

**Institute for Advanced Management Systems Research
Department of Information Technologies
Åbo Akademi University**

Fuzzy Logic Controllers - Tutorial

Robert Fullér

Directory

- **Table of Contents**
- **Begin Article**

Table of Contents

- 1. A basic feedback system**
- 2. Fuzzy logic controller**
- 3. Defuzzification methods**
- 4. Effectivity of fuzzy logic control systems.**

1. A basic feedback system

Conventional controllers are derived from control theory techniques based on mathematical models of the open-loop process, called *system*, to be controlled.

The purpose of the feedback controller is to guarantee a desired response of the output y .

The process of keeping the output y close to the setpoint (reference input) y^* , despite the presence disturbances of the system parameters, and noise measurements, is called regulation. The output of the controller (which is the input of the system) is the control action u .

The general form of the discrete-time control law is

$$u(k) = f(e(k), e(k-1), \dots, e(k-\tau), u(k-1), \dots, u(k-\tau))$$

providing a control action that describes the relationship between the input and the output of the controller.

- e represents the error between the desired setpoint y^* and the output of the system y ,
- parameter τ defines the order of the controller,
- f is in general a nonlinear function.

The seminal work by L.A. Zadeh (1973) on fuzzy algorithms introduced the idea of formulating the control algorithm by logical rules.

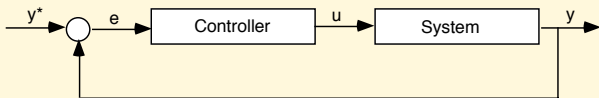


Figure 1: A basic feedback control system.

- L.A. Zadeh, Outline of a new approach to the analysis of complex systems and decision processes, *IEEE Transactions on Systems, Man and Cybernetics*, 3(1973) 28-44.

2. Fuzzy logic controller

In a fuzzy logic controller (FLC), the dynamic behavior of a fuzzy system is characterized by a set of linguistic description

rules based on expert knowledge. The expert knowledge is usually of the form

IF (a set of conditions are satisfied) **THEN** (a set of consequences can be inferred).

Since the antecedents and the consequents of these IF-THEN rules are associated with fuzzy concepts (linguistic terms), they are often called *fuzzy conditional statements*.

In our terminology, a *fuzzy control rule* is a fuzzy conditional statement in which the antecedent is a condition in its application domain and the consequent is a control action for the system under control.

Basically, fuzzy control rules provide a convenient way for expressing control policy and domain knowledge.

Furthermore, several linguistic variables might be involved in the antecedents and the conclusions of these rules.

When this is the case, the system will be referred to as a multi-input-multi-output (MIMO) fuzzy system. For example, in the case of two-input-single-output (MISO) fuzzy systems, fuzzy control rules have the form

\mathfrak{R}_1 : if x is A_1 and y is B_1 then z is C_1

also

\mathfrak{R}_2 : if x is A_2 and y is B_2 then z is C_2

also

...

also

\mathfrak{R}_n : if x is A_n and y is B_n then z is C_n

where x and y are the process state variables, z is the control

variable, A_i , B_i , and C_i are linguistic values of the linguistic variables x , y and z in the universes of discourse U , V , and W , respectively, and an implicit sentence connective *also* links the rules into a rule set or, equivalently, a rule-base.

We can represent the FLC in a form similar to the conventional control law

$$u(k) = F(e(k), e(k-1), \dots, e(k-\tau), u(k-1), \dots, u(k-\tau))$$

where the function F is described by a fuzzy rule-base. However it does not mean that the FLC is a kind of transfer function or difference equation.

The knowledge-based nature of FLC dictates a limited usage of the past values of the error e and control u because it is rather unreasonable to expect meaningful linguistic statements

for $e(k-3), e(k-4), \dots, e(k-\tau)$.

A typical FLC describes the relationship between the change of the control

$$\Delta u(k) = u(k) - u(k-1)$$

on the one hand, and the error $e(k)$ and its change

$$\Delta e(k) = e(k) - e(k-1).$$

on the other hand. Such control law can be formalized as

$$\Delta u(k) = F(e(k), \Delta(e(k)))$$

and is a manifestation of the general FLC expression with

$$\tau = 1.$$

The actual output of the controller $u(k)$ is obtained from the

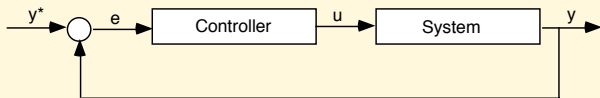
previous value of control $u(k - 1)$ that is updated by $\Delta u(k)$

$$u(k) = u(k - 1) + \Delta u(k).$$

This type of controller was suggested originally by *Mamdani and Assilian* in 1975 and is called the *Mamdani-type FLC*. A prototypical rule-base of a simple FLC realising the control law above is listed in the following

- \mathfrak{R}_1 : **If** e is "positive" and Δe is "near zero" **then** Δu is "positive"
- \mathfrak{R}_2 : **If** e is "negative" and Δe is "near zero" **then** Δu is "negative"
- \mathfrak{R}_3 : **If** e is "near zero" and Δe is "near zero" **then** Δu is "near zero"
- \mathfrak{R}_4 : **If** e is "near zero" and Δe is "positive" **then** Δu is "positive"
- \mathfrak{R}_5 : **If** e is "near zero" and Δe is "negative" **then** Δu is "negative"

So, our task is the find a crisp control action z_0 from the fuzzy rule-base and from the actual crisp inputs x_0 and y_0 :

Figure 2: Membership functions for the *error*.

\mathfrak{R}_1 : if x is A_1 and y is B_1 then z is C_1

also

\mathfrak{R}_2 : if x is A_2 and y is B_2 then z is C_2

also

...

also

\mathfrak{R}_n : if x is A_n and y is B_n then z is C_n

input x is x_0 and y is y_0

output

z_0

Of course, the inputs of fuzzy rule-based systems should be given by fuzzy sets, and therefore, we have to fuzzify the crisp inputs. Furthermore, the output of a fuzzy system is always a fuzzy set, and therefore to get crisp value we have to defuzzify it.

Fuzzy logic control systems usually consist from four major parts: *Fuzzification interface*, *Fuzzy rule-base*, *Fuzzy inference machine* and *Defuzzification interface*.

A fuzzification operator has the effect of transforming crisp data into fuzzy sets. In most of the cases we use fuzzy singletons as fuzzifiers

$$\text{fuzzifier}(x_0) := \bar{x}_0$$

where x_0 is a crisp input value from a process.

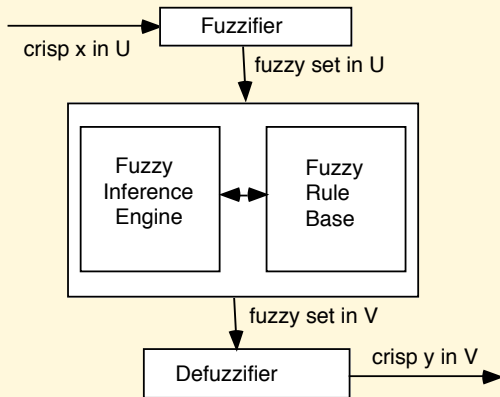


Figure 3: Fuzzy logic controller.

Suppose now that we have two input variables x and y . A fuzzy

control rule

\mathfrak{R}_i : if (x is A_i and y is B_i) then (z is C_i)

is implemented by a *fuzzy implication* R_i and is defined as

$$R_i(u, v, w) = [A_i(u) \text{ and } B_i(v)] \rightarrow C_i(w)$$

where the logical connective *and* is implemented by the minimum operator, i.e.

$$[A_i(u) \text{ and } B_i(v)] \rightarrow C_i(w) =$$

$$[A_i(u) \times B_i(v)] \rightarrow C_i(w) = \min\{A_i(u), B_i(v)\} \rightarrow C_i(w)$$

Of course, we can use any t-norm to model the logical connective *and*.

Fuzzy control rules are combined by using the sentence connective *also*.

Since each fuzzy control rule is represented by a fuzzy relation, the overall behavior of a fuzzy system is characterized by these fuzzy relations. In other words, a fuzzy system can be characterized by a single fuzzy relation which is the combination in question involves the sentence connective *also*.

Symbolically, if we have the collection of rules

\mathfrak{R}_1 : if x is A_1 and y is B_1 then z is C_1

also

\mathfrak{R}_2 : if x is A_2 and y is B_2 then z is C_2

also

...

also

\mathfrak{R}_n : if x is A_n and y is B_n then z is C_n

The procedure for obtaining the fuzzy output of such a knowledge base consists from the following three steps:

- Find the firing level of each of the rules.
- Find the output of each of the rules.
- Aggregate the individual rule outputs to obtain the overall system output.

To infer the output z from the given process states x , y and fuzzy relations R_i , we usually apply the compositional rule of inference

\mathcal{R}_1 : if x is A_1 and y is B_1 then z is C_1

\mathcal{R}_2 : if x is A_2 and y is B_2 then z is C_2

.....

\mathcal{R}_n : if x is A_n and y is B_n then z is C_n

fact : x is \bar{x}_0 and y is \bar{y}_0

conseq. z is C

where the consequence is computed by

consequence = **Agg** ($fact \circ \mathcal{R}_1, \dots, fact \circ \mathcal{R}_n$).

That is,

$$C = \mathbf{Agg}(\bar{x}_0 \times \bar{y}_0 \circ R_1, \dots, \bar{x}_0 \times \bar{y}_0 \circ R_n)$$

taking into consideration that

$$\bar{x}_0(u) = 0, \quad u \neq x_0$$

and

$$\bar{y}_0(v) = 0, \quad v \neq y_0,$$

the computation of the membership function of C is very simple:

$$C(w) = \mathbf{Agg}\{A_1(x_0) \times B_1(y_0) \rightarrow C_1(w), \\ \dots, A_n(x_0) \times B_n(y_0) \rightarrow C_n(w)\}$$

for all $w \in W$.

The procedure for obtaining the fuzzy output of such a knowledge base can be formulated as

- The firing level of the i -th rule is determined by

$$A_i(x_0) \times B_i(y_0).$$

- The output of of the i -th rule is calculated by

$$C'_i(w) := A_i(x_0) \times B_i(y_0) \rightarrow C_i(w)$$

for all $w \in W$.

- The overall system output, C , is obtained from the individual rule outputs C'_i by

$$C(w) = \mathbf{Agg}\{C'_1, \dots, C'_n\}$$

for all $w \in W$.

Example 2.1. *If the sentence connective also is interpreted as oring the rules by using minimum-norm then the membership function of the consequence is computed as*

$$C = (\bar{x}_0 \times \bar{y}_0 \circ R_1) \cup \dots \cup (\bar{x}_0 \times \bar{y}_0 \circ R_n).$$

That is,

$$C(w) = A_1(x_0) \times B_1(y_0) \rightarrow C_1(w) \vee \\ \dots \vee A_n(x_0) \times B_n(y_0) \rightarrow C_n(w)$$

for all $w \in W$.

3. Defuzzification methods

The output of the inference process so far is a fuzzy set, specifying a possibility distribution of control action.

In the on-line control, a nonfuzzy (crisp) control action is usually required.

Consequently, one must defuzzify the fuzzy control action (output) inferred from the fuzzy control algorithm, namely:

$$z_0 = \text{defuzzifier}(C),$$

where z_0 is the nonfuzzy control output and *defuzzifier* is the defuzzification operator.

Definition 3.1. *Defuzzification is a process to select a representative element from the fuzzy output C inferred from the fuzzy control algorithm.*

The most often used defuzzification operators are

- **Center-of-Area/Gravity.** The defuzzified value of a fuzzy set C is defined as its fuzzy centroid:

$$z_0 = \frac{\int_W zC(z) dz}{\int_W C(z) dz}.$$

The calculation of the Center-of-Area defuzzified value is simplified if we consider finite universe of discourse W and thus discrete membership function $C(w)$

$$z_0 = \frac{\sum z_j C(z_j)}{\sum C(z_j)}.$$

- **Center-of-Sums, Center-of-Largest-Area**
- **First-of-Maxima.** The defuzzified value of a fuzzy set C is its smallest maximizing element, i.e.

$$z_0 = \min\{z \mid C(z) = \max_w C(w)\}.$$

- **Middle-of-Maxima.** The defuzzified value of a discrete fuzzy set C is defined as a mean of all values of the uni-

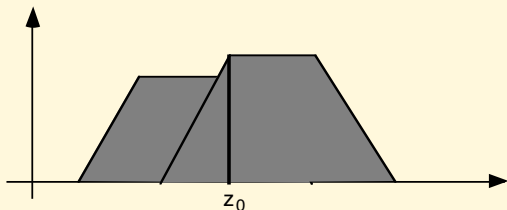


Figure 4: First-of-Maxima defuzzification method.

verse of discourse, having maximal membership grades

$$z_0 = \frac{1}{N} \sum_{j=1}^N z_j$$

where $\{z_1, \dots, z_N\}$ is the set of elements of the universe W which attain the maximum value of C .

If C is not discrete then defuzzified value of a fuzzy set C is defined as

$$z_0 = \frac{\int_G z dz}{\int_G dz}$$

where G denotes the set of maximizing element of C .

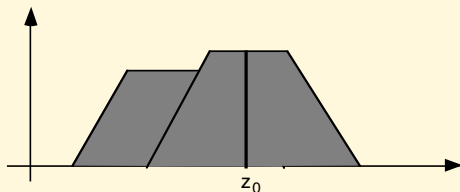


Figure 5: Middle-of-Maxima defuzzification method.

- **Max-Criterion.** This method chooses an arbitrary value,

from the set of maximizing elements of C , i.e.

$$z_0 \in \{z \mid C(z) = \max_w C(w)\}.$$

- Height defuzzification** The elements of the universe of discourse W that have membership grades lower than a certain level α are completely discounted and the defuzzified value z_0 is calculated by the application of the Center-of-Area method on those elements of W that have membership grades not less than α :

$$z_0 = \frac{\int_{[C]^\alpha} zC(z) dz}{\int_{[C]^\alpha} C(z) dz}.$$

where $[C]^\alpha$ denotes the α -level set of C as usually.

Example 3.1. *Consider a fuzzy controller steering a car in a way to avoid obstacles. If an obstacle occurs right ahead, the*

plausible control action depicted in Figure could be interpreted as

”turn right or left”.

Both Center-of-Area and Middle-of-Maxima defuzzification methods results in a control action ”drive ahead straightforward” which causes an accident.

A suitable defuzzification method would have to choose between different control actions (choose one of two triangles in the Figure) and then transform the fuzzy set into a crisp value.

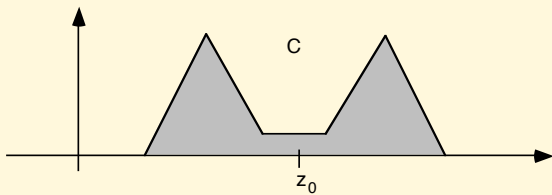


Figure 6: Undesired result by Center-of-Area and Middle-of-Maxima defuzzification methods.

4. Effectivity of fuzzy logic control systems.

Using the Stone-Weierstrass theorem Wang (1992) showed that fuzzy logic control systems of the form

$$\mathfrak{R}_i : \text{if } x \text{ is } A_i \text{ and } y \text{ is } B_i \text{ then } z \text{ is } C_i, \quad i = 1, \dots, n$$

with

- Gaussian membership functions

$$A_i(u) = \exp \left[-\frac{1}{2} \left(\frac{u - \alpha_{i1}}{\beta_{i1}} \right)^2 \right],$$

$$B_i(v) = \exp \left[-\frac{1}{2} \left(\frac{v - \alpha_{i2}}{\beta_{i2}} \right)^2 \right],$$

$$C_i(w) = \exp \left[-\frac{1}{2} \left(\frac{w - \alpha_{i3}}{\beta_{i3}} \right)^2 \right],$$

- Singleton fuzzifier

$$\text{fuzzifier}(x) := \bar{x}, \quad \text{fuzzifier}(y) := \bar{y},$$

- Product fuzzy conjunction

$$[A_i(u) \text{ and } B_i(v)] = A_i(u)B_i(v)$$

- Product fuzzy implication (Larsen implication)

$$[A_i(u) \text{ and } B_i(v)] \rightarrow C_i(w) = A_i(u)B_i(v)C_i(w)$$

- Centroid defuzzification method

$$z = \frac{\sum_{i=1}^n \alpha_{i3} A_i(x) B_i(y)}{\sum_{i=1}^n A_i(x) B_i(y)}$$

where α_{i3} is the center of C_i .

are universal approximators, i.e. they can approximate any continuous function on a compact set to arbitrary accuracy.

Namely, Wang proved the following theorem

L.-X. Wang. Fuzzy systems are universal approximators. In Proc. of the 1st IEEE Conf. on Fuzzy Systems, pages 1163-1170, San Deigo, CA, March 1992.

Theorem 4.1. *For a given real-valued continuous function g on the compact set U and arbitrary $\epsilon > 0$, there exists a fuzzy logic control system with output function f such that*

$$\sup_{x \in U} \|g(x) - f(x)\| \leq \epsilon.$$

In 1995 Castro (Fuzzy logic controllers are universal approximators, IEEE Transactions on Syst. Man Cybernetics, 25(1995) 629-635.) showed that Mamdani's fuzzy logic controllers

$$\mathfrak{R}_i : \text{if } x \text{ is } A_i \text{ and } y \text{ is } B_i \text{ then } z \text{ is } C_i, \quad i = 1, \dots, n$$

with

- Symmetric triangular membership functions

$$A_i(u) = \begin{cases} 1 - |a_i - u|/\alpha_i & \text{if } |a_i - u| \leq \alpha_i \\ 0 & \text{otherwise} \end{cases}$$

$$B_i(v) = \begin{cases} 1 - |b_i - v|/\beta_i & \text{if } |b_i - v| \leq \beta_i \\ 0 & \text{otherwise} \end{cases}$$

$$C_i(w) = \begin{cases} 1 - |c_i - w|/\gamma_i & \text{if } |c_i - w| \leq \gamma_i \\ 0 & \text{otherwise} \end{cases}$$

- Singleton fuzzifier

$$\text{fuzzifier}(x_0) := \bar{x}_0$$

- Minimum norm fuzzy conjunction

$$[A_i(u) \text{ and } B_i(v)] = \min\{A_i(u), B_i(v)\}$$

- Minimum-norm fuzzy implication

$$[A_i(u) \text{ and } B_i(v)] \rightarrow C_i(w) = \min\{A_i(u), B_i(v), C_i(w)\}$$

- Maximum t-conorm rule aggregation

$$\mathbf{Agg}(\mathfrak{R}_1, \mathfrak{R}_2, \dots, \mathfrak{R}_n) = \max(\mathfrak{R}_1, \mathfrak{R}_2, \dots, \mathfrak{R}_n)$$

- Centroid defuzzification method

$$z = \frac{\sum_{i=1}^n c_i \min\{A_i(x), B_i(y)\}}{\sum_{i=1}^n \min\{A_i(x), B_i(y)\}}$$

where c_i is the center of C_i .

are also universal approximators.