Fuzzy Logic Control

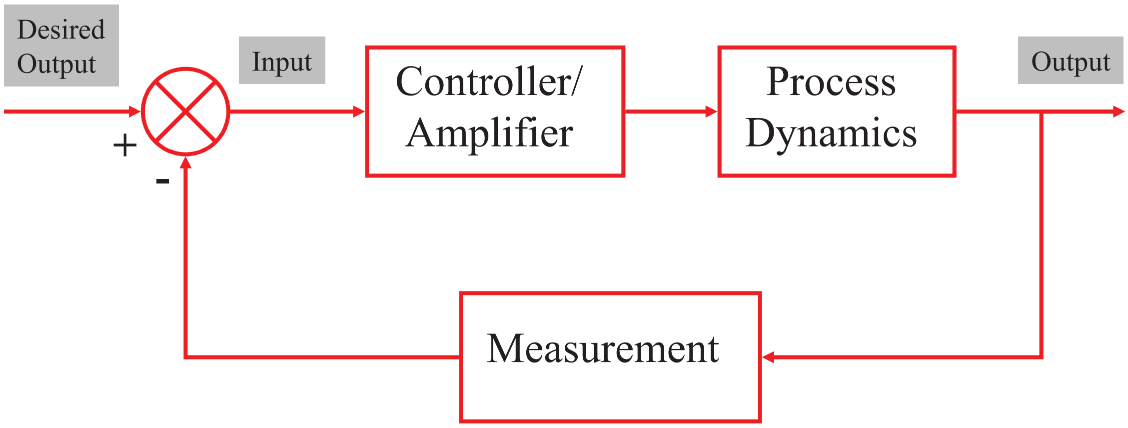
Lect 5 Fuzzy Logic Control

Content

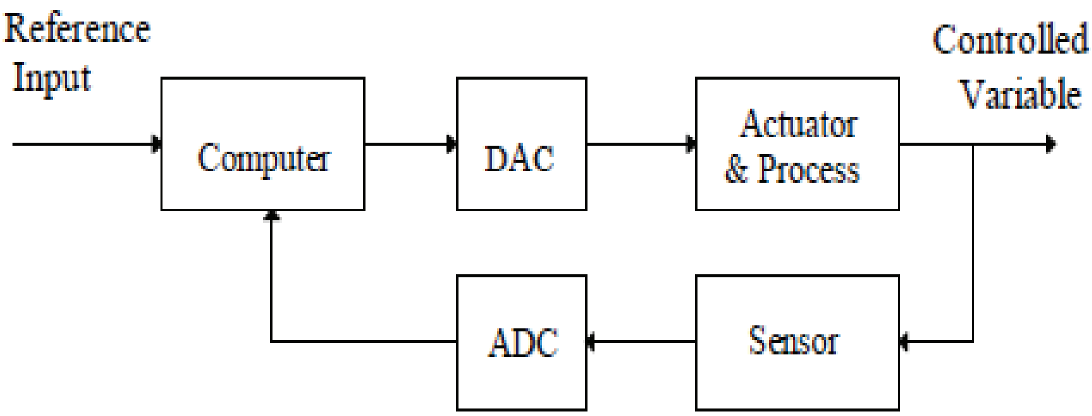
- Classical Control
- Fuzzy Logic Control
- The Architecture of Fuzzy Inference Systems
- Fuzzy Control Model
  - Mamdani Fuzzy models
  - Larsen Fuzzy Models
  - Sugeno Fuzzy Models
  - Tsukamoto Fuzzy models
- Examples

CONVENTIONAL CONTROL

- **Open-loop** control is 'blind' to actual output
- **Closed-loop** control takes account of actual output and compares this to desired output



Digital Control System Configuration



CONVENTIONAL CONTROL

Example: design a cruise control system

After gaining an intuitive understanding of the plant's dynamics and establishing the design objectives, the control engineer typically solves the cruise control problem by doing the following:

1. Developing a model of the automobile dynamics (which may model vehicle and power train dynamics and tire, the effect of road grade variations, etc.).

2. Using the mathematical model, or a simplified version of it, to design a controller (e.g., via a linear model, develop a linear controller with techniques from classical control).

CONVENTIONAL CONTROL

3. Using the mathematical model of the closed-loop system and mathematical or simulation-based analysis to study its performance (possibly leading to redesign).

4. Implementing the controller via, for example, a microprocessor, and evaluating the performance of the closed-loop system (again, possibly leading to redesign).

CONVENTIONAL CONTROL

Mathematical model of the plant:

– never perfect

– an abstraction of the real system

– "is accurate enough to be able to design a controller that will work."!

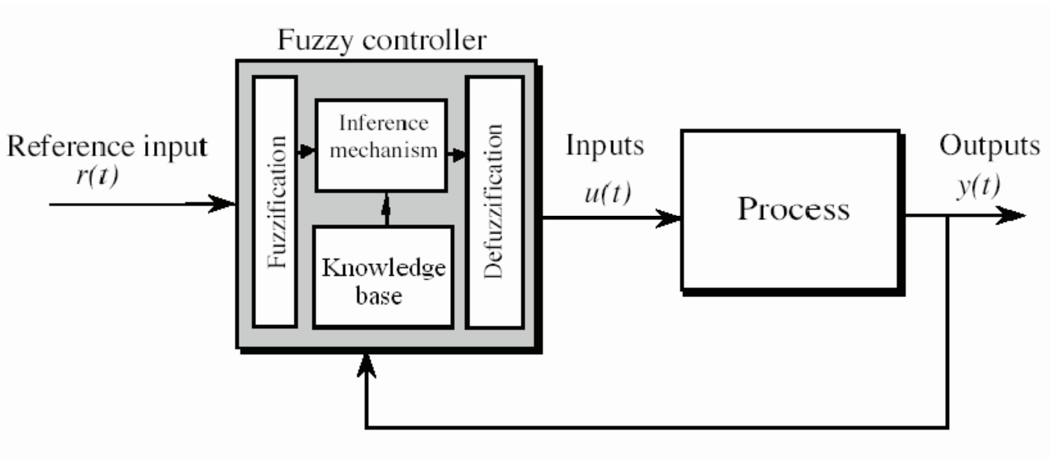
– based on a system of differential equations

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

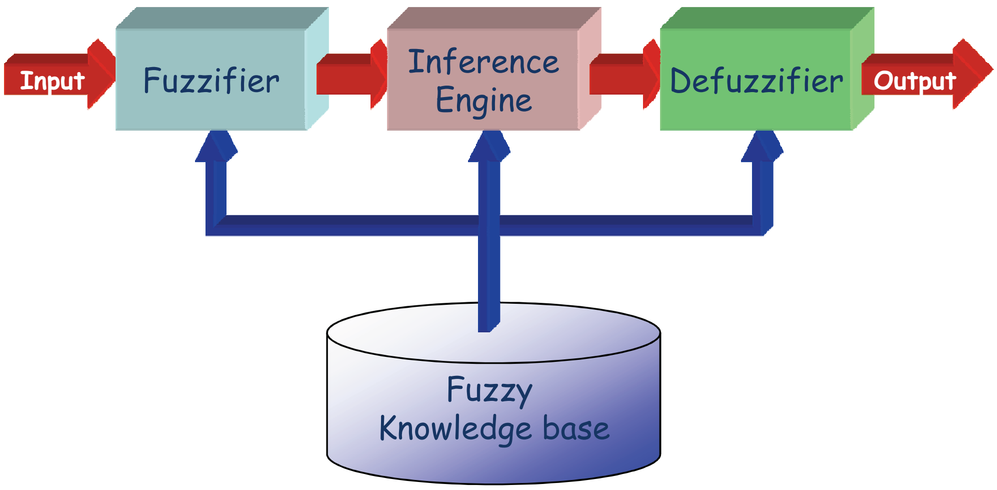
In this case u is the m-dimensional input; x is the n-dimensional state ($\dot{x} = \frac{dx(t)}{dt}$); y is the p dimensional output; and A, B, C, and D are matrices of appropriate dimension.

Fuzzy Control

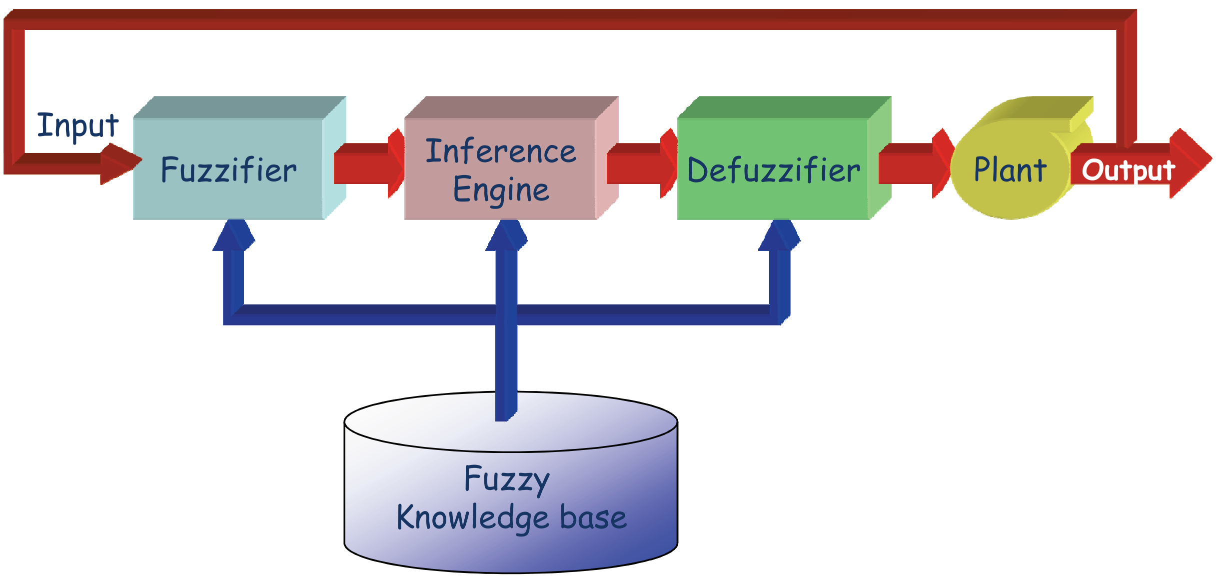
Fuzzy control provides a formal methodology for representing, manipulating, and implementing a human's heuristic knowledge about how to control a system.



Fuzzy Systems



Fuzzy Control Systems



Fuzzy Logic Control

- **Fuzzy controller design** consist of turning **intuitions**, and any **other information** about how to control a system, into set of rules.
- These rules can then be applied to the system.
- If the rules **adequately control the system**, the design work is done.
- If the rules are inadequate, **the way they fail** provides information to change the rules.

Components of Fuzzy system

- The components of a conventional expert system and a fuzzy system are the same.
- Fuzzy systems *though* contain `fuzzifiers'.
  - Fuzzifiers convert crisp numbers into fuzzy numbers,
- Fuzzy systems contain `defuzzifiers',
  - Defuzzifiers convert fuzzy numbers into crisp numbers.

In order to process the input to get the output reasoning there are six steps involved in the creation of a rule based fuzzy system:

1. Identify the inputs and their ranges and name them.
2. Identify the outputs and their ranges and name them.
3. Create the degree of fuzzy membership function for each input and output.
4. Construct the rule base that the system will operate under
5. Decide how the action will be executed by assigning strengths to the rules
6. Combine the rules and defuzzify the output

Fuzzy Logic Control

Type of Fuzzy Controllers:

- Mamdani
- Larsen
- TSK (Takagi Sugeno Kang)
- Tsukamoto
- Other methods

Fuzzy Control Systems

Mamdani
Fuzzy models

Mamdani Fuzzy models

- The most commonly used fuzzy inference technique is the so-called Mamdani method.

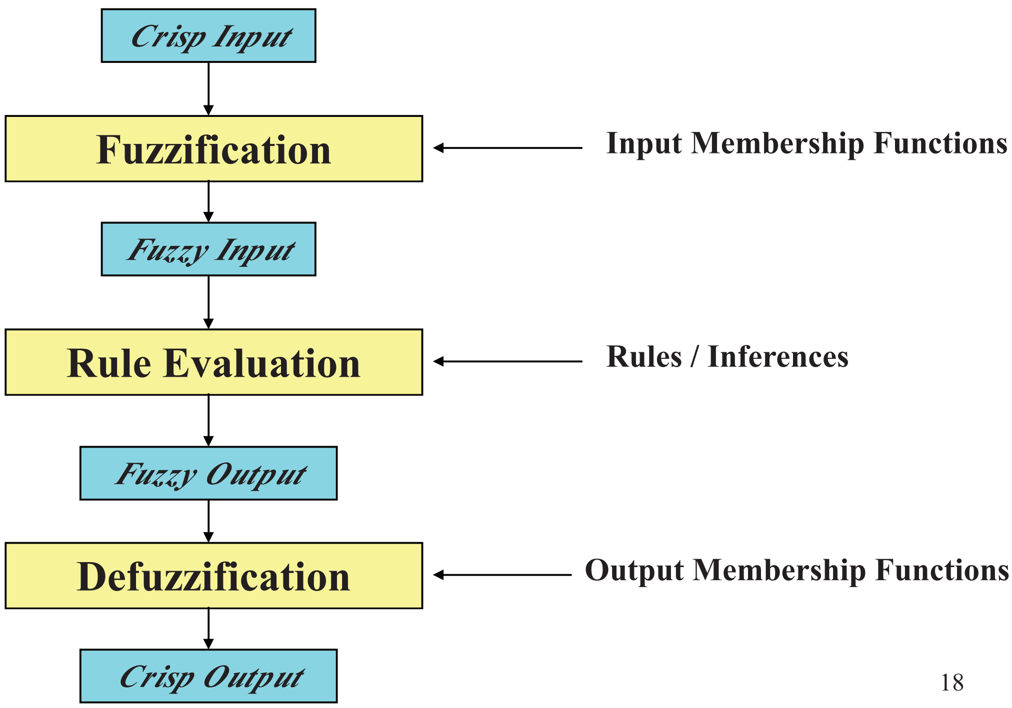
- In 1975, Professor **Ebrahim Mamdani** of London University built one of the first fuzzy systems to control a steam engine and boiler combination.
- Original Goal: Control a steam engine & boiler combination by a set of linguistic control rules obtained from experienced human operators.

Mamdani fuzzy inference

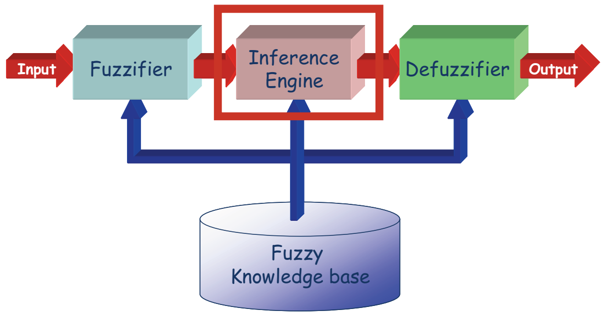
The Mamdani-style fuzzy inference process is performed in four steps:

1. Fuzzification of the input variables,
2. Rule evaluation;
3. Aggregation of the rule outputs, and finally
4. Defuzzification.

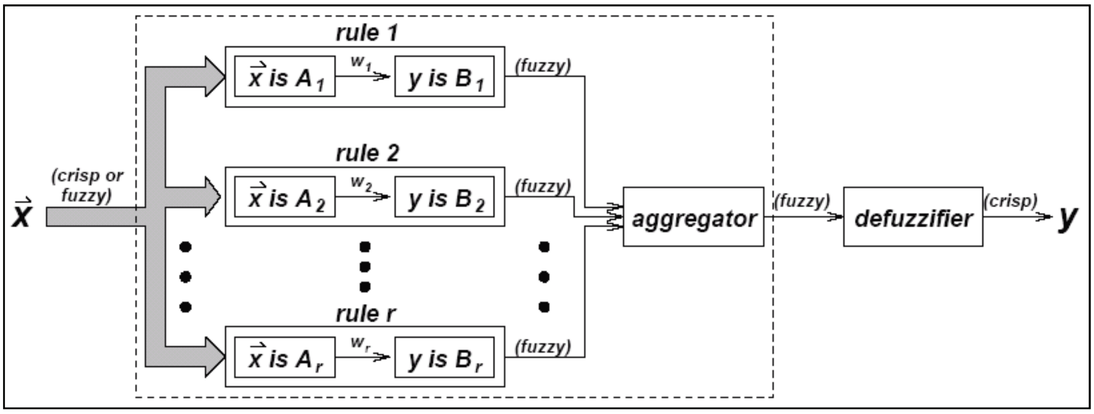
Operation of Fuzzy System



Inference Engine



Using If-Then type fuzzy rules converts the fuzzy input to the fuzzy output.



We examine a simple two-input one-output problem that includes three rules:

Rule: 1
IF x is $A3$
OR y is $B1$
THEN z is $C1$

Rule: 2
IF x is $A2$
AND y is $B2$
THEN z is $C2$

Rule: 3
IF x is $A1$
THEN z is $C3$

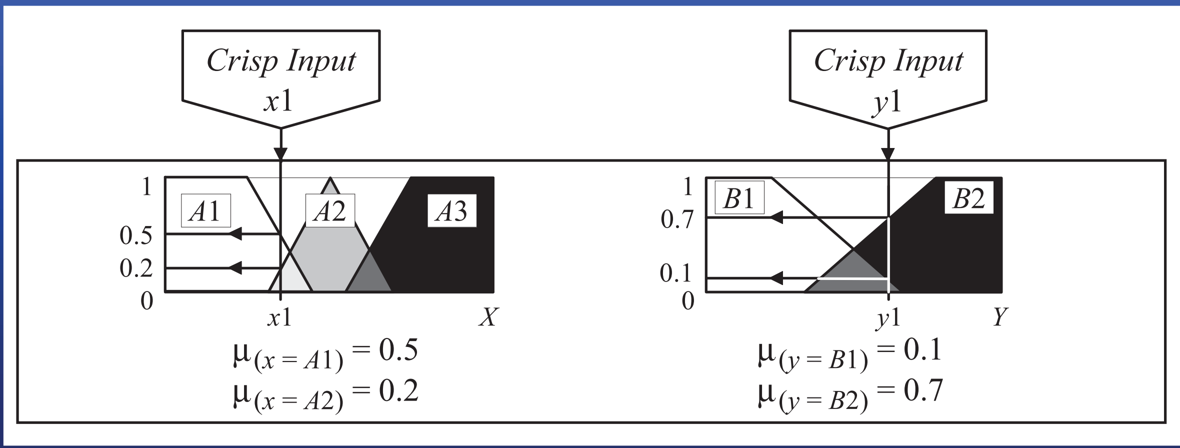
Rule: 1
IF *project_funding* is *adequate*
OR *project_staffing* is *small*
THEN *risk* is *low*

Rule: 2
IF *project_funding* is *marginal*
AND *project_staffing* is *large*
THEN *risk* is *normal*

Rule: 3
IF *project_funding* is *inadequate*
THEN *risk* is *high*

Step 1: Fuzzification

- Take the crisp inputs, $x1$ and $y1$ (*project funding* and *project staffing*)
- Determine the degree to which these inputs belong to each of the appropriate fuzzy sets.



project funding *project staffing*

Step 2: Rule Evaluation

- take the fuzzified inputs, $\mu_{(x=A1)} = 0.5$, $\mu_{(x=A2)} = 0.2$, $\mu_{(y=B1)} = 0.1$ and $\mu_{(y=B2)} = 0.7$
- apply them to the antecedents of the fuzzy rules.
- If a given fuzzy rule has multiple antecedents, the fuzzy operator (AND or OR) is used to obtain a single number that represents the result of the antecedent evaluation. This number (the truth value) is then applied to the consequent membership function.

Step 2: Rule Evaluation

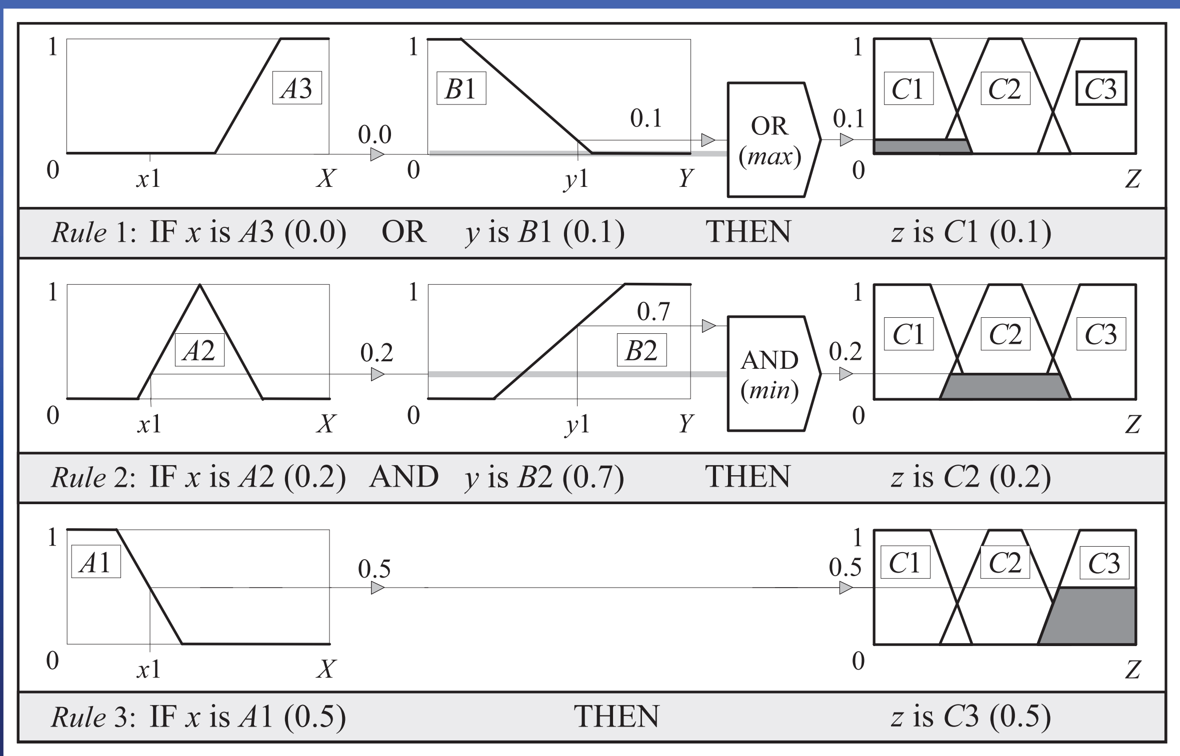
To evaluate the disjunction of the rule antecedents, we use the **OR fuzzy operation**. Typically, fuzzy expert systems make use of the classical fuzzy operation **union**:

$$\mu_{A \cup B}(x) = max\,[\mu_A(x),\ \mu_B(x)]$$

Similarly, in order to evaluate the conjunction of the rule antecedents, we apply the **AND fuzzy operation intersection**:

$$\mu_{A \cap B}(x) = min\,[\mu_A(x),\ \mu_B(x)]$$

Mamdani-style rule evaluation



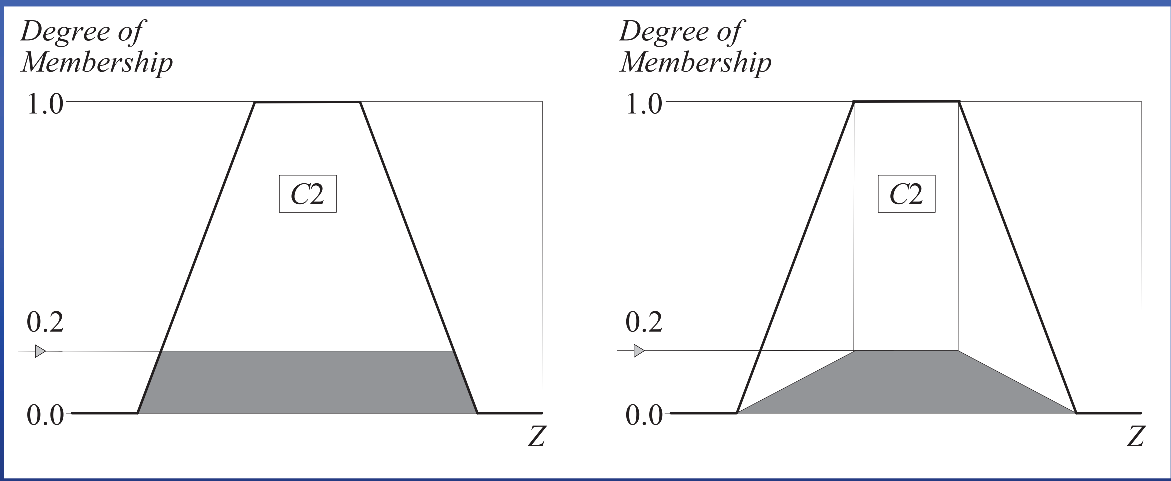
- Now the result of the antecedent evaluation can be applied to the membership function of the consequent.
- The most common method is to cut the consequent membership function at the level of the antecedent truth.
- This method is called **clipping (**Max-Min Composition) .
  - The clipped fuzzy set loses some information.
  - Clipping is still often preferred because:
    - it involves less complex and faster mathematics
    - it generates an aggregated output surface that is easier to defuzzify.

- While clipping is a frequently used method, **scaling** (Max-Product Composition) offers a better approach for preserving the original shape of the fuzzy set.

- The original membership function of the rule consequent is adjusted by multiplying all its membership degrees by the truth value of the rule antecedent.

- This method, which generally loses less information, can be very useful in fuzzy expert systems.

Clipped and scaled membership functions



Max-Min Composition

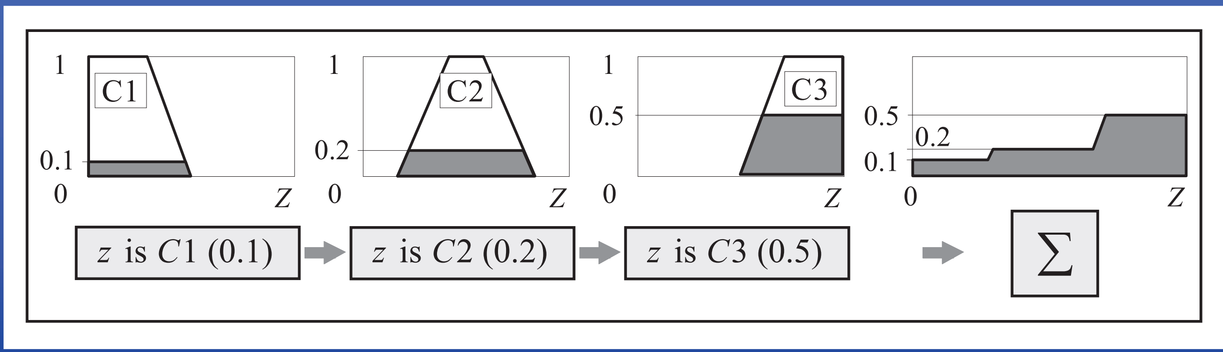
Max-Product Composition

Step 3: Aggregation of The Rule Outputs

- Aggregation is the process of unification of the outputs of all rules.

- We take the membership functions of all rule consequents previously clipped or scaled and combine them into a single fuzzy set.

Aggregation of the rule outputs



Step 4: Defuzzification

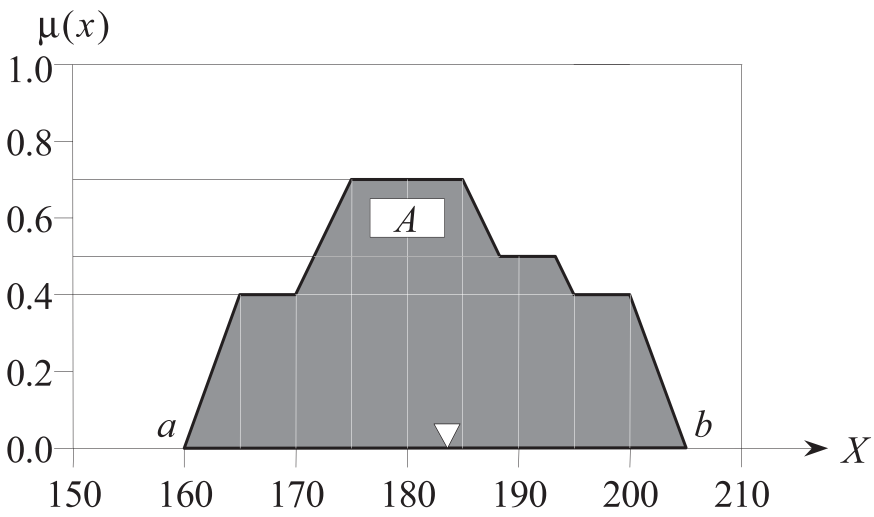
- Fuzziness helps us to evaluate the rules, but the final output of a fuzzy system has to be a crisp number.

- The input for the defuzzification process is the aggregated output fuzzy set and the output is a single number.

- There are several defuzzification methods, but probably the most popular one is the **centroid technique**.
- It finds the point where a vertical line would slice the aggregate set into two equal masses. Mathematically this **centre of gravity (COG)** can be expressed as:

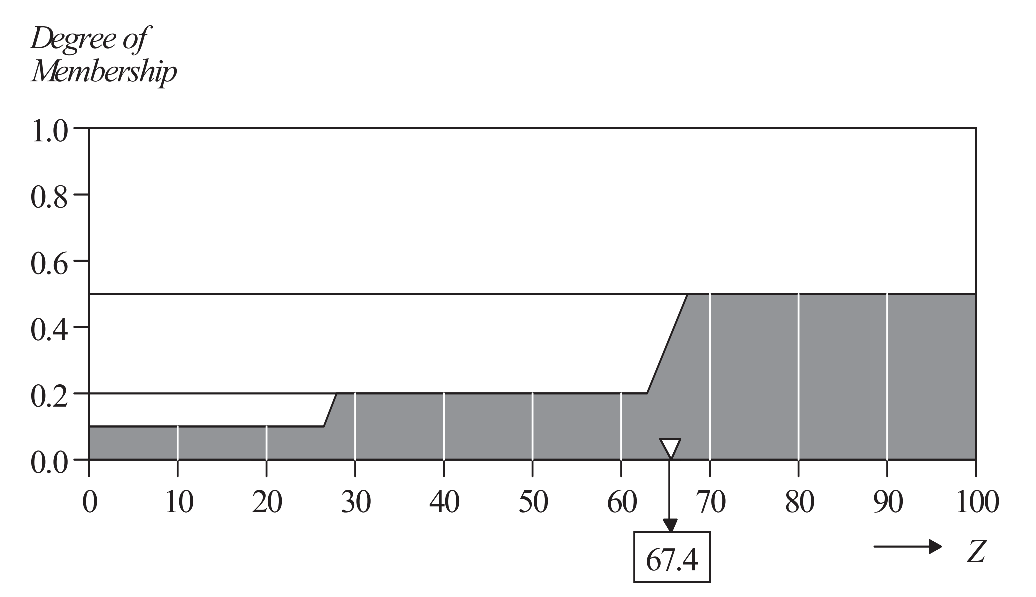
$$COG = \frac{\int_a^b \mu_A(x)\, x\, dx}{\int_a^b \mu_A(x)\, dx}$$

- Centroid defuzzification method finds a point representing the centre of gravity of the fuzzy set, *A*, on the interval, *ab*.
- ***A reasonable estimate*** can be obtained by calculating it over a sample of points.



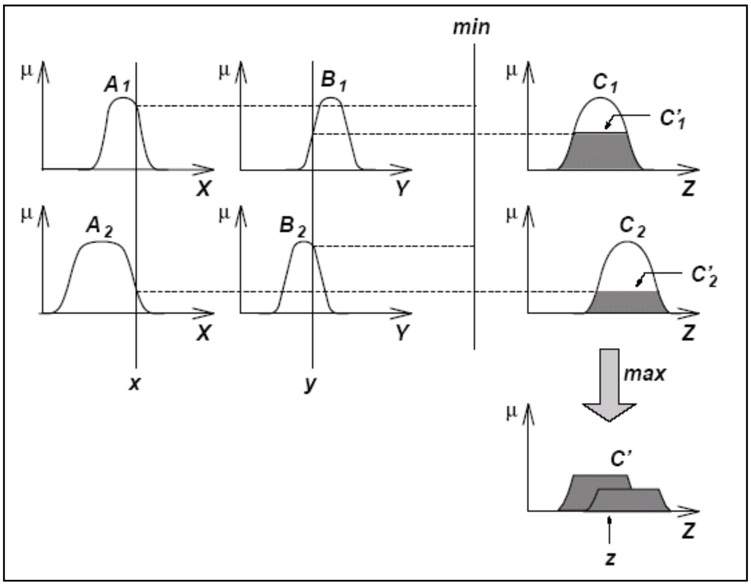
Centre of gravity (COG):

$$COG = \frac{(0+10+20)\times 0.1 + (30+40+50+60)\times 0.2 + (70+80+90+100)\times 0.5}{0.1+0.1+0.1+0.2+0.2+0.2+0.2+0.5+0.5+0.5+0.5} = 67.4$$



Max-Min Composition is used.

The Reasoning Scheme



Examples for Mamdani Fuzzy Models

Example #1

Single input single output Mamdani fuzzy model with 3 rules:

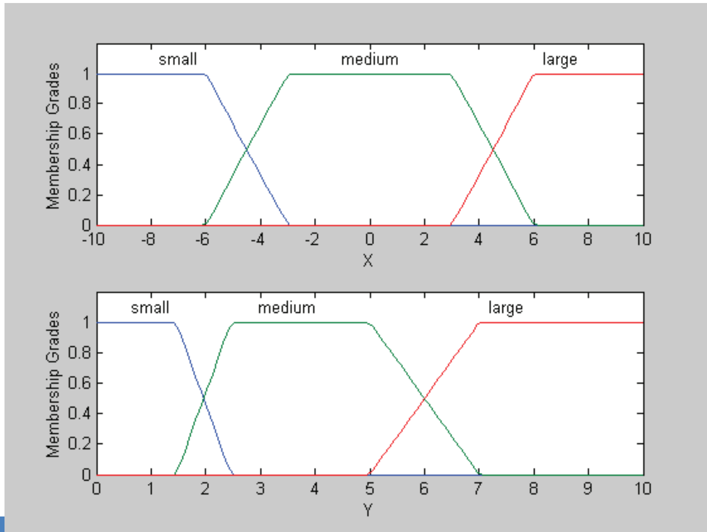
If X is small then Y is small $\rightarrow$ R_1

If X is medium then Y is medium $\rightarrow$ R_2

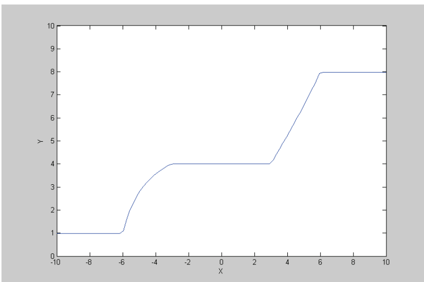
Is X is large then Y is large $\rightarrow$ R_3

X = input $\in$ [-10, 10] Y = output $\in$ [0,10]

Using centroid defuzzification, we obtain the following overall input-output curve



Single input single output antecedent & consequent MFs



Example #2 (Mamdani Fuzzy models)

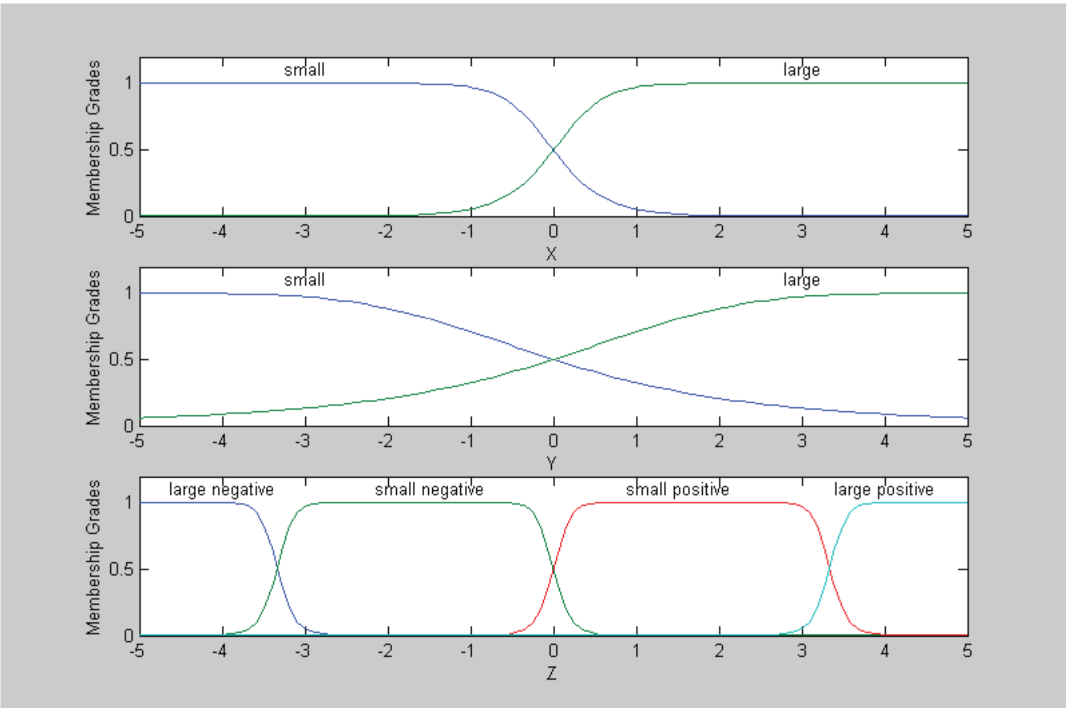
Two input single-output Mamdani fuzzy model with 4 rules:

If X is small & Y is small then Z is negative large

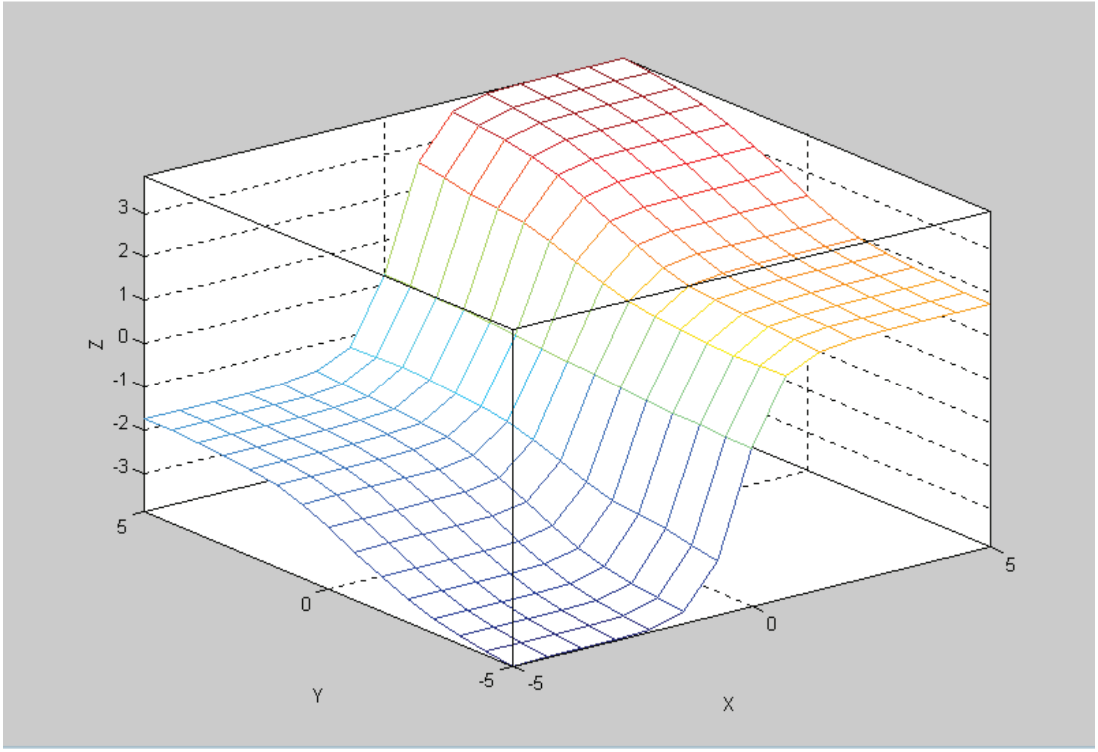
If X is small & Y is large then Z is negative small

If X is large & Y is small then Z is positive small

If X is large & Y is large then Z is positive large



Two-input single output antecedent & consequent MFs



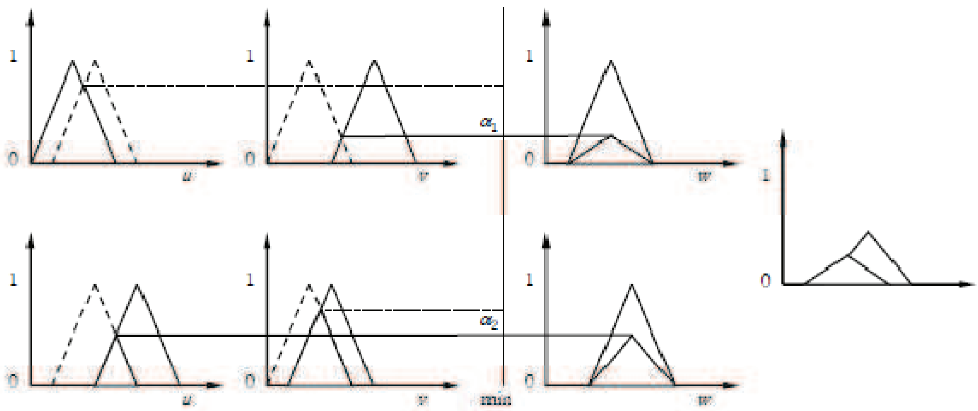
Overall input-output surface

Larsen Fuzzy models

Inference method: Larsen

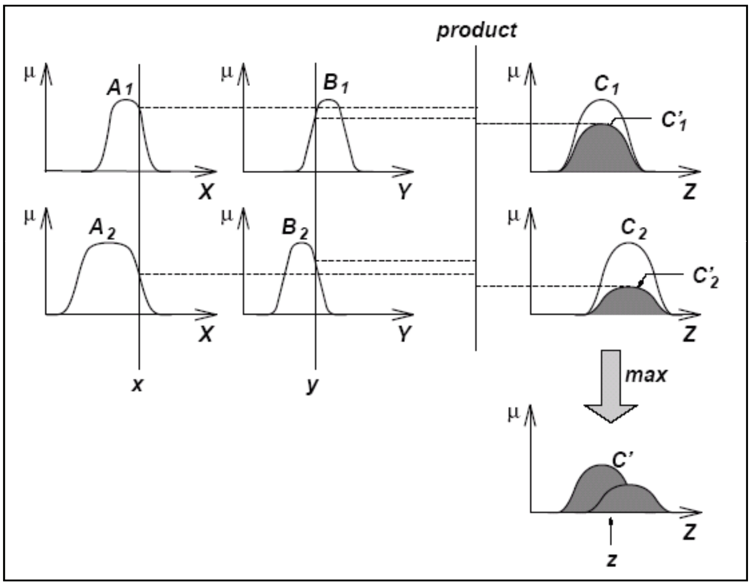
– product operator(•) for a fuzzy implication

– max-product operator for the composition



Max-Product Composition is used.

The Reasoning Scheme



Fuzzy Control Systems

Sugeno
Fuzzy Models

Sugeno Fuzzy Models

- Also known as TSK fuzzy model
  - Takagi, Sugeno & Kang, 1985

- Goal: Generation of fuzzy rules from a given input-output data set.

Sugeno Fuzzy Control

- Mamdani-style inference, requires to find the centroid of a two-dimensional shape
  - by integrating across a continuously varying function.
  - In general, this process is not computationally efficient.
- **Michio Sugeno** suggested to use a single spike, a *singleton*, as the membership function of the rule consequent.
- A **fuzzy singleton**, is a fuzzy set with a membership function that is unity at a single particular point on the universe of discourse and zero everywhere else.

- Sugeno-style fuzzy inference is very similar to the Mamdani method.
- Sugeno changed only a rule consequent. Instead of a fuzzy set, he used a mathematical function of the input variable.
- The format of the **Sugeno-style fuzzy rule** is

IF x is A AND y is B
THEN z is $f(x, y)$

- where x, y and z are linguistic variables
- A and B are fuzzy sets on universe of discourses X and Y
- $f(x, y)$ is a mathematical function

The most commonly used **zero-order Sugeno fuzzy model** applies fuzzy rules in the following form:

IF *x* is *A* AND *y* is *B*
THEN *z* is *k*

where k is a constant.

- In this case, the output of each fuzzy rule is constant.
- All consequent membership functions are represented by singleton spikes.

Fuzzy Rules of TSK Model

If x is A and y is B then $z = f(x, y)$

Fuzzy Sets

Crisp Function

$f(x, y)$ is very often a polynomial function w.r.t. x and y.

Examples

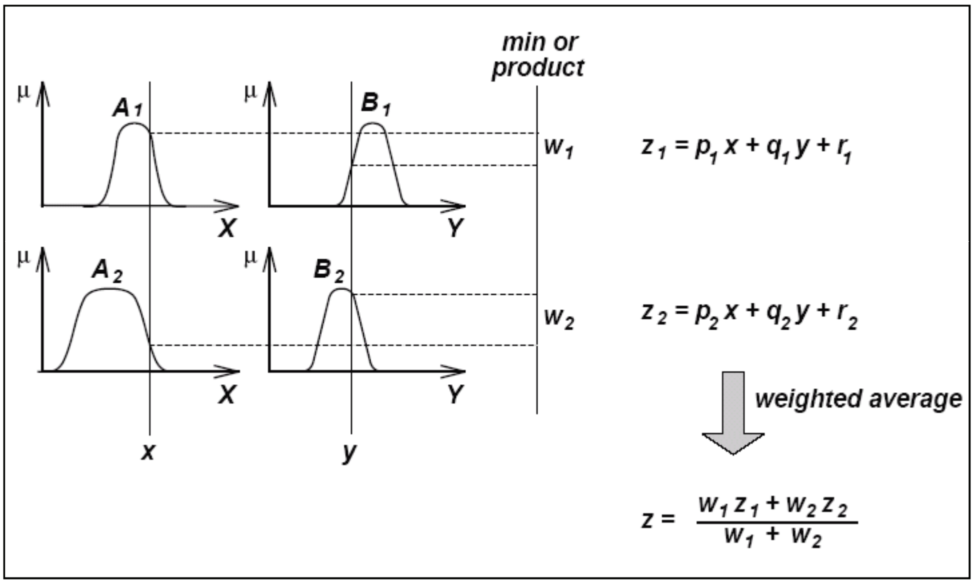
R1: if X is small and Y is small then $z = -x + y + 1$

R2: if X is small and Y is large then $z = -y + 3$

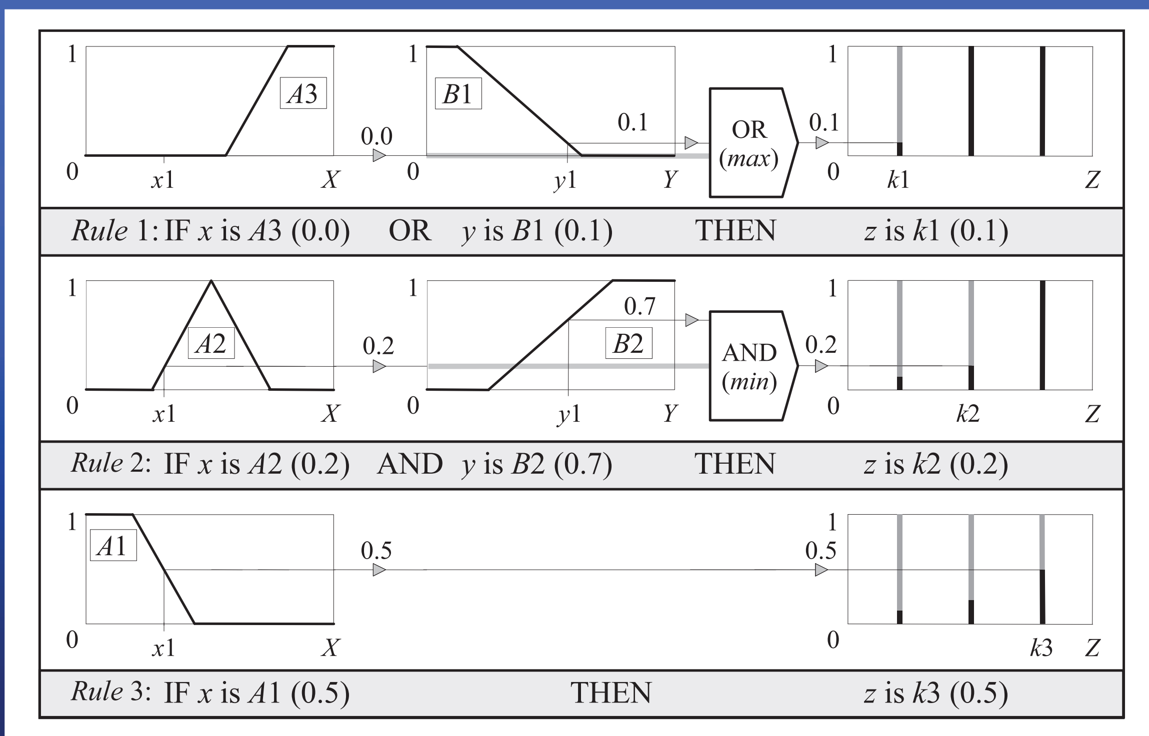
R3: if X is large and Y is small then $z = -x + 3$

R4: if X is large and Y is large then $z = x + y + 2$

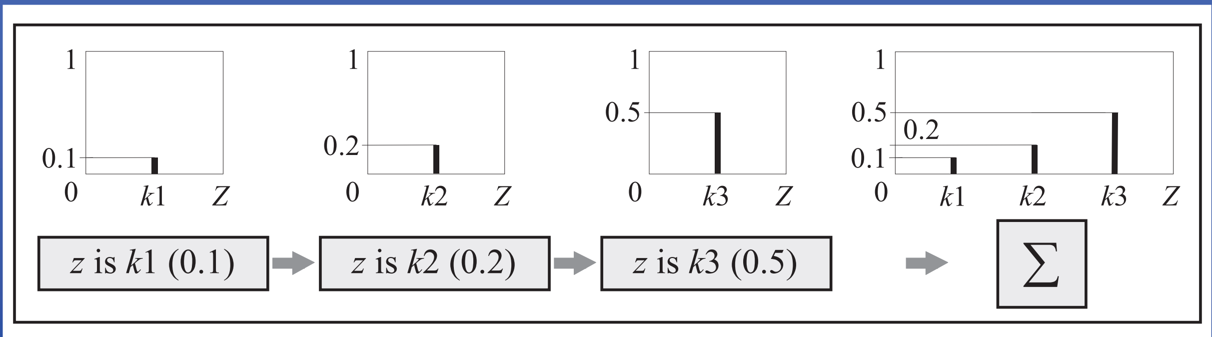
The Reasoning Scheme



Sugeno-style rule evaluation



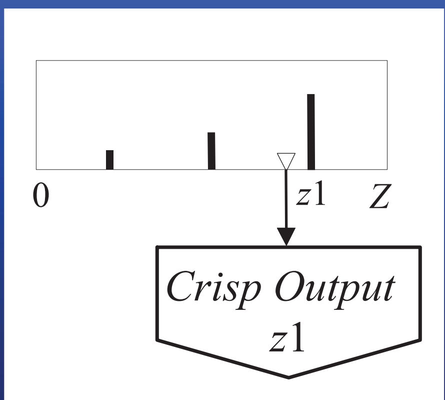
Sugeno-style aggregation of the rule outputs



Weighted average (WA):

$$WA = \frac{\mu(k1)\times k1 + \mu(k2)\times k2 + \mu(k3)\times k3}{\mu(k1)+\mu(k2)+\mu(k3)} = \frac{0.1\times 20 + 0.2\times 50 + 0.5\times 80}{0.1+0.2+0.5} = 65$$

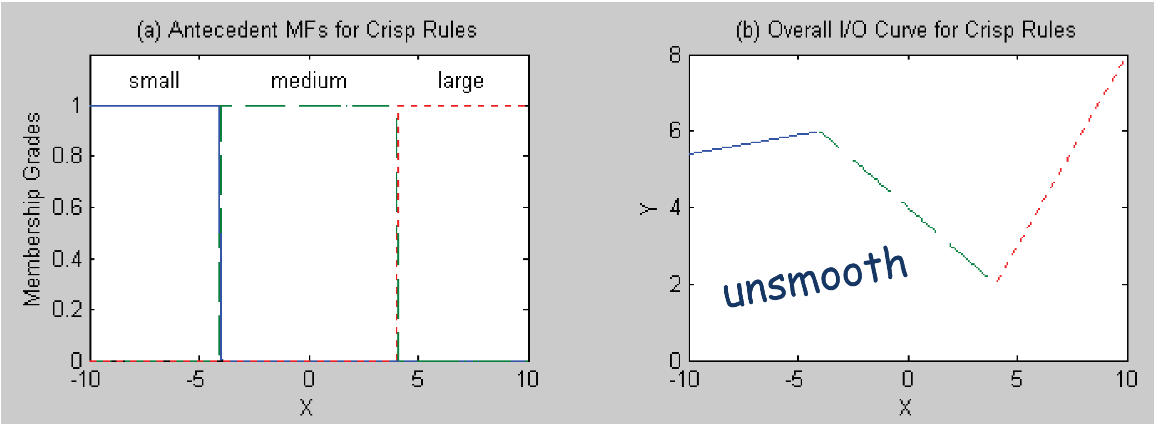
Sugeno-style defuzzification



Example

R1: If X is small then $Y = 0.1X + 6.4$
R2: If X is medium then $Y = -0.5X + 4$
R3: If X is large then $Y = X - 2$

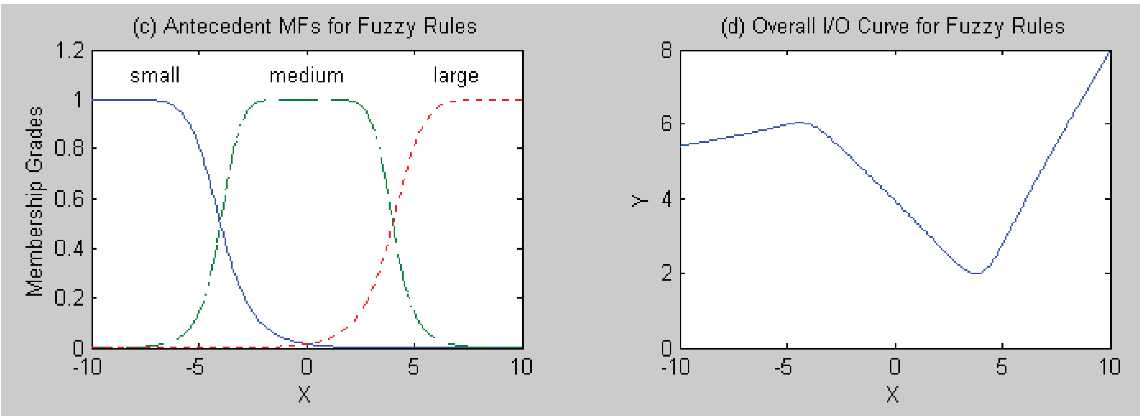
X = input $\in [-10, 10]$



Example

R1: If X is small then $Y = 0.1X + 6.4$
R2: If X is medium then $Y = -0.5X + 4$
R3: If X is large then $Y = X - 2$

X = input $\in [-10, 10]$

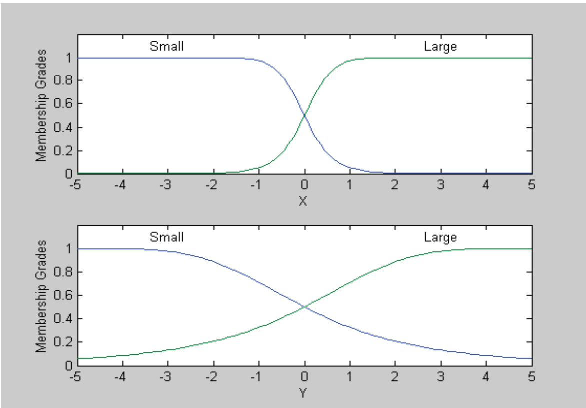


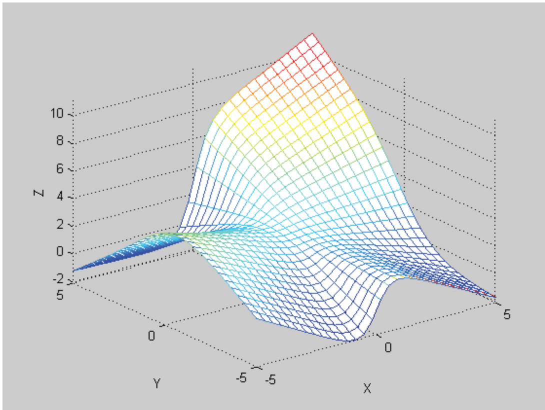
If we have smooth membership functions (fuzzy rules) the overall input-output curve becomes a smoother one.

Example

R1: if X is small and Y is small then $z = -x + y + 1$
R2: if X is small and Y is large then $z = -y + 3$
R3: if X is large and Y is small then $z = -x + 3$
R4: if X is large and Y is large then $z = x + y + 2$

$X, Y \in [-5, 5]$





Tsukamoto Fuzzy Model

The consequent of each fuzzy if-then rule:

- a fuzzy set with a monotonical MF.
- Overall output: the weighted average of each rule's output.
- No defuzzification.
- Not as transparent as mamdani's or Sugeno's fuzzy model.
- Not follow strictly the compositional rule of inference: the output is always crisp.

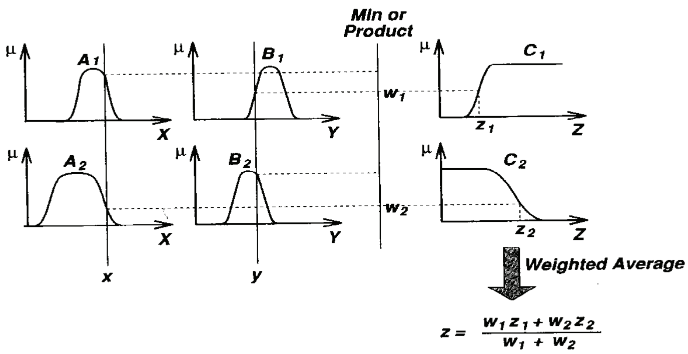
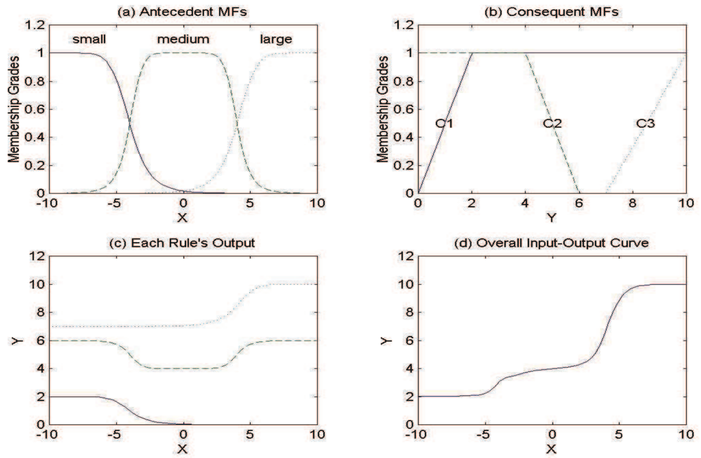


Figure 4.11. *The Tsukamoto fuzzy model.*

Example: Tsukamoto Fuzzy Model

Single-input Tsukamoto fuzzy model

If X is small *then Y is* C_1.
If X is medium *then Y is* C_2.
If X is large *then Y is* C_3.



Review Fuzzy Models

If <antecedence> then <consequence>.

The same style for
- Mamdani Fuzzy Models
- Larsen Fuzzy Models
- Sugeno Fuzzy Models
- Tsukamoto Fuzzy Models

Different styles for
- Mamdani Fuzzy Models
- Larsen Fuzzy Models
- Sugeno Fuzzy Models
- Tsukamoto Fuzzy models

How to make a decision on which method to apply – Mamdani or Sugeno?

- Mamdani method is widely accepted for capturing expert knowledge.
  - It allows us to describe the expertise in human-like manner.

- Sugeno method is computationally effective and works well with optimization and adaptive techniques, which makes it very attractive in control problems, particularly for dynamic nonlinear systems.

Tuning Fuzzy Systems

1. Review model input and output variables, and if required redefine their ranges.

2. Review the fuzzy sets, and if required define additional sets on the universe of discourse.
   - The use of wide fuzzy sets may cause the fuzzy system to perform roughly.

3. Provide sufficient overlap between neighbouring sets.
   - It is suggested that triangle-to-triangle and trapezoid-to-triangle fuzzy sets should overlap between 25% to 50% of their bases.

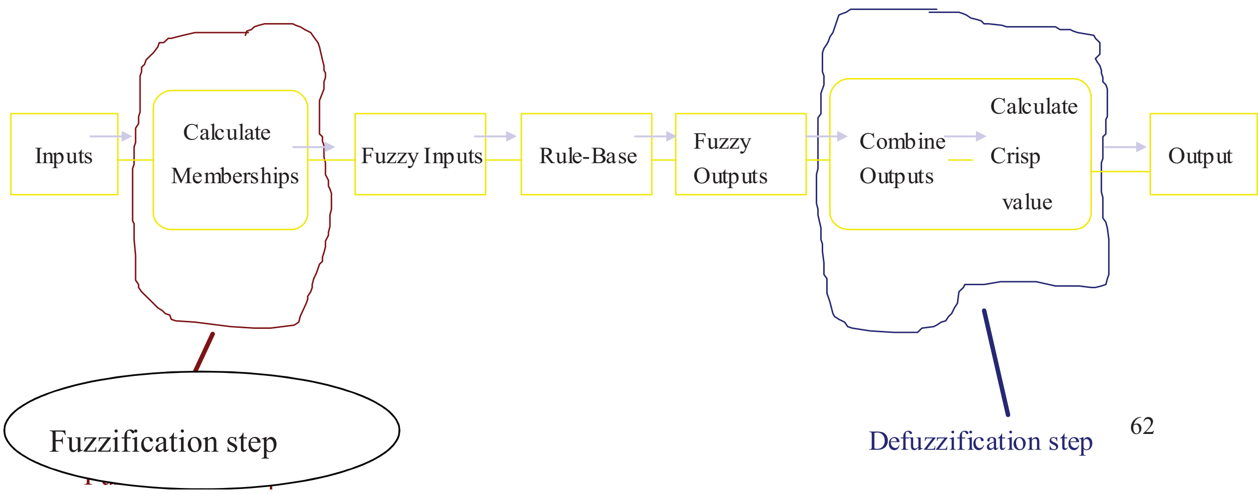
4. Review the existing rules, and if required add new rules to the rule base.

5. Adjust the rule execution weights. Most fuzzy logic tools allow control of the importance of rules by changing a weight multiplier.

6. Revise shapes of the fuzzy sets. In most cases, fuzzy systems are highly tolerant of a shape approximation.

Steps in Designing a Fuzzy Logic Control System

1. Identify the system input variables, their ranges, and membership functions.
2. Identify the output variables, their ranges, and membership functions.
3. Identify the rules that describe the relations of the inputs to the outputs.
4. Determine the de-fuzzifier method of combining fuzzy rules into system outputs.



EXAMPLES

Building a Fuzzy Expert System: Case Study

- A service centre keeps spare parts and repairs failed ones.
- A customer brings a failed item and receives a spare of the same type.
- Failed parts are repaired, placed on the shelf, and thus become spares.
- The objective here is to advise a manager of the service centre on certain decision policies to keep the customers satisfied.

Process of Developing a Fuzzy Expert System

1. Specify the problem and define linguistic variables.
2. Determine fuzzy sets.
3. Elicit and construct fuzzy rules.
4. Encode the fuzzy sets, fuzzy rules and procedures to perform fuzzy inference into the expert system.
5. Evaluate and tune the system.

Step 1: Specify the problem and define linguistic variables

There are four main linguistic variables: average waiting time (mean delay) m, repair utilisation factor of the service centre ρ (is the ratio of the customer arrival day to the customer departure rate), number of servers s, and initial number of spare parts n .

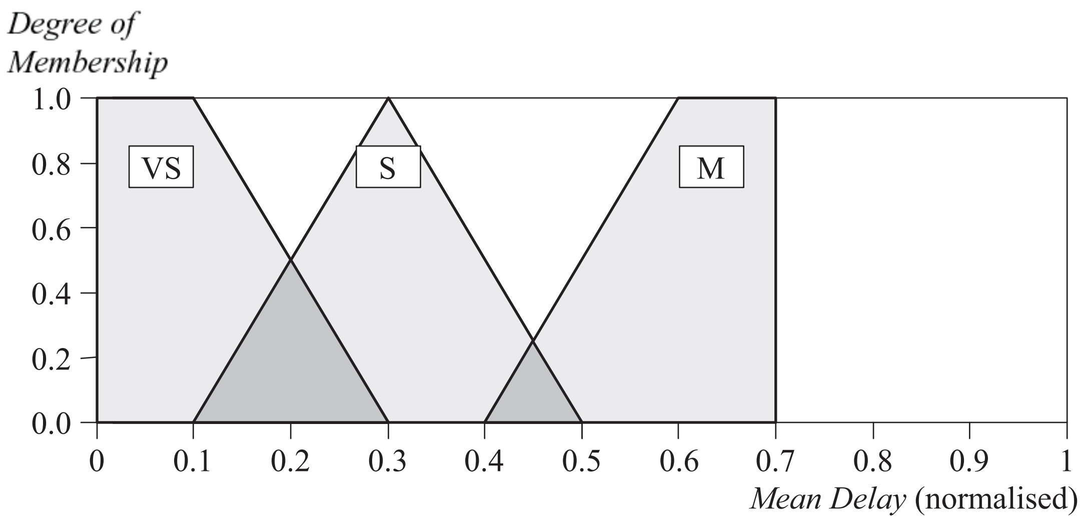
Linguistic variables and their ranges

Linguistic Variable: *Mean Delay*, *m*		
Linguistic Value	Notation	Numerical Range (normalised)
Very Short	VS	[0, 0.3]
Short	S	[0.1, 0.5]
Medium	M	[0.4, 0.7]
Linguistic Variable: *Number of Servers*, *s*		
Linguistic Value	Notation	Numerical Range (normalised)
Small	S	[0, 0.35]
Medium	M	[0.30, 0.70]
Large	L	[0.60, 1]
Linguistic Variable: *Repair Utilisation Factor*, ρ		
Linguistic Value	Notation	Numerical Range
Low	L	[0, 0.6]
Medium	M	[0.4, 0.8]
High	H	[0.6, 1]
Linguistic Variable: *Number of Spares*, *n*		
Linguistic Value	Notation	Numerical Range (normalised)
Very Small	VS	[0, 0.30]
Small	S	[0, 0.40]
Rather Small	RS	[0.25, 0.45]
Medium	M	[0.30, 0.70]
Rather Large	RL	[0.55, 0.75]
Large	L	[0.60, 1]
Very Large	VL	[0.70, 1]

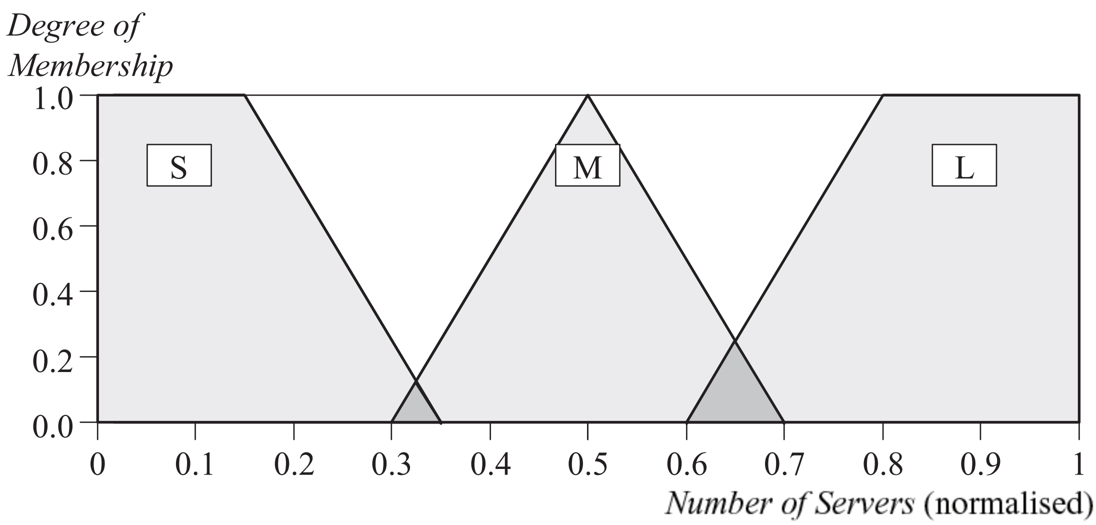
Step 2: Determine Fuzzy Sets

Fuzzy sets can have a variety of shapes. However, a triangle or a trapezoid can often provide an adequate representation of the expert knowledge, and at the same time, significantly simplifies the process of computation.

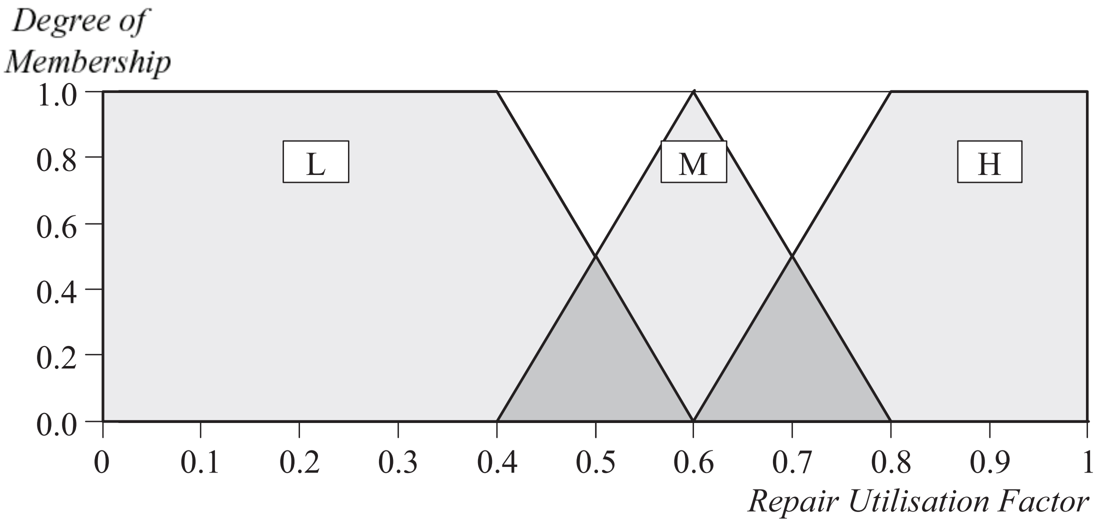
Fuzzy sets of *Mean Delay* ***m***



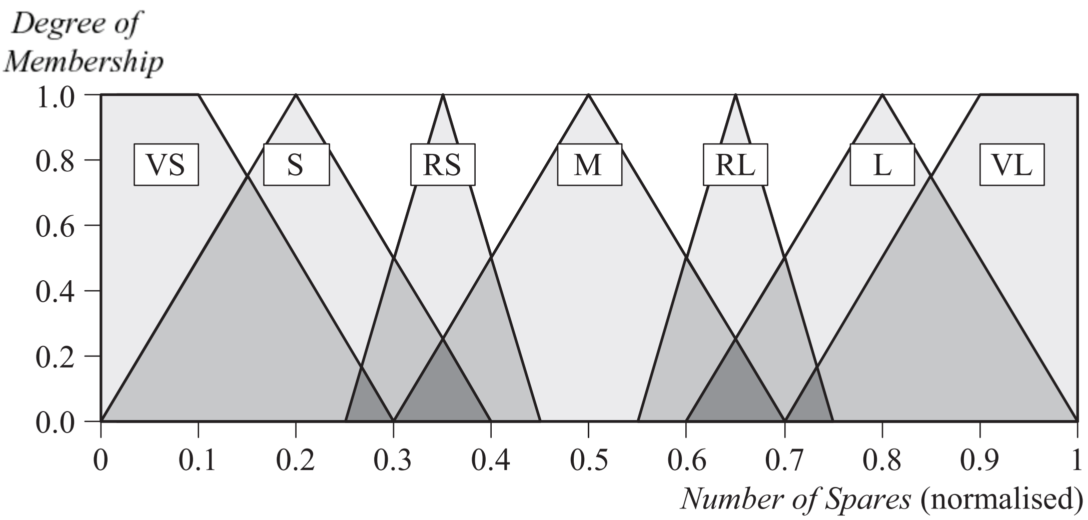
Fuzzy sets of *Number of Servers* ***s***



Fuzzy sets of *Repair Utilisation Factor* ρ



Fuzzy sets of *Number of Spares* n



Step 3: Elicit and construct fuzzy rules

To accomplish this task, we might ask the expert to describe how the problem can be solved using the fuzzy linguistic variables defined previously.

Required knowledge also can be collected from other sources such as books, computer databases, flow diagrams and observed human behavior.

The matrix form of representing fuzzy rules is called fuzzy associative memory (FAM).

The square FAM representation

s				
L	**M**	**S**	**VS**	
M	**RL**	**RS**	**S**	
S	**VL**	**L**	**M**	
	VS	S	M	*m*

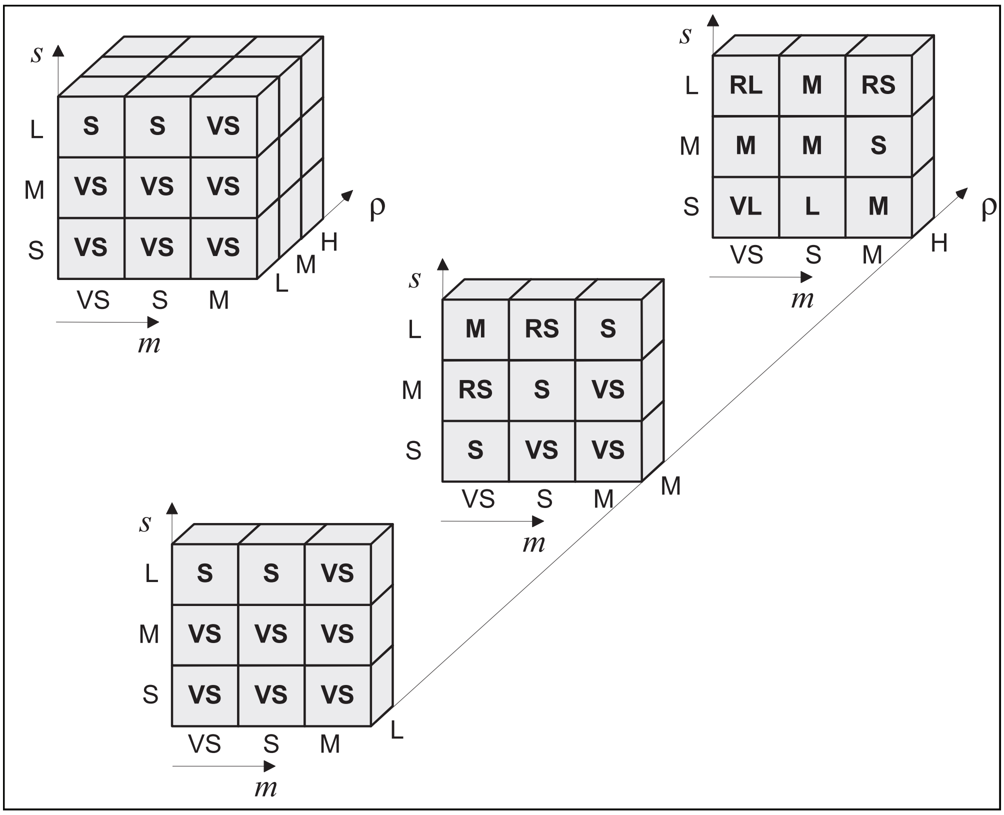
The rule table

Rule	*m*	*s*	ρ	*n*	Rule	*m*	*s*	ρ	*n*	Rule	*m*	*s*	ρ	*n*
1	VS	S	L	VS	10	VS	S	M	S	19	VS	S	H	VL
2	S	S	L	VS	11	S	S	M	VS	20	S	S	H	L
3	M	S	L	VS	12	M	S	M	VS	21	M	S	H	M
4	VS	M	L	VS	13	VS	M	M	RS	22	VS	M	H	M
5	S	M	L	VS	14	S	M	M	S	23	S	M	H	M
6	M	M	L	VS	15	M	M	M	VS	24	M	M	H	S
7	VS	L	L	S	16	VS	L	M	M	25	VS	L	H	RL
8	S	L	L	S	17	S	L	M	RS	26	S	L	H	M
9	M	L	L	VS	18	M	L	M	S	27	M	L	H	RS

Rule Base 1

1. If (utilisation_factor is L) then (number_of_spares is S)
2. If (utilisation_factor is M) then (number_of_spares is M)
3. If (utilisation_factor is H) then (number_of_spares is L)
4. If (mean_delay is VS) and (number_of_servers is S) then (number_of_spares is VL)
5. If (mean_delay is S) and (number_of_servers is S) then (number_of_spares is L)
6. If (mean_delay is M) and (number_of_servers is S) then (number_of_spares is M)
7. If (mean_delay is VS) and (number_of_servers is M) then (number_of_spares is RL)
8. If (mean_delay is S) and (number_of_servers is M) then (number_of_spares is RS)
9. If (mean_delay is M) and (number_of_servers is M) then (number_of_spares is S)
10. If (mean_delay is VS) and (number_of_servers is L) then (number_of_spares is M)
11. If (mean_delay is S) and (number_of_servers is L) then (number_of_spares is S)
12. If (mean_delay is M) and (number_of_servers is L) then (number_of_spares is VS)

Cube FAM of Rule Base 2



Step 4: Encode the fuzzy sets, fuzzy rules and procedures to perform fuzzy inference into the expert system

To accomplish this task, we may choose one of two options: to build our system using a programming language such as C/C++, Java, or to apply a fuzzy logic development tool such as MATLAB Fuzzy Logic Toolbox or Fuzzy Knowledge Builder.

Step 5: Evaluate and Tune the System

The last task is to evaluate and tune the system. We want to see whether our fuzzy system meets the requirements specified at the beginning.

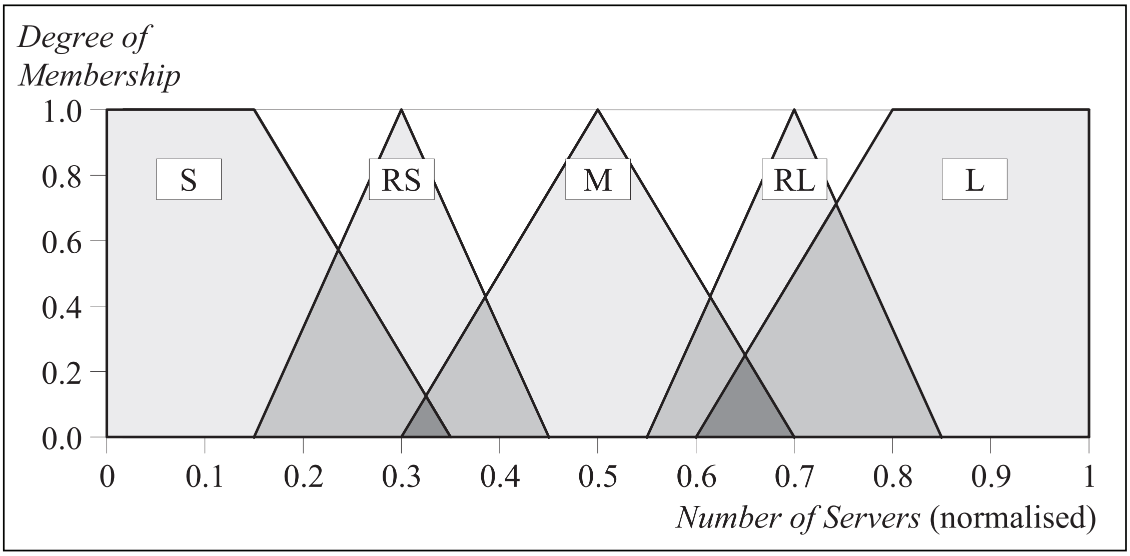
Several test situations depend on the mean delay, number of servers and repair utilisation factor.

The Fuzzy Logic Toolbox can generate surface to help us analyse the system's performance.

However, even now, the expert might not be satisfied with the system performance.

To improve the system performance, we may use additional sets – *Rather Small* and *Rather Large* – on the universe of discourse *Number of Servers*, and then extend the rule base.

Modified Fuzzy Sets of *Number of Servers s*



Cube FAM of Rule Base 3