Chapter 7

Rule Based Fuzzy Systems

7.1. Canonical form of IF THEN rules

7.2. Inference graphical techniques

7.3. Defuzzification methods

7.4. Nonlinear fuzzy modelling

7.5. Introduction to fuzzy control

7.1. Canonical form IF THEN

$$\text{IF } \underbrace{\text{premise (antecedent)}}_{\text{Fuzzy set}} \text{ THEN } \underbrace{\text{conclusion (consequent)}}_{\text{Fuzzy set}}$$

Any rule can be reduced to this canonical form

Reduction to the canonical form

IF THEN ELSE

IF p is big THEN y is small ELSE y is not small

IF p is big THEN y is small

IF p not big THEN y is not small

$$\text{IF } \underset{\sim}{A}^1 \text{ THEN } \underset{\sim}{B}^1 \text{ ELSE } \underset{\sim}{B}^2 \equiv \begin{cases} \text{IF } \underset{\sim}{A}^1 \text{ THEN } \underset{\sim}{B}^1 \\ \text{IF } \overline{\underset{\sim}{A}^1} \text{ THEN } \underset{\sim}{B}^2 \end{cases}$$

Reduction to the canonical form

IF THEN UNLESS

$$\text{IF } \underset{\sim}{A}^1 \text{ THEN } \underset{\sim}{B}^1 \text{ UNLESS } \underset{\sim}{A}^2 \equiv \begin{cases} \text{IF } \underset{\sim}{A}^1 \text{ THEN } \underset{\sim}{B}^1 \\ \text{IF } \underset{\sim}{A}^2 \text{ THEN } \overline{\underset{\sim}{B}^1} \end{cases}$$

Chained IF THEN rules

$$\text{IF } \underset{\sim}{A}^1 \text{ THEN (IF } \underset{\sim}{A}^2 \text{ THEN } \underset{\sim}{B}^2) \equiv \text{IF } (\underset{\sim}{A}^1 \text{ AND } \underset{\sim}{A}^2) \text{ THEN } \underset{\sim}{B}^2$$

Case of several antecedents

conjunctives

IF p is $\underset{\sim}{A}^1$ and $\underset{\sim}{A}^2$ and... $\underset{\sim}{A}^n$ THEN y é $\underset{\sim}{B}$

$$\underset{\sim}{A}^S = \underset{\sim}{A}^1 \cap \underset{\sim}{A}^2 \cap \ldots \cap \underset{\sim}{A}^n$$

$$\mu_{\underset{\sim}{A}^S}(p) = min(\mu_{\underset{\sim}{A}^1}(p), \mu_{\underset{\sim}{A}^2}(p), \ldots, \mu_{\underset{\sim}{A}^n}(p))$$

IF p is $\underset{\sim}{A}^S$ THEN y is $\underset{\sim}{B}$

disjunctives

$$\text{IF } p \text{ is } \underset{\sim}{A}^1 \ or\ \underset{\sim}{A}^2 \ or \ldots or\ \underset{\sim}{A}^n \text{ THEN } y \text{ is } \underset{\sim}{B}$$

$$\underset{\sim}{A}^S = \underset{\sim}{A}^1 \cup \underset{\sim}{A}^2 \cup \ldots \cup \underset{\sim}{A}^n$$

$$\mu_{\underset{\sim}{A}^S}(p) = max(\mu_{\underset{\sim}{A}^1}(p), \mu_{\underset{\sim}{A}^2}(p), \ldots, \mu_{\underset{\sim}{A}^n}(p))$$

$$\text{IF } p \text{ is } \underset{\sim}{A}^S \text{ THEN } y \text{ is } \underset{\sim}{B}$$

- In the canonical form IF THEN, each rule is an implication and can be reduced to a relation and to a relational matrix.

- A set of rules can then be reduced to a set of relations.

Aggregation of fuzzy rules

Several rules (r):

Global consequent = aggregation of the consequents of the individual rules.

Conjunctive rules

(simultaneously satisfied, linked by the connective AND)

$y^i \equiv$ consequent of rule i (fuzzy set)

$y = y^1$ AND y^2 AND ... AND y^r

$y = y^1 \cap y^2 \cap \ldots \cap y^r$

$$\mu(y) = min[\mu(y^1), \mu(y^2), \ldots, \mu(y^r)]$$

Disjunctive Rules

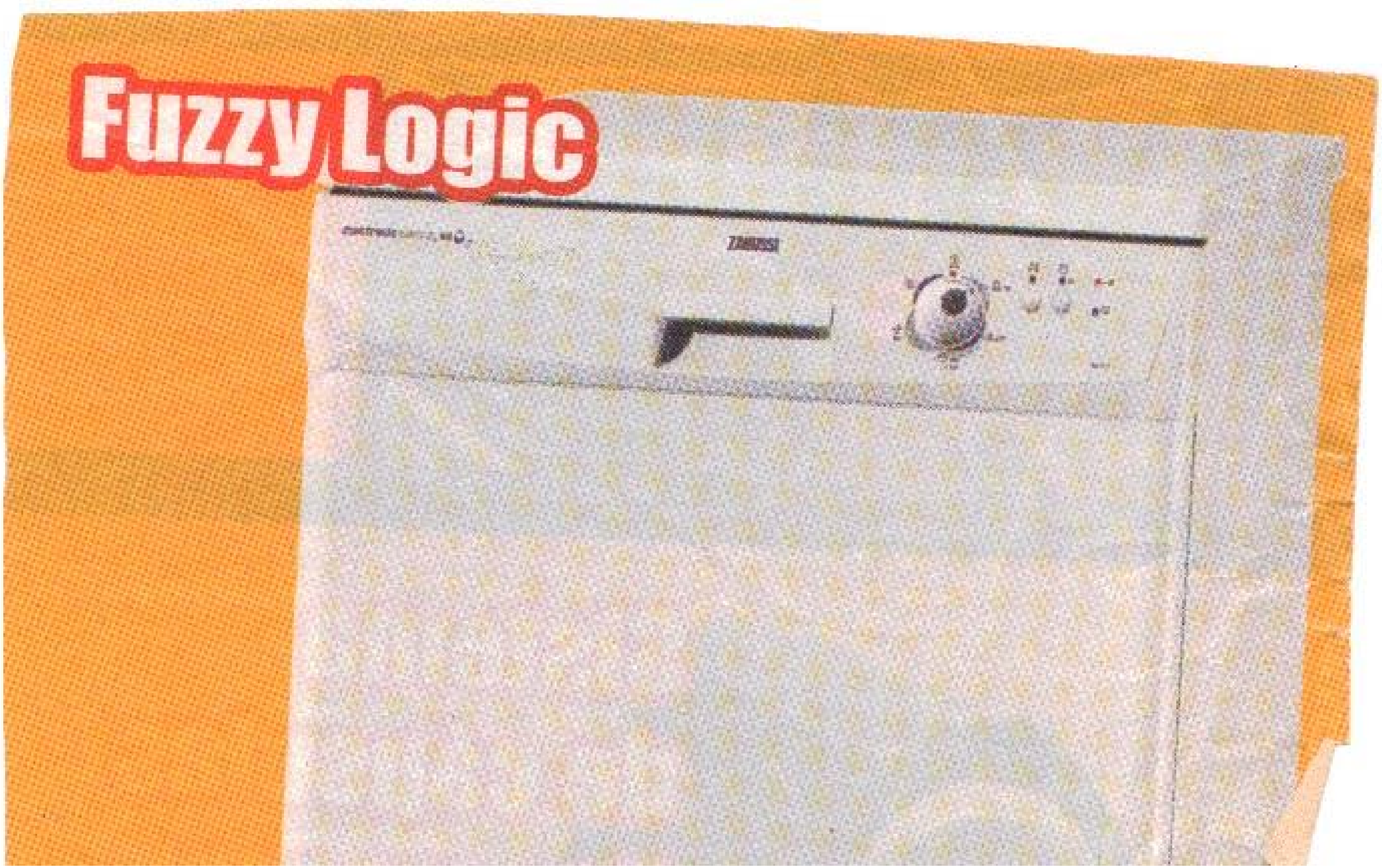
(alternatively satisfied, linked by the connective OR)

$y^i \equiv$ consequent of rule i (fuzzy set)

$y = y^1$ OR y^2 OR … OR y^r

$y = y^1 \cup y^2 \cup \ldots \cup y^r$

$$\mu(y) = max[\mu(y^1), \mu(y^2), \ldots, \mu(y^r)]$$

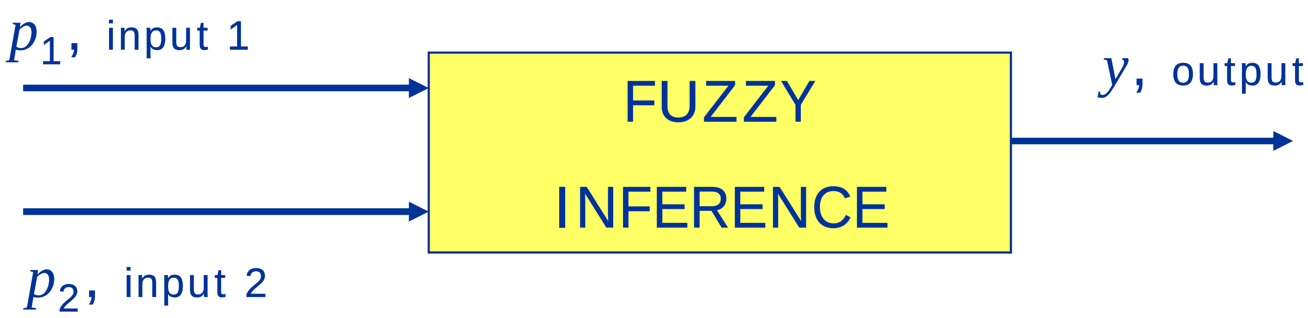


Thousands of fuzzy logic related patents issued, principally from industrial companies.

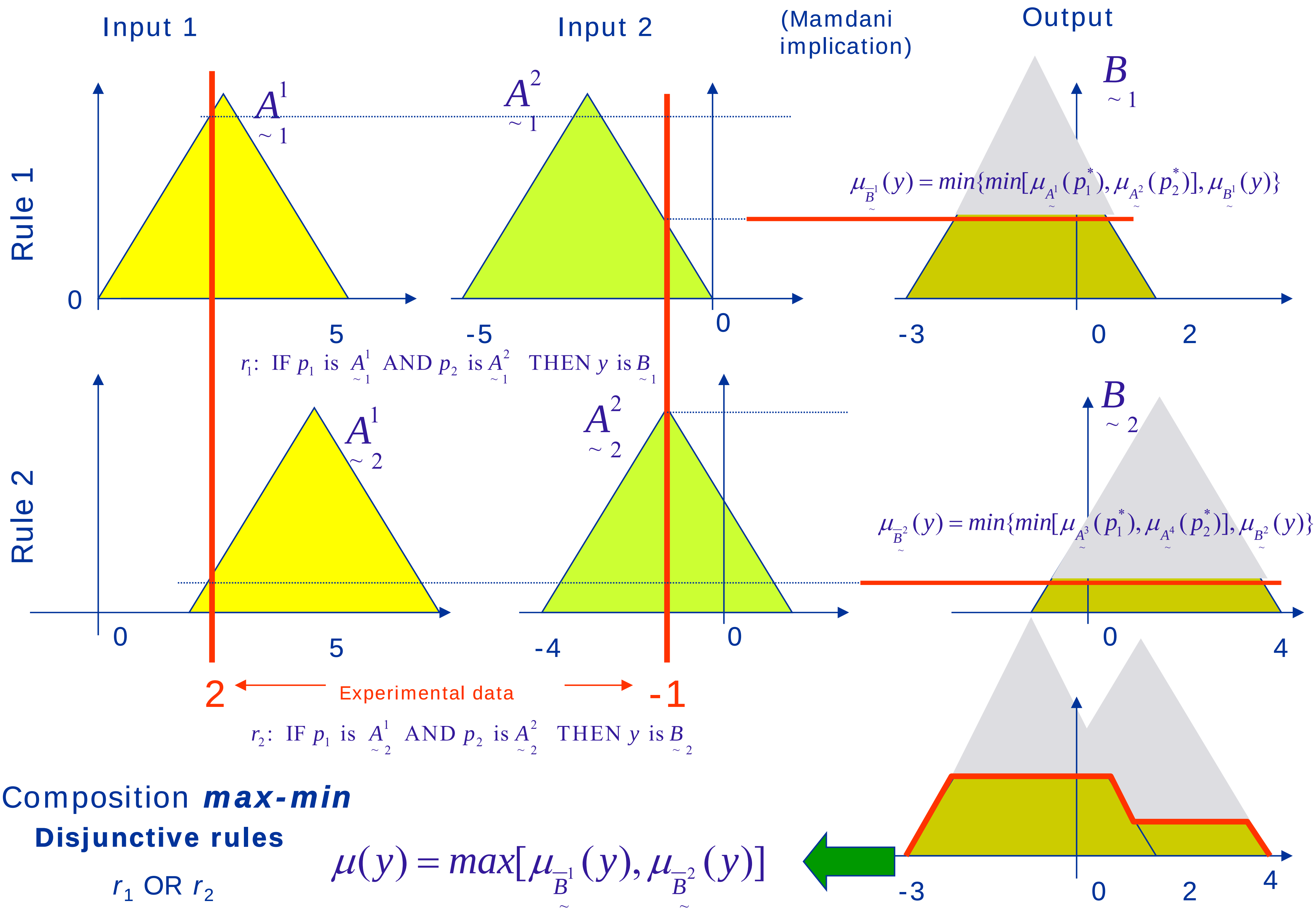
7.2. Graphical inference techniques

Practical, simple and intuitive, an alternative to the analytical techniques based on the matricial operations of composition.

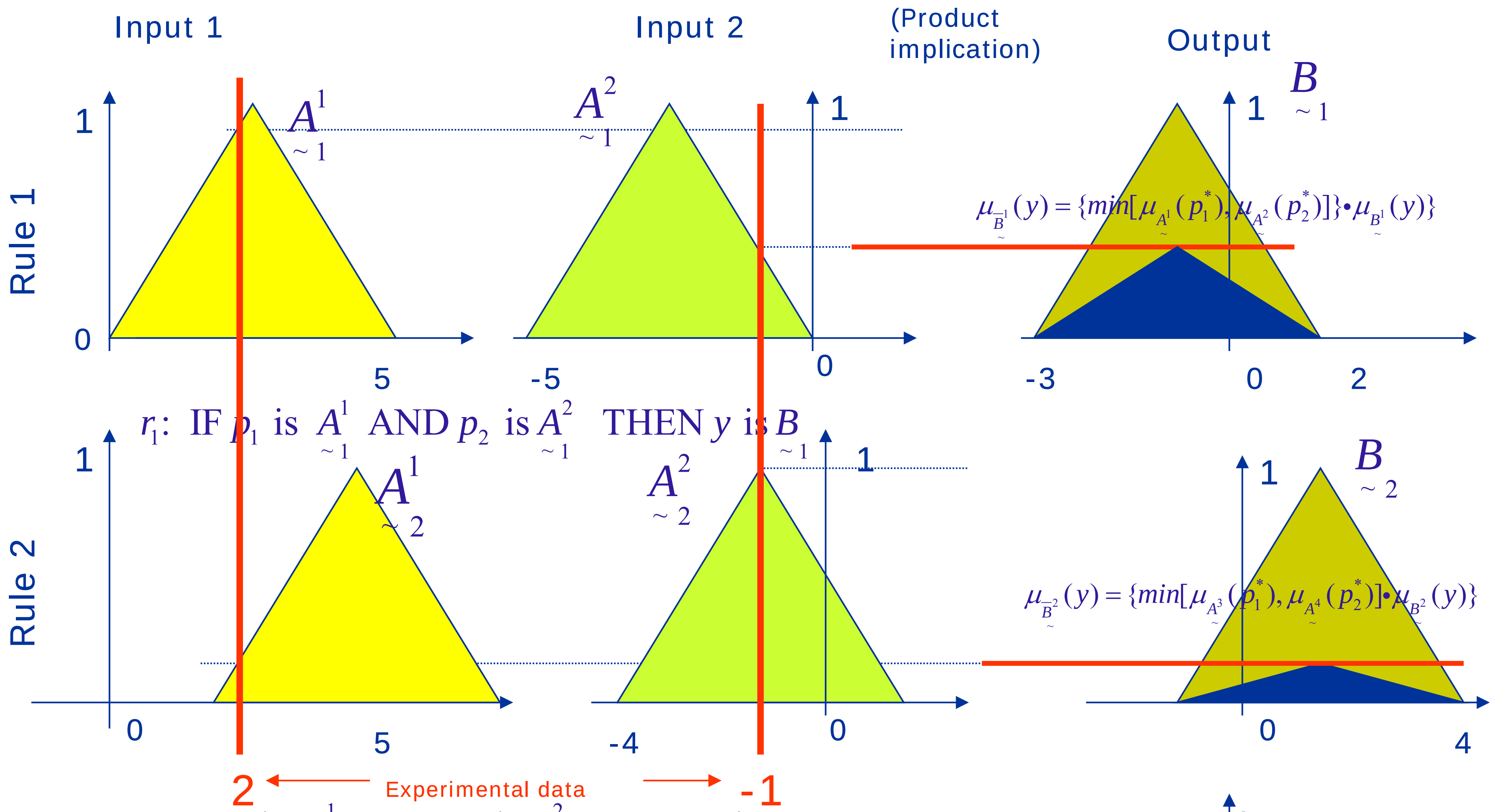
Case of two antecedents and one consequent:



Mamdani implication



Input 1
Input 2
(Mamdani implication)
Output
Rule 1
Rule 2
$\underset{\sim}{A}_1^1$
$\underset{\sim}{A}_1^2$
$\underset{\sim}{B}_1$
$\underset{\sim}{A}_2^1$
$\underset{\sim}{A}_2^2$
$\underset{\sim}{B}_2$
0
5
-5
0
-3
0
2
0
5
-4
0
0
4
-3
0
2
4
$\mu_{\underset{\sim}{B}^1}(y) = min\{min[\mu_{\underset{\sim}{A}^1}(p_1^*), \mu_{\underset{\sim}{A}^2}(p_2^*)], \mu_{\underset{\sim}{B}^1}(y)\}$
r_1: IF p_1 is $\underset{\sim}{A}_1^1$ AND p_2 is $\underset{\sim}{A}_1^2$ THEN y is $\underset{\sim}{B}_1$
$\mu_{\underset{\sim}{B}^2}(y) = min\{min[\mu_{\underset{\sim}{A}^3}(p_1^*), \mu_{\underset{\sim}{A}^4}(p_2^*)], \mu_{\underset{\sim}{B}^2}(y)\}$
2
Experimental data
-1
r_2: IF p_1 is $\underset{\sim}{A}_2^1$ AND p_2 is $\underset{\sim}{A}_2^2$ THEN y is $\underset{\sim}{B}_2$
Composition ***max-min***
Disjunctive rules
r_1 OR r_2
$\mu(y) = max[\mu_{\underset{\sim}{B}^1}(y), \mu_{\underset{\sim}{B}^2}(y)]$

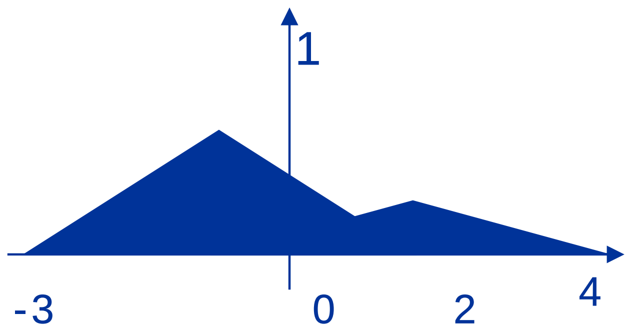


r_2: IF p_1 is $\underset{\sim 2}{A^1}$ AND p_2 is $\underset{\sim 2}{A^2}$ THEN y is $\underset{\sim 2}{B}$

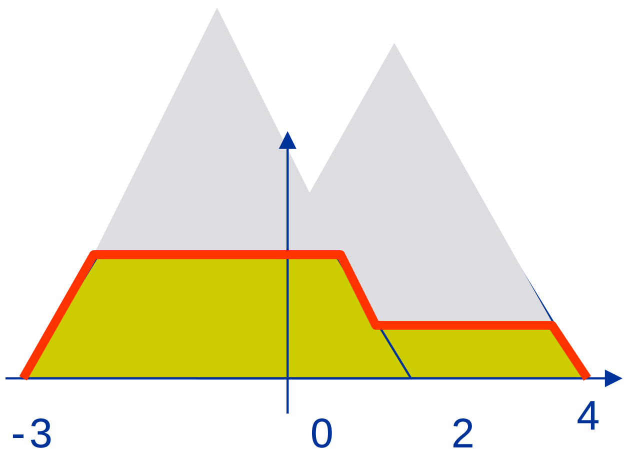
Composition ***max-prod***

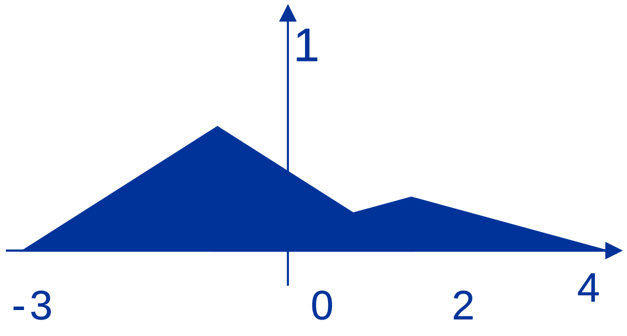
Disjunctive rules $\quad \mu(y) = max[\mu_{\underset{\sim}{\bar{B}^1}}(y)\}, \mu_{\underset{\sim}{\bar{B}^2}}(y)]$

r_1 OR r_2



Which is the output ?





Composition ***max-min***

Composition ***max-prod***

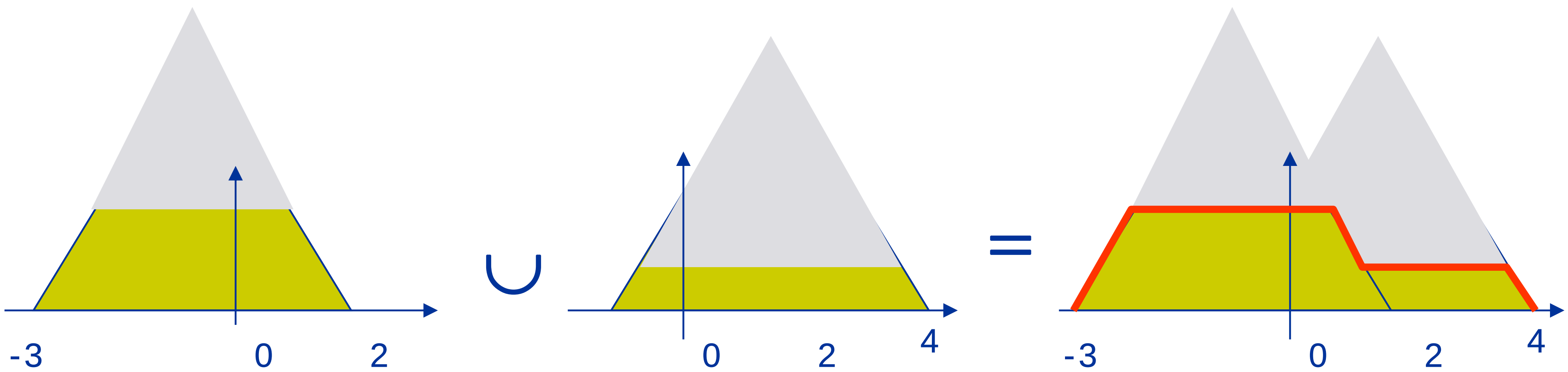
And if one needs a crisp value ? 0? 1? -1 ? ...

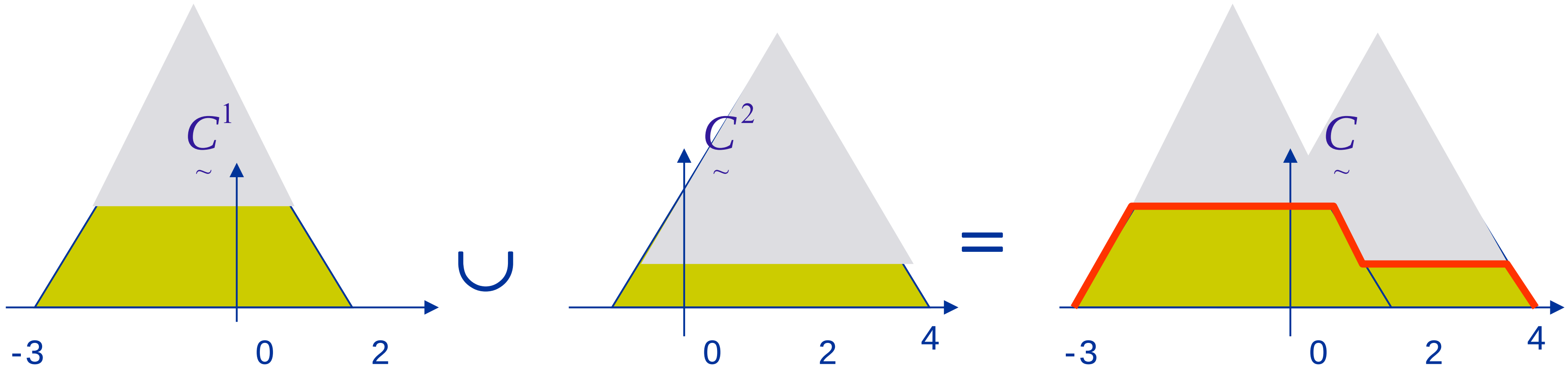
... defuzzification

7.3. Defuzzification methods

defuzzification:

- Conversion of a fuzzy (imprecise) quantity to a crisp (precise) quantity.
- The result of the inference of a set of fuzzy rules is the union of the fuzzy sets defined in the universe of discourse of the output.





$$\mu(y) = max[\mu_{\underset{\sim}{C}^1}(y)\}, \mu_{\underset{\sim}{C}^2}(y)]$$

And in general for r rules

$$\underset{\sim}{C} = \bigcup_{i=1}^{r} \underset{\sim}{C}^i$$

In general one has to defuzzify $\underset{\sim}{C}$

Some defuzzification methods

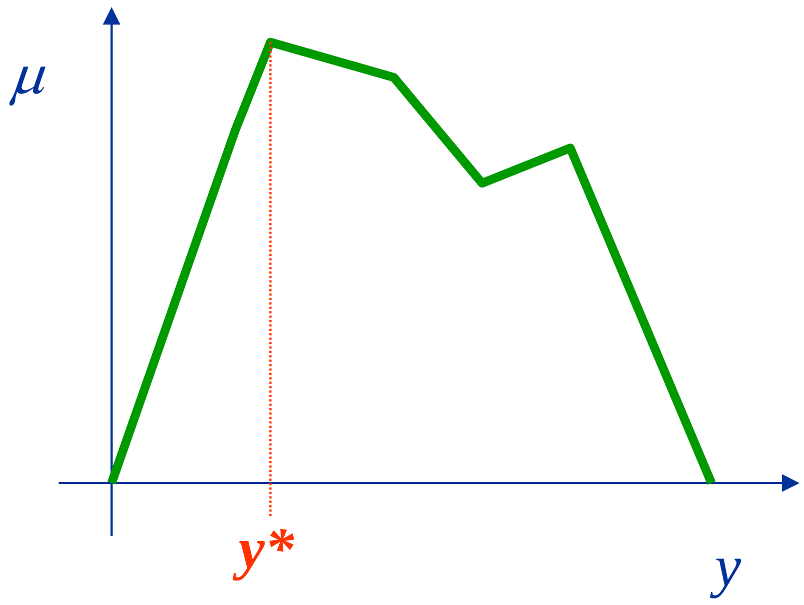
- height (or of maximum membership)
- the centroid (or of the center of gravity)
- the weighted average
- the mean of the maxima (or middle of the maxima)
- the first (or last) of the maxima (FOM or LOM)
- the center of the sums
- the bisector

The height method (maximum membership)

Applied to functions with picks

$$\mu_{\underset{\sim}{C}}(y^*) \geq \mu_{\underset{\sim}{C}}(y), \forall y \in Y$$

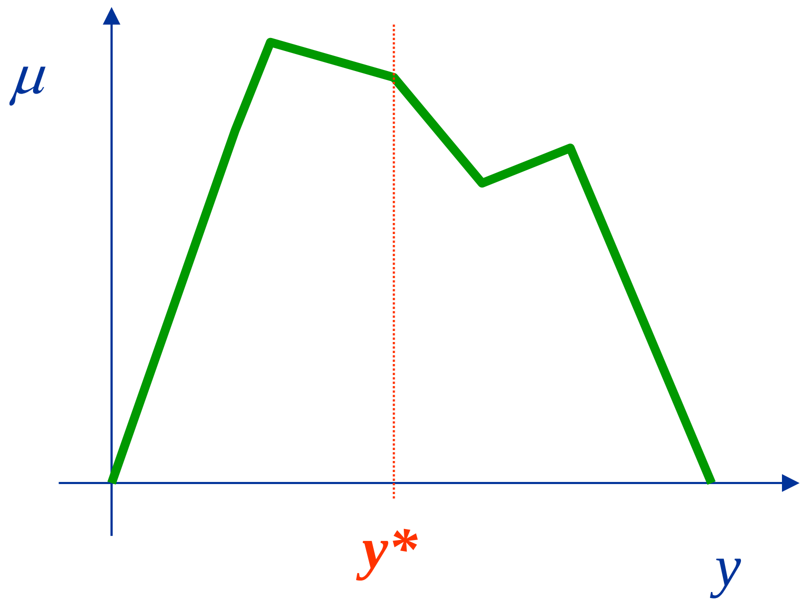
$$y^* : \mu_{\underset{\sim}{C}}(y^*) = \max_{Y} \mu_{\underset{\sim}{C}}(y)$$



Method of centroid, center of mass, or center of gravity

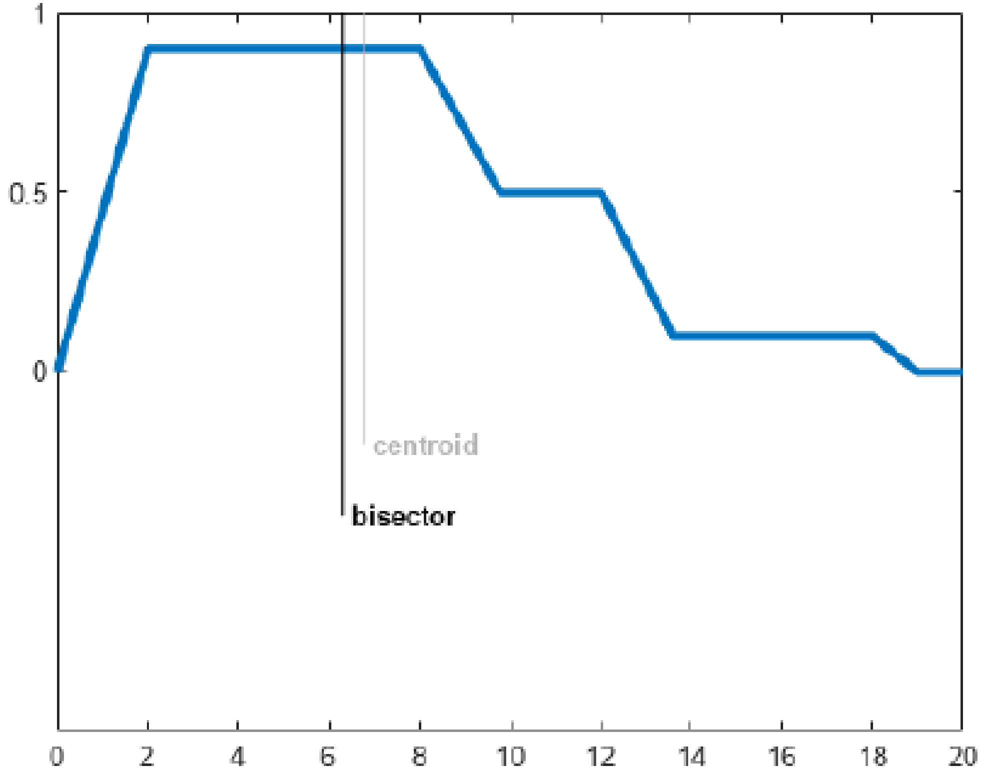
Calculates the center of gravity of the figure

$$y^* = \frac{\int \mu_{\underset{\sim}{C}}(y).y.dy}{\int \mu_{\underset{\sim}{C}}(y)dy}$$



The bisector method

Divides the output fuzzy set into two parts of equal área.
It produces results similar to the centroid method, but not always.



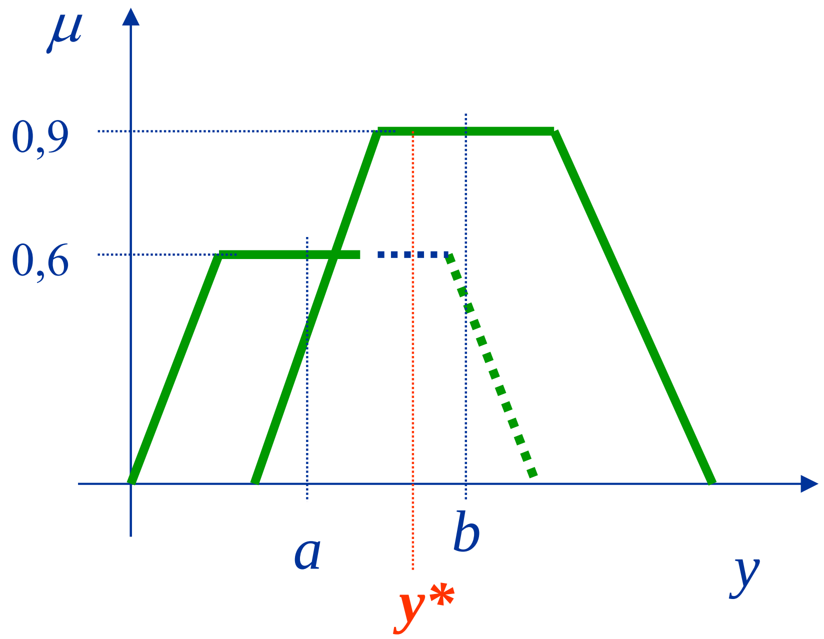
(from Matlab 2022b online help)

Method of the weighted average

Valid only for symmetric membership functions

$$y^* = \frac{\sum \mu_{\underset{\sim}{C}}(\bar{y})\bar{y}}{\sum \mu_{\underset{\sim}{C}}(\bar{y})}$$

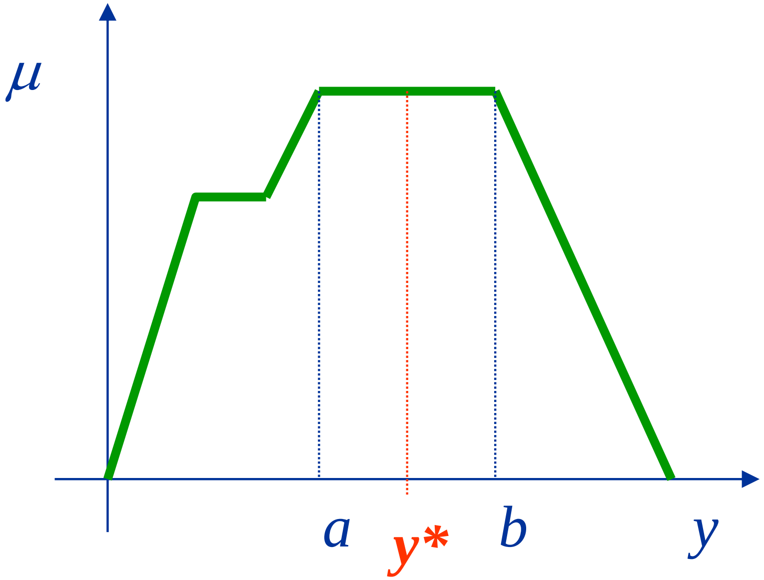
$$y^* = \frac{0,6a + 0,9b}{0,6 + 0,9}$$



Method of the mean of maxima

When the maximum is not unique (but a plateau)

$$y^* = \frac{a+b}{2}$$



Method of the first (or last) of the maxima

First of maxima:

the least value of de y having maximum membership

$$y^* = \inf\{y \in Y : \mu_{\underset{\sim}{C}}(y) = height(\underset{\sim}{C})\}$$

Last of the maxima:

the biggest value of y having maximum membership

$$y^* = \sup\{y \in Y : \mu_{\underset{\sim}{C}}(y) = height(\underset{\sim}{C})\}$$

Method of the center of sums

- One calculates the area of each fuzzy set resulting from each rule.

- One calculates their sum being each one weighted by its area

$$y^* = \frac{\int_y y.\sum_{k=1}^{r} \mu_{\underset{\sim}{C_k}}(y)dy}{\int_y \sum_{k=1}^{r} \mu_{\underset{\sim}{C_k}}(y)dy}$$

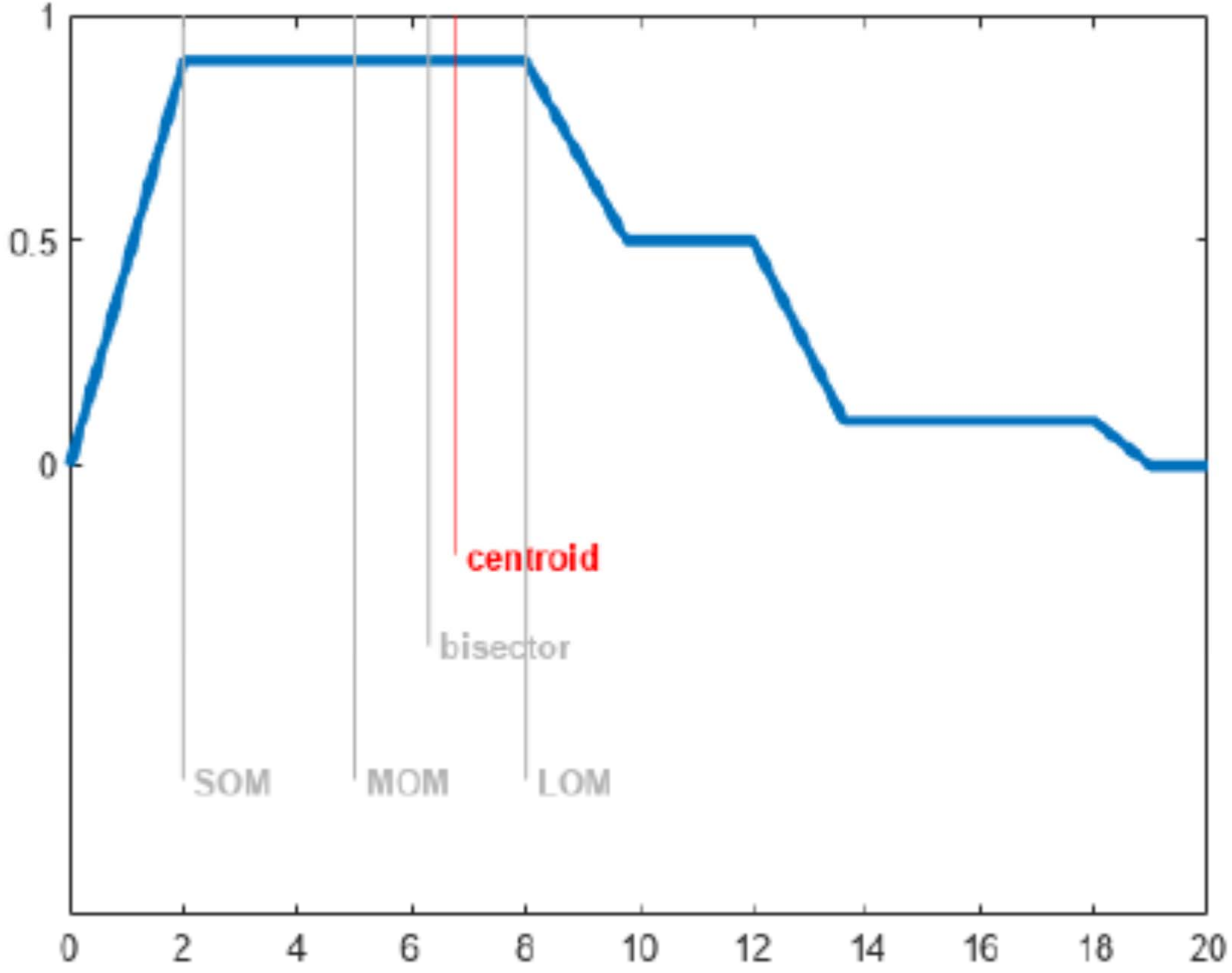
In the case of symmetrical membership functions

$$y^* = \frac{\sum_{k=1}^{r} \overline{y}_k \int_y \mu_{\underset{\sim}{C_k}}(y_k)dy}{\sum_{k=1}^{r} \int_y \mu_{\underset{\sim}{C_k}}(y_k)dy}$$

- It is a simplified version of the centroid method. It considers individually each fuzzy set from each rule. As a consequence, the intersection areas account twice or more. It treats the algebraic sum of the output fuzzy sets instead of their union.

- It is similar to the weighted average method in the case of symmetrical membership functions, but here the weights are the areas of the membership functions of the sets, while in there the weights are the individual values of the membership of the averages. For more see Ross, p. 105.

Comparison of methods (from Matlab 2022b online help)



7.4. Nonlinear fuzzy modelling

- When we do not have deep knowledge about the system that we want to model, preventing the modelling by differential or difference equations, for example.

- But there is information either numerical (from experimentation) or linguist (qualitative, expert knowledge) that can be formalized in a fuzzy sets' framework.

- Rules IF THEN or relational equations are used.

7.4.1. Modeling by IF THEN rules

The rules are built from:

- available knowledge, based on experimentation, on the empirical observation, on the intuition;

- input-output data, either numerical on non-numerical (verbal, qualitative)

For the construction of the rules:

(i) Find the scale of each input $[p_{i\min}, p_{i\max}]$ and the scale of each output $[y_{mín}, y_{máx}]$

(ii) Normalize the scales to the interval [-1,1] (or [0, 1])

Normalized inputs:
$$\overline{p}_i = -1 + \frac{2(p_i - p_{i\min})}{p_{i\max} - p_{i\min}}$$

Normalized output:
$$\overline{y} = -1 + \frac{2(y - y_{\min})}{y_{\max} - y_{\min}}$$

Normalization ....

$$\overline{p}_i = -1 + \frac{2(p_i - p_{i\min})}{p_{i\max} - p_{i\min}} \Leftrightarrow \overline{p}_i = \frac{p_i - \dfrac{(p_{i\max} + p_{i\min})}{2}}{\dfrac{p_{i\max} - p_{i\min}}{2}}$$

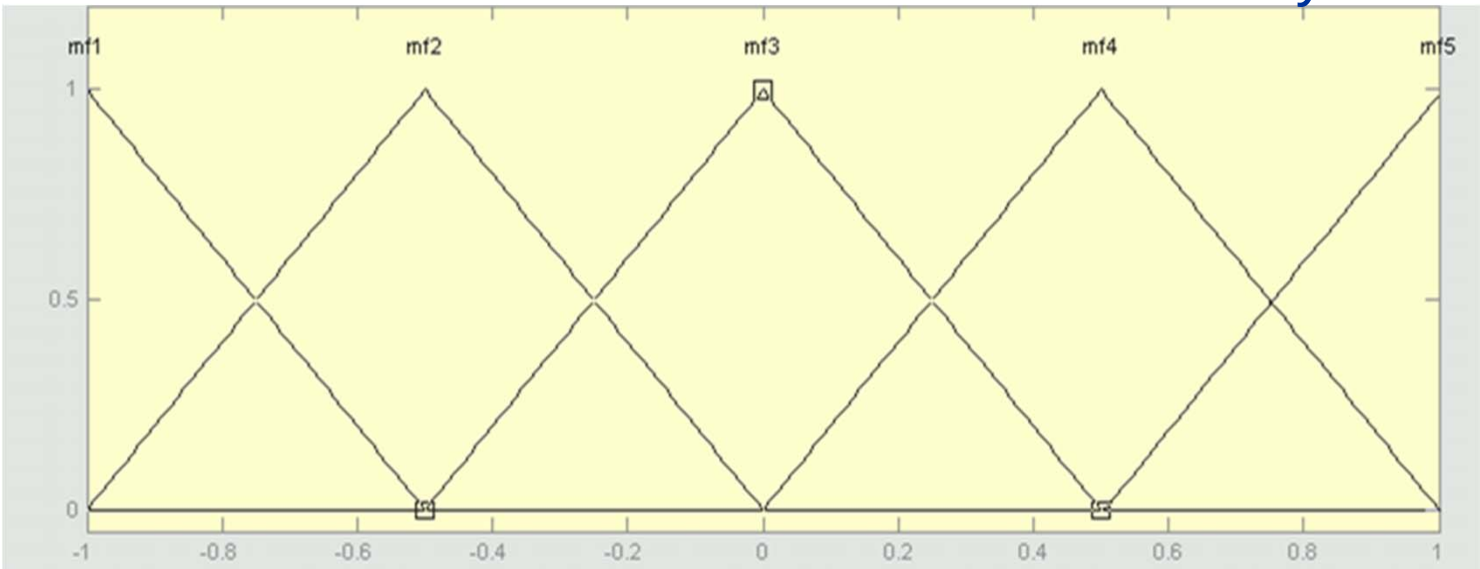
$$\overline{y} = -1 + \frac{2(y - y_{\min})}{y_{\max} - y_{\min}} \Leftrightarrow \overline{y} = \frac{y - \dfrac{(y_{\max} + y_{\min})}{2}}{\dfrac{y_{\max} - y_{\min}}{2}}$$

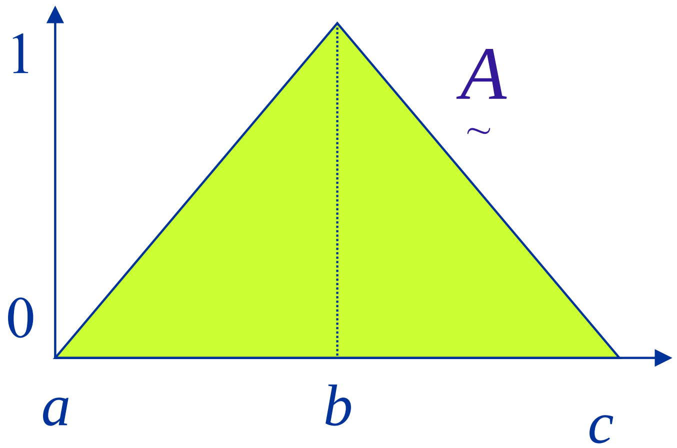
...and denormalization

$$\overline{y} = \frac{y - \dfrac{(y_{\max} + y_{\min})}{2}}{\dfrac{y_{\max} - y_{\min}}{2}} \Leftrightarrow y = \overline{y} \bullet \frac{(y_{\max} - y_{\min})}{2} + \frac{(y_{\max} + y_{\min})}{2}$$

(iii) Partition of the input spaces , P, and output spaces Y, into fuzzy sets

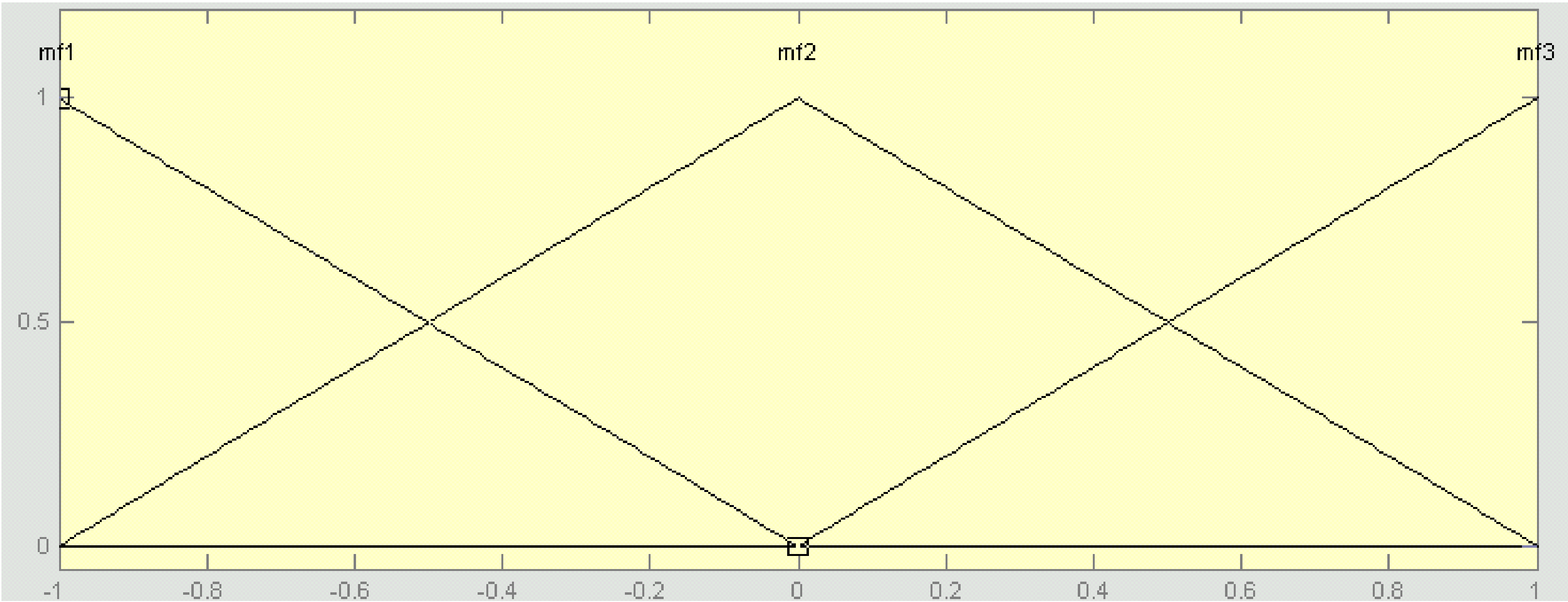
five fuzzy sets





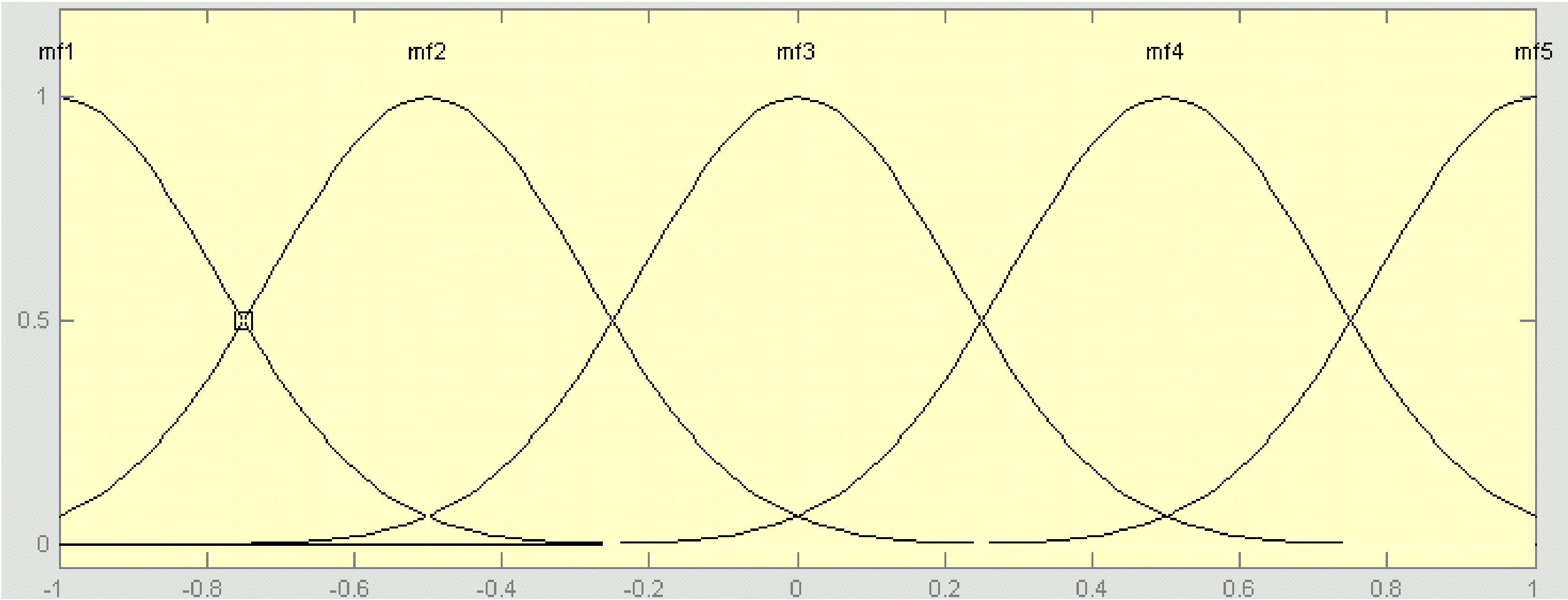
$$\mu_{\underset{\sim}{A}}(x) = \begin{cases} 0, & x \le a \\ \dfrac{x-a}{b-a}, & a \le x \le b \\ \dfrac{c-x}{c-b}, & b \le x \le c \\ 0, & x \ge c \end{cases}$$

three fuzzy sets

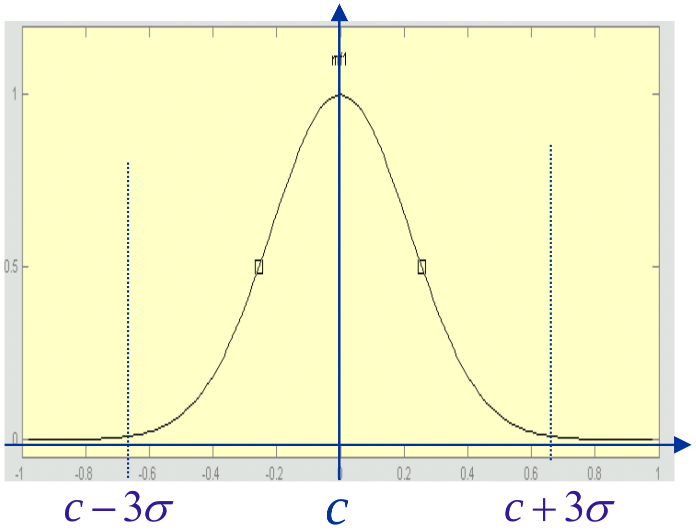


gaussian

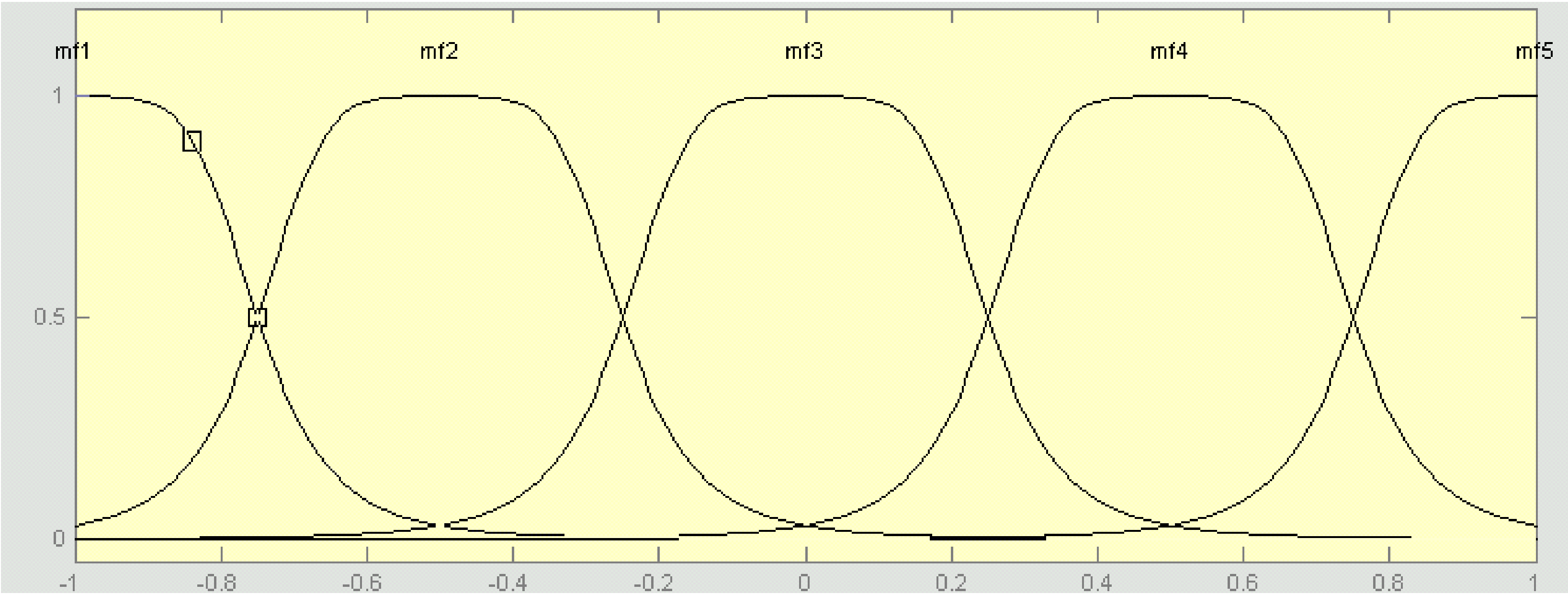
GAUSSMF(X, [SIGMA, C]) = EXP(-(X - C).^2/(2*SIGMA^2));



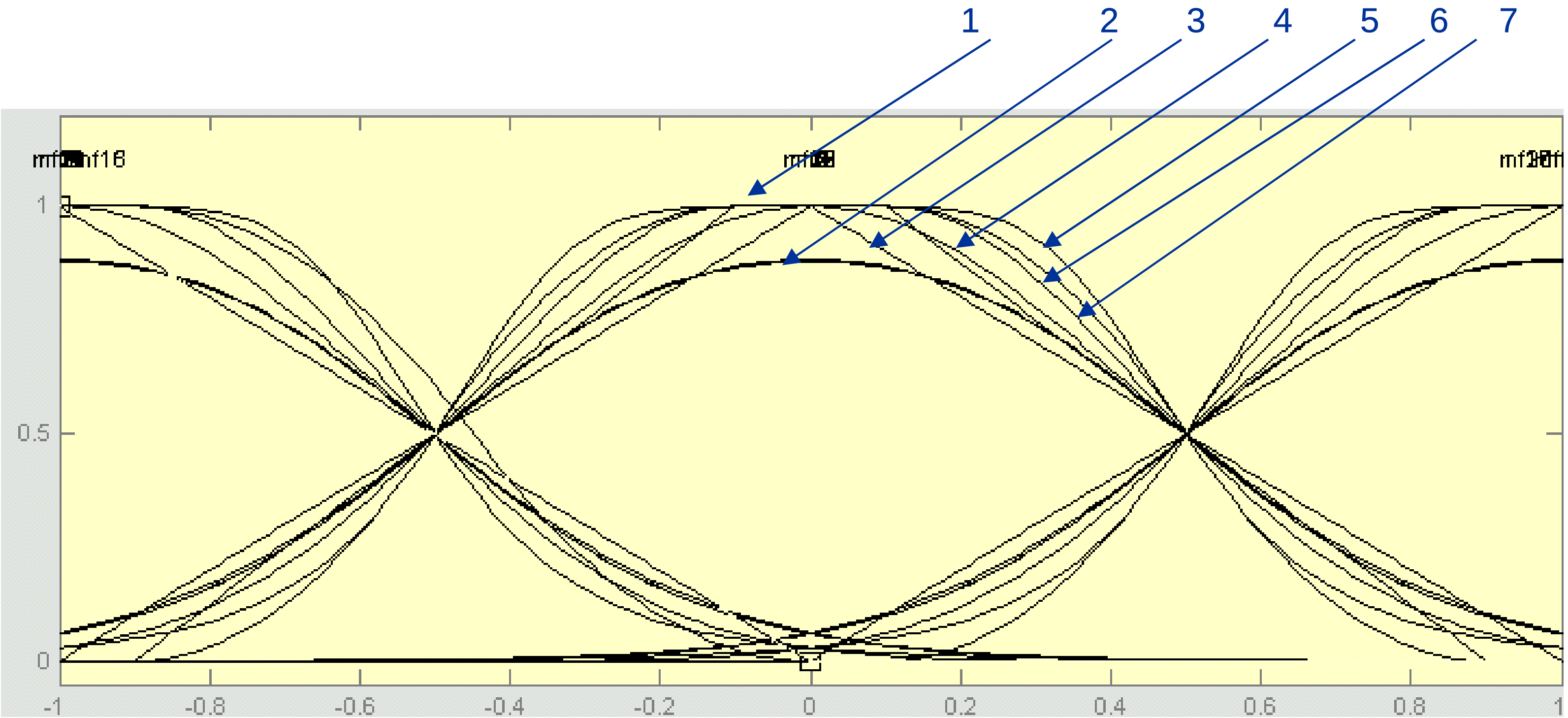
$$f(x) = e^{-\frac{(x-c)^2}{2\sigma^2}}$$



bellshaped

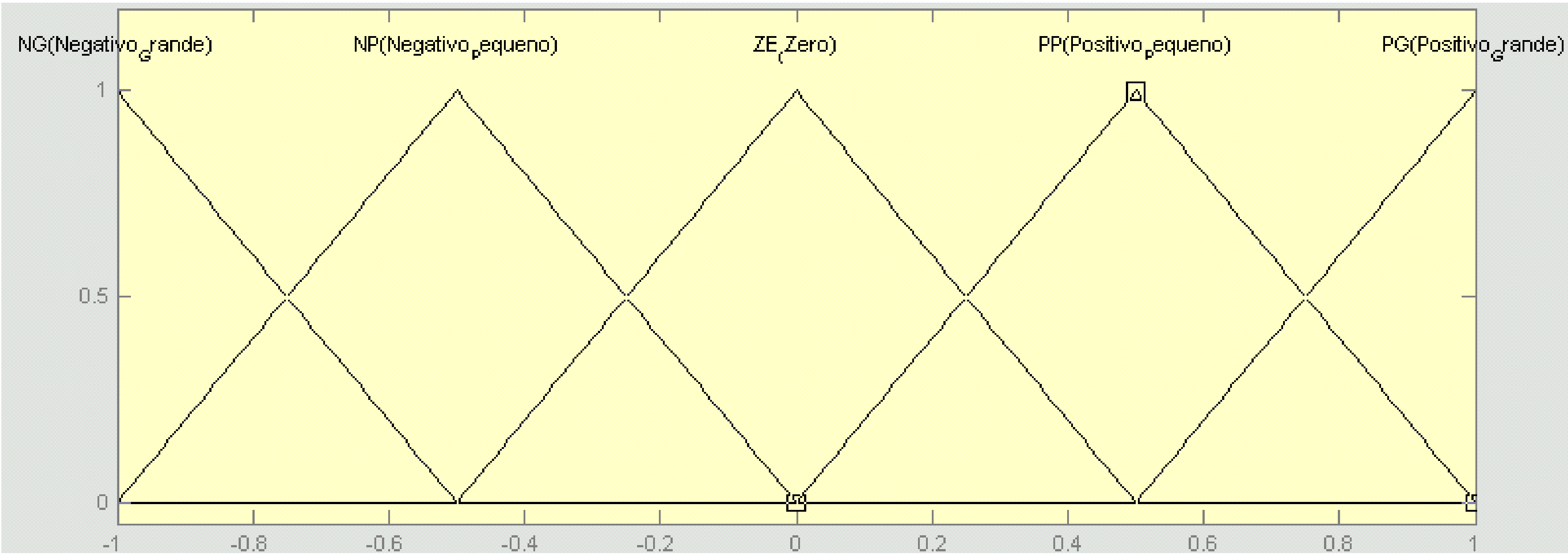


Comparison of the membership functions implemented in the graphical interface of the *Fuzzy Logic Toolbox*:



Actually there are more than 7, but these are the most used.

(iv) Label each of the fuzzy sets with a linguistic value , for example (in Portuguese) :



(v) write IF THEN rules using the available knowledge, for example,

IF input 1 is NG and input 2 is NG THEN Output1 is PP

IF input 1 is ZE and input 2 is PP THEN Output1 is NP

IF input 1 is PP and input 2 is PP THEN Output1 is NP

...

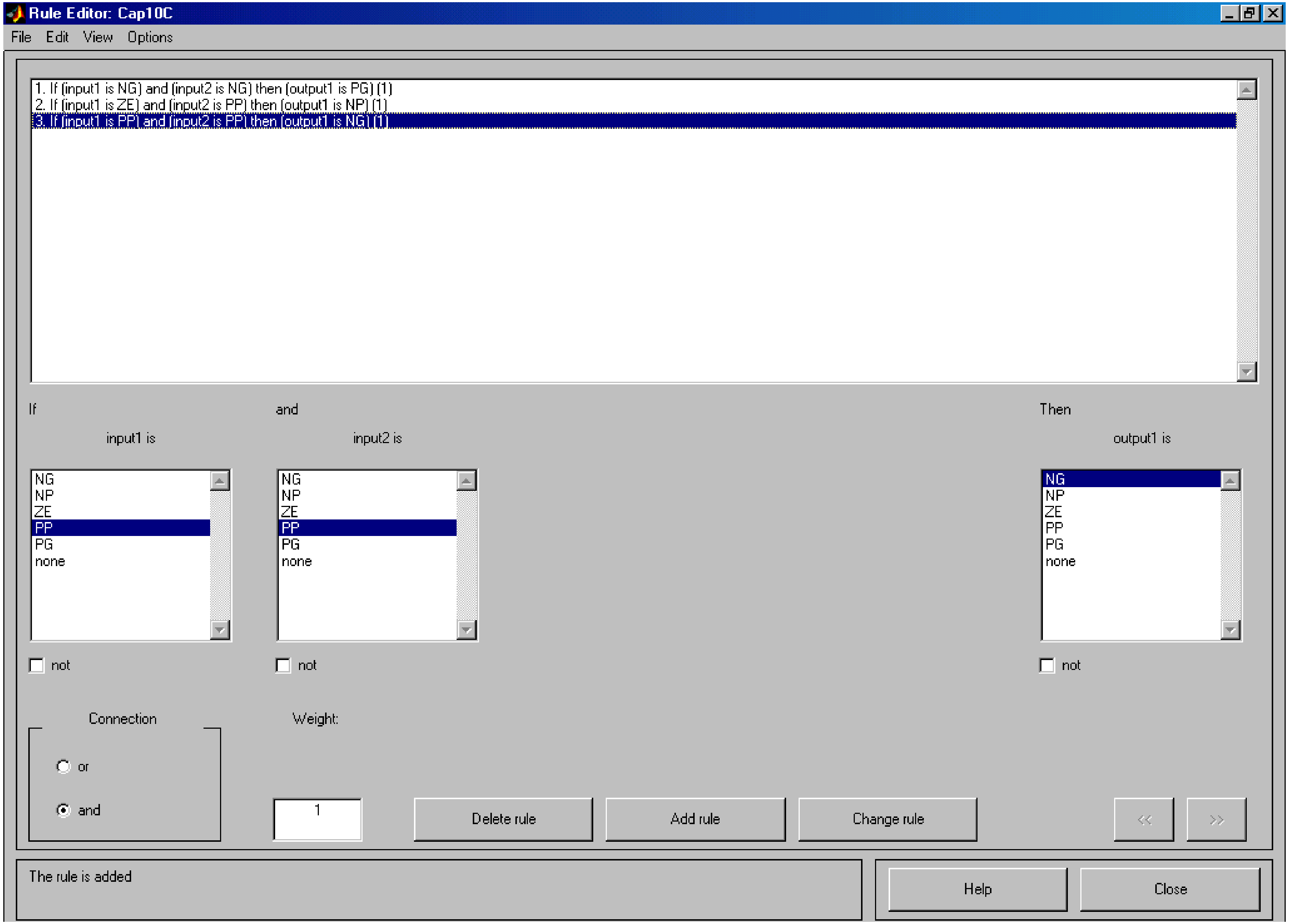
Maximum number of rules:

number of fuzzy sets of input 1 multiplied by the number of fuzzy sets of input 2, in the example,

$$5x5=25$$

Rule editor *(Fuzzy Logic Toolbox*, Mathworks)

>fuzzy



To compute the output y^* corresponding to a crisp input p^* :

(i) fuzzify the input p^*, identifying all the fuzzy sets to which p^* belongs, i.e., all $\underset{\sim}{A_i}$ such that

$$\mu_{\underset{\sim}{A_i}}(p^*) > 0$$

(ii) fire all the rules were these $\underset{\sim}{A_i}$ are antecedents, by the graphical method (*max-min* or *max-prod*), obtaining an output (fuzzy set) for each fired rule,

(iii) aggregate all the fuzzy sets obtained in (ii),

(iv) apply a defuzzification method to compute the corresponding crisp value y^*.

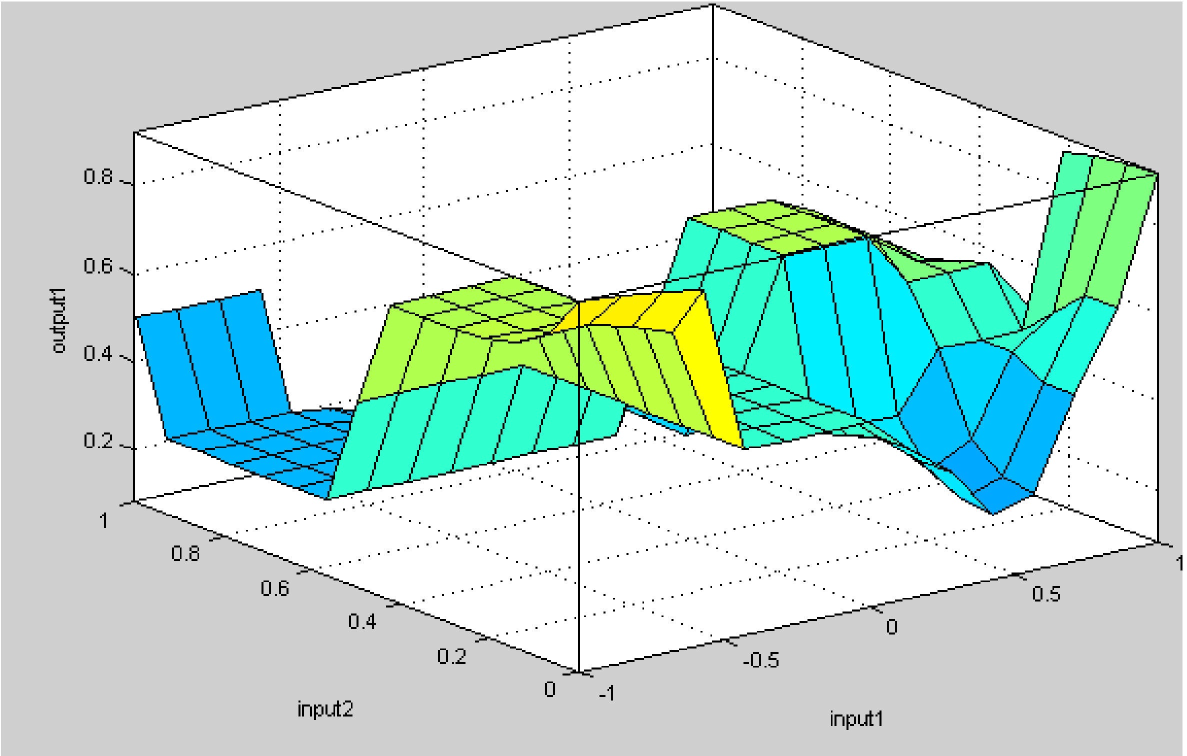
- If there are two inputs and one output, each rule has two antecedents and one consequent.

- If there are r inputs and one output, each rule has r antecedents and one consequent.

IF p_1 is PS and ... and p_r is NP THEN y is NB

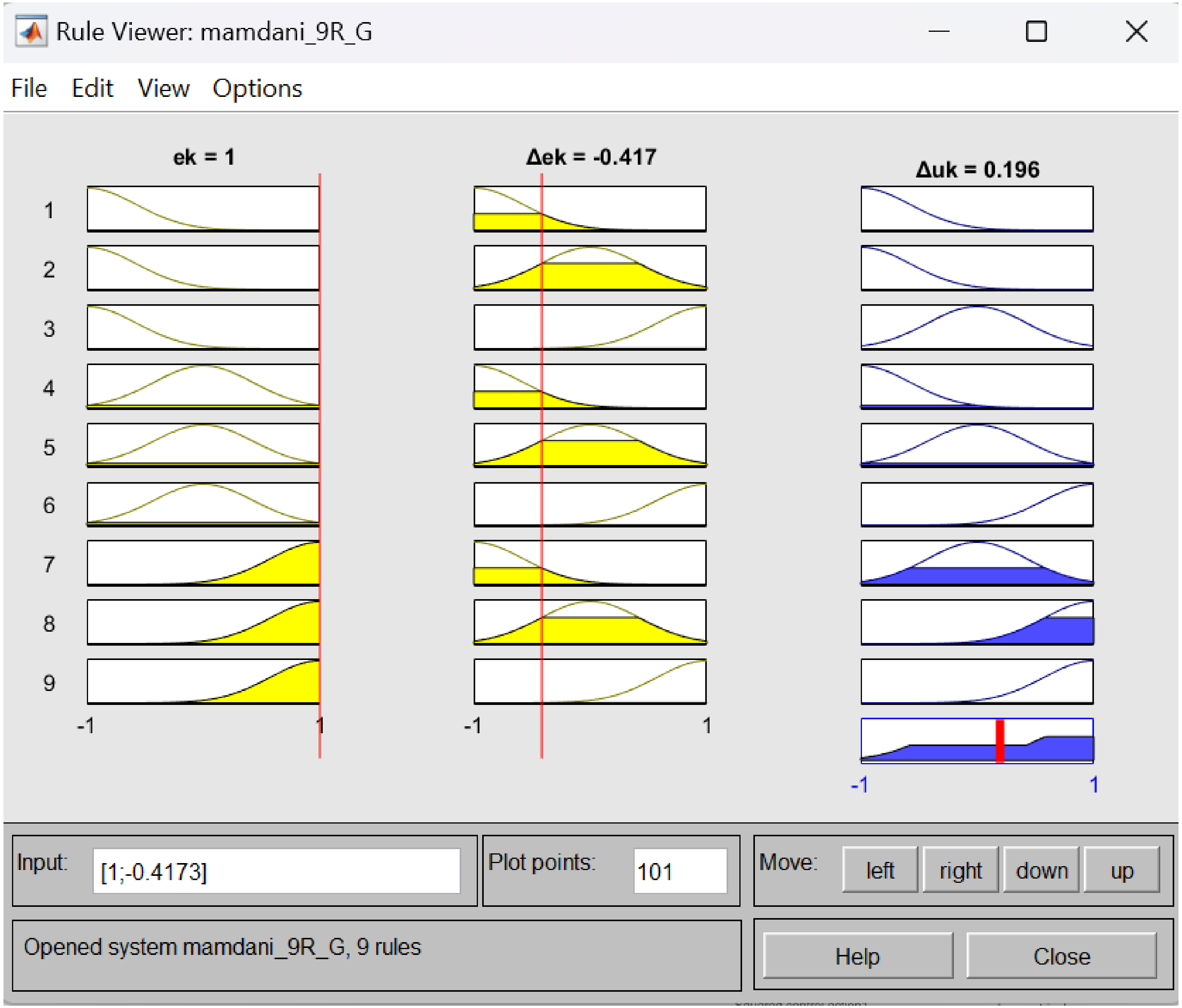
The surface of the output (*view surface*)

After obtaining the rules, the output can be computed varying the inputs along their scales and plotting. For example, for two inputs and one output

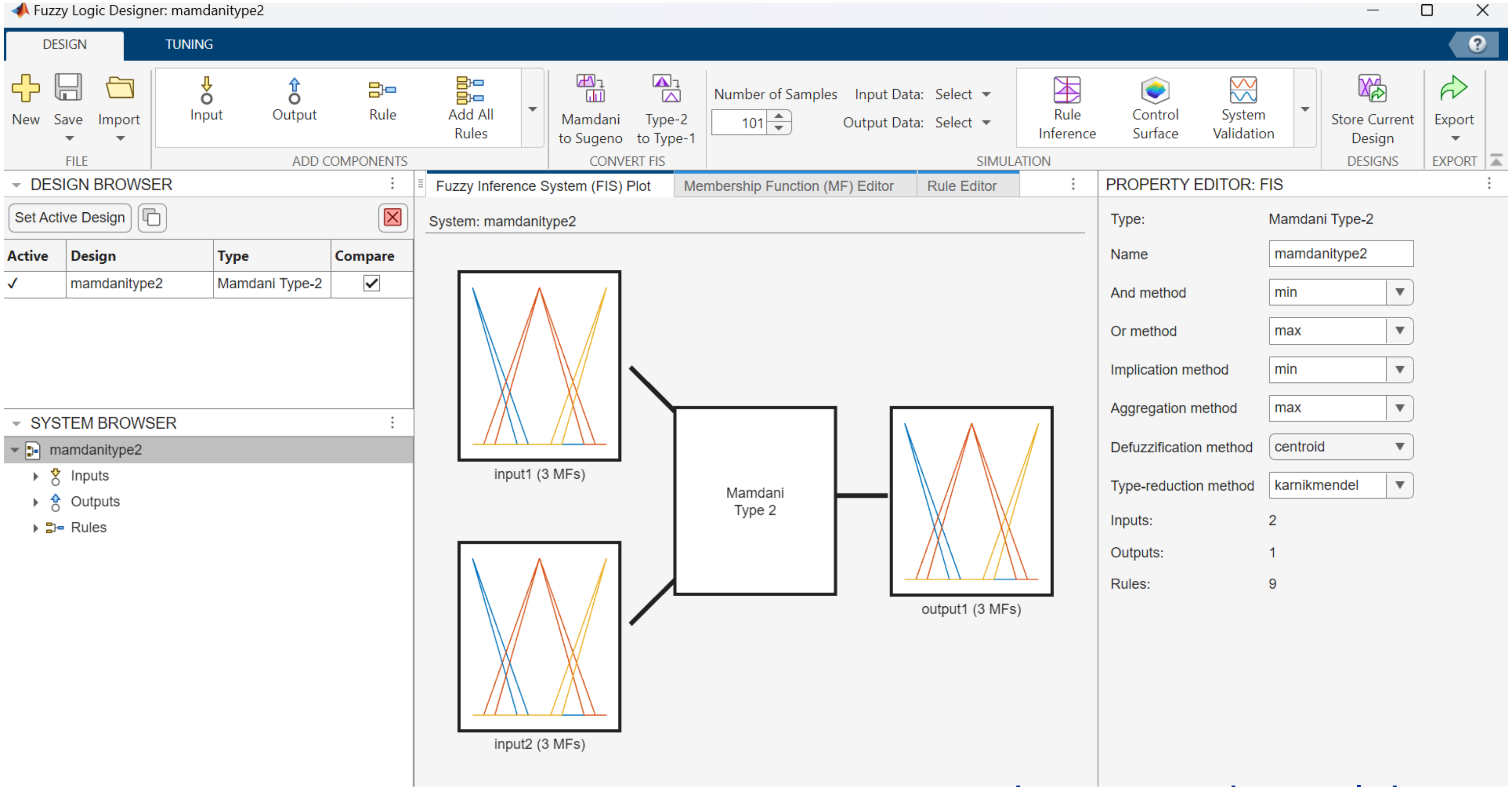


...allows to visualize the system's behavior.

Simulation (*view rules*)



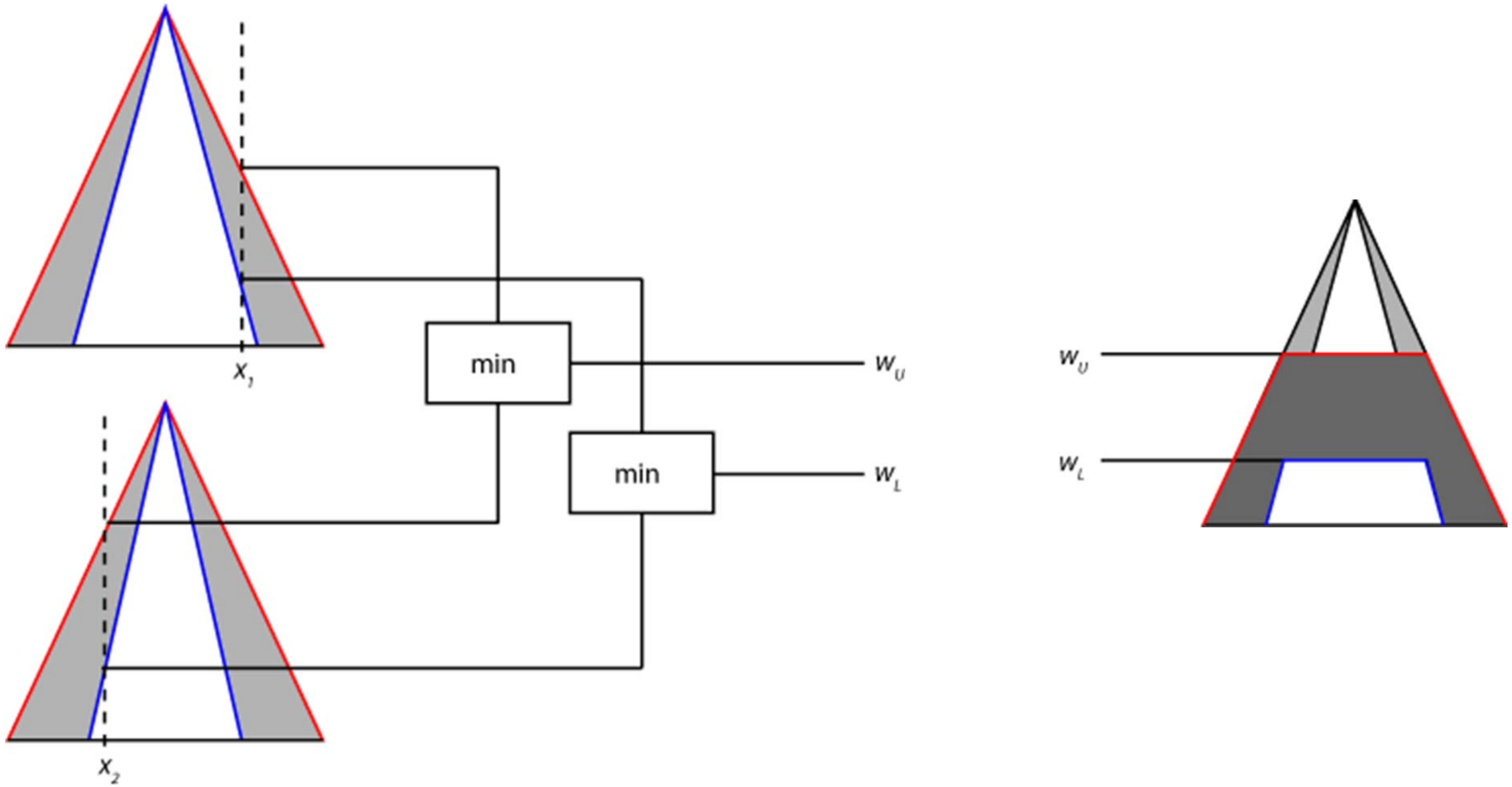
Actually there exist a new interface reachable by
> fuzzyLogicDesigner



Printscreen do Matlab

It includes Type-2 fuzzy systems and more utilities

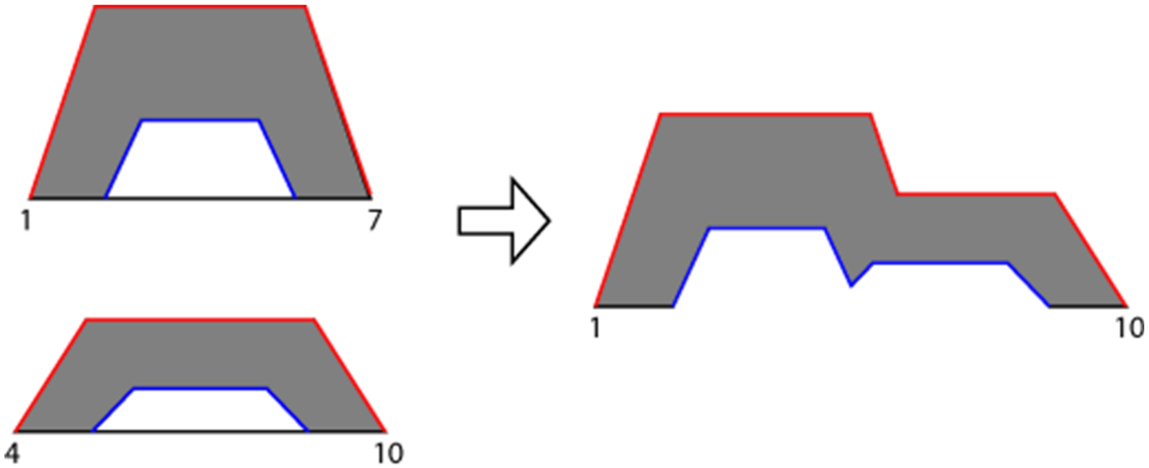
Type-2 Fuzzy logic



A rule fires with two intensities, that will be used in the other operations. See Matlab help

https://www.mathworks.com/help/fuzzy/type-2-fuzzy-inference-systems.html

Aggregation



Defuzzification

Produces two values, the final output is the average of these two.

7.4.2. Takagi-Sugeno-Kang inference (type-1 fuzzy systems)

TSK (Takagi-Sugeno-Kang) fuzzy systems

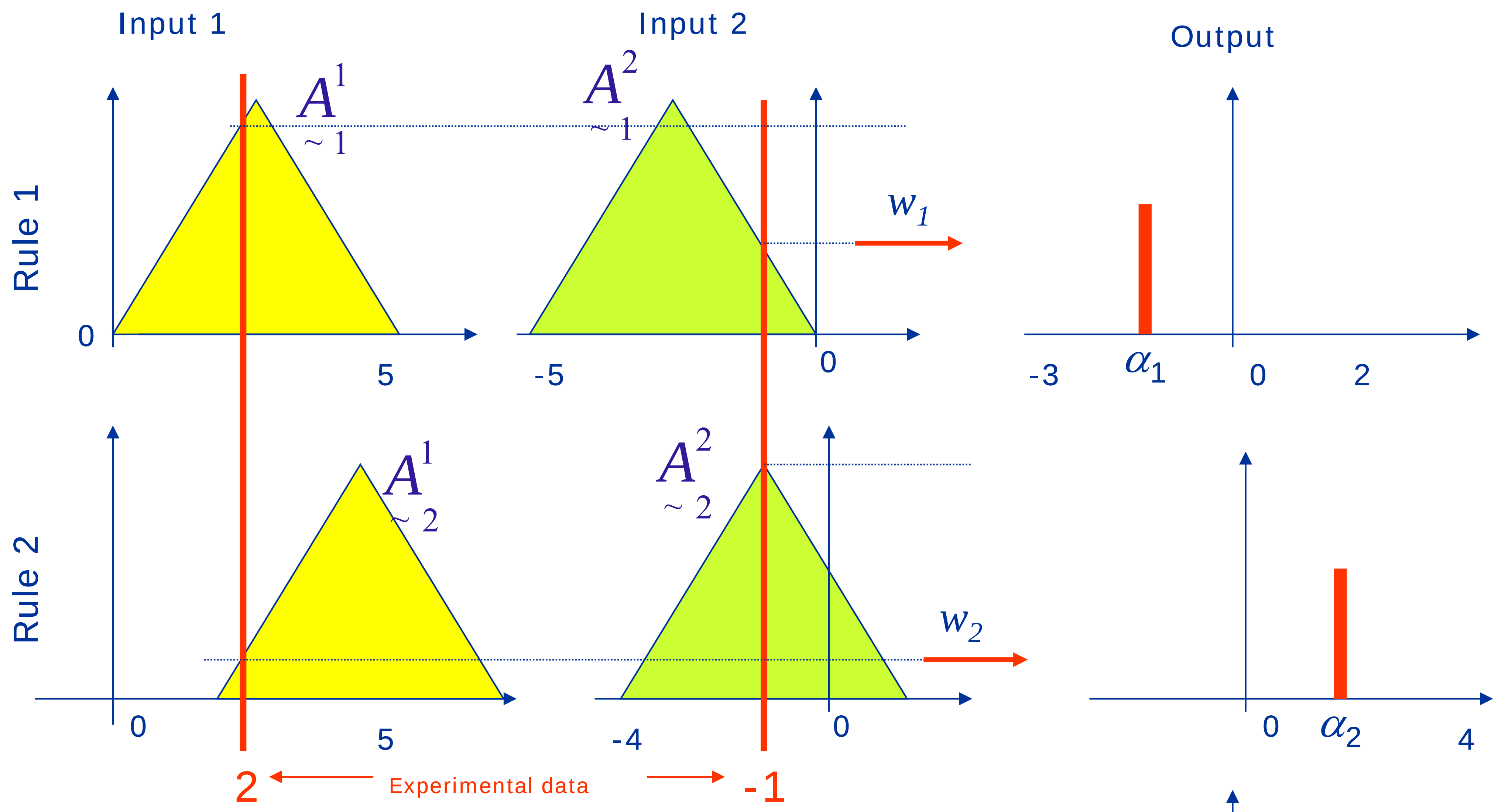
Fuzzy antecedents (as in the Mamdani ones)

Crisp consequents

$$r_1\text{: IF } p_1 \text{ is } \underset{\sim 1}{A}^{1} \text{ AND } p_2 \text{ is } \underset{\sim 1}{A}^{2} \text{ THEN } y_1 = f_1(p_1, p_2)$$

$$r_2\text{: IF } p_1 \text{ is } \underset{\sim 2}{A}^{1} \text{ AND } p_2 \text{ is } \underset{\sim 2}{A}^{2} \text{ THEN } y_2 = f_2(p_1, p_2)$$

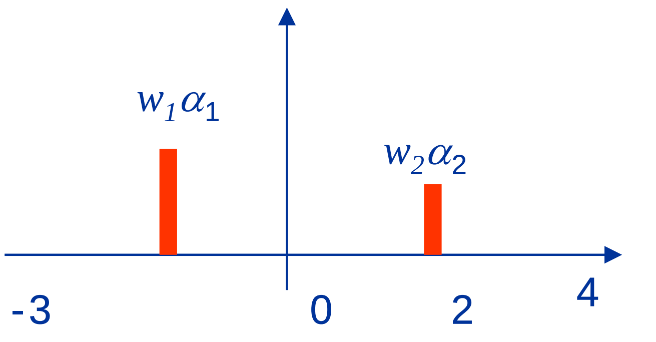
$$f_i(p_1, p_2) = \begin{cases} \alpha_i, constant, \text{TSK order 0.} \\ \alpha_i + \beta_i p_1 + \gamma_i p_2, \text{TSK order 1.} \end{cases}$$



r_1: IF p_1 is $\underset{\sim}{A}{}_1^1$ AND p_2 is $\underset{\sim}{A}{}_1^2$ THEN $y_1 = \alpha_1$

r_2: IF p_1 is $\underset{\sim}{A}{}_2^1$ AND p_2 is $\underset{\sim}{A}{}_2^2$ THEN $y_2 = \alpha_2$

$$y^* = \frac{w_1\alpha_1 + w_2\alpha_2}{w_1 + w_2}$$



$$r_1\text{: IF } p_1 \text{ is } \underset{\sim 1}{A^1} \text{ AND } p_2 \text{ is } \underset{\sim 1}{A^2} \text{ THEN } y_1 = f_1(p_1, p_2)$$

$$r_2\text{: IF } p_1 \text{ is } \underset{\sim 2}{A^1} \text{ AND } p_2 \text{ is } \underset{\sim 2}{A^2} \text{ THEN } y_2 = f_2(p_1, p_2)$$

… … …

$$r_r\text{: IF } p_1 \text{ is } \underset{\sim r}{A^1} \text{ AND } p_2 \text{ is } \underset{\sim r}{A^2} \text{ THEN } y_r = f_r(p_1, p_2)$$

$$y^* = \frac{w_1 f_1^* + w_2 f_2^* + \ldots + w_r f_r^*}{w_1 + w_2 + \ldots + w_r}$$

$$f_i^* = f_i(p_1^*, p_2^*)$$

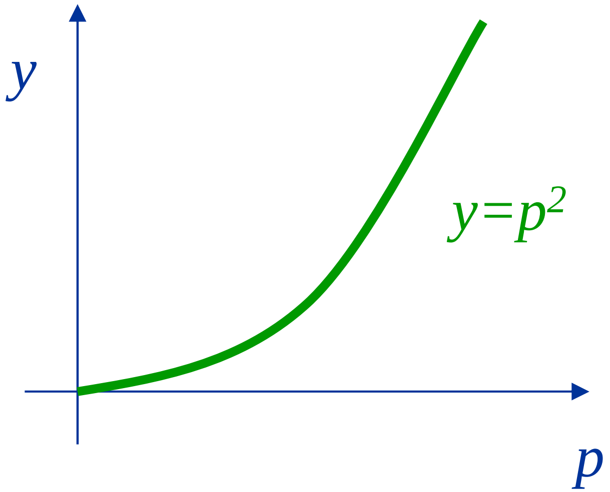
TSK systems are very used

in fuzzy control

for modelling dynamical systems

in neuro-fuzzy systems (Chapter 8)

7.4.3. Modelling by relational fuzzy equations



Each value of p in P is related to a value of y in Y.

$P=\{-2,-1,0,1,2\}$ $\qquad$ $y=\{y: y=p^2, p\in P\}=\{4,1, 0,1,4\}=\{0,1,4\}$

$$R(p,y)=\begin{array}{c|ccc|} & 0 & 1 & 4 \\ \hline -2 & 0 & 0 & 1 \\ -1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 2 & 0 & 0 & 1 \\ \hline \end{array}$$

Transfer function between p and y

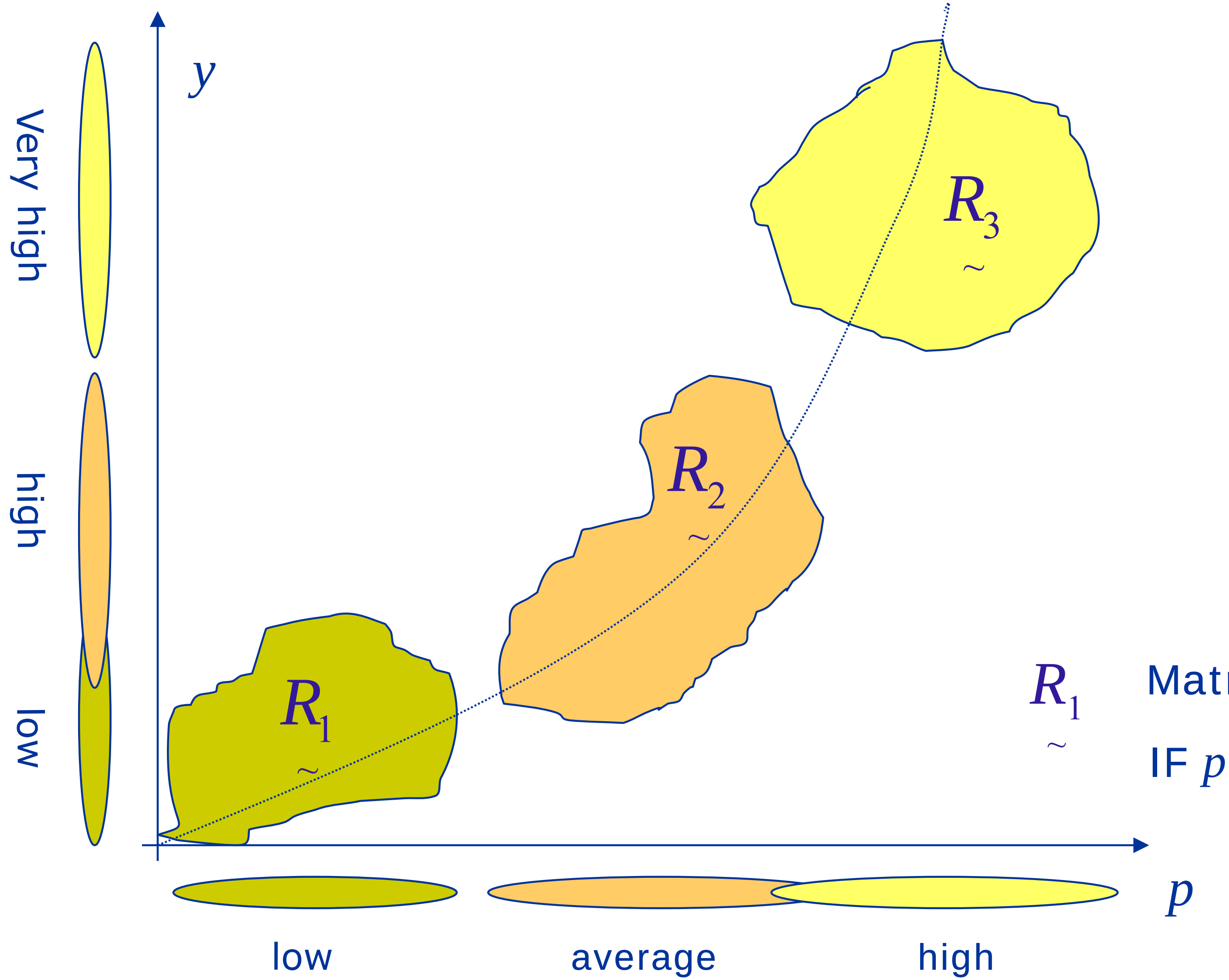
To compute the output corresponding to one input $p^* = -1$

$$p^* = -1 = \left\{ \frac{0}{-2} + \frac{1}{-1} + \frac{0}{0} + \frac{0}{1} + \frac{0}{2} \right\}$$

$$y^* = p^* \circ R = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \end{bmatrix} \circ \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$$

$$= \left\{ \frac{0}{0} + \frac{1}{1} + \frac{0}{4} \right\} = 1$$

With imprecise knowledge



$\underset{\sim}{R}_1$ Matrix of implication

IF p is low THEN y is low

Given one input, p^*

(i) compute, by max-min composition (or max-prod) the outputs resulting from each input $\underset{\sim}{R_i}$

$$y^1 = p^* \circ \underset{\sim}{R}^1 \qquad y^2 = p^* \circ \underset{\sim}{R}^2 \qquad y^3 = p^* \circ \underset{\sim}{R}^3$$

(ii) the total output is, in the case of disjunctive rules,

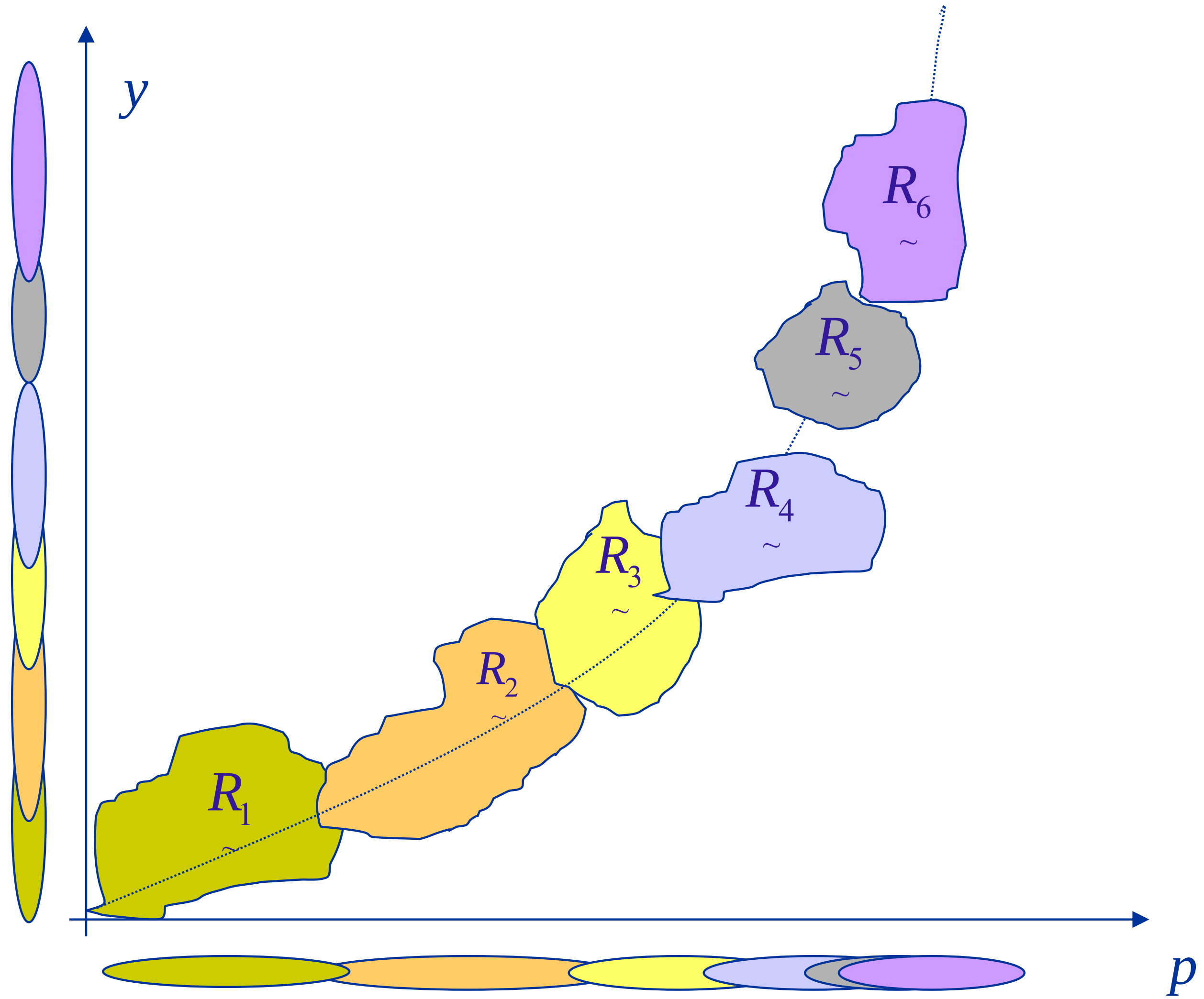
$$y = y^1 \vee y^2 \vee y^3 = (p^* \circ \underset{\sim}{R_1}) \vee (p^* \circ \underset{\sim}{R_2}) \vee (p^* \circ \underset{\sim}{R_3}) =$$

$$= p^* \circ (\underset{\sim}{R_1} \vee \underset{\sim}{R_2} \vee \underset{\sim}{R_3}) = p^* \circ (\underset{\sim}{R_1} \cup \underset{\sim}{R_2} \cup \underset{\sim}{R_3})$$

$$= p^* \underset{\sim}{R}$$

$$\underset{\sim}{R} = \underset{\sim}{R_1} \cup \underset{\sim}{R_2} \cup \underset{\sim}{R_3}$$

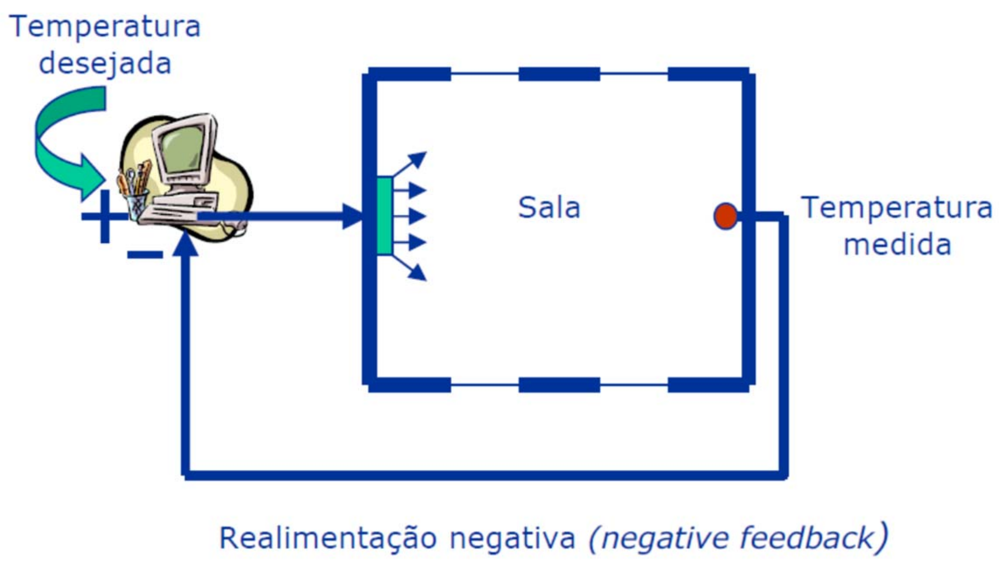
Fuzzy transfer function



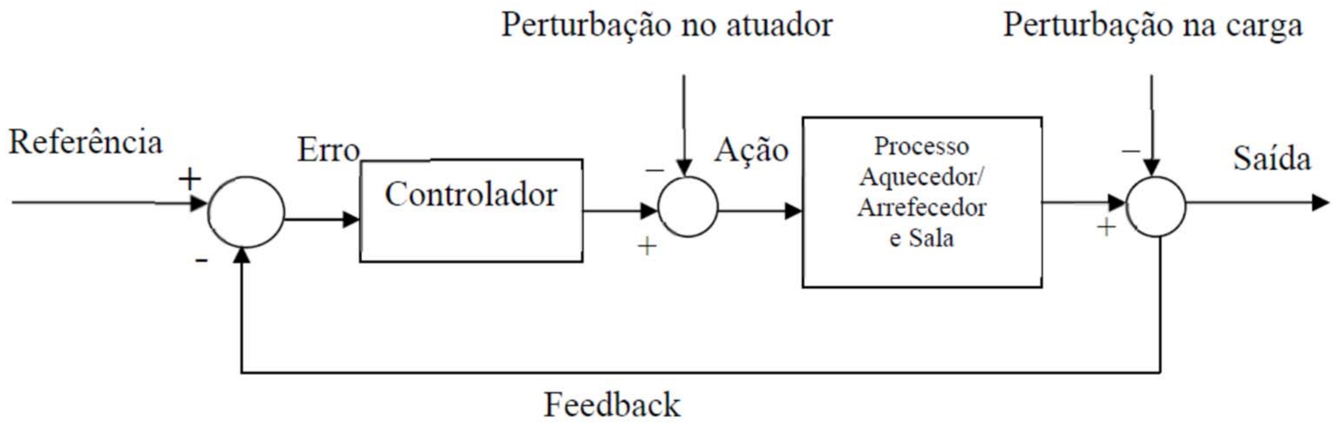
The higher the number of relations, the more precise the transfer functions will be.

Feedback control systems

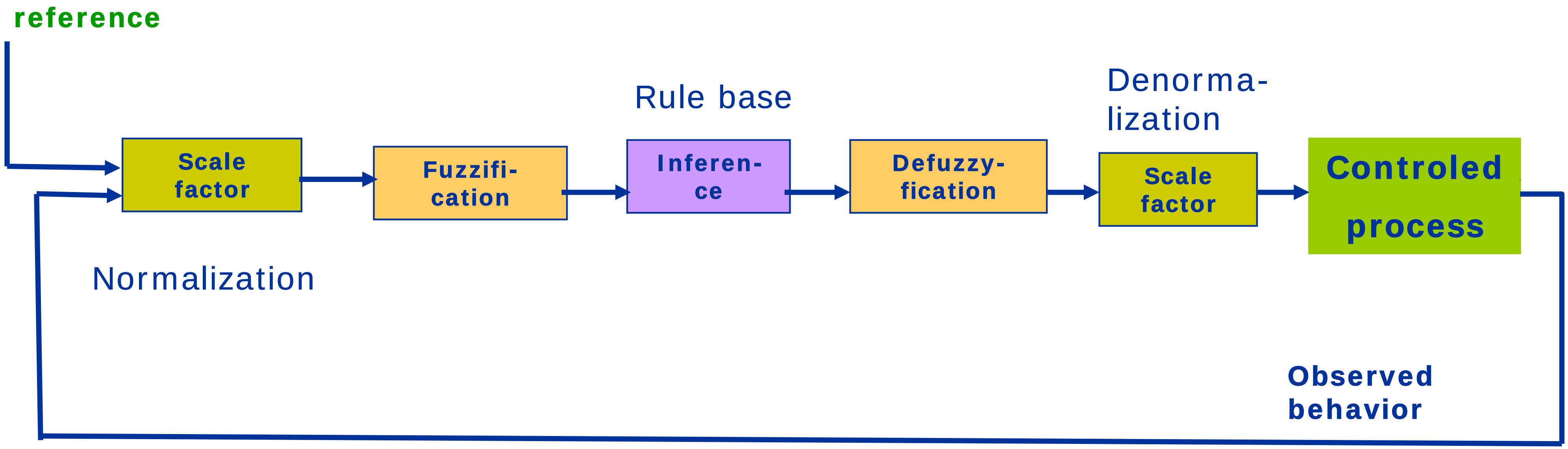
Read the note "Noções básicas de controlo" or "Introduction to Automatic Control" in the course materials (PL)



Room air conditioning



7.5 Introduction to fuzzy control



Reference: the desired behavior of the process (reference behavior)

Error: difference between the reference and the observed behavior (the process output)

Stages of the development of a basic fuzzy controller

1- Identification of the process variables (inputs, outputs, disturbances, etc.) and of the controller input variables (error, variation of error,...).

2- Partition of the universe of discourse (scales of the inputs and outputs of the controller) in a certain number (3, 5, 7, ...) of fuzzy sets, giving to each one a linguistic label; these fuzzy sets must cover all the universe of discourse with some level of superposition.

3- Affect to each fuzzy set a membership function.

4- Write the rules: define a fuzzy relation between the controller input fuzzy sets (antecedents) and the controller output fuzzy sets (consequents)

5- Chose appropriate scale factors for the controller inputs obliging them to remain in the interval [-1,1]. At the same time chose appropriate scale factors for the controller outputs in order to put them in accordance with the scale of the process input. This is a trial and error procedure.

6- Fuzzify the controller inputs (usually the error and the variation of the error).

7- Fire the rules and obtain the output fuzzy set from each one.

8- Aggregate the fuzzy outputs produced by each rule

9- Apply a method of defuzzification to compute a crisp output.

Finding appropriate scale factors is frequently a challenge.

How to obtain the fuzzy rules

- From the knowledge of an expert.

- From experimental data conveniently processed (ex. by clustering).

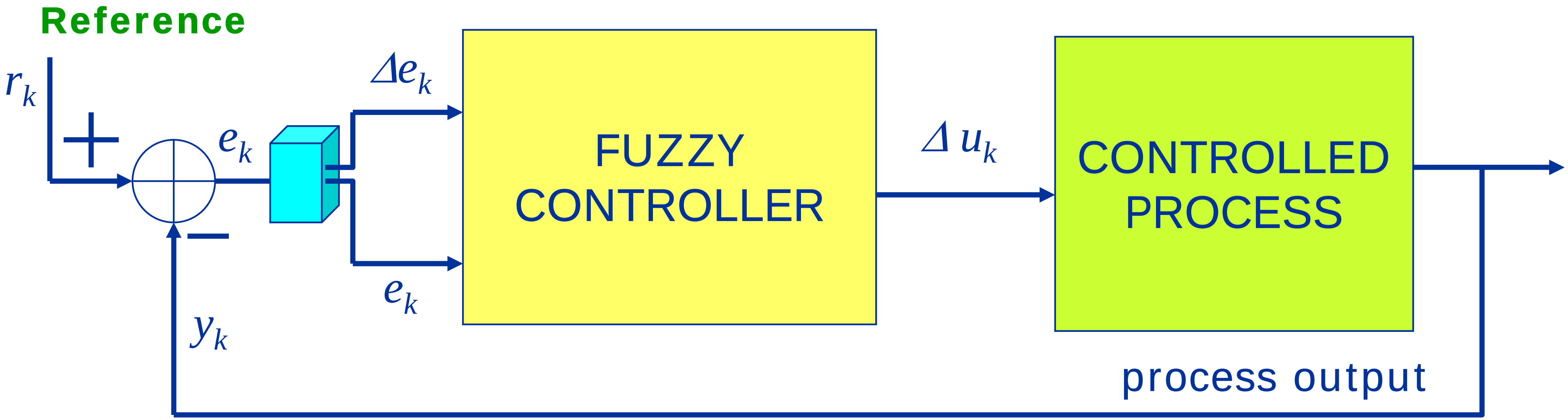
- From simulations with a mathematical model of the process (seldom available...).

The controller may be adjusted afterwards, either manually, or automatically (adaptive control).

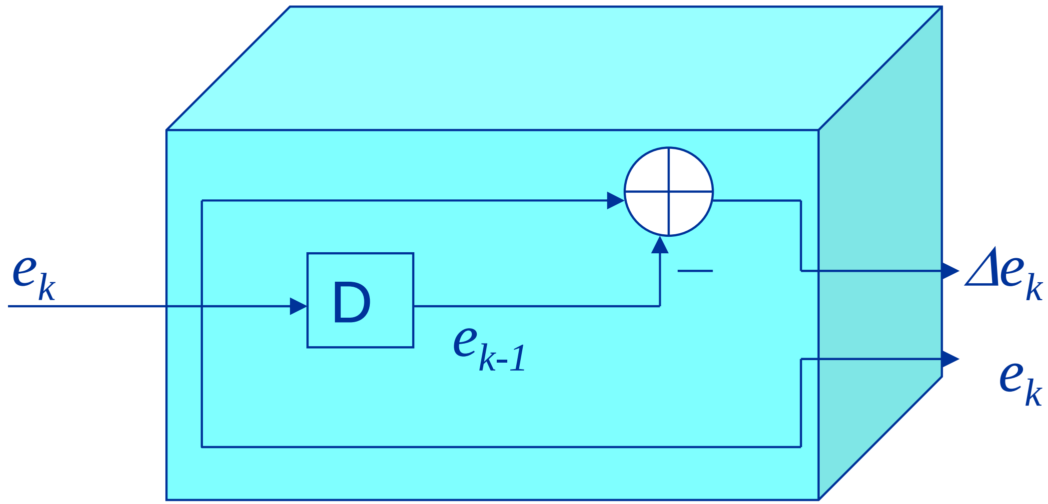
Degrees of freedom: scale factors, membership functions (shape, placement) of the antecedents and consequents.

Fuzzy controller with two inputs and one output

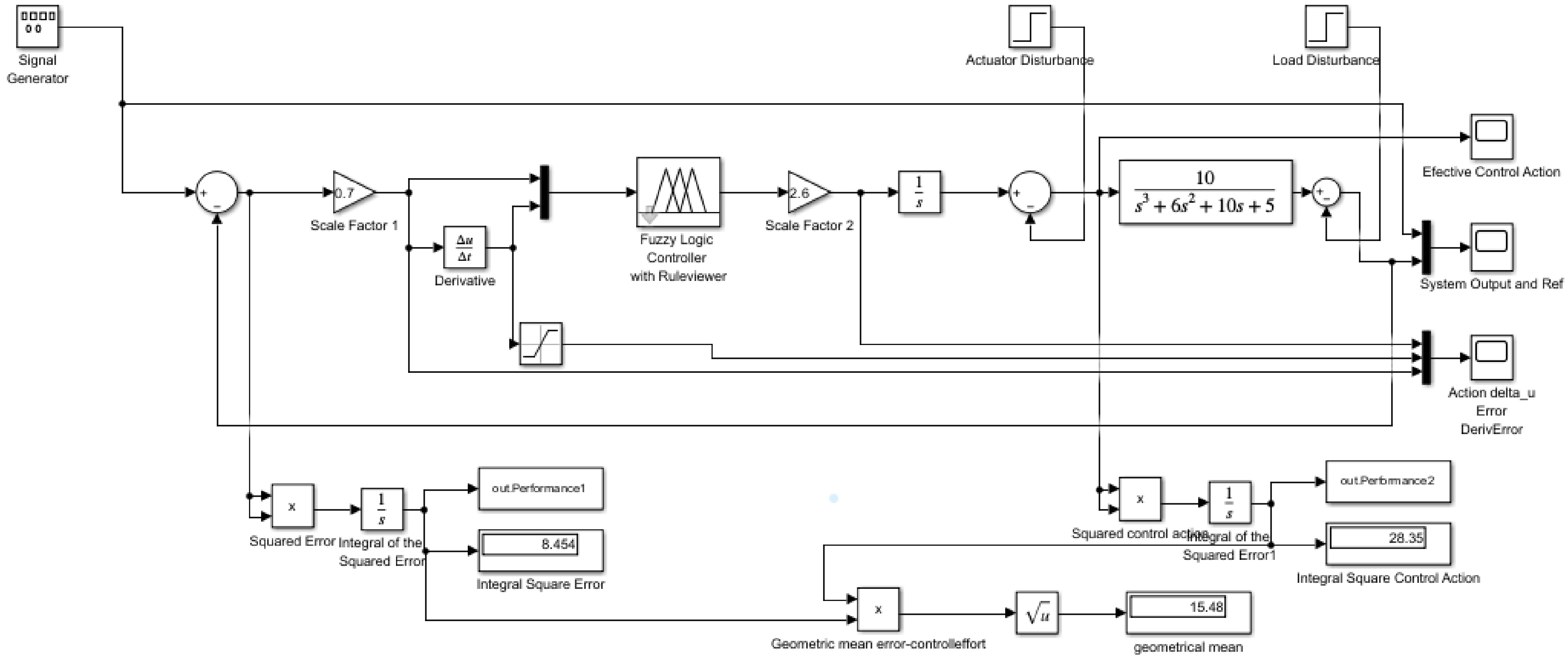
$$r_1\text{: IF } e_k \text{ is } \underset{\sim}{A}{}_1^1 \text{ AND } \Delta e_k \text{ is } \underset{\sim}{A}{}_1^2 \text{ THEN } \Delta u_k \text{ is } \underset{\sim}{B}{}_1$$



$$\Delta e_k = e_k - e_{k-1}$$



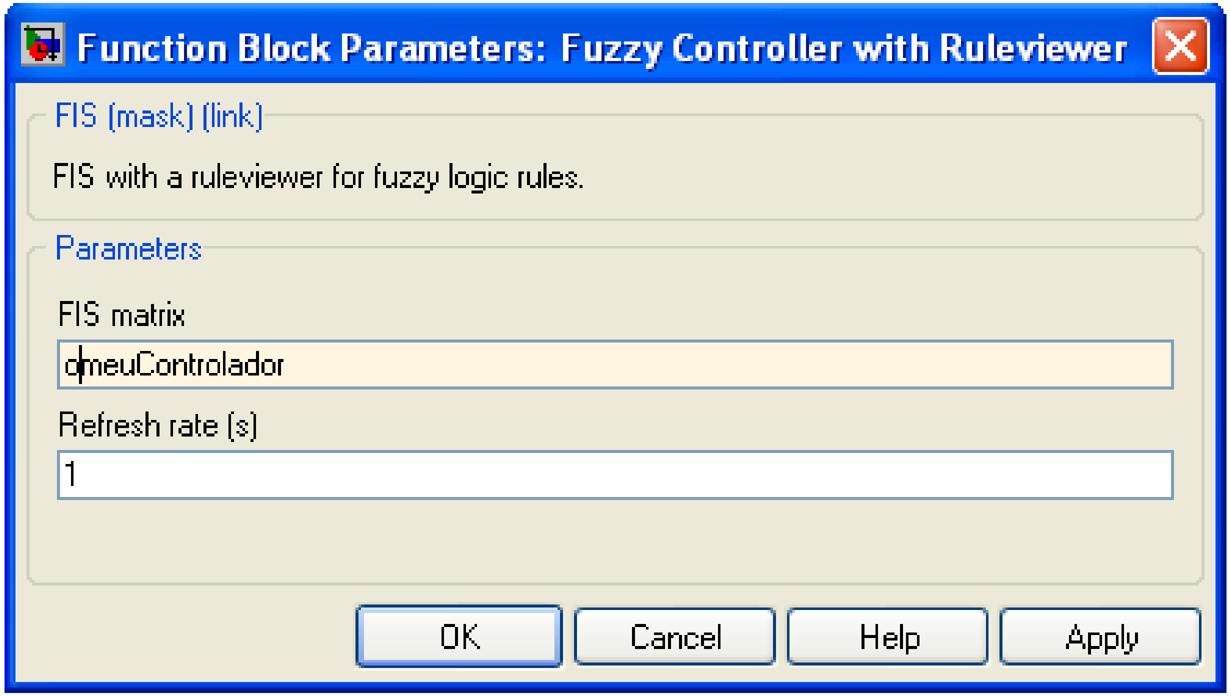
Implementation in Simulink for a continuous process



The block Fuzzy Controller with Ruleviewer, from the Simulink library (Fuzzy Control Toolbox). When working, we can see the rules being fired, with animation. The frequency of the animation is given by the parameter "Refresh Rate" in the controller block; its value specifies the number of samples between two consecutive animations.

Note that the saturation of the derivative of the error before the scope 3, prevents that in the graph the derivative, when it is very high, "smaches" the error.

Clicking twice in the controller block the following dialog window appears, were we write the name of the controller, a fis matrix, in the case omeuControlador.fis.



omeuControlador is a *.fis* file, developed with the fiseditor (>fuzzy or fuzzyLogicDesigner) and exported to the *workspace* (it is not enough to have it in the working directory).

The need of the derivative of the error (error variation)

The action over the controlled process at an instant *k* depends not only on the error in that instant *k* but also on the way the error is varying: is it becoming lower or higher (with respect to previous instant *k-1*) ?

The derivative of the error is a good measure of that evolution. That is why the error and its derivative are the two inputs of the controller. In discrete time the derivative is approximated by $\Delta e(k)=e(k)-e(k-1)$.

Introduction of an integrator

For a good performance, the controller must calculate not the amplitude of the control action, but its **variation** from an instant to the next. This is called **incremental control** mode.

To compute the amplitude of the control from the incremental control, an integrator is introduced: the applied control action in each instant is the integral (the sum) of all increments since the beginning.

Controller type Mamdani:

$$r_1\text{: IF } e_k \text{ is } \underset{\sim}{A}{}_1^1 \text{ AND } \Delta e_k \text{ is } \underset{\sim}{A}{}_1^2 \text{ THEN } \Delta u_k \text{ is } \underset{\sim}{B}{}_1$$

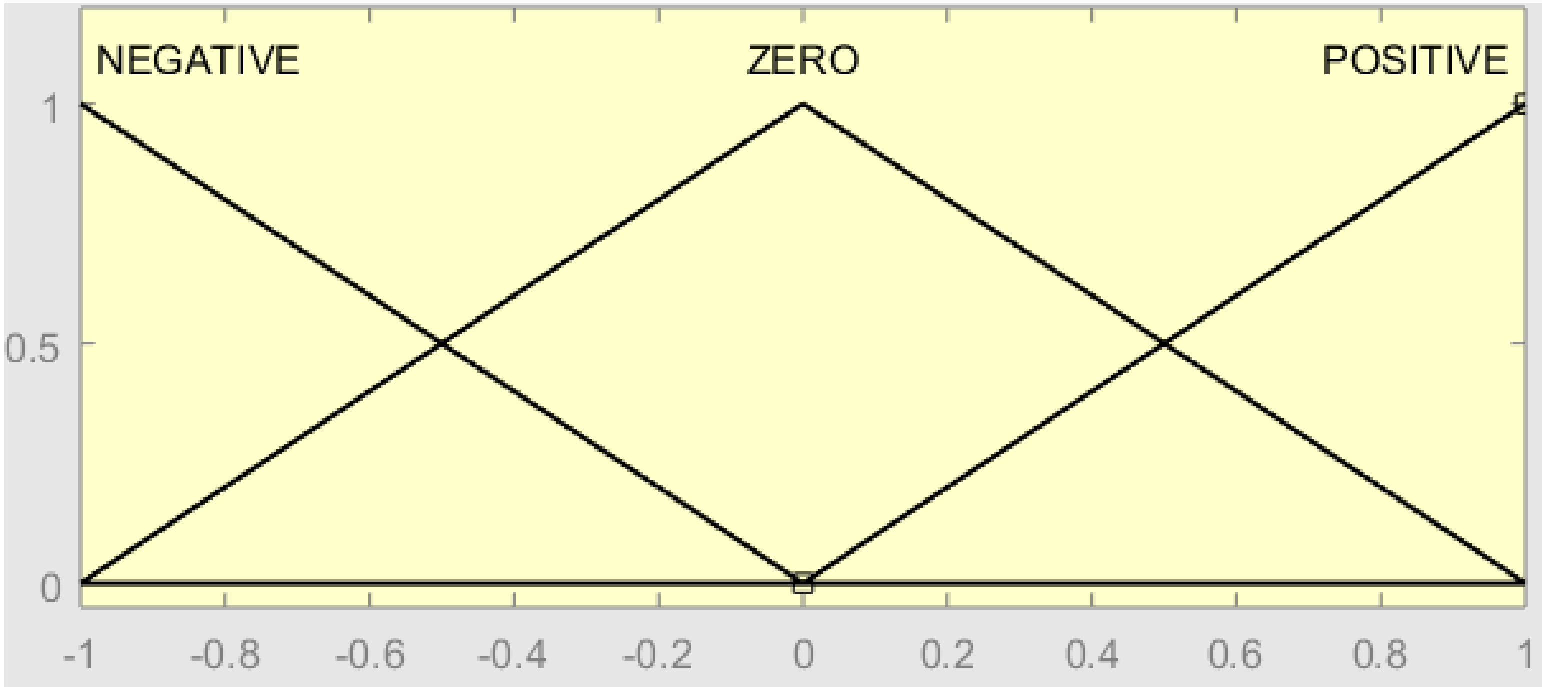
Controller type Sugeno:

$$r_1\text{: IF } e_k \text{ is } \underset{\sim}{A}{}_1^1 \text{ AND } \Delta e_k \text{ is } \underset{\sim}{A}{}_1^2 \text{ THEN } \Delta u_k = f_1(e_k, \Delta e_k)$$

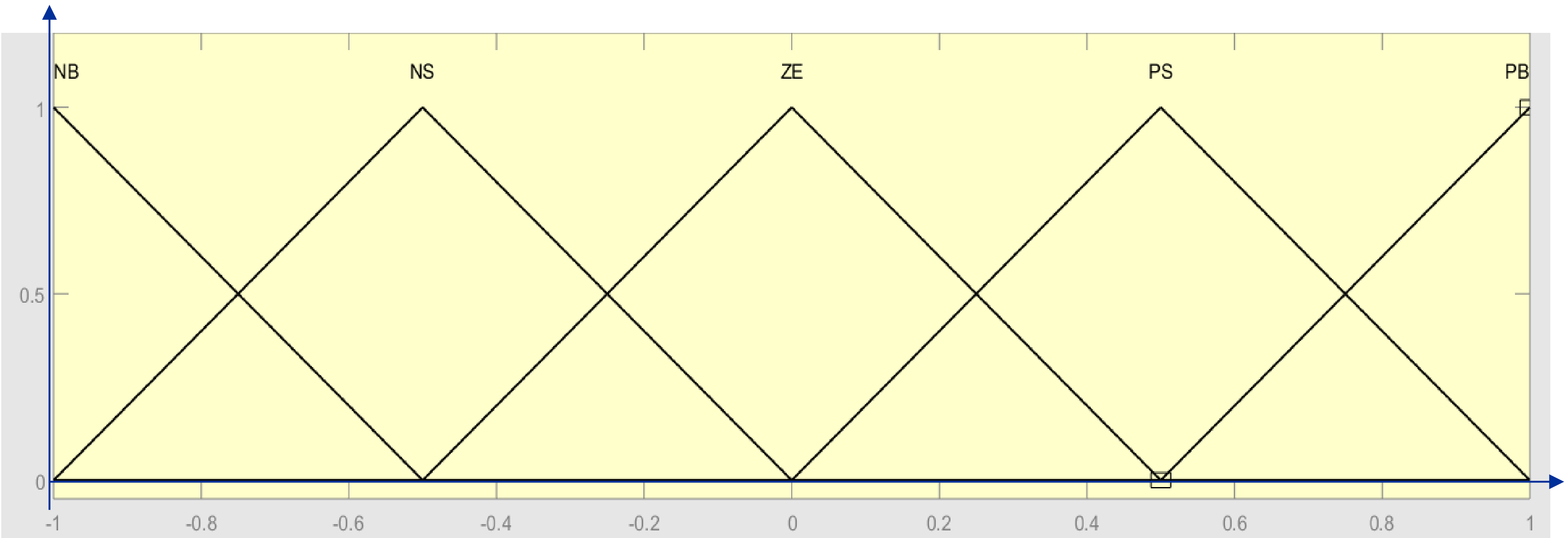
Choosing the linguistic labels

After normalization of e_k and Δe_k to the scale $[-1,1]$, a certain number of intervals in that scale is defined and linguistic labels are associated to them:

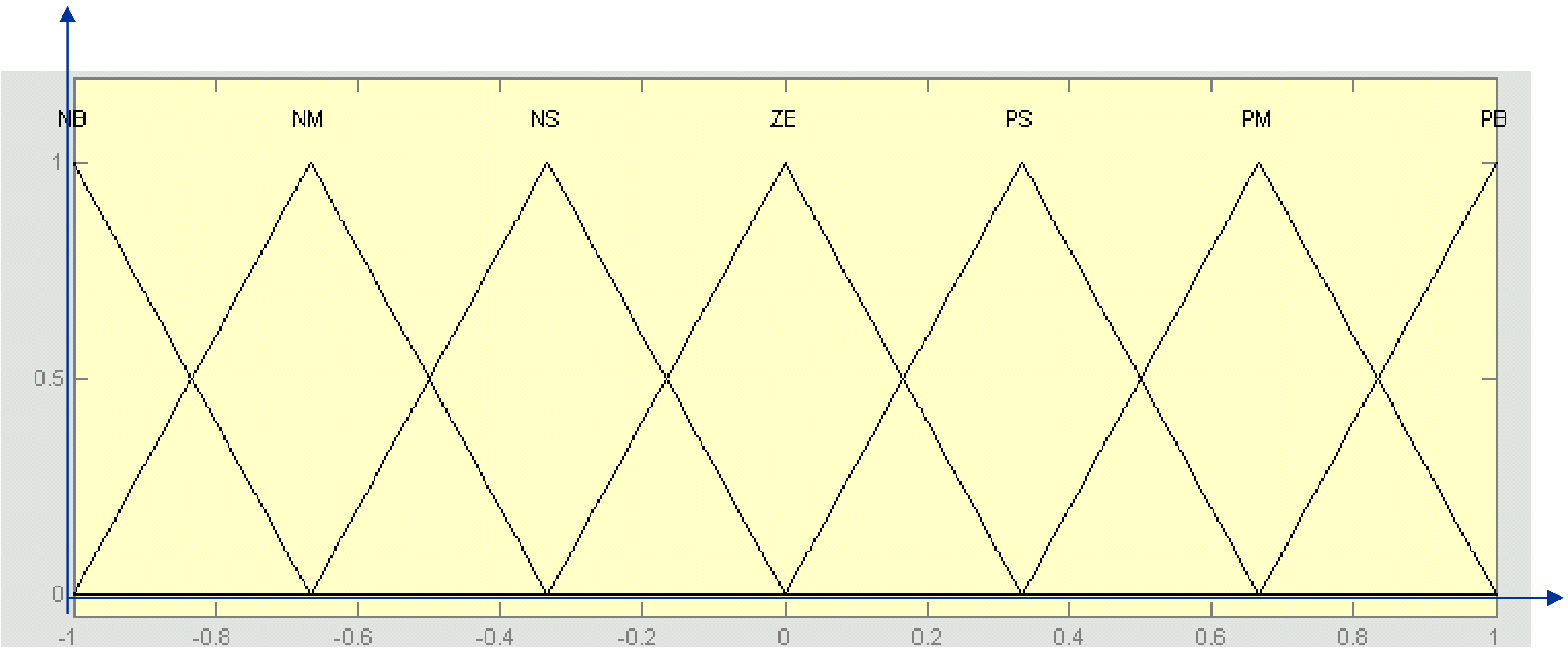
3 terms :{negative (N), zero(Z), positive(P)}



5 terms :{negative big(NB), negative small (NS), zero(ZE), positive small (PS), positive big (PB)}

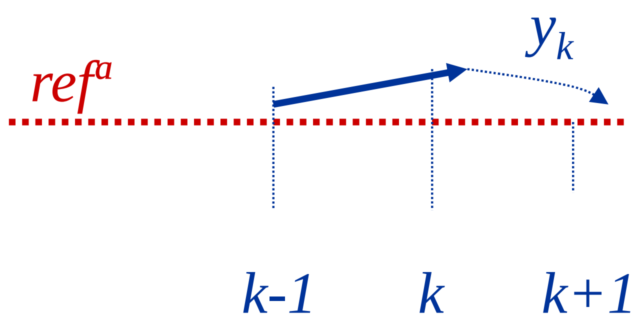


7 terms :{negative big(NB), negative medium (NM), negative small (NS), zero(ZE), positive small (PS), positive medium (PM), positive big (PB)}



Suppose the case where e_k e Δe_k are small: the output deviated from the reference, but it is near and is evolving slowly; as a consequence the control variation Δu_k should be small to correct that deviation,

$$\left.\begin{cases} e_k \ \text{NS} \Rightarrow y_k > \text{reference} \\ \Delta e_k \ \text{NS} \Rightarrow e_k < e_{k-1} \end{cases}\right\} \Rightarrow \text{the output is going away from the reference}$$

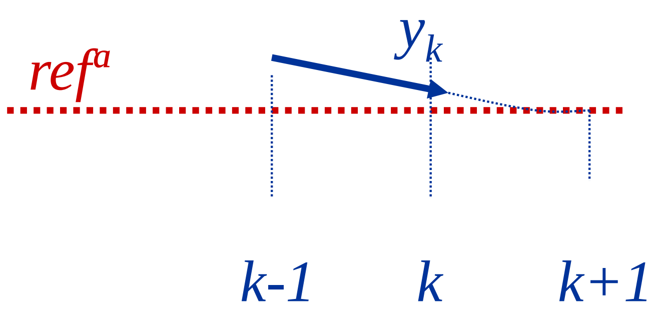


$$e_k = ref_k - y_k \text{, negative}$$
$$\Delta e_k = e_k - e_{k-1}\text{, negative}$$

One needs to reduce the output and make it come back to the reference. For that the control action must be smaller, diminishing u_k significantly, by making Δu_k NM (in the hypothesis that lowering the process input will lower the process output). This leads to the rule

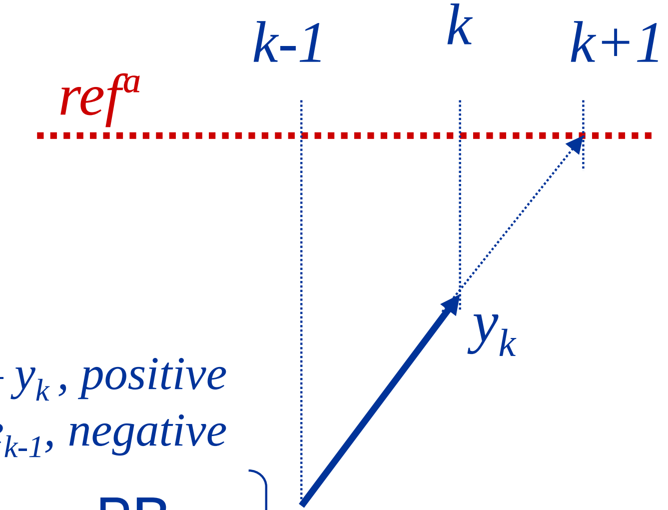
IF e_k is NS and Δe_k is NS THEN Δu_k is NM

Other situations:



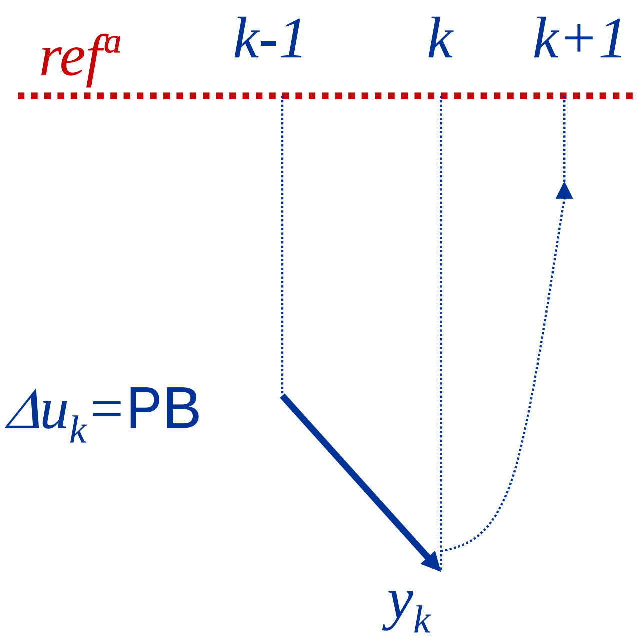
e_k NS
Δe_k *PS*
$\Big\}\ \Delta u_k$=*ZE*

$e_k = ref_k - y_k$,*negative*
$\Delta e_k = e_k$-e_{k-1}, *positive*



$e_k = ref_k - y_k$, *positive*
$\Delta e_k = e_k$-e_{k-1}, *negative*

e_k PB
Δe_k *NB*
$\Big\}\ \Delta u_k$=ZE



$e_k = ref_k - y_k$, *positive*
$\Delta e_k = e_k$-e_{k-1}, *positive*

e_k PB
Δe_k *PB*
$\Big\}\ \Delta u_k$=PB

For all cases, RULE BASE (Dryankov, 113)

FAM

Fuzzy

Associative

Memory

49 rules

e_k \ Δe_k	NB	NM	NS	ZE	PS	PM	PB
NB	NB	NB	NB	NB	NM	NS	ZE
NM	NB	NB	NB	NM	NS	ZE	PS
NS	NB	NB	NM	NS	ZE	PS	PM
ZE	NB	NM	NS	ZE	PS	PM	PB
PS	NM	NS	ZE	PS	PM	PB	PB
PM	NS	ZE	PS	PM	PB	PB	PB
PB	ZE	PS	PM	PB	PB	PB	PB

e_k \ Δe_k	NB	NS	ZE	PS	PB
NB	NB	NB	NB	NS	ZE
NS	NB	NB	NS	ZE	PS
ZE	NB	NS	ZE	PS	PB
PS	NS	ZE	PS	PB	PB
PB	ZE	PS	PB	PB	PB

25 rules

e_k \ Δe_k	N	ZE	P
N	N	N	Z
ZE	N	Z	P
P	Z	P	P

9 rules

e_k \ Δe_k	N	ZE	P
N	NB	N	Z
ZE	N	Z	P
P	Z	P	PB

If the antecedents have three values (N ZE P) and the consequents five (NB N Z P PB) the ex-students André Fonseca and Fábio Mestre) proposed this table, with good results. For the Sugeno order zero case the consequents are (-1, -0.5, 0, 0.5, 1).

Fuzzy controller database:

- Membership functions: type, parameters
- Scale factors: numerical values

Membership functions most frequent: triangular, Gaussians, both with superposition at 0.5.

Scale factors:

- Heuristic choice, by trial and error, after estimation of the maximum and minimum values of e_k, Δe_k and of u_k.
- More elaborated methods (Dryankov, 127); self organizing controllers.
- some patents on it, ex. https://www.google.com/patents/US5687076

Bibliography

Fuzzy Logic With Engineering Applications, Timothy Ross, *4th Ed.,* Wiley, 2016.

Fuzzy Logic Toolbox Users' Guide, The Mathworks, 2023b.

Fuzzy Logic, H. Zimmermann, Springer Verlag, 1997.

An Introduction to Fuzzy Sets Analysis and Design, Pedrycz, Witold and Fernando Gomide,, MIT Press, 1998.

An Introduction to Fuzzy Control, D. Driankov, H. Hellendoorn and M. Reinfrank, Springer Verlag 1996.

Fuzzy Modelling and Control, Andrzej Piegat, Springer Verlag, 2001.

*Introduction to Type-2 Fuzzy Logic Control: Theory and Applications*Mendel, Jerry M., Hani Hagras, Woei-Wan Tan, William W. Melek, and Hao Ying., IEEE Press, John Wiley & Sons, 2014.