Diploma Thesis

Bilinear Neuro-Fuzzy Direct Adaptive Control of unknown Nonlinear Systems in Brunovski Canonical Form

D.G. Christeas

Author

Technical University of Crete
Department of Electronic and Computer Engineering
Systems Division

Acknowledgement-Dedication

I would like to thank especially my supervisor professor Mr. Manolis Christodoulou for his unlimited help and patience to fulfill this diploma thesis.

Furthermore,great thanks to Mr.Yannis Boutalis and Mr.Dimitris Theodoridis for their support and help with the theoretical part of this scientific statement.

Also,great thanks to my friend Fillip Andreadis,student in Department of Electronic and Computer Engineering.

Finally,I would like to dedicate this thesis to my loving parents who have been supporting me all these years and to my grandmother Panagiota who died recently...

Preface

Fuzzy sets were introduced by Zadeh (1965) as a means of representing and manipulating data that was not precise, but rather fuzzy.

Fuzzy logic provides an inference morphology that enables approximate human reasoning capabilities to be applied to knowledge-based systems. The theory of fuzzy logic provides a mathematical strength to capture the uncertainties associated with human cognitive processes, such as thinking and reasoning. The conventional approaches to knowledge representation lack the means for representating the meaning of fuzzy concepts. Artificial neural systems can be considered as simplified mathematical models of brain-like systems and they function as parallel distributed computing networks. However, in contrast to conventional computers, which are programmed to perform specific task, most neural networks must be taught, or trained.

Perhaps the most important advantage of neural networks is their adaptivity. Neural networks can automatically adjust their weights to optimize their behavior as pattern recognizers, decision makers, system controllers, predictors, etc. Adaptivity allows the neural network to perform well even when the environment or the system being controlled varies over time. 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, the selected type of the membership functions and the proper number of fuzzy rules.

List of Figures

Contents

Part I

Abstract

1 Abstract

The direct adaptive regulation of unknown nonlinear dynamical systems is considered in this DIPLOMA THESIS.

The method 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).

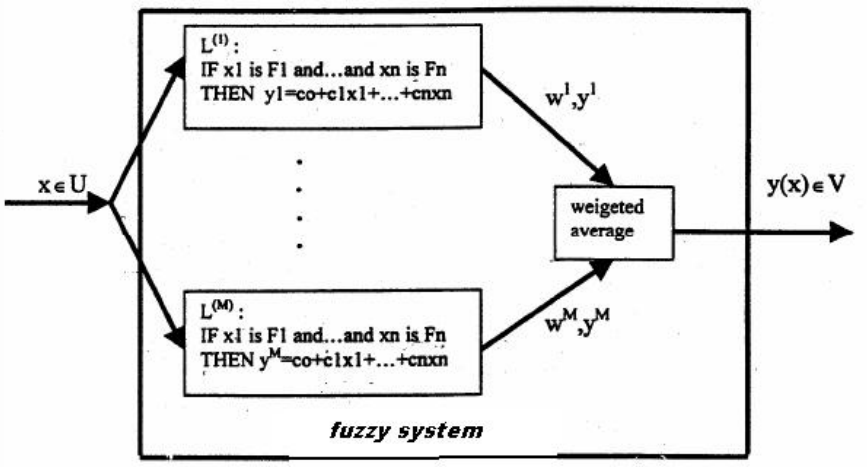

Figure 1. Fuzzy Dynamical System

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.

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

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.

Once the system is identified around an operation point, it is regulated to zero adaptively.

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 existence of the control signal is always assured by introducing a method of parameter hopping,which is incorporated in the weight updating law.

Part II

Introduction

2 Introduction

Nonlinear time invariant 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 [(1)], [(2)], 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 [(3)] - [(4)].

The neural and fuzzy approaches are most of the time equivalent, differing between each other mainly in the structure of the approximator chosen.

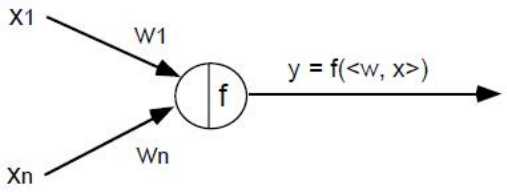

Figure 2. Neural Network

Indeed, in order to bridge the gap between the neural and fuzzy approaches several researchers introduce adaptive schemes using a class of parameterized functions that include both neural networks and fuzzy systems [(5)] - [(9)].

Regarding the approximator structure, linear in the parameters approximators are used in [(9)], [(12)], and nonlinear in [(13)], [(14)], [(15)].

In the neuro or neuro fuzzy control approaches, most of the already presented works [(9)] - [(15)] deal with indirect adaptive control (trying first to identify the dynamics of the systems and then generating a control input according to the certainty equivalence principle), whereas few authors [(16)] and [(12)] face the direct approach (directly generating the control input to guarantee stability), because it is not always clear how to construct the control law without knowledge of the system dynamics.

Recently [(18)], [(19)], 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. In this diploma thesis HONNFs are also used for the neuro fuzzy direct control of nonlinear dynamical systems, which comprises of two interrelated phases:

First the identification of the model and second the control of the plant.

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 [(8)], mountain clustering [(10)], and subtractive clustering [(11)]. 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.

In the proposed approach structure identification is also made off-line based either on human expertise or on gathered data.

However, 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.
In this thesis HONNFs are also used for the neuro fuzzy direct control
of nonlinear dynamical systems in Brunovsky canonical form with modeling errors.
In the proposed approach the underlying fuzzy model is of Mamadani type.
The structure identification is also made off-line based either on human
expertise or on gathered data.
However [(20)], 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

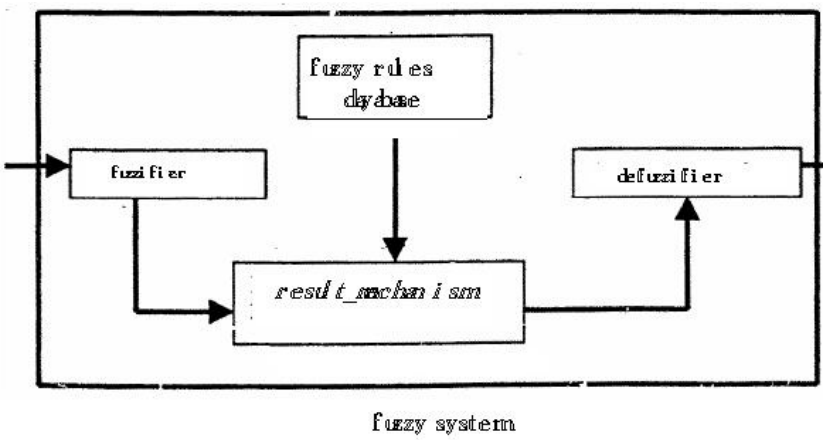

Figure 3. Mamadani's Fuzzy model

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 can be expressed in Brunovsky canonical form. We also consider that its unknown nonlinearities could be approximated with the help of two independent fuzzy subsystems.

We also assume the existence of disturbance expressed as modeling error terms depending on both input and system states.

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 and the modeling error terms are within a certain region depending on the input and state values, then the error reaches zero very fast.

Also, an appropriate state feedback 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, which is incorporated in the weight updating law.

References

[1] K. Hornic, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators", *Neural Networks*, vol. 2, pp. 359-366, 1989.

[2] L. Wang, *Adaptive Fuzzy Systems and Control*, Prentice Hall, Englewood Cliffs, NJ, 1994.

[3] J. S. R. Jang, "ANFIS: Adaptive-network-based fuzzy inference system", *IEEE Trans. Syst. Man. Cyber.*, vol. 23, pp. 665-684, 1993.

[4] C.T. Lin, "A neural fuzzy control system with structure and parameter learning", *Fuzzy Sets and Systems*, vol. 70, pp. 183-212, 1995.

[5] 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*, vol. 83, pp. 325-339, 1996.

[6] C.F.Juang and C. T. Lin, "An on-line self-constructing neural fuzzy inference network and its applications", *IEEE Trans. Fuzzy Syst.*, vol. 6, pp. 12-32, 1998.

[7] R. P. Li and M. Mukaidono, "A new approach to rule learning based on fusion of fuzzy logic and neural networks", *IEICE Trans. Fuzzy Syst*, vol. E78-d, pp. 1509-1514, 1995.

[8] S. L. Chiu, "Fuzzy model identification based on cluster estimation", *Journal of Intelligent and Fuzzy Systems*, vol. 2, 1994.

[9] Y. H. Lin and G. A. Cunningham, "A new approach to fuzzy-neural system modelling", *IEEE Trans. Fuzzy Syst.*, vol. 3, pp. 190-197, 1995.

[10] C. F. Jang and C. T. Lin, "An online self-constructing neural fuzzy inference network and its applications", *IEEE Trans. Fuzzy Syst.*, vol. 6, pp. 12-32, 1998.

[11] S. Mitra and Y. Hayashi, "Neuro-fuzzy rule generation: survey in soft computing framework", *IEEE Trans. On Neural Networks*, vol. 11, pp. 748-768, 2000.

[12] B. S. Chen, C. H. Lee, and Y. C. Chang, "Tracking design of uncertain nonlinear siso systems: Adaptive fuzzy approach", *IEEE Trans. Fuzzy Syst.*, vol. 4, pp. 32-43, Feb. 1996.

[13] J. T. Spooner and K. M. Passino, "Stable adaptive control using fuzzy systems and neural networks", *IEEE Trans. Fuzzy Syst.*, vol. 4, pp. 339-359, June 1996.

[14] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks", *IEEE Trans. Neural Networks*, vol. 1, pp. 4-27, Feb.1990.

[15] M. M. Polycarpou and M. J. Mears, "Stable adaptive tracking of uncertain systems using nonlinearly parameterized online approximators", *Int. J. Control*, vol. 70, no. 3, pp. 363-384, 1998.

[16] G. A. Rovithakis and M. A. Christodoulou, "Adaptive control of unknown plants using dynamical neural networks", *IEEE Trans. Syst, Man, Cybern.*, vol. 24, , pp. 400-412 Mar. 1994.

[17] N. Golea, A. Golea, K. Benmahammed, "Stable indirect fuzzy adaptive control", *Fuzzy Sets and Systems*, vol. 137, pp. 353-366, 2003.

[18] E. B. Kosmatopoulos and M. A. Christodoulou, "Recurrent neural networks for approximation of fuzzy dynamical systems", *Int. Journal of Intelligent Control and Systems*, vol. 1, pp. 223-233, 1996.

[19] 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.

[20] W. Yu, X. Li, "Fuzzy neural identification by online clustering with application on crude oil blending", *Int. Conf. on Fuzzy Systems*, Vancouver, BC, Canada, July 16-21, 2006.

Part III

Neuro-Fuzzy Adaptive Systems

3 Neuro-Fuzzy Adaptive Systems

3.1 Fuzzy Systems-Theory-Historical Background

Fuzzy sets were introduced by Zadeh (1965) as a means of representing

and manipulating data that was not precise, but rather fuzzy.

Fuzzy logic provides an inference morphology that enables approximate

human reasoning capabilities to be applied to knowledge-based systems.

The theory of fuzzy logic provides a mathematical strength to capture

the uncertainties associated with human cognitive processes, such as

thinking and reasoning. The conventional approaches to knowledge

representation lack the means for representating the meaning of fuzzy concepts.

As a consequence, the approaches based on first order logic and classical

probablity theory do not provide an appropriate conceptual framework for

dealing with the representation of commonsense knowledge, since such knowledge

is by its nature both lexically imprecise and noncategorical.The developement of

fuzzy logic was motivated in large measure by the need for a conceptual framework

which can address the issue of uncertainty and lexical imprecision.

Some of the essential characteristics of fuzzy logic relate to the following :

- In fuzzy logic, exact reasoning is viewed as a limiting case of approximate reasoning.
- In fuzzy logic, everything is a matter of degree.
- In fuzzy logic, knowledge is interpreted a collection of elastic or, equivalently, fuzzy constraint on a collection of variables.

- Inference is viewed as a process of propagation of elastic constraints.

- Any logical system can be fuzzified.
- There are two main characteristics of fuzzy systems that give them better performance for specific applications.
- Fuzzy systems are suitable for uncertain or approximate reasoning, especially for the system with a mathematical model that is difficult to derive.
- Fuzzy logic allows decision making with estimated values under incomplete or uncertain information.

3.2 Neural Systems

While fuzzy logic performs an inference mechanism under cognitive uncertainty, computational neural networks offer exciting advantages, such as learning, adaptation, fault-tolerance, parallelism and generalization. A brief comparative study between fuzzy systems and neural networks in their operations in the context of knowledge acquisition, uncertainty, reasoning and adaptation is presented in figure 4

Artificial neural systems can be considered as simplified mathematical models of brain-like systems and they function as parallel distributed computing networks. However, in contrast to conventional computers, which are programmed to perform specific task, most neural networks must be taught, or trained.
They can learn new associations, new functional dependencies and new patterns.
The study of brain-style computation has its roots over 50 years ago in the work of McCulloch and Pitts (1943) and slightly later in Hebbs famous Organization of Behavior (1949). The early work in artificial intelligence was torn between those

Skills		Fuzzy Systems	Neural Nets
Knowledge acquisition	Inputs Tools	Human experts Interaction	Sample sets Algorithms
Uncertainty	Information	Quantitive and Qualitive	Quantitive
	Cognition	Decision making	Perception
Reasoning	Mechanism Speed	Heuristic search Low	Parallel computations High
Adaption	Fault-tolerance Learning	Low Induction	Very high Adjusting weights
Natural language	Implementation Flexibility	Explicit High	Implicit Low

Figure 4. Properties of fuzzy systems and neural networks

who believed that intelligent systems could best be built on computers modeled after brains, and those like Minsky and Papert who believed that intelligence was fundamentally symbol processing of the kind readily modeled on the von Neumann computer.For a variety of reasons, the symbol-processing approach became the dominant theme in artifcial intelligence.

The 1980s showed a rebirth in interest in neural computing:

- Hopfield (1985) provided the mathematical foundation for understanding the dynamics of an important class of networks.
- Rumelhart and McClelland (1986) introduced the backpropagation learning algorithm for complex, multi-layer networks and thereby provided an answer to one of the most severe criticisms of the original perceptron work.

To enable a system to deal with cognitive uncertainties in a manner more like humans, one may incorporate the concept of fuzzy logic into

the neural networks. The resulting hybrid system is called fuzzy neural, neural fuzzy, neuro-fuzzy or fuzzy-neuro network.

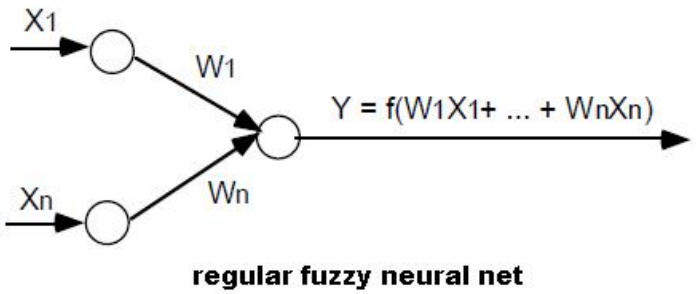

Figure 5. Fuzzy-Neuro network

Neural networks are used to tune membership functions of fuzzy systems that are employed as decision-making systems for controlling equipment. Although fuzzy logic can encode expert knowledge directly using rules with linguistic labels, it usually takes a lot of time to design and tune the membership functions which quantitatively define these linquistic labels. Neural network learning techniques can automate this process and substantially reduce development time and cost while improving performance. In theory, neural networks, and fuzzy systems are equivalent in that they are convertible, yet in practice each has its own advantages and disadvantages. For neural networks, the knowledge is automatically acquired by the backpropagation algorithm, but the learning process is relatively slow and

analysis of the trained network is difficult (black box). Neither is it possible to extract structural knowledge (rules) from the trained neural network, nor can we integrate special information about the problem into the neural network in order to simplify the learning procedure. Fuzzy systems are more favorable in that their behavior can be explained based on fuzzy rules and thus their performance can be adjusted by tuning the rules. But since, in general, knowledge acquisition is difficult and also the universe of discourse of each input variable needs to be divided into several intervals, applications of fuzzy systems are restricted to the fields where expert knowledge is available and the number of input variables is small. To overcome the problem of knowledge acquisition, neural networks are extended to automatically extract fuzzy rules from numerical data. Cooperative approaches use neural networks to optimize certain parameters of an ordinary fuzzy system, or to preprocess data and extract fuzzy (control) rules from data.

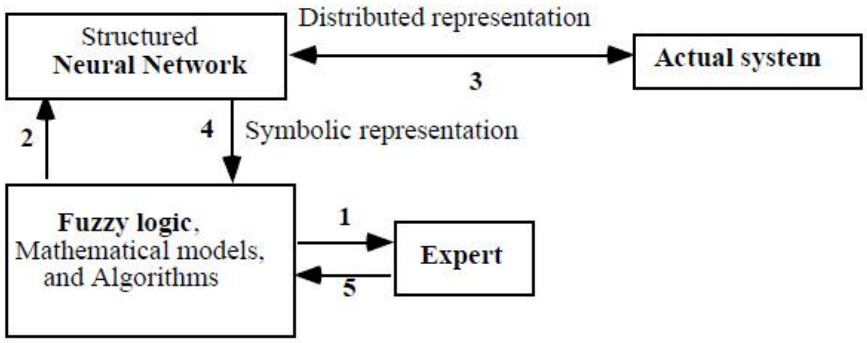

Figure 6. Neural fuzzy system

1 **Translate experts knowledge into a symbolic representation.**

2 **Initialize the neural net by symbolic representation.**

3 **Decrease errors between actual system and neural net by learning.**

4 **Translate the distributed representation based upon the structure of neural net.**

5 **Acquire knowledge from the modified symbolic representation.**

The basic processing elements of neural networks are called artificial neurons, or simply neurons. We emphasize here that all inputs, outputs and the weights of a hybrid neural net are real numbers taken from the unit interval $[0, 1]$.

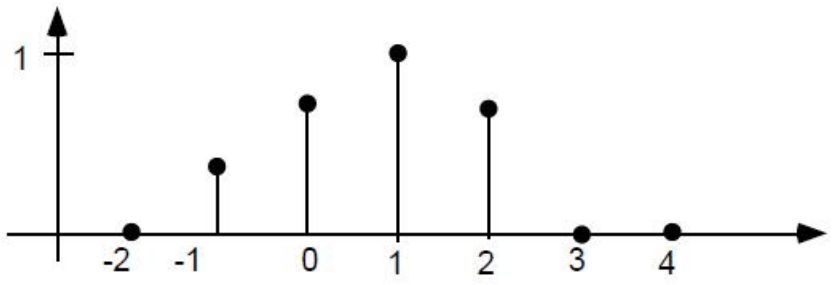

Figure 7. A discrete membership function for x is close to 1

A processing element of a hybrid neural net is called fuzzy neuron.

It is well-known that regular nets are universal approximators,i.e. they can approximate any continuous function on a compact set to arbitrary accuracy.

In a discrete fuzzy expert system one inputs a discrete approximation to the fuzzy

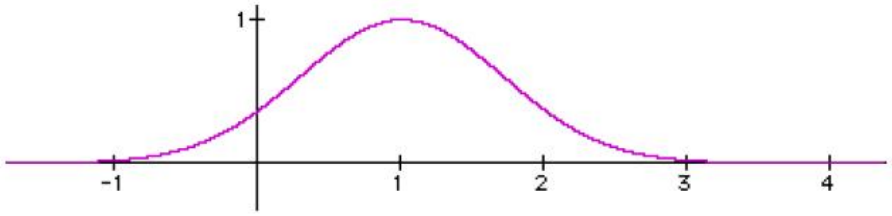

Figure 8. A membership function for x is close to 1

sets and obtains a discrete approximation to the output fuzzy set.

Usually discrete fuzzy expert systems and fuzzy controllers

are continuous mappings. Thus we can conclude that given a continuous fuzzy

expert system, or continuous fuzzy controller, there is a regular net that

can uniformly approximate it to any degree of accuracy on compact sets.

The problem with this result that it is non-constructive and does not tell you

how to build the net.

A fuzzification operator has the effect of transforming crisp data into fuzzy sets.

In most of the cases we use fuzzy singletons as fuzzifiers

$fuzzifier(x0) := x0$

where x0 is a crisp input value from a process.

Hybrid neural nets can be used to implement fuzzy IF-THEN rules in a constructive

way. Though hybrid neural nets can not use directly the standard error

backpropagation algorithm for learning, they can be trained by steepest

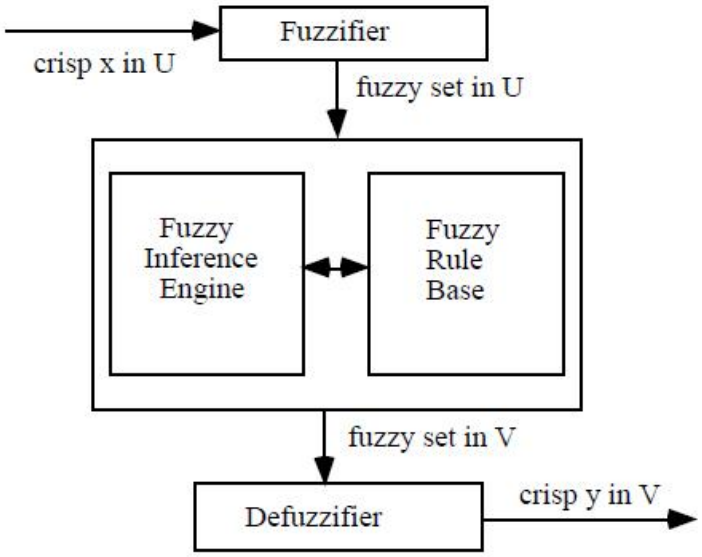

Figure 9. Fuzzy Logic Controller

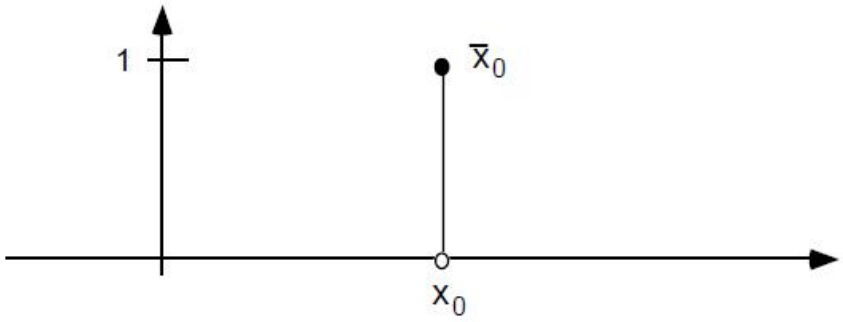

Figure 10. Fuzzy singleton as fuzzifier

descent methods to learn the parameters of the membership functions representing the linguistic terms in the rules. The direct fuzzification of conventional neural networks is to extend connection weigths and/or inputs and/or fuzzy desired outputs (or targets) to fuzzy numbers. The computational process envisioned for fuzzy neural systems is as follows. It starts with the development of a fuzzy neuron based on the understanding of biological neuronal morphologies, followed by learning mechanisms.This leads to the following three steps in a fuzzy neural computational process development of fuzzy neural models motivated by biological neurons, models of synaptic connections which incorporates fuzziness into neural network, development of learning algorithms (that is the method of adjusting the synaptic weights). Two possible models of fuzzy neural systems are :

- In response to linguistic statements, the fuzzy interface block provides an input vector to a multi-layer neural network. The neural network can be adapted (trained) to yield desired command outputs or decisions.
- A multi-layered neural network drives the fuzzy inference mechanism.

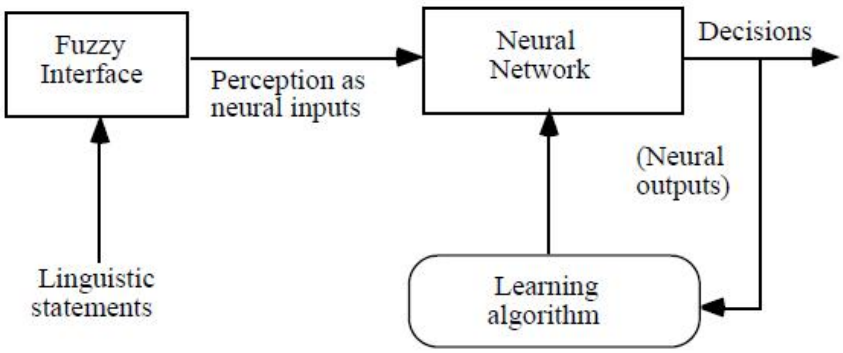

Figure 11. The first model of fuzzy neural system.

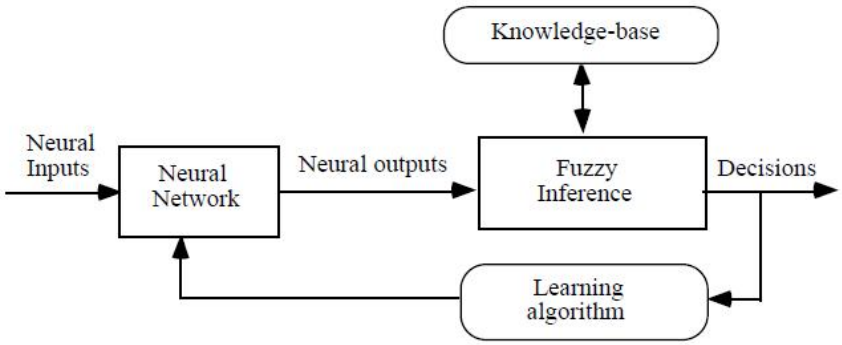

Figure 12. The second model of fuzzy neural system.

3.3 Applications of artificial neural networks

There are large classes of problems that appear to be more amenable to solution by neural networks than by other available techniques. These tasks often involve ambiguity, such as that inherent in handwritten character recognition. Problems of this sort are difficult to tackle with conventional methods such as matched filtering or nearest neighbor classification, in part because

the metrics used by the brain to compare patterns may not be very closely related to those chosen by an engineer designing a recognition system. Likewise, because reliable rules for recognizing a pattern are usually not at hand, fuzzy logic and expert system designers also face the difficult and sometimes impossible task of finding acceptable descriptions of the complex relations governing class inclusion. In trainable neural network systems, these relations are abstracted directly from training data. Moreover, because neural networks can be constructed with numbers of inputs and outputs ranging into thousands, they can be used to attack problems that require consideration of more input variables than could be feasibly utilized by most other approaches. It should be noted, however, that neural networks will not work well at solving problems for which sufficiently large and general sets of training data are not obtainable:

- **The telecommunications industry** :

  Many neural network applications are under development in the telecommunications industry for solving problems ranging from control of a nationwide switching network to management of an entire telephone company. Other aplications at the telephone circuit level turn out to be the most significant commercial applications of neural networks in the world today.Modems, commonly used for computer-to-commputer communications and in every fax machine, have adaptive circuits for telephone line equalization and for echo cancellation.

- **Control of sound and vibration** :

  Active control of vibration and noise is accomplished by using an adaptive actuator to generate equal and opposite vibration and noise.This is being used in

air-conditioning systems, in automotive systems, and in industrial applications.

- **Particle accelerator beam control**:

  The Stanford linear accelerator Center is now using adaptive techniques to cancel disturbances that diminish the positioning accuracy of opposing beams of positrons and electrons in a particle colloder.

- **Quality control in manufacturing** :

  Neural networks are being used in a large number of quality control and quality assurance programs throughout industry.Applications include contaminant-level detection from spectroscopy data at chemical plants and loudspeaker defect classification by CTS Electronics.

- **Medical applications** :

  Commercial products by Neuromedical Systems Inc. are used for cancer screening and other medical applications. The company markets electrocardiograph and pap smear systems that rely on neural network technology. The pap smear system. Papnet,is able to help cytotechnologists spot cancerous cells, drastically reducing false/negative classifications.

- **Automobile applications** :

  Ford Motor Co., General Motors, and other automobile manufacturers are currently researching the possibility of widespread use of neural networks in automobiles and in automobile production. Some of the areas that are yielding promising results in the laboratory include engine fault detection and diagnosis, antilock brake control, active-suspension control, and idle-speed control. General Motors is having preliminary success using neural networks to model subjective customer ratings of automobiles

based on their dynamic characteristics to help engineers tailor vehicles to the market.

- **Biomedical applications** :

  Neural networks are rapidly finding diverse applications in the biomedical sciences. They are being used widely in research on amino acid sequencing in RNA and DNA, ECG and EEG waveform classification, prediction of patients reactions to drug treatments, prevention of anesthesia-related accidents, arrhythmia recognition for implantable defibrillators patient mortality predictions, quantitative cytology, detection of breast cancer from mammograms, modeling schizophrenia, clinical diagnosis of lowerback pain, enhancement and classification of medical images, lung nodule detection, diagnosis of hepatic masses, prediction of pulmonary embolism likelihood from ventilation-perfusion lung scans, and the study of interstitial lung disease.

3.4 Adaptivity of Neural Networks

Perhaps the most important advantage of neural networks is their adaptivity.

Neural networks can automatically adjust their weights to optimize their behavior as pattern recognizers, decision makers, system controllers, predictors, etc. Adaptivity allows the neural network to perform well even when the environment or the system being controlled varies over time. There are many control problems that can benefit from continual nonlinear modeling and adaptation.While fuzzy logic performs an inference mechanism under cognitive uncertainty, computational neural networks offer exciting

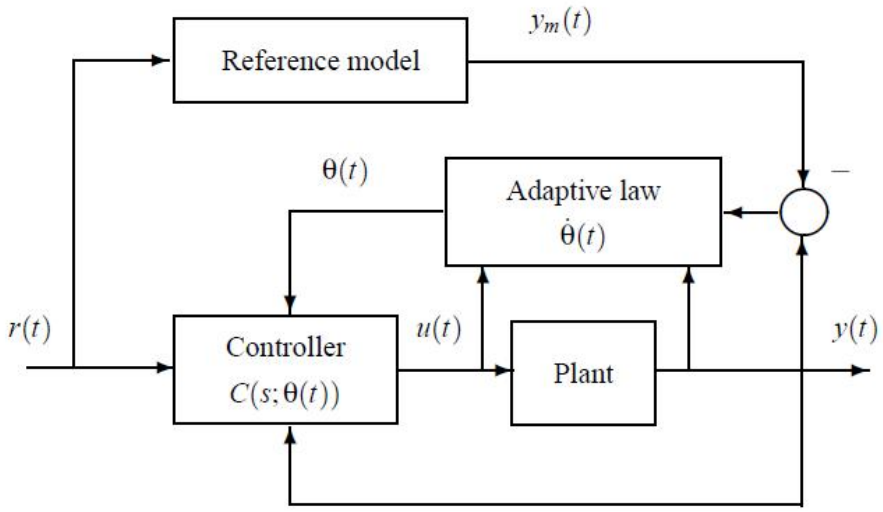

Figure 13. Direct Adaptive Control System

advantages, such as learning, adaptation, fault-tolerance, parallelism and generalization. Self-optimization allows the neural network to design itself. The system designer first defines the neural network architecture, determines how the network connects to other parts of the system, and chooses a training methodology for the network. The neural network then adapts to the application. Neural networks, such as those used by Pavilion in chemical process control, and by Neural Application Corp. in arc furnace control, are ideally suited to track problem solutions in changing environments. Additionally, with some programmability, such as the choices regarding the number of neurons per layer and number of layers, a practitioner can use the same neural network in a wide variety of applications. Engineering time is thus saved.

Another example of the advantages of self-optimization is in the field of Expert Systems. In some cases, instead of obtaining a set of rules through interaction between an experienced expert and a knowledge engineer, a neural

system can be trained with examples of expert behavior.

A good understanding of adaptive control involves good knowledge of control design for linear time-invariant systems, basic stability theory of nonlinear systems, and some mathematical maturity. Several books and research monographs as well as numerous papers on the theory and application of adaptive control already exist. Despite the maturity of the field and the numerous publications, the field of adaptive control appears to many as a collection of unrelated techniques, intelligent tricks, and fixes, and very few researchers can really follow the long and technical stability proofs. On the implementation side, designing stable adaptive control systems and simulating them on a digital computer to demonstrate their theoretical properties could also be an adventure if one does not have a good understanding of the basic theoretical properties and limitations of adaptive control.

The performance, complexity, and adaptive law of an adaptive fuzzy system representation can be quite different depending 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.

According to [(2)], we classify adaptive fuzzy controllers into two types:

- If the fuzzy logic systems used in an adaptive fuzzy controller are linear in their adjustable parameters, this adaptive fuzzy controller is called *a first -type adaptive fuzzy controller*
- If the fuzzy logic systems used in an adaptive fuzzy controller are nonlinear in their adjustable parameters, this adaptive fuzzy controller is called *a second -type adaptive fuzzy controller*

Notice that both first and second types of adaptive fuzzy controllers are nonlinear adaptive controllers.

Suppose that the adaptive fuzzy system is intended to approximate the nonlinear function $f(x)$.

In the first-type adaptive fuzzy controller, Wang [(1)] uses the following fuzzy logic representation:

$$f(x) = \sum_{l=1}^{M} \theta_l \xi_l(x) = \theta^T \xi(x) \tag{2}$$

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

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

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).

Clearly, Eq. (2) is equivalent to the following equation assuming that $\mu_{F_i^l}$ are given: that is, $\mu_{F_i^l}$ will not change during the adaptation procedure.

$$f(x) = \frac{\sum_{l=1}^{M} y^l \left(\prod_{i=1}^{n} \mu_{F_i^l}(x) \right)}{\sum_{l=1}^{M} \left(\prod_{i=1}^{n} \mu_{F_i^l}(x) \right)} \tag{3}$$

In the second-type adaptive fuzzy controller, the following fuzzy logic system is used:

$$f(x) = \frac{\sum_{l=1}^{M} y^l \left(\prod_{i=1}^{n} \exp(-(\frac{x_i - x_i^l}{\sigma_i^l})^2) \right)}{\sum_{l=1}^{M} \left(\prod_{i=1}^{n} \exp(-(\frac{x_i - x_i^l}{\sigma_i^l})^2) \right)} \tag{4}$$

where y^l, x_i^l, σ_i^l are the adjustable parameters.

From the above definitions it is apparent 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,

the selected type of the membership functions and the proper number of fuzzy rules.

In approximating complex nonlinear functions, this number may become very large [(2)] leading to parameter explosion.

References

[1] L. Wang, *Adaptive Fuzzy Systems and Control*, Prentice Hall, Englewood Cliffs, NJ, 1994.

[2] H. L. Hiew, C. P. Tsang, *Fuzzy Chaos and Recursive Partitioning*, in B. Kosko (Ed.), Fuzzy Engineering, Prentice-Hall International Inc. New Jersey, 1997

Part IV

Indicator Functions for Fuzzy Systems

4 Indicator Functions for Fuzzy Systems

Let us consider the system with input space $u \subset R^m$ and state - space $x \subset R^n$,

with its i/o relation being governed by the following equation

$$z^t = f(x^t, u^t) \tag{5}$$

where $f(\cdot)$ is a continuous function and the superscript t denotes the

temporal variable.

In case the system is dynamic the above equation could be replaced by the following difference equation

$$x^{t+1} = f(x^t, u^t) \tag{6}$$

where the superscript t denotes the temporal variable, $t = 1, 2, ...$

By setting $y = [x, u]$ and omiting superscript t, Eq. (26) may be rewritten as follows

$$z = f(y) \tag{7}$$

In many practical situations, we are unable to measure accurately the states

and inputs of a system of the form in (26);

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 *ith* element of the set L_i.

Then we may define a function $\tilde{h}_i : R \to L_i$ to be the output function of the system in Eq. (28) in the case that

$$\tilde{y}_i = \tilde{h}_i(y_i) \tag{8}$$

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 $\tilde{h}_i : R \to \{1, 2, ..., k_i\}$ as follows

$$\tilde{h}_i(y_i) = \tilde{y}_{ij} \Longleftrightarrow h_i(y_i) = j \tag{9}$$

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. (28).

Also, $\tilde{h}_i(\cdot)$ and consequently $h_i(\cdot)$ is related with the structural identification part mentioned in section 2 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

$$\tilde{\mu}_{ij}(y_i) = \max_l \tilde{\mu}_{il}(y_i) \Longleftrightarrow h_i(y_i) = j \tag{10}$$

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

$$\mathcal{Y}_{ij} = \{y_i \in R : h_i(y_i) = j\} \tag{11}$$

i.e. 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 $y_{j_1,j_2,...,j_{n+m}}$ defined as follows

$$y_{j_1,j_2,...,j_{n+m}} := y_{1j_1} \times \cdots \times y_{(n+m)j_{n+m}}, j_i \in \{1,2,...,k_i\} \tag{12}$$

In a similar way we may define the sets x_{ij}, u_{ij}, z_{ij} and the sets $x_{j_1,j_2,...,j_n}$, $u_{j_1,j_2,...,j_n}$ and $z_{j_1,j_2,...,j_n}$.

Note now the following fact:

for two vectors $(x^{(1)},u^{(1)}) \in y_{j_1,j_2,...,j_{n+m}}$ and $(x^{(2)},u^{(2)}) \in y_{j_1,j_2,...,j_{n+m}}$

there maybe

$$h_i(f_i(x^{(1)},u^{(1)})) \neq h_i(f_i(x^{(2)},u^{(2)})) \tag{13}$$

for some $i \in \{1,2,...,n\}$, i.e. two input vectors belonging to the same subset $y_{j_1,j_2,...,j_{n+m}}$ may point - through the vector - field $f(\cdot)$, to different subsets $z_{l_1,l_2,...,l_n}$.

Let now $\Omega^{l_1,l_2,...,l_n}_{j_1,j_2,...,j_{n+m}}$ be defined as the subset of $y_{j_1,j_2,...,j_{n+m}}$ that points - through the vector - field $f(\cdot)$, to the subsets $z_{l_1,l_2,...,l_n}$,

i.e
$\Omega^{l_1,l_2,...,l_n}_{j_1,j_2,...,j_{n+m}} := \{(x,u) \in y_{j_1,j_2,...,j_{n+m}} : h_1(z_1) = l_1, ..., h_n(z_n) = l_n\}$
and define the transition possibilities $\pi^{l_1,...,l_n}_{j_1,...,j_{n+m}}$ as follows

$$\pi^{l_1,...,l_n}_{j_1,...,j_{n+m}} := \frac{\int_{(x,u)\in\Omega^{l_1,...,l_n}_{j_1,...,j_{n+m}}} dXdU}{\int_{(x,u)\in y_{j_1,...,j_{n+m}}} dXdU} \tag{14}$$

where $\pi^{l_1,...,l_n}_{j_1,...,j_{n+m}}$ is a number belonging to a set [0,1] that represents the fraction

of the vectors (x,u) in $y_{j_1,...,j_{n+m}}$ that points - through the vector field $f(\cdot)$

to the set $\chi_{l_1,...,l_n}$.

Obviously

$$\sum_{l_1,\ldots,l_n} \pi^{l_1,\ldots,l_n}_{j_1,\ldots,j_{n+m}} = 1 \tag{15}$$

In order to present the lemma , we define the indicator function:

Let $I^{l_1,l_2,\ldots,l_n}_{j_1,j_2,\ldots,j_{n+m}}$ denote the indicator function of the subset $\Omega^{l_1,l_2,\ldots,l_n}_{j_1,j_2,\ldots,j_{n+m}}$, that is,

$$I^{l_1,\ldots,l_n}_{j_1,\ldots,j_{n+m}}(x,u) = \begin{cases} 1 \; if \; (x,u) \in \Omega^{l_1,\ldots,l_n}_{j_1,\ldots,j_{n+m}} \\ 0 \qquad otherwise \end{cases} \tag{16}$$

Using the above definitions, we can see that the system in Eq. (28) is described by fuzzy rules of the form

$$R^{l_1,\ldots,l_n}_{j_1,\ldots,j_{n+m}} \Leftrightarrow \left\{ \begin{array}{c} IF \; y_1 \; is \; \tilde{y}_{1j_1} \; AND... \\ AND \; y_{n+m} \; is \; \tilde{y}_{(n+m)j_{n+m}} \\ THEN \\ z_1 \; is \; \tilde{z}_{1l_1} \; AND...AND \; z_n \; is \; \tilde{z}_{nl_n} \\ with \; possibility \; \pi^{l_1,\ldots,l_n}_{j_1,\ldots,j_{n+m}} \end{array} \right\}. \tag{17}$$

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

In the above notation, if $j_1 = l_1$, $j_2 = l_2$ and $\ldots$ 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.

4.1 Describing Fuzzy Systems

A Fuzzy System - (FS) is a set of Fuzzy Rules of the form ($R^{l_1,l_2,\ldots,l_n}_{j_1,j_2,\ldots,j_{n+m}}$);

the system in Eq. (26) is called the Underlying System - (US) of the previously defined FS.

Alternatively, the system in Eq. (26) will be called a Generator of the FS that is described by the rules

($R^{l_1,l_2,\ldots,l_n}_{j_1,j_2,\ldots,j_{n+m}}$).

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. (28) to be generators for the FS that is described by the rules

$(R^{l_1,l_2,\ldots,l_n}_{j_1,j_2,\ldots,j_{n+m}})$.

Define now the following system

$$z = \sum \bar{z}^{l_1,\ldots,l_n}_{j_1,\ldots,j_{n+m}} \times I^{l_1,\ldots,l_n}_{j_1,\ldots,j_{n+m}}(\chi, u) \tag{18}$$

Where $\bar{z}^{l_1,\ldots,l_n}_{j_1,\ldots,j_{n+m}} \in R^n$ be any vector satisfying $h_i(\bar{z}^{l_1,\ldots,l_n}_{j_1,\ldots,j_{n+m}}(i)) = l_i$

where $\bar{z}^{l_1,\ldots,l_n}_{j_1,\ldots,j_{n+m}}(i)$ denotes the i^{th} entry of $\bar{z}^{l_1,\ldots,l_n}_{j_1,\ldots,j_{n+m}}$

Then, according to [(1)], [(2)] the system in (30) is a generator for the FS

$(R^{l_1,l_2,\ldots,l_n}_{j_1,j_2,\ldots,j_{n+m}})$.

It is obvious that Eq. (30) can be also valid for dynamic systems. In its dynamical form it becomes

$$\chi^{t+1} = \sum \bar{x}^{l_1,\ldots,l_n}_{j_1,\ldots,j_{n+m}} \times I^{l_1,\ldots,l_n}_{j_1,\ldots,j_{n+m}}(\chi^t, u^t) \tag{19}$$

Where $\bar{x}^{l_1,\ldots,l_n}_{j_1,\ldots,j_{n+m}} \in R^n$ be any vector satisfying $h_i(\bar{x}^{l_1,\ldots,l_n}_{j_1,\ldots,j_{n+m}}(i)) = l_i$

where $\bar{x}^{l_1,\ldots,l_n}_{j_1,\ldots,j_{n+m}}(i)$ denotes the i^{th} entry of $\bar{x}^{l_1,\ldots,l_n}_{j_1,\ldots,j_{n+m}}$

References

[1] E. B. Kosmatopoulos and M. A. Christodoulou, "Recurrent neural networks for approximation of fuzzy dynamical systems", *Int. Journal of Intelligent Control and Systems*, vol. 1, pp. 223-233, 1996.

[2] 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.

Part V

Using High Order Neural Network Functions in order to approximate the indicator functions

5 Using High Order Neural Network Functions in order to approximate the indicator functions

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

$I^{l_1,\ldots,l_n}_{j_1,\ldots,j_{n+m}}$.

However, in order the approximation problem to make sense the space $y := x \times u$

must be compact. Thus, our first assumption is the following:

(**A.1**) $y := x \times u$ is a compact set.

Notice that since $y \subset \Re^{n+m}$ the above assumption is identical to

the assumption that it is closed and bounded. Also, it is noted that even if y

is not compact we may assume that there is a time instant T such that (x^t, u^t)

remain in a compact subset of y for all $t < T$;

i.e. if $y_T := \{(x^t, u^t) \in y, t < T\}$

We may replace assumption (A.1) by the following assumption

(**A.2**) y_T is a compact set.

It is worth noticing, that while assumption (A.1) requires the system in Eq. (27)

solutions to be bounded for all $u^t \in U$ and $x^0 \in X$,assumption (A.2)

requires the system in Eq. (27) solutions to be bounded for a finite time period;

thus, assumption (A.1) requires the system in Eq. (27) to be 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 $I^{l_1,...,l_n}_{j_1,...,j_{n+m}}$

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

$$N(x,u;w,L) = \sum_{k=1}^{L} w_k \prod_{j\in I_k} \Phi_j^{d_j(k)} \tag{20}$$

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

$$Phi = \begin{bmatrix} Phi_1 \\ . \\ . \\ . \\ Phi_n \\ Phi_{n+1} \\ . \\ . \\ . \\ Phi_{m+n} \end{bmatrix} = \begin{bmatrix} S(x_1) \\ . \\ . \\ . \\ S(x_n) \\ S(u_1) \\ . \\ . \\ . \\ S(u_m) \end{bmatrix} \tag{21}$$

where

$$S(u)\, or\, S(x) = a\frac{1}{1+e^{-\beta x}} - \gamma \tag{22}$$

and $w := [w_1 \cdots w_L]^T$ are the HONNF weights.

Eq. (32) can also be written

$$N(x,u;w,L) = \sum_{k=1}^{L} w_k s_k(x,u) \tag{23}$$

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

The next lemma [(3)] states that a HONNF of the form in Eq. (35)

can approximate the indicator function $I^{l_1,...,l_n}_{j_1,...,j_{n+m}}$.

Lemma 1. *Consider the indicator function* $I^{l_1,...,l_n}_{j_1,...,j_{n+m}}$ *and the family of the HONNFs* $N(x,u;w,L)$.

Then for any $\epsilon > 0$ *there is a vector of weights* $w^{j_1,...,j_{n+m};l_1,...,l_n}$

and a number of $L^{j_1,...,j_{n+m};l_1,...,l_n}$ *high order connections such that*

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

where $\bar{\mathrm{y}} \equiv \mathrm{y}$ *if assumption (A.1) is valid*

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

Let us now keep $L^{j_1,...,j_{n+m};l_1,...,l_n}$ constant, i.e. let us preselect the number of

high order connections, and let us define the optimal weights of the HONNF

with $L^{j_1,...,j_{n+m};l_1,...,l_n}$ high order connections as follows

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

and the modelling error as follows

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

It is worth noticing that from Lemma 1 we have that

$$\sup_{(x,u)\in\bar{y}} \left|\nu^{l_1,...,l_n}_{j_1,...,j_{n+m}}(x,u)\right|$$

can be made arbitrarily small by simply selecting appropriately the number of

high order connections. Using the approximation Lemma 1

it is natural to approximate system in Eq. (31) by the following dynamical system

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

Let now $\chi^t(\chi^0,u^t)$ denote the solution in Eq. (31) given that the initial state

at $t=0$ is equal to χ^0 and the input is u^t. Similarly we define $z^t(z^0,u^t)$.

Also let

$$\nu(z^t, u^t) = \sum (\bar{x}^{l_1,\ldots,l_n}_{j_1,\ldots,j_{n+m}}(x,u) \times \nu^{l_1,\ldots,l_n}_{j_1,\ldots,j_{n+m}}(z^t, u^t)) \tag{24}$$

Then, it can be easily shown that

$$z^t(z^0, u^t) = \chi^t(z^0, u^t) + \nu(z^t, u^t) \tag{25}$$

Note now that from the approximation Lemma 1 and the definition of $\nu(z^t, u^t)$ we have that modeling error can be made arbitrarily small provided that (z^t, u^t) remain in a compact set (e.g. $\bar{y}$).

Theorem 1 ((3)). *,[(4)] Consider the FDS ($R^{l_1,\ldots,l_n}_{j_1,\ldots,j_{n+m}}$) and suppose that system in Eq. (27) is its underlying system.*

*Assume that either assumptions (**A.1**) or (**A.2**) hold.*

Also consider the RHONN in [(4)].

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

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

where

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

5.1 Fuzzy system description using rule firing indicator functions and HONNF

In this section, we are briefly introducing the representation of fuzzy systems using the rule firing indicator functions (RFIF),or simply indicator functions (IF), which is used for the development of the proposed method.

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

$$z(k) = f(x(k), u(k)) \tag{26}$$

where $f(\cdot)$ is a continuous function and k denotes the temporal variable.

In case the system is dynamic the above equation could be replaced by the following differential equation

$$\dot{x}(k) = f(x(k), u(k)) \tag{27}$$

By setting $y(k) = [x(k), u(k)]$, Eq. (26) may be rewritten as follows

$$z(k) = f\left(y(k)\right) \tag{28}$$

with $y \subset R^{m+n}$

In case f in (28) is unknown we may wish to approximate it by using a fuzzy representation. In this case both $y(k) = [x(k), u(k)]$ and $z(k)$ are initially replaced by fuzzy linguistic variables.

Experts or data depended techniques may determine the form of the membership functions of the fuzzy variables and fuzzy rules will determine the fuzzy relations between $y(k)$ and $u(k)$.

Sensor input data, possibly noisy and imprecise,enter the fuzzy system, are fuzzified, are processed by the fuzzy rules and the fuzzy implication engine and are in the sequel defuzzified to produce the estimated $z(k)$ [(1)], [(2)].

We assume here that a Mamadani type fuzzy system is used.

Let now $\Omega^{l_1,l_2,...,l_n}_{j_1,j_2,...,j_{n+m}}$ be defined as the subset

of (x,u) pairs, belonging to the $(j_1,j_2,...,j_{n+m})^{th}$ input fuzzy patch

and pointing - through the vector field $f(\cdot)$ - to the subset of $z(k)$,

which belong to the $(j_1,j_2,...,j_{n+m})^{th}$ output fuzzy patch.

In other words, $\Omega^{l_1,l_2,...,l_n}_{j_1,j_2,...,j_{n+m}}$ contains input value pairs that are associated

through a fuzzy rule with output values.

According to the above notation the Indicator Function (IF) connected to $\Omega^{l_1,l_2,...,l_n}_{j_1,j_2,...,j_{n+m}}$

is defined as follows:

$$I^{l_1,...,l_n}_{j_1,...,j_{n+m}}(x(k),u(k)) = \begin{cases} \alpha \; if \; (x(k),u(k)) \in \Omega^{l_1,...,l_n}_{j_1,...,j_{n+m}} \\ 0 \qquad\qquad otherwise \end{cases} \tag{29}$$

where α denotes the firing strength of the rule.

Define now the following system

$$z(k) = \sum \bar{z}^{l_1,...,l_n}_{j_1,...,j_{n+m}} \times I^{l_1,...,l_n}_{j_1,...,j_{n+m}}(x(k),u(k)) \tag{30}$$

Where $\bar{z}^{l_1,...,l_n}_{j_1,...,j_{n+m}} \in R^n$ be any constant vector consisting of the centers of

the membership functions of each output variable z_i and $I^{l_1,...,l_n}_{j_1,...,j_{n+m}}(x(k),u(k))$

is the IF.

Then, according to [(3)], [(4)] the system in (30) is a generator for

the fuzzy system (FS).

It is obvious that Eq. (30) can be also valid for dynamic systems.

In its dynamical form it becomes

$$\dot{x}(k) = \sum \bar{x}^{l_1,...,l_n}_{j_1,...,j_{n+m}} \times I^{l_1,...,l_n}_{j_1,...,j_{n+m}}(x(k),u(k)) \tag{31}$$

Where $\bar{x}^{l_1,...,l_n}_{j_1,...,j_{n+m}} \in R^n$ be again any constant vector consisting of

the centers of fuzzy partitions of every variable x_i and $I^{l_1,...,l_n}_{j_1,...,j_{n+m}}(x(k), u(k))$

is the IF.

Based on the fact that functions of high order neurons are capable of

approximating discontinuous functions [(3)] and [(4)]

use high order neural network functions HONNFs in order to approximate the

IF $I^{l_1,...,l_n}_{j_1,...,j_{n+m}}$.

A HONNF is defined as:

$$N(x(k), u(k); w, L) = \sum_{hot=1}^{L} w_{hot} \prod_{j \in I_{hot}} \Phi_j^{d_j(hot)} \tag{32}$$

where $I_{hot} = \{I_1, I_2, ..., I_L\}$ is a collection of L not-ordered subsets

of $\{1, 2, ..., m+n\}$, $d_j(hot)$ are non-negative integers, Φ_j are sigmoid functions

of the state or the input, which are the elements of the following vector :

$\Phi = \left[\Phi_1 \ \dots \ \Phi_n \ \Phi_{n+1} \ \dots \ \Phi_{m+n}\right]^T =$

$$= \left[S(x_1) \ \dots \ S(x_n) \ S(u_1) \ \dots S(u_m)\right]^T \tag{33}$$

where

$$S(x) = a\frac{1}{1+e^{-\beta x}} - \gamma \tag{34}$$

and $w := [w_1 \cdots w_L]^T$ are the HONNF weights.

Eq. (32) can also be written

$$N(x(k), u(k); w, L) = \sum_{hot=1}^{L} w_{hot} S_{hot}(x(k), u(k)) \tag{35}$$

where $S_{hot}(x(k), u(k))$ are high order terms of sigmoid functions of the state and/or input.

References

[1] L. Wang, *Adaptive Fuzzy Systems and Control*, Prentice Hall, Englewood Cliffs, NJ, 1994.

[2] K. M. Passino and S. Yurkovich, *Fuzzy Control*, Addison Wesley Longman, Menlo Park, CA, 1998. S. Tong, T. Chai, "Direct adaptive control and robust analysis for unknown multivariable nonlinear systems with fuzzy logic systems", *Fuzzy Sets and Systems*, Vol. 106, pp. 309-319, 1999.

[3] E. B. Kosmatopoulos and M. A. Christodoulou, "Recurrent neural networks for approximation of fuzzy dynamical systems", *Int. Journal of Intelligent Control and Systems*, vol. 1, pp. 223-233, 1996.

[4] 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.

Part VI

Direct adaptive neuro-fuzzy control

6 Direct adaptive neuro-fuzzy control

6.1 Problem formulation and neuro-fuzzy representation

Problem formulation

We consider nonlinear dynamical systems of the Brunovski canonical form

$$\dot{x} = A_c x + b_c[f(x) + g(x) \cdot u] \tag{36}$$

where the state $x \in R^n$ is assumed to be completely measured,

the control input $u \in R$, f and g are scalar nonlinear functions of the state

being only involved in the dynamic equation of x_n.

Also,

$$A_c = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots \\ \cdots & \cdots & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix}$$

and

$b_c = \begin{bmatrix} 0 \cdots 0 \ 1 \end{bmatrix}^T$

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 (36), is very difficult

or even impossible to be solved since the f, g are assumed to be completely unknown.

To overcome this problem we assume that the unknown plant can be described by

the following model arriving from a neuro-fuzzy representation described below.

$$\dot{x} = A_c x + b_c[XW^* S(x) + X_1 W_1^* S_1(x) u] \tag{37}$$

where the weight values W^* and W_{1n}^* are unknown.

Therefore, the state regulation problem is analyzed for the system (37) instead of (36). 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, X and X_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 (36), to guarantee the existence and uniqueness of solution for any finite initial condition and $u \in U$.

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, $|x(T)| < \infty$.

The f, g are continuous with respect to their arguments and satisfy a local Lipchitz condition so that the solution $x(t)$ of (36) is unique for any finite initial condition and $u \in U$.

Neuro-fuzzy representation

We are using a fuzzy approximation of the system in (36), which uses two fuzzy subsystem blocks for the description of $f(x)$ and $g(x)$ as follows

$$f(\chi) = \sum \bar{f}_{j_n}^{l_1,\ldots,l_n} \times I_{j_n}^{l_1,\ldots,l_n}(\chi) \tag{38}$$

$$g(\chi) = \sum \bar{g}_{j_n}^{l_1,\ldots,l_n} \times I_1{}_{j_n}^{l_1,\ldots,l_n}(\chi) \tag{39}$$

where the summation is carried out over the number of all available fuzzy rules, I, I_1 are appropriate IF and the meaning of indices $\bullet_{j_1,\ldots,j_n}^{l_1,\ldots,l_n}$

has already been described in Section 4.

According to Section 4, every IF can be approximated with the help of a suitable HONNF.

Therefore, every I, I_1 can be replaced with a corresponding HONNF as follows

$$f(\chi) = \sum \bar{f}_{j_n}^{l_1,\ldots,l_n} \times N_{j_n}^{l_1,\ldots,l_n}(\chi) \tag{40}$$

$$g(\chi) = \sum \bar{g}_{j_n}^{l_1,\ldots,l_n} \times N_1{}_{j_n}^{l_1,\ldots,l_n}(\chi) \tag{41}$$

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 (40),(41) are carried out over the number of the corresponding output partitions.

Therefore, the system of (36) is replaced by the following equivalent Brunovsky form Fuzzy - Recurrent High Order Neural Network (F-RHONN), which depends on the centers of the fuzzy output partitions $\bar{f}_l$ and $\bar{g}_l$

$$\dot{\hat{\chi}} = A_c\hat{\chi} + b_c\left[\sum_{l=1}^{Npf} \bar{f}_l \times N_l(\chi) + \left(\sum_{l=1}^{Npg} \bar{g}_l \times N_{1l}(\chi)\right) u\right] \tag{42}$$

where Npf and Npg are the number of fuzzy partitions of f and g respectively.

Or in a more compact form

$$\dot{\hat{\chi}} = A_c\hat{\chi} + b_c[XWS(\chi) + X_1W_1S_1(\chi)u] \tag{43}$$

where X, X_1 are matrices containing the centers 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
(35) and (42).

The dimensions and the contents of all the above matrices are chosen
so that both $XWS(\chi)$ and $X_1W_1S_1(\chi)$ are scalar. For notational simplicity we also
assume that all output fuzzy variables are partitioned to the same number,
m, of partitions.

Under these specifications X is a $1 \times m$ vector of the form

$$X = \left[\bar{f}_1 \ \bar{f}_2 \cdots \bar{f}_m\right]$$

where $\bar{f}_p$ denotes the center or the fuzzy p-th partition of f.

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 mentioned in the introduction.

Also, $S(\chi) = \left[s_1(\chi) \ldots s_k(\chi)\right]^T$,
where each $s_i(\chi)$ with $i = \{1, 2, ..., k\}$, is a high order combination
of sigmoid functions of the state variables and W is a $m \times k$
matrix with neural weights.

W can be also written as a collection of column vectors W^l,
that is $W = \left[W^1 W^2 \cdots W^l\right]$,
where $l = 1, 2, ..., k$.

Similarly, X_1 is a $1 \cdot m$ raw vector of the form

$$X_1 = \left[\bar{g}_1 \ \bar{g}_2 \cdots \bar{g}_m\right],$$

where $\bar{g}_k$ denotes the center or the k-th partition of g. W_1, $S_1(\chi)$ have the same dimensions as W, $S(\chi)$ respectively.

References

[1] L. Wang, *Adaptive Fuzzy Systems and Control*, Prentice Hall, Englewood Cliffs, NJ, 1994.

[2] K. M. Passino and S. Yurkovich, *Fuzzy Control*, Addison Wesley Longman, Menlo Park, CA, 1998.

[3] J. S. R. Jang, "ANFIS: Adaptive-network-based fuzzy inference system", *IEEE Trans. Syst. Man. Cyber.*, vol. 23, pp. 665-684, 1993.

[4] C.T. Lin, "A neural fuzzy control system with structure and parameter learning", *Fuzzy Sets and Systems*, vol. 70, pp. 183-212, 1995.

[5] G. A. Rovithakis and M. A. Christodoulou, "Adaptive control of unknown plants using dynamical neural networks", *IEEE Trans. Syst, Man, Cybern.*, vol. 24, , pp. 400-412 Mar. 1994.

[6] Y. Diao, K.M. Passino, Adaptive Neural/Fuzzy Control for Interpolated Nonlinear Systems, *IEEE Trans. on Fuzzy Systems*, Vol. 10, No. 5, pp. 583-595, Oct. 2002.

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

[8] S. Tong, T. Chai, "Direct adaptive control and robust analysis for unknown multivariable nonlinear systems with fuzzy logic systems", *Fuzzy Sets and Systems*, Vol. 106, pp. 309-319, 1999.

[9] Y. Yang, "Direct robust adaptive fuzzy control (DRAFC) for uncertain nonlinear systems using small gain theorem", *Fuzzy Sets and Systems*, Vol. 151, May 2004, pp. 7997, May 2004.

[10] M. Chemachema,K. Belarbi, "Robust direct adaptive controller for a class of nonlinear systems based on neural networks and fuzzy logic system", *International Journal on Artificial Intelligence Tools*, Vol. 16, pp. 553-560, 2007.

[11] 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, Springer Verlag London Limited, 2000.

Part VII

Adaptive regulation with modeling error effects

7 Adaptive regulation with modeling error effects

In this section we present a solution to the adaptive regulation problem and investigate the modeling error effects when the dynamical equations have the Brunovski canonical form. Assuming the presence of modeling error the unknown system can be written as (37). The regulation of the system can be achieved by selecting the control input to be

$$u = -\frac{XWS(x) + \upsilon}{X_1 W_1 S_1(x)} \tag{44}$$

with

$$\upsilon = -kx \tag{45}$$

where k is a vector of the form $k = [k_n \cdots k_2 \; k_1] \in R^n$ be such that all roots of the polynomial $h(s) = s^n + k_1 s^{n-1} + \cdots + k_n$ are in the open left half-plane.

Define now, the regulation error as

$$\xi = -x \tag{46}$$

After substituting Eq. (44) to the $n-th$ state equation of Eq. (37)

and straightforward manipulations we have that

$$\dot{x} = Ax + b_c[X^* W^* S(x) + X_1^* W_1^* S_1(x)u] \tag{47}$$

To this end add and subtract to the above error equation the terms $b_c kx$ and define: $\tilde{W} = W - W^*$ and $\tilde{W}_1 = W_1 - W_1^*$.

$$\dot{x} = Ax + b_c kx - b_c kx + b_c[X^* W^* S(x) + X_1^* W_1^* S_1(x)u]$$

$$= [A - b_c k]x + b_c[X^* W^* S(x) + X_1^* W_1^* S_1(x)u - X_1^* W_1^* S_1(x)u - X^* W^* S(x)]$$

$$= L_c x + b_c[X^* W^* S(x) + X_1^* W_1^* S_1(x)u - X_1^* W_1^* S_1(x)u - XWS(x)]$$

where $\Lambda_c = A_c - b_c k$ is a matrix with its eigenvalues on the left half plane.

$\tilde{W} = W - W^*$ and $\tilde{W}_1 = W_1 - W_1^*$. W and W_1 are estimates of

W^* and W_1^* respectively and are obtained by update laws which are

to be designed in the sequel.

After substituting Eq. (49), (50) becomes

$$\dot{x} = \Lambda_c x + b_c[-X^* \tilde{W} S(x) - \tilde{X} WS(x) - X_1^* \tilde{W}_1 S_1(x)u - \tilde{X}_1 W_1 S_1(x)u] \tag{48}$$

$$= \Lambda_c x - b_c[X^* \tilde{W} S(x) + \tilde{X} WS(x) + X_1^* \tilde{W}_1 S_1(x)u + \tilde{X}_1 W_1 S_1(x)u]$$

Define now, the regulation error as

$$\xi = -x \tag{49}$$

So,by manipulating $\xi = -x$ to the previous equation, we have :

$$-\dot{\xi} = \Lambda_c[-\xi] - b_c[X^* \tilde{W} S(x) + \tilde{X} WS(x) + X_1^* \tilde{W}_1 S_1(x)u + \tilde{X}_1 W_1 S_1(x)u] \tag{50}$$

$$\dot{\xi} = \Lambda_c[\xi] + b_c[X^* \tilde{W} S(x) + \tilde{X} WS(x) + X_1^* \tilde{W}_1 S_1(x)u + \tilde{X}_1 W_1 S_1(x)u]$$

To continue, consider the **Lyapunov** candidate function

$$V(\xi, \hat{x}, \tilde{X}, \tilde{W}, \tilde{X}_1, \tilde{W}_1) = \frac{1}{2}\xi^T P\xi + \frac{1}{2}tr\{\tilde{X}^T \tilde{X}\} + \frac{1}{2}tr\{\tilde{W}^T \Delta \tilde{W}\} + \frac{1}{2}tr\{\tilde{X}_1^T \tilde{X}_1\} + \frac{1}{2}tr\{\tilde{W}_1^T \Delta_1 \tilde{W}_1\} \tag{51}$$

Where $P > 0$ is chosen to satisfy the **Lyapunov** equation

$$P\Lambda_c + \Lambda_c^T P = -I$$

and matrices Δ and Δ_1 are both diagonal $n \cdot m \times n \cdot m$ and defined as follows:

$\Delta = diag\{(|\bar{f}_1^{1*}|, |\bar{f}_2^{1*}|, \ldots, |\bar{f}_m^{1*}|), (|\bar{f}_1^{2*}|, |\bar{f}_2^{2*}|, \ldots, |\bar{f}_m^{2*}|), \ldots,$
$(|\bar{f}_1^{m*}|, |\bar{f}_2^{m*}|, \ldots, |\bar{f}_m^{m*}|)\}$

and

$\Delta_1 = diag\{(|\bar{g}_1^{1,1*}|, |\bar{g}_2^{1,1*}|, \ldots, |\bar{g}_m^{1,1*}|),$

$(|\bar{g}_1^{2,2*}|, |\bar{g}_2^{2,2*}|, \ldots, |\bar{g}_m^{2,2*}|), \ldots,$

$(|\bar{g}_1^{m,m*}|, |\bar{g}_2^{m,m*}|, \ldots, |\bar{g}_m^{m,m*}|)\}$

Thus $\Delta \geq 0$ and $\Delta_1 \geq 0$.

Taking the derivative of the **Lyapunov** function candidate we get

$$\dot{V} = \tfrac{1}{2}\dot{\xi^T} P\xi + \tfrac{1}{2} tr\{\dot{\tilde{X}}^T \tilde{X}\} + \tfrac{1}{2} tr\{\dot{\tilde{W}}^T \Delta \tilde{W}\} + \tfrac{1}{2} tr\{\ \dot{\tilde{X}_1}^T \tilde{X}_1\} + \tfrac{1}{2} tr\{\ \dot{\tilde{W}_1}^T \Delta_1 \tilde{W}_1\}$$

In this point,we can show that :

$\dot{\xi^T} P\xi = \tfrac{1}{2}\dot{\xi^T} P\xi + \tfrac{1}{2}\ \xi^T P\dot{\xi}$

$= \tfrac{1}{2}[\xi^T L_c^T + [b_c[X^* \tilde{W} S(x) + \tilde{X} W S(x) +$

$+ X_1^* \tilde{W}_1 S_1(x) u + \tilde{X}_1 W_1 S_1(x) u]^T]] P\xi +$

$+ \tfrac{1}{2}[\xi^T P[L_c \xi + b_c[X^* \tilde{W} S(x) + \tilde{X} W S(x) +$

$+ X_1^* \tilde{W}_1 S_1(x) u + \tilde{X}_1 W_1 S_1(x) u]]$

So, we have:

$$\dot{\xi^T} P\xi = \tfrac{1}{2}[\xi^T L_c^T P\xi + \tfrac{1}{2}[\xi^T P L_c \xi + \tfrac{1}{2}[b_c[X^*\tilde{W}S(x) + \tilde{X}WS(x)+$$

$$+X_1^*\tilde{W}_1 S_1(x)u + \tilde{X}_1 W_1 S_1(x)u]]^T P\xi + \tfrac{1}{2}[\xi^T P b_c[X^*\tilde{W}S(x)+$$

$$\tilde{X}WS(x) + X_1^*\tilde{W}_1 S_1(x)u + \tilde{X_1} W_1 S_1(x)u]$$

$$= \tfrac{1}{2}[\xi^T[P\text{Ł}_c + \text{Ł}_c^T P]\xi + \tfrac{1}{2}[b_c[X^*\tilde{W}S(x) + \tilde{X}$$

$$WS(x) + X_1^*\tilde{W}_1 S_1(x)u + \tilde{X}_1 W_1 S_1(x)u]]^T P\xi +$$

$$\tfrac{1}{2}[\xi^T P b_c[X^*\tilde{W}S(x) + \tilde{X}WS(x)+$$

$$X_1^*\tilde{W}_1 S_1(x)u + \tilde{X_1} W_1 S_1(x)u]$$

But,we know that

$$P\Lambda_c + \Lambda_c^T P = -I$$

so the last equation becomes :

$$\dot{\xi^T} P\xi = -\tfrac{1}{2}\left\|\xi\right\|^2 + \tfrac{1}{2}[b_c[X^*\tilde{W}S(x)+$$

$$+\tilde{X}WS(x) + X_1^*\tilde{W}_1 S_1(x)u + \tilde{X_1} W_1 S_1(x)u]]^T P\xi +$$

$$+\tfrac{1}{2}[\xi^T P b_c[X^*\tilde{W}S(x) + \tilde{X}WS(x)+$$

$$+X_1^*\tilde{W}_1 S_1(x)u + \tilde{X_1} W_1 S_1(x)u]$$

Furthermore,we have that:

$$\tfrac{1}{2}[b_c[X^*\tilde{W}S(x) + \tilde{X}WS(x) + X_1^*\tilde{W}_1 S_1(x)u+$$

$$\tilde{X_1} W_1 S_1(x)u]]^T P\xi + \tfrac{1}{2}[\xi^T P b_c[X^*\tilde{W}S(x)+$$

$\tilde{X}WS(x) + X_1^*\tilde{W}_1S_1(x)u + \tilde{X}_1W_1S_1(x)u]]$

$= \frac{1}{2}[b_c[X^*\tilde{W}S(x)]^T + b_c[\tilde{X}WS(x)]^T +$

$+b_c[X_1^*\tilde{W}_1S_1(x)u]^T + [b_c[\tilde{X}_1W_1S_1(x)u]^T]]P\xi +$

$+\frac{1}{2}[\xi^T Pb_c[X^*\tilde{W}S(x) + \tilde{X}WS(x) + X_1^*\tilde{W}_1S_1(x)u +$

$+\tilde{X}_1W_1S_1(x)u]]$

So, with further manipulations, we have :

$= [\frac{1}{2}S(x)^T\tilde{W}^TX^{*T}b_c^T + \frac{1}{2}S(x)^TW^T\tilde{X}^Tb_c^T +$

$+\frac{1}{2}u^TS_1(x)^T\tilde{W}_1^TX_1^{*T}b_c^T +$

$+\frac{1}{2}u^TS_1(x)^TW_1^T\tilde{X}_1^{\ T}b_c^T]P\xi +$

$+\frac{1}{2}[\xi^T Pb_c[X^*\tilde{W}S(x) + \tilde{X}WS(x) +$

$X_1^*\tilde{W}_1S_1(x)u + \tilde{X}_1W_1S_1(x)u]$

$= \frac{1}{2}[\xi^T Pb_c[X^*\tilde{W}S(x) + \frac{1}{2}S(x)^T\tilde{W}^TX^{*T}b_c^TP\xi +$

$+\frac{1}{2}[\xi^T Pb_c\tilde{X}WS(x) + \frac{1}{2}S(x)^TW^T\tilde{X}^Tb_c^TP\xi +$

$+\frac{1}{2}[\xi^T Pb_cX_1^*\tilde{W}_1S_1(x)u + \frac{1}{2}u^TS_1(x)^T\tilde{W}_1^TX_1^{*T}b_c^TP\xi$

$+\frac{1}{2}[\xi^T Pb_c\tilde{X}_1W_1S_1(x)u + \frac{1}{2}u^TS_1(x)^TW_1^T\tilde{X}_1^{\ T}b_c^TP\xi$

$= \xi^T Pb_cX^*\tilde{W}S(x) + \xi^T Pb_c\tilde{X}WS(x) +$

$+\xi^T Pb_cX_1^*\tilde{W}_1S_1(x)u + \xi^T Pb_c\tilde{X}_1W_1S_1(x)u$

So,we finally have that the `derivative of V` is:

$$\dot{V} = -\tfrac{1}{2}\left\|\xi\right\|^2 + \xi^T P b_c X^* \tilde{W} S(x) + \xi^T P b_c \tilde{X} W S(x) +$$
$$+\xi^T P b_c X_1^* \tilde{W}_1 S_1(x) u + \xi^T P b_c \tilde{X}_1 W_1 S_1(x) u$$

Hence, if we choose:

$$\begin{cases} tr\{\dot{\tilde{W}}^T \Delta \tilde{W}\} = -\xi^T P b_c X^* \tilde{W} S(x) \\ tr\{\dot{\tilde{X}}^T \tilde{X}\} = -\xi^T P b_c \tilde{X} W S(x) \\ tr\{\dot{\tilde{W}}_1^T \Delta_1 \tilde{W}_1\} = -\xi^T P b_c X_1^* \tilde{W}_1 S_1(x) u \\ tr\{\dot{\tilde{X}}_1^T \tilde{X}_1\} = -\xi^T P b_c \tilde{X}_1 W_1 S_1(x) u \end{cases}$$

and using the fact that whenever $tr\{\dot{X}^T \tilde{X}\} = A\tilde{X}B$, where A is a row and B is a column vector, $\Rightarrow \dot{\tilde{X}} = A^T B^T$, we finally get that the update laws will take the form:

$$\begin{cases} \Delta \dot{W} = -[\xi^T P b_c [X^*]]^T S(x)^T \\ \dot{X} = -b_c^T P \xi S(x)^T W^T \\ \Delta_1 \dot{W}_1 = -[\xi^T P b_c [X_1^*]]^T [S_1(x) u]^T \\ \dot{X}_1 = -b_c^T P \xi u^T S_1(x)^T W_1^T \end{cases} \tag{52}$$

We write $X^{*T} = \Delta\{sgn(X^*)\}^T$ and $X_1^{*T} = \Delta_1\{sgn(X_1^*)\}^T$

where:

$sgn(X^*) = diag\{sgn(X^{1*}), sgn(X^{2*}), \ldots, sgn(X^{n*})\}$

where:

$sgn(X^{i*}) = [sgn(\bar{f_1}^{i,*}), sgn(\bar{f_2}^{i,*}), \ldots, sgn(\bar{f_m}^{i,*})]$

and:

$sgn(X_1^*) = diag\{sgn(X_1^{1*}), sgn(X_1^{2*}), \ldots, sgn(X_1^{n*})\}$

where:

$sgn(X_1^{i*}) = [sgn(\bar{g_1}^{i,i,*}), sgn(\bar{g_2}^{i,i,*}), \ldots, sgn(\bar{g_m}^{i,i,*})]$

Then equations (52) become:

$$\begin{cases} \dot{W} = -\gamma_1 sgn(X^*)^T b_c^T P \xi S(x)^T \\ \dot{X} = -\gamma_3 b_c^T P \xi S(x)^T W^T \\ \dot{W_1} = -\gamma_2 sgn(X_1^*)^T b_c^T P \xi u^T S_1(x)^T \\ \dot{X}_1 = -\gamma_4 b_c^T P \xi u^T S_1(x)^T W_1^T \end{cases} \tag{53}$$

where γ_1,γ_2,γ_3,γ_4 are constants related to the bilinear neuro fuzzy adaptive control problem.These constants are chosen, after careful selection, to be 0.1 or less.

Proof. The update laws (53) are implementable, provided we know the signs of the partitions, which is a very reasonable assumption. However the centers of the partitions are automatically selected by our algorithm optimally.

Using the above **Lyapunov** function candidate V and proving that $\dot{V} \leq 0$ all properties of the theorem are assured [(3)].

Hence and since $u, \dot{\xi} \in L_\infty$, the sigmoidals are bounded by definition, $\tilde{W}, \tilde{W}_1 \in L_\infty$, so since $\xi \in L_2 \cap L_\infty$ and $\dot{\xi} \in L_\infty$, applying Barbalat's Lemma 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.

Hence we have that

$$\lim_{t\to\infty} x(t) = -\lim_{t\to\infty} \xi(t) = 0$$

Thus,

$$\lim_{t\to\infty} x(t) = 0.$$

Remark 1. We can't conclude anything about the convergence of the synaptic weights W and W_1 to their optimum values W^* and W_1^* respectively from the above analysis.

The existence of signal u is associated with conditions which guarantee that $X_1 W_1 S_1 \neq 0$. It can be shown that using appropriate projection method [(3)], the weight updating laws can be modified so that the existence of the control signal can be assured.

However, this development is not presented in this thesis,but the relevant results along with the ones given here are to be presented shortly in a forthcoming work.

References

[1] L. Wang, *Adaptive Fuzzy Systems and Control*, Prentice Hall, Englewood Cliffs, NJ, 1994.

[2] J. S. R. Jang, "ANFIS: Adaptive-network-based fuzzy inference system", *IEEE Trans. Syst. Man. Cyber.*, vol. 23, pp. 665-684, 1993.

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

[4] S. Tong, T. Chai, "Direct adaptive control and robust analysis for unknown multivariable nonlinear systems with fuzzy logic systems", *Fuzzy Sets and Systems*, Vol. 106, pp. 309-319, 1999.

[5] 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, Springer Verlag London Limited, 2000.

Part VIII

Conclusions

A direct adaptive control scheme was considered in this thesis, aiming at the regulation of non linear unknown plants of Brunovsky canonical form with the presence of modeling errors.

The approach is based on a new Neuro-Fuzzy Dynamical Systems definition, which uses the concept of Fuzzy Adaptive Systems (FAS) operating in conjunction with High Order Neural Network Functions (R-HONNFs).

Since the plant is considered unknown, we first propose its approximation by a special form of a Brunovsky type fuzzy dynamical system (FDS) where the fuzzy rules are approximated by appropriate HONNFs.

This practically transforms the original unknown system into a neuro-fuzzy 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.

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.

I hereby affirm that I composed this work independently and used no other than the specified sources and tools and that I marked all quotes as such.