – field $f(\cdot)$, to different subsets $\mathcal{Z}_{l_1, l_2, \ldots, l_n}$. Let now $\Omega^{l_1, l_2, \ldots, l_n}_{j_1, j_2, \ldots, j_{n+m}}$ be defined as the subset of $y_{j_1, j_2, \ldots, j_{n+m}}$ that points – through the vector – field $f(\cdot)$, to the subsets $z_{l_1, l_2, \ldots, l_n}$, i. e.

$$\Omega^{l_1, l_2, \ldots, l_n}_{j_1, j_2, \ldots, j_{n+m}} := \{(x, u) \in \mathcal{Y}_{j_1, j_2, \ldots, j_{n+m}} : h_1(z_1) = l_1, \ldots, h_n(z_n) = l_n\}$$

and define the transition possibilities $\pi^{l_1,\dots,l_n}_{j_1,\dots,j_{n+m}}$ as follows

$$\pi^{l_1,\dots,l_n}_{j_1,\dots,j_{n+m}} := \frac{\int_{(x,u)\in\Omega^{l_1,\dots,l_n}_{j_1,\dots,j_{n+m}}} \mathrm{d}X\mathrm{d}U}{\int_{(x,u)\in y_{j_1,\dots,j_{n+m}}} \mathrm{d}X\mathrm{d}U} \tag{15}$$

where $\pi^{l_1,\dots,l_n}_{j_1,\dots,j_{n+m}}$ is a number belonging to a set [0,1] that represents the fraction of the vectors (x,u) in $y_{j_1,\dots,j_{n+m}}$ that points – through the vector field $f(\cdot)$ to the set $x_{l_1,\dots,l_n}$. Obviously

$$\sum_{l_1,\dots,l_n} \pi^{l_1,\dots,l_n}_{j_1,\dots,j_{n+m}} = 1. \tag{16}$$

In order to present the lemma of Section 3, we define the indicator function: Let $I^{l_1,l_2,\dots,l_n}_{j_1,j_2,\dots,j_{n+m}}$ denote the indicator function of the subset $\Omega^{l_1,l_2,\dots,l_n}_{j_1,j_2,\dots,j_{n+m}}$, that is,

$$I^{l_1,\dots,l_n}_{j_1,\dots,j_{n+m}}(x,u) = \begin{cases} \alpha & \text{if } (x,u)\in\Omega^{l_1,\dots,l_n}_{j_1,\dots,j_{n+m}} \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

where α denotes the firing strength of the rule.

Using the above definitions, we can see that the system in Eq. (8) is described by fuzzy rules of the form

$$R^{l_1,\dots,l_n}_{j_1,\dots,j_{n+m}} \Longleftrightarrow \begin{array}{l} \text{IF } y_1 \text{ is } \tilde{y}_{1j_1} \text{ AND}\dots \\ \text{AND } y_{n+m} \text{ is } \tilde{y}_{(n+m)j_{n+m}} \\ \text{THEN} \\ z_1 \text{ is } \tilde{z}_{1l_1} \text{ AND}\cdots\text{AND } z_n \text{ is } \tilde{z}_{nl_n} \\ \text{with possibility } \pi^{l_1,\dots,l_n}_{j_1,\dots,j_{n+m}} \end{array} \tag{18}$$

where obviously $\tilde{y}_{ij_i} = \tilde{h}_i(y_i(k))$ and $\tilde{z}_{il_i} = \tilde{h}_i(z_i) = \tilde{h}_i(f_i(x,u))$.

In the above notation, if $j_1 = l_1$ and $j_2 = l_2$ and ... and $j_n = l_n$, then these points participate to the definition of the same fuzzy rule. If $j_1 \neq l_1$ or $j_2 \neq l_2$ or or $j_n \neq l_n$, then these points define alternative fuzzy rules describing this transition. Consider now the next definition.

Definition 2.1. (FS) A Fuzzy System – (FS) is a set of Fuzzy Rules of the form $(R^{l_1,l_2,\dots,l_n}_{j_1,j_2,\dots,j_{n+m}})$; the system in Eq. (6) is called the Underlying System – (US) of the previously defined FS. Alternatively, the system in Eq. (8) will be called a Generator of the FS that is described by the rules $(R^{l_1,l_2,\dots,l_n}_{j_1,j_2,\dots,j_{n+m}})$.

Due to the linguistic description of the variables of the FS it is not rare to have more than one systems of the form in Eq. (8) to be generators for the FS that is described by the rules $(R^{l_1,l_2,\dots,l_n}_{j_1,j_2,\dots,j_{n+m}})$.

Define now the following system

$$z = \sum \bar{z}^{l_1,\dots,l_n}_{j_1,\dots,j_{n+m}} \times I^{l_1,\dots,l_n}_{j_1,\dots,j_{n+m}}(x,u). \tag{19}$$

Where $\bar{z}^{l_1,\dots,l_n}_{j_1,\dots,j_{n+m}} \in \mathbb{R}^n$ be any constant vector containing the centers of each fuzzy variable z_i and satisfy $h_i(\bar{z}^{l_1,\dots,l_n}_{j_1,\dots,j_{n+m}}(i)) = l_i$ where $\bar{z}^{l_1,\dots,l_n}_{j_1,\dots,j_{n+m}}(i)$ denotes the ith entry of $\bar{z}^{l_1,\dots,l_n}_{j_1,\dots,j_{n+m}}$ Then, according to [5, 24] the system in (19) is a generator for the FS $(R^{l_1,l_2,\dots,l_n}_{j_1,j_2,\dots,j_{n+m}})$.

It is obvious that Eq. (19) can be also valid for dynamic systems. In its dynamical form it becomes

$$x(k+1) = \sum \bar{x}^{l_1,\dots,l_n}_{j_1,\dots,j_{n+m}} \times I^{l_1,\dots,l_n}_{j_1,\dots,j_{n+m}}(x(k), u(k)) \tag{20}$$

Where $\bar{x}^{l_1,\dots,l_n}_{j_1,\dots,j_{n+m}} \in \mathbb{R}^n$ be any vector satisfying $h_i(\bar{x}^{l_1,\dots,l_n}_{j_1,\dots,j_{n+m}}(i)) = l_i$ where $\bar{x}^{l_1,\dots,l_n}_{j_1,\dots,j_{n+m}}(i)$ denotes the ith entry of $\bar{x}^{l_1,\dots,l_n}_{j_1,\dots,j_{n+m}}$

3. THE HONNF'S AS FUZZY RULE APPROXIMATORS

The main idea in presenting the main result of this section lies on the fact that functions of high order neurons are capable of approximating discontinuous functions; thus, we use high order neural network functions in order to approximate the indicator functions $I^{l_1,\dots,l_n}_{j_1,\dots,j_{n+m}}$. However, in order the approximation problem to make sense the space $\mathcal{Y} := \mathcal{X} \times \mathcal{U}$ must be compact. Thus, our first assumption is the following:

(**A.1**) $\mathcal{Y} := \mathcal{X} \times \mathcal{U}$ is a compact set.

Notice that since $\mathcal{Y} \subset \mathbb{R}^{n+m}$ the above assumption is identical to the assumption that it is closed and bounded. Also, it is noted that even if $\mathcal{Y}$ is not compact we may assume that there is a time instant T such that $(x(k), u(k))$ remain in a compact subset of $\mathcal{Y}$ for all $k < T$; i.e. if $\mathcal{Y}_T := \{(x(k), u(k)) \in \mathcal{Y}, k < T\}$ We may replace assumption (A.1) by the following assumption

(**A.2**) $\mathcal{Y}_T$ is a compact set.

It is worth noticing, that while assumption (A.1) requires the system in Eq. (7) solutions to be bounded for all $u(k) \in U$ and $x(0) \in X$, assumption (A.2) requires the system in Eq. (7) solutions to be bounded for a finite time period; thus, assumption (A.1) requires the system in Eq. (7) to be bounded input bounded state (BIBS) stable while assumption (A.2) is valid for systems that are not BIBS stable and, even more, for unstable systems and systems with finite escape times.

We are now ready to show that high order neural network functions are capable of approximating the indicator functions $I^{l_1,\dots,l_n}_{j_1,\dots,j_{n+m}}$. Let us define the following high order neural network functions (HONNFs).

$$N(x, u; w, L) = \sum_{k=1}^{L} w_k \prod_{j \in I_k} \Phi_j^{d_j(k)}. \tag{21}$$

Where $\{I_1, I_2, \ldots, I_L\}$ is a collection of L not-ordered subsets of $\{1, 2, \ldots, m+n\}$, $d_j(k)$ are non-negative integers, Φ_j are sigmoid functions of the state or the input, which are the elements of the following vector

$$\Phi = \begin{bmatrix} \Phi_1 \\ \vdots \\ \Phi_n \\ \Phi_{n+1} \\ \vdots \\ \Phi_{m+n} \end{bmatrix} = \begin{bmatrix} S(x_1) \\ \vdots \\ S(x_n) \\ S(u_1) \\ \vdots \\ S(u_m) \end{bmatrix} \tag{22}$$

where

$$S(u) \quad \text{or} \quad S(x) = a\frac{1}{1+e^{-\beta x}} - \gamma \tag{23}$$

and $w := [w_1 \cdots w_L]^T$ are the HONNF weights. Eq. (21) can also be written

$$N(x, u; w, L) = \sum_{k=1}^{L} w_k s_k(x, u). \tag{24}$$

Where $s_k(x, u)$ are high order terms of sigmoid functions of the state and/or input.

The next lemma [24] states that a HONNF of the form in Eq. (24) can approximate the indicator function $I^{l_1,\ldots,l_n}_{j_1,\ldots,j_{n+m}}$.

Lemma 3.1. Consider the indicator function $I^{l_1,\ldots,l_n}_{j_1,\ldots,j_{n+m}}$ and the family of the HONNFs $N(x, u; w, L)$. Then for any $\epsilon > 0$ there is a vector of weights $w^{j_1,\ldots,j_{n+m};l_1,\ldots,l_n}$ and a number of $L^{j_1,\ldots,j_{n+m};l_1,\ldots,l_n}$ high order connections such that

$$\sup_{(x,u)\in\bar{y}} \{I^{l_1,\ldots,l_n}_{j_1,\ldots,j_{n+m}}(x, u) - N(x, u; w^{j_1,\ldots,j_{n+m};l_1,\ldots,l_n}, L^{j_1,\ldots,j_{n+m};l_1,\ldots,l_n})\} < \varepsilon$$

where $\bar{y} \equiv y$ if assumption (A.1) is valid and $\bar{y}_T \equiv y$ if assumption (A.2) is valid.

Let us now keep $L^{j_1,\ldots,j_{n+m};l_1,\ldots,l_n}$ constant, i. e. let us preselect the number of high order connections, and let us define the optimal weights of the HONNF with $L^{j_1,\ldots,j_{n+m};l_1,\ldots,l_n}$ high order connections as follows

$$\begin{aligned} \bar{w}^{j_1,\ldots,j_{n+m};l_1,\ldots,l_n} &:= \arg \min_{w\in\mathbb{R}^{j_1,\ldots,j_{n+m};l_1,\ldots,l_n}} \\ &\times \left\{ \sup_{(x,u)\in\bar{y}} \left| I^{l_1,\ldots,l_n}_{j_1,\ldots,j_{n+m}}(x, u) - N(x, u; w, L^{j_1,\ldots,j_{n+m};l_1,\ldots,l_n}) \right| \right\} \end{aligned}$$

and the modelling error as follows

$$\nu^{l_1,\ldots,l_n}_{j_1,\ldots,j_{n+m}}(x, u) = I^{l_1,\ldots,l_n}_{j_1,\ldots,j_{n+m}}(x, u) - N(x, u; w^{j_1,\ldots,j_{n+m};l_1,\ldots,l_n}, L^{j_1,\ldots,j_{n+m};l_1,\ldots,l_n})$$

It is worth noticing that from Lemma 3.1, we have that $\sup_{(x,u)\in\bar{y}}\left|\nu^{l_1,\ldots,l_n}_{j_1,\ldots,j_{n+m}}(x,u)\right|$ can be made arbitrarily small by simply selecting appropriately the number of high order connections.

Using the approximation Lemma 3.1, it is natural to approximate system in Eq. (20) by the following dynamical system

$$z(k+1) = \sum \bar{x}^{l_1,\ldots,l_n}_{j_1,\ldots,j_{n+m}}(x,u) \times N(z(k),u(k);w^{j_1,\ldots,j_{n+m};l_1,\ldots,l_n},L^{j_1,\ldots,j_{n+m};l_1,\ldots,l_n}).$$

Let now $x(k)[x(0),u(k)]$ denote the solution in Eq. (20) given that the initial state at $t=0$ is equal to $\chi(0)$ and the input is $u(k)$. Similarly we define $z(k)[z(0),u(k)]$. Also let

$$\nu(z(k),u(k)) = \sum (\bar{x}^{l_1,\ldots,l_n}_{j_1,\ldots,j_{n+m}}(x,u) \times \nu^{l_1,\ldots,l_n}_{j_1,\ldots,j_{n+m}}(z(k),u(k))). \tag{25}$$

Then, it can be easily shown that

$$z(k)[z(0),u(k)] = x(k)[z(0),u(k)] + \nu[z(k),u(k)]. \tag{26}$$

Note now that from the approximation Lemma 3.1 and the definition of $\nu(z(k),u(k))$ we have that modeling error can be made arbitrarily small provided that $(z(k),u(k))$ remain in a compact set (e. g. $\bar{y}$).

Theorem 3.2. (Kosmatopoulos and Christodoulou [24], Christodoulou et al. [5]) Consider the FDS $(R^{l_1,\ldots,l_n}_{j_1,\ldots,j_{n+m}})$ and suppose that system in Eq. (7) is its underlying system. Assume that either assumptions (**A.1**) or (**A.2**) hold. Also consider the HONNF in [5]. Then, for any $\varepsilon>0$ there exists a matrix Θ^* and a number L^* high order connections and $\Theta=\Theta^*$ is a generator for the FDS described by the rules

$$R^{l_1,\ldots,l_n}_{j_1,\ldots,j_{n+m}} \Longleftrightarrow \begin{array}{l} \text{IF } y_1 \text{ is } \tilde{y}_{1j_1} \text{ AND } \cdots \\ \text{AND } y_{n+m} \text{ is } \tilde{y}_{(n+m)j_{n+m}} \\ \text{THEN} \\ \chi_1 \text{ is } \tilde{y}_{1l_1} \text{ AND } \cdots \text{AND } \chi_n \text{ is } \tilde{y}_{nl_n} \\ \text{with possibility } \widehat{\pi}^{l_1,\ldots,l_n}_{j_1,\ldots,j_{n+m}} \end{array}$$

where

$$\max\left|\pi^{l_1,\ldots,l_n}_{j_1,\ldots,j_{n+m}} - \widehat{\pi}^{l_1,\ldots,l_n}_{j_1,\ldots,j_{n+m}}\right| < \varepsilon.$$

4. DIRECT ADAPTIVE NEURO–FUZZY CONTROL

4.1. Problem formulation and neuro-fuzzy representation

4.1.1. Problem formulation

We consider affine in the control, nonlinear dynamical systems of the form

$$\dot{x} = f(x) + G(x)\cdot u$$

$$y = x \tag{27}$$

where the state $x \in \mathbb{R}^n$ is assumed to be completely measured, the control u is in $\mathbb{R}^n$, f is an unknown smooth vector field called the drift term and G is a matrix with columns the unknown smooth controlled vector fields g_i, $i = 1, 2, \ldots, n$ and $G = [g_1, g_2, \ldots, g_n]$. The above class of continuous-time nonlinear systems are called affine, because in (27) the control input appears linear with respect to G. The main reason for considering this class of nonlinear systems is that most of the systems encountered in engineering, are by nature or design, affine. Furthermore, we note that non affine systems of the form given in (1) can be converted into affine, by passing the input through integrators, a procedure known as dynamic extension.

The state regulation problem is known as our attempt to force the state to zero from an arbitrary initial value by applying appropriate feedback control to the plant input. However, the problem as it is stated above for the system (27), is very difficult or even impossible to be solved since the vector fields f, g_i, $i = 1, 2, \ldots, n$, are assumed to be completely unknown. To overcome this problem we assume that the unknown plant can be modeled by the following neuro-fuzzy model, where the weight values W^* and W_1^* are unknown.

$$\dot{x} = -Ax + XW^*S(x) + X_1W_1^*S_1(x)u \tag{28}$$

Therefore, the state regulation problem is analyzed for the system (28) instead of (27). Since, W^* and W_1^* are unknown, our solution consists of designing a control law $u(W, W_1, x)$ and appropriate update laws for W and W_1 to guarantee convergence of the state to zero and in some cases,which will be analyzed in the following sections, boundedness of x and of all signals in the closed loop.

The following mild assumptions are also imposed on (27), to guarantee the existence and uniqueness of solution for any finite initial condition and $u \in U$.

Proposition 4.1. Given a class U of admissible inputs, then for any $u \in U$ and any finite initial condition, the state trajectories are uniformly bounded for any finite $T > 0$. Hence, $|x(T)| < \infty$.

Proposition 4.2. The vector fields f, g_i, $i = 1, 2, \ldots, n$ are continuous with respect to their arguments and satisfy a local Lipchitz condition so that the solution $x(t)$ of (27) is unique for any finite initial condition and $u \in U$.

4.1.2. Neuro-fuzzy representation

We are using an affine in the control fuzzy dynamical system, which approximates the system in (27) and uses two fuzzy subsystem blocks for the description of $f(x)$ and $G(x)$ as follows

$$\hat{f}(x) = -A\hat{x} + \sum \bar{f}_{j_1,\ldots,j_n}^{l_1,\ldots,l_n} \times I_{j_1,\ldots,j_n}^{l_1,\ldots,l_n}(x) \tag{29}$$

$$\hat{g}_i(x) = \sum (\bar{g}_i)_{j_1,\ldots,j_n}^{l_1,\ldots,l_n} \times I1_{j_1,\ldots,j_n}^{l_1,\ldots,l_n}(x) \tag{30}$$

where the summation is carried out over the number of all available fuzzy rules, I, I_1 are appropriate fuzzy rule indicator functions and the meaning of indices $\bullet_{j_1,\ldots,j_n}^{l_1,\ldots,l_n}$ has already been described in Section 2.

According to Lemma 3.1, every indicator function can be approximated with the help of a suitable HONNF. Therefore, every I, I_1 can be replaced with a corresponding HONNF as follows

$$\hat{f}(x) = -A\hat{x} + \sum \bar{f}^{l_1,\ldots,l_n}_{j_1,\ldots,j_n} \times N^{l_1,\ldots,l_n}_{j_1,\ldots,j_n}(x) \tag{31}$$

$$\hat{g}_i(x) = \sum (\bar{g}_i)^{l_1,\ldots,l_n}_{j_1,\ldots,j_n} \times N_1{}^{l_1,\ldots,l_n}_{j_1,\ldots,j_n}(x) \tag{32}$$

where N, N_1 are appropriate HONNFs.

In order to simplify the model structure, since some rules result to the same output partition, we could replace the NNs associated to the rules having the same output with one NN and therefore the summations in (31), (32) are carried out over the number of the corresponding output partitions. Therefore, the affine in the control fuzzy dynamical system in (29), (30) is replaced by the following equivalent affine Recurrent High Order Neural Network (RHONN), which depends on the centers of the fuzzy output partitions $\bar{f}_l$ and $\bar{g}_{i,l}$

$$\dot{\hat{x}} = -A\hat{x} + \sum_{l=1}^{Npf} \bar{f} \times N_l(x) + \sum_{i=1}^{n} \left(\sum_{l=1}^{Npg_i} (\bar{g}_i)_l \times N_{1l}(x) \right) u_i. \tag{33}$$

Or in a more compact form

$$\dot{\hat{x}} = -A\hat{x} + XWS(x) + X_1 W_1 S_1(x) u. \tag{34}$$

Where A is a $n \times n$ stable matrix which for simplicity can be taken to be diagonal as $A = \text{diag}[a_1, a_2, \ldots, a_n]$, X, X_1 are matrices containing the centres of the partitions of every fuzzy output variable of $f(x)$ and $g(x)$ respectively, $S(\chi), S_1(\chi)$ are matrices containing high order combinations of sigmoid functions of the state χ and W, W_1 are matrices containing respective neural weights according to (24) and (33). The dimensions and the contents of all the above matrices are chosen so that $XWS(\chi)$ is a $n\times 1$ vector and $X_1 W_1 S_1(\chi)$ is a $n\times n$ matrix. Without compromising the generality of the model we assume that the vector fields in (30) are such that the matrix G is diagonal. For notational simplicity we assume that all output fuzzy variables are partitioned to the same number, m, of partitions. Under these specifications X is a $n \times n\cdot m$ block diagonal matrix of the form $X = \text{diag}(X^1, X^2, \ldots, X^n)$ with each X^i being an m-dimensional raw vector of the form

$$X^i = \begin{bmatrix} \bar{f}^i_1 & \bar{f}^i_2 & \cdots & \bar{f}^i_m \end{bmatrix}$$

or in a more detailed form

$$X = \begin{bmatrix} \bar{f}^1_1 & \cdots & \bar{f}^1_m & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \bar{f}^2_1 & \cdots & \bar{f}^2_m & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 & \bar{f}^n_1 & \cdots & \bar{f}^n_m \end{bmatrix}$$

where $\bar{f}_p^i$ denotes the center or the fuzzy pth partition of f_i. These centers can be determined manually or automatically with the help of a fuzzy c-means clustering algorithm as a part of the off-line structural identification procedure. Also, $S(\chi) = \begin{bmatrix} s_1(\chi) & \ldots & s_k(\chi) \end{bmatrix}^T$, where each $s_i(\chi)$ with $i = \{1, 2, \ldots, k\}$, is a high order combination of sigmoid functions of the state variables and W is a $n \cdot m \times k$ matrix with neural weights. W assumes the form $W = \begin{bmatrix} W^1 & \cdots & W^n \end{bmatrix}^T$, where each W^i is a matrix $\left[w_{j\,l}^i\right]_{m\times k}$. X_1 is a $n \times n \cdot m$ block diagonal matrix $X_1 = \mathrm{diag}({}^1X^1, {}^1X^2, \ldots, {}^1X^n)$ with each ${}^1X^i$ being an m-dimensional raw vector of the form

$${}^1X^i = \begin{bmatrix} \bar{g}_1^{i,i} & \bar{g}_2^{i,i} & \cdots & \bar{g}_m^{i,i} \end{bmatrix},$$

or in a more detailed form

$${}^1X = \begin{bmatrix} \bar{g}_1^{1,1} & \cdots & \bar{g}_m^{1,1} & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \bar{g}_1^{2,2} & \cdots & \bar{g}_m^{2,2} & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 & \bar{g}_1^{n,n} & \cdots & \bar{g}_m^{n,n} \end{bmatrix}$$

where $\bar{g}_k^{i\,i}$ denotes the center or the kth partition of g_{ii}. W_1 is a $m \cdot n \times n$ block diagonal matrix $W_1 = \mathrm{diag}({}^1W^1, {}^1W^2, \ldots, {}^1W^n)$, where each ${}^1W^i$ is a column vector $\left[{}^1w_{j\,l}^i\right]_{m\times 1}$ of neural weights. Finally, $S_1(\chi)$ is a $n \times n$ diagonal matrix with each diagonal element $s_i(\chi)$ being a high order combination of sigmoid functions of the state variables.

According to the above definitions the configuration of the F-RHONN approximator is shown in Figure 1. When the inputs are given into the fuzzy-neural network shown in Figure 1, the output of layer IV gives indicator function outputs which activate the corresponding rules and are calculated by Eq. (24). At layer V, each node performs a fuzzy rule while layer VI gives the function output.

The approximator of indicator functions, has four layers. At layer I, the input nodes represent input and state measurable variables. At layer II, the nodes represent the values of the sigmoidal functions. At layer III, the nodes are the values of high order sigmoidal combinations. The links between layer III and layer IV are fully connected by the weighting vectors $W = \begin{bmatrix} W^1 & \cdots & W^n \end{bmatrix}^T$, the adjustable parameters. Finally, at layer IV the output represents the values of indicator functions.

It has to be mentioned here that the proposed neuro-fuzzy representation, finally given by (34), offers some advantages over other fuzzy or neural adaptive representations. Considering the proposed approach from the adaptive fuzzy system (AFS) point of view, the main advantage is that the proposed approach is much less vulnerable in initial AFS design assumptions because there is no need for a-priori information related to the IF part of the rules (type and centers of membership functions, number of rules). This information is replaced by the existence of HONNFs. Considering the proposed approach from the NN point of view, the final representation of the dynamic equations is actually a combination of High Order Neural Networks, each one being specialized in approximating a function related

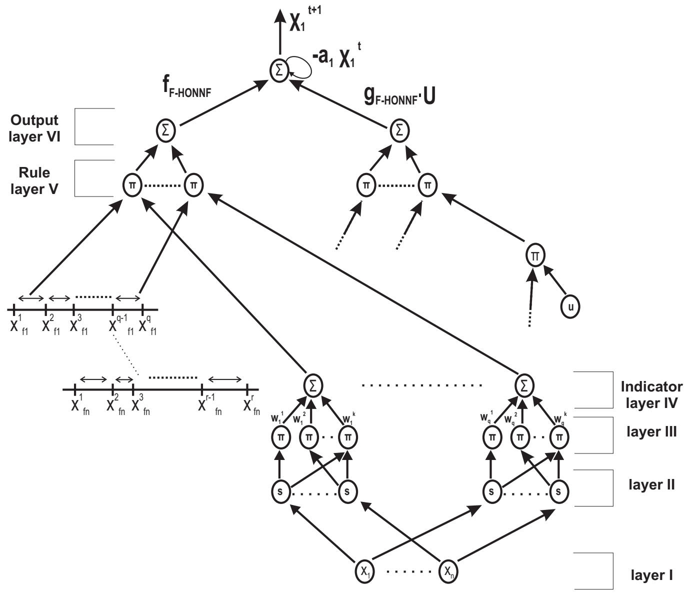

Fig. 1. Configuration of a neuro-fuzzy (F-RHONN) approximator.

to a corresponding center of output state membership function. This way, instead of having one large HONNF trying to approximate "everything" we have many, probably smaller, specialized HONNFs. Conceptually, this strategy is expected to present better approximation results; this is also verified in the simulations section. Moreover, as it will be seen in Section 4.2.1, due to the particular bond of each HONNF with one center of an output state membership function, the existence of the control law is assured by introducing a novel technique of parameter "hopping" in the corresponding weight updating laws.

4.2. Adaptive regulation – Complete matching

In this subsection we investigate the adaptive regulation problem when the modeling error term is zero, or in other words, when we have complete model matching. Under this assumption the unknown system can be written as (28), where $x \in \mathbb{R}^n$ is the system state vector, $u \in \mathbb{R}^n$ are the control inputs, X, X_1 are $n \times n \cdot m$ block diagonal matrices, W^* is a $n \cdot m \times k$ matrix of synaptic weights, W_1^* is a $m \cdot n \times n$ block diagonal matrix and A is a $n \times n$ matrix with positive eigenvalues which for simplicity can be taken diagonal. Finally, $S(x)$ is a n-dimensional vector and $S_1(x)$ is a $n \times n$ diagonal matrix with each diagonal element $s_i(x)$ being a high order combination of sigmoid functions of the state variables.

Let us take a function $h(x) : \mathbb{R}^n \to \mathbb{R}^+$ of class C^2 having the following form

$$h(x) = \frac{1}{2} |x|^2 = \frac{1}{2} x^T x \tag{35}$$

which involves the state variables. The derivative of the above equation with respect to time according to Eq. (28) is

$$\dot{h}(x) = \frac{\partial h^T}{\partial x}\frac{\partial x}{\partial t} = \frac{\partial h^T}{\partial x}\left[-Ax + XW^*S(x) + X_1W_1^*S_1(x)u\right]$$

which is linear with respect to W^* and W_1^* and can be written as

$$\dot{h} + \frac{\partial h^T}{\partial x}Ax = \frac{\partial h^T}{\partial x}XW^*S(x) + \frac{\partial h^T}{\partial x}X_1W_1^*S_1(x)u. \tag{36}$$

Define now, the estimation error as

$$v \triangleq \frac{\partial h^T}{\partial x}XWS(x) + \frac{\partial h^T}{\partial x}X_1W_1S_1(x)u - \dot{h} - \frac{\partial h^T}{\partial x}Ax \tag{37}$$

$$\begin{aligned} v &\triangleq \frac{\partial h^T}{\partial x}XWS(x) + \frac{\partial h^T}{\partial x}X_1W_1S_1(x)u - \frac{\partial h^T}{\partial x}XW^*S(x) - \frac{\partial h^T}{\partial x}X_1W_1^*S_1(x)u \\ \Rightarrow v &\triangleq \frac{\partial h^T}{\partial x}X\tilde{W}S(x) + \frac{\partial h^T}{\partial x}X_1\tilde{W}_1S_1(x)u \end{aligned} \tag{38}$$

where W and W_1 are estimates of W^* and W_1^* respectively, obtained by update laws which are to be designed in the sequel. This signal cannot be measured since $\dot{h}$ is unknown. To overcome this problem, we use the following filtered version of error v

$$v = \dot{\xi} + \kappa\xi$$

and according to Eq. (37) we have that

$$v = -\dot{h} + \frac{\partial h^T}{\partial x}\left[-Ax + XWS(x) + X_1W_1S_1(x)u\right] \tag{39}$$

$$\dot{\xi} + \kappa\xi = -\dot{h} + \frac{\partial h^T}{\partial x}\left[-Ax + XWS(x) + X_1W_1S_1(x)u\right] \tag{40}$$

where κ a strictly positive constant. To implement Eq. (40), we take

$$\xi \triangleq \varsigma - h. \tag{41}$$

Employing Eq. (41), Eq. (40) can be written as

$$\dot{\varsigma} + \kappa\varsigma = \kappa h + \frac{\partial h^T}{\partial x}\left[-Ax + XWS(x) + X_1W_1S_1(x)u\right] \tag{42}$$

with state $\varsigma \in \mathbb{R}$. This method is referred to as error filtering. Furthermore, Eq. (42) substituting Eq. (35) becomes

$$\dot{\varsigma} + \kappa\varsigma = \kappa h - x^TAx + x^TXWS(x) + x^TX_1W_1S_1(x)u. \tag{43}$$

To continue, consider the Lyapunov-like function

$$L = \frac{1}{2}\xi^2 + \frac{1}{2}\mathrm{tr}\left\{\dot{\tilde{W}}^T\tilde{W}\right\} + \frac{1}{2}\mathrm{tr}\left\{\dot{\tilde{W}}_1^T\tilde{W}_1\right\}. \tag{44}$$

Where $\tilde{W} = W - W^*$ and $\tilde{W}_1 = W_1 - W_1^*$
If we take the derivative of Eq. (44) with respect to time we obtain

$$\dot{L} = \xi\dot{\xi} + \mathrm{tr}\left\{\dot{W}^T\tilde{W}\right\} + \mathrm{tr}\left\{\dot{W}_1^T\tilde{W}_1\right\}. \tag{45}$$

Employing Eqs. (40), (45) becomes

$$\dot{L} = -\kappa\xi^2 + \xi[-\dot{h} - x^TAx + x^TXWS(x) + x^TX_1W_1S_1(x)] + \mathrm{tr}\left\{\dot{W}^T\tilde{W}\right\} + \mathrm{tr}\left\{\dot{W}_1^T\tilde{W}_1\right\}$$

which together with Eq. (36) gives

$$\begin{aligned}\dot{L} \quad = \quad & -\kappa\xi^2 + \xi[-x^TXW^*S(x)x^TX_1W_1^*S_1(x)u + x^TXWS(x) + x^TX_1W_1S_1(x)u] \\ & + \mathrm{tr}\left\{\dot{W}^T\tilde{W}\right\} + \mathrm{tr}\left\{\dot{W}_1^T\tilde{W}_1\right\}\end{aligned}$$

or equivalently

$$\dot{L} = -\kappa\xi^2 + \xi x^TX\tilde{W}S(x) + \xi x^TX_1\tilde{W}_1S_1(x)u + \mathrm{tr}\left\{\dot{W}^T\tilde{W}\right\} + \mathrm{tr}\left\{\dot{W}_1^T\tilde{W}_1\right\}. \tag{46}$$

Hence, if we choose

$$\mathrm{tr}\left\{\dot{W}^T\tilde{W}\right\} = -\xi x^TX\tilde{W}S(x) \tag{47}$$

and

$$\mathrm{tr}\left\{\dot{W}_1^T\tilde{W}_1\right\} = -\xi x^TX_1\tilde{W}_1S_1(x)u \tag{48}$$

$\dot{L}$ becomes

$$\dot{L} = -\kappa\xi^2 \leq 0 \tag{49}$$

which means that $\dot{L}$ is negative semidefinite. It can be easily verified that Eqs. (47), (48) after using property (2) of Section 2.1 and making the appropriate operations, can be element wise written as

a) for the elements of W^i

$$w_{jl}^i = -\xi\bar{f}_j^i x_i s_l(x) \tag{50}$$

b) for the elements of ${}^1W^i$

$$\dot{w}_{j1}^i = -\xi\bar{g}_j^{ii}x_iu_is_i(x) \tag{51}$$

or equivalently ${}^1\dot{W}^i = -({}^1X^i)^T\xi x_iu_is_i(x)$ for all $i, j = 1, 2, \ldots, n$ and $l = 1, 2, \ldots, k$.

Equations (50) and (51) can be finally written in a compact form as

$$\dot{W} = -\xi X^TxS^T(x) \tag{52}$$

$$\dot{W}_1 = -\xi X_1^T x' U S_1(x) \tag{53}$$

where ξ is a scalar magnitude, x' is a diagonal matrix such that $x' = \mathrm{diag}\,[x_1, x_2, \ldots, x_n]$ and $U = \mathrm{diag}\,[u_1, u_2, \ldots, u_n]$.

Now we can prove the following lemma

Lemma 4.3. Consider the system

$$\dot{x} = -Ax + XW^* S(x) + X_1 W_1^* S_1(x) u$$

$$\dot{\varsigma} = -\kappa\varsigma + \kappa h - x^T A x + x^T X W S(x) + x^T X_1 W_1 S_1(x) u$$

$$\xi \triangleq \varsigma - h$$

$$h(x) = \tfrac{1}{2} |x|^2 .$$

The update laws

$$\dot{W} = -\xi X^T x S^T(x), \qquad \dot{W}_1 = -\xi X_1^T x' U S_1(x)$$

guarantee the following properties

- $\xi, |x|, W, W_1, \varsigma \in L_\infty$
- $|\xi| \in L_2$
- $\lim_{t\to\infty} \xi(t) = 0,\ \lim_{t\to\infty} \dot{W}(t) = 0,\ \lim_{t\to\infty} \dot{W}_1(t) = 0$

provided that $u \in L_\infty$.

Proof. From Eq. (49) we have that $L \in L_\infty$ which implies $\xi, W, W_1 \in L_\infty$. Since $u \in L_\infty$ then $x \in L_\infty$, hence $h \in L_\infty$. Furthermore, $\xi = \zeta - h$, hence $\zeta \in L_\infty$. Since L is a monotone decreasing function of time and bounded from below, the $\lim_{t\to\infty} L(t) = L_\infty$ exists so by integrating $\dot{L}$ from 0 to ∞ we have

$$\int_0^\infty |\xi|^2 \, \mathrm{d}t = \frac{1}{r} |L(0) - L_\infty| < \infty$$

which implies that $|\xi| \in L_2$. We also have that

$$\dot{\xi} = -\kappa\xi + x^T X \tilde{W} S(x) + x^T X_1 \tilde{W}_1 S_1(x) u$$

Hence, $\dot{\xi} \in L_\infty$ provided that $u \in L_\infty$. Having in mind that $\xi \in L_2 \cap L_\infty$ and $\dot{\xi} \in L_\infty$, applying Barbalat's Lemma [17] we conclude that $\lim_{t\to\infty} \xi(t) = 0$. Now, using the boundedness of $u, S(x), S_1(x)$, x and the convergence of $\xi(t)$ to zero, we have that $\dot{W}, \dot{W}_1$ also converge to zero. □

To proceed further, we observe that $\dot{h}$ can be written as

$$\dot{h} = x^T \left[-Ax + XWS(x) + X_1 W_1 S_1(x) u\right] - x^T X \tilde{W} S(x) - x^T X_1 \tilde{W}_1 S_1(x) u.$$

Hence, if we choose the control input u to be

$$u = -\left[X_1 W_1 S_1(x)\right]^{-1} XWS(x) \tag{54}$$

then $\dot{h}$ becomes

$$\dot{h} = -x^T Ax - x^T X\tilde{W}S(x) - x^T X_1 \tilde{W}_1 S_1(x)u. \tag{55}$$

Moreover, Eq. (55) can be written

$$\dot{h} \leq -\frac{c}{2}|x|^2 - \dot{\xi} - \kappa\xi, \tag{56}$$

where $c = 2n\lambda_{\min}(A)$, with $\lambda_{\min}(A)$ denoting the minimum eigenvalue of matrix A. Observe that Eq. (56) is equivalent to

$$\dot{h} \leq -ch - \dot{\xi} - \kappa\xi. \tag{57}$$

Furthermore,

$$h = \zeta - \xi$$

hence, Eq. (57) becomes

$$\begin{aligned} \dot{\varsigma} &\leq -c\varsigma + c\xi - \kappa\xi \\ &\leq -c\varsigma + (c+\kappa)|\xi|, \end{aligned} \tag{58}$$

which as it will be seen later, can be used to prove that $x(t) \to 0$.

4.2.1. Introduction to the parameter hopping

It is important to say that, in order to apply the control law given by Eq. (54) we have to assure the existence of $\left[X_1 W_1 S_1(\chi)\right]^{-1}$. Since $S_1(\chi)$ is diagonal with its elements $s_i(\chi) \neq 0$ and X_1, W_1 are block diagonal the existence of the inverse is assured when ${}^1X^i \cdot {}^1W^i \neq 0$, $\forall\, i = 1, \ldots n$. Therefore, W_1 has to be confined such that $\left|{}^1X^i \cdot {}^1W^i\right| \geq \theta_i > 0$, with θ_i being a design parameter. In case the boundary defined by the above confinement is nonlinear the updating W_1 can be modified by using a projection algorithm [17]. However, in our case the boundary surface is linear and the direction of updating is normal to it because $\nabla\left[{}^1X^i \cdot {}^1W^i\right] = {}^1X^i$. Therefore, the projection of the updating vector on the boundary surface is of no use. Instead, using concepts from multidimensional vector geometry we modify the updating law such that, when the weight vector approaches (within a safe distance θ_i) the forbidden hyper-plane ${}^1X^i \cdot {}^1W^i = 0$ and the direction of updating is toward the forbidden hyper-plane, it introduces a *hopping* which drives the weights in the direction of the updating but on the other side of the space, where here the weight space is divided into two sides by the forbidden hyper-plane. This procedure is depicted in Figure 2, where a simplified 2-dimensional representation is given. The magnitude of the hopping can be determined following the vectorial proof given bellow.

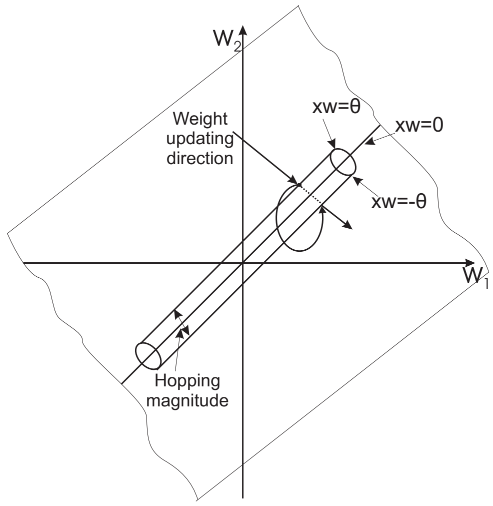

Fig. 2. Pictorial Representation of parameter hopping.

4.2.2. Vectorial proof of parameter hopping

In selecting the terms involved in parameter hopping we start from the vector definition of a line, of a plane and the distance of a point to a plane. The equation of a line in vector form is given by $r = a + \lambda t$, where a is the position vector of a given point of the line, t is a vector in the direction of the line and λ is a real scalar. By giving different numbers to λ we get different points of the line each one represented by the corresponding position vector r. The vector equation of a plane can be defined by using one point of the plane and a vector normal to it. In this case $r \cdot n = a \cdot n = d$ is the equation of the plane, where a is the position vector of a given point on the plane, n is a vector normal to the plane and d is a scalar. When the plane passes through zero, then apparently $d = 0$. To determine the distance of a point B with position vector b from a given plane we consider Figure 4 and combine the above definitions as follows. Line BN is perpendicular to the plane and is described by vector equation $r = b + \lambda n$, where n is the normal to the plane vector. However, point N also lies on the plane and in case the plane passes through zero

$$r \cdot n = 0 \Rightarrow (b + \lambda n) \cdot n = 0 \Rightarrow \lambda = \tfrac{-b \cdot n}{\|n\|}.$$

Apparently, if one wants to get the position vector of B′ (the symmetrical of B in respect to the plane), this is given by

$$r = b - 2\tfrac{b \cdot n}{\|n\|} n.$$

In our problem $b = {}^1W^i$, our plane is described by the equation ${}^1X^i \cdot {}^1W^i = 0$ and as it has already been mentioned the normal to it is the vector ${}^1X^i$.

Theorem 4.4 below introduces this *hopping* in the weight updating law.

Theorem 4.4. Consider the control scheme described from equations 52, 54, 55 and 60. The updating law:

a) For the elements of W^i given by (50)

b) For the elements of ${}^1W^i$ given by 51 modified according to the hopping method:

$$
{}^1\dot{W}^i = \begin{cases} -\xi\left({}^1X^i\right)^T x_i u_i s_i(x) & \begin{array}{l}\text{if } \left|{}^1X^i \cdot {}^1W^i\right| > \theta_i \\ \text{or } {}^1X^i \cdot {}^1W^i = \pm\theta_i \\ \text{and } {}^1X^i \cdot {}^1\dot{W}^i <> 0\end{array} \\ \\ \begin{array}{l} -\xi\left({}^1X^i\right)^T x_i u_i s_i(x) - \\ -\frac{2\left({}^1X^i\,{}^1W^i\left({}^1X^i\right)^T\right)}{\mathrm{tr}\left\{({}^1X^i)^T\,{}^1X^i\right\}} \end{array} & \begin{array}{l}\text{if } 0 < {}^1X^i \cdot {}^1W^i < \theta_i \\ \text{and } {}^1X^i \cdot {}^1\dot{W}^i > 0 \\ \text{or if } -\theta_i < {}^1X^i \cdot {}^1W^i < 0 \\ \text{and } {}^1X^i \cdot {}^1\dot{W}^i < 0\end{array} \end{cases} \tag{59}
$$

guarantees the properties of Lemma 4.3 and assures the existence of the control signal.

Proof. In order the properties of lemma 4.3 to be valid it suffices to show that by using the modified updating law for ${}^1W^i$ the negativeness of the Lyapunov function is not compromised. Indeed the first part of the modified form of ${}^1\dot{W}^i$ is exactly the same with (51) and therefore according to the development of (51) the negativeness of V is in effect. The first part is used when the weights are at a certain distance (condition if $\left|{}^1X^i \cdot {}^1W^i\right| > \theta_i$) from the forbidden plane or at the safe limit (condition ${}^1X^i \cdot {}^1W^i = \pm\theta_i$) but with the direction of updating moving the weights far from the forbidden plane (condition ${}^1X^i \cdot {}^1\dot{W}^i <> 0$).

In the second part of ${}^1\dot{W}^i$, term $-\frac{2}{\mathrm{tr}\{({}^1X^i)^T\,{}^1X^i\}}\,{}^1X^i\,{}^1W^i\,({}^1X^i)^T$ determines the magnitude of weight *hopping*, which as explained in the vectorial proof of "hopping" and is depicted in Figure 4 has to be two times the distance of the current weight vector to the forbidden hyper-plane. Therefore the *existence* of the control signal is assured because the weights never reach the forbidden plane. Regarding the *negativeness* of $\dot{V}$ we proceed as follows.

Let that ${}^1W^{*i}$ contains the actual unknown values of ${}^1W^i$ such that

$$\left|{}^1X^i \cdot {}^1W^{*i}\right| \gg \theta_i$$

and that ${}^1\tilde{W}^i = {}^1W^i - {}^1W^{*i}$. Then, the weight hopping can be equivalently written with respect to ${}^1\tilde{W}^i$ as $-2\theta_i{}^1\tilde{W}^i/\|{}^1\tilde{W}^i\|$. Under this consideration the modified updating law is rewritten as ${}^1\dot{W}^i = -\xi\left({}^1X^i\right)^T x_i u_i s_{i1}(x) - 2\theta_i{}^1\tilde{W}^i/\|{}^1\tilde{W}^i\|$. With this updating law it can be easily verified that

$$\dot{L} = -\kappa\xi^2 - \gamma_1\Theta$$

with Θ being a positive constant expressed as

$$\Theta = \sum 2\theta_i\left(({}^1\tilde{W}^i)^T)^1\tilde{W}^i)\right)/\|{}^1\tilde{W}^i\| \geq 0$$

where the summation includes all weight vectors which require hopping.

Therefore, the negativeness of $\dot{L}$ is actually strengthened due to the last negative term. □

The inclusion of weight hopping in the weights updating law guarantees that the control signal does not go to infinity. Apart from that, it is also of practical use to assure that $X_1 W_1 S_1(x)$ does not go even temporarily to infinity because in this case the method may become algorithmically unstable driving at the same time the control signal to zero failing to control the system. To assure that this situation does not happen we have again to assure that $\left|{}^1X^i \cdot {}^1W^i\right| < \rho_i$ with ρ_i being again a design parameter determining an external limit for ${}^1X^i \cdot {}^1W^i$. Following the same lines of thought with the case of weight hopping introduced above we could again consider the forbidden hyperplanes being defined by the equation $\left|{}^1X^i \cdot {}^1W^i\right| = \rho_i$. When the weight vector reaches one of the forbidden hyper-planes ${}^1X^i \cdot {}^1W^i = \rho_i$ and the direction of updating is toward the forbidden hyper-plane, a new *hopping* is introduced which moves the weights to the other forbidden hyper-plane keeping however the direction of the updating intact. This procedure is depicted in Figure 3, in a simplified 2-dimensional representation. The magnitude of hopping is $-\frac{2}{\mathrm{tr}\{({}^1X^i)^T\,{}^1X^i\}}\,{}^1X^i\,{}^1W^i\,({}^1X^i)^T$ being determined by following again the same vectorial proof and Figure 4. By performing *hopping* when ${}^1X^i \cdot {}^1W^i$ reaches either the inner or outer forbidden planes, ${}^1X^i \cdot {}^1W^i$ is confined to lie in space P defined by these hyper-planes. The weight updating law for ${}^1W^i$ incorporating the two hopping conditions can now be expressed as

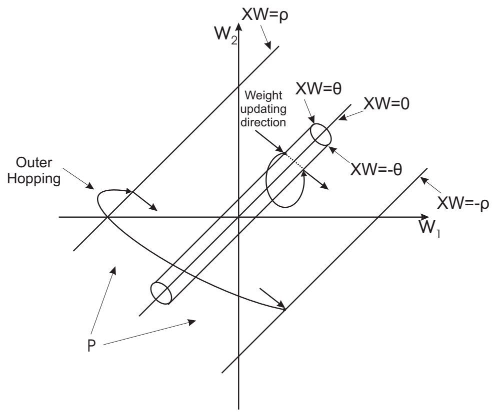

Fig. 3. Pictorial Representation of inner and outer parameter hopping.

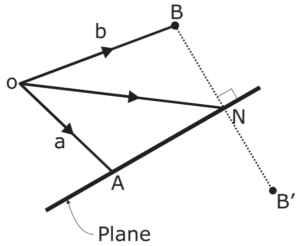

Fig. 4. Vector explanation of parameter hopping.

$$
{}^1\dot{W}^i = \begin{cases} -\xi\left({}^1X^i\right)^T x_i u_i s_i(x) & \begin{array}{l}\text{if } {}^1X^i \cdot {}^1W^i \in P \\ \text{or } {}^1X^i \cdot {}^1W^i = (\pm\theta_i \text{ or } \pm\rho_i) \\ \text{and } {}^1X^i \cdot {}^1\dot{W}^i <> \text{ or} \\ >< 0\end{array} \\ \begin{array}{l} -\xi\left({}^1X^i\right)^T x_i u_i s_i(x) - \\ -\frac{2\left({}^1X^i\,{}^1W^i\left({}^1X^i\right)^T\right)}{\mathrm{tr}\left\{({}^1X^i)^T\,{}^1X^i\right\}}\end{array} & \begin{array}{l}\text{if } 0 < {}^1X^i \cdot {}^1W^i < \theta_i \\ \text{and } {}^1X^i \cdot {}^1\dot{W}^i > 0 \\ \text{or if } -\theta_i < {}^1X^i \cdot {}^1W^i < 0 \\ \text{and } {}^1X^i \cdot {}^1\dot{W}^i < 0 \\ \text{or if } {}^1X^i \cdot {}^1W^i < -\rho_i \\ \text{and } {}^1X^i \cdot {}^1\dot{W}^i > 0 \\ \text{or if } {}^1X^i \cdot {}^1W^i > \rho_i \\ \text{and } {}^1X^i \cdot {}^1\dot{W}^i < 0\end{array} \end{cases} \tag{60}
$$

The following lemma presents the properties of the hopping algorithm in detail.

Lemma 4.5. The updating law (60) incorporating the two 'hopping' conditions, can only make Lyapunov derivative more negative and in addition guarantee that ${}^1X^i \cdot {}^1W^i \in P$ for all $i = 1, 2, \ldots, n$, provided that ${}^1X^i(0) \cdot {}^1W^i(0) \in P$.

Proof. In order to prove that the hopping modification given by (60) can only make $\dot{L}$ more negative, we go with the following cases.

Case 1: Activation of first (inner) hopping condition $\left|{}^1X^i \cdot {}^1W^i\right| \leq \theta_i$.

This case has already been examined in Theorem 4.4. $\dot{L}$ is augmented by the following quantity

$$\Re_a = -\gamma_1\Theta = -\gamma_1 \sum 2\theta_i \left(({}^1\tilde{W}^i)^T)^1\tilde{W}^i)\right) / \|{}^1\tilde{W}^i\|$$

where the summation includes all weight vectors which require inner hopping. Obviously $\Re_a < 0$.

Case 2: Activation of second (outer) hopping condition $\left|{}^1X^i \cdot {}^1W^i\right| \geq \rho_i$.

Following the same lines of proof as with Theorem 4.4 it is easy to conclude that in this case $\dot{L}$ is augmented by the following quantity

$$\Re_b = -\gamma_1\Theta_1 = -\gamma_1 \sum 2\rho_i \left(({}^1\tilde{W}^i)^T)^1\tilde{W}^i) \right) / \|{}^1\tilde{W}^i\|$$

where the summation includes all weight vectors which require outer hopping. Obviously $\Re_b < 0$

It is obvious that by following the procedure of inner or outer hopping, once the initial weights are such that ${}^1X^i(0) \cdot {}^1W^i(0) \in P$ then ${}^1X^i \cdot {}^1W^i$ will never leave P. □

Lemma 4.6. Let η be a C^1 time function defined on $[0, T)$ where $(0 \leq T \leq \infty)$, satisfying

$$\dot{\eta} \leq -c\eta + a(t) + \beta(t),$$

where c is a strictly positive constant and $a(t)$ and $\beta(t)$ are two positive time functions belonging to $L_2(0, T)$ that is

$$\int_0^T a^2(t)\,\mathrm{d}t \leq M_1 < \infty,$$

and

$$\int_0^T \beta^2(t)\,\mathrm{d}t \leq M_2 < \infty.$$

Under this assumption, $\eta(t)$ is upper bounded on $(0, T)$ and precisely

$$\eta(t) \leq \xi^{\frac{M_1}{c}} \left[\eta(0) + \sqrt{\tfrac{2}{c}}\sqrt{M_2} \right], \quad \forall\, t \in [0, T),$$

moreover, if T is infinite then

$$\limsup_{t\to\infty} \eta(t) \leq 0.$$

Observe that (58) with $\kappa = 1$ becomes

$$\dot{\varsigma} \ \leq -c\varsigma + (1+\kappa)\,|\xi|\,, \tag{61}$$

however, from Lemma 2 we have that $|\xi| \in L_2$ so $(1+c)\,|\xi| \in L_2$. Furthermore, observe that obviously T can be extended to be infinite. Hence, Lemma 4 can be applied in (61) with $M_1 = 0$ to obtain

$$\limsup_{t\to\infty} \varsigma(t) \leq 0. \tag{62}$$

Moreover, since $h = \zeta - \xi$ and $h \geq 0$ we have $\varsigma(t) \geq \xi(t)$, or

$$-\varsigma(t) \leq -\xi(t), \tag{63}$$

but from Lemma 2 we have
$$\lim_{t\to\infty} \xi(t) = 0. \tag{64}$$
Hence, (63) together with (62) and (64) prove
$$\lim_{t\to\infty} \varsigma(t) = 0. \tag{65}$$
Furthermore, since $h = \zeta - \xi$, (64) and (65) yield
$$\lim_{t\to\infty} h\left(x\left(t\right)\right) = 0,$$
which by the definition of $h(x)$ finally implies that
$$\lim_{t\to\infty} \left|x\left(t\right)\right| = 0,$$
therefore, we have proven the following theorem.

Theorem 4.7. The closed loop system
$$\begin{gathered}
\dot{x} = -Ax + XW^{*}S(x) + X_1 W_1^{*} S_1(x)u, \\
\dot{\varsigma} = -\kappa\varsigma + \kappa h - x^T Ax + x^T XWS(x) + x^T X_1 W_1 S_1(x)u, \\
u = -\left[X_1 W_1 S_1(x)\right]^{-1} XWS(x), \xi \triangleq \varsigma - h, \\
h(x) = \tfrac{1}{2}\left|x\right|^2, \\
\kappa = 1,
\end{gathered}$$
together with update laws given by (50) and (60) guarantee that
$$\lim_{t\to\infty} \left|x\left(t\right)\right| = 0.$$

From the aforementioned analysis it is obvious that different choices of $h(x)$ and κ, lead to different adaptive regulators. It is anticipated that appropriate selection of κ, could attenuate the effects of the uncertainties that may be present.

Remark 4.8. The control law (54) can be also extended to the following form
$$u = -\left[X_1 W_1 S_1(x)\right]^{-1}\left(XWS(x) + Kx\right) \tag{66}$$
where K is a positive definite diagonal matrix defined by the designer. It can be easily verified that with this control law $\dot{h}$ becomes $\dot{h} = -x^T \Lambda x - x^T X\tilde{W}S(x) - x^T X_1 \tilde{W}_1 S_1(x)u$. where $\Lambda = A + K$ is a stable matrix. Therefore, c in (56) becomes $c = 2n\lambda_{\min}(\Lambda)$, with $\lambda_{\min}(\Lambda)$ denoting the minimum eigenvalue of the Λ matrix. Since $\lambda_{\min}(A) \le \lambda_{\min}(\Lambda)$ the proof of theorem 4.7 is still valid with the property of $\lim_{t\to\infty} \left|x\left(t\right)\right| = 0$ being actually enhanced. Therefore term Kx is actually acting as a robustifying term.

Conclusively, in our approach, referred to as *direct adaptive fuzzy-RHONN control*, the control law parameters are estimated on-line except from the centers of the membership function partitions of vector fields f and g_i, which are initially determined off-line. Moreover, it is assumed that the actual system is parametrized using these parameters and this is an essential part of direct control schemes. The basic structure of the *direct adaptive fuzzy-RHONN controller* is shown in Figure 5.

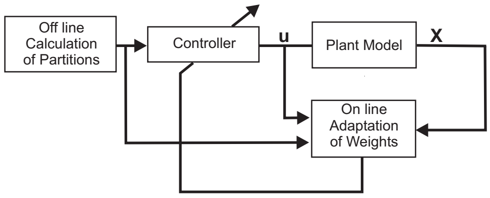

Fig. 5. Overall scheme of the proposed direct adaptive neuro-fuzzy control system.

5. SIMULATION RESULTS

To justify our motivation for using the proposed neuro-fuzzy system representation, its function approximation abilities are first compared to two well established approaches. The one uses only recurrent high order neural networks (RHONN) for system function approximation [46], while the other uses conventional fuzzy adaptive function approximation(Wang [52]). To demonstrate the potency of the proposed scheme in controlling an unknown plant, we present three simulation results which both assume only parametric uncertainty in the control phase. The first aims to test the ability of the proposed direct control scheme to regulate a 'Dc Motor' and shows a very good behavior. The other presents a comparison between the proposed method and a simple RHONN direct controller [46] on the same 'Dc Motor', which shows off the performance superiority of the proposed method. Finally, the proposed scheme is tested on a benchmark problem presenting chaotic behavior, which is also regulated using the proposed scheme.

5.1. Function approximation abilities of F-HONNF against RHONN's and Wang's approximations, using a well known benchmark

Van der Pol oscillator is usually used as a simple benchmark problem for testing identification and control schemes. It's dynamical equations are given by

$$\dot{x}_1 = x_2 \tag{67}$$
$$\dot{x}_2 = x_2 \cdot \left(a - x_1^2\right) \cdot b - x_1 + u. \tag{68}$$

It is our intention to compare the approximation abilities of the proposed Neuro-Fuzzy approach with Wang [52] adaptive Fuzzy approach and RHONN [46]. Eq. (68) is similar with Eq. (27), so we assume that $f(x)$ and $g(x)$ can be approximated using Wang's approach and Eq. (3) or alternatively by the XWS and $X_1W_1S_1$ term of Eq. (34) in the proposed approach, or WS and W_1S_1 for RHONN approach [46] respectively. The weight updating laws are chosen to be: For the Wang approach ([52], page 115)

$$\dot{\theta}_f = -\gamma_1 e^T P b_c \xi(x) \tag{69}$$
$$\dot{\theta}_g = -\gamma_2 e^T P b_c \xi(x) u_c \tag{70}$$

where only the simplified approach, without parameter projection case was necessary to be used.

For the RHONN approach we use the adaptive laws, which are described in [46], page 37. For the proposed F-HONNF approach we use the following adaptive laws [51]

$$\dot{W} = -X^T P e S^T \quad (71)$$
$$\dot{W}_1 = -X_1^T P E U S_1^T \quad (72)$$

where $E = \mathrm{diag}(e_1, \ldots, e_n)$ is a diagonal matrix containing the state variable errors. Numerical training data were obtained by using Eq. (68) with initial conditions $\begin{bmatrix} x_1(0) & x_2(0) \end{bmatrix} = \begin{bmatrix} 1 & 1 \end{bmatrix}$, and a persistently exciting input $u = 1 + 0.8\sin(0.001t)$.

The approximation of the dynamical equations using conventional fuzzy system approach requires a very large number of fuzzy rules for the approximation of the unknown functions. Choosing 40 or more membership functions for each variable x_i results in very accurate fuzzy representation. This representation requires 1600 rules, which in turn leads to a parameter explosion when using an adaptive scheme like that of Eq. (3) and consequently, it takes plenty of time for the simulations.

We are using the proposed approach with Eq. (34) to approximate Van der Pol oscillator. The proposed Neuro-Fuzzy model was chosen to use 5 output partitions of f and 5 output partitions of g. The number of high order sigmoidal terms (HOST) used in HONNF's were chosen to be first 2 $(s(x_1), s(x_2))$ and secondly 5 $(s(x_1), s(x_2), s(x_1) \cdot s(x_2), s^2(x_1), s^2(x_2))$ for two different simulations with the same benchmark. Therefore, the number of adjustable weights is 20 or 50 respectively, which is a much smaller number to that used in the conventional fuzzy approach.

In order our model to be equivalent with regard to other parameters except the adjustable weights we have chosen terms $\gamma_1 P b_c$ in Eq. (69), (70) and P (the updating learning rate) in Eq. (71), (72) to have the same values. Also, the RHONN model given from [46] is constructed with the same learning parameters and number of high order terms with these of F-HONNF approach. The parameters of the sigmoidal terms were chosen to be $a_1 = 0.1$, $a_2 = 6$, $b_1 = b_2 = 1$ and $c_1 = c_2 = 0$. Figures 6 and 7 shows the approximation of states x_1 and x_2 respectively while Figure 8 and 9 gives the evolution of errors x_1 and x_2.

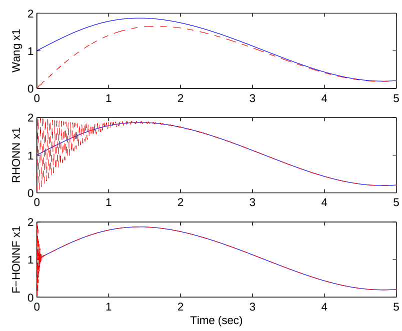

Fig. 6. Evolution of variable x_1 for Wang, RHONN and F-HONNF approach.

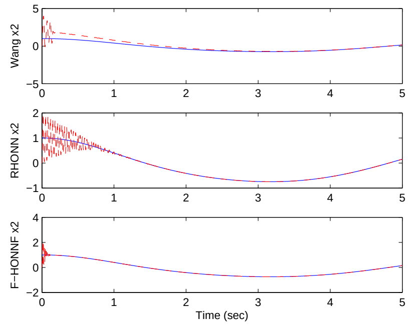

Fig. 7. Evolution of variable x_2 for Wang, RHONN and F-HONNF approach.

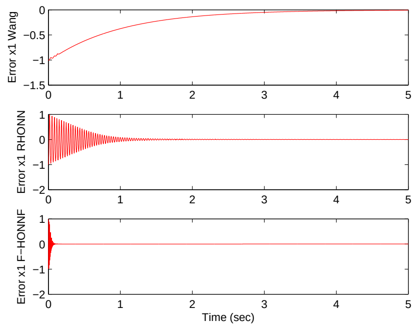

Fig. 8. Approximation Error of variable x_1 for Wang, RHONN and F-HONNF approach.

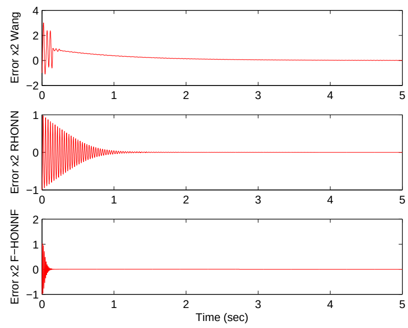

Fig. 9. Approximation Error of variable x_2 for Wang, RHONN and F-HONNF approach.

Table 1. Comparison of Wang, RHONN and FHONNF approaches for Van der Pol oscillator with 2 HOST.

	Wang	RHONN	FHONNF
MSE x_1	0.1038	0.0303	0.0058
MSE x_2	0.1401	0.0259	0.0087

Table 2. Comparison of Wang, RHONN and FHONNF approaches for Van der Pol oscillator with 5 HOST.

	Wang	RHONN	FHONNF
MSE x_1	0.1038	0.0180	0.0013
MSE x_2	0.1401	0.0149	0.0018

The mean squared error (MSE) for Wang's, RHONN and F-HONNF approaches were measured and are shown in Tables 1 and 2, demonstrating a significant (order of magnitude) increase in the approximation performance , although no a-priori information regarding fuzzy partitions and membership functions of the inputs were used.

Conclusively, the comparison between Wang and F-HONNF's leads to a huge superiority of F-HONNF's regarding the number of adjustable parameters and the approximation abilities. With respect to the RHONN approach the proposed F-HONNF approach is also much better.

5.2. Direct control of DC Motor with parametric uncertainties

In this section we apply the proposed approach to control the speed of a 1 KW DC motor with a normalized model described by the following dynamical equations [46]

$$\begin{aligned} T_a \frac{\mathrm{d}I_a}{\mathrm{d}t} &= -I_a - \Phi\Omega + V_a \\ T_m \frac{\mathrm{d}\Omega}{\mathrm{d}t} &= \Phi I_a - K_0\Omega - m_L \\ T_f \frac{\mathrm{d}\Phi}{\mathrm{d}t} &= -I_f + V_f \\ \Phi &= \frac{aI_f}{1+bI_f}. \end{aligned} \tag{73}$$

The states are chosen to be the armature current, the angular speed and the stator flux $x = \begin{bmatrix} I_a & \Omega & \Phi \end{bmatrix}$. As control inputs the armature and the field voltages

$u = \begin{bmatrix} V_a & V_f \end{bmatrix}$ are used. With this choice, we have

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -\frac{1}{T_a}x_1 - \frac{1}{T_a}x_2x_3 \\ \frac{1}{T_m}x_1x_3 - \frac{K_0}{T_m}x_2 - \frac{m_L}{T_m} \\ -\frac{1}{T_f}\frac{x_3}{a-\beta x_3} \end{bmatrix} + \begin{bmatrix} \frac{1}{T_a} & 0 \\ 0 & 0 \\ 0 & \frac{1}{T_f} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}. \tag{74}$$

Which is of a nonlinear, affine in the control form.

In many control schemes of the literature V_f is assumed constant. This may naturally occur when the field is produced by a permanent magnet or when it may be separately excited but is intentionally kept constant. This assumption may facilitate things because if V_f is constant then Φ is constant and the above nonlinear 3^{rd} order system can be linearized and reduced to a second order form having 2 states ($x_1 = I_a$ and $x_2 = \Omega$), with the value Φ being included as a constant parameter.

$$\begin{aligned} T_a\frac{\mathrm{d}I_a}{\mathrm{d}t} &= -I_a - \Phi\Omega + V \\ T_m\frac{\mathrm{d}\Omega}{\mathrm{d}t} &= \Phi I_a - K_0\Omega - m_L. \end{aligned} \tag{75}$$

In the more general case however V_f is not considered constant and this scheme can also be used for armature and field weakening control of the separately excited Dc motor. Moreover, if the motor characteristics are not exactly known we may consider that the nonlinear model is unknown and therefore its control can be accomplished using the proposed neuro-fuzzy approach. In this case the regulation problem of a DC motor is translated as follows: Find a state feedback to force the angular velocity Ω and the armature current I_a to go to zero, while the magnetic flux varies.

We first assume that the system is described, within a degree of accuracy, by a 2nd order nonlinear neuro-fuzzy system of the form (34), where $x_1 = I_a$ and $x_2 = \Omega$. So, the number of states and inputs is $n = 2$, the number of fuzzy output partitions of each f_i is $m = 6$ and the depth of high order sigmoid terms $k = 5$. In this case $s_i(x)$ assume high order connection up to the second order. The number of fuzzy partitions of each g_{ii} is selected to be $m = 3$ using only first order sigmoid term.

$$\dot{\hat{x}}_1 = -a_1\hat{x}_1 + X^1W^{*1}S(x) + {}^1X^{11}W^{*1}s(x_1)u_1 \tag{76}$$

$$\dot{\hat{x}}_2 = -a_2\hat{x}_2 + X^2W^{*2}S(x) + {}^1X^{21}W^{*2}s(x_2)u_2 \tag{77}$$

or in a more detailed form according to the simulation parameters given above

$$\begin{aligned} \dot{x}_1 =& -a_1\hat{x}_1 + \bar{f}_1^1\left(W_{1,1}^{1*}s_1(x) + \cdots + W_{1,5}^{1*}s_5(x)\right) + \cdots \\ & \cdots + \bar{f}_6^1\left(W_{6,1}^{1*}s_1(x) + \cdots + W_{6,5}^{1*}s_5(x)\right) + \left(\bar{g}_1^{1,11}W_{1,1}^{1*} + \cdots + \bar{g}_3^{1,11}W_{3,1}^{1*}\right)s(x_1)u_1 \\ \dot{x}_2 =& -a_2x_2 + \bar{f}_1^2\left(W_{1,1}^{2*}s_1(x) + \cdots + W_{1,5}^{2*}s_5(x)\right) + \cdots + \bar{f}_6^2\left(W_{6,1}^{2*}s_1(x) + \cdots \right. \\ & \left. \cdots + W_{6,5}^{2*}s_5(x)\right) + \left(\bar{g}_1^{2,21}W_{1,1}^{2*} + \cdots + \bar{g}_3^{2,21}W_{3,1}^{2*}\right)s(x_2)u_2. \end{aligned}$$

However, in the simulations carried out, the actual system is simulated by using the complete set of equations (74). The produced control law (54) is applied on this

Table 3. Parameter values for the DC motor.

Parameter	Value
$1/T_a$	$148.88\,\mathrm{sec}^{-1}$
$1/T_m$	$42.91\,\mathrm{sec}^{-1}$
K_0/T_m	$0.0129\ N\cdot\mathrm{m/rad}$
T_f	31.88 sec
m_L	0.0
a	2.6
β	1.6

Table 4. Initial variable values for the DC motor.

Parameter	Value
I_a	$0.3\,p\cdot u$
Ω	$0.3\,p\cdot u$
Φ	$0.98\,p\cdot u$
W^i	0
${}^1W^i$	1

system, which in turn produces states x_1, x_2, which are in the sequel used in the updating laws of the controller's weights. In our example control laws are having the form

$$u_1 = -\frac{\bar{f}_1^1\left(W_{1,1}^1 s_1(x) + \cdots + W_{1,5}^1 s_5(x)\right) + \cdots + \bar{f}_6^1\left(W_{6,1}^1 s_1(x) + \cdots + W_{6,5}^1 s_5(x)\right)}{\left(\bar{g}_1^{1,1}{}^1W_{1,1}^1 + \cdots + \bar{g}_3^{1,1}{}^1W_{3,1}^1\right) s(x_1)}$$

$$u_2 = -\frac{\bar{f}_1^2\left(W_{1,1}^2 s_1(x) + \cdots + W_{1,5}^2 s_5(x)\right) + \cdots + \bar{f}_6^2\left(W_{6,1}^2 s_1(x) + \cdots + W_{6,5}^2 s_5(x)\right)}{\left(\bar{g}_1^{2,2}{}^1W_{1,1}^2 + \cdots + \bar{g}_3^{2,2}{}^1W_{3,1}^2\right) s(x_2)}.$$

We simulated a 1KW DC motor with parameter values that can be seen in Table 3.

In the control phase, we use the parameters $a_i = 0.2$ and the range of partitions $f_1[-182.5667, 0]$, $f_2[-19.3627, 30.0566]$, $g_{11}[148, 150]$ and $g_{22}[42, 44]$. The initial values of all variables can be seen in Table 4, where p.u (per unit) denotes relevance to the nominal value of the variable. $W^i = 0$ and ${}^1W^i = 1$ denote that weight matrices W^i and ${}^1W^i$ have initial values $0's$ and $1's$ respectively. Figures 10 and 11 show the convergence of states x_1 and x_2 to zero.

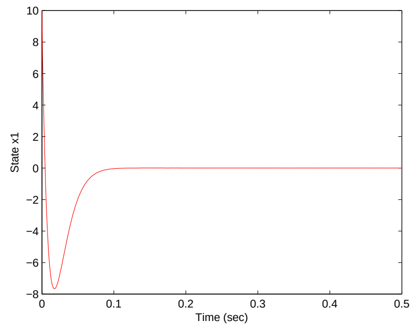

Fig. 10. Convergence of x_1 to zero for the Fuzzy-RHONN Model without dynamic uncertainties.

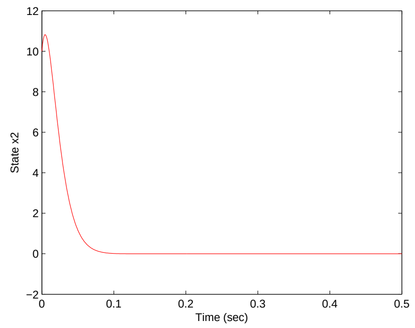

Fig. 11. Convergence of x_2 to zero for the Fuzzy-RHONN Model without dynamic uncertainties.

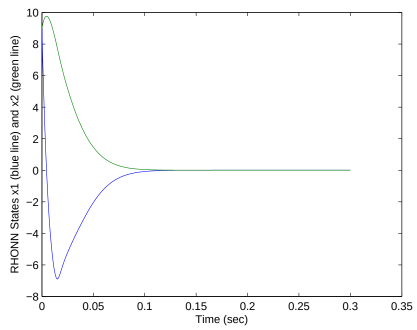

Fig. 12. Convergence of x_1 and x_2 to zero for RHONN Model.

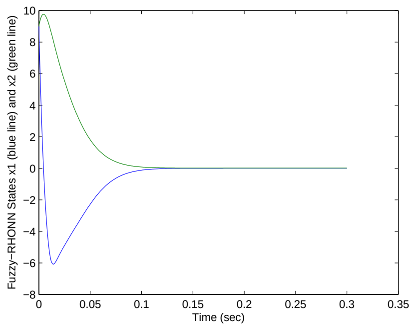

Fig. 13. Convergence of x_2 and x_2 to zero for Fuzzy-RHONN Model.

5.3. Comparison between RHONN and Fuzzy-RHONN Models for controlling a DC Motor

As concerning comparison abilities about controlling a DC Motor, Figures 12 and 13 give the evolution of the states x_1 and x_2, which are the angular velocity and armature current of the RHONN and Fuzzy-RHONN models, with time respectively. During the simulations we observe that the RHONN Model converge to zero despairingly slow (after 8 minutes and 32 seconds for 0.3 seconds simulation time) according to the time we measured with a computer clock, while the proposed adaptive control algorithm converge to zero very fast (after 2,66 seconds for the same simulation time) following the same steps. This shows that our method is superior when it comes to real time problems. Hence applications such as plane control, ESP automobile control etc. can be handled out by our method very efficiently. Specially, a huge difference between the two approximations can be seen in Figures 14 and 15 where we can observe that the evolution of the control input u_1 for the RHONN Model has been exploded while the Fuzzy-RHONN has smooth development, a fact that shows the superiority of our neuro-fuzzy method. Our method thus can be used in several critical engineering applications.

5.4. Direct control of Lorenz system with parametric uncertainties

The Lorenz system was derived to model the two-dimensional convection flow of a fluid layer heated from below and cooled from above. The model represents the Earth's atmosphere heated by the ground's absorption of sunlight and losing heat into space. It can be described by the following dynamical equations

$$\begin{aligned} \dot{x}_1 &= \sigma(x_2 - x_1) \\ \dot{x}_2 &= \rho x_1 - x_2 - x_1 x_3 \\ \dot{x}_3 &= -\beta x_3 + x_1 x_2 \end{aligned} \tag{78}$$

where x_1, x_2 and x_3 represent measures of fluid velocity, horizontal and vertical temperature variations, correspondingly. The parameters σ, ρ and β are positive which

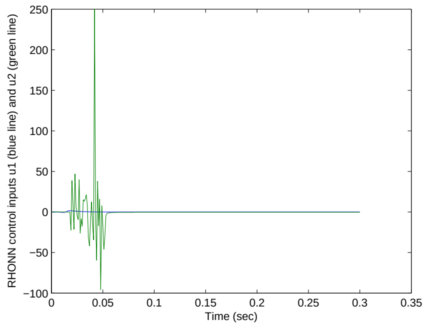

Fig. 14. Evolution of u_1 and u_2 for RHONN Model.

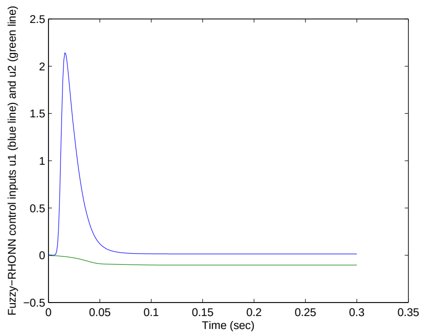

Fig. 15. Evolution of u_1 and u_2 for Fuzzy-RHONN Model.

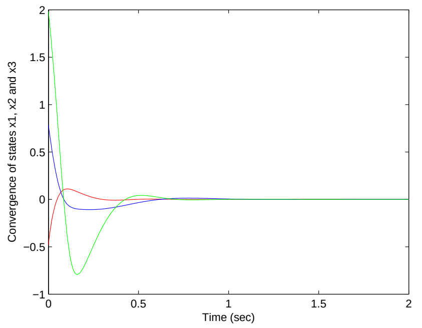

Fig. 16. Convergence of states x_1 (red line), x_2 (blue line) and x_3 (green line) to zero for the Lorenz model.

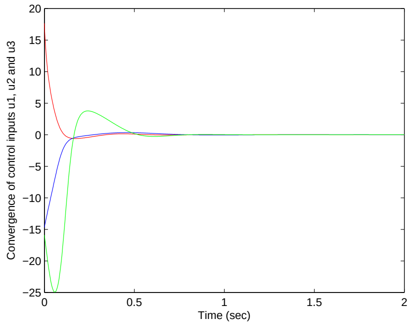

Fig. 17. Evolution of control inputs u_1 (red line), u_2 (blue line) and u_3 (green line) for the Lorenz model.

represent the Prandtl number, Rayleigh number and geometric factor, correspondingly. Selecting $\sigma = 10$, $\rho = 28$ and $\beta = 8/3$ the system presents three unstable equilibrium points and the system trajectory wanders forever near a strange invariant set called strange attractor presenting thus a chaotic behavior [55].

However, the Lorenz system including control inputs can be expressed as [55]

$$\begin{aligned} \dot{x}_1 &= \sigma(x_2 - x_1) + u_1 \\ \dot{x}_2 &= \rho x_1 - x_2 - x_1 x_3 + u_2 \\ \dot{x}_3 &= -\beta x_3 + x_1 x_2 + u_3. \end{aligned} \tag{79}$$

The control objective is to derive appropriate state feedback control law to regulate the system to one of its equilibria, which is $(0, 0, 0)$. In particular, we consider that (79) has the following initial condition

$$x_0 = [-0.5, 0.8, 2]^T.$$

The main parameters for the control law (66) and the learning laws (50), (60) are selected as

$$\begin{aligned} A &= \mathrm{diag}(20, 10, 40) \\ K &= \mathrm{diag}(21, 38, 40). \end{aligned}$$

The parameters of the sigmoidals that have been used are $\alpha_1 = \alpha_2 = \alpha_3 = 1$, $\beta_1 = \beta_2 = \beta_3 = 1$ and $\gamma_1 = \gamma_2 = \gamma_3 = 0$.

In the sequel, we first assume that the system is described, within a degree of accuracy, by a neuro-fuzzy system of the form (34) with the number of states being $n = 3$, the number of fuzzy partitions being $p = 5$ and the depth of high order sigmoid terms $k = 9$. In this case $s_i(x)$ assume high order connection up to the second order. Figure 16 shows the convergence of states x_1, x_2 and x_3 to zero exponentially fast. Also, Figure 17 shows the smooth evolution of the control inputs.

6. CONCLUSIONS

A direct adaptive control scheme was considered in this paper, aiming at the regulation of non linear unknown plants. The approach is based on a new Neuro-Fuzzy Dynamical Systems definition, which uses the concept of Fuzzy Dynamical Systems (FDS) operating in conjunction with High Order Neural Network Functions (F-HONNFs). Since the plant is considered unknown, we first propose its approximation by a special form of an affine in the control fuzzy dynamical system (FDS) and in the sequel the fuzzy rules are approximated by appropriate HONNFs. The fuzzy-recurrent high order neural networks are used as models of the unknown plant, practically transforming the original unknown system into a F-RHONN model which is of known structure, but contains a number of unknown constant value parameters known as synaptic weights. The proposed scheme does not require a-priori experts' information on the number and type of input variable membership functions making it less vulnerable to initial design assumptions, is computationally extremely fast and thus can be used in several critical and real-time engineering applications. Weight updating laws for the involved HONNFs are provided, which guarantee that the system states reach zero exponentially fast, while keeping all signals in the closed loop bounded. A novel method of parameter hopping developed for the first time by the authors, assures the existence of the control signal and is incorporated in the weight updating law. Simulations illustrate the potency of the method both in approximation abilities as well as in controlling unknown nonlinear plants such as the DC-Motor and Lorentz system. Compared to simple RHONN direct control, the proposed method proves to be superior.

(Received July 11, 2008.)

REFERENCES

[1] Y. S. Boutalis, D. C. Theodoridis, and M. A. Christodoulou: A new Neuro FDS definition for indirect adaptive control of unknown nonlinear systems using a method of parameter hopping. IEEE Trans. Neural Networks, to appear 2009.

[2] F. C. Chen and C. C. Liu: Adaptively controlling nonlinear continuous time systems using multilayer neural networks. IEEE Trans. Automat. Control *39* (1994), 1306–1310.

[3] B. S. Chen, C. H. Lee, and Y. C. Chang: Tracking design of uncertain nonlinear SISO systems: Adaptive fuzzy approach. IEEE Trans. Fuzzy Systems *4* (1996), 32–43.

[4] K. B. Cho and B. H. Wang: Radial basis function based adaptive fuzzy systems and their applications to system identification and prediction. Fuzzy Sets and Systems *83* (1996), 325–339.

[5] M. A. Christodoulou, D. C. Theodoridis, and Y. S. Boutalis: Building optimal fuzzy dynamical systems description based on recurrent neural network approximation. In: Proc. Internat. Conference of Networked Distributed Systems for Intelligent Sensing and Control, Kalamata 2007.

[6] S. L. Chui: Fuzzy model identification based on cluster estimation. J. Intelligent and Fuzzy Systems *2* (1994), 267–284.

[7] F. P. Da and W. S. Song: Fuzzy neural networks for direct adaptive control. IEEE Trans. Industrial Electronics *50* (2003), 507–513.

[8] Y. Diao and K. M. Passino: Adaptive neural/fuzzy control for interpolated nonlinear systems. IEEE Trans. Fuzzy Systems *10* (2002), 583–595.

[9] N. Golea, A. Golea, and K. Benmahammed: Stable indirect fuzzy adaptive control. Fuzzy Sets and Systems *137* (2003), 353–366.

[10] Y. Gao and M. J. Er: Online adaptive fuzzy neural identification and control of a class of MIMO nonlinear systems. IEEE Trans. Fuzzy Systems *4* (2003), 462–477.

[11] H. L. Hiew and C. P. Tsang: Fuzzy chaos and recursive partitioning. In: Fuzzy Engineering (B. Kosko, ed.), Prentice-Hall, Englewood Cliffs, N. J. 1997.

[12] K. Hornic, M. Stinchcombe, and H. White: Multilayer feedforward networks are universal approximators. Neural Networks *2* (1989), 359–366.

[13] M. Hojati and S. Gazor: Hybrid adaptive fuzzy identification and control of nonlinear systems. IEEE Trans. Fuzzy Systems *10* (2002), 198–210.

[14] P. A. Ioannou and K. S. Tsakalis: A robust direct adaptive controller. IEEE Trans. Automat. Control *AC-31* (1986), 1033–1043.

[15] P. A. Ioannou and A. Datta: Robust adaptive control: A unified approach. Proc. IEEE *79* (1991), 1735–1768.

[16] P. A. Ioannou and J. Sun: Robust Adaptive Control. Prentice-Hall, Englewood Cliffs, N. J. 1996.

[17] P. Ioannou and B. Fidan: Adaptive control tutorial. SIAM: Advances in Design and Control Series, 2006.

[18] A. Isidori: Nonlinear Control System. Third edition. Springer-Verlag, Berlin – Heigelberg – New York 1995.

[19] B. Jakubczyk: Feedback linearization of discrete time system. System Control Lett. *9* (1987), 411–416.

[20] J. S. R. Jang: ANFIS: Adaptive-network-based fuzzy inference system. IEEE Trans. Systems Man Cybernet. *23* (1993), 665–684.

[21] C. F. Juang and C. T. Lin: An on-line self-constructing neural fuzzy inference network and its applications. IEEE Trans. Fuzzy Systems *6* (1998), 12–32.

[22] P. V. Kokotovic, H. K. Khalil, and J. O'Reilly: Singular Perturbation Methods in Control: Analysis and Design. Academic Press, New York 1986.

[23] E. B. Kosmatopoulos and M. A. Christodoulou: Structural properties of gradient recurrent high order neural networks. IEEE Trans. Circuits and Systems, Part II, Express Briefs *42* (1995), 592–603.

[24] E. B. Kosmatopoulos and M. A. Christodoulou: Recurrent neural networks for approximation of fuzzy dynamical systems. Internat. J. Intelligent Control and Systems *1* (1996), 223–233.

[25] S. Labiod, M. S. Boucherit, and T. M. Guerra: Adaptive fuzzy control of a class of MIMO nonlinear systems. Fuzzy Sets and Systems *151* (2005), 59–77.

[26] C. Lee and C. Teng: Identification and control of dynamic systems using recurrent fuzzy neural networks. IEEE Trans. Fuzzy Systems *8* (2000), 349–366.

[27] W. Leonhard: Control of Electrical Drives. Springer-Verlag, Berlin 1985.

[28] Y. Leu, W. Wang, and T. Lee: Observer-based direct adaptive fuzzy-neural control for nonaffine nonlinear systems. IEEE Trans. Neural Networks *16* (2005), 853–861.

[29] R. P. Li and M. Mukaidono: A new approach to rule learning based on fusion of fuzzy logic and neural networks. IEICE Trans. Fuzzy Systems *E78-d* (1995), 1509–1514.

[30] Y. H. Lin and G. A. Cunningham: A new approach to fuzzy-neural system modelling. IEEE Trans. Fuzzy Systems *3* (1995), 190–197.

[31] C. T. Lin: A neural fuzzy control system with structure and parameter learning. Fuzzy Sets and Systems *70* (1995), 183–212.

[32] T. Lina, C. Wang, and H. Liub: Observer-based indirect adaptive fuzzy-neural tracking control for nonlinear SISO systems using VSS and H_1 approaches. Fuzzy Sets and Systems *143* (2004), 211–232.

[33] S. Mitra and Y. Hayashi: Neuro-fuzzy rule generation: survey in soft computing framework. IEEE Trans. Neural Networks *11* (2000), 748–768.

[34] S. Monaco and D. Normand-Cyrot: Minimum-phase nonlinear discretetime systems and feedback stabilization. In: Proc. IEEE Conference Decision Control, Los Angeles 1987, pp. 979–986.

[35] R. V. Monopoli: Model reference adaptive control with an augmented error signal. IEEE Trans. Automat. Control *AC-19* (1974), 474–484.

[36] S. P. Moustakidis, G. A. Rovithakis, and J. B. Theocharis: An adaptive neuro-fuzzy tracking control for multi-input nonlinear dynamic systems. Automatica, to appear 2008.

[37] K. S. Narendra and A. M. Annaswamy: Stable Adaptive Systems. Prentice-Hall, Englewood Cliffs, N. J. 1989.

[38] K. S. Narendra and K. Parthasarathy: Identification and control of dynamical systems using neural networks. IEEE Trans. Neural Networks *1* (1990), 4–27.

[39] J. H. Park, S. H. Huh, S. J. Seo, and G. T. Park: Direct adaptive controller for nonaffine nonlinear systems using self-structuring neural networks. IEEE Trans. Neural Networks *16* (2005), 414–422.

[40] K. Passino and S. Yurkovich: Fuzzy Control. Addison Wesley Longman, 1998.

[41] P. Phan and T. J. Gale: Direct adaptive fuzzy control with a self-structuring algorithm. Fuzzy Sets and Systems *159* (2008), 871–899.

[42] M. M. Polycarpou and M. J. Mears: Stable adaptive tracking of uncertain systems using nonlinearly parameterized online approximators. Internat. J. Control *70* (1998), 363–384.

[43] G. A. Rovithakis and M. A. Christodoulou: Adaptive control of unknown plants using dynamical neural networks. IEEE Trans. Systems Man Cybernet. *24* (1994), 400–412.

[44] G. A. Rovithakis and M. A. Christodoulou: Direct adaptive regulation of unknown nonlinear dynamical systems via dynamic neural networks. IEEE Trans. Systems Man Cybernet. *25* (1995), 1578–1594.

[45] G. A. Rovithakis: Performance of a neural adaptive tracking controller for multi-input nonlinear dynamical systems in the presence of additive and multiplicative external disturbances. IEEE Trans. Systems Man Cybernet. Part A *30* (2000), 720–730.

[46] G. A. Rovithakis and M. A. Christodoulou: Adaptive control with recurrent high order neural networks (Theory and industrial applications). In: Advances in Industrial Control (M. A. Grimble and M. A. Johnson, eds.), Springer Verlag, London Limited 2000.

[47] S. S. Sastry and M. Bodson: Adaptive Control: Stability, Convergence, and Robustness. Prentice-Hall, Englewood Cliffs, N. J. 1989.

[48] S. S. Sastry and A. Isidori: Adaptive control of linearizable systems. IEEE Trans. Automat. Control *34* (1989), 1123–1131.

[49] J. J. E. Slontine and W. Li: Applied Nonlinear Control. Prentice-Hall, Englewood Cliffs, N. J. 1991.

[50] J. T. Spooner and K. M. Passino: Stable adaptive control using fuzzy systems and neural networks. IEEE Trans. Fuzzy Systems *4* (1996), 339–359.

[51] D. C. Theodoridis, M. A. Christodoulou, and Y. S. Boutalis: Indirect adaptive neuro-fuzzy control based on high order neural network function approximators. In: Proc. 16th Mediterranean Conference on Control and Automation – MED08, Ajaccio 2008, pp. 386–393.

[52] L. Wang: Adaptive Fuzzy Systems and Control. Prentice Hall, Englewood Cliffs, N. J. 1994.

[53] C. Wang, H. Liu, and T. Lin: Direct adaptive fuzzy-neural control with state observer and supervisory controller for unknown nonlinear dynamical systems. IEEE Trans. Fuzzy Systems *10* (2002), 39–49.

[54] S. Wu and M. Joo Er: Dynamic fuzzy neural networks. A novel approach to function approximation. IEEE Trans. Systems, Man Cybernet, Part B *30* (2000), 358–364.

[55] T. H. Yeap and N. U. Ahmed: Feedback control of chaotic systems. Dynamic and Control *4* (1994), 97–114.

[56] W. Yu and X. Li: Fuzzy neural identification by online clustering with application on crude oil blending. In: Internat. Conference on Fuzzy Systems, Vancouver 2006.

Dimitris Theodoridis, Department of Electrical and Computer Engineering, Democritus University of Thrace, Xanthi 67100. Greece.
e-mail: dtheodo@ee.duth.gr

Yiannis Boutalis, Department of Electrical and Computer Engineering, Democritus University of Thrace, Xanthi 67100, Greece and Department of Electrical, Electronic and Communication Engineering, Chair of Automatic Control, University of Erlangen, Nuremberg 91058, Germany.
e-mail: ybout@ee.duth.gr

Manolis Christodoulou, Department of Electronic and Computer Engineering, Technical University of Crete, Chania, Crete 73100. Greece.
e-mail: manolis@ece.tuc.gr