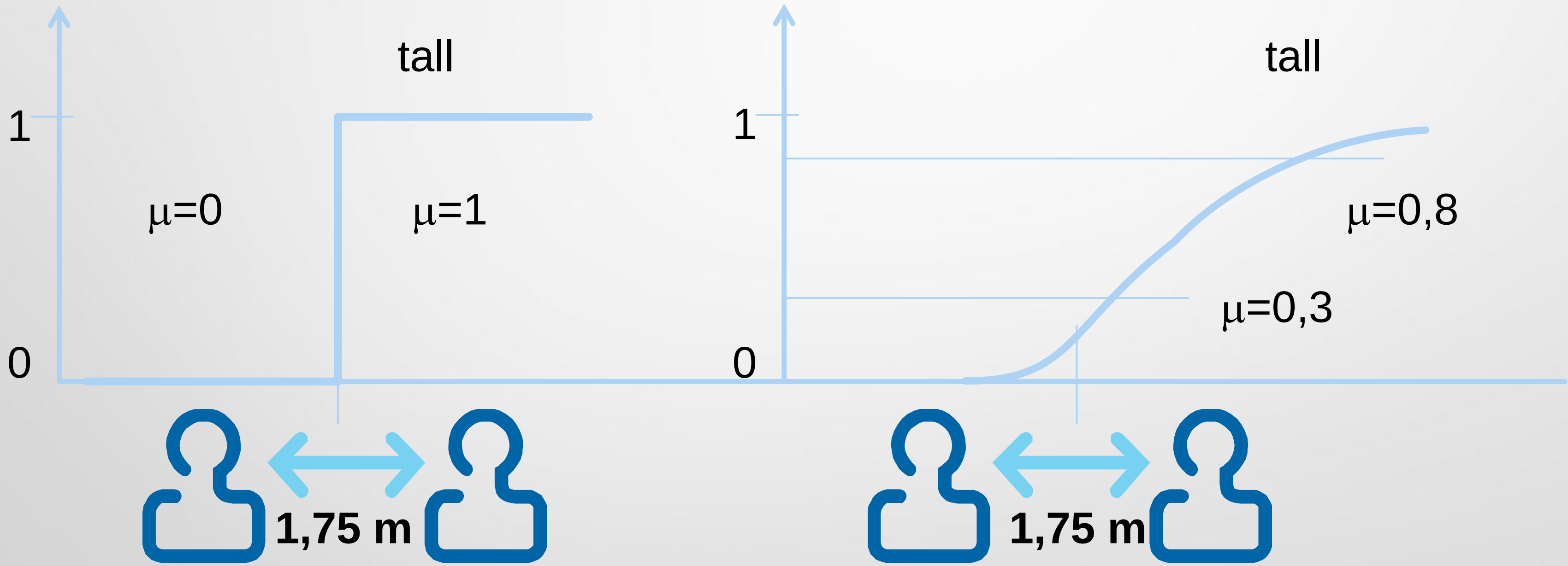
Neo-fuzzy neural networks for modeling of complex systems

Yancho Todorov, Ph.D.

yancho.todorov@ieee.org

What fuzziness is?

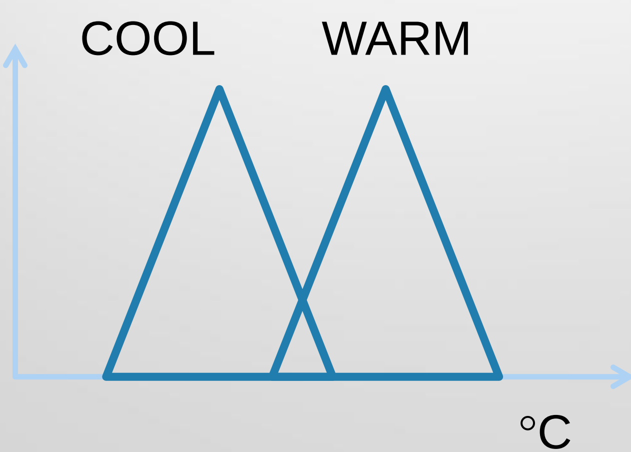
If we separate a group of people assuming that every person of height above 1.75 m is TALL :

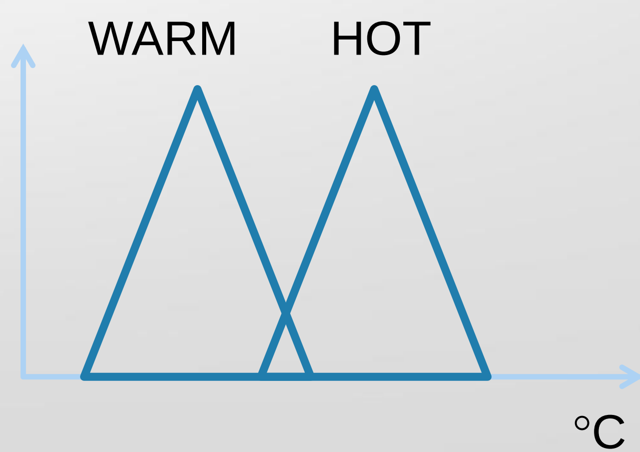


What fuzziness is?

COLD	COOL	WARM	HOT
0 – 15 °C	15-25 °C	25 – 38 °C	40-100 °C

COLD	COOL	WARM	HOT
0 – 20 °C	20-27 °C	27 – 45 °C	45-100 °C





Fuzzy Set

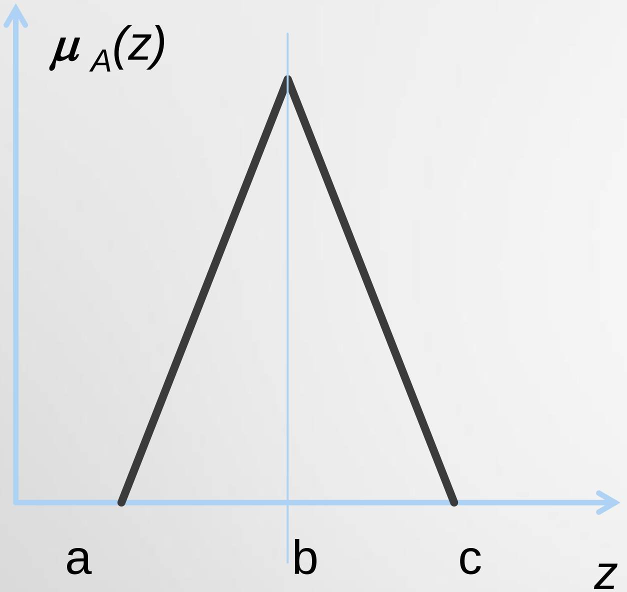
If **Z** is a space of elements, with main component **Z** described by ***z***, such that:

Z = {***z***}

Thus a fuzzy set **A** in **Z** is characterized by a **membership function $\mu_A(z)$**, which associate each point in **z** with real number in **[0,1]**, along with the elements $\mu_A(z)$ for ***z***, which are called **degree of membership** of ***z*** in **A**.

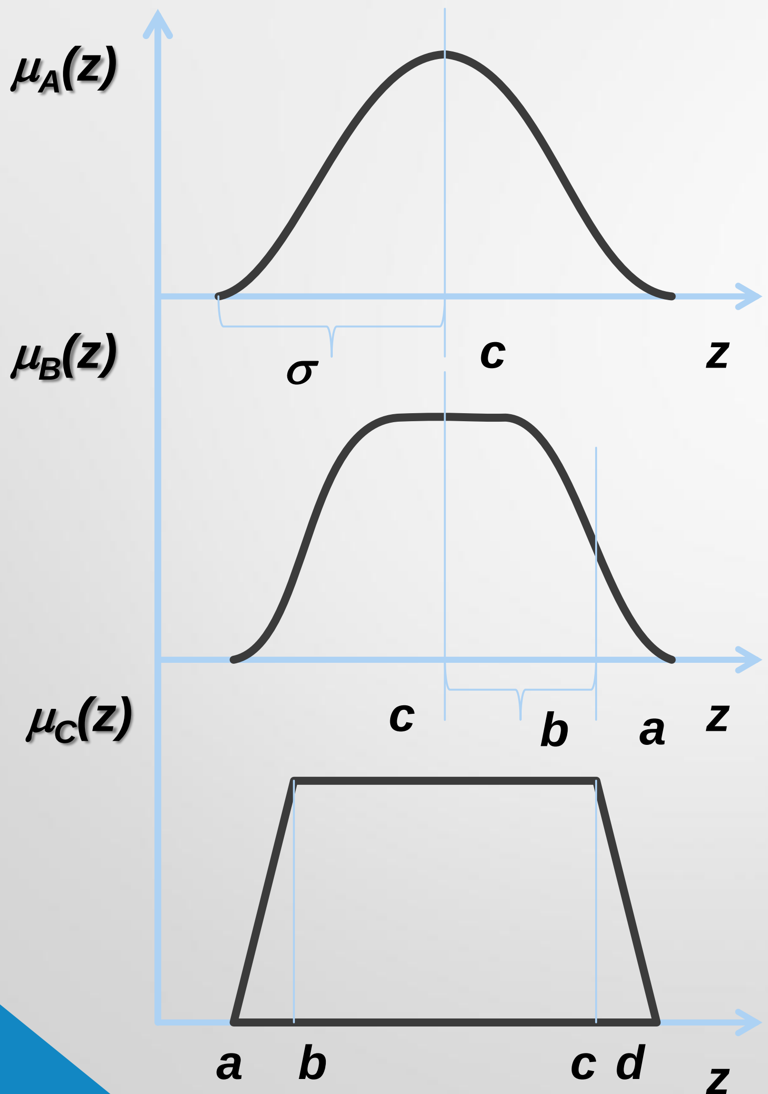
If the value of $\mu_A(z)$ is closer to **one**, then a greater value of the degree of membership of ***z*** to **A** **is assigned**.

A={z, μ(z)}, z ∈ Z, μ_A(z)∈ [0,1]



$$\mu_A(z) = \begin{cases} \frac{z-a}{b-a} & b > z > a \\ 1 & z = b \\ \frac{c-z}{c-b} & c > z > b \end{cases}$$

Fuzzy Sets- type 1



$$\mu_A(z) = \exp\left(-\frac{z-c}{2\sigma^2}\right)$$

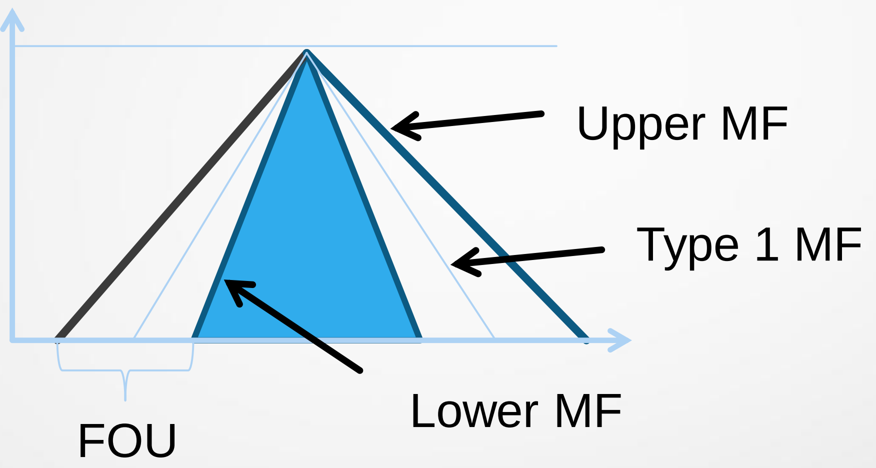
$$\mu_B(z) = \frac{1}{1+\left|\frac{z-c}{a}\right|^{2b}}$$

$$\mu_C(z) = \begin{cases} \frac{z-a}{b-a} & b > z > a \\ 1 & c > z > b \\ \frac{d-z}{d-c} & c > x > b \end{cases}$$

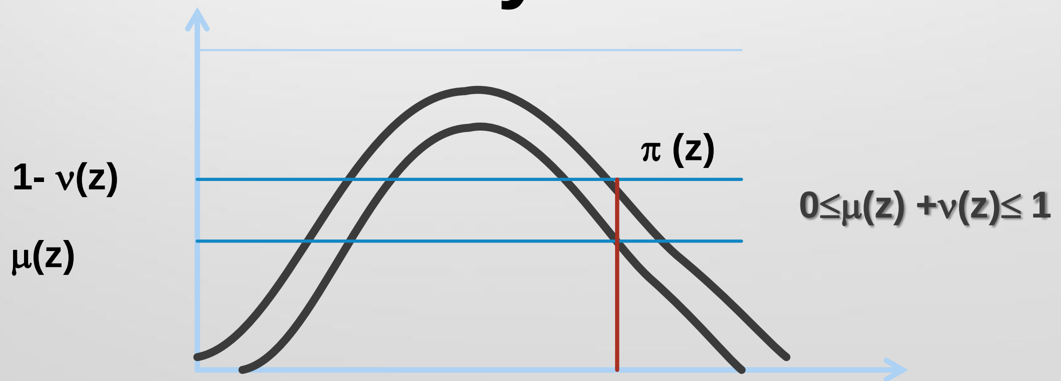
Other types of Fuzzy Sets

Type 2 Interval Fuzzy Set

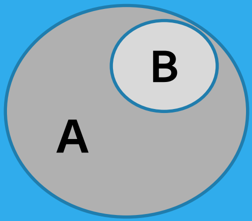
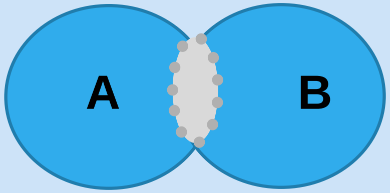
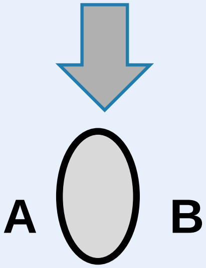
Footprint **O**f
Uncertainty



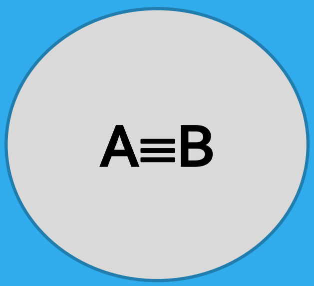
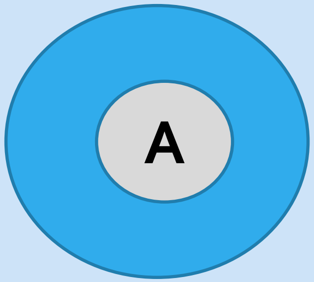
Intuitionistic Fuzzy Set



Fuzzy sets operations

SUBSET 	$A \in B \rightarrow \forall\, z \in Z : \mu_A(z) < \mu_B(z)$
UNION 	$A \cup B\ \forall\, z \in Z : \max\{\mu_A(z), \mu_B(z)\}$
INTERSECTION 	$A \cap B\ \forall\, z \in Z : \min\{\mu_A(z), \mu_B(z)\}$

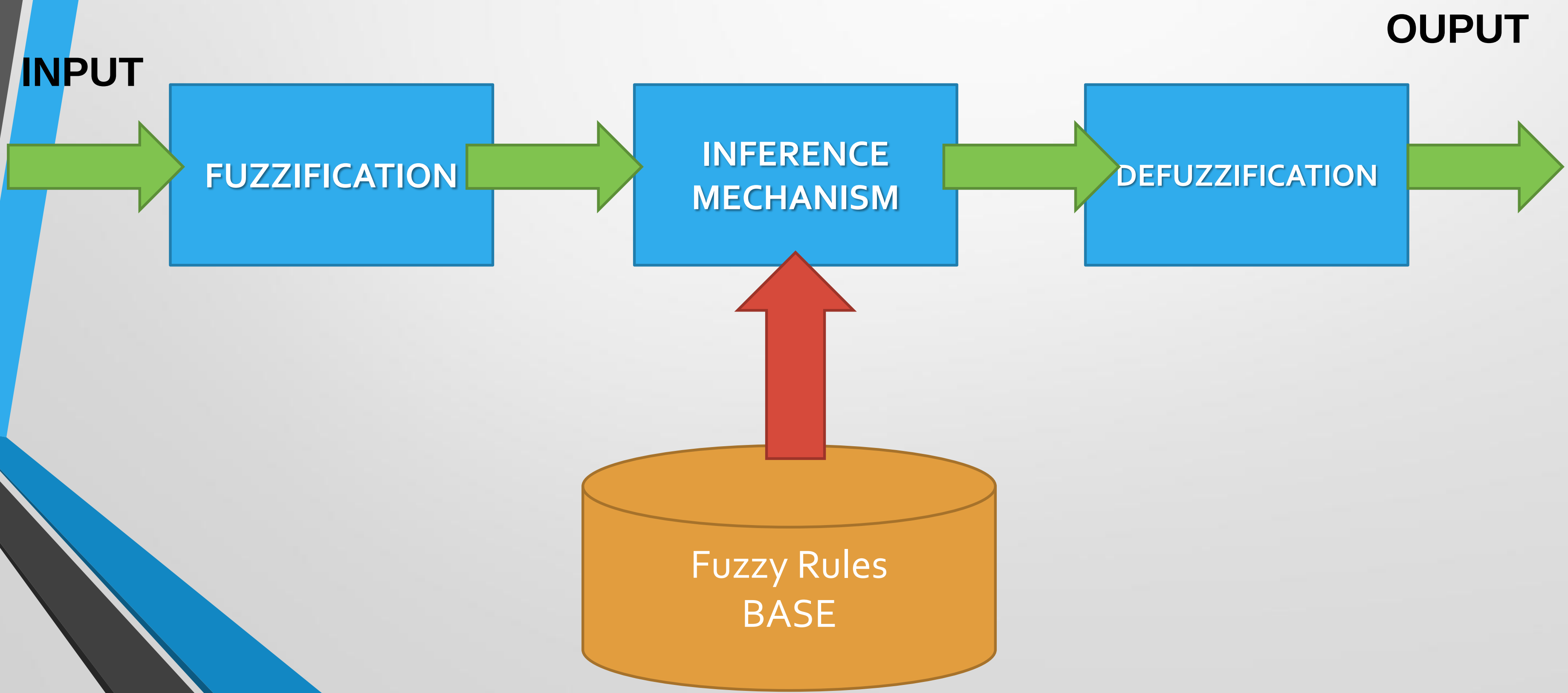
Fuzzy sets operations

EQUITY 	$A \equiv B \rightarrow \forall z \in Z : \mu_A(z)=\mu_B(z)$
COMPLEMENT 	$\tilde{A} \rightarrow \forall z \in Z : \mu_{\tilde{A}}(z)=1-\mu_A(z)$

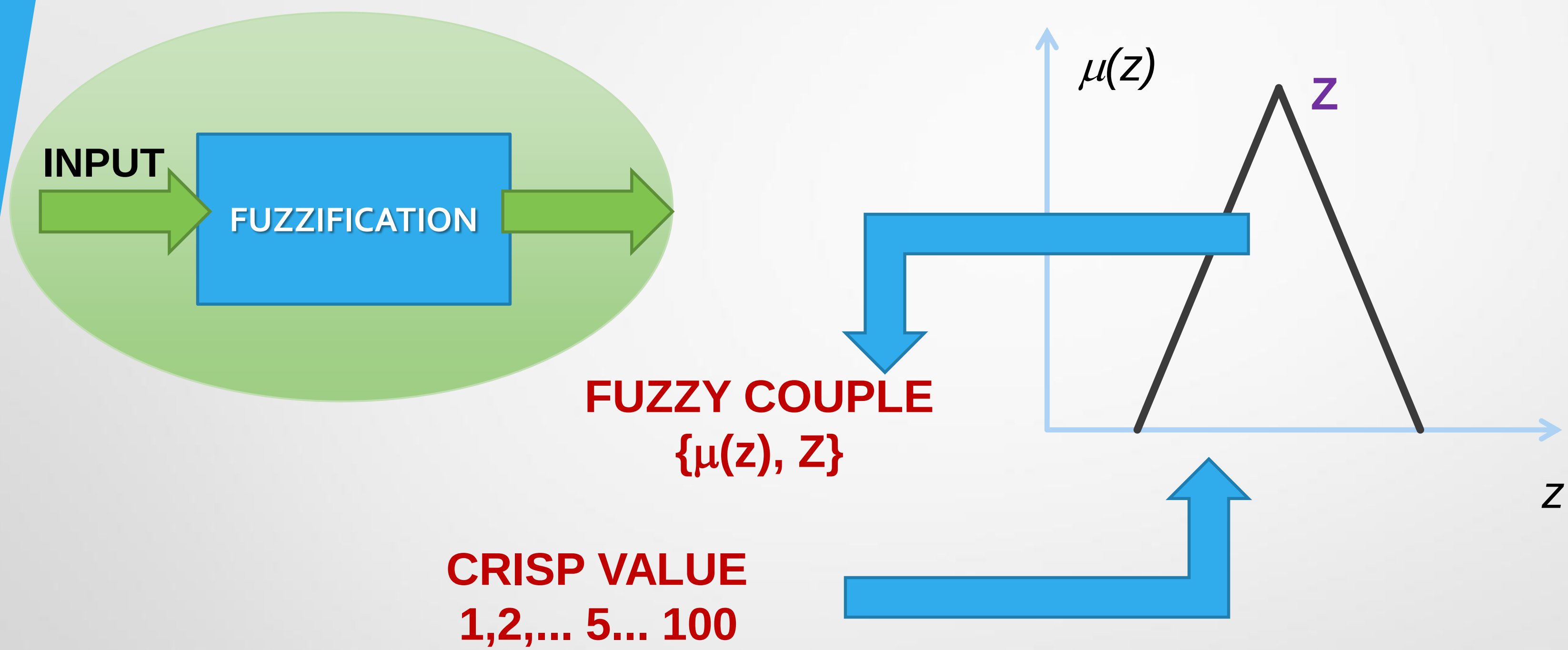
Types of Fuzzy inferences

- Mamdani Fuzzy Inference
- Sugeno Fuzzy Inference
  - ANFIS (Adaptive Neuro-Fuzzy Inference System)
  - Takagi-Sugeno Neuro-Fuzzy model
- Tsukamoto Fuzzy Inference

Generalized structure of a Fuzzy Inference



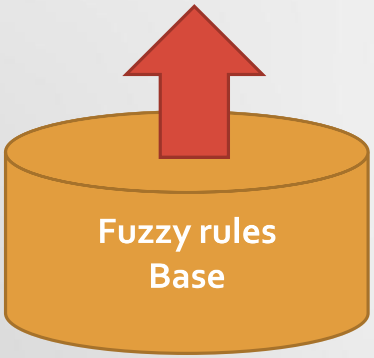
Sugeno Fuzzy Inference: Fuzzification



The main purpose of the "fuzzification" procedure is to transform the crisp input values into fuzzy couples – linguistic variable of a set and a corresponding membership degree!

Sugeno Fuzzy Inference: Rules base

$$R^{(i)} : if \ z_1 \ is \ \tilde{Z}_1^{(i)} \ and.......z_s \ is \ \tilde{Z}_s^{(i)} \ then \ F_i(z)$$



$$F(z) = a_1 z_1 + a_2 z_2 + ... + a_s z_s$$

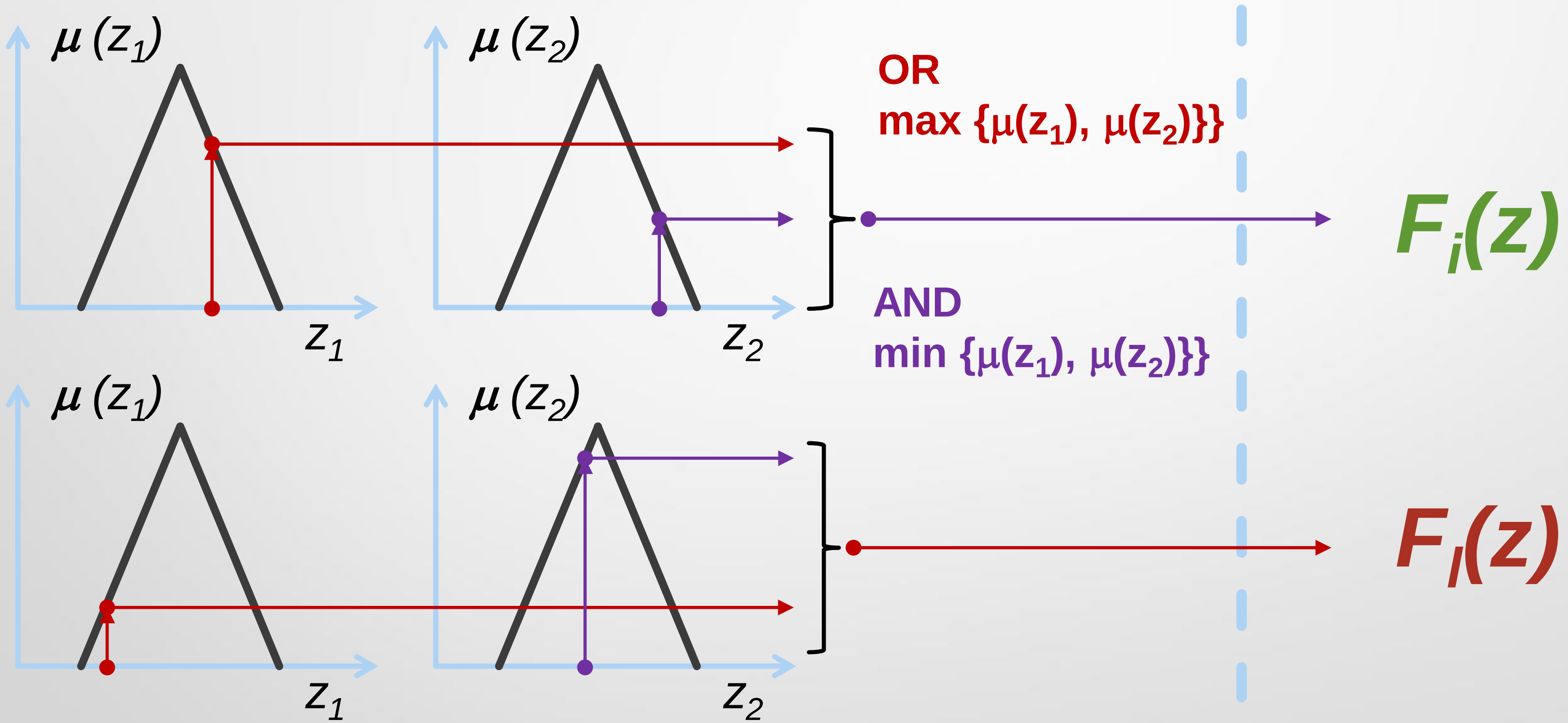
HOW many fuzzy rules are being generated?

FR= **N**^{P}

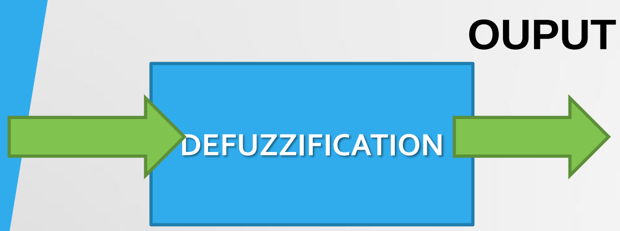
N – number of the membership functions per input

P – number of the input parameters

Sugeno Fuzzy Inference mechanism



Sugeno Fuzzy Inference: Defuzzification



$$\mu_y(u) = \mu_{1j}(u) * \mu_{2j}(u) * \ldots\ldots * \mu_{pj}(u)$$

$$u = \frac{\sum_1^{FR} F_i(z)\mu^{(i)}{}_y(u)}{\sum_1^{FR} \mu^{(i)}{}_y(u)}$$

Neo-fuzzy neuron

- **The NEO-Fuzzy Neuron concept enables the possibility to model complex dynamics with less computational effort, compared to classical Fuzzy-Neural Networks.**
- **Unfortunately, its application in purpose to process modeling and control under uncertainties/ data variations, have not been studied yet.**
- **In the presented approach, the conventional concept is extended with Type-2 Interval Fuzzy Logic in order to be achieved overall robustness of the proposed model**
- **Thus, introducing Type-2 Fuzzy Logic in purpose of handling uncertain variations is beneficial for modeling different plant processes with complex dynamics.**
- **To overcome some deficiencies in the classical gradient learning approach, a simple heuristic approach is introduced.**

Neo-fuzzy neuron

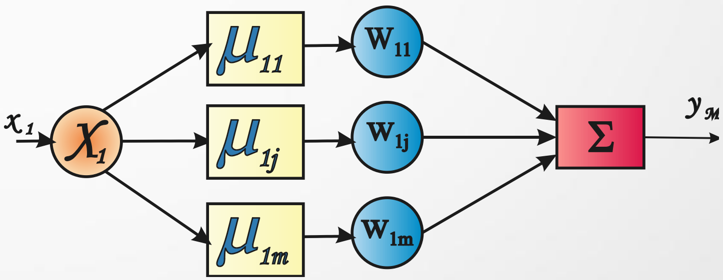
✓The NEO-Fuzzy neuron is similar to a o-th order Sugeno fuzzy system, in which only one input is included in each fuzzy rule, and to a radial basis function network (RBFN) with scalar arguments of basis functions

✓In fact the NFN network is a multi-input single-output system – MISO !

✓The NEO-Fuzzy neuron has a nonlinear synaptic transfer characteristic.

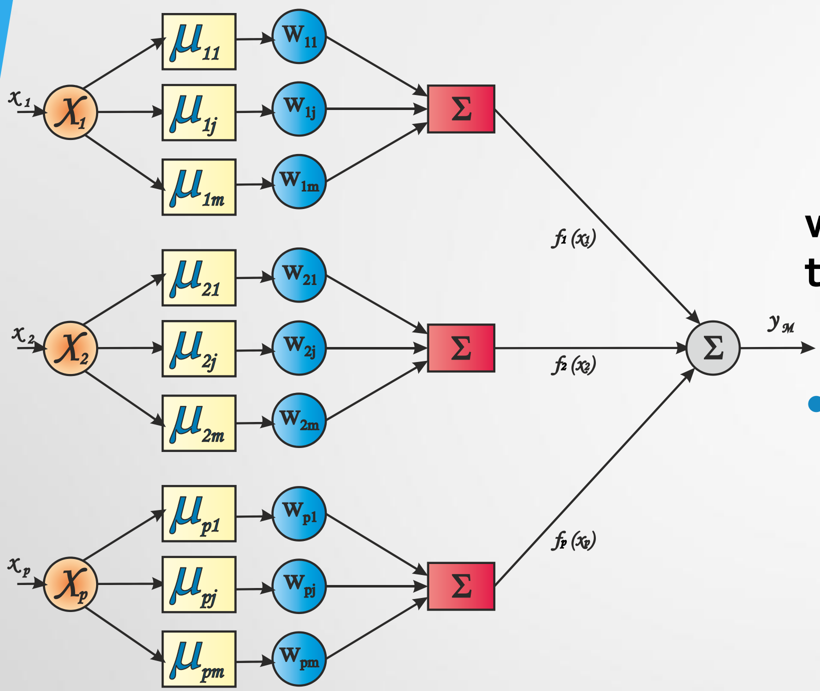
✓The nonlinear synapse is realized by a set of fuzzy implication rules.

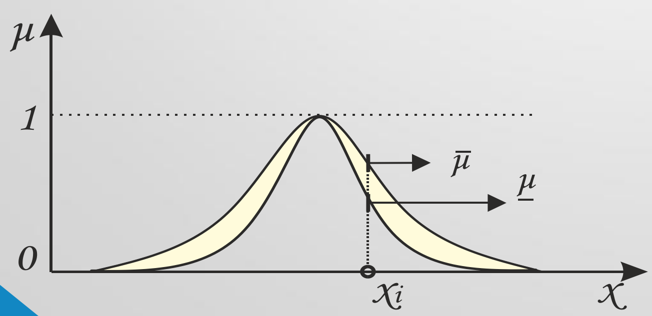
✓The output of the NEO-Fuzzy neuron is obtained by the following equation:



$$f(x) = \sum_{j=1}^{m} \mu_j(x(k))w_j$$

Type-2 Neo-fuzzy network





- The MISO NEO-fuzzy neural network topology can be represented as:

$$\hat{y}(k) = f(x(k))$$

where x(k) is an input vector of the states in terms of different time instants.

- Each Neo-Fuzzy Neuron comprises a simple fuzzy inference which produces reasoning to singleton weighting consequents:

$$R^{(i)} : if \;\; x_i \;\; is \;\; \tilde{A}_i^{(i)} \quad then \quad f_i(x_i)$$

- Each element of the input vector is being fuzzified using Type-2 Interval Fuzzy set:

$$\mu_{ij}(x_i) = -\exp\left(\frac{x_i - c_{ij}}{2\sigma_{ij}}\right)^2 = \begin{cases} \bar{\mu}_{ij} \text{ as } \sigma_{ij} = \bar{\sigma}_{ij} \\ \underline{\mu}_{ij} \text{ as } \sigma_{ij} = \underline{\sigma}_{ij} \end{cases}$$

Type-2 Neo-fuzzy network

- The fuzzy inference should match the output of the fuzzifier with fuzzy logic rules performing fuzzy implication and approximation reasoning in the following way:

$$\mu_{ij}* = \begin{cases} \bar{\mu}_{ij}* = \prod_{i=1}^{n} \bar{\mu}_{ij} \\ \underline{\mu}_{ij}* = \prod_{i=1}^{n} \underline{\mu}_{ij} \end{cases}$$

- The output of the network is produced by implementing consequence matching and linear combination as follows:

$$\hat{y}(k) = \frac{1}{2}\sum_{j=1}^{l}(\bar{\mu}_{ij}* + \underline{\mu}_{ij}*) f_i(x_i) = \frac{1}{2}\sum_{i=1}^{l}(\bar{\mu}_{ij}* + \underline{\mu}_{ij}*) w_{ij}$$

which in fact represents a weighted product composition of the *i-th* input to *j-th* synaptic weight.

Learning Algorithm

- To train the proposed modeling structure an unsupervised learning scheme has been used. Therefore, a defined error cost term is being minimized at each sampling period in order to update the weights:

$$E = \varepsilon^2/2 \text{ and } \varepsilon(k) = y_d(k) - \hat{y}(k)$$

- As learning approach of the proposed modeling structure a simple *heuristic backpropagation* approach , where the scheduled parameters depend on the *signum* of the gradient and defined learning rate, is adopted:

$$w_{ij}(k+1) = w_{ij}(k) + \Delta w_{ij}(k) = w_{ij}(k) + \eta_{ij}(k) sign\left(\frac{\partial E(k)}{\partial w_{ij}(k)}\right)$$

$$\Delta w_{ij}(k) = -\eta_{ij}(k) sign\left(\frac{\partial E(k)}{\partial w_{ij}(k)}\right) = -\eta_{ij}(k) sign\left(\frac{\partial E(k)}{\partial \hat{y}(k)}\frac{\partial \hat{y}(k)}{\partial w_{ij}(k)}\right) = -\eta_{ij}(k) sign\left(\varepsilon(k)\frac{\partial \hat{y}(k)}{\partial w_{ij}(k)}\right)$$

Learning Algorithm

- The learning rate is local to each synaptic weight and it is adjusted by taking into account the extent of the gradient in the current and the past sample period as:

$$\eta_{ij}(k)\begin{cases} \min\left(a\eta_{ij}(k-1),\eta_{\max}\right) & if \quad \Delta E_{ij}(k)\Delta E_{ij}(k\text{ -}1)>0 \\ \max\left(b\eta_{ij}(k-1),\eta_{\min}\right) & if \quad \Delta E_{ij}(k)\Delta E_{ij}(k\text{ -}1)<0 \\ \eta_{ij}(k-1) & if \quad \Delta E_{ij}(k)\Delta E_{ij}(k\text{ -}1)=0 \end{cases}$$

where the constants are: a=1.2, b=0.5 and $\eta_{min}=10^{-3}$, $\eta_{max}=5$.

The main advantage of the proposed approach is that the information about the gradient is neglected, which accelerates significantly the learning process!

Numerical examples

- To test the modeling capabilities of the proposed NEO-fuzzy neural network, a numerical experiments in prediction of two common chaotic time series (Mackey-Glass a and Rossler)are investigated.
- The Rossler chaotic time series are described by three coupled first-order differential equations:

$$\frac{dx}{dt} = -y - z \qquad \frac{dy}{dt} = x + ay \qquad \frac{dz}{dt} = b + z(x - c)$$

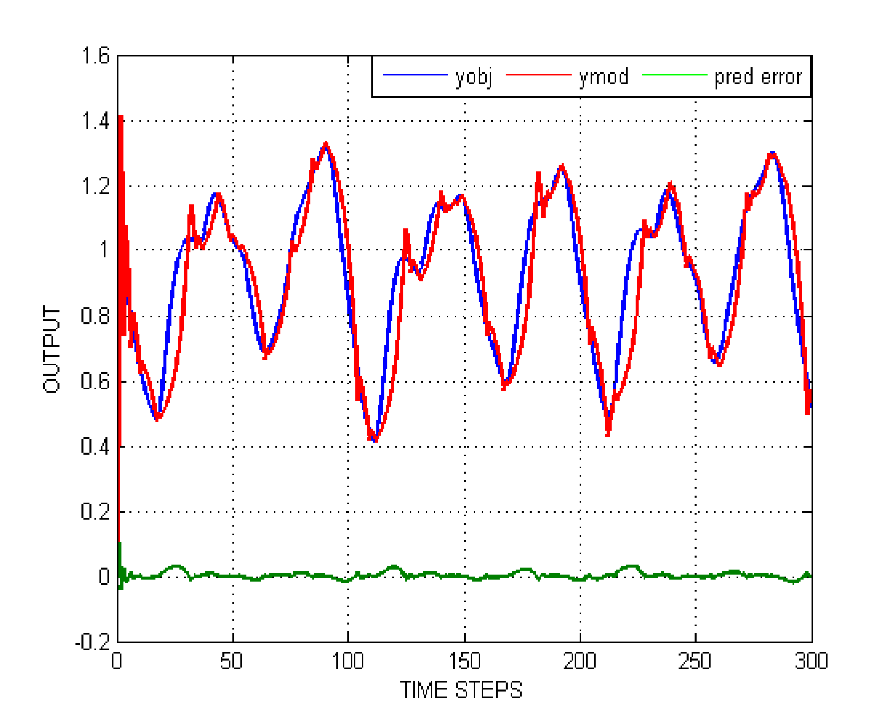
a=0.2; b=0.4; c=5.7 and initial conditions x_0=0.1; y_0=0.1; z_0=0.1

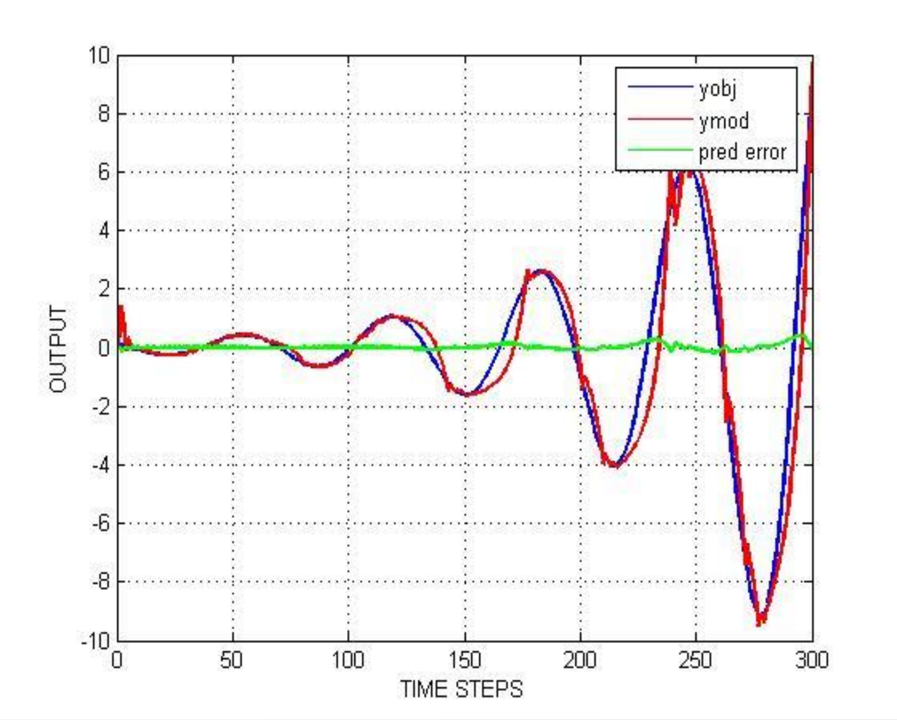
- The Mackey-Glass (MG) chaotic time series is described by the following time-delay differential equation:

$$x(i+1) = \frac{x(i) + ax(i-s)}{(1 + x^c(i-s)) - bx(i)}$$

a=0.2; b=0.1; C=10; initial conditions x_0=0.1 and τ= 17s.

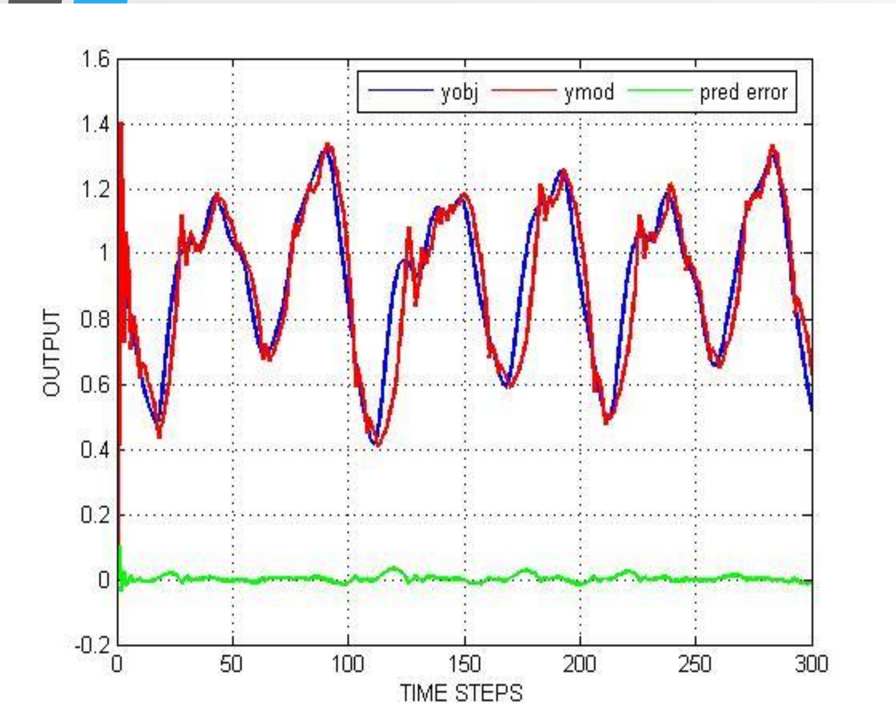
Numerical examples

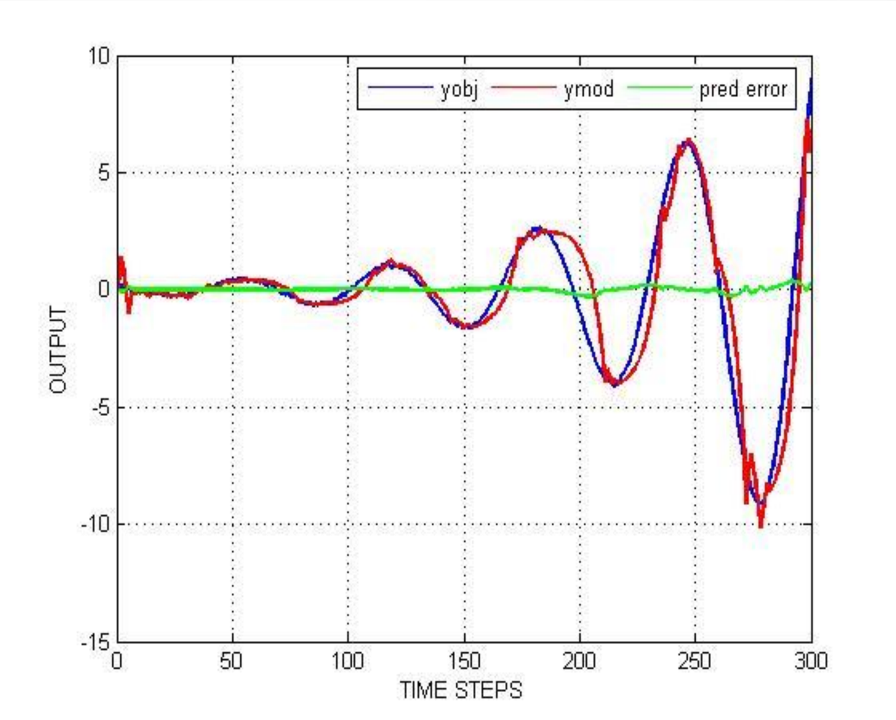




Modeling of Mackey-Glass and Rossler chaotic time series and the estimated error in the noiseless case.

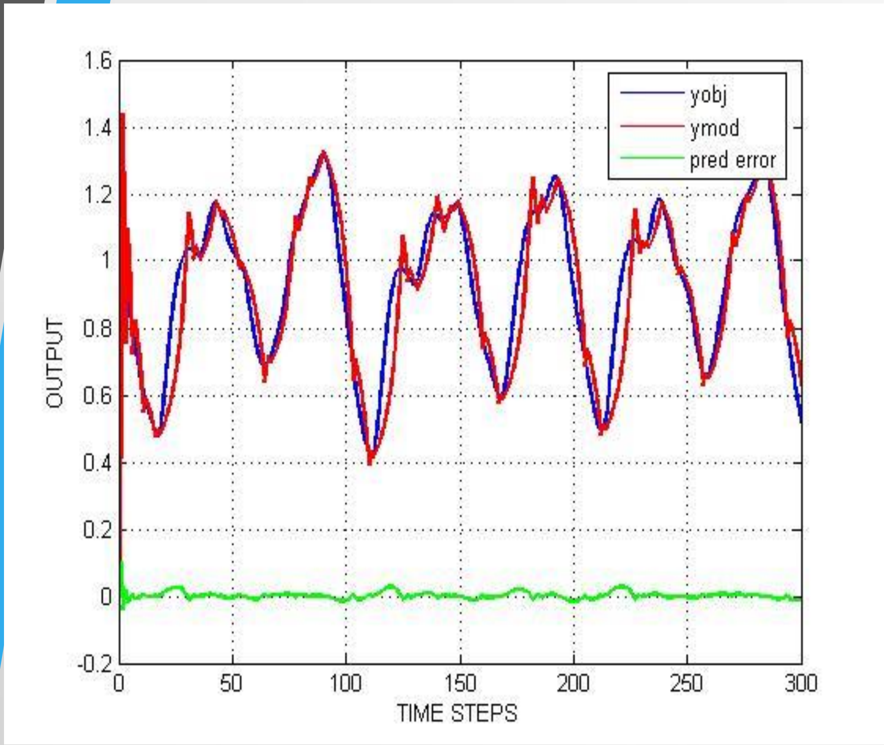
Numerical examples

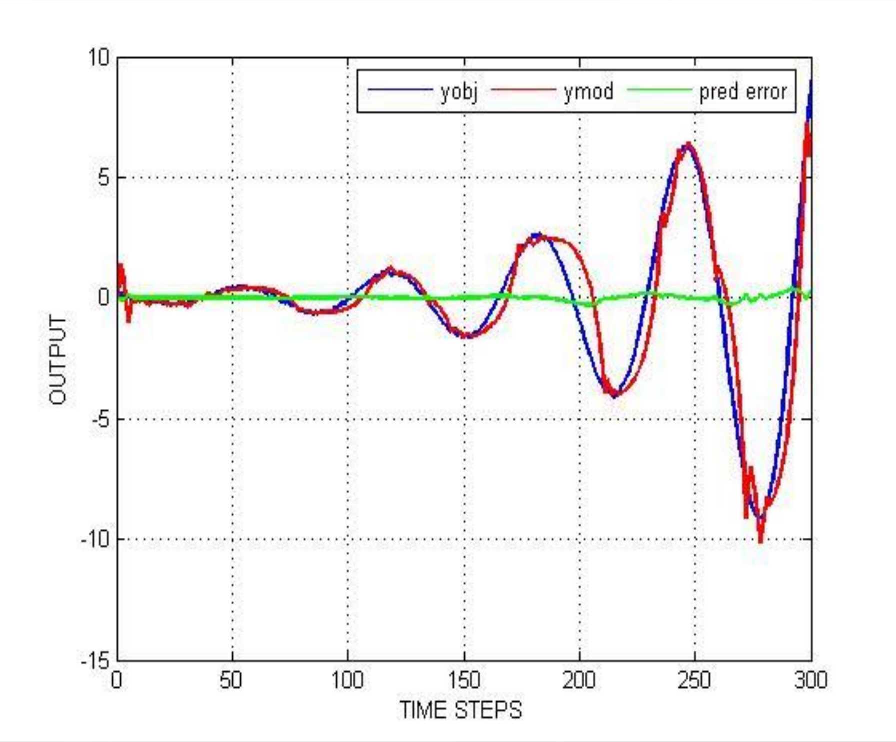




Modeling of Mackey-Glass and Rossler chaotic time series and the estimated error in the case of 5% additive noise and 5% FOU.

Numerical examples

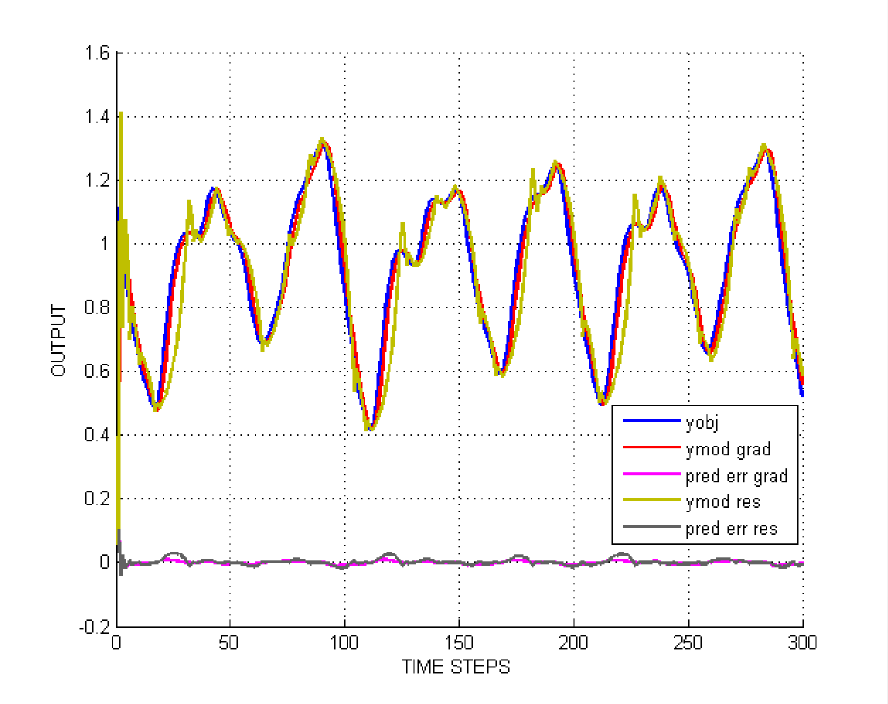




Modeling of Mackey-Glass chaotic and Rossler time series and the estimated error in the case of 5% additive noise and 10% FOU.

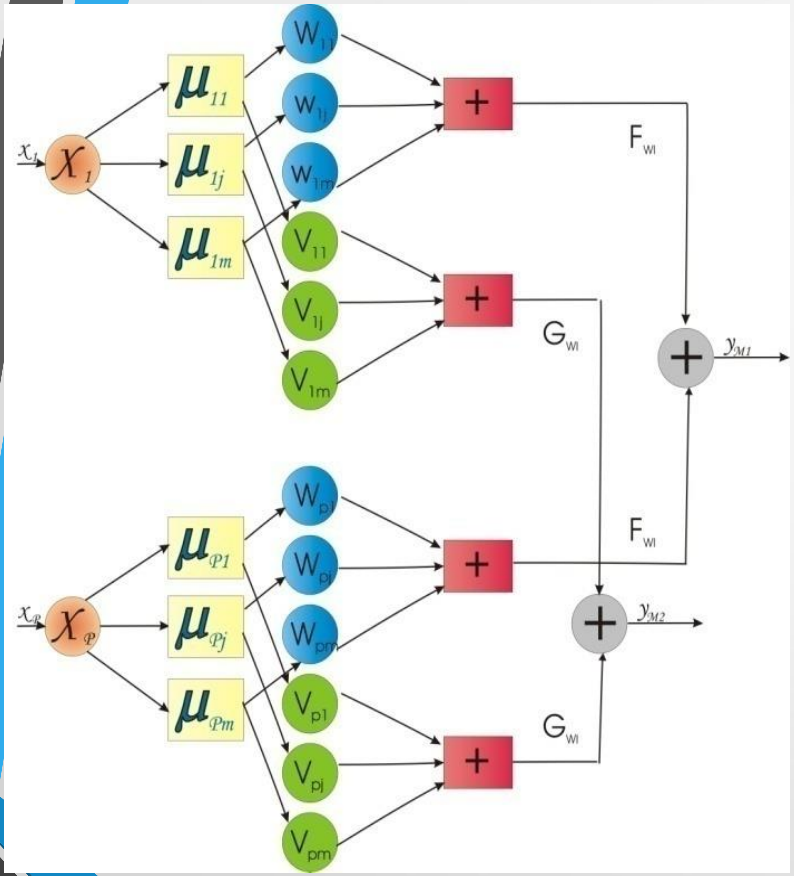
Numerical examples

Mean Squared Errors			
Time step	Without noise 10^{-4}	With noise and 5% FOU, 10^{-4}	With noise and 10% FOU, 10^{-4}
50	4.70	4.66	4.62
100	2.86	2.70	2.64
150	3.37	3.90	2.95
200	8.07	7.47	6.97
250	39.88	22.33	21.82
300	81.71	72.81	70.13



Comparison of the proposed heuristic algorithm to the classical Gradient Descent.

MIMO NEO-fuzzy network



MIMO Neo-Fuzzy Network

The MIMO NEO-Fuzzy Network is five-layer structure.

On the third layer the obtained membership degrees are multiplied by two different group weight coefficients $w_{ji}(k)$ and $v_{ji}(k)$. On the fourth layer are computed two groups of functions:

$$F_w(x) = \sum_{j=1}^{m} \mu_j(x(k))w_j \qquad G_w(x) = \sum_{j=1}^{m} \mu_j(x(k))v_j$$

The outputs of the MIMO NEO-Fuzzy Network are obtained on the last fifth layer:

$$y_{m1} = \sum_{j=1}^{p} F_{wj}(x) \qquad y_{m2} = \sum_{j=1}^{p} G_{wj}(x)$$

Learning of MIMO NEO-fuzzy network

- In the MIMO Neo-Fuzzy Network is need to be adjusted only one group of parameters – the consequents.
- The defined error cost terms are being minimized at each sampling period in order to update the synaptic weights:

$$E_2 = {\varepsilon^2}_2 / 2 \quad \varepsilon_2 = y_{m2}(k) - \hat{y}_{m2}(k) \qquad E_1 = {\varepsilon^2}_1 / 2 \quad \varepsilon_1 = y_{m1}(k) - \hat{y}_{m1}(k)$$

- The updating rules in which **w** and **v** are vectors of the trained parameters: the synaptic links in the consequent part of the rules and **η** is an adaptive learning rate:

$$\eta(k) = 0.1 * \left\| \mu_{ij}(x_i(k)) \right\|^2$$

$$v_{ij}(k+1) = v_{ij}(k) + \Delta v = v_{ij}(k) + \eta\left(\frac{\partial E_2(k)}{\partial v(k)}\right)$$
$$= v_{ij}(k) + \eta e(k) \mu_{ij}(x_i(k))$$

$$w_{ij}(k+1) = w_{ij}(k) + \Delta w = w_{ij}(k) + \eta\left(\frac{\partial E_1(k)}{\partial w(k)}\right)$$
$$= w_{ij}(k) + \eta e(k) \mu_{ij}(x_i(k))$$

Learning of MIMO NEO-fuzzy network

- To demonstrate the ability of the proposed MIMO NEO-Fuzzy Network it is chosen to model the following nonlinear system:

$$y_1(k) = \frac{y_1^2(k-1)}{y_1^2(k-1)+1} + 0.5y_2(k-1)$$

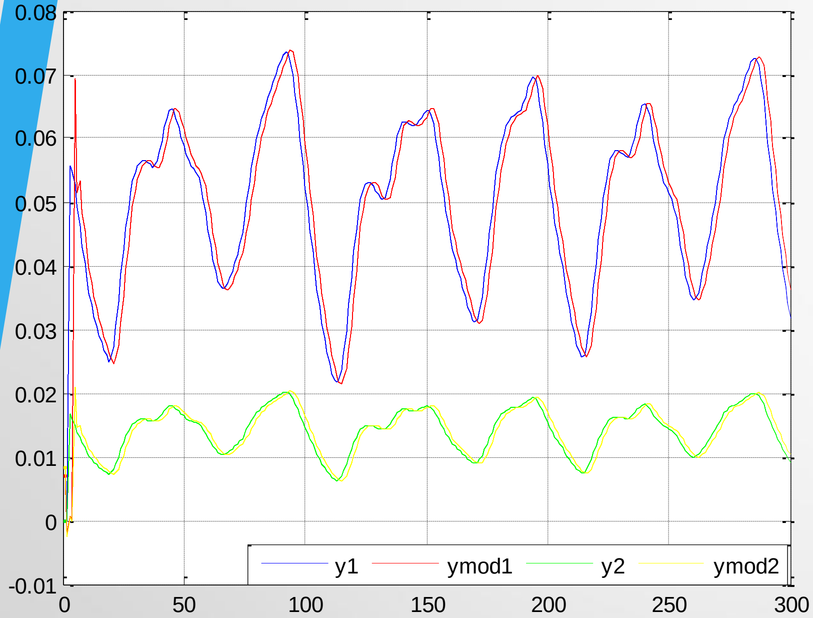
$$y_2(k) = \frac{y_1^2(k-1)}{y_2^2(k-1)+y_3^2(k-1)+y_4^2(k-1)} + u_1(k-1)$$

$$y_3(k) = \frac{y_3^2(k-1)}{y_3^2(k-1)+1} + 0.3y_4(k-1)$$

$$y_4(k) = \frac{y_3^2(k-1)}{y_1^2(k-1)+y_2^2(k-1)+y_4^2(k-1)} + 0.5u_2(k-1)$$

where u1(k) and u2(k) are the system inputs, y1(k) and y3(k) are the outputs. Two benchmark chaotic systems (Mackey-Glass and Rossler chaotic time series) are used as inputs:

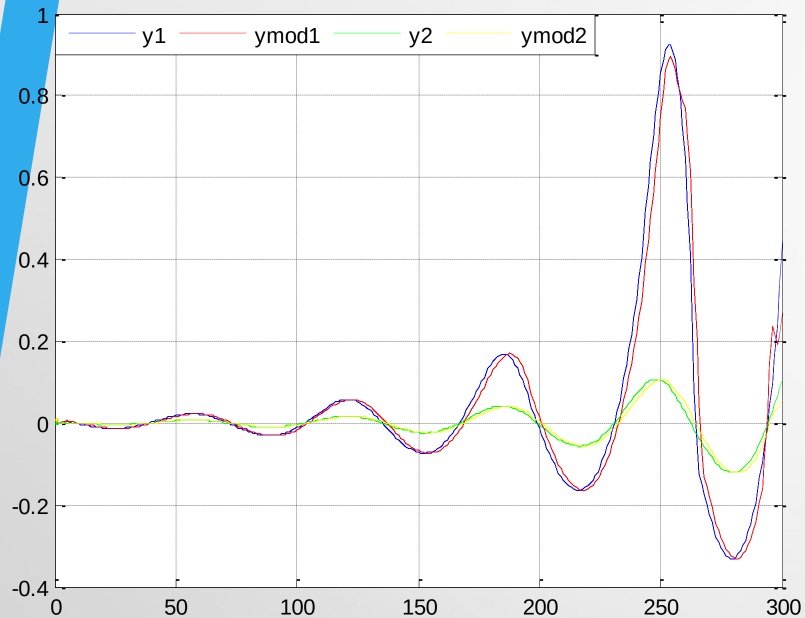
Numerical Examples



Steps	RMSE1	MSE1	RMSE2	MSE2
50	2.0e-4	5.5e-8	6.1e-5	3.7e-9
100	1.8e-4	4.8e-8	4.6e-5	3.4e-9
150	5.5e-5	3.7e-8	3.4e-5	2.8e-9
200	4.2e-5	3.1e-8	3.1e-5	2.2e-9
250	3.8e-5	2.4e-8	2.7e-5	1.8e-9
300	3.5e-5	1.9e-8	2.1e-5	1.3e-9

Model validation by using Mackey-Glass chaotic time series as inputs

Numerical Examples



Model validation by using Rossler chaotic time series as inputs

Steps	RMSE1	MSE1	RMSE2	MSE2
50	1.69e-4	2.85e-8	4.8e-5	2.32e-9
100	1.62e-4	2.6e-8	4.97e-5	3.1e-9
150	1.58e-4	2.5e-8	5.6e-5	6.2e-9
200	9.8e-3	9.7e-7	2.7e-4	7.5e-8
250	6.4e-3	4.1e-5	1.5e-4	2.1e-7
300	3.8e-3	1.5e-6	1.2e-4	4.4e-6

THANK YOU
FOR YOUR ATTENTION!