SYMPOSIOTECNICO EN HONOR A BILL WOLOVICH - UNA FIESTA PARA WOLOVICH

Symposium to honor Bill Wolovich-Sunday December 7, 2008 Cancun Mexico

New Trends in Neuro-Fuzzy Adaptive Control Systems

Authors: Manolis A. Christodoulou

Technical University of Crete Greece

and Yiannis S. Boutalis

Democritus University of Thrace Greece and University of Erlangen – Nuremberg Chair of Automatic Control Systems, Germany

Outline of the presentation

- Preliminaries
- Fuzzy approaches of Mamdani and TSK type
- Fuzzy representation with indicator functions
- Approximation of indicator functions with HONNF’s
- Neuro-Fuzzy modelling for affine in the control systems
- Weight updating laws for identification and indirect control
- Introduction to weight hopping
- Some simulations
- Conclusions

Preliminaries

- Computational Intelligence techniques
- Neural Networks - RHONNs
- Fuzzy Systems and their Functional Representation

Computational Intelligence and the use of “Intelligent” Techniques

We are using the so-called intelligent techniques when the mathematical model of the process is not sufficient or it does not exist. Instead, there are operating data or human knowledge expressed with linguistic rules.

Computational Intelligence comprises the following research areas

- **Expert Systems**
- **Fuzzy Systems**
- **Neural Network Systems**
- **Fuzzy – Neuro (or Neuro-Fuzzy) Systems**
- **Evolutionary Computation**

Modern History on Computational Intelligence

DARPA recently awarded $4.9M to IBM and 5 other Universities through the SyNAPSE Program for the buildup of BRAIN-LIKE computers

•Production for the first time of an electronic system that behaves as the simulations do

•180 degree shift in perspective: Seek algorithm first, problem second.

•Difference between von Neumann and current machine: the current does not separate memory and computation

•Time ripe now: Neuroscience mature, supercomputing methods available, nanotechnology allowing deployment of enough micro-operations and synapses put in small surface areas.

•If the project succeeds: Birth to novel cognitive systems, ubiquitously deployed computers, imbued with new intelligence, respond in context-dependent way, learn over time, action and cognition in complex real-world environments.

•Goal: inevitable but unpredictable

Past History on Computational Intelligence

When the linear separability problem existed back in 1980’s still prominent scientists were insisting on doing research in this area, most of the times underfunded and frustrated, like

•David Rumelhart

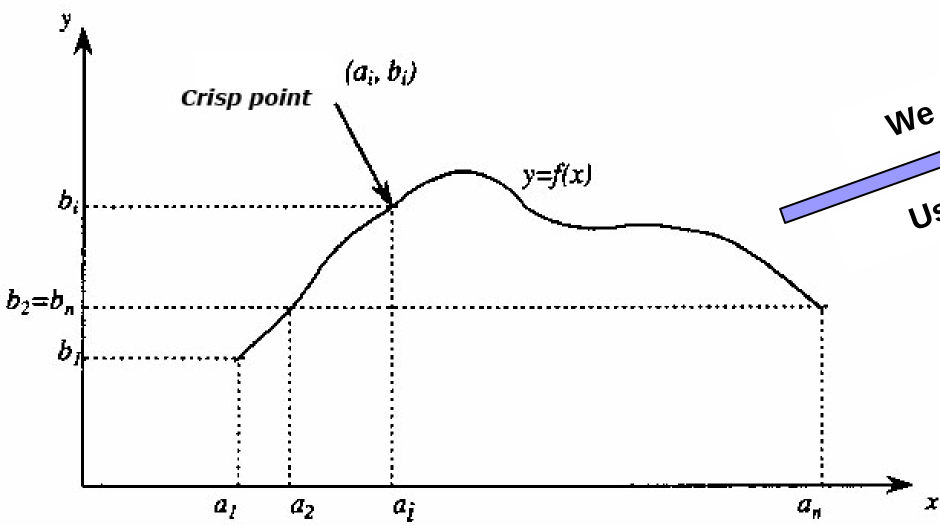
•Geoffrey Hinton

•Roland Williams

•Bob Newcomb

And many other pioneers, just a few to name of

When the mathematical model is imprecise or does not exist



We describe it

Using the rules

If x is a1 Then y is b1
If x is a2 Then y is b2
…
If x is ai Then y is bi
…
If x is an Then y is bn

One Significant Observation

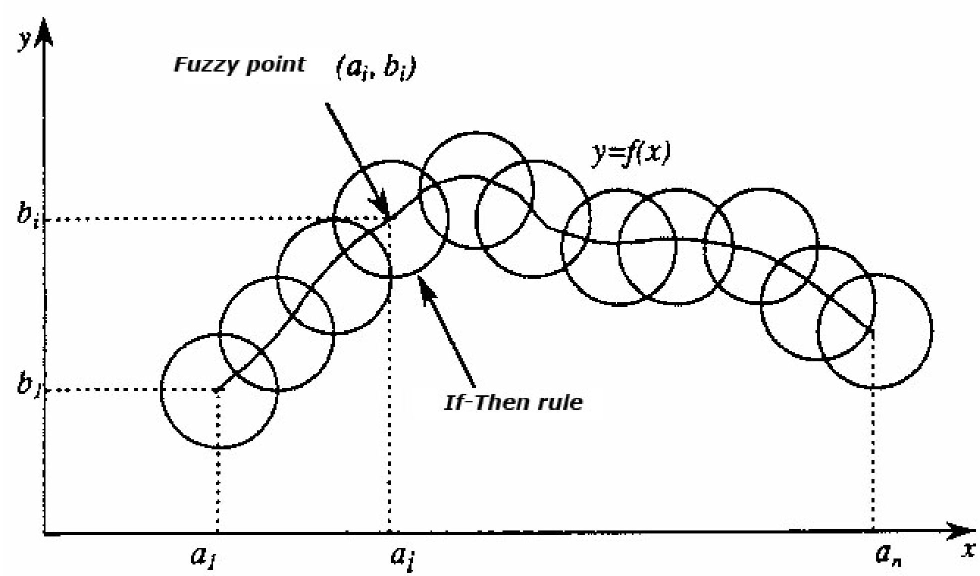
One could describe the function with two alternative ways

1. Based on I/O data using Neural Networks
2. Based on Fuzzy rules building Fuzzy Linguistic Description

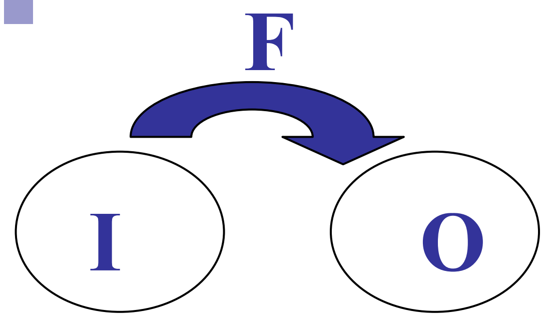
Building

Linguistic

Descriptions



What does a NN do?



In reality it is an emulator of the linear or non-linear mapping between inputs and outputs

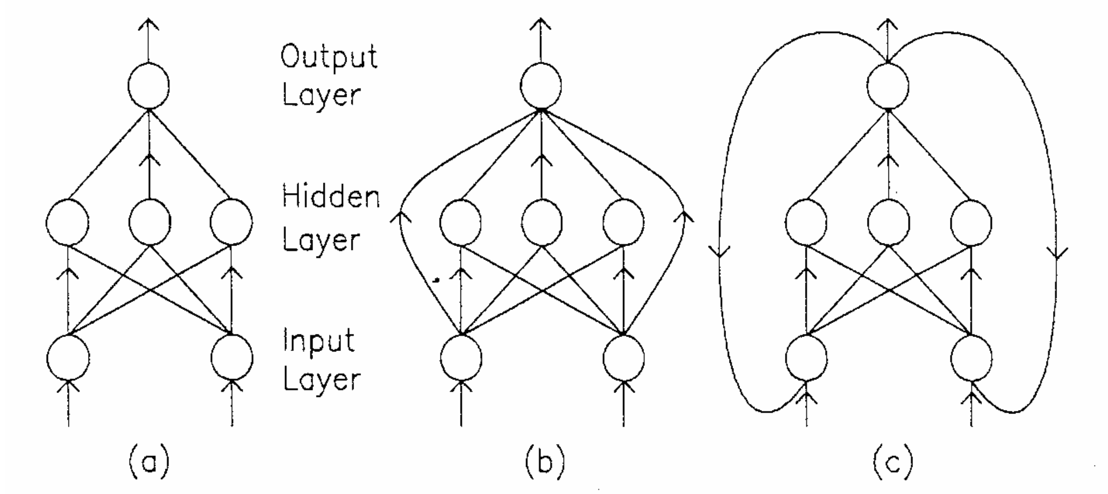
Where are they used ?

In many fields!!!

- **Modelling / Parameter Estimation**
- **Filtering (Signal / Image Processing)**
- **Pattern Recognition / Classification / Identification**
- **Forcasting / Prediction**
- **Control**

Topologies and Training of NNs

$$\hat{f}(x) = \hat{W}^T \sigma(\hat{V}^T x)$$



Partially – connected Feedforward

Fully - connected Feedforward

Feedback / Recurrent Dynamical

<u>Training</u>

In conventional FF NNs the <u>backpropagation</u> algorithm and its varieties are well established

In 1-layer recurrent NNs <u>the linearity in the parameters property</u> is exploited to train the NN to reach its goal exponentially fast

Advantage of NN Control over Conventional Adaptive Control

Let the dynamic equations of rigid-link robot arms.

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{V}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{F}_{\mathbf{v}}(\dot{\mathbf{q}}) + \mathbf{F}_{\mathbf{d}}(\dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) = \boldsymbol{\tau} \xrightarrow{\text{LIP}} \boldsymbol{\tau} = \mathbf{W}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})\boldsymbol{\varphi}$$

Conventional Adaptive Control

$$\hat{f}(x) = R(x)\hat{\phi}$$

Linear in respect to the tunable parameters (masses and friction coefs). The regression matrix *R* depends on *f(x)* and must be recomputed for each different robot arm [*f(x)]*.

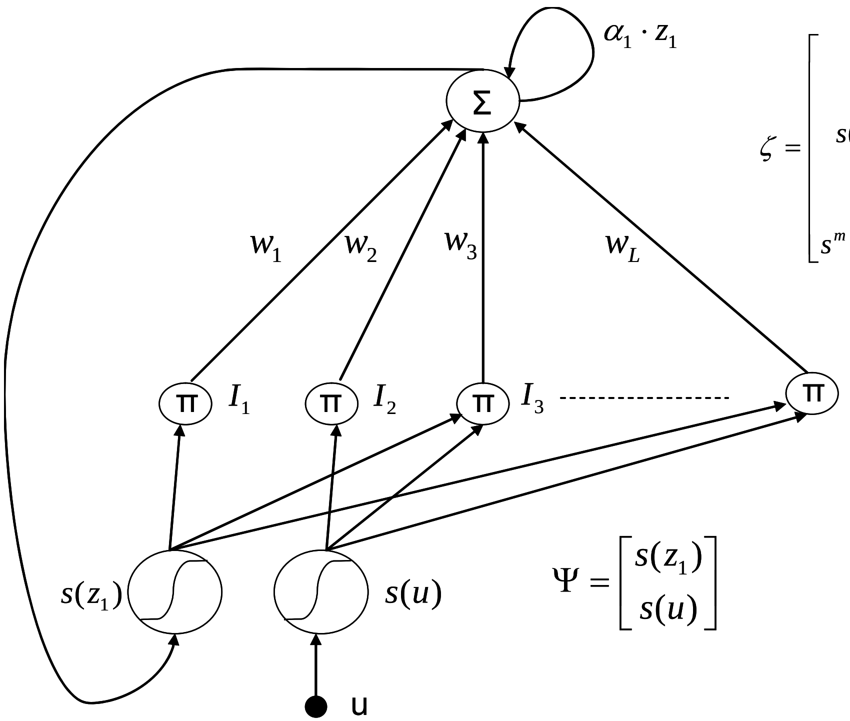
Neural Network Adaptive Control

$$\hat{f}(x) = \hat{W}^T \sigma(x)$$

Linear in respect to the tunable parameters (neural weights). The same basis functions *σ(x) suffices for every f(x).*

A Neural network controller is much more powerful than conventional adaptive controller. The former, is a universal controller for all rigid-link robot arms.

Recurrent High Order NNs

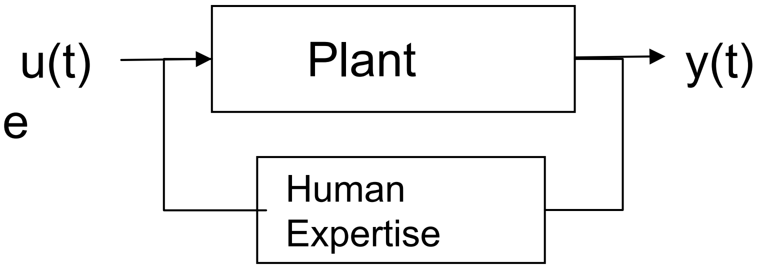


$$\zeta = \begin{bmatrix} s(u) \\ s(z_1) \\ s(u) \cdot s(z_1) \\ \cdot \\ \cdot \\ s^m(u) \cdot s^n(z_1) \end{bmatrix}$$

$$\Psi = \begin{bmatrix} s(z_1) \\ s(u) \end{bmatrix}$$

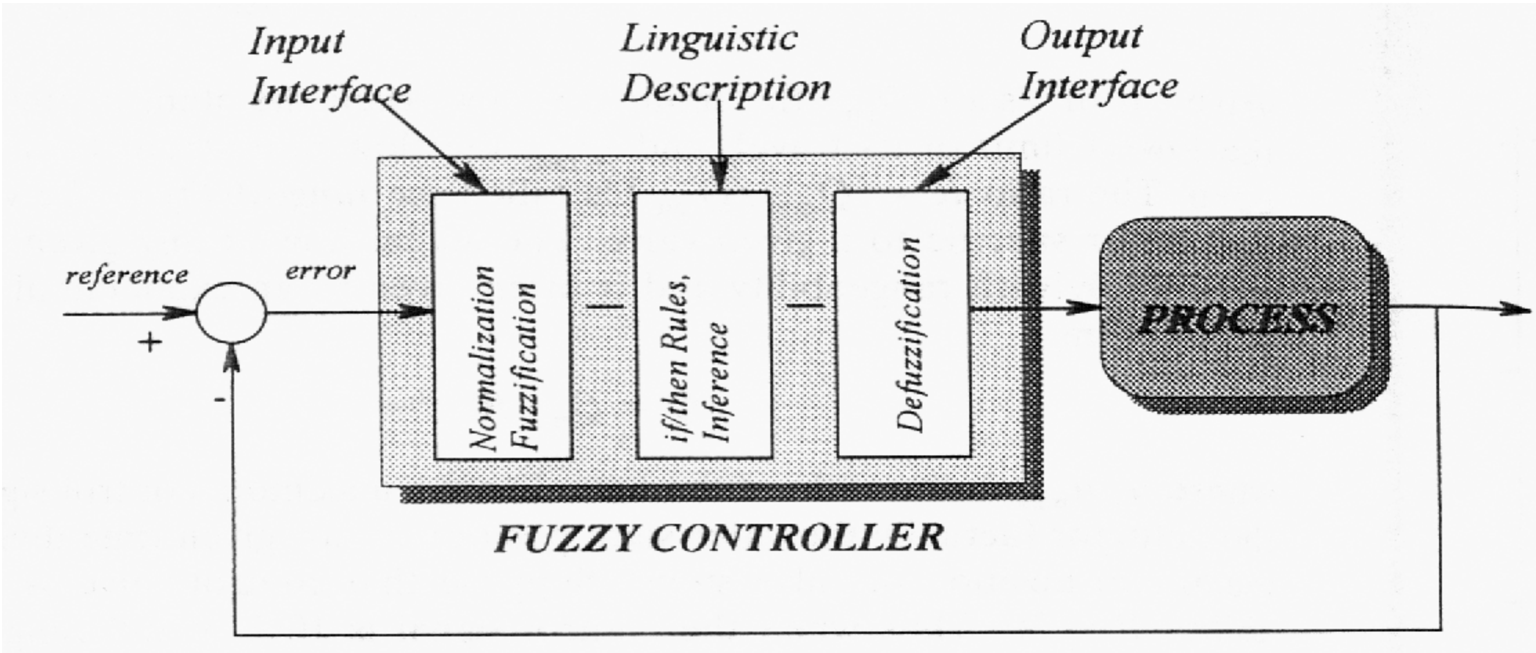
- Recurrent Higher order Neural Networks are very suitable for the approximation of the dynamics of non linear systems
- They constitute the so-called Π – Σ networks. In the first layer(s) there are the inputs, the feedback and higher order products of them. In the second layer the next state value is estimated by weighting and summing the elements from the first layer.
- Their functional representation is linear with respect to the tunable parameters wij, therefore their estimate can be made using existing knowledge from adaptive estimation theory
- They can be used as the first part of a direct or indirect controller scheme

Fuzzy Controllers



In fuzzy control the controller incorporates the experience of the human manipulator of a plant

A typical use of a fuzzy controller and its components



Functional Representation of Fuzzy Systems

Depending on the type of the Fuzzy system (Mamdani – or Tagaki-Sugeno) and on the implication operator used in the inference procedure a fuzzy system can be represented in a functional form and can be used as a function approximator.

$$f(x)=\frac{\sum_{l=1}^{M} y^{l}\left(\prod_{i=1}^{n}\mu_{F_i^l}(x)\right)}{\sum_{l=1}^{M}\left(\prod_{i=1}^{n}\mu_{F_i^l}(x)\right)}$$

Typical functional representation of a Mamdani-type fuzzy system with the Larsen's product implication operator

Gausian M.F

$$f(x)=\frac{\sum_{l=1}^{M} y^{l}\left(\prod_{i=1}^{n}\exp\left(-\left(\frac{x-x_i^l}{\sigma_i^l}\right)^2\right)\right)}{\sum_{l=1}^{M}\left(\prod_{i=1}^{n}\exp\left(-\left(\frac{x-x_i^l}{\sigma_i^l}\right)^2\right)\right)}$$

Tunable parameters enabling the fuzzy functional representation to be an adaptive system approximator which can be trained Based on data of operation

Our Neuro-Fuzzy Approach

☑ We consider the indirect adaptive control of unknown nonlinear dynamical systems. The proposed scheme uses the concept of Adaptive Fuzzy Systems (AFS) operating in conjunction with High Order Neural Network Functions (F-HONNFs).

☑ 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 identification scheme leads up to a Recurrent High Order Neural Network, which however takes into account the centers of the fuzzy output partitions of the initial FDS.

☑ Weight updating laws for the involved HONNFs are provided, which guarantee that both the identification error and the system states reach zero exponentially fast, while keeping all signals in the closed loop bounded.

☑ The applicability of the method is tested on a DC Motor system where it is shown that by following the proposed procedure one can obtain asymptotic regulation.

A. Fuzzy approaches of Mamdani and TSK type

- Suppose that the the nonlinear function $f(x)$ is approximated by an adaptive (Mamdani type) fuzzy system which is linear in the adjustable parameters

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

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:

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

where θ_l are adjustable parameters, and $\mu_{F_i^l}$ are given membership functions of the input variables.

- Obviously, the combination of the above equations, are equivalent to the following equation:

$$f(x) = \frac{\sum_{l=1}^{M} \theta_l \left(\prod_{i=1}^{n} \mu_{F_i^l}(x_i) \right)}{\sum_{l=1}^{M} \left(\prod_{i=1}^{n} \mu_{F_i^l}(x_i) \right)}$$

- In Tagaki-Sugeno formulation *f*(*x*) is given by:

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

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 $\xi_l(x)$ as being defined previously.

B. Fuzzy representation with Indicator functions

- $I_{j_1,\dots,j_{n+m}}^{l_1,\dots,l_n}$ denotes the indicator function of the subset $\Omega_{j_1,\dots,j_{n+m}}^{l_1,\dots,l_n}$, that is,

$$\boxed{I_{j_1,\dots,j_{n+m}}^{l_1,\dots,l_n}(x,u) = \begin{cases} \alpha & if\,(x,u) \in \Omega_{j_1,\dots,j_{n+m}}^{l_1,\dots,l_n} \\ 0 & otherwise \end{cases}}$$

Let $\bar{x}_{j_1,\dots,j_{n+m}}^{l_1,\dots,l_n} \in R^n$ be any vector satisfying

$$\boxed{h_i(\bar{x}_{j_1,\dots,j_{n+m}}^{l_1,\dots,l_n}(i)) = l_i}$$

Define now the following dynamical system:

$$\boxed{\chi^{t+1} = \sum \bar{x}_{j_1,\dots,j_{n+m}}^{l_1,\dots,l_n} \times I_{j_1,\dots,j_{n+m}}^{l_1,\dots,l_n}(\chi^t, u^t)}$$

- The system in above equation is a generator for the FDS ($R_{j_1,\dots,j_{n+m}}^{l_1,\dots,l_n}$).

However, in order the approximation problem to make sense the space $y := \chi \times u$ must be compact.

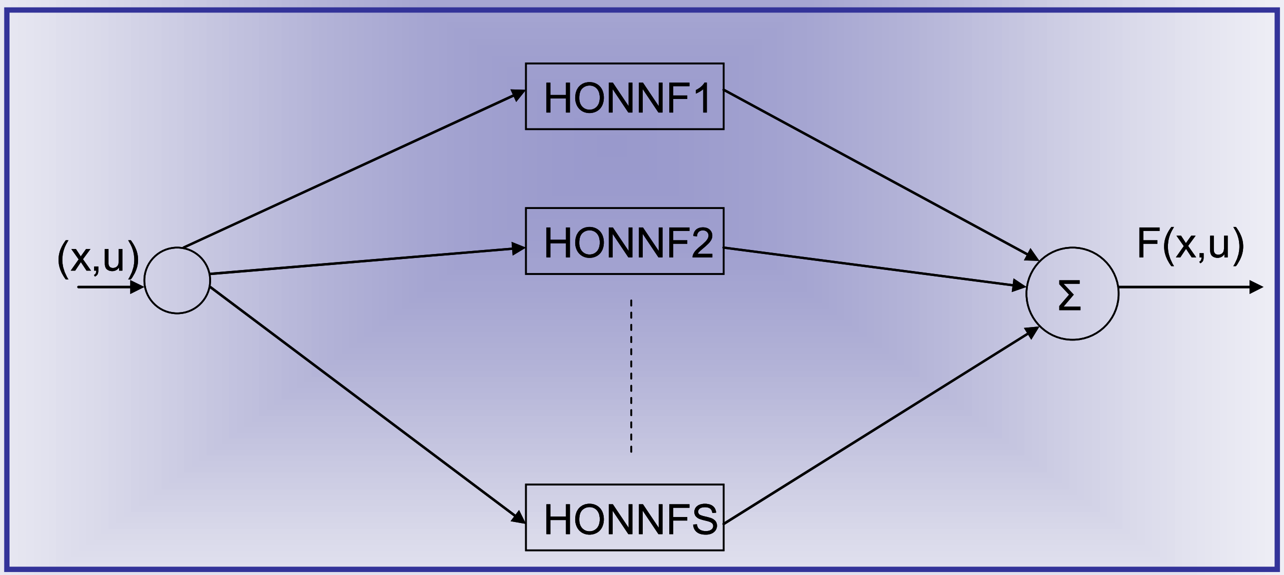
C. Approximation of Indicator functions with HONNF’s

Let us define the following high order neural network functions (HONNFs) that approximate the Indicator functions.

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

Where Φj are elements of:

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



- As it is reported in the literature the HONNF N(·) satisfies the Stone – Weierstrass theorem and therefore can approximate bounded and measurable functions. It is worth noticing that while the Stone – Weierstrass theorem is valid for continuous functions we can extend it by using the Lusin theorem to functions that are discontinuous like indicator functions.

D. Neuro-Fuzzy modelling for affine in the control systems

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

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

where the state $x \in R^n$, is assumed to be completely measured, the control u is in 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 $G = [g_1\, g_2 \quad g_n]$.

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

$$\boxed{f(\chi) = A\chi + \sum \bar{f}_{j_1,\ldots,j_n}^{l_1,\ldots,l_n} \times I_{j_1,\ldots,j_n}^{l_1,\ldots,l_n}(\chi)}$$

$$\boxed{g_i(\chi) = \sum (\bar{g}_i)_{j_1,\ldots,j_n}^{l_1,\ldots,l_n} \times I_{1 j_1,\ldots,j_n}^{l_1,\ldots,l_n}(\chi)}$$

D. Neuro-Fuzzy modelling for affine in the control systems

Every indicator function can be approximated with the help of a suitable HONNF. I, I_1

$$f(\chi) = A\chi + \sum \bar{f}_{j_1,\ldots,j_n}^{l_1,\ldots,l_n} \times N_{j_1,\ldots,j_n}^{l_1,\ldots,l_n}(\chi)$$

$$g_i(\chi) = \sum (\bar{g}_i)_{j_1,\ldots,j_n}^{l_1,\ldots,l_n} \times N_{1\,j_1,\ldots,j_n}^{l_1,\ldots,l_n}(\chi)$$

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 are carried out over the number of the corresponding output partitions.

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

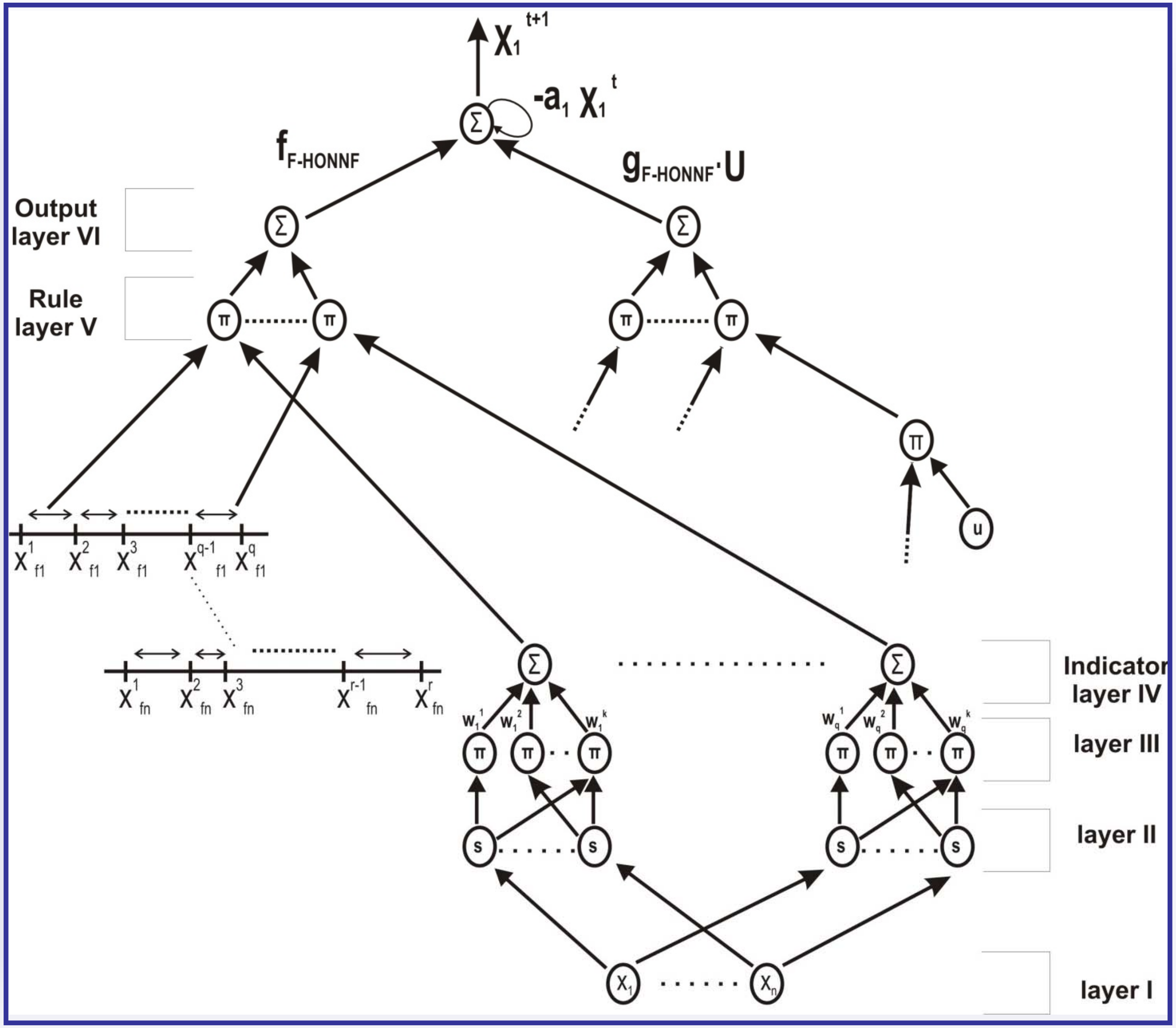
Or in a more compact form:

$$\dot{\hat{\chi}} = A\hat{\chi} + XWS(\chi) + X_1 W_1 S_1(\chi)u$$

Where A is a $n \times n$ stable matrix which for simplicity can be taken to be diagonal as $A = diag[a_1, a_2, ..., a_n]$.X ,$X_1$ are matrices containing the centres of the partitions of every fuzzy output variable of f(x) and g(x) respectively, as it can be seen below. S(x), S_1(x) are matrices containing high order combinations of sigmoid functions of the state χ and W, W_1 are matrices containing respective neural weights.

$$X = \begin{bmatrix} \bar{f}_1^1 & \bar{f}_2^1 & \cdots & \bar{f}_m^1 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & \bar{f}_1^2 & \bar{f}_2^2 & \cdots & \bar{f}_m^2 & 0 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & \bar{f}_1^n & \bar{f}_2^n & \cdots & \bar{f}_m^n \end{bmatrix}$$

$$X_1 = \begin{bmatrix} \bar{g}_1^{1,1} & \bar{g}_2^{1,1} & \cdots & \bar{g}_m^{1,1} & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & \bar{g}_1^{2,2} & \bar{g}_2^{2,2} & \cdots & \bar{g}_m^{2,2} & 0 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & \bar{g}_1^{n,n} & \bar{g}_2^{n,n} & \cdots & \bar{g}_m^{n,n} \end{bmatrix}$$



Configuration of a neuro-fuzzy (F-RHONN) approximator

E. Identification Scheme

We assume the existence of only parameter uncertainty, so, we can take into account that the actual system can be modeled by the following neural form

$$\boxed{\dot{\chi} = A\chi + XW^{*}S(\chi) + X_1 W_1^{*} S_1(\chi)u}$$

Define now, the error between the identifier states and the real states as

$$\boxed{e = \widehat{\chi} - \chi}$$

Then we obtain the error equation

$$\boxed{\dot{e} = Ae + X\tilde{W}S(\chi) + X_1 \tilde{W}_1 S_1(\chi)u}$$

$$\tilde{W} = W - W^{*}$$

$$\tilde{W}_1 = W_1 - W_1^{*}$$

Regarding the identification of W W_1
we are now able to state the following theorem

E.Weight updating laws for identification and indirect control

Consider the identification scheme given by:

$$\boxed{\dot{e} = Ae + X\tilde{W}S(\chi) + X_1\tilde{W}_1S_1(\chi)u}$$

The learning law:

a) For the elements of W^i

$$\dot{w}^i_{jl} = -\overline{f}^i_j p_i e_i s_l(\chi)$$

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

$${}^1\dot{w}^i_{j1} = -\overline{g}^{ii}_j p_i e_i u_i s_i(\chi)$$

with $i = 1,\ldots,n \quad , j = 1,\ldots,m \quad , l = 1,\ldots,k$ guarantees the following properties:

$$\boxed{\begin{array}{c} e, \hat{\chi}, \tilde{W}, \tilde{W}_1 \in L_\infty, \quad e \in L_2 \\ \lim_{t\to\infty} e(t) = 0, \quad \lim_{t\to\infty} \dot{\tilde{W}}(t) = 0, \quad \lim_{t\to\infty} \dot{\tilde{W}}_1(t) = 0 \end{array}}$$

E.Weight updating laws for system identification

The proof is provided by considering the Lyapunov function candidate:

$$\boxed{V(e,\tilde{W},\tilde{W}_1)=\frac{1}{2}\cdot e^T\cdot P\cdot e+\frac{1}{2}\cdot tr\{\tilde{W}^T\cdot\tilde{W}\}+\frac{1}{2}\cdot tr\{\tilde{W}_1^T\cdot\tilde{W}_1\}}$$

And choosing the weight updating law so that $\dot{V}\leq 0$

Control Objective

Our purpose is to find suitable control and learning laws to drive the error and the state to zero, while all other signals in the closed loop remain bounded. Taking u to be equal to

$$\boxed{u=-\left[X_1W_1S_1(\chi)\right]^{-1}XWS(\chi)}$$

We finally obtain $\dot{\widehat{\chi}}=A\widehat{\chi}$

E. Introduction to weight hopping

- However, regarding the control law: $u = -\left[X_1 W_1 S_1(\chi)\right]^{-1} XWS(\chi)$

the existence of $\left[X_1 W_1 S_1(\chi)\right]^{-1}$ has to be assured.

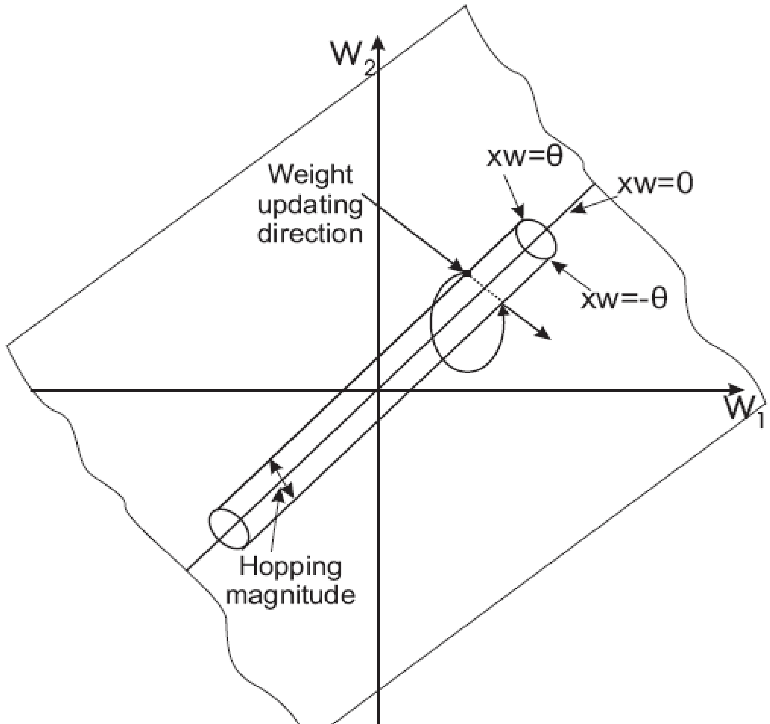
Since $S_1(\chi)$ is diagonal with its element $s_i(\chi) \neq 0$ and X_1, W_1 are block diagonal the existence of the inverse is assured when ${}^1X^i \cdot {}^1W^i \neq 0, \forall i = 1, \ldots n$

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.

$$W_1 = \begin{bmatrix} \begin{matrix} w_1^{11} \\ \vdots \\ w_m^{11} \end{matrix} & 0 & 0 & 0 \\ 0 & \begin{matrix} w_1^{22} \\ \vdots \\ w_m^{22} \end{matrix} & 0 & 0 \\ & & \ddots & \\ 0 & 0 & 0 & \begin{matrix} w_1^{nn} \\ \vdots \\ w_m^{nn} \end{matrix} \end{bmatrix}$$

$$S_1(\chi) = \begin{bmatrix} s_1(\chi) & 0 & \cdots & 0 \\ 0 & s_2(\chi) & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & s_n(\chi) \end{bmatrix}$$

$$X_1 = \begin{bmatrix} \bar{g}_1^{1,1} & \bar{g}_2^{1,1} & \cdots & \bar{g}_m^{1,1} & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & \bar{g}_1^{2,2} & \bar{g}_2^{2,2} & \cdots & \bar{g}_m^{2,2} & 0 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & \bar{g}_1^{n,n} & \bar{g}_2^{n,n} & \cdots & \bar{g}_m^{n,n} \end{bmatrix}$$



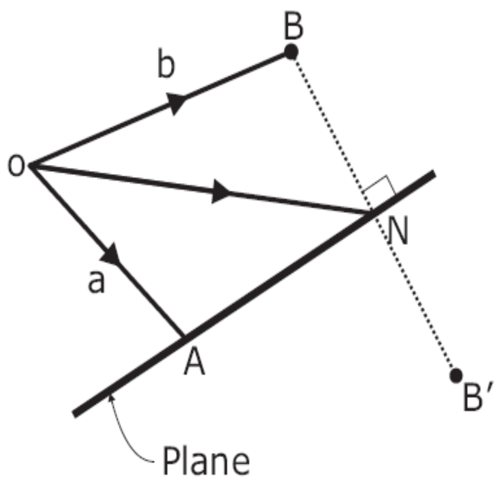
Pictorial Representation of parameter hopping

Term

$$\frac{2}{tr\{({}^1X^i)^T\;{}^1X^i\}}\;{}^1X^i\;{}^1W^i\;({}^1X^i)^T$$

determines the magnitude of weight hopping, which 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.

- In our case the boundary surface is linear and the direction θ of updating is normal to it. 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) the forbidden plane ${}^1X^i \cdot {}^1W^i = 0$, it 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 plane.



Vector explanation of parameter hopping

Vector equation of a line $r = b + \lambda n$

Vector equation of a plane
Passing through 0 $r \cdot n = 0$

Point N of BN is also a point of the plane

$$r \cdot n = 0 \Rightarrow (b + \lambda n) \cdot n = 0$$

$$\Rightarrow \lambda = -\frac{b \cdot n}{\|n\|}$$

Apparently, if one wants to get the position vector of $B^{/}$ (the symmetrical of B in respect to the plane), this is given by

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

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

E.Weight updating laws for identification and indirect control

- Consider the control scheme given by:

$$\boxed{\dot{e} = Ae + X\tilde{W}S(\chi) + X_1\tilde{W}_1S_1(\chi)U} \quad \boxed{u = -\left[X_1W_1S_1(\chi)\right]^{-1} XWS(\chi)} \quad \boxed{\dot{\hat{\chi}} = A\hat{\chi}}$$

The learning law: a) For the elements of W^i

$$\boxed{\dot{w}^{i}_{jl} = -\overline{f}^{\,i}_{j}\, p_i e_i s_l(\chi)}$$

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

$$
{}^1\dot{W}^i = \begin{cases}
-({}^1X^i)^T p_i e_i u_i s_i(\chi) & \text{if } \left|{}^1X^i \cdot {}^1W^i\right| > \theta_i > 0 \\
& \text{or } \left|{}^1X^i \cdot {}^1W^i\right| = \theta_i \\
& \text{and } {}^1X^i \cdot {}^1\dot{W}^i \le 0 \\
-({}^1X^i)^T p_i e_i u_i s_i(\chi) - \dfrac{2}{tr\{({}^1X^i)^T\, {}^1X^i\}}\, {}^1X^i\, {}^1W^i\, ({}^1X^i)^T & \text{if } \left|{}^1X^i \cdot {}^1W^i\right| = \theta_i \\
& \text{and } {}^1X^i \cdot {}^1\dot{W}^i > 0
\end{cases}
$$

guarantees the following properties:

- ✓ $e, \hat{\chi}, \tilde{W}, \tilde{W}_1 \in L_\infty, \quad e, \hat{\chi} \in L_2$
- ✓ $\lim_{t\to\infty} e(t) = 0, \quad \lim_{t\to\infty} \hat{\chi}(t) = 0$
- ✓ $\lim_{t\to\infty} \dot{\tilde{W}}(t) = 0, \quad \lim_{t\to\infty} \dot{\tilde{W}}_1(t) = 0$

E. Simulation Results

- The dynamics of our system which is a 1 KW DC motor are described as follows:

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

- 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.

- The regulation problem of a DC motor is translated as follows: Find a state feedback to force the angular velocity and the armature current to go to zero, while the magnetic flux varies.

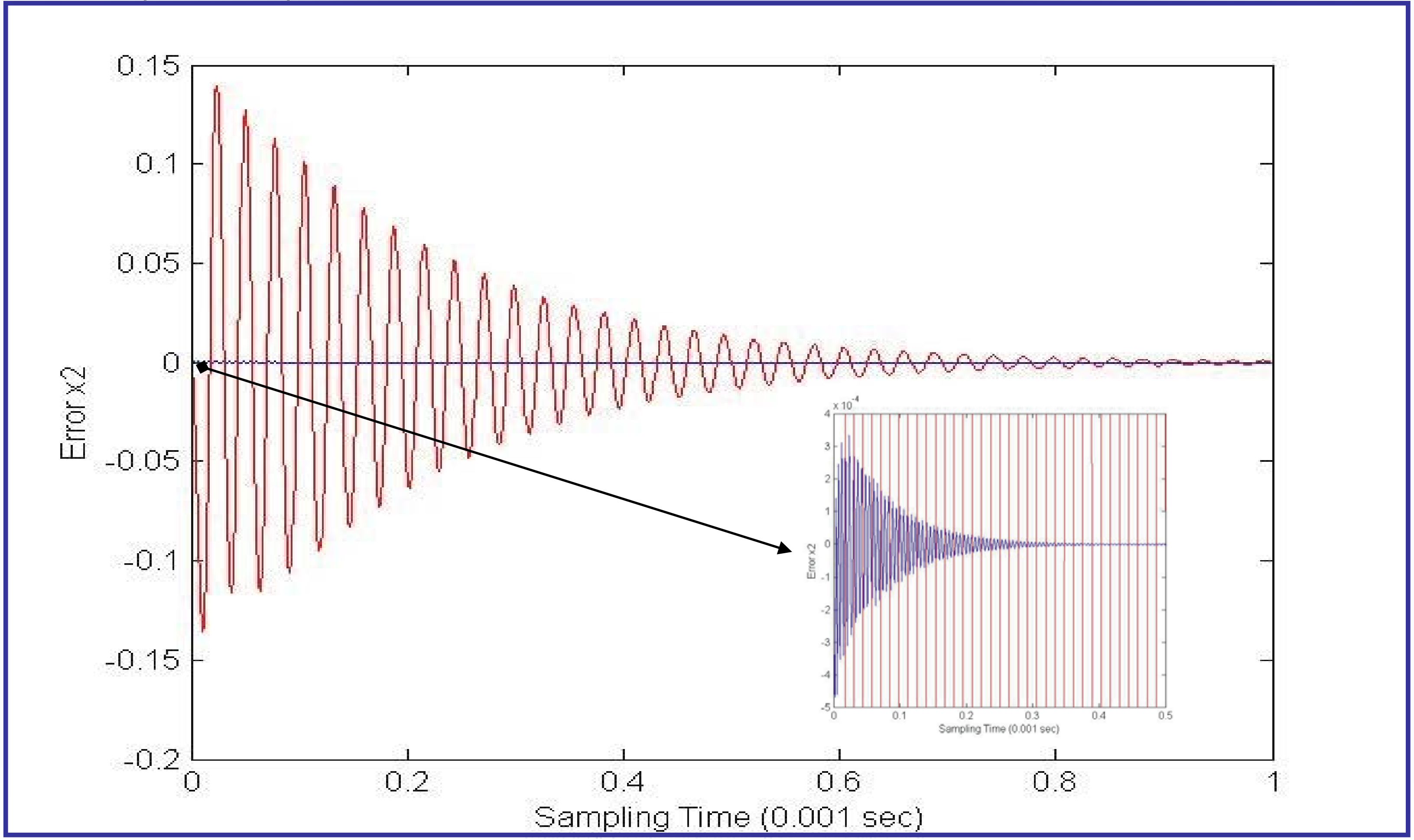
- When Φ is considered constant, the above nonlinear 3rd order system can be linearized and reduced to a second order form having 2 states (I and Ω) Inspired by that, we first assume that the system is described, within a degree of accuracy, by a 2*nd* order linear neuro-fuzzy system of the form

$$\boxed{\begin{aligned}\dot{\chi}_1 &= -a_1\chi_1 + X\,{}^1W^{*1}S(\chi) + {}^1X\,{}^{1\,1}W^{*1}s(\chi_1)u_1 \\ \dot{\chi}_2 &= -a_2\chi_2 + X\,{}^2W^{*2}S(\chi) + {}^1X\,{}^{2\,1}W^{*2}s(\chi_2)u_2\end{aligned}}$$

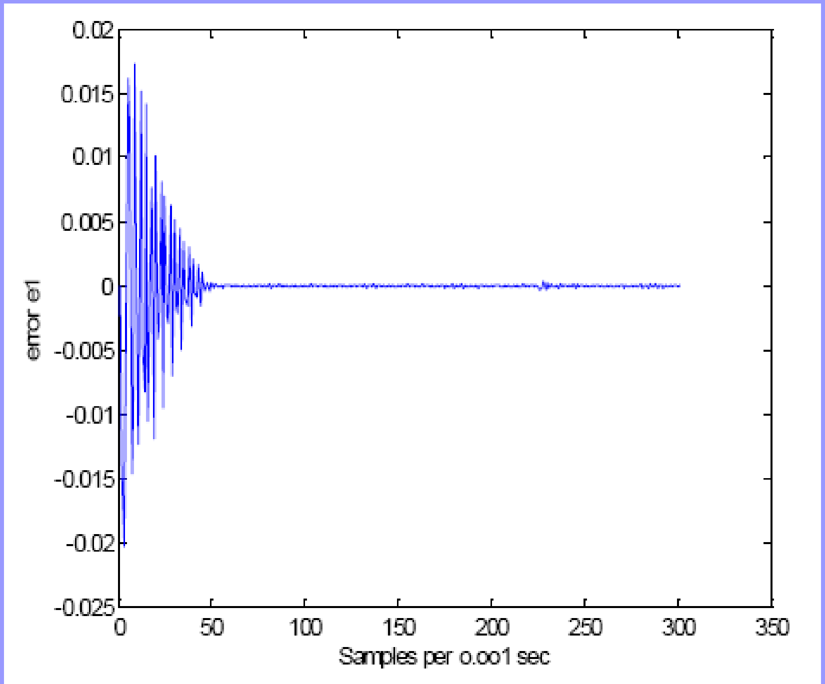
- In the above neuro-fuzzy (F-HONNF) model, the number of fuzzy partitions of each f_i is selected to be *m* = 5 and the depth of high order sigmoid terms *k*=9. In this case $s_i(x)$ assume high order connection up to the third order. Also, the number of fuzzy partitions of each g_{ii} is *m* = 3 with high order connection up to the first order. Thus, the dynamic equations can be written in a more detailed form

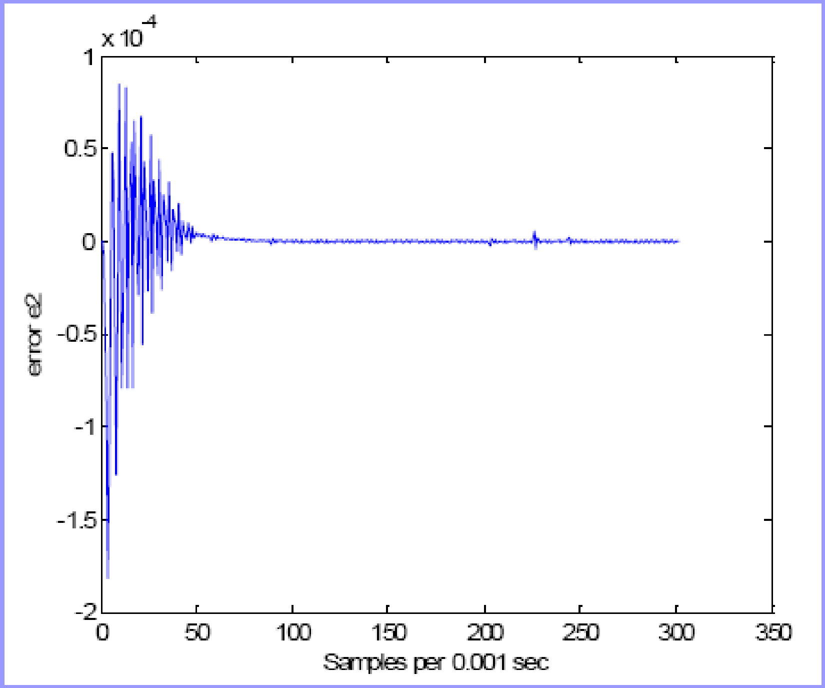
$$\boxed{\begin{aligned}\dot{\chi}_1 &= -a_1\chi_1 + \bar{f}_1^1\left(W^*_{1,1}s_1(\chi)+\cdots+W^*_{1,9}s_9(\chi)\right)+\cdots+\bar{f}_6^1\left(W^*_{6,1}s_1(\chi)+\cdots+W^*_{6,9}s_9(\chi)\right)+\left(\bar{g}_1^{1,1}W^*_{1,1}+\cdots+\bar{g}_3^{1,1}W^*_{3,1}\right)s(\chi_1)u_1 \\ \dot{\chi}_2 &= -a_2\chi_2 + \bar{f}_1^2\left(W^*_{7,1}s_1(\chi)+\cdots+W^*_{7,9}s_9(\chi)\right)+\cdots+\bar{f}_6^2\left(W^*_{12,1}s_1(\chi)+\cdots+W^*_{12,9}s_9(\chi)\right)+\left(\bar{g}_1^{2,2}W^*_{4,2}+\cdots+\bar{g}_3^{2,2}W^*_{6,2}\right)s(\chi_2)u_2\end{aligned}}$$

For comparison purposes we test the identification abilities of the proposed F-HONNF model against the conventional RHONN approximator, using equivalent parameters regarding learning rate and number of high order terms used. The following scheme shows the performance of the proposed scheme (blue line) against the corresponding performance of RHONN (red line).



- In the identification phase, we use the parameters $a_i = 15$ and the range of f_1 [-182.5667 , 0], f_2[-19.3627 , 30.0566], g_{11} [148 , 150] and g_{22}[42 , 44]. The inputs were chosen to be $u_1 = u_2 = 1 + 0.8\sin(0.001t)$. All initial values were set to zero, except that of the magnetic flux which was taken to be equal to 0.98. The following figures gives the evolution of the errors e_1 and e_2 respectively.



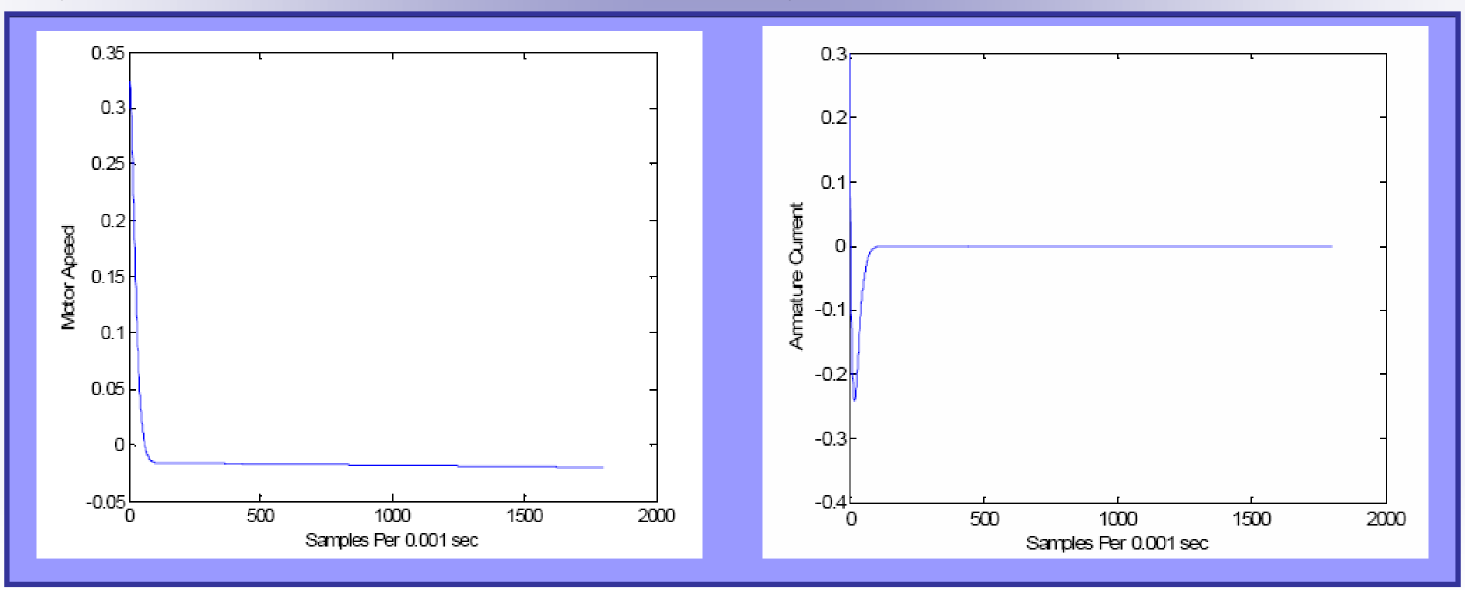


The produced control law is applied on the system, which in turn produces states x_1, x_2, which in the sequel are used for the computation of the estimation errors that are employed by the updating laws. Therefore, the control inputs has the form:

$$u_2 = -\frac{\bar{f}_1^2\left(W_{7,1}s_1(\chi)+\cdots+W_{7,9}s_9(\chi)\right)+\cdots+\bar{f}_6^2\left(W_{12,1}s_1(\chi)+\cdots+W_{12,9}s_9(\chi)\right)}{\left(\bar{g}_1^{2,2}W_{4,2}+\cdots+\bar{g}_3^{2,2}W_{6,2}\right)s(\chi_2)}$$

$$u_1 = -\frac{\bar{f}_1^1\left(W_{1,1}s_1(\chi)+\cdots+W_{1,9}s_9(\chi)\right)+\cdots+\bar{f}_6^1\left(W_{6,1}s_1(\chi)+\cdots+W_{6,9}s_9(\chi)\right)}{\left(\bar{g}_1^{1,1}W_{1,1}+\cdots+\bar{g}_3^{1,1}W_{3,1}\right)s(\chi_1)}$$

In the control phase, the following figures show the evolution of the angular velocity and armature current respectively.



F. Conclusions

➲ The results of this paper indicate that the concept of Fuzzy Dynamical Systems (FDS) operating in conjunction with Recurrent High Order Neural Networks (RHONNs) can be successfully used in the indirect control of general unknown nonlinear dynamical systems.

➲A novel method of parameter hopping was introduced to assure the existence of the control signal

➲ The two stage procedure that was followed, was tested on the control of a ‘Dc Motor’. It was shown that the proposed Neuro-Fuzzy model lead the states to zero very fast, while all signals remain bounded.

➲ The proposed approach can be extended to direct control and to systems with parameter and dynamics uncertainties.

Publications

- M. A. Christodoulou, D. C. Theodoridis, and Y. S. Boutalis, "Building Optimal Fuzzy Dynamical Systems Description Based on Recurrent Neural Network Approximation", in Proc. Int. Conf. of Networked Distributed Systems for Intelligent Sensing and Control, Kalamata, Greece, June, 2007.
- D. C. Theodoridis, Y. S. Boutalis and M. A. Christodoulou, "Indirect Adaptive Neuro - Fuzzy Control Based On High Order Neural Network Function Approximators", in Proc. 16th Mediterranean Conference on Control and Automation - MED08, pp. 386-393, Ajaccio, Corsica, France, June 25-27, 2008.
- 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, accepted for publication, 2008.

- Y.S.Boutalis, M.A.Christodoulou and D.C.Theodoridis, “A new Neuro-Fuzzy intelligent method for indirect adaptive control of nonlinear systems via a novel approach of parameter hopping”, Joint 4th International Conference on Soft Computing and Intelligent Systems and 9th International Symposium on Advanced Intelligent Systems, pp. 918-923, September 17-21, 2008, Nagoya University, Nagoya, JAPAN.
- D.C.Theodoridis, Y.S.Boutalis, and M.A.Christodoulou, “Direct Adaptive Control of Unknown Nonlinear Systems using a new Neuro-Fuzzy Method Together with a Novel Approach of Parameter Hopping”, Kybernetica, 2008 (to appear).

Thank you!