# Chapter 5. Fuzzy Logic Control System

~ In contrast to conventional control techniques, fuzzy logic control (FLC) is best utilized in complex ill-defined processes that can be controlled by a skilled human operator without much knowledge of their underlying dynamics.

~ The basic idea behind FLC is to incorporate the "expert experience" of a human operator in the design of the controller in controlling a process whose input – output relationship is described by collection of fuzzy control rules (e.g., IF-THEN rules) involving linguistic variables rather than a complicated dynamic model.

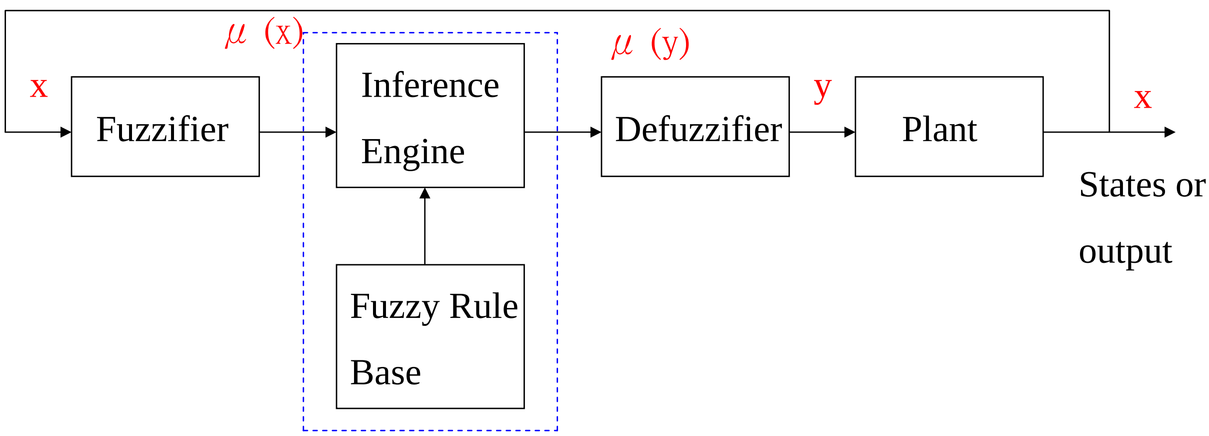
~ The utilization of linguistic variables, fuzzy control rules, and approximate reasoning provides a means to incorporate human expert experience in designing the controller.



~ FLC is strongly based on the concepts of fuzzy sets, linguistic variables and approximate reasoning introduced in the previous chapters.

~ This chapter will introduce the basic architecture and functions of fuzzy logic controller, and some practical application examples.

~ A typical architecture of FLC is shown below, which comprises of four principal comprises: a fuzzifier, a fuzzy rule base, inference engine, and a defuzzifier.



~ If the output from the defuzzifier is not a control action for a plant, then the system is fuzzy logic decision system.

~ The *fuzzifier* has the effect of transforming crisp measured data (e.g. speed is 10 mph) into suitable linguistic values (i.e. fuzzy sets, for example, speed is too slow).

~ The *fuzzy rule base* stores the empirical knowledge of the operation of the process of the domain experts.

~ The *inference engine* is the kernel of a FLC, and it has the capability of simulating human decision making by performing approximate reasoning to achieve a desired control strategy.

~ The *defuzzifier* is utilized to yield a nonfuzzy decision or control action from an inferred fuzzy control action by the inference engine.



· **Input and output spaces.**

~ A proper choice of process state variables and control variables is essential to characterization of the operation of a fuzzy logic control system (FLCS).

~ Expert experience and engineering knowledge play an important role during this state variables and control variables selection process.

~ Typically, the input variables in a FLC are the state, state error, state error derivative, state error integral, and so on.

~ The input vector x and the output state vector y can be defined, respectively, as

$$x = \left\{ \left( x_i, \mathrm{U}_i, \left\{ \mathrm{T}_{x_i}^{\,1}, \mathrm{T}_{x_i}^{\,2}, \cdots, \mathrm{T}_{x_i}^{\,k_i} \right\}, \left\{ \mu_{x_i}^{\,1}, \mu_{x_i}^{\,2}, \cdots, \mu_{x_i}^{\,k_i} \right\} \right) \Big|_{i=1,\cdots n} \right\}$$

$$y = \left\{ \left( y_i, \mathrm{V}_i, \left\{ \mathrm{T}_{y_i}^{\,1}, \mathrm{T}_{y_i}^{\,2}, \cdots, \mathrm{T}_{y_i}^{\,l_i} \right\}, \left\{ \mu_{y_i}^{\,1}, \mu_{y_i}^{\,2}, \cdots, \mu_{y_i}^{\,l_i} \right\} \right) \Big|_{i=1,\cdots m} \right\}$$

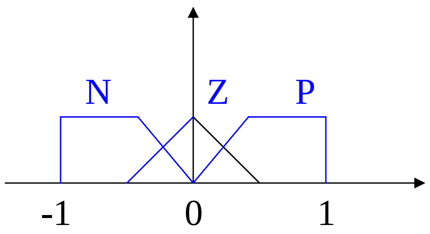
where the input linguistic variables $x_i$ form a fuzzy input space $U=U_1 \times U_2 \ldots \times U_n$ and the output linguistic variables $y_i$ form a fuzzy output space $V=V_1 \times V_2 \ldots V_m$.

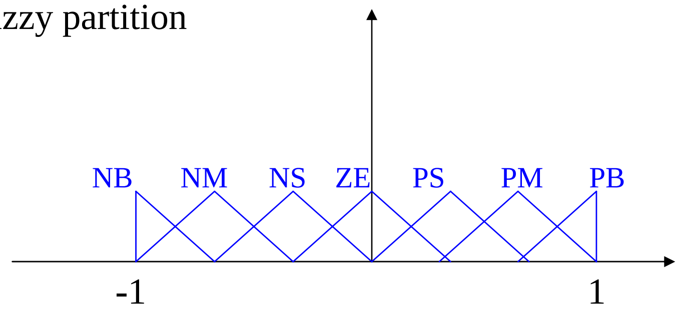
~ An input linguistic variable, variable $x_i$, is associated with a term set

$$T(x_i) = \left\{T_{x_i}^{1}, T_{x_i}^{2}, \cdots, T_{x_i}^{k_i}\right\}.$$

~ The size (or cardinality) of a term set, $|T(x_i)| = k_i$, is called the fuzzy partition number of $x_i$.

~ Diagrammatic representation of a fuzzy partition

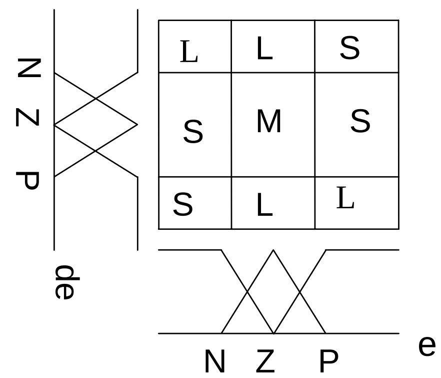





~ For a two-input FLC, the fuzzy input space is divided into many overlapping grids.

~ Grid-type partition:
N: Negative
Z: Zero
P: Positive
L: Large
S: Small
M: Medium




Rule set.
R1: IF e is N And de is N Then u is L
R2: IF e is Z And de is N Then u is L
R3: IF e is P And de is N Then u is S
R4: IF e is N And de is Z Then u is S
R5: IF e is Z And de is Z Then u is M
R6: IF e is P And de is Z Then u is S
R7: IF e is N And de is P Then u is S
R8: IF e is Z And de is P Then u is L
R9: IF e is P And de is P Then u is L

~ Furthermore, the fuzzy partitions in a fuzzy input space determine the maximum number of fuzzy control rules in a FLCS.

~ In the case of a two-input-one-output fuzzy logic control system, if $|T(x_1)| = 3$ and $|T(x_2)| = 7$, then the maximum number of fuzzy control rules is $3 \times 7$.

~ The input membership functions $\mu_{x_i}^{k}$, $k = 1, 2, \cdots k_i$ and the output membership functions $\mu_{y_i}^{\ell}$, $l = 1, 2, \cdots \ell_i$ used in a FLC are usually parametric functions such as triangular functions, trapezoid functions, and bell-shaped functions.

~ The triangular-shaped functions and the trapezoidal-shaped functions, can be represented by L-R fuzzy numbers, while the bell-shaped membership functions can be defined as



$$\mu_{x_i}(x) = \exp\left(-\frac{(x - m_i)^2}{\sigma_i^2}\right).$$

where $m_i$ and $\sigma_i$ specify the center location and the width of the bell-shaped function, respectively.

~ Proper fuzzy partition of input and output spaces and a correct choice of membership functions play an essential role in achieving a successful FLC design.

~ Traditionally, a heuristic trial-and-error procedure is usually used to determine an optimal fuzzy partition.

~ A promising approach to automating and speeding up these design choices is to provide a FLC with the ability to learn its input and output membership functions and fuzzy control rules.

· ***Fuzzifier***

~ A fuzzifier performs the function of fuzzification which is a subjective valuation to transform measurement data into valuation of a subjective value.

~ It can be defined as a mapping from an observed input space to labels of fuzzy sets in a specified input universe of discourse.

~ In fuzzy control application the observed data are usually crisp (though they may be corrupted by noise).

~ A natural and simple fuzzification approach is to convert a crisp value, $x_0$ ,into a fuzzy singleton, A, within the specified universe of discourse.

That is, the membership function of A, $\mu_A(x)$, is equal to 1 at the point $x_0$, and zero at other places. This approach is widely used in FLC applications because it greatly simplifies the fuzzy reasoning process.

In this case , for a specific value $x_i(t)$ at time t , it is mapped to the fuzzy set $T^1_{x_1}$ with degree $\mu^1_{x_1}(x_i(t))$ and to the fuzzy set $T^2_{x_i}$ with degree $\mu^2_{x_i}(x_i(t))$

and so on.



~ In a more complex case, where observed data are disturbed by random noise, a fuzzifier should convert the probabilistic data into fuzzy numbers, that is, fuzzy (possibility) data.

· ***Fuzzy Rule Base***

~ Fuzzy control rules are characterized by a collection of fuzzy IF-THEN rules in which the preconditions (antecedents) and consequents involve linguistic variables.

~ The general form of the fuzzy control rules in the case of multi-input-single-output systems (MISO) is:

$R^i$: IF x is $A_i$ , …, AND y is $B_i$ , THEN z is $C_i$ . i=1~ n.

where x, …, y and z are linguistic variables representing the process state variable and the control variable, respectively, and $A_i$ ,… ,$B_i$ , $C_i$ are the linguistic values of the linguistic values of the linguistic variables x, …, y and z in the universe of discourse U, …,V and W.

~ Another form :

$R^i$ : If x is $A_i$ , … . AND y is $B_i$ , THEN $z = f_i(x, \ldots, y)$.

where $f_i(x, \ldots, y)$ is a function of the process state variables x, …,y.

~ Both fuzzy control rules have linguistic values as inputs and either linguistic values or crisp values as outputs.

- ***Inference Engine :***

~ The inference engine is the kernel of FLC in modeling human decision making within the conceptual framework of fuzzy logic and approximate reasoning.

~ The generalized modus pones (forward data-driven inference) plays an especially important role in approximate reasoning.

~ The generalized modus pones can be rewritten as



Premise 1: IF x is A, THEN y is B. ------- (*)

Premise 2: x is A'

Conclusion: y is B'

where A, A', B and B'are fuzzy predicates (fuzzy sets or relations) in the universal sets U, U ,V and V, respectively.

~ In general, a fuzzy control rule ( e.g. premise 1 in Eq (*)) is a fuzzy relation which is expressed as a fuzzy implication, $R = A \rightarrow B$.

~ According to the compositional rule of inference conclusion, B' can be obtained by taking the composition of fuzzy set A'and the fuzzy relation (here the fuzzy relation is a fuzzy implication) $A \rightarrow B$:

$$B' = A' \circ R = A' \circ (A \rightarrow B). \quad \text{---------} \quad (*)$$

~ In addition to the definitions of fuzzy composition and implication given in Chap. 6, there are four types of compositional operators that can be used in the compositional rule of inference.These correspond to the four operations associated with the t-norms.

— Max-min operation.

— Max product operation.

— Max bounded product (max — $\odot$ ) operation.

— Max drastic product (max — $\dot{\wedge}$ ) operation.

~ In FLC applications, the *max-min* and *max-product* compositional operators are the most commonly and frequently used due to their computational efficiency.

Let max—☆ represent any one of the above four composition operations. Then (*) becomes :



B'= A' ☆ R = A' ☆ (A → B )

$$\mu_{B'}(v) = \max_{u}[\mu_{A'}(u) \;☆\; \mu_{A \to B}(u,v)]$$

~ As for the fuzzy implication A → B, there are nearly 40 distinct fuzzy implication functions described in the existing literature, e.g. (see Table 7.1 )

| Rule of Fuzzy Implication | Implication Formulas | Fuzzy Implication $\mu_{A \to B}(u, v)$ |
|---|---|---|
| $R_C$ : min operation[Mamdani] | a → b= a∧b | $= \mu_A(u) \wedge \mu_B(v)$ |
| $R_P$ : product operation[Larsen] | a → b= a • b | $= \mu_A(u) \bullet \mu_B(v)$ |
| $R_{bP}$:bounded product | a → b= 0∨ (a+b-1) | $= 0 \vee [\mu_A(u) + \mu_B(v) - 1]$ |
| $R_{dP}$:drastic product | a → b= $\begin{cases} a & b=1 \\ b & a=1 \\ 0 & a, b<1 \end{cases}$ | ⋮ |
| Ra: arithmetic rule [Zadeh] | a → b=1∧ (1-a+b) | |

~ The fuzzy implication rules defined in Table 7.1 are generated from the fuzzy conjunction, fuzzy disjunction, or fuzzy implication by employing various t-norms or t-conorms.

~ The above four implications are all t-norms. For example, Mamdani's min fuzzy implication $R_c$ is obtained if the intersection operator is used in the fuzzy conjunction. (fuzzy conjunction $\mu_{A\cap B}(x,y) \triangleq t(\mu_A(x),\mu_B(y))$. )

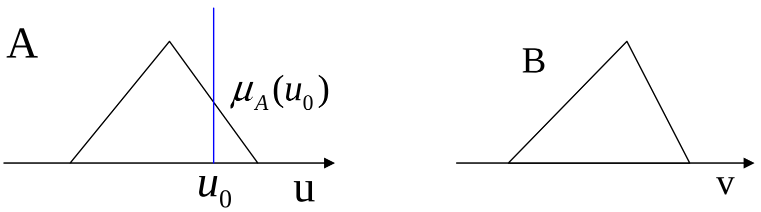
~ Larsen 's product fuzzy implication $R_P$ is obtained if the algebraic product is used in the fuzzy conjunction.

~ Rbp and Rdp are obtained if the bounded product and the drastic product are used in the fuzzy conjunction.

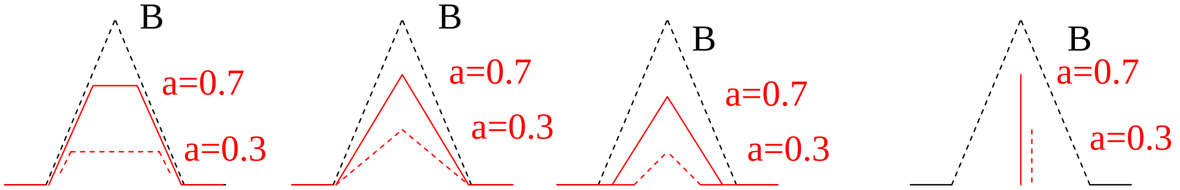
Ex : Assume fuzzy set A' is a singleton at $u_0$; i.e.

$$\mu_{A'}(u_0) = 1 \text{ and } \mu_{A'}(u) = 0 \quad \text{for } u \neq u_0$$






The consequent B ' under the fuzzy implications A → B is as follows , where $\mu_A(u_0) = a$ with $a$ =0.3 (dotted line) And $a$=0.7 ( solid line).




(1)$R_C$ : $a \wedge b$ (2)$R_P$ : $a \cdot b$ (3) $R_{bp}$ :0∨ $(a+b-1)$ (4) $R_{dp}$: $\begin{cases} a & b=1 \\ b & a=1 \\ 0 & a,b<1 \end{cases}$

$$\boxed{\begin{aligned} \mu_{B'}(v) &= \max_u \{\mu_{A'}(u) ☆ [\mu_A(u) \to \mu_B(v)]\} \\ &= 1 ☆ (\mu_A(u_0) \to \mu_B(v)) = \mu_A(u_0) \to \mu_B(v) \end{aligned}}$$

~ Among the various fuzzy implications in Table7.1, Mamdani's fuzzy implication method $R_C$ associated with the max-min composition is the most frequently used in fuzzy logic control.

Ex 7.2 : Premise 1: IF x is A, THEN y is B

Premise 2: x is A'

Conclusion : y is B'.

derive the conclusion B', when

A'=A, A'= VERY A, A'=MORE OR LESS A, A'=NOT A.

Here, Mamdani's method $R_C$ and max-min composition are adopted.

(a) For A'= A $\qquad B' = A \circ R_C$

$$\mu_{B'}(v) = \bigvee_u \{ \mu_A(u) \wedge (\mu_A(u) \wedge \mu_B(v)) \} = \bigvee_u \{ \mu_A(u) \wedge \mu_B(v) \}$$

$$= \bigvee_u \mu_A(u) \wedge \mu_B(v) = 1 \wedge \mu_B(v) = \mu_B(v)$$



(b) For A'= $A^2$ (VERY A).

$$\mu_{B'}(v) = \bigvee_u \{ \mu^2_A(u) \wedge (\mu_A(u) \wedge \mu_B(v)) \} = \bigvee_u \{ \mu^2_A(u) \wedge \mu_B(v) \}$$

$$= 1 \wedge \mu_B(v) = \mu_B(v)$$

(c) For A'= 1 − A (NOT A).

$$\mu_{B'}(v) = \bigvee_u \{ (1 - \mu_A(u)) \wedge (\mu_A(u) \wedge \mu_B(v)) \}$$

$$= \bigvee_u \{ (1 - \mu_A(u)) \wedge \mu_A(u) \} \wedge \mu_B(v) = 0.5 \wedge \mu_B(v).$$

Table 7.2 Summary of inference results for generalized modus pones (max-min composition).

| | A | Very A ($A^2$) | More or Less A ($A^{1/2}$) | Not A. |
|---|---|---|---|---|
| $R_c$ | $\mu_B$ | $\mu_B$ | $\mu_B$ | $0.5 \wedge \mu_B$ |
| $R_p$ | $\mu_B$ | $\mu_B$ | $\mu_B$ | $\frac{\mu_B}{(1+\mu_B)}$ |
| $R_a$ | $\frac{1+\mu_B}{2}$ | $\frac{3+2\mu_B-\sqrt{5+4\mu_B}}{2}$ | $\frac{\sqrt{5+4\mu_B}-1}{2}$ | 1 |
| $R_s$ | $\mu_B$ | $\mu^2_B$ | $\sqrt{\mu_B}$ | 1 |

~ The above results are all based on the max-min composition.

~ for A'=A, B'=A o $R_a$ = (1+B)/2 $\neq$ B.

$\therefore$ $R_a$ does not satisfy normal modus pones if max-min composition is used.

~ Let$\square$ and$\triangle$ denote the max bounded product and the max drastic product compositions, respectively. Then we have

$$B' = A \square R_a = A \triangle R_a = B$$

that is, it satisfies the modus ponens.

- ***Application of the generalized modus pones in the inference engine of a FLC***

~ In most cases, the fuzzy rule base has the form of multi-input-multi-output (MIMO) system.

R= {$R^1_{MIMO}$, $R^2_{MIMO}$, $R^3_{MIMO}$, … , $R^n_{MIMO}$ }.

where $R^i_{MIMO}$ represents the ith rule:

IF(x is $A_i$ AND … AND y is $B_i$) THEN ($Z_1$ is $C^1_i$, …, $Z_q$ is $C^q_i$ )



$Z_q$: qth control variable.

$C^q_i$: output predicate of the qth control variable.

The precondition of $R^i_{MIMO}$ forms a fuzzy set $A_i \times \ldots \times B_i$ in the product space $U \times \ldots \times V$, and the consequent is the union of q independent control actions.

~ $R^i_{MIMO}$ may be represented as a fuzzy implication

$$R^i_{MIMO}: (A_i \times \cdots \times B_i) \rightarrow (C^1_i + \cdots + C^q_i) = \bigcup_{k=1}^{q} (A_i \times \cdots \times B_i) \rightarrow C^k_i$$

Where + represents the union of q independent control actions or variable.

$$R = \left\{ \bigcup_{i=1}^{n} R^i_{MIMO} \right\} = \left\{ \bigcup_{i=1}^{n} \left[ (A_i \times \cdots \times B_i) \rightarrow (C^1_i + \cdots + C^q_i) \right] \right\}$$

$$= \left\{ \bigcup_{i=1}^{n} \bigcup_{k=1}^{q} \left[ (A_i \times \cdots \times B_i) \rightarrow C^k_i \right] \right\}$$

$$= \left\{ \bigcup_{k=1}^{q} \bigcup_{i=1}^{n} \left[ (A_i \times \cdots \times B_i) \rightarrow C^k_i \right] \right\}$$

$$= \left\{ RB^1_{MISO} , RB^2_{MISO} , \cdots , RB^q_{MISO} \right\}.$$

e.g.,

$R_{MIMO}:$ IF x is $A_1$ AND y is $B_1$ THEN $Z_1$ is $C_1^1$ AND $Z_2$ is $C_1^2$

IF x is $A_2$ AND y is $B_2$ THEN $Z_1$ is $C_2^1$ AND $Z_2$ is $C_2^2$

IF x is $A_3$ AND y is $B_3$ THEN $Z_1$ is $C_3^1$ AND $Z_2$ is $C_3^2$

can be represented by

$RB_{MISO}^1:$ IF x is $A_1$ AND y is $B_1$ THEN $Z_1$ is $C_1^1$

IF x is $A_2$ AND y is $B_2$ THEN $Z_1$ is $C_2^1$

IF x is $A_3$ AND y is $B_3$ THEN $Z_1$ is $C_3^1$

$RB_{MISO}^2:$ IF x is $A_1$ AND y is $B_1$ THEN $Z_2$ is $C_1^2$

IF x is $A_2$ AND y is $B_2$ THEN $Z_2$ is $C_2^2$

IF x is $A_3$ AND y is $B_3$ THEN $Z_2$ is $C_3^2$



~ Observation:

(1)The above equation shows that the fuzzy rule base R of a FLC is composed of a set of q subrules bases $RB^i_{MISO}$, with each subrule base $RB^i_{MISO}$ consisting of *n* fuzzy control rules with multiple process state variables and a single control variable.

(2)Thus, a MIMO system with n inputs and q outputs can be decomposed into *q n*-input-single-output subsystems, and each of these *q* subsystems has a set of subrule base $\{RB^i_{MIMO}\}$.

(3)Hence, instead of considering fuzzy control rules for a MIMO system, we will consider fuzzy control rules only for a MISO system.

~ The general form of a MISO fuzzy control rules in the case of two-input-single-output fuzzy systems is

| | | |
|---|---|---|
| | Input | : x is A' AND y is B' |
| | $R^1$ | : IF x is $A_1$ AND y is $B_1$, THEN Z is $C_1$ |
| (*) | ALSO $R^2$ | : IF x is $A_2$ AND y is $B_2$ , THEN Z is $C_2$ |
| | ⋮ | ⋮ |
| | ALSO $R^n$ | : IF x is $A_N$ AND y is $B_n$, THEN Z is $C_n$ |
| | Conclusion | : Z is C'. |

~ In the above rules, the connectives AND and ALSO may be interpreted as either intersection ($\cap$) or union ($\cup$) for different definitions of fuzzy implication in Table 7.1.



Theorem 1:

Consider the whole set of rules in Eq (*), with minimum t-norm and Mamdani 's minimum fuzzy implication $R_C$, the conclusion $C'$ can be expressed as a unification of the individual conclusions of fuzzy control rules. That is,

$$C' = (A', B') \text{ o } \bigcup_{i=1}^{n} R_C(A_i , B_i ; C_i ) = \bigcup_{i=1}^{n}(A', B') \text{ o } R_C(A_i,B_i;C_i)$$

pf:
$$\begin{aligned}
\mu_{C'}(w) &= (\mu_{A'}(u) \wedge \mu_{B'}(v)) \text{ o } V(\mu_{R_1}(u,v,w),\cdots,\mu_{R_n}(u,v,w)) \\
&= \underset{u,v}{V}\left\{(\mu_{A'}(u) \wedge \mu_{B'}(v)) \wedge [V(\mu_{R_1}(u,v,w),\cdots,\mu_{R_n}(u,v,w)) \;]\right\} \\
&= V \underset{u,v}{V}\left\{[(\mu_{A'}(u) \wedge \mu_{B'}(v)) \wedge \mu_{R_1}(u,v,w)],\cdots,[(\mu_{A'}(u) \wedge \mu_{B'}(v)) \wedge \mu_{R_n}(u,v,w)]\right\} \\
&= V\left\{[(\mu_{A'}(u) \wedge \mu_{B'}(v)) \text{ o } \mu_{R_1}(u,v,w)],\cdots,[(\mu_{A'}(u) \wedge \mu_{B'}(v)) \text{ o } \mu_{R_n}(u,v,w)]\right\}.
\end{aligned}$$

where $\mu_{R_i}(u,v,w) = \mu_{A_i}(u) \wedge \mu_{B_i}(v) \wedge \mu_{C_i}(w)$ .

Remark:



(1) The above theorem is also true for $R_P$. $R_{bP}$, $R_{dP}$.

(2) The rule connective ALSO is interpreted as the union operator ( $\cup$ ) for $R_C$, $R_P$ , $R_{bP}$ and $R_{dP}$ fuzzy implication.

(3) On the other hand, the connective ALSO is interpreted as the intersection operator ( $\cap$ ) for $R_a$ , $R_m$ , $R_s$ , $R_b$,and $R_D$ fuzzy implications, so, we have

$$C' = (A', B') \circ \bigcap_{i=1}^{n} R_a(A_i, B_i; C_i) = \bigcap_{i=1}^{n} (A', B') \circ R_a(A_i, B_i; C_i).$$

(4) The above theorem still holds if we use the max-product composition instead of the max-min composition.

~ We shall focus on two special fuzzy implication rules $R_C$ and $R_P$, which are most commonly used in FLCS.

~ Since in fuzzy control the inputs are usually fuzzy singletons, namely, $A' = u_0$ and $B' = v_0$ in Eq. (*), the following theorem plays an important role in FLC applications.



Theorem 2:

Consider the max-min compositional operator $\circ$ and minimum t-norm. If the inputs are fuzzy singletons, namely, $A' = u_0$ and $B' = v_0$ , then the results $C'$ in Eq. (*) derived by employing Mamdani's minimum operation rule $R_c$ and Larsen 's product operation rule $R_P$, respectively, may be expressed simply as:

$$(**) \qquad R_C : \mu_{C'}(w) = \bigvee_{i=1}^{n} \Phi_i \wedge \mu_{C_i}(w) \equiv \bigvee_{i=1}^{n} \left[ \mu_{A_i}(u_0) \wedge \mu_{B_i}(v_0) \right] \wedge \mu_{C_i}(w).$$

$$R_P : \mu_{C'}(w) = \bigvee_{i=1}^{n} \Phi_i \bullet \mu_{C_i}(w) \equiv \bigvee_{i=1}^{n} \left[ \mu_{A_i}(u_0) \wedge \mu_{B_i}(v_0) \right] \bullet \mu_{C_i}(w)$$

where $\Phi_i$ denotes the *firing strength* of the i-th rule which is a measure of the contribution of the i-th rule to the fuzzy control action.

Pf: For $R_C$ :



$$C' = (A', B') \circ R_C(A_i, B_i; C_i),\ i = 1, 2, \cdots n$$

$$\mu_{C_i'}(w) = \bigvee_{u,v}\left[\mu_{A'}(u) \wedge \mu_{B'}(v)\right] \wedge \left[\mu_{A_i}(u) \wedge \mu_{B_i}(v) \wedge \mu_{C_i}(w)\right]$$

$$= \{\bigvee_{u}[\mu_{A'}(u) \wedge \mu_{A_i}(u)] \wedge \bigvee_{v}[\mu_{B}(v) \wedge \mu_{B_i}(v)]\} \wedge \mu_{C_i}(w)$$

$$= [\mu_{A_i}(u_0) \wedge \mu_{B_i}(v_0)] \wedge \mu_{C_i}(w)$$

$$\mu_{C'}(w) = \bigvee_{i=1}^{n}\mu_{C_i'}(w) = \bigvee_{i=1}^{n}\left[\mu_{A_i}(u_0) \wedge \mu_{B_i}(v_0)\right] \wedge \mu_{C_i}(w).$$

For $R_P$ :

$$C' = (A', B') \circ R_P(A_i, B_i; C_i),\ i = 1, 2, \cdots, n$$

$$\mu_{C_i'}(w) = \bigvee_{u,v}\left[\mu_{A'}(u) \wedge \mu_{B'}(v)\right] \wedge \{[\mu_{A_i}(u) \wedge \mu_{B_i}(v)] \bullet \mu_{C_i}(w)\}$$

$$= \bigvee_{u}\{\mu_{A'}(u) \wedge [\mu_{A_i}(u) \bullet \mu_{C_i}(w)]\} \wedge \bigvee_{v}\{\mu_{B'}(v) \wedge [\mu_{B_i}(v) \bullet \mu_{C_i}(w)]\}$$

$$= \mu_{A_i}(u_0) \bullet \mu_{C_i}(w) \wedge \mu_{B_i}(v_0) \bullet \mu_{C_i}(w)$$

$$= [\mu_{A_i}(u_0) \wedge \mu_{B_i}(u_0)] \bullet \mu_{C_i}(w)$$

$$\mu_{C'}(w) = \bigvee_{i=1}^{n}\mu_{C_i'}(w) = \bigvee_{i=1}^{n}\left[\mu_{A_i}(u_0) \wedge \mu_{B_i}(v_0)\right] \bullet \mu_{C_i}(w).$$

Remarks:



1. Eq (**) is the most frequently used in fuzzy control applications
2. For fuzzy input $A'$ and $B'$

(1)AND: use $t(a,b) = a \wedge b$, max-min composition, implication: $R_C$

$$C' = (A', B') \circ R_C(A_i, B_i; C_i)$$

$$\mu_{C_i'}(w) = \bigvee_{u,v}\left[\mu_{A'}(u) \wedge \mu_{B'}(v)\right] \wedge \left[\mu_{A_i}(u) \wedge \mu_{B_i}(v) \wedge \mu_{C_i}(w)\right]$$

$$= \{\bigvee_{u}[\mu_{A'}(u) \wedge \mu_{A_i}(u)] \wedge \bigvee_{v}[\mu_{B}(v) \wedge \mu_{B_i}(v)]\} \wedge \mu_{C_i}(w)$$

$$= (\mu_1' \wedge \mu_2') \wedge \mu_{C_i}(w)$$

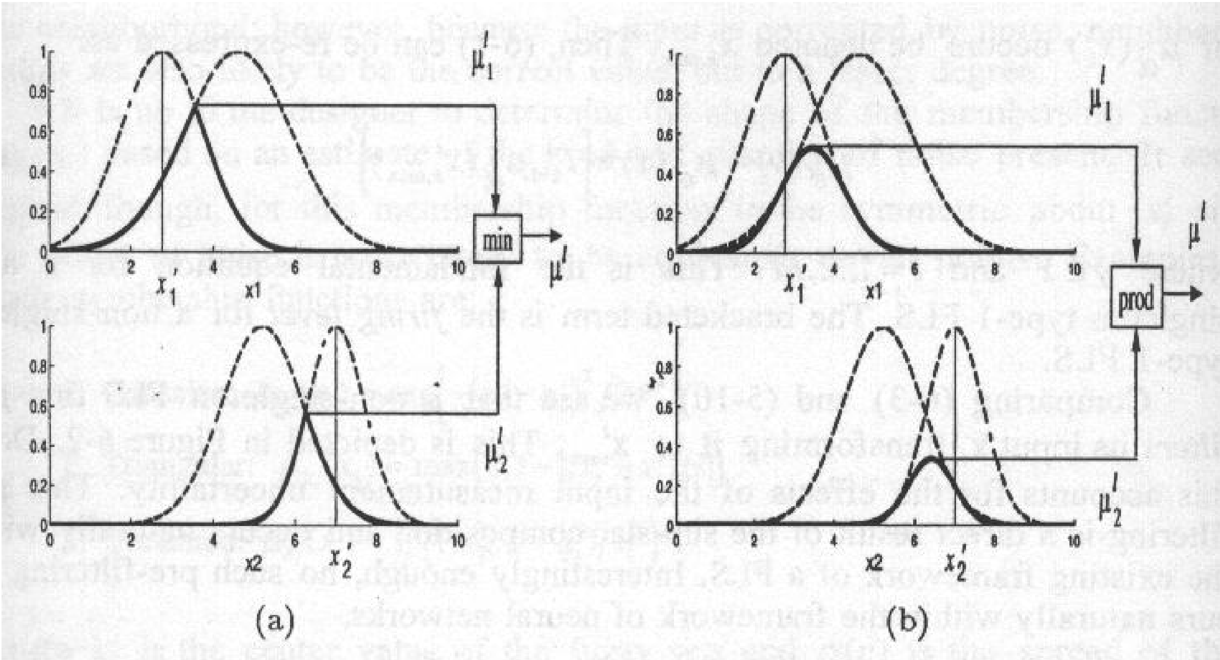
(2)AND: use $t(a,b) = a \bullet b$, max-product composition, implication: $R_P$

$$C' = (A', B') * R_P(A_i, B_i; C_i)$$

$$\mu_{C_i'}(w) = \bigvee_{u,v}\left[\mu_{A'}(u) \bullet \mu_{B'}(v)\right] \bullet \left[\mu_{A_i}(u) \bullet \mu_{B_i}(v) \bullet \mu_{C_i}(w)\right]$$

$$= \{\bigvee_{u}[\mu_{A'}(u) \bullet \mu_{A_i}(u)] \bullet \bigvee_{v}[\mu_{B}(v) \bullet \mu_{B_i}(v)]\} \bullet \mu_{C_i}(w)$$

$$= (\mu_1' \bullet \mu_2') \bullet \mu_{C_i}(w)$$






• ***Types of fuzzy reasoning***

~ The two types of fuzzy reasoning currently employed in FLC applications are as follows:

-First type (Mamdani type)

Assume that we have two fuzzy control rules as follows:

$R^1$ : IF x is $A_1$ AND y is $B_1$ ,THEN Z is $C_1$.

$R^2$ : IF x is $A_2$ AND y is $B_2$ ,THEN Z is $C_2$.

then the firing strengths $\Phi_1$ and $\Phi_2$ of the first and second rules may be expressed as

$$\Phi_1 = \mu_{A_1}(x_0) \wedge \mu_{B_1}(y_0) \text{ and } \Phi_2 = \mu_{A_2}(x_0) \wedge \mu_{B_2}(y_0).$$

where $\mu_{A1}(x_0)$ and $\mu_{B1}(y_0)$ indicate the degrees of partial match between the user-supplied data and the data in the fuzzy rule base.

(1a) Mamdani 's minimum fuzzy implication rule, $R_{C.}$

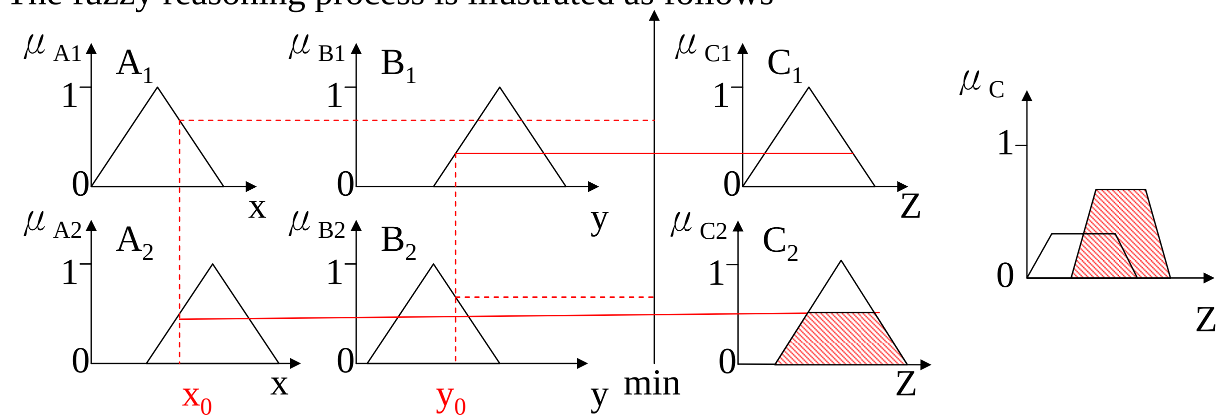
In this mode of reasoning. the ith fuzzy control rule leads to the control decision

$$\mu_{C_i'}(w) = \Phi_i \wedge \mu_{C_i}(w).$$

The final inferred consequent C is given by

$$\mu_C(w) = \mu_{C_1'} \vee \mu_{C_2'} = \left[\Phi_1 \wedge \mu_{C_1}(w)\right] \vee \left[\Phi_2 \wedge \mu_{C_2}(w)\right]$$

The fuzzy reasoning process is illustrated as follows



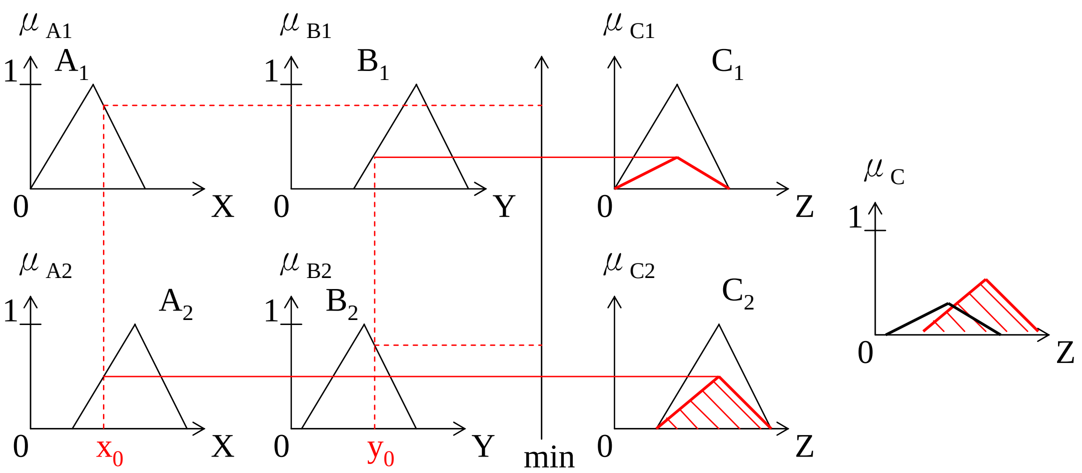



(1b). Larsen's product fuzzy implication rule, $R_{P.}$

In this case, the ith fuzzy control rule leads to the control decision

$$\mu_{C_i'}(w) = \Phi_i \bullet \mu_{C_i}(w).$$

$$\mu_C(w) = \mu_{C_1'} \vee \mu_{C_2'} = \left[\Phi_1 \bullet \mu_{C_1}(w)\right] \vee \left[\Phi_2 \bullet \mu_{C_2}(w)\right]$$



- Second type (*TSK-type* or *TS-type*)

The consequent of a rule is a function of input linguistic variables.

~ $R^i$ : IF x is $A_i$ … AND y is $B_i$ , THEN z = $f_i$ (x , … , y). Consider two fuzzy control rules as follows:

$R^1$: IF x is $A_1$ AND y is $B_1$ , THEN z is $f_1$(x,y)

$R^2$: IF x is $A_2$ AND y is $B_2$ , THEN z is $f_2$(x,y)

~ When the inputs are $x_0$ and $y_0$, the inferred values of the control action from the first and second rules are $f_1(x_0,y_0)$ and $f_2(x_0,y_0)$ , respectively. A crisp control action is given by

$$z_0 = \frac{\Phi_1 f_1(x_0, y_0) + \Phi_2 f_2(x_0, y_0)}{\Phi_1 + \Phi_2}$$

~ This method was proposed by Takagi, Sugeno and Kang, and is usually called *TSK-type* or *TS-type.*

~ In general, $f_i\ (x , \ldots , y)$ is a linear combination of the input variables plus a constant. i.e. $f_i = a_i^0 + a_i^1 x + \cdots + a_i^n y$



- ***Defuzzifier***

~A mapping from a space of fuzzy control action defined over an output universe of discourse into a space of non-fuzzy (crisp) control actions.

~ A defuzzifier is necessary when fuzzy reasoning of the first type is used.

~ Two commonly used methods of defuzzification are the *center of area (COA)* method and the *mean of maximum (MOM)* method.

~ COA:

(1) In the case of a discrete universe of discourse.

$$z^*_{COA} = \frac{\sum_{j=1}^{n} \mu_c(z_j) z_j}{\sum_{j=1}^{n} \mu_c(z_j)}$$

where n is the number of quantization levels of the output , $z_j$ is the amount of control output at the quantization level j , and $\mu_c(z_j)$ represents its membership value in the output fuzzy set c.

(2) a continuous universe of discourse.

$$z^*_{COA} = \frac{\int_z \mu_c(z) z dz}{\int_z \mu_c(z) dz}$$

~ MOM:

~ Generates a control action that represents the mean value of all local control action whose membership functions reach the maximum.

~ In the case of a discrete universe,

$$z^*_{MOM} = \sum_{j=1}^{m} \frac{z_j}{m}$$

where $z_j$ is the support value at which the membership function reaches the maximum value $\mu_c(z_j)$ and m is the number of such support values.



Ex 7.3:

We are given a fuzzy logic control system with the following two fuzzy control rules:

Rule 1 : IF x is $A_1$ AND y is $B_1$ , THEN z is $C_1$

Rule 2 : IF x is $A_2$ AND y is $B_2$ , THEN z is $C_2$

Suppose $x_0$ and $y_0$ are the sensor readings for linguistic input variables x and y and

$$\mu_{A1}(x) = \begin{cases} (x-2)/3 & 2 \le x \le 5 \\ (8-x)/3 & 5 < x \le 8 \end{cases} \qquad \mu_{A2}(x) = \begin{cases} (x-3)/3 & 3 \le x \le 6 \\ (9-x)/3 & 6 < x \le 9 \end{cases}$$

$$\mu_{B1}(y) = \begin{cases} (y-5)/3 & 5 \le y \le 8 \\ (8-y)/3 & 8 < y \le 11 \end{cases} \qquad \mu_{B2}(y) = \begin{cases} (y-4)/3 & 4 \le y \le 7 \\ (10-y)/3 & 7 < y \le 10 \end{cases}$$

$$\mu_{C1}(z) = \begin{cases} (z-1)/3 & 1 \le z \le 4 \\ (7-z)/3 & 4 < z \le 7 \end{cases} \qquad \mu_{C2}(z) = \begin{cases} (z-3)/3 & 3 \le z \le 6 \\ (9-z)/3 & 6 < z \le 9 \end{cases}$$

AND: minimum operation, Implication: Rc

at time $t_1$, we have $x_0 = 4$ and $y_0 = 8$.

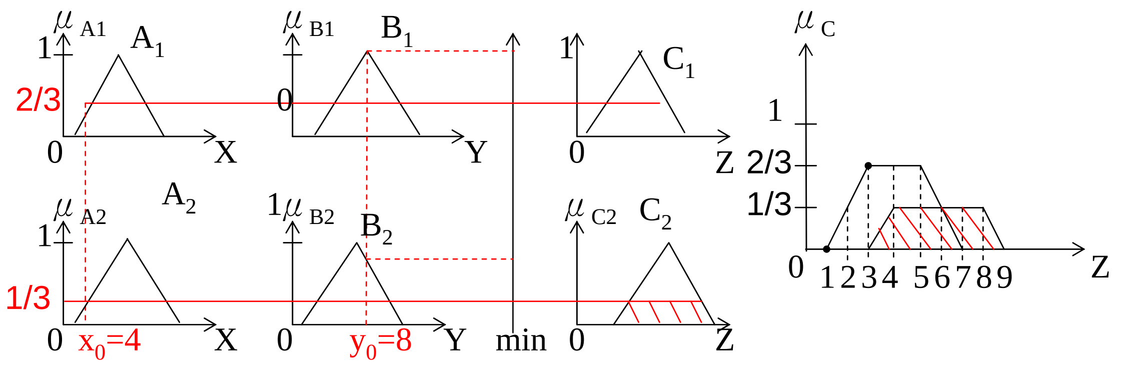
$\mu_{A1}(4) = 2/3$, $\mu_{B1}(8) = 1$, $\mu_{A2}(4) = 1/3$, $\mu_{B2}(8) = 2/3$.

$\Phi_1 = \min(\mu_{A1}(x_0), \mu_{B1}(y_0)) = \min(2/3, 1) = 2/3$

$\Phi_2 = \min(\mu_{A2}(x_0), \mu_{B2}(y_0)) = \min(1/3, 2/3) = 1/3$

$$z^*_{COA} = \frac{2(\frac{1}{3})+3(\frac{2}{3})+4(\frac{2}{3})+5(\frac{2}{3})+6(\frac{1}{3})+7(\frac{1}{3})+8(\frac{1}{3})}{\frac{1}{3}+\frac{2}{3}+\frac{2}{3}+\frac{2}{3}+\frac{1}{3}+\frac{1}{3}+\frac{1}{3}} = 4.7$$

$$z^*_{MOM} = (3+4+5)/3 = 4.0$$




Ex: (TSK- type fuzzy rules)

Rule 1 : IF x is $A_1$ AND y is $B_1$ , THEN z=2+2x-4y

Rule 2 : IF x is $A_2$ AND y is $B_2$ , THEN z=1+3x+y

Suppose $x_0$= 4 and $y_0$ = 8, find z=?

Sol:

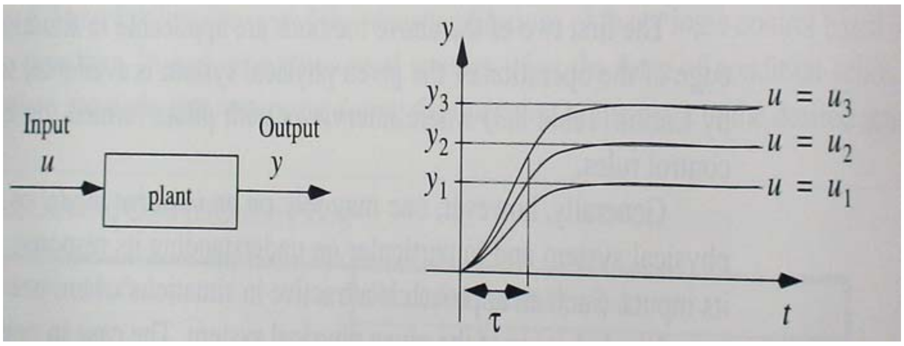
$\Phi_1$=2/3 and $\Phi_2$=1/3 (from previous example)

Rule 1: $z = 2+2(4)-4(8) = -22$

Rule 2: $z = 1+3(4)+(8) = 21$

Final output: $z = \frac{\frac{2}{3}(-22)+\frac{1}{3}(21)}{\frac{2}{3}+\frac{1}{3}} = \frac{-23}{3}$

Example:




Response of a simple generic plant

error $e = y_d - y$, control action: $u$

Normalized error: $e_n(t) = n_e \cdot e(t),$

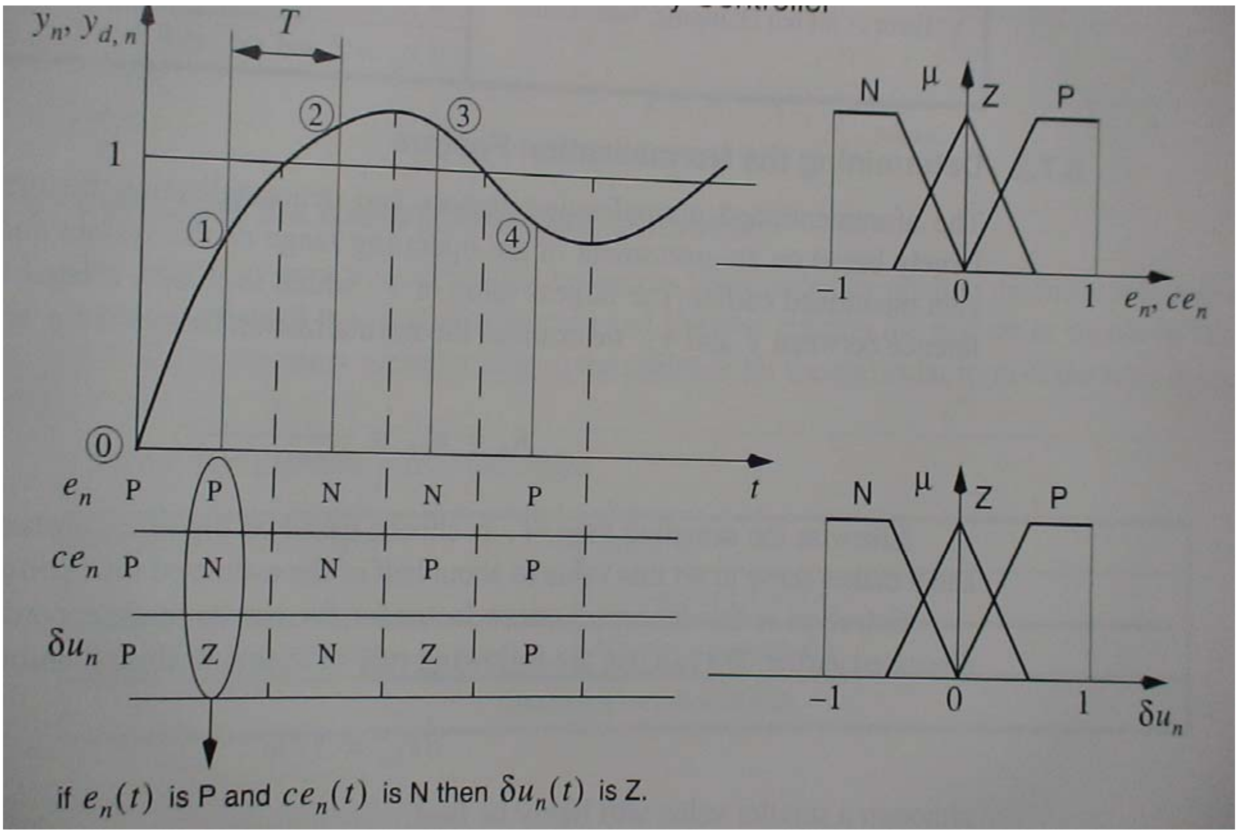
Normalized change in error: $ce_n(t) = n_{ce} \cdot (e(t) - e(t-T)),$

Normalized increment in control action: $\delta u_n(t) = \delta u(t) \cdot n_{\delta u}$

$n_e, n_{ce}$ and $n_{\delta u}$: normalization (scaling) factor

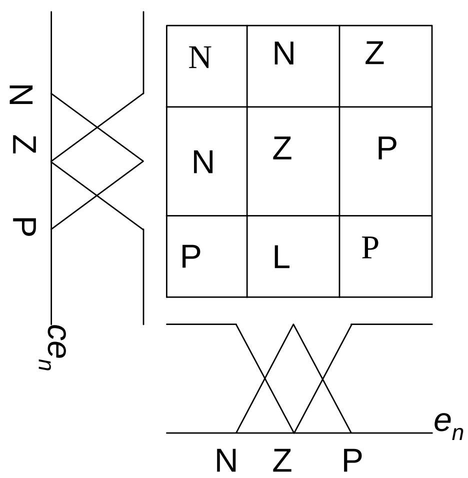
$u(t) = u(t-T) + \delta u(t)$

Fuzzy rule: IF $e_n(t)$ is P AND $ce_n(t)$ is N THEN $\delta u_n(t)$ is Z



## Rules for the Generic Fuzzy Controller

| | |
|---|---|
| Observable Attributes | error, $e_n$ |
| | Change-in-error, $ce_n$ |
| Controllable Attributes | Change-in-input, $\delta u_n$ |
| Condition→Action Rules | |
| I: Starting up, change the input in response to the setpoint change | If $e_n$ is P and $ce_n$ is P then $\delta u_n$ is P |
| | If $e_n$ is N and $ce_n$ is N then $\delta u_n$ is N |
| II: Plant is not responding; adjust input. | If $e_n$ is P and $ce_n$ is Z then $\delta u_n$ is P |
| | If $e_n$ is N and $ce_n$ is Z then $\delta u_n$ is N |
| III: Plant is responding normally, keep input the same | If $e_n$ is P and $ce_n$ is N then $\delta u_n$ is Z |
| | If $e_n$ is N and $ce_n$ is P then $\delta u_n$ is Z |
| IV: Reached equilibrium | If $e_n$ is Z and $ce_n$ is Z then $\delta u_n$ is Z |
| V: Error is nil but changing, take action | If $e_n$ is Z and $ce_n$ is N then $\delta u_n$ is N |
| | If $e_n$ is Z and $ce_n$ is P then $\delta u_n$ is P. |



• ***Implementation of fuzzy controller:***

(1) The consequent is a fuzzy singleton.

~ Rule i : IF $x_1$ is $A_{i1}$ AND … AND $x_n$ is $A_{in}$

Then $u$ is $w_i$.

$x_1, \dots, x_n$: input variables.

$u$ : control output variable.

$A_{ij}$: fuzzy set, where Gaussian membership function is used.

$$\mu_{A_{ij}}(x_j) = \exp\{-(\frac{x_j - m_{ij}}{\sigma_{ij}})^2\},$$

$m_{ij}$ : center of fuzzy set $A_{ij}$.

$\sigma_{ij}$ : width of fuzzy set $A_{ij}$.



~ The firing strength $\Phi$ of rule i is

$$\Phi_i(\vec{x}) = \prod_{j=1}^{n} \mu_{ij}(x_j) = \exp\{-\sum_{j=1}^{n}(\frac{x_j - m_{ij}}{\sigma_{ij}})^2\}$$

where the “ algebraic product” is used for fuzzy AND operation.

or $$\Phi_i(\vec{x}) = \min(\mu_{i1}(x_1), \dots, \mu_{in}(x_n))$$

where the “ min” operation is used for fuzzy AND operation.

~ The final output of the fuzzy controller consisting of r rules is :

$$u(\vec{x}) = \frac{\sum_{i=1}^{r} \Phi_i(\vec{x}) w_i}{\sum_{i=1}^{r} \Phi_i(\vec{x})}$$

Ex: Rule 1 : IF x is A1 AND y is B1 , THEN u is 4
Rule 2 : IF x is A2 AND y is B2 , THEN u is 6
Find z when inputs x0=4 and y0=8.
(1) Use product for AND operation.
(2) Use minimum for AND operation

Sol:

(1)

$$\Phi_1 = \mu_{A1}(x_0)\cdot\mu_{B1}(y_0) = \frac{2}{3}\cdot 1 = \frac{2}{3}, \qquad u = \frac{(\frac{2}{3})\cdot 4 + (\frac{2}{9})\cdot 6}{\frac{2}{3}+\frac{2}{9}} = \frac{9}{2} = 4.5$$

$$\Phi_2 = \mu_{A2}(x_0)\cdot\mu_{B2}(y_0) = \frac{1}{3}\cdot\frac{2}{3} = \frac{2}{9}$$

(2)

$$\Phi_1 = \mu_{A1}(x_0)\wedge\mu_{B1}(y_0) = \frac{2}{3}\wedge 1 = \frac{2}{3}, \qquad u = \frac{(\frac{2}{3})\cdot 4 + (\frac{1}{3})\cdot 6}{\frac{2}{3}+\frac{1}{3}} = \frac{14}{3}$$

$$\Phi_2 = \mu_{A2}(x_0)\wedge\mu_{B2}(y_0) = \frac{1}{3}\wedge\frac{2}{3} = \frac{1}{3}$$

(2) The consequent is TSK type. 

~ Rule i : IF $x_1$ is $A_{i1}$ AND … AND $x_n$ is $A_{in}$

THEN $u = a_i^0 + \sum_{k=1}^{n} a_i^k x_k = f_i(\vec{x})$

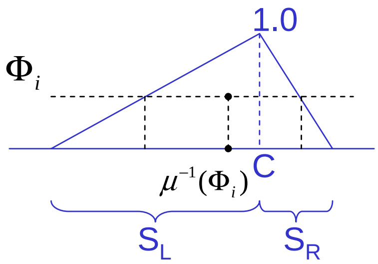
control output $$u(\vec{x}) = \frac{\sum_{i=1}^{r} \Phi_i(\vec{x})\, f_i(\vec{x})}{\sum_{i=1}^{r} \Phi_i(\vec{x})}$$

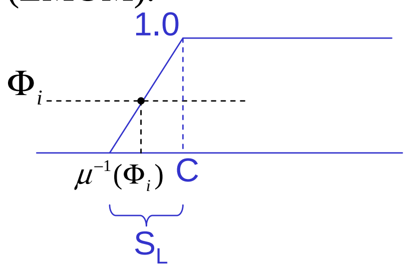
(3) The consequent is a fuzzy set.

~ Rule i : IF $x_1$ is $A_{i1}$ AND … AND $x_n$ is $A_{in}$

Then u is $B_i$. $B_i$ : a fuzzy set.

① The local mean-of-maximum method. (LMOM).



For triangular functions , LMOM gives

$$\mu^{-1}(\Phi_i) = c + \frac{1}{2}(S_R - S_L)(1 - \Phi_i).$$

$\mu^{-1}(\Phi_i)$ is the $u$-coordinate of the centroid of $B_i$

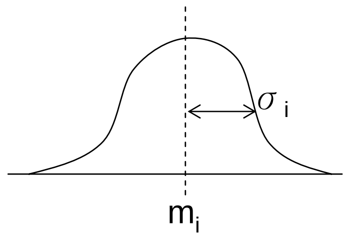
output $$u(\vec{x}) = \frac{\sum_{i=1}^{r} \Phi_i(\vec{x}) \cdot \mu^{-1}(\Phi_i(\vec{x}))}{\sum_{i=1}^{r} \Phi_i(\vec{x})}$$

② Based on center of area

IF $B_i$ is a fuzzy set , with Gaussian membership fun.

$$\mu_{B_i} = \exp\{-(\frac{x - m_i}{\sigma_i})^2\}$$

then $$u(\vec{x}) = \frac{\sum_{i=1}^{r} (m_i \sigma_i) \Phi_i(\vec{x})}{\sum_{i=1}^{r} \sigma_i \Phi_i(\vec{x})}$$

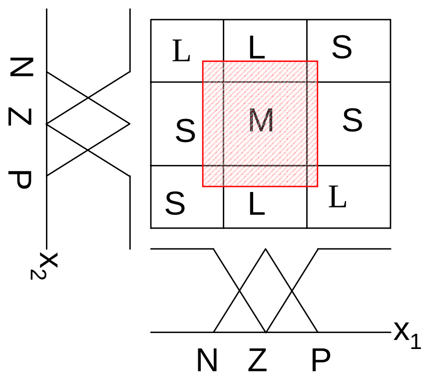





**• *Antecedent part partition:***

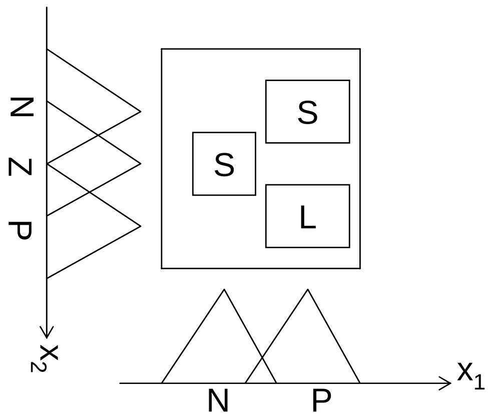
① Grid-type partition:

N: Negative
Z: Zero
P: Positive
L: Large
S: Small
M: Medium




$R_1$:
IF $x_1$ is N and $x_2$ is N
Then u is L

② Flexible partition:



**• *Design Methodology of Fuzzy Control Systems.***

~ The principal elements of designing a FLC include:

(1) defining input and output variables.

(2) deciding on the fuzzy partition of the input and output spaces and choosing the membership functions for the input and output linguistic variable.

(3) deciding on the types and the derivation of fuzzy control rules.

(4) designing the inference mechanism , which includes a fuzzy implication and a compositional operator, and the interpretation of sentence connectives AND and ALSO.

(5) choosing a defuzzification operator.

~ For (1) and (2) , there are two methods for making the choice



① We can use experience and engineering knowledge to select the possible and proper input-out variables and then use a heuristic cut-and-try procedure to find a proper fuzzy partition and a trial-and-error approach to find suitable membership functions.

② We can use learning or self-organization techniques.

~ For (4) and (5) , three is no systematic method. Most practitioners use empirical studies and results to provide guidelines for these choices.

~ For (3) the determination of fuzzy control rules depends heavily on the nature of the controlled plants. The followings are methods for the derivation of fuzzy control rules.

① Expert experience and control engineering knowledge:

~ Fuzzy control rules are designed by referring to a human operator's and/or a control engineer's knowledge.

~ More specifically, we can ask a human expert to express his or her knowledge in terms of fuzzy implications , that is , to express his know-how in fuzzy IF-THEN rules.

~ Finally , a heuristic cut-and-try procedure is used to fine-tune the fuzzy control rules.

~ Disadvantage

(a) An operator may not be able to verbalize his or her knowledge.

(b) It may be difficult for a control engineer to write down control rules because the controlled process is too complex.

② Modeling an operator's control actions:

~ We can model an operator's skilled actions or control behavior in terms of fuzzy implications using the input-output data connected with his control actions.



~ Then we can use the obtained " input-output model" as a fuzzy controller.

~ Ex: Sugeno's fuzzy car

[ Sugeno and Murakami, 1985, Sugeno and Nishida , 1985. ].

③Based on learning (or self-organizing):

~ Currently many research efforts are focused on emulating human learning mainly on the ability to create fuzzy controls rules and to modify them based on experience.

~ Examples are designs by neural networks, evolutionary algorithms, or swarm intelligence algorithms.

(4) Mathematical derivation based on plant mathematical model.

• ***Stability analysis of fuzzy control systems.***

~ Tanaka and Sugeno [1992] used Lyapunov's direct method to perform stability analysis of fuzzy control systems where the fuzzy rules are of TS type

~ Consider the following fuzzy system with zero input:

$$\begin{aligned} R^i:\ & \text{IF}\ \ x(k)\ \text{is}\ M_1^i\ \text{AND}\ \cdots\ \text{AND}\ x(k-n+1)\ \text{is}\ M_n^i\ , \\ & \text{THEN}\ \ x(k+1)=a_1^i x(k)+\ \cdots\ +a_n^i x(k-n+1) \\ & \text{where}\ \ i=1,\ 2,\ \cdots\ ,r. \end{aligned}$$

The consequent can be written as

$$\vec{x}(k+1)=A_i\vec{x}(k)\ \ ,\ \ \vec{x}(k)=[x(k),\ x(k-1),\ \cdots\ ,x(k-n+1)]^T$$



and
$$A_i=\begin{bmatrix} a_1^i & a_2^i & \cdots & a_{n-1}^i & a_n^i \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}$$

$$\vec{x}(k+1)=\frac{\sum_{i=1}^{r}\Phi_i A_i\vec{x}(k)}{\sum_{i=1}^{r}\Phi_i}\qquad\cdots\cdots\cdots(*)$$

$$\Phi_i=\mu_{M_1^i}(x_0(k))\cdot\mu_{M_2^i}(x_0(k-1))\cdots\ \mu_{M_n^i}(x_0(k-n+1)).$$

where $x_0(\bullet)$ denotes the crisp value of $x(\bullet)$ at a specific instance.

Theorem 1:

Consider a discrete system $\vec{x}(k+1)=f(\vec{x}(k))$.
where $\vec{x}(k)\in\Re^n$, $f(\vec{x}(k))$ is an $n\times 1$ function vector with the property that $f(\vec{0})=\vec{0}$ for all $k$. Suppose that there exists a scalar function $V(\vec{x}(k))$ continuous in $\vec{x}(k)$ such that

(a) $V(\vec{0}) = 0$

(b) $V(\vec{x}(k)) > 0$ for $\vec{x}(k) \neq \vec{0}$

(c) $V(\vec{x}(k))$ approaches infinity as $\|\vec{x}(k)\| \to \infty$

(d) $\Delta V(\vec{x}(k)) < 0$ for $\vec{x}(k) \neq \vec{0}$

Then the equilibrium state $\vec{x}(k) = \vec{0}$ for all $k$ is asymptotically stable in the large , and $V(\vec{x}(k))$ is a Lyapunov function.

Theorem 2:

If $P$ is a positive-definite matrix such that

$$A^T PA - P < 0 \quad and \quad B^T PB - P < 0$$

where $A,\ B,\ P \in \Re^{n\times n}$, then $A^T PB + B^T PA - 2P < 0$

pf: $A^T PB + B^T PA - 2P = -(A-B)^T P(A-B) + A^T PA + B^T PB - 2P$

$$= -(A-B)^T P(A-B) + A^T PA - P + B^T PB - P$$

$$\boxed{\begin{aligned} &-\vec{x}^T (A-B)^T P(A-B)\vec{x} \\ &= -[(A-B)\vec{x}]^T P[(A-B)\vec{x}] \\ &= -\vec{v}^T P\vec{v} \leq 0 \end{aligned}}$$

since $P$ is a positive-definite matrix.

$$-(A-B)^T P(A-B) \leq 0.$$

$\therefore$ the conclusion of the theorem follows.



Theorem 3:

[Tanaka and Sugeno, 1992] The equilibrium of a fuzzy system described by Eq.(*) is globally asymptotically stable, if $\exists$ a common positive-definite matrix $P$ for all the subsystems such that

$$A_i^T PA_i - P < 0\ , \qquad i \in \{1,\ 2, \cdots, r\}$$

pf: Choose a scalar function $V(\vec{x}(k))$ such that

$V(\vec{x}(k)) = \vec{x}^T(k)\ P\ \vec{x}(k),\ V \to \infty\ as\ \|\vec{x}(k)\| \to \infty$

where $P$ is a p.d. matrix.

$$\Delta V(\vec{x}(k)) = V(\vec{x}(k+1)) - V(\vec{x}(k))$$

$$= \vec{x}^T(k+1)\ P\ \vec{x}(k+1) - \vec{x}^T(k)\ P\ \vec{x}(k)$$

$$= \left(\frac{\sum_{i=1}^{r} \Phi_i A_i \vec{x}(k)}{\sum_{i=1}^{r} \Phi_i}\right)^T P \left(\frac{\sum_{i=1}^{r} \Phi_i A_i \vec{x}(k)}{\sum_{i=1}^{r} \Phi_i}\right) - \vec{x}^T(k)\ P\ \vec{x}(k)$$

$$= \vec{x}^T(k)[(\frac{\sum_{i=1}^{r}\Phi_i A_i^T}{\sum_{i=1}^{r}\Phi_i})\,P\,(\frac{\sum_{i=1}^{r}\Phi_i A_i}{\sum_{i=1}^{r}\Phi_i}) - P]\,\vec{x}(k)$$

$$= \frac{\sum_{i,j=1}^{r}\Phi_i\Phi_j\vec{x}^T(k)(A_i^T P A_j - P)\vec{x}(k)}{\sum_{i,j=1}^{r}\Phi_i\Phi_j}$$

$$= \frac{\sum_{i=1}^{r}(\Phi_i)^2\vec{x}^T(k)\{A_i^T P A_i - P\}\vec{x}(k) + \sum_{i<j}^{r}\Phi_i\Phi_j\vec{x}^T(k)\{A_i^T P A_j + A_j^T P A_i - 2P\}\vec{x}(k)}{\sum_{i,j=1}^{r}\Phi_i\Phi_j} < 0$$

where $\Phi_i \geq 0$ , and $\sum_{i=1}^{r}\Phi_i > 0$

$\therefore$ $V(\vec{x}(k))$ is a Lyapunov function and the fuzzy system is globally asymptotically stable.



•Parallel distributed compensation (PDC)

~A TS fuzzy controller can be designed using TS-fuzzy model by using the antecedent part of the TS fuzzy model as that of the TS fuzzy contrtoller

~In this case, we can use a proper linear control method for each pair of plant and controller rules. This design approach is called *PDC*.

Ref: *K. Tanaka and H. O. Wang, Fuzzy control systems design and analysis: a linear matrix inequality approach, New York, Wiley,* 2001.

~Discrete-time TS fuzzy model

$\mathrm{R}^i$ : IF $x(k)$ is $M_1^i$ AND $\cdots$ AND $x(k-n+1)$ is $M_n^i$ ,

THEN $\vec{x}(k+1) = A_i\,\vec{x}(k) + B_i\vec{u}(k),$ $\quad i = 1,\ 2,\ \cdots\ ,\ r.$

The output is

$$\vec{x}(k+1) = \frac{\sum_{i=1}^{r}\Phi_i[A_i\vec{x}(k) + B_i\vec{u}(k)]}{\sum_{i=1}^{r}\Phi_i} \qquad (1)$$

~ A PDC-type TS fuzzy controller which uses full state feedback is

Control rule *i*: IF $x(k)$ is $M_1^i$ AND $\cdots$ AND $x(k-n+1)$ is $M_n^i$ ,

THEN $\vec{u}(k) = K_i\,\vec{x}(k),$ $\quad i = 1, \ldots, r$

Output of the TS fuzzy controller 

$$\vec{u}(k) = \frac{\sum_{i=1}^{r} \Phi_i K_i \vec{x}(k)}{\sum_{i=1}^{r} \Phi_i} \qquad (2)$$

By substituting Eq. (2) to Eq. (1), we have the controlled output of the plant

$$\vec{x}(k+1) = \frac{\sum_{i=1}^{r}\sum_{j=1}^{r} \Phi_i \Phi_j (A_i + B_i K_j)\vec{x}(k)}{\sum_{i=1}^{r}\sum_{j=1}^{r} \Phi_i \Phi_j} \qquad (3)$$

*Theorem 4: Stability condition.*

The equilibrium of the discrete-time TS fuzzy control system (i.e. $\vec{x} = \vec{0}$) is globally asymptotically stable if there exists a common positive definite matrix $P$ such that

$$(A_i + B_i K_j)^T P (A_i + B_i K_j) - P < 0, \quad \text{for all } i, j = 1, 2, \ldots, r. \qquad (4)$$

Remark: When the $K_i$'s, $i = 1, 2, \ldots, r$, are pre-determined, it is suggested that $P$ can be determined numerically by solving the Linear Matrix Inequalities (LMIs).

~ It is noted that Eq. (4) has $r^2$ LMIs. By grouping the same terms in Eq. (3), we have 

$$\vec{x}(k+1) = \frac{\sum_{i=1}^{r} \Phi_i^2 (A_i + B_i K_i)\vec{x}(k) + 2\sum_{i<j}^{r} \Phi_i \Phi_j G_{ij} \vec{x}(k)}{\sum_{i=1}^{r}\sum_{j=1}^{r} \Phi_i \Phi_j} \qquad (5)$$

where

$$G_{ij} = \frac{(A_i + B_i K_j) + (A_j + B_j K_i)}{2}, \quad i < j \le r,$$

This yields a less conservative stability condition.

*Theorem 5: Less conservative stability condition*

The equilibrium state of the discrete-time TS fuzzy control system in Eq. (5) (namely, $\vec{x} = \vec{0}$) is globally asymptotically stable if there exists a common symmetric positive definite matrix $P$ such that

$$(A_i + B_i K_i)^T P (A_i + B_i K_i) - P < 0, \; i = 1, 2, \ldots, r, \qquad (6)$$

$$G_{ij}^T P G_{ij} - P < 0, \quad i < j \le r$$

Remarks:

1. The number of LMIs for Eq. (6) is $r(r+1)/2$.

   Eq. (6) also has the advantage of the relation of the stability criterion of Eq. (4). Some standard feasibility problems that are infeasible from Eq. (4) can be solved from Eq. (4).

2. The sufficient condition for the stability of Eq. (6) can be used only for the purpose of checking of the stability of the TS fuzzy control system in which the feedback gains, $K_i$'$s, i=1,....r,$ are pre-determined by a proper controller design method.



Example:

Consider the following fuzzy model:

$R^1$ : IF $x(k)$ is $M^1$, THEN $x^1(k+1) = 2.178x(k) - 0.588x(k-1) + 0.603u(k)$

$R^2$ : IF $x(k)$ is $M^2$, THEN $x^2(k+1) = 2.256x(k) - 0.361x(k-1) + 1.120u(k)$

$$\Rightarrow \quad x(k+1) = \frac{\Phi_1 x^1(k+1) + \Phi_2 x^2(k+1)}{\Phi_1 + \Phi_2} \qquad \Phi_1, \Phi_2 \text{ : firing strength.}$$

FLC:

R[1]: IF $x(k)$ is $M^1$, THEN $u^1(k) = -2.109x(k) + 0.475x(k-1)$

R[2]: IF $x(k)$ is $M^2$, THEN $u^2(k) = -1.205x(k) + 0.053x(k-1)$

$$\Rightarrow \quad u(k) = \frac{\Phi_1 u^1(k) + \Phi_2 u^2(k)}{\Phi_1 + \Phi_2}$$

$$\Rightarrow \quad x^1(k+1) = \frac{0.906\Phi_1 + 1.451\Phi_2}{\Phi_1 + \Phi_2} x(k) + \frac{-0.302\Phi_1 - 0.556\Phi_2}{\Phi_1 + \Phi_2} x(k-1)$$

$$x^2(k+1) = \frac{-0.106\Phi_1 + 0.906\Phi_2}{\Phi_1 + \Phi_2} x(k) + \frac{0.171\Phi_1 - 0.302\Phi_2}{\Phi_1 + \Phi_2} x(k-1)$$

Method 1: by Theorem 3



Hence ,

$$x(k+1)=\frac{\Phi_1 x^1(k+1)+\Phi_2 x^2(k+1)}{\Phi_1+\Phi_2}$$

$$=\frac{0.906\Phi_1^2+1.345\Phi_1\Phi_2+0.906\Phi_2^2}{(\Phi_1+\Phi_2)^2}x(k)-\frac{0.302\Phi_1^2+0.385\Phi_1\Phi_2+0.302\Phi_2^2}{(\Phi_1+\Phi_2)^2}x(k-1)$$

$$=\frac{[0.906\Phi_1^2 x(k)-0.302\Phi_1^2 x(k-1)]+[1.345\Phi_1\Phi_2 x(k)-0.385\Phi_1\Phi_2 x(k-1)]+[0.906\Phi_2^2 x(k)-0.302\Phi_2^2 x(k-1)]}{\Phi_1^2+2\Phi_1\Phi_2+\Phi_2^2}$$

$$\Rightarrow \quad \left.\begin{array}{l} S^{11}: \text{IF } x(k) \text{ is } (M^1 \text{ AND } M^1), \text{ THEN } \vec{x}^{11}(k+1)=A_{11}\vec{x}(k) \\ 2\times \quad S^{12}: \text{IF } x(k) \text{ is } (M^1 \text{ AND } M^2), \text{ THEN } \vec{x}^{12}(k+1)=A_{12}\vec{x}(k) \\ S^{22}: \text{IF } x(k) \text{ is } (M^2 \text{ AND } M^2), \text{ THEN } \vec{x}^{22}(k+1)=A_{22}\vec{x}(k) \end{array}\right\}(**)$$

where

$$A_{11}=\begin{bmatrix}0.906 & -0.302\\ 1 & 0\end{bmatrix},\ A_{12}=\begin{bmatrix}0.672 & -0.193\\ 1 & 0\end{bmatrix},\ A_{22}=\begin{bmatrix}0.906 & -0.302\\ 1 & 0\end{bmatrix}$$

Let $\quad P=\begin{bmatrix}4.19 & -0.88\\ -0.88 & 1.38\end{bmatrix}$



Then we can verify

$$A_{11}^T P A_{11}-P<0\ ,\ A_{12}^T P A_{12}-P<0\ ,\ A_{22}^T P A_{22}-P<0.$$

From Theorem 3, the fuzzy control system is globally asymptotiaclly stable.

Method 2: (By Theorem 5)

$$A_1=\begin{bmatrix}2.178 & -0.588\\ 1 & 0\end{bmatrix},\ B_1=\begin{bmatrix}0.603\\ 0\end{bmatrix},\ K_1=\begin{bmatrix}-2.109 & 0.475\end{bmatrix}$$

$$A_2=\begin{bmatrix}2.256 & -0.361\\ 1 & 0\end{bmatrix},\ B_2=\begin{bmatrix}1.120\\ 0\end{bmatrix},\ K_2=\begin{bmatrix}-1.205 & 0.053\end{bmatrix}$$

$$A_1+B_1K_1=\begin{bmatrix}0.9064 & -0.302\\ 1 & 0\end{bmatrix},A_2+B_2K_2=\begin{bmatrix}0.9063 & -0.302\\ 1 & 0\end{bmatrix}$$

$$G_{12}=\frac{(A_1+B_1K_2)+(A_2+B_2K_1)}{2}=\begin{bmatrix}0.6727 & -0.1925\\ 1 & 0\end{bmatrix}$$

Let $P=\begin{bmatrix}4.19 & -0.88\\ -0.88 & 1.38\end{bmatrix}$, Theorem 5 is satisfied.

***Remarks:*** *(with u(k) absent in consequent)*

1. All the $A_i$ matrices are stable matrices if $\exists$ a common positive-definite matrix $P$. However, a common positive-definite matrix $P$ obviously does not always exist even if all the $A_i$ matrices are stable matrices.
2. A fuzzy system may be globally asymptotically stable even if there does not exist a common positive-definite matrix $P$.
3. A fuzzy system is not always globally asymptotically stable even if all the $A_i$ matrices are stable matrices.
4. In the following theorem , a necessary condition for ensuring the existence of a common positive-definite matrix $P$ is given.



Theorem 6: [Tanaka and Sugeno , 1992]

Assume that $A_i$ is stable , nonsingular matrix for $i=1, 2, \ldots ,r$. $A_iA_j$ is a stable matrix for $i , j=1, 2, \ldots ,r$ if $\exists$ a common positive-definite matrix $P$ such that

$$A_i^T P A_i - P < 0. \qquad \cdots(*)$$

pf: from (*) , we have

$$P - (A_i^{-1})^T P A_i^{-1} < 0, \quad \text{since} \quad (A_i^{-1})^T = (A_i^T)^{-1}.$$

Therefore, $P < (A_i^{-1})^T P (A_i^{-1}), \quad \text{for } i = 1,\ 2,\ \cdots, r.$

from (*) $A_i^T P A_i < P$

$$\Rightarrow \qquad A_i^T P A_i < (A_j^{-1})^T P (A_j^{-1}) , \qquad i, j = 1,\ 2,\ \cdots ,\ r.$$

$$\Rightarrow \qquad A_j^T A_i^T P A_i A_j - P < 0$$

$$\Rightarrow \qquad A_i A_j \ \text{ must be a stable matrix for } i\ ,\ j = 1,\ 2,\ \cdots ,r.$$

Remark:

Theorem 6 shows that if one of the $A_iA_j$ matrices is not a stable matrix , then $\nexists$ a common positive-definite matrix $P$.

**• *Parameter region (PR):***

~ The PR representation graphically shows locations of fuzzy IF-THEN rules in consequent parameter space.

Example:

Consider the following fuzzy system ( fuzzy system 1 ):

Rule 1 : IF $x(k)$ is $C_1$ , THEN $x_1(k+1) = 0.1x(k) + 0.1x(k-1)$

Rule 2 : IF $x(k)$ is $C_2$ , THEN $x_2(k+1) = 0.3x(k) + 0.1x(k-1)$

Rule 3 : IF $x(k)$ is $C_3$ , THEN $x_3(k+1) = 0.1x(k) + 0.3x(k-1)$

$$\Rightarrow \quad A_1 = \begin{bmatrix} 0.1 & 0.1 \\ 1 & 0 \end{bmatrix} \quad A_2 = \begin{bmatrix} 0.3 & 0.1 \\ 1 & 0 \end{bmatrix} \quad A_3 = \begin{bmatrix} 0.1 & 0.3 \\ 1 & 0 \end{bmatrix}$$

Consider another fuzzy system ( fuzzy system 2 ):

Rule 1 : IF $x(k)$ is $B_1$ , THEN $x_1(k+1) = 0.1x(k) + 0.1x(k-1)$

Rule 2 : IF $x(k)$ is $B_2$ , THEN $x_2(k+1) = 0.3x(k) + 0.1x(k-1)$

Rule 3 : IF $x(k)$ is $B_3$ , THEN $x_3(k+1) = 0.1x(k) + 0.3x(k-1)$

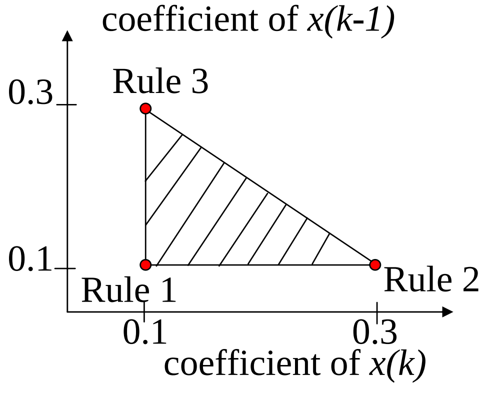


Rule 4 : IF $x(k)$ is $B_4$ , THEN $x_4(k+1) = 0.2x(k) + 0.2x(k-1)$

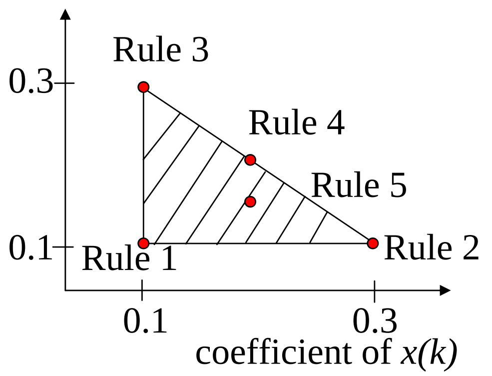
Rule 5 : IF $x(k)$ is $B_5$ , THEN $x_5(k+1) = 0.1x(k) + 0.15x(k-1)$

$$\Rightarrow \quad A_1 = \begin{bmatrix} 0.1 & 0.1 \\ 1 & 0 \end{bmatrix} \quad A_2 = \begin{bmatrix} 0.3 & 0.1 \\ 1 & 0 \end{bmatrix} \quad A_3 = \begin{bmatrix} 0.1 & 0.3 \\ 1 & 0 \end{bmatrix}$$

$$A_4 = \begin{bmatrix} 0.2 & 0.2 \\ 1 & 0 \end{bmatrix} \quad A_5 = \begin{bmatrix} 0.15 & 0.15 \\ 1 & 0 \end{bmatrix}$$




(a) PR of fuzzy system 1.




(b) PR of fuzzy system 2.

~ in fuzzy system 1 , each plotted point corresponds to each edge of the parameter region.

~ in fuzzy system 2 , the PR constructed using the plotted points of rules 1-3 includes plotted points of rules 4 and 5.

~ Rules 1~3 of fuzzy system 1 or fuzzy system 2 are edges of the PR , they are said to be edge rules. The consequent matrices $A_1$ , $A_2$ and $A_3$ in edge rules are said to be edge matrices.

~ A fuzzy system that consists of only of edge rules is said to be a minimum representation.

~ Obviously , fuzzy system 1 is a minimum representation , while fuzzy system 2 is not.

~ The following theorem is important for checking stability in the case of non-minimum representation.



Theorem 5: [ Tanaka and Sano , 1993 ]

Assume that $P$ is a positive-definite matrix. If $A_i^T P A_i - P < 0$ for $i = 1,\ 2,\ \cdots\ ,\ r$, then $A^{*T} P A^* - P < 0$, where $A^*$ is nonedge matrix such that

$$A^* = \sum_{i=1}^{r} s_i A_i, \quad \text{where} \quad \sum_{i=1}^{r} s_i = 1 \ \text{ and } \ s_i \geq 0.$$

Remark:

1. The above theorem indicates that the stability of a fuzzy system can be checked by applying the Tanaka-Sugeno theorem (Thm. 3) to a minimum representation of fuzzy system.
2. In fuzzy system 2 of the above example ,

   $A_4=0.5A_2+0.5A_3$ and $A_5=0.5A_1+0.25A_2+0.25A_3$. Therefore , a minimum representation of fuzzy system 2 is equivalent to fuzzy system 1. Hence , it's found from Thm. 5 that fuzzy system 2 is stable if fuzzy system 1 is stable.