Intelligent Control Systems
(0640734)

Lecture (8)

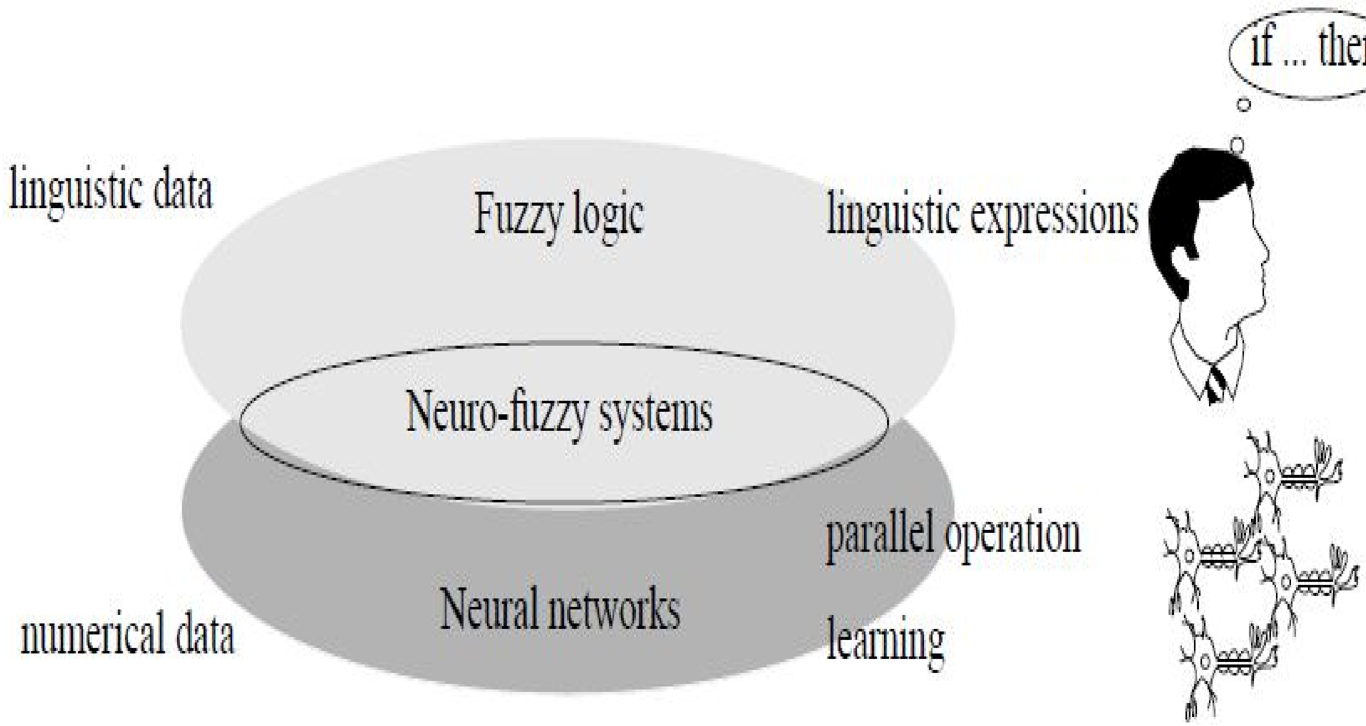
Neuro-Fuzzy Systems in Control

Prof. Kasim M. Al-Aubidy
Philadelphia University-Jordan

Introduction:

➢ Neuro-fuzzy systems combine the properties and the benefits of NNs and FL. This lecture will cover:

1. the background of neuro-fuzzy systems, and
2. neural fuzzy logic inference systems which are capable of extracting fuzzy rules and sets from numerical input-output information.



Introduction:

- Fuzzy systems transfer the vague fuzzy form of human reasoning to mathematical systems. The use of *IF–THEN rules in fuzzy systems gives us the* possibility of easily understanding the information modeled by the system. In most of the fuzzy systems the knowledge is obtained from human experts. This method has a great disadvantage; not every human expert can (want to) share their knowledge.
- ANNs can learn from experience but most of the topologies do not allow us to clearly understand the information learned by the networks.
- ANNs are incorporated into fuzzy systems to form neuro-fuzzy systems, which can acquire knowledge automatically by learning algorithms of NNs.
- Neuro-fuzzy systems have the advantage over fuzzy systems that the acquired knowledge is more meaningful to humans.
- Clustering is another technique used with neuro-fuzzy systems, it is used to initialize unknown parameters such as the number of fuzzy rules or the number of membership functions for the premise part of the rules. They are also used to create dynamic systems and update the parameters of the system.

- The operation of the neuro-fuzzy system is expressed as linguistic fuzzy expressions and learning schemes of NNs are used to learn the system.
- Neuro-fuzzy systems allow incorporation of both numerical and linguistic data into the system.
- The neuro-fuzzy system is also capable of extracting fuzzy knowledge from numerical data.
- The neuro-fuzzy systems can be divided into two main groups:
  1. Neural fuzzy inference systems, and
  2. Fuzzy neural networks.

1. Neural fuzzy inference systems:

- The origin of neural fuzzy inference systems is to incorporate neural concepts, such as learning and parallelism, into fuzzy logic inference systems.
- The fuzzy inference can be implemented in two ways:
  1. To use one network which realize the whole fuzzy inference.
  2. Each fuzzy rule is realized using a neural network, when the fuzzy inference is the result of several neural networks.

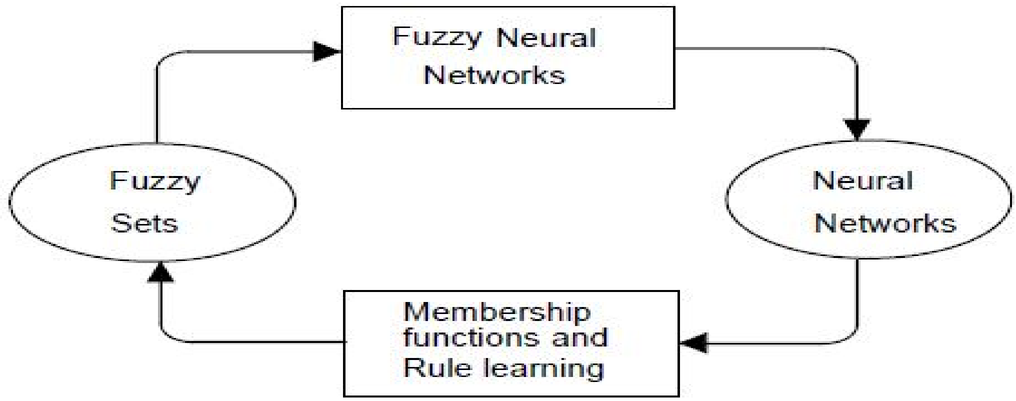
2. Fuzzy neural networks:

- The fuzzy ideas are incorporated into NNs.
- The approach replaced the weighted sum of the neuron by an corresponding fuzzy operation. The operation of the neuro-fuzzy system was exactly the same as the NN.
- The fuzzy neural networks consist of two components in same system:
  1. a fuzzy system, and
  2. a neural network.
- The fuzzy system can be either a fuzzy inference block which converts linguistic information for the NN, or the NN can drive the fuzzy inference block.
- A common approach to neuro-fuzzy systems is to fuzzify the learning algorithms of different NNs paradigms. For example, the learning rate coefficient of Kohonen's self-organizing map is treated as a fuzzy membership value of the current input sample in the class of each neuron.

How ANNs and FSs can interact?

There are two main ways in which ANNs and fuzzy systems can interact:

1. The fuzzification of NNs; weight level, TF level, learning algorithm level.
2. Giving fuzzy systems features of NNs; the ANN is used to learn membership functions or rules for a given fuzzy system.



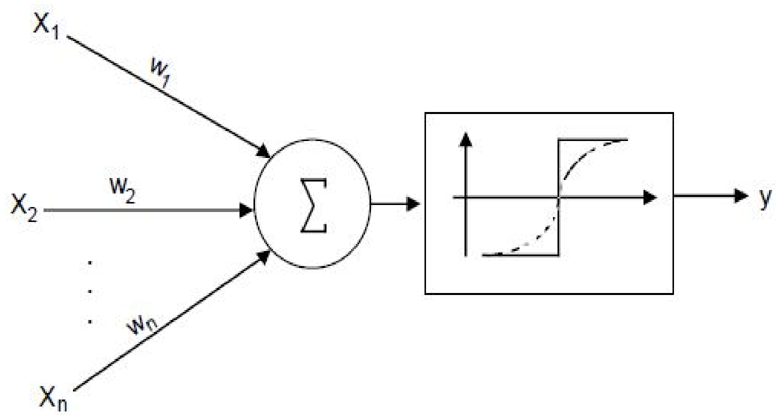
Classes of Neuro Fuzzy Systems:

1. Co-operative: Neural algorithms learn membership functions or rules or both.
2. Concurrent: the two techniques are applied after one another as pre- or post-processing.
3. Hybrid: the fuzzy system is represented as a NN to take advantage of learning algorithms inherited from ANNs.

Fuzzy Neural Networks:

Integrating NNs and fuzzy systems into a single new soft computing model gives hopes of exploiting their complementary nature by reinforcing the good points and avoiding their respective shortcomings.

Fuzzy Neurons:

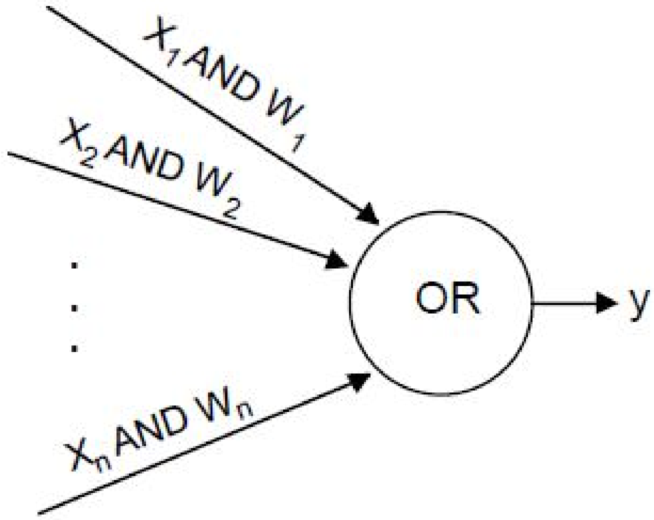


- Fuzzy models of ANNs can be constructed by using fuzzy operations at the single neuron level.
- The fuzzy operations that have been used for that purpose are the union and the intersection of fuzzy sets and, more generally, t-norms and t-conorms.
- A variety of fuzzy neurons can be obtained by applying fuzzy operations to connection weights, to aggregation functions or to both of them.
- A step toward the fuzzification of an ANN can be done by considering other forms A of the aggregation function according to the more general equation:

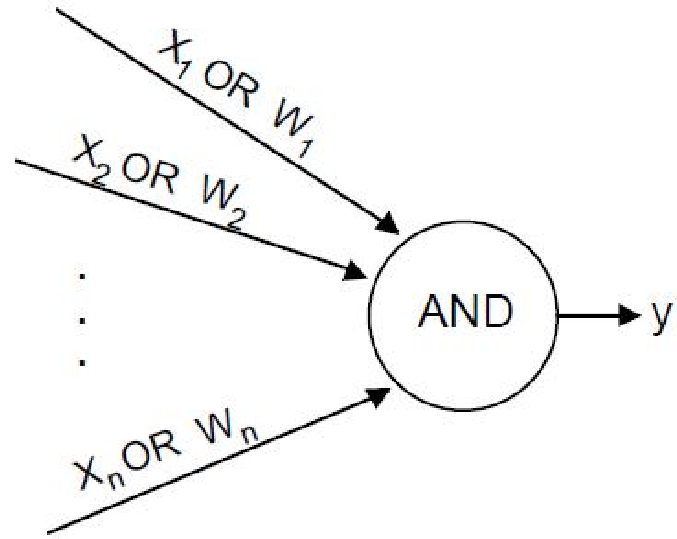
$$y = g\,(A(w;\, x))$$

where g is the transfer function and y is the output signal of the neuron.

OR Fuzzy Neuron



AND Fuzzy Neuron

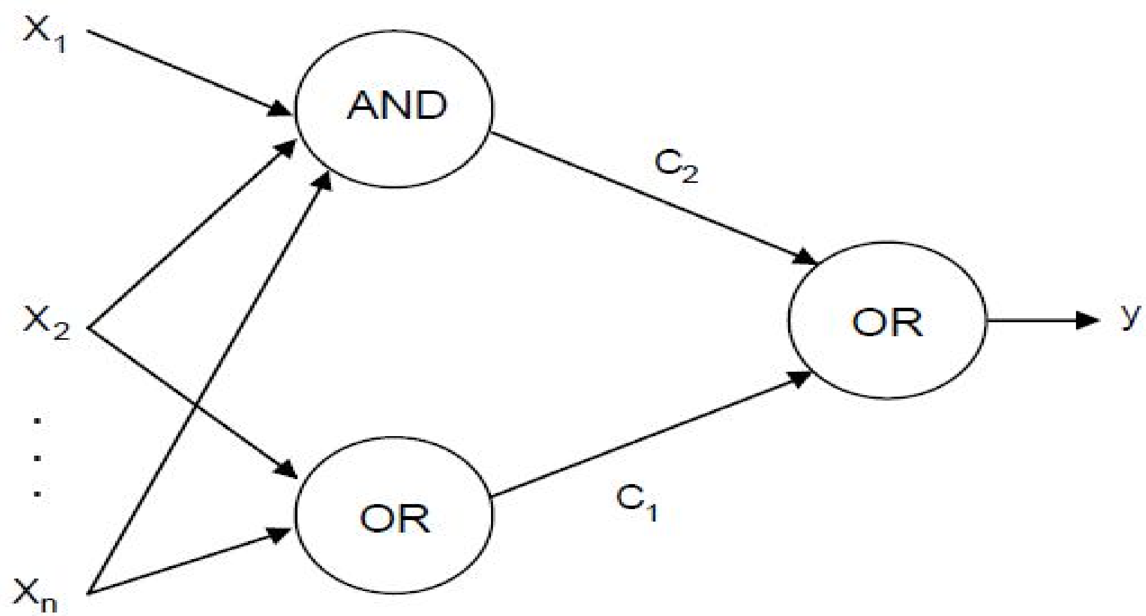


$$y = OR(x_1\ AND\ w_1, x_2\ AND\ w_2, \ldots, x_n\ AND\ w_n)$$

$$y = AND(x_1\ OR\ w_1, x_2\ OR\ w_2, \ldots, x_n\ OR\ w_n)$$

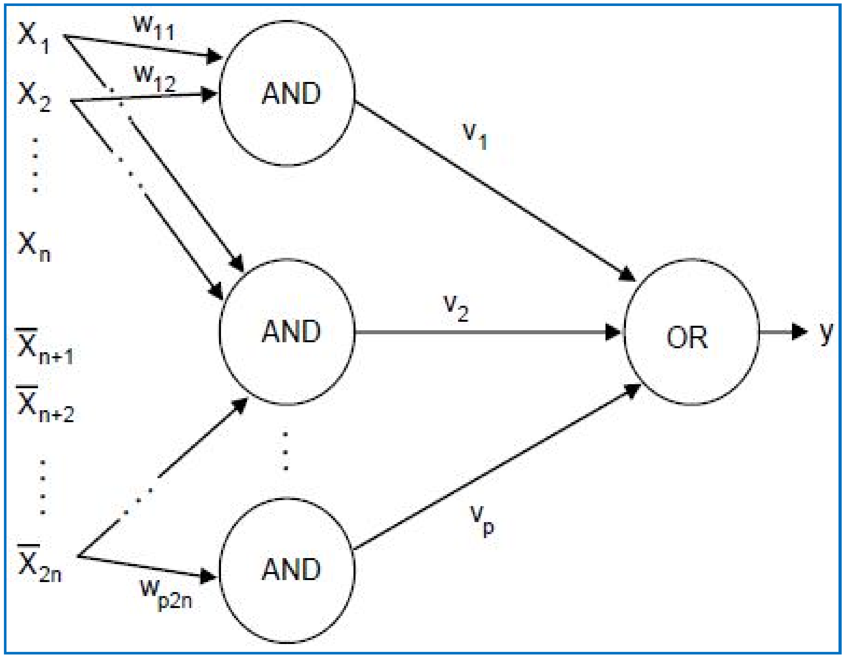
The OR/AND Neuron:

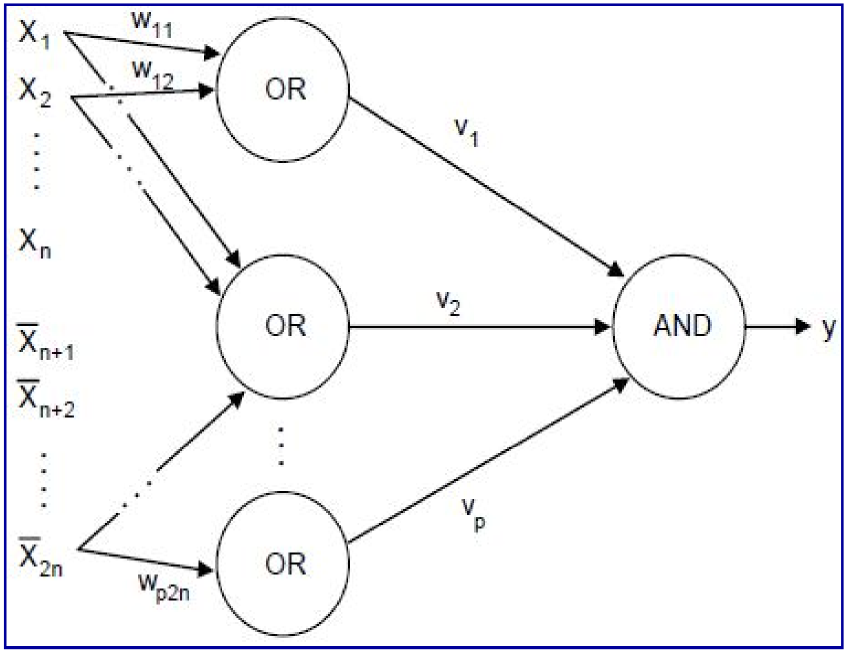
- It is a combination of the AND and OR neurons into a two-layer structure.
- The behavior of the net can be modulated by suitably weighting the output signals from the OR or the AND parts through setting or learning the connection weights c1 and c2.
- The limiting cases are;
  - ✓ c1 = 0 and c2 = 1: the system reduces itself to a pure AND neuron.
  - ✓ c1 = 1, and c2 = 0:, the system reduces itself to a pure OR neuron.



Multilayered Fuzzy Neural Networks:

- Fuzzy neurons can be assembled together into multilayered networks.
- Since the neurons are in general different, the construction gives rise to non-homogeneous neural networks.
- For example; the first NN is composed of a hidden layer with p neurons (AND type) and an output layer with a single OR neuron. The input is constituted by 2n values.
- The second possibility is to have OR neurons in the hidden layer and a single AND neuron in the output layer.





Learning in Fuzzy Neural Networks:

- Supervised learning in fuzzy neural networks consists in modifying their connection weights in a such a manner that an error measure is reduced by using sets of known input/output data pairs.
- Another important requirement is that the network thus obtained be capable of generalization, i.e. its performance should remain acceptable when it is presented with new data.
- A single fuzzy neuron adapts its connection weights in order to reduce a measure of error averaged over the training set:

$$\mathbf{w}^{t+1} = \mathbf{w}^{t} + \Delta\mathbf{w}^{t}$$

where the weight change is a given function F of the difference between the target response d and the calculated node output y:

$$\Delta\mathbf{w}^{t} = F(|d^{t} - y^{t}|)$$

- A criterion function E is defined such that it gives a measure of how well the fuzzy network maps input data into the corresponding output. A common form for E is the sum of the squared errors:

$$E(\mathbf{w}) = 1/2\sum_{k=1}^{n}(d_k - y_k)^2 \qquad \Delta w_{i,j} = -\eta\,\frac{\partial E}{\partial w_{i,j}}$$

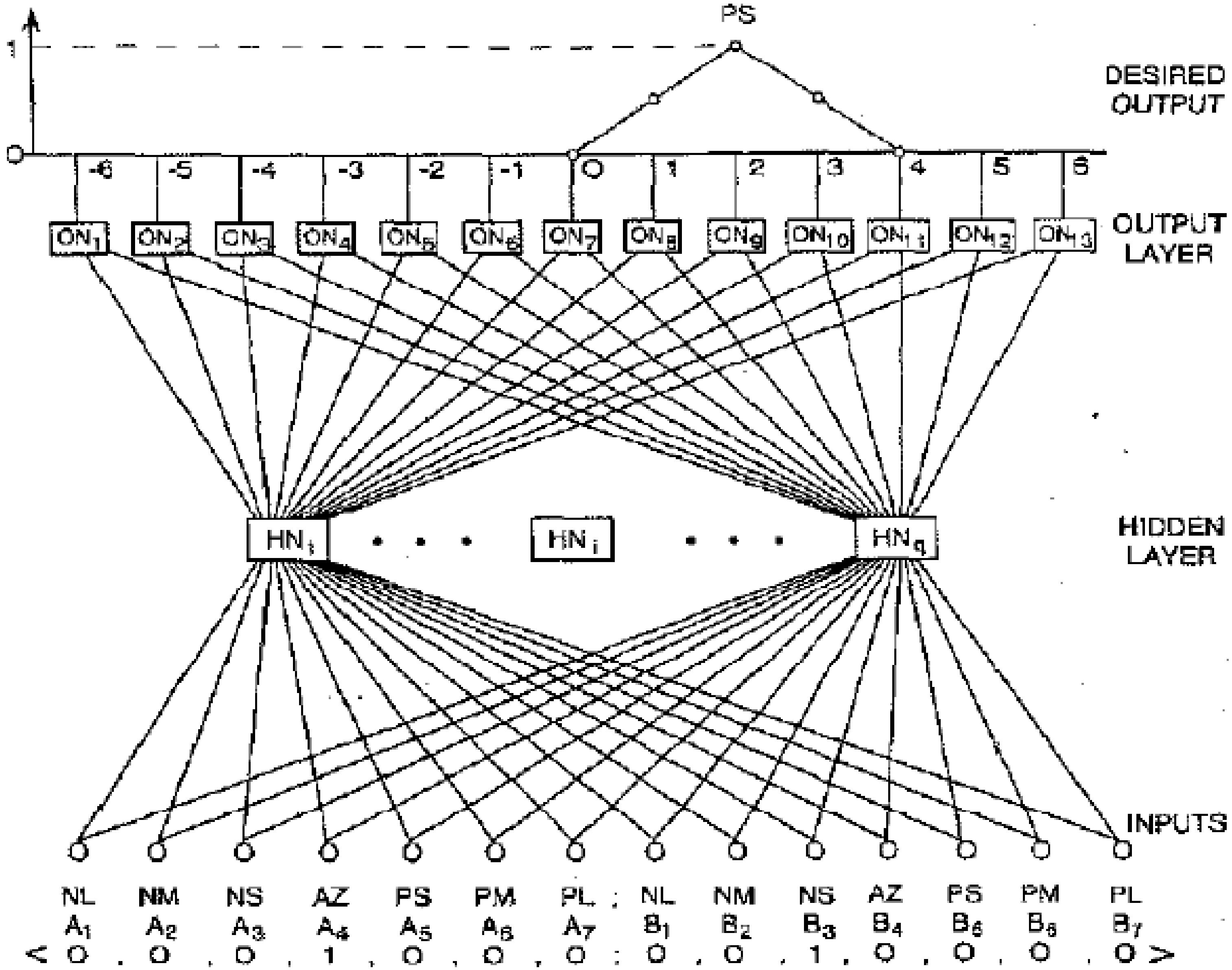
Example: ANN approximating a Fuzzy Inference Engine:

A FLC with 2 i/ps (e, Δe) and single o/p (V). Each i/p variable has 7 fuzzy sets, and the o/p has 13 discrete points (-6 to 6).

- The ANN has q neurons in the hidden layer.
- Each fuzzy inference rule is represented in the ANN by;
- I/P Vector: $a_1,a_2,a_3,a_4,a_5,a_6,a_7;b_1,b_2,b_3,b_4,b_5,b_6,b_7$
- O/P Vector: $C_{-6},C_{-5},C_{-4},C_{-3},C_{-2}, C_{-1}, C_0, C_1, C_2, C_3, C_4, C_5, C_6,$
- For the rule:

IF e is AZ and Δe is NS THEN V is PS

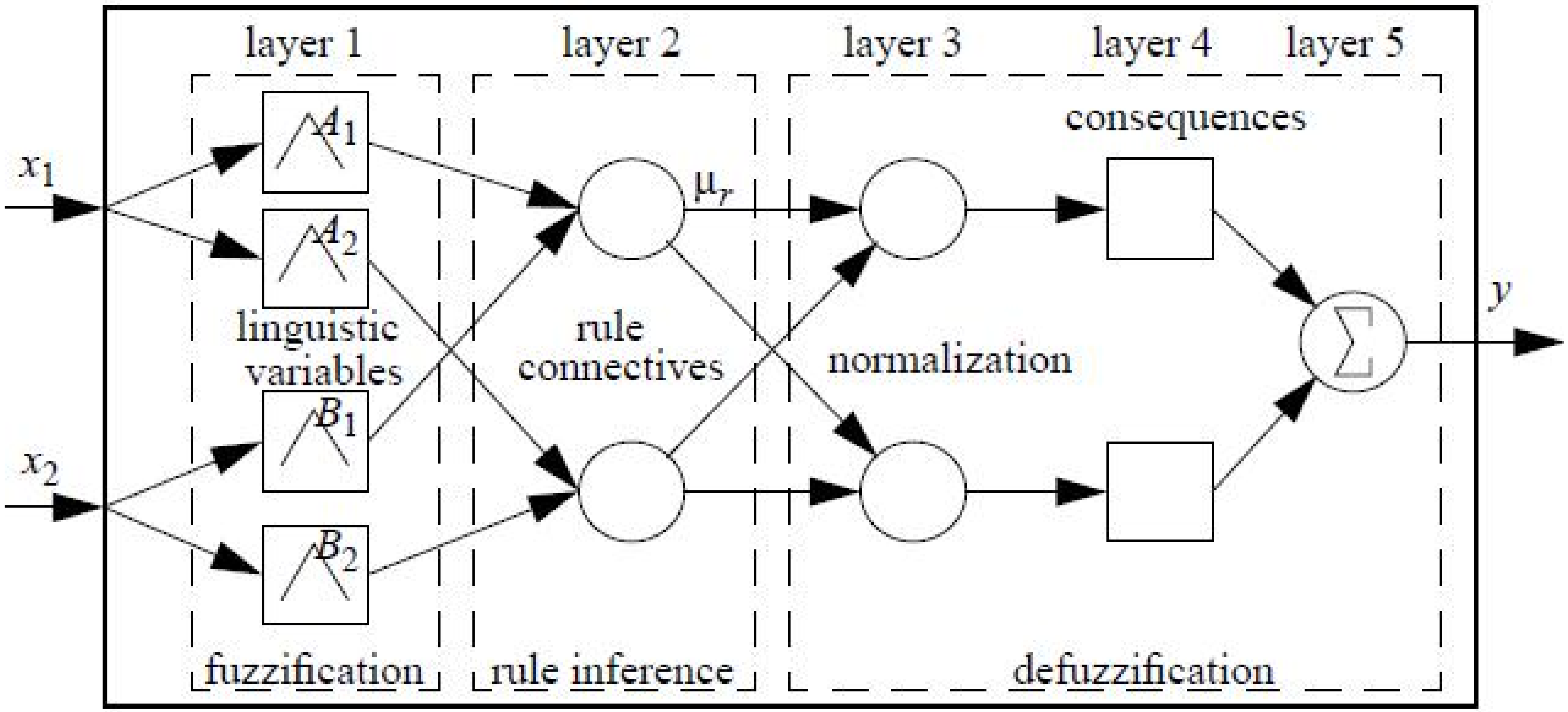
- I/P Vector: 0,0,0,1,0,0,0;0,0,1,0,0,0,0
- O/P Vector: 0,0,0,0,0,0,0,0.5,1,0.5,0,0,0
- The two vectors represent an input/output pair of training set. The full training set consists of all 49 fuzzy rules.


PS
1
0
DESIRED OUTPUT
-6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6
ON1 ON2 ON3 ON4 ON5 ON6 ON7 ON8 ON9 ON10 ON11 ON12 ON13
OUTPUT LAYER
HN1 HNi HNq
HIDDEN LAYER
INPUTS
NL NM NS AZ PS PM PL ; NL NM NS AZ PS PM PL
A1 A2 A3 A4 A5 A6 A7 B1 B2 B3 B4 B5 B6 B7
< 0 , 0 , 0 , 1 , 0 , 0 , 0 ; 0 , 0 , 1 , 0 , 0 , 0 , 0 >

- The following features (or some of them), distinguish fuzzy NNs from others;
  1. Inputs are fuzzy numbers,
  2. Outputs are fuzzy numbers,
  3. Weights are fuzzy numbers,
  4. Weighted inputs of each neuron are not aggregated by summation but by other aggregation functions.
- The following are basic features of the fuzzy NNs:
  1. All real numbers that characterize a classical ANN become fuzzy numbers.
  2. The i/ps and weights of each neuron are fuzzy numbers, and the sum must be calculated by fuzzy arithmetic.
  3. Error function employed in the BP learning algorithm is calculated using fuzzy arithmetic.
  4. The stopping criterion for fuzzy NNs must be properly fuzzified.
  5. We need to fuzzify the BP learning algorithm.

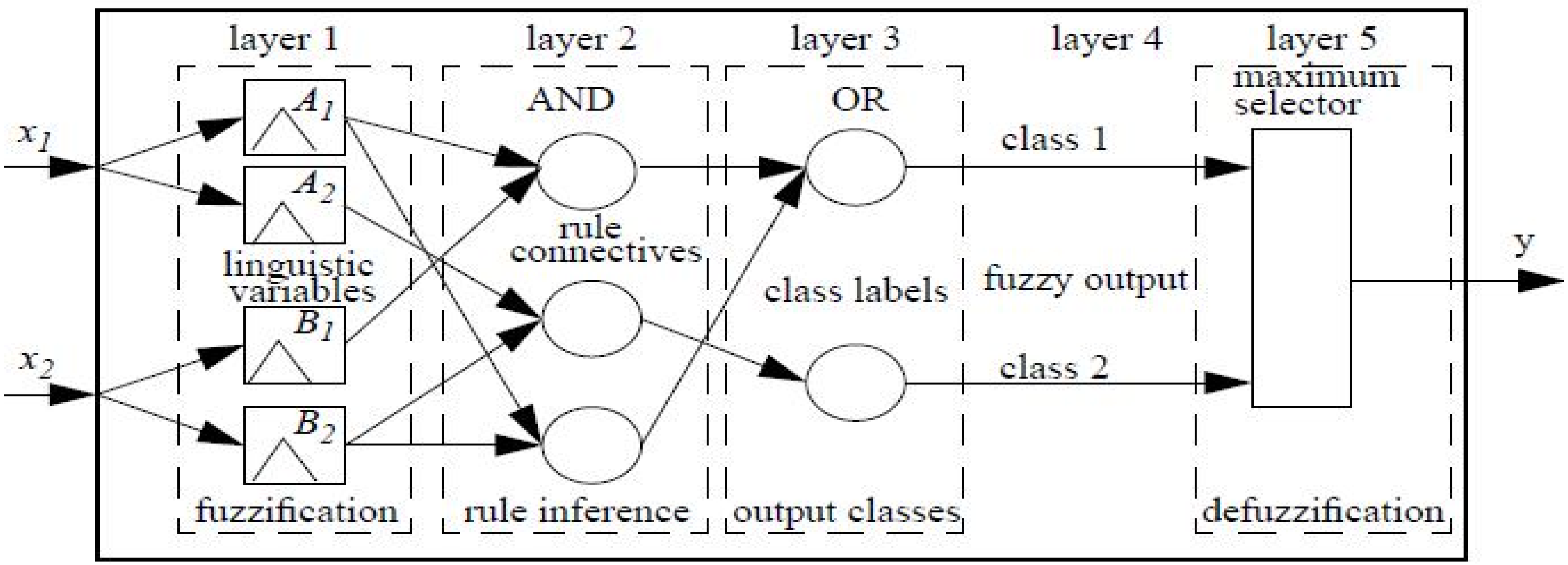
Network Architecture:

- The fuzzy logic inference system can be implemented as a five-layer NN.
- Consider a system which has 2 inputs X1 and X2 and only one output y.
  Rule 1: **IF x1 is A1 AND x2 is B1, THEN y is z1,**
  Rule 2: **IF x1 is A2 AND x2 is B2, THEN y is z2.**



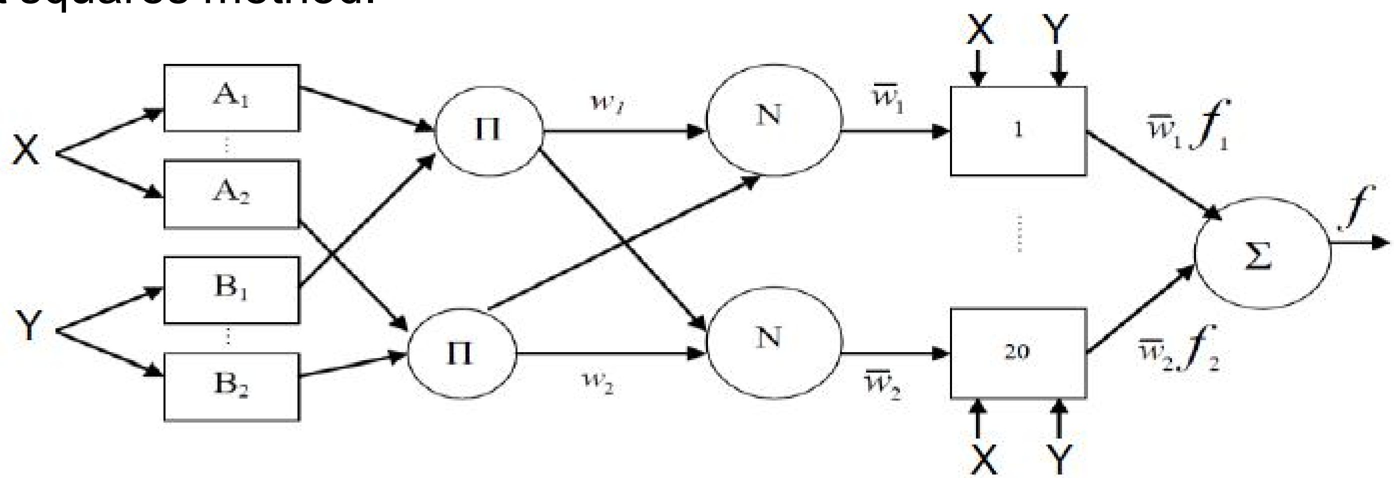
Neuro-fuzzy classifier architecture:

- The architecture of the neuro-fuzzy classifier is slightly different from the architecture used in function approximator;
  Layer 1&2: have the identical function with the approximator.
  Layer 3. Combination of firing strengths
  Layer 4. Fuzzy outputs
  Layer 5. Defuzzification Two first layers
- Consider a neuro-fuzzy system using following fuzzy rules,
  Rule 1: **IF x1 is A1 AND x2 is B1, THEN CLASS is 1,**
  Rule 2: **IF x2 is A2 AND x2 is B2, THEN CLASS is 2,**
  Rule 3: **IF x1 is A1 AND x2 is B2, THEN CLASS is 1.**



Adaptive Neuro Fuzzy Inference System (ANFIS):

- ANFIS implements a first order Sugeno-style fuzzy system.
- It is a method for tuning an existing rule with a learning algorithm based on a collection of training data.
- ANFIS constructs a fuzzy inference system whose membership function parameters are tuned using either a back propagation algorithm alone, or in combination with least squares method.



- In this NN, the neuro fuzzy controller has 2 inputs (X & Y) and it has an o/p (Y). For each input 20 membership functions and also 20 rules in the rules base is considered, such as;

If X is A1 and Y is B1 then f1=p1 X+q1 Y+r1.

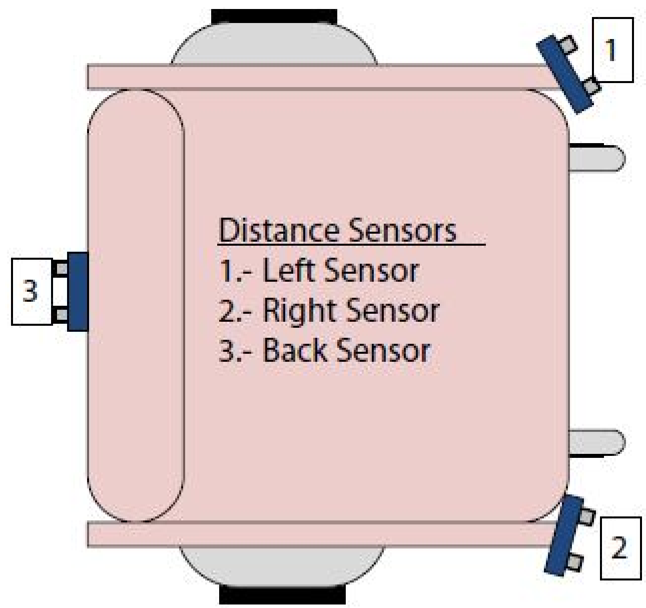
If X is A2 and Y is B2 then f2=p2 X+q2 Y+r2.

- **Layer 1:** consists of membership functions.
- **Layer 2:** implemented the fuzzy AND operator.
- **Layer 3:** acts to scale the firing strengths.
- **Layer 4:** the output of this layer is comprised of a linear combination of the inputs multiplied by the normalized firing strength.
- **Layer 5:** is simple summation of the outputs of layer 4.
- The adjustment of modifiable parameters is a two step process;
  1. Information is propagated forward in the network until layer 4, where the parameters are identified by a least squares estimator.
  2. Then, the parameters in layer 2 are modified using gradient descent.
- The indirect method for adjusting the tunable parameters in the ANFIS network is used with the training data.
- The optimal fuzzy system uses one rule for one i/o pair, thus if number of i/o pairs is large, various clustering can be used to group the i/o pairs so that a group can be represented by one rule.
- **Clustering** means partitioning of a collection of data into disjoint subset or clusters, with the data in a cluster having some properties them distinguish from the data in other clusters.

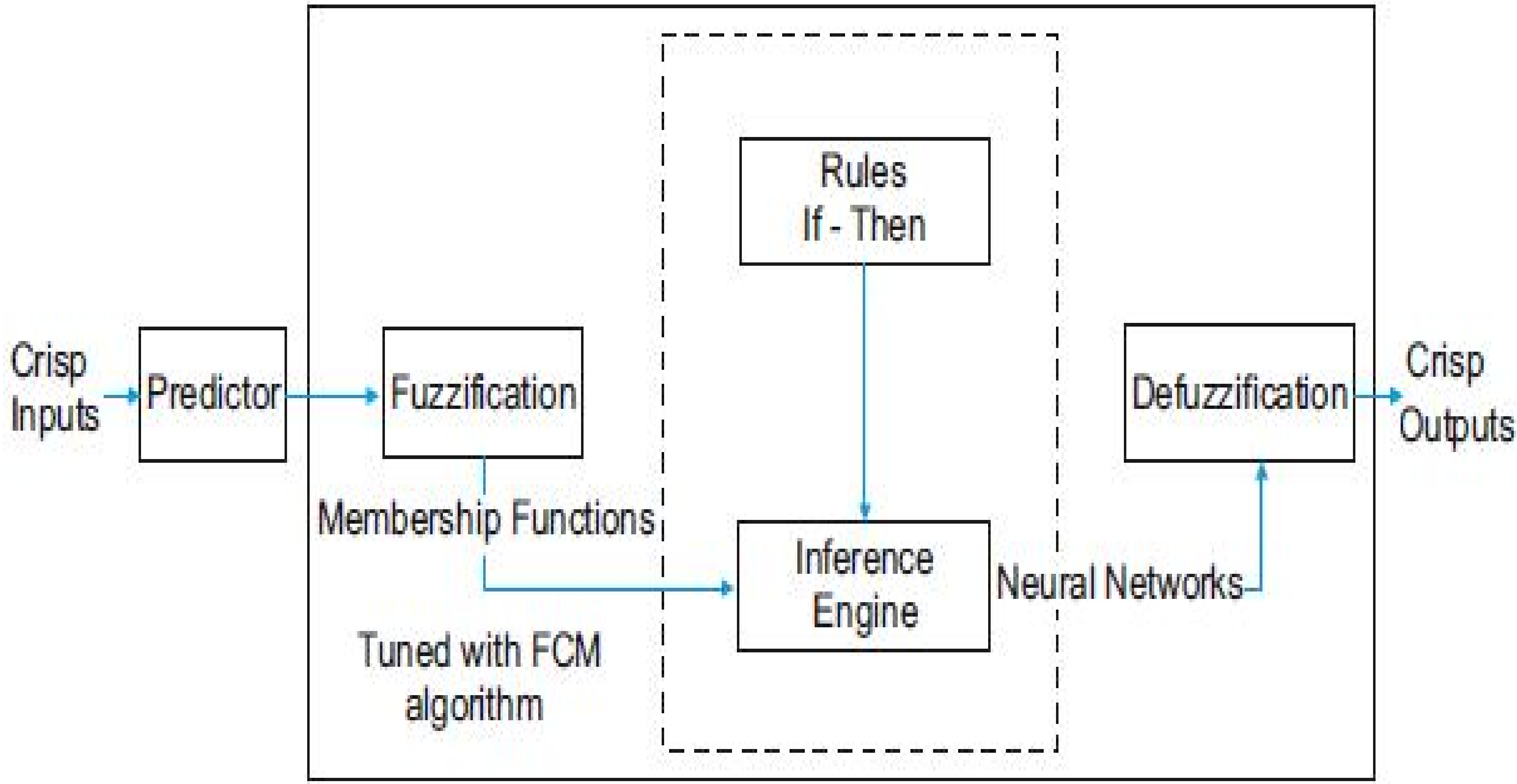
Applications: Wheelchair Neuro-fuzzy Controller:

Ref: P. Ponce-Cruz and F. D. Ramírez-Figueroa, "Intelligent Control Systems with LabVIEW", Springer, 2010.

- The position of the chair is manipulated so that it will avoid static and dynamic obstacles. The controller takes information from three ultrasonic sensors, and then decides the best direction that the wheelchair must follow.
- The outputs of the neuro-fuzzy controller are the voltages sent to a system that generates a PWM to move the motors and the directions in which the wheel will turn.
- The controller is based on trigonometric NNs and fuzzy cluster means. It follows a Takagi–Sugeno inference method, but instead of using polynomials on the defuzzification process it also uses trigonometric neural networks (T-ANNs).



Wheelchair Neuro-fuzzy Controller:



The *IF–THEN* rules for the direction controller

1. *IF* Degree is Small & Direction is Left *THEN* PWM_R IS Very Few, PWM_L IS Very Few, DIR_R is CCW, DIR_L is CW.
2. *IF* Degree is Small & Direction is Center *THEN* PWM_R IS Very Few, PWM_L IS Very Few, DIR_R is NC, DIR_L is NC.
3. *IF* Degree is Small & Direction is Right *THEN* PWM_R IS Very Few, PWM_L IS Very Few, DIR_R is CW, DIR_L is CCW.
4. *IF* Degree is Medium & Direction is Left *THEN* PWM_R IS Some, PWM_L IS Some, DIR_R is CCW, DIR_L is CW.
5. *IF* Degree is Medium & Direction is Center *THEN* PWM_R IS Some, PWM_L IS Some, DIR_R is NC, DIR_L is NC.
6. *IF* Degree is Medium & Direction is Right *THEN* PWM_R IS Some, PWM_L IS Some, DIR_R is CW, DIR_L is CCW.
7. *IF* Degree is Large & Direction is Left *THEN* PWM_R IS Very Much, PWM_L IS Very Much, DIR_R is CCW, DIR_L is CW.
8. *IF* Degree is Large & Direction is Center *THEN* PWM_R IS Very Much, PWM_L IS Very Much, DIR_R is NC, DIR_L is NC.
9. *IF* Degree is Large & Direction is Right *THEN* PWM_R IS Very Much, PWM_L IS Very Much, DIR_R is CW, DIR_L is CCW.

Trigonometric Artificial Neural Networks:

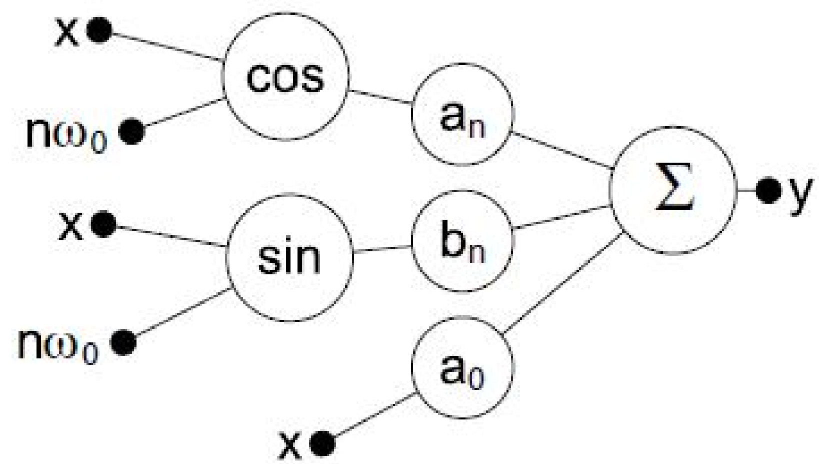
- The trigonometric Fourier series consists of the sum of functions multiplied by a coefficient plus a constant; a neural network can thus be built based on the following equations.
- The topology of this network is composed of two layers. On the first layer the activation function of the neurons are trigonometric functions. On the second layer the results of the activation functions multiplied by its weights plus a constant are summed. This constant is the mean value of the function; the weights are the coefficients of the Fourier trigonometric series

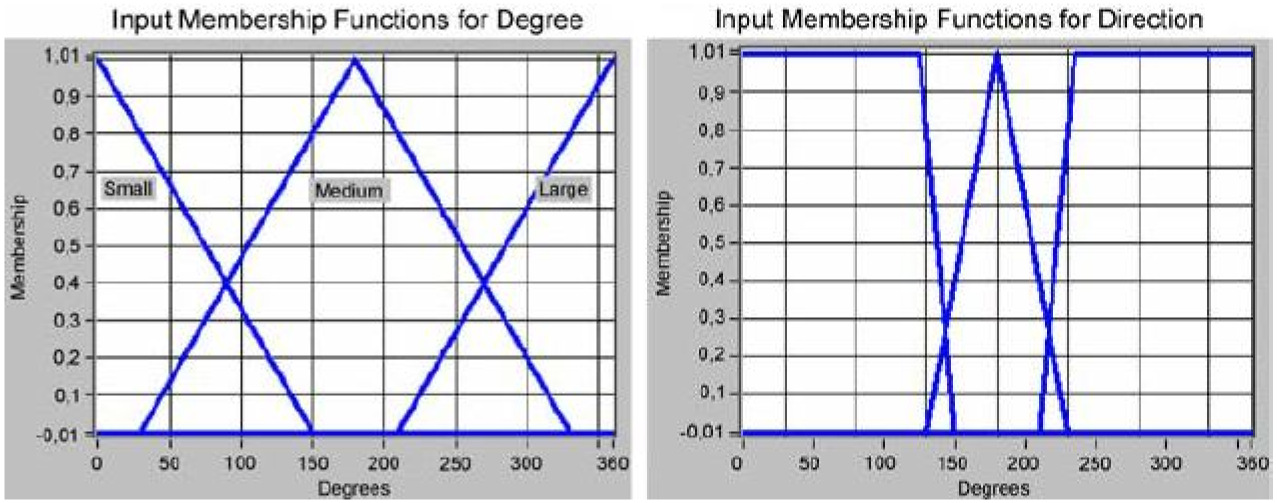
$$f(x) \sim \frac{a_0}{2} + \sum_{n=1}^{\infty} \left(a_n \cos(nx) + b_n \sin(nx)\right)$$

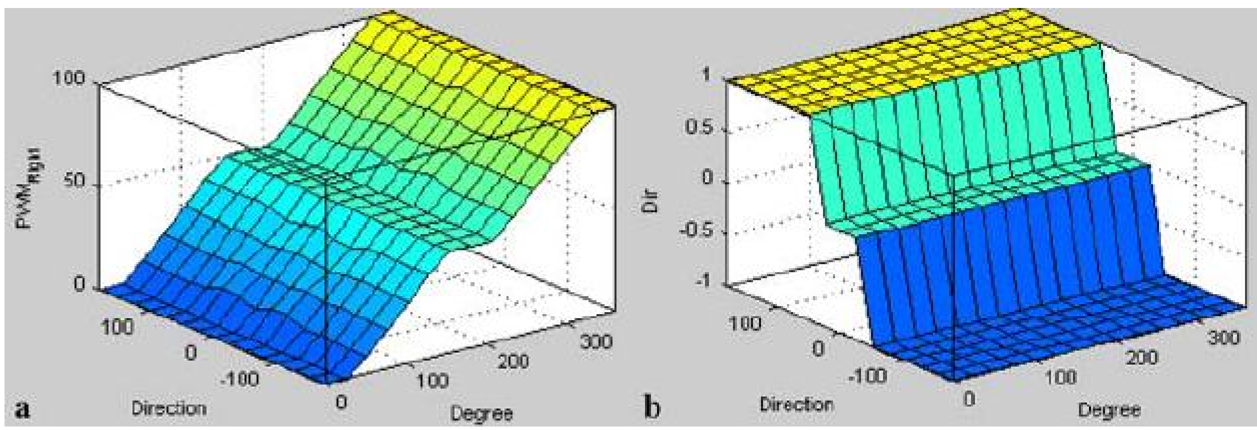
$$a_0 = \frac{1}{T} \int_0^T f(x)\,\mathrm{d}x$$

$$a_n = \frac{1}{T} \int_0^T f(x) \cos(n\omega x)\,\mathrm{d}x$$

$$b_n = \frac{1}{T} \int_0^T f(x) \sin(n\omega x)\,\mathrm{d}x\,.$$

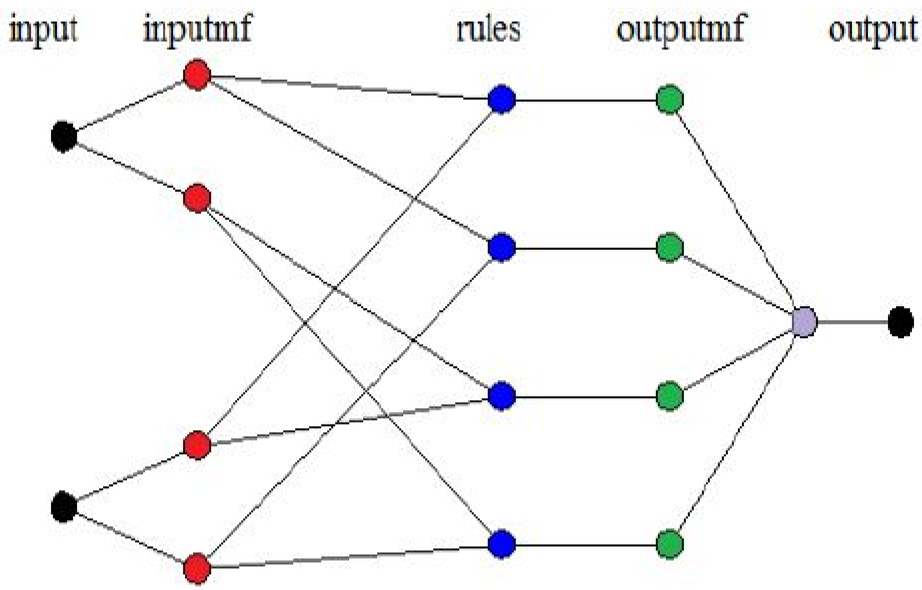



Input Membership Functions for Degree
Membership
1,01
0,9
0,8
0,7
0,6
0,5
0,4
0,3
0,2
0,1
-0,01
Small
Medium
Large
0
50
100
150
200
250
300
360
Degrees
Input Membership Functions for Direction
Membership
1,01
0,9
0,8
0,7
0,6
0,5
0,4
0,3
0,2
0,1
-0,01
0
50
100
150
200
250
300
360
Degrees


100
50
0
PWM_Right
100
0
-100
Direction
0
100
200
300
Degree
a
1
0.5
0
-0.5
-1
Dir
100
0
-100
Direction
0
100
200
300
Degree
b

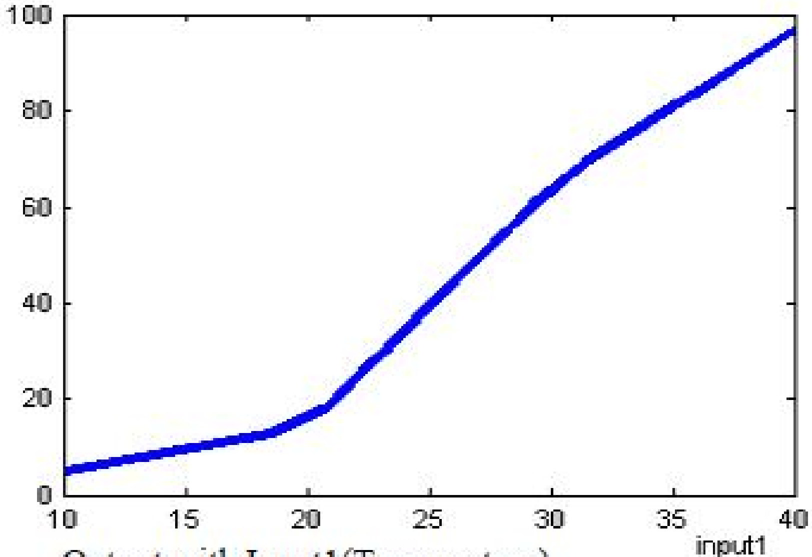
DEVELOPMENT OF NEURO FUZZY CONTROLLER ALGORITHM FOR AIR CONDITIONING SYSTEM, *By: A. KAUR and A.KAUR, International Journal of Engineering Science and Technology (IJEST), Vol. 4 No.04 April 2012, pp.1666-1671.*

The neurofuzzy controller for air conditioning system takes two inputs (each has 4 fuzzy sets) from temperature and humidity sensors and controls the compressor speed.

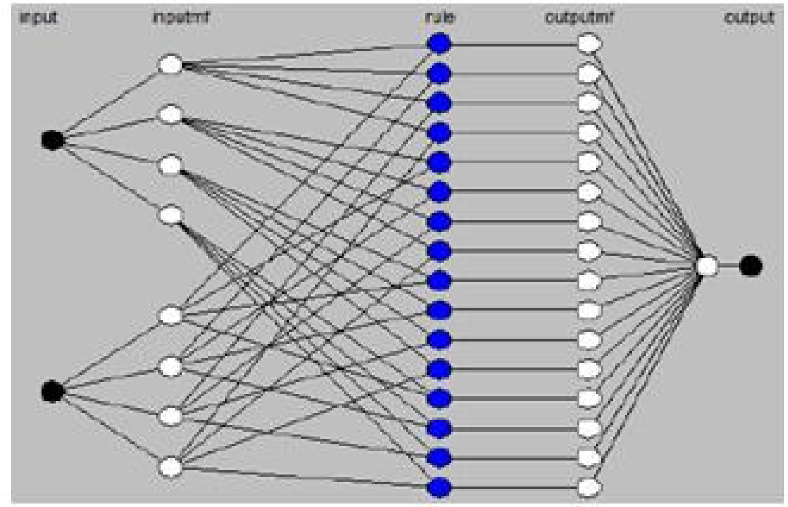


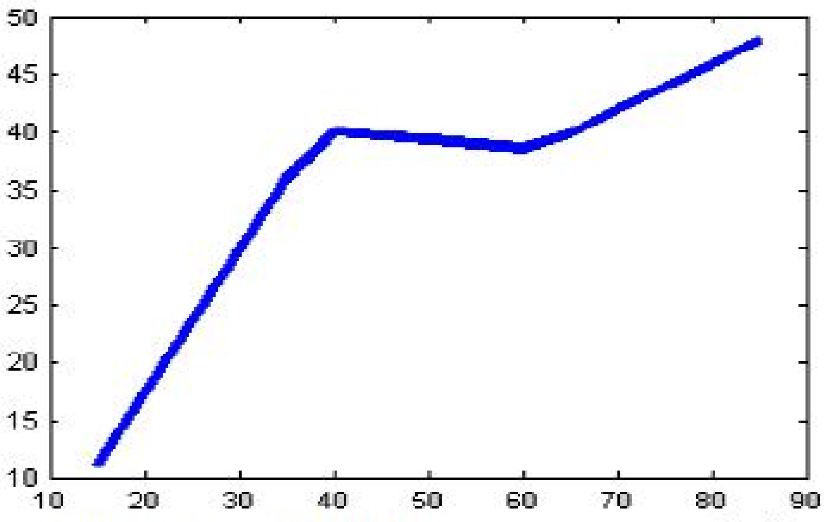
Rule base of neuro-fuzzy controller

Rules	Temperature	Humidity	Compressor speed
1.	Very Low	Dry	Off
2.	Very Low	Comfortable	Off
3.	Very Low	Humid	Off
4.	Very Low	Sticky	Low
5.	Low	Dry	Off
6.	Low	Comfortable	Off
7.	Low	Humid	Low
8.	Low	Sticky	Medium
9.	High	Dry	Low
10.	High	Comfortable	Medium
11.	High	Humid	Fast
12.	High	Sticky	Fast
13.	Very High	Dry	Medium
14.	Very High	Comfortable	Fast
15.	Very High	Humid	Fast
16.	Very High	Sticky	Fast

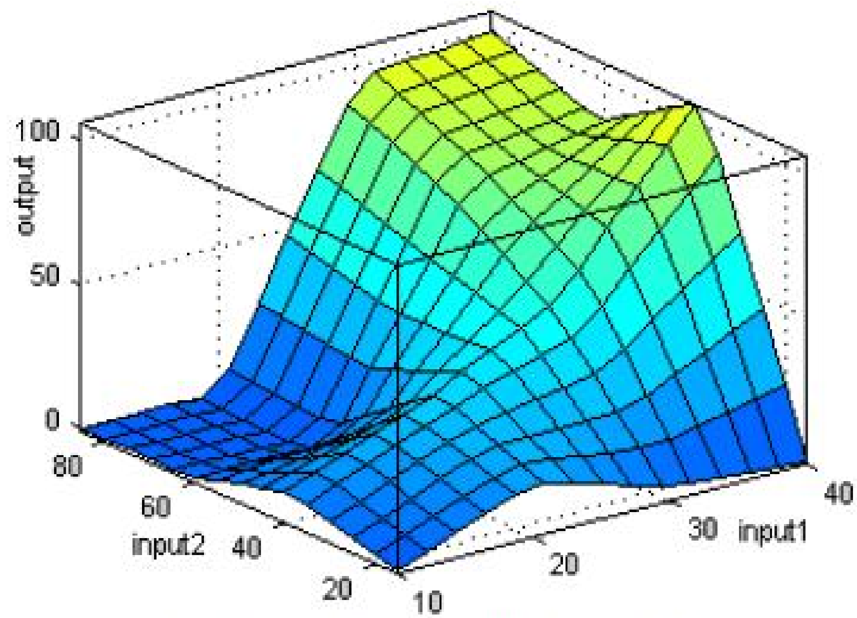


Output with Input1(Temperature)





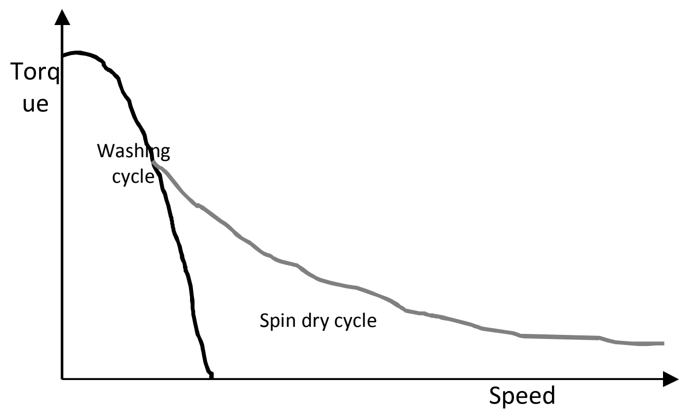
Output with Input2(Humidity)

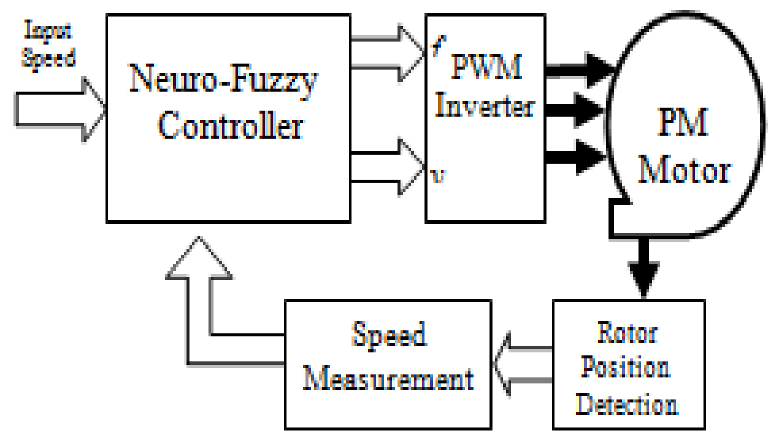


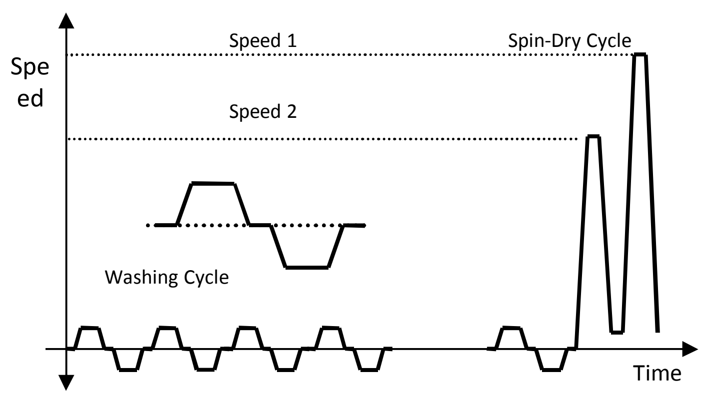
Surface view of neuro-fuzzy controller

Neuro-Fuzzy Controller of a Sensorless PM Motor Drive for Washing Machines, *By: Kasim M. Al-Aubidy and Mohummad M. Ali, IEEE-SSD7 Multiconference, Tunisia, 2007.*

A simple neuro-fuzzy controller has been used to control speed and position of a sensorless PM motor. The results demonstrate the capability of such a drive system in washing machine applications where simplicity, reliability and stability are more important issues.

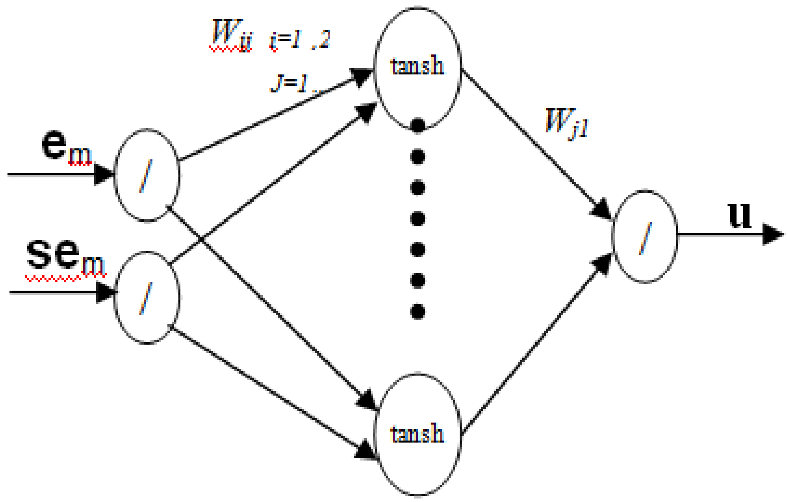


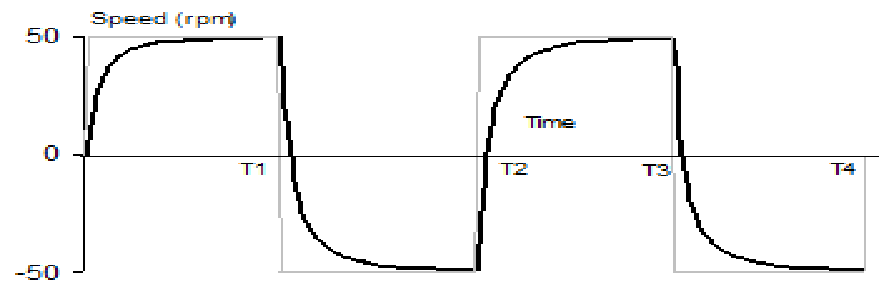




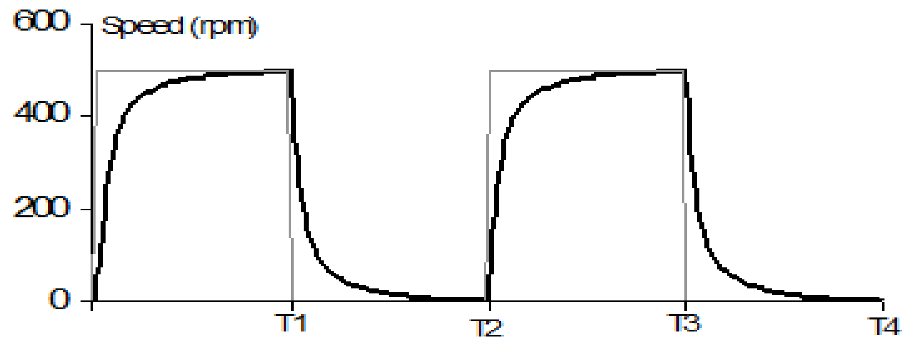
The neuro-fuzzy controller will be formed by training a back propagation NN on the bases of fuzzy number rules described by their central values which are extracted using clustering algorithm from the available i/o collected from other control strategies.

- 2-node linear input layer receiving error and sum of error.
- 10-node nonlinear hidden layer with tansh activation function.
- One node linear output layer that generates the actuating signal.
- It needs 5620 iterations to learn the given i/p pattern, with 0.1 learning rate.





a). during washing cycle.



b). during drying cycle.

References:

1. T. L. Huntsberger and P. Ajjimarangsee, “Parallel self-organizing feature maps for unsupervised pattern recognition,” Int. J. General Systems, vol. 16, no. 4, pp. 357–372, 1990.
2. P. Ponce-Cruz and F. D. Ramírez-Figueroa, “Intelligent Control Systems with LabVIEW”, Springer, 2010, ISBN 978-1-84882-683-0. Available online: http://files.instrument.com.cn/bbs/upfile/files/20111226/20111226101936.pdf
3. Kasim M. Al-Aubidy and Mohummad M. Ali, “Neuro-Fuzzy Controller of a Sensorless PM Motor Drive for Washing Machines”, IEEE-SSD7 Multiconference, Tunisia, 2007.