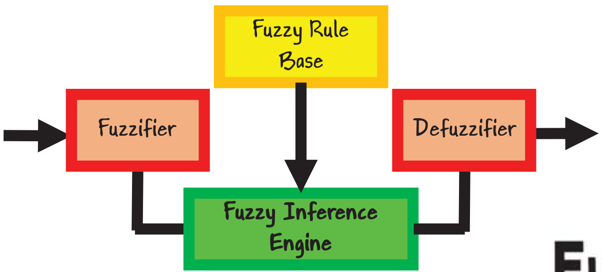
Intelligent Control



Fuzzy Rules and Fuzzy Inference Systems

By:
Barmak Baigzadehnoe
b.baigzadeh@uok.ac.ir

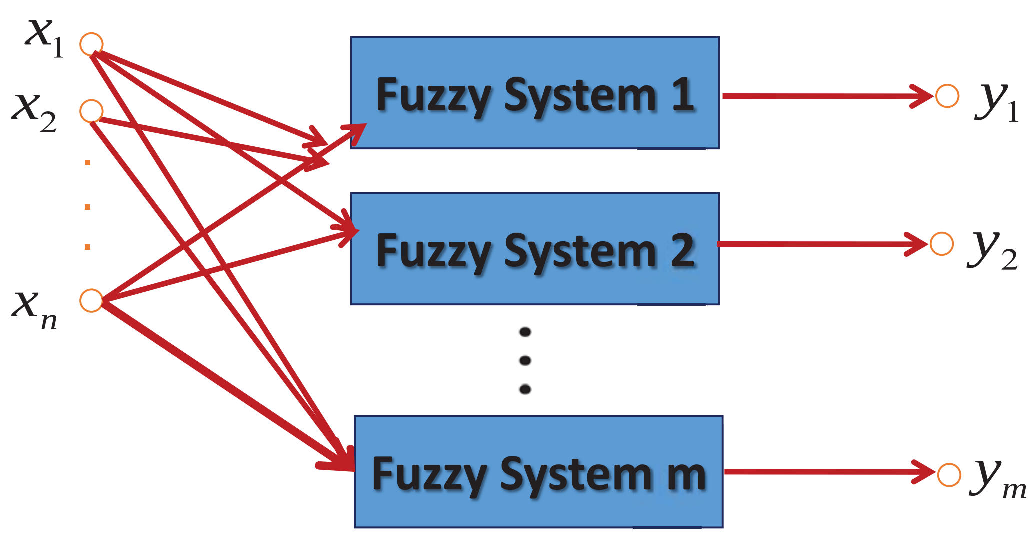
Smart/Micro Grids Research Center, University of Kurdistan

Content

- Fuzzy Systems
- Fuzzy Rules
- Fuzzy Inference Engine

Fuzzy Systems

We consider only the multi-input-single-output case, because a multi-output system can always be decomposed into a collection of single-output systems:



Content

- Fuzzy Systems
- Fuzzy Rules
- Fuzzy Inference Engine

Fuzzy Rule Base

A ***fuzzy rule base*** is a set of fuzzy IF-THEN rules. It is the heart of the fuzzy system in the sense that all other components are used to implement these rules in a reasonable and efficient manner. The ***fuzzy rule base*** comprises the following fuzzy IF-THEN rules:

Fuzzy sets

$$Ru^{(l)}: \text{IF } x_1 \text{ is } A_1^l \text{ and } \cdots \text{ and } x_n \text{ is } A_n^l \text{ THEN } y \text{ is } B^l$$

$$x_1, \cdots, x_n \in U = U_1 \times \cdots \times U_n$$

$$y \in V$$

Note that this form of fuzzy IF-THEN rule is known as *canonical* or *Mamdani* fuzzy rules.

Fuzzy Rule Base

The canonical fuzzy IF-THEN rules include the following as special cases:

- ***Partial Rules***

$$\text{IF } x_1 \text{ is } A_1^l \text{ and } \cdots \text{ and } x_m \text{ is } A_m^l \text{ THEN } y \text{ is } B^l \ (m < n)$$

- ***OR Rules***

$$\text{IF } x_1 \text{ is } A_1^l \text{ and } \cdots \text{ and } x_m \text{ is } A_m^l \text{ OR } x_{m+1} \text{ is } A_{m+1}^l \text{ and } \cdots \text{ and } x_n \text{ is } A_n^l \text{ THEN } y \text{ is } B^l$$

- ***Single Fuzzy Statements***

$$y \text{ is } B^l$$

Fuzzy Rule Base

- ***Gradual Rules***

  The smaller the x, the bigger the y

- ***Non-Fuzzy Rules***

  If the membership functions of A_i^l and B_l can only take values 1 or 0, then the rules become non-fuzzy rules,

Fuzzy Rule Base

A set of fuzzy IF-THEN rules is ***complete*** if for any linguistic variables in the input space, there exists at least one rule in the fuzzy rule base, such that:

$$\forall i: \ \mu_{A_i^l}(x_i) \neq 0$$

Note: The completeness of a set of rules means that at any point in the input space there is at least one rule that "fires"; that is, the membership value of the IF part of the rule at this point is non-zero.

Fuzzy Rules

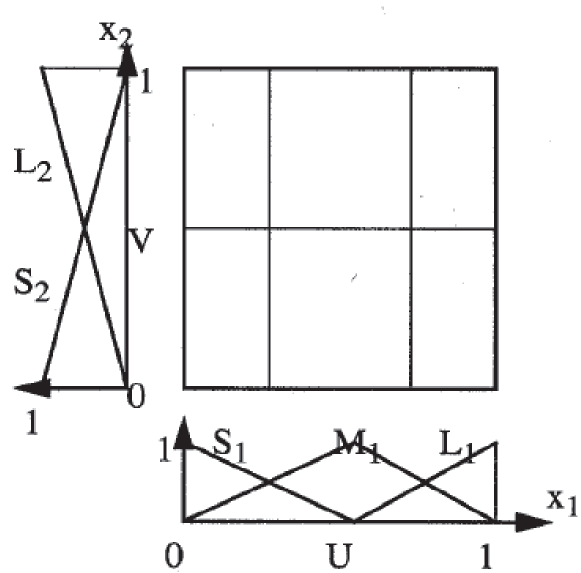
Example: Consider a two-input-one-output fuzzy system with $U = U_1 \times U_2 = [\,0,1\,] \times [\,0,1\,]$ and $V = [\,0,1\,]$. Define three fuzzy sets S_1, M_1 and L_1 in U_1, and two fuzzy sets S_2 and L_2 in U_2. In order for a fuzzy rule base to be complete, it must contain the following six rules whose IF parts constitute all the possible combinations of S_1, M_1 and L_1 with S_2 and L_2:

$$IF\ x_1\ is\ S_1\ and\ x_2\ is\ S_2, THEN\ y\ is\ B^1$$
$$IF\ x_1\ is\ S_1\ and\ x_2\ is\ L_2, THEN\ y\ is\ B^2$$
$$IF\ x_1\ is\ M_1\ and\ x_2\ is\ S_2, THEN\ y\ is\ B^3$$
$$IF\ x_1\ is\ M_1\ and\ x_2\ is\ L_2, THEN\ y\ is\ B^4$$
$$IF\ x_1\ is\ L_1\ and\ x_2\ is\ S_2, THEN\ y\ is\ B^5$$
$$IF\ x_1\ is\ L_1\ and\ x_2\ is\ L_2, THEN\ y\ is\ B^6$$

in which $B^l\ (l = 1,2,\dots,6)$ are fuzzy sets in V.

Fuzzy Rules

Example:



Content

- Fuzzy Systems
- Fuzzy Rules
- Fuzzy Inference Engine

Fuzzy Inference Engine

First consider Let $Ru^{(l)}$ be a fuzzy relation in $U \times V$, which represent the fuzzy IF-THEN rule:

$$R_u^{(l)}: \text{IF } x_1 \text{ is } A_1^l \text{ and} \cdots \text{and } x_n \text{ is } A_n^l \text{ THEN } y \text{ is } B^l$$

$$Ru^{(l)} = A_1^l \times A_2^l \times A_3^l \times \ldots \times A_n^l \rightarrow B^l$$

where $A_1^l \times A_2^l \times A_3^l \times \ldots \times A_n^l$ is a fuzzy relation in $U = U_1 \times U_2 \times \ldots \times U_n$ defined by

$$\mu_{A_1^l \times A_2^l \times A_3^l \times \ldots \times A_n^l}(x_1, x_2, \ldots, x_n) = \mu_{A_1^l} * \mu_{A_2^l} * \mu_{A_3^l} * \ldots * \mu_{A_n^l}$$

Fuzzy Inference Engine

$$\mu_{A_1^l \times A_2^l \times A_3^l \times \dots \times A_n^l}(x_1, x_2, \dots, x_n) = \mu_{A_1^l} * \mu_{A_2^l} * \mu_{A_3^l} * \dots * \mu_{A_n^l}$$

where $*$ is represents any t-norm operator. The implication $\rightarrow$ in $Ru^{(l)}$ is defined according to various implications such as Dienes-Rescher Implication, Lukasiewicz Implication,….

Example: Determine $\mu_{A_1^l \times A_2^l \times A_3^l \times \dots \times A_n^l \rightarrow B^l}(x_1, x_2, \dots, x_n, y)$ by employing product t-norm and Dienes-Rescher Implication.

Fuzzy Inference Engine

In a fuzzy inference engine, there are two opposite arguments for what a set of rules should mean:

- Views the rules as independent conditional statements:

  " Union Combination "

- Views the rules as strongly coupled conditional statements:

  "Intersection Combination"

Fuzzy Inference Engine

In a fuzzy inference engine, fuzzy logic principles are used to combine the fuzzy IFTHEN rules in the fuzzy rule base into a mapping from a fuzzy set in input space to a fuzzy set in output space. There are two ways to infer with a set of rules:

- ***Composition based inference***
- ***Individual-rule based inference***

Note that If the fuzzy rule base consists of only a single rule, then the *generalized modus ponens* specifies the mapping from fuzzy set in input space to a fuzzy set in output space.

Union Composite Based Inference

If we accept the rules as independent conditional statements, then the M rules in the canonical form are interpreted as a single fuzzy relation Q_M in $U \times V$ defined by:

$$Q_M = \bigcup_{l=1}^{M} R_u^{(l)}$$

This combination is called the ***Mamdani combination***. If we use the symbol + to represent the s-norm, then

$$\mu_{Q_M}(x_1, x_2, \ldots, x_n, y) = \mu_{R_u^{(1)}}(x, y) + \mu_{R_u^{(2)}}(x, y) + \ldots + \mu_{R_u^{(M)}}(x, y)$$

Intersection Composite Based Inference

If we accept the rules as strongly coupled conditional statements, then the M rules in the canonical form are interpreted as a single fuzzy relation Q_G in $U \times V$ defined by:

$$Q_G = \bigcap_{l=1}^{M} R_u^{(l)}$$

This combination is called the ***Godal combination***. If we use the symbol $*$ to represent the t-norm, then

$$\mu_{Q_G}\left(x_1, x_2, \ldots, x_n, y\right) = \mu_{R_u^{(1)}}(x, y) * \mu_{R_u^{(2)}}(x, y) * \ldots * \mu_{R_u^{(M)}}(x, y)$$

Composite Based Inference

Finally, let A' be an arbitrary fuzzy set in U and be the input to the fuzzy inference engine. Then, by viewing Q_M or Q_G as a single fuzzy IF-THEN rule and using the ***generalized modus ponens***, we obtain the output of the ***fuzzy inference engine*** as

Mamdani Combination

$$\mu_{B'}(y) = \sup_{x \in U} t[\mu_{A'}(x), \mu_{Q_M}(x, y)]$$

Godel Combination

$$\mu_{B'}(y) = \sup_{x \in U} t[\mu_{A'}(x), \mu_{Q_G}(x, y)]$$

$$\text{generalized modus ponens}: \mu_{B'}(y) = \sup_{x \in U} t[\mu_{A'}(x), \mu_{A \to B}(x, y)]$$

Individual Rule Based Inference

In ***individual-rule based inference***, each rule in the fuzzy rule base determines an output fuzzy set and the output of the whole fuzzy inference engine is the combination of the M individual fuzzy sets. For given input fuzzy set A' in U, compute the output fuzzy set B' in V for each individual rule $Ru^{(l)}$ according to the ***generalized modus ponens***:

$$\mu_{B'_l}(y) = \sup_{x \in U} t[\mu_{A'}(x), \mu_{R_u^{(l)}}(x, y)] \qquad ; \; l = 1, 2, \ldots, M$$

$$generalized\ modus\ ponens: \mu_{B'}(y) = \sup_{x \in U} t[\mu_{A'}(x), \mu_{A \to B}(x, y)]$$

Individual Rule Based Inference

The output of the fuzzy inference engine is the combination of the M fuzzy sets $\{B'_1, B'_2, \ldots, B'_M\}$ according to two opposite arguments:

- Union Individual Rule Based Inference

$$\mu_{B'}(y) = \mu_{B'_1}(y) + \mu_{B'_2}(y) + \ldots. + \mu_{B'_M}(y)$$

- Intersection Individual Rule Based Inference

$$\mu_{B'}(y) = \mu_{B'_1}(y) * \mu_{B'_2}(y) * \ldots. * \mu_{B'_M}(y)$$

Fuzzy Inference Engine

Some known fuzzy inference engines are as follows:

- ***Product Inference Engine***
- ***Minimum Inference Engine***
- ***Lukasiewicz Inference Engine***
- ***Zadeh Inference Engine***
- ***Dienes-Rescher Inference Engine***

Product Inference Engine

In ***product inference engine***, we use:

- Individual rule based inference with union combination
- Mamdani's product implication
- Algebraic product for all the t-norm operators and max for all the s-norm operators.

$$\mu_{B'}(y) = \max_{l=1}^{M}\left[\sup_{x\in U}\left(\mu_{A'}(x)\prod_{i=1}^{n}\mu_{A_i^l}(x_i)\mu_{B^l}(y)\right)\right]$$

Minimum Inference Engine

In ***minimum inference engine***, we use:

- Individual rule based inference with union combination
- Mamdani's minimum implication
- min product for all the t-norm operators and max for all the s-norm operators.

$$\mu_{B'}(y) = \max_{l=1}^{M}\left[\sup_{x\in U}\left(\min\left(\mu_{A'}(x), \mu_{A_1^l}(x_1)\mu_{A_2^l}(x_2), \ldots, \mu_{A_n^l}(x_n), \mu_{B^l}(y)\right)\right)\right]$$

Lukasiewicz Inference Engine

In ***Lukasiewicz inference engine***, we use:

- Individual rule based inference with intersection combination
- Lukasiewicz implication
- min product for all the t-norm operators

$$\mu_{B'}(y) = \min_{l=1}^{M}\left[\sup_{x\in U}\left(\min\left(\mu_{A'}(x), 1-\min_{i=1}^{n}\left(\mu_{A_i^l}(x_i)\right)+\mu_{B^l}(y)\right)\right)\right]$$

Zadeh Inference Engine

In ***Zadeh inference engine***, we use:

- Individual rule based inference with intersection combination
- Zadeh implication
- min product for all the t-norm operators

$$\mu_{B'}(y) = \min_{l=1}^{M}\left[\sup_{x\in U}\left(\max\left(\min\left(\mu_{A'}(x), \mu_{A_1^l}(x_1)\mu_{A_2^l}(x_2),\ldots,\mu_{A_n^l}(x_n),\mu_{B^l}(y)\right), 1-\min_{i=1}^{n}\left(\mu_{A_i^l}(x_i)\right)\right)\right)\right]$$

Dienes-Rescher Inference Engine

In ***Dienes-Rescher Inference Engine***, we use:

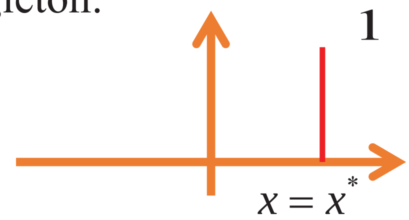
- Individual rule based inference with intersection combination
- Dienes-Rescher implication
- min product for all the t-norm operators

$$\mu_{B'}(y) = \min_{l=1}^{M}\left[\sup_{x\in U}\left(\min\left(\mu_{A'}(x), \max\left(1-\min_{i=1}^{n}\left(\mu_{A_i^l}(x_i)\right), \mu_{B^l}(y)\right)\right)\right)\right]$$

Fuzzy Inference Engine

Theorem: If the fuzzy set A' is a fuzzy singleton:

$$\mu_{A'}(x) = \begin{cases} 1 & x = x^* \\ 0 & o.w. \end{cases}$$



then:

- ***Product Inference Engine:***

$$\mu_{B'}(y) = \max_{l=1}^{M} \left[\prod_{i=1}^{n} \mu_{A_i^l}(x_i^*) \mu_{B^l}(y) \right]$$

- ***Minimum Inference Engine***

$$\mu_{B'}(y) = \max_{l=1}^{M} \left[\min\left(\mu_{A_1^l}(x_1^*) \mu_{A_2^l}(x_2^*), \ldots, \mu_{A_n^l}(x_n^*), \mu_{B^l}(y) \right) \right]$$

Fuzzy Inference Engine

Theorem:

- ***Lukasiewicz Inference Engine***

$$\mu_{B'}(y) = \min_{l=1}^{M} \left[\min\left(1, 1 - \min_{i=1}^{n} \left(\mu_{A_i^l}(x_i^*) \right) + \mu_{B^l}(y) \right) \right]$$

- ***Zadeh Inference Engine***

$$\mu_{B'}(y) = \min_{l=1}^{M} \left[\max\left(\min\left(\mu_{A_1^l}(x_1^*) \mu_{A_2^l}(x_2^*), \ldots, \mu_{A_n^l}(x_n^*), \mu_{B^l}(y) \right), 1 - \min_{i=1}^{n} \left(\mu_{A_i^l}(x_i^*) \right) \right) \right]$$

- ***Dienes-Rescher Inference Engine***

$$\mu_{B'}(y) = \min_{l=1}^{M} \left[\max\left(1 - \min_{i=1}^{n} \left(\mu_{A_i^l}(x_i^*) \right), \mu_{B^l}(y) \right) \right]$$

Fuzzy Inference Engine

Example: Suppose that a fuzzy rule base consists of only one rule:

$$IF\ x_1\ is\ A_1\ and\ x_2 is\ A_2\ and\ \ldots, and\ x_n\ is\ A_n, THEN\ y\ is\ B$$

where

$$\mu_B(y) \begin{cases} 1-|y| & -1 \le y \le 1 \\ 0 & o.w. \end{cases}$$

Assume that A' is a fuzzy singleton defined as follows:

$$\mu_{A'}(x) = \begin{cases} 1 & x = x^* \\ 0 & o.w. \end{cases}$$

and let $\mu_{A_p}(x_p^*) = \min\left(\mu_{A_1^l}(x_1^*)\mu_{A_2^l}(x_2^*),\ldots,\mu_{A_n^l}(x_n^*)\right)$ and $\mu_A(x^*) = \prod_{i=1}^{n} \mu_{A_i^l}(x_i^*)$.

Obtain $\mu_{B'}(y)$ by using the product, minimum, Lukasiewicz, Zadeh and Dienes-Rescher inference engines.

Thanks