Fuzzy Logic - II

Neural Networks & Fuzzy Logic

(BITS F312)

Neuro-Fuzzy-GA Hybrid Systems

Why Hybridization?

- The three members have strengths in different contexts
- Objective is to combine their strengths
- And overcome their weaknesses
- Improved performance over wider ranges of variables
- Higher complexity

GA based Tuning of Fuzzy Systems

- Success of an FLC heavily depends on how appropriately the membership functions and the rule base have been defined

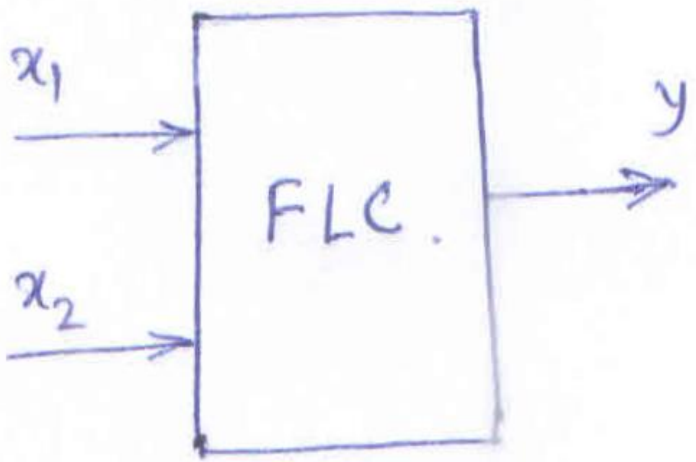
- They are designed from domain expertise
- For a few input-output variables may be manageable
- Difficult and less accurate when the variables and/or fuzzy parameters grow
- An optimizer helps in arriving at an optimal setting of the parameters

- Because of large number of parameters, traditional optimization tools are nearly inapplicable

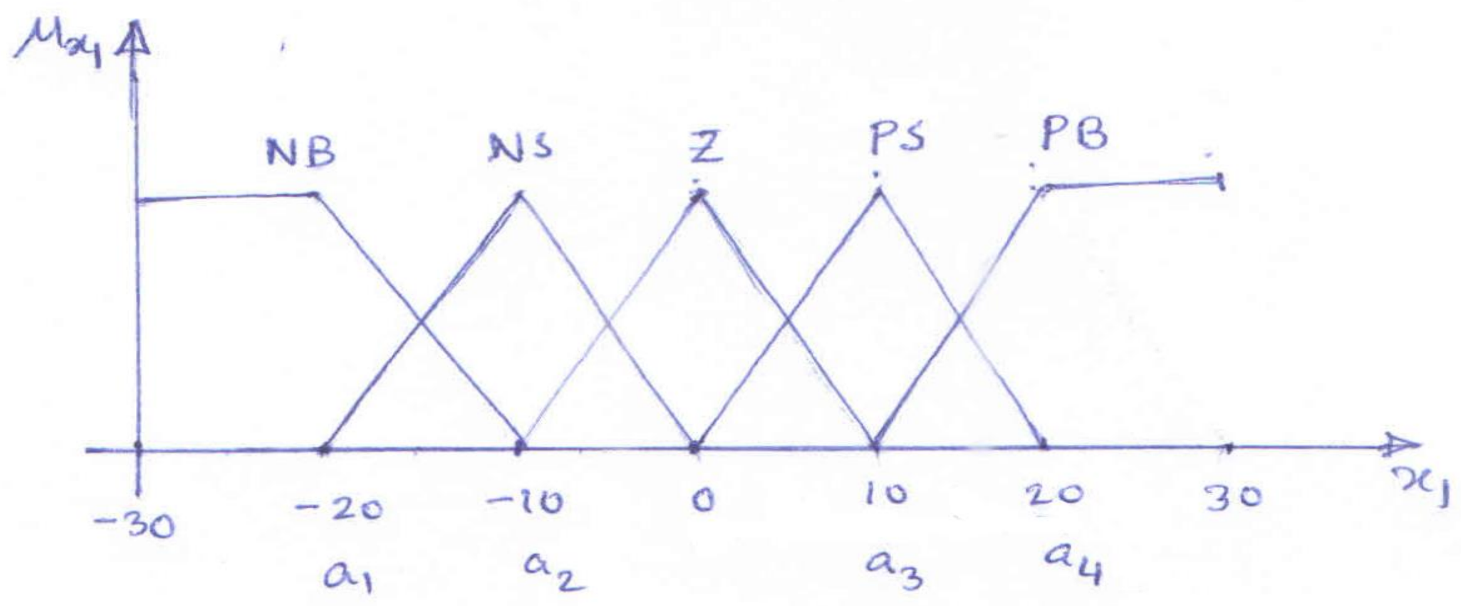
- Requires some input-output data to be acquired along with the qualitative understanding about the behaviour of the phenomenon

- Membership functions, scaling factors and rule base are tuned

Tuning of MF:



Sl. No.	x_1	x_2	y
1.	5.0	2.5	10.1
2.	3.5	-4.5	-8.2
3.	⋮	⋮	⋮
⋮			
T			



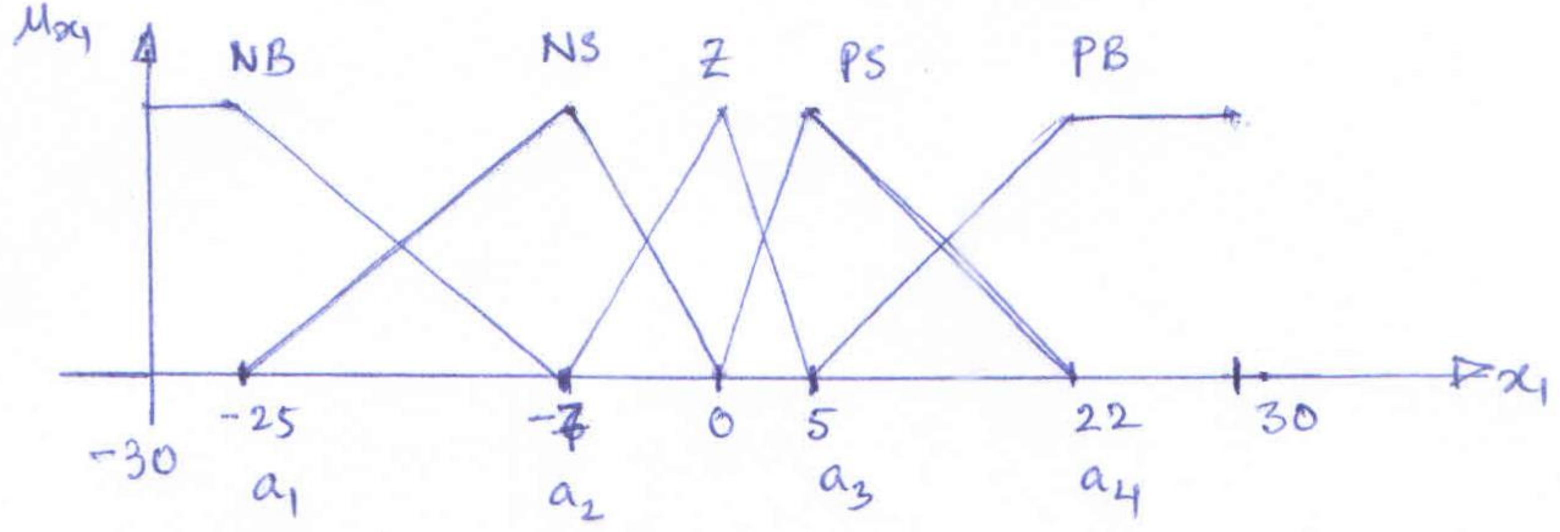
Let,

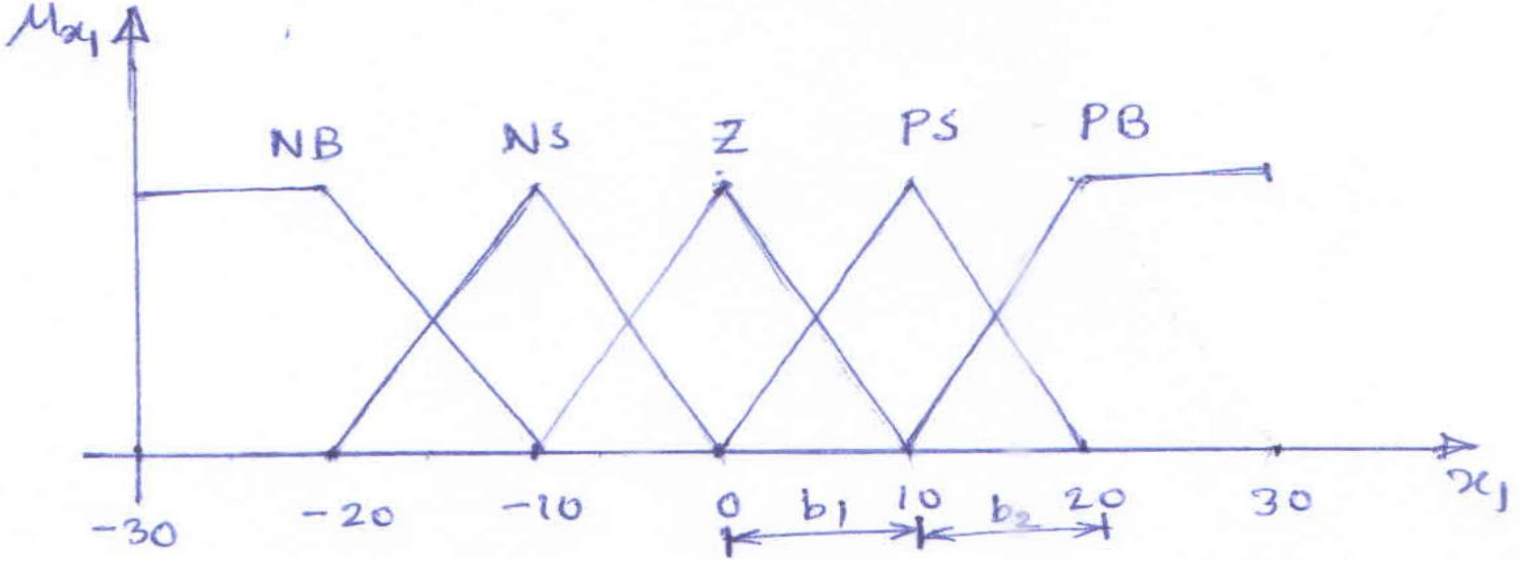
$a_1 \in [-25, -15]$

$a_2 \in [-15, -5]$

$a_3 \in [5, 15]$

$a_4 \in [15, 25]$

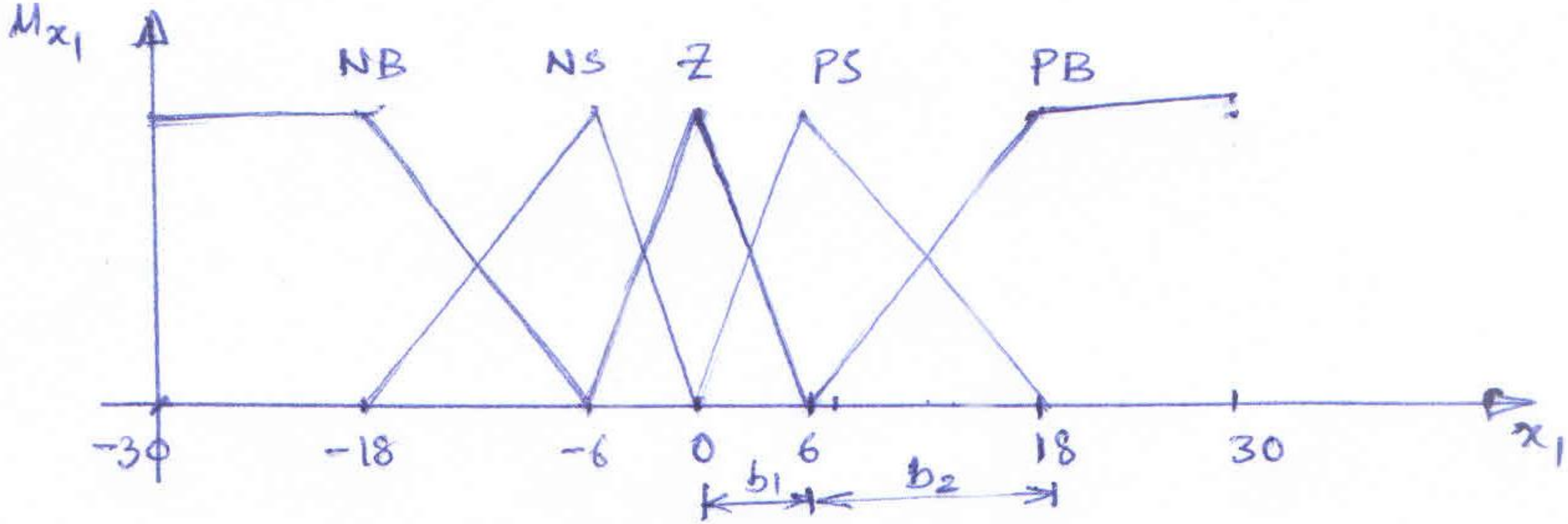


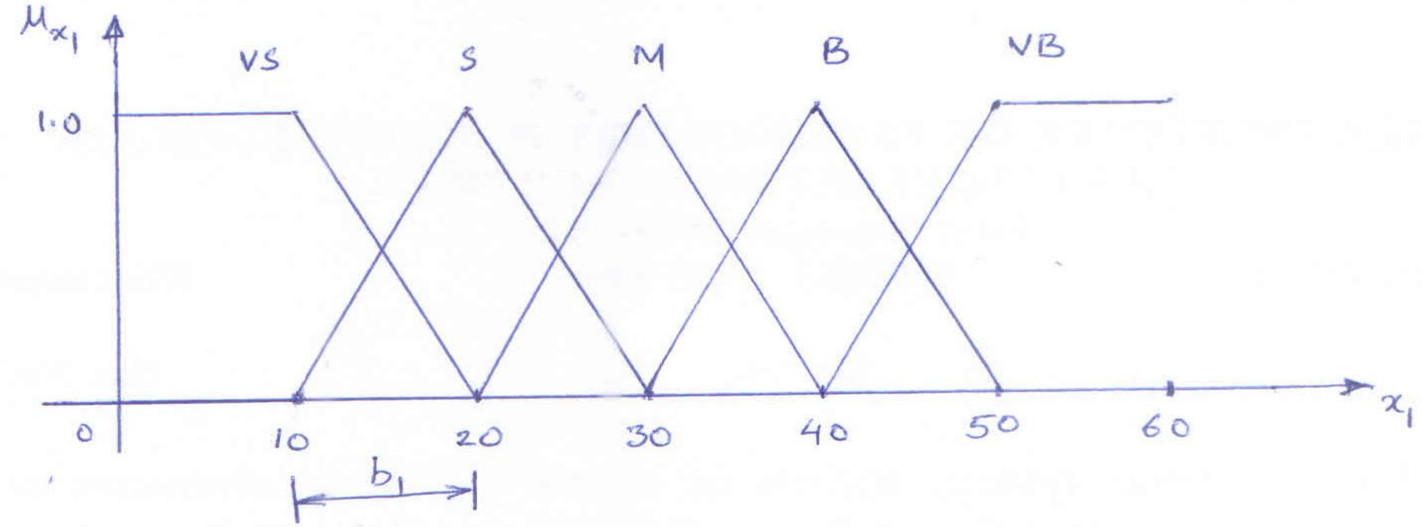


let,

$b_1 \in [5, 15]$

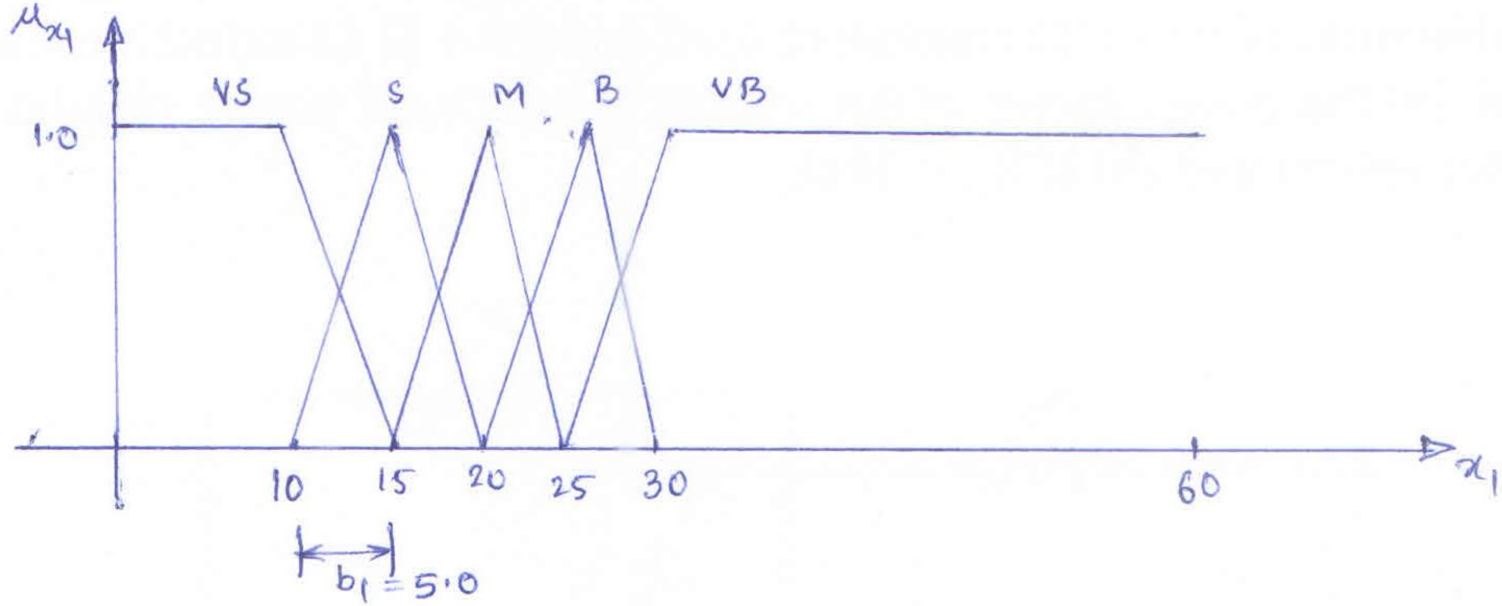
$b_2 \in [5, 15]$





μ_{x_1}
VS
S
M
B
VB
1.0
0
10
20
30
40
50
60
x_1
b_1

$b_1 \in [5, 15]$



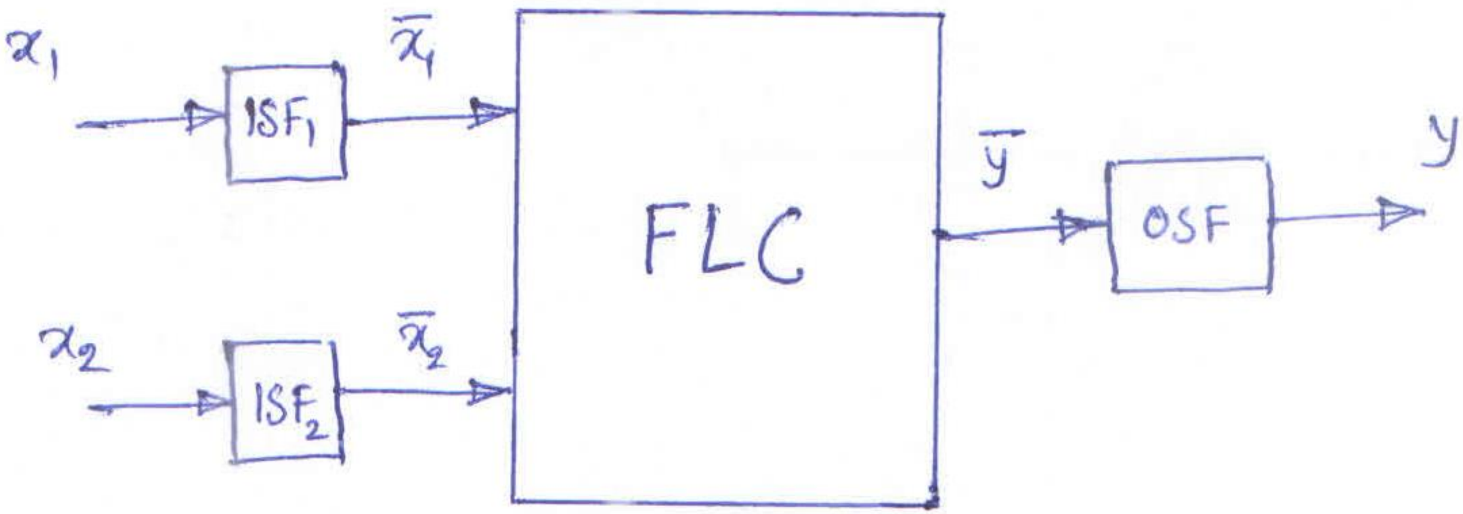
μ_{x_1}
VS
S
M
B
VB
1.0
10
15
20
25
30
60
x_1
$b_1 = 5.0$

x_1 \ x_2	NB	NS	Z	PS	PB
NB	NB	NB	NS	NS	Z
NS	NB	NS	NS	Z	PS
Z	NS	NS	Z	PS	PS
PS	NS	Z	PS	PS	PB
PB	Z	PS	PS	PB	PB

Initial Population :-

Sl. No.	GA String (36 bit)	Deviation	fitness
1	$\underbrace{010011}_{b_1}\ \underbrace{111000}_{b_2} \cdots \underbrace{001111}_{b_6}$	$\bar{d}_1 = \frac{1}{T}\sum \lvert d_i \rvert$	$f_1 = \frac{1}{\bar{d}_1}$
~~2~~			
2	⋮	⋮	⋮
⋮	⋮	⋮	⋮
N	⋮	⋮	⋮

Tuning of Scaling Factors:



Tuning of Rule Base:

- ✓ Some rules may be redundant
- ✓ They may be removed taking help from the database

x_1 \ x_2	NB	NS	Z	PS	PB
NB	NB	NB	NS	NS	Z
NS	NB	NS	NS	Z	PS
Z	NS	NS	Z	PS	PS
PS	NS	Z	PS	PS	PB
PB	Z	PS	PS	PB	PB

Sl. No.	x_1	x_2	y
1.	5.0	2.5	10.1
2.	3.5	-4.5	-8.2
3.	⋮	⋮	⋮
⋮			
T			

- ✓ Let '0' imply absence and '1' imply presence of a rule
- ✓ Hence a 25 bit long string will represent the entire rule base

Initial GA population —

Sl. No.	GA String (61 bit)	fitness
1.	$\underbrace{110010}_{b_1} \cdots \underbrace{011110}_{b_6} \; \underbrace{1011010001011110011011100}_{\text{RB (25 bit)}}$	f_1
2.		,
⋮	⋮	⋮
N		

String #1 corresponds to the following RB —

x_1 \ x_2	NB	NS	Z	PS	PB
NB	NB	—	NS	NS	—
NS	NB	—	—	—	PS
Z	—	NS	Z	PS	PS
PS	—	—	PS	PS	—
PB	Z	PS	PS	—	—

fitness, $f_1 = \dfrac{1}{\bar{d}_1 + P_1}$ where P_1 = no. of rules present

Generation of Rule Base:

- ✓ Designer does not have enough intuition to design the initial rule base
- ✓ GA can design the rule base that will fit the input-output data set

Let

NB	≡	000
NS	≡	001
Z	≡	010
PS	≡	011
PB	≡	100

- ✓ Three bits are assigned to the consequent part of each rule
- ✓ Hence a 75 bit string for the consequent parts of the entire rule base

Sl.No	GA String (136 bit)	fitness
1.	$\underbrace{110010}_{b_1} \cdots \underbrace{011110}_{b_6}\ \underbrace{10110 \cdots 11100}_{RB\ (25\ bits)}\ \underbrace{001}_{\text{Con. of } R1}\ \underbrace{100}_{\text{Con. of } R_2} \cdots \underbrace{011}_{\text{Con. of } R25}$	f_1
⋮	⋮	⋮
N		

NB ≡ 000

NS ≡ 001

Z ≡ 010

PS ≡ 011

PB ≡ 100

Some Observations:

- ✓ Too long GA strings
- ✓ Switch to Real Coded GA

✓ Mixed Real and Integer Valued Optimization

NB	≡	000
NS	≡	001
Z	≡	010
PS	≡	011
PB	≡	100

NB	≡	1
NS	≡	2
Z	≡	3
PS	≡	4
PB	≡	5

$$\underset{R1}{\underset{\downarrow}{2}}\;\underset{R2}{\underset{\downarrow}{5}}\;,\,,\,,\;\underset{R25}{\underset{\downarrow}{4}}$$

Sl.No	GA String (136 bit)	fitness
1.	$\underbrace{110010}_{b_1} \cdots \underbrace{011110}_{b_6}\ \underbrace{10110 \cdots 11100}_{\text{RB (25 bits)}}\ \underbrace{001}_{\text{Con. of R1}}\ \underbrace{100}_{\text{Con. of } R_2} \cdots \underbrace{011}_{\text{Con. of R25}}$	f_1
⋮	⋮	⋮
N		

- ✓ After going through the GA operations, we may not arrive at a valid consequent
- ✓ Hence number of output fuzzy sets may be increased to 8 or decreased to 4
- ✓ But neither will be symmetrical
- ✓ May be made symmetrical by dropping the Zero fuzzy set
- ✓ However, may lead to complexity in forming some Rules

Sl.No	GA String (136 bit)	fitness
1.	$\underbrace{110010}_{b_1} \cdots \underbrace{011110}_{b_6}\ \underbrace{10110 \cdots 11100}_{\text{RB (25 bits)}}\ \underbrace{001}_{\text{Con. of R1}}\ \underbrace{100}_{\text{Con. of } R_2} \cdots \underbrace{011}_{\text{Con. of R25}}$	f_1
⋮	⋮	⋮
N		

Michigan Approach:

- ✓ Each Rule is represented by a GA string
- ✓ Population size = Number of rules
- ✓ Hence RB is represented by the whole population

x_1 \ x_2	NB	NS	Z	PS	PB
NB	NB	NB	NS	NS	Z
NS	NB	NS	NS	Z	PS
Z	NS	NS	Z	PS	PS
PS	NS	Z	PS	PS	PB
PB	Z	PS	PS	PB	PB

R1: 1 1 1
R2: 1 2 1
R3: 1 3 2
.
.
R25: 5 5 5

NB ≡ 1
NS ≡ 2
Z ≡ 3
PS ≡ 4
PB ≡ 5

Pittsburgh Approach:

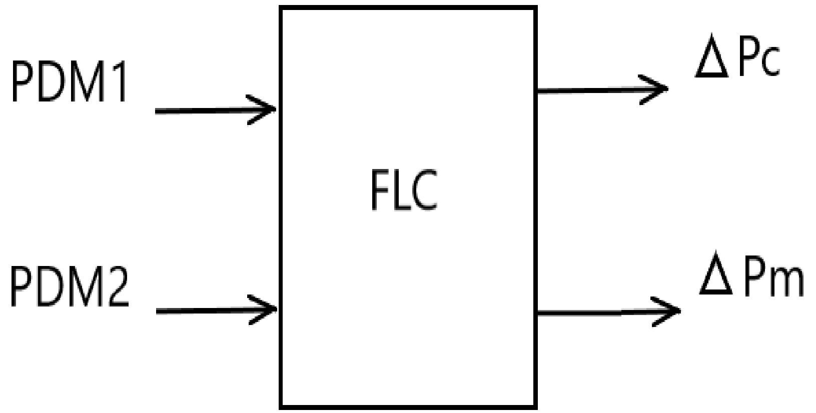
- ✓ Entire RB is represented by a single GA string

Fuzzy based Tuning of GA Parameters:

- ✓ GA parameters $\mathrm{p_c}, \mathrm{p_m}$ and N are set beforehand
- ✓ They are chosen intuitively
- ✓ An FLC is brought in to make the search more effective (i.e. to avoid premature convergence)
- ✓ Two diversity measure PDM1 and PDM2 are defined in Dynamic Parametric GA (DPGA)

✓ $PDM_1 = \frac{\bar{f}}{f_{best}} \in [0,1]$

$PDM_2 = \frac{f_{worst}}{\bar{f}} \in [0,1]$



Typical Rules: If PDM1 is High Then ΔPc is Positive

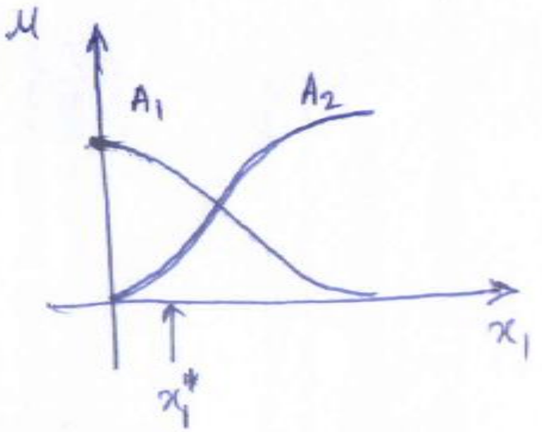
If PDM2 is Low Then ΔPm is Negative

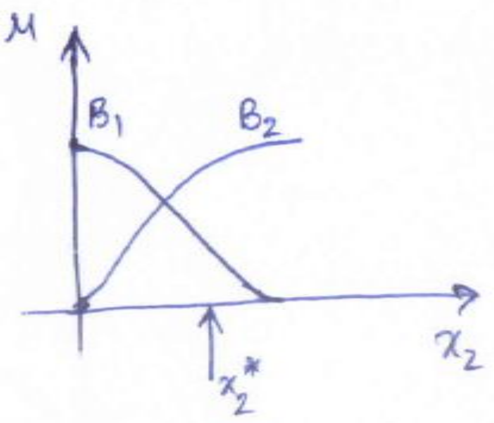
Neuro-Fuzzy Systems

- ✓ Fuzzy parameters are learned
- ✓ Adaptivity and generalization
- ✓ On-line applications

Adaptive Neuro-Fuzzy Inference System (ANFIS)

- ✓ Sugeno Inference
- ✓ Algebraic Product T-norm
- ✓ Backpropagation learning





x_1 \ x_2	B_1	B_2
A_1	$f_1(\cdot,\cdot)$	$f_2(\cdot,\cdot)$
A_2	$f_3(\cdot,\cdot)$	$f_4(\cdot,\cdot)$

$$A_1(x_1) = \frac{1}{1+e^{b_1(x_1-a_1)}}, \quad A_2(x_1) = \frac{1}{1+e^{-b_1(x_1-a_1)}}$$

$$B_1(x_2) = \frac{1}{1+e^{b_2(x_2-a_2)}}, \quad B_2(x_2) = \frac{1}{1+e^{-b_2(x_2-a_2)}}$$

$$y_1 = f_1 = c_{11}x_1 + c_{12}x_2 + c_{13}$$

$$y_2 = f_2 = c_{21}x_1 + c_{22}x_2 + c_{23}$$

$$y_3 = f_3 = c_{31}x_1 + c_{32}x_2 + c_{33}$$

$$y_4 = f_4 = c_{41}x_1 + c_{42}x_2 + c_{43}$$

$$\omega_1 = A_1(x_1)\, B_1(x_2)$$

$$\omega_2 = A_1(x_1) \cdot B_2(x_2)$$

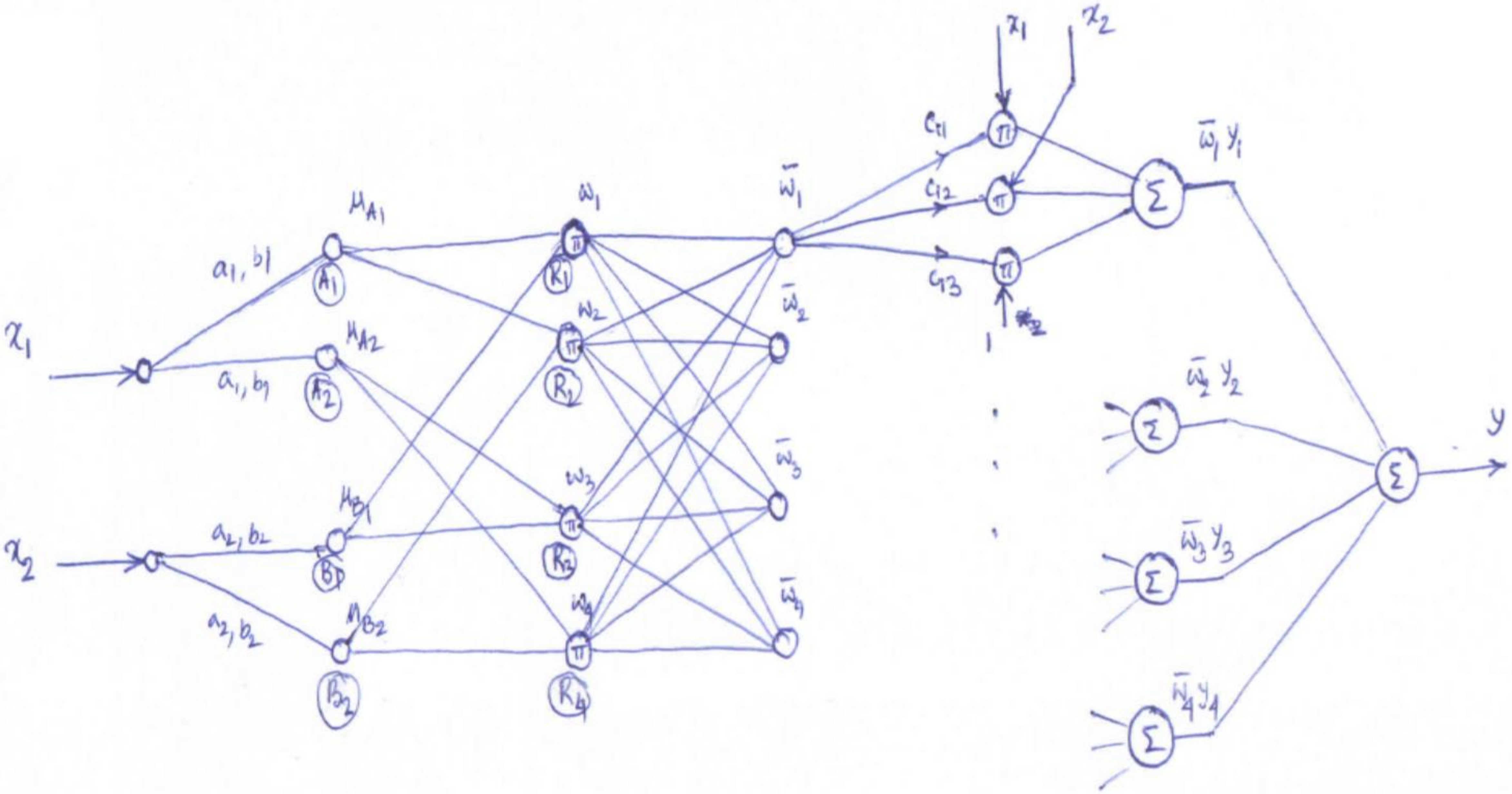
$$\omega_3 = A_2(x_1) \cdot B_1(x_2)$$

$$\omega_4 = A_2(x_1) \cdot B_2(x_2)$$

Sl. No.	x_1	x_2	y
1.	5.0	2.5	10.1
2.	3.5	-4.5	-8.2
3.	⋮	⋮	⋮
⋮			

$$y = \frac{\omega_1 y_1 + \omega_2 y_2 + \omega_3 y_3 + \omega_4 y_4}{\omega_1 + \omega_2 + \omega_3 + \omega_4}$$

$$= \bar{\omega}_1 y_1 + \bar{\omega}_2 y_2 + \bar{\omega}_3 y_3 + \bar{\omega}_4 y_4$$



x_1
x_2
a_1, b_1
a_1, b_1
a_2, b_2
a_2, b_2
μ_{A_1}
μ_{A_2}
μ_{B_1}
μ_{B_2}
A_1
A_2
B_1
B_2
w_1
w_2
w_3
w_4
R_1
R_2
R_3
R_4
$\bar{w}_1$
$\bar{w}_2$
$\bar{w}_3$
$\bar{w}_4$
c_{11}
c_{12}
c_{13}
x_1
x_2
x_2
1
$\bar{w}_1 y_1$
$\bar{w}_2 y_2$
$\bar{w}_3 y_3$
$\bar{w}_4 y_4$
y

$$E = \frac{1}{2}(T-y)^2$$

$$\Delta c_{11} = -\eta \frac{\partial E}{\partial c_{11}} = -\eta \frac{\partial E}{\partial y_1} \cdot \frac{\partial y_1}{\partial c_{11}} = -\eta \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial y_1} \cdot \frac{\partial y_1}{\partial c_{11}} = \eta (T-y) \cdot \bar{\omega}_1 \cdot x_1$$

$$\Delta c_{12} = -\eta \frac{\partial E}{\partial c_{12}} = -\eta \frac{\partial E}{\partial y_1} \cdot \frac{\partial y_1}{\partial c_{12}} = -\eta \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial y_1} \cdot \frac{\partial y_1}{\partial c_{12}} = \eta (T-y) \cdot \bar{\omega}_1 \cdot x_2$$

$$\Delta c_{13} = -\eta \frac{\partial E}{\partial c_{13}} = -\eta \frac{\partial E}{\partial y_1} \cdot \frac{\partial y_1}{\partial c_{13}} = -\eta \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial y_1} \cdot \frac{\partial y_1}{\partial c_{13}} = \eta (T-y) \bar{\omega}_1 \cdot$$

$$\Delta c_{21} = -\eta \frac{\partial E}{\partial c_{21}} = -\eta \frac{\partial E}{\partial y_2} \cdot \frac{\partial y_2}{\partial c_{21}} = -\eta \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial y_2} \cdot \frac{\partial y_2}{\partial c_{21}} = \eta (T-y) \cdot \bar{\omega}_2 \cdot x_1$$

Next tunable parameters are : a_1, b_1, a_2, b_2.

$$\Delta a_1 = -\eta \frac{\partial E}{\partial a_1} = -\eta \cdot \left[\frac{\partial E}{\partial A_1} \frac{\partial A_1}{\partial a_1} + \frac{\partial E}{\partial A_2} \cdot \frac{\partial A_2}{\partial a_1} \right]$$

Now, $\frac{\partial A_1}{\partial a_1} = \frac{\partial}{\partial a_1} \left(\frac{1}{1+e^{b_1(x_1-a_1)}} \right) = b_1 \cdot A_1 \cdot (1-A_1) = b_1 A_1 \cdot A_2$

& $\frac{\partial A_2}{\partial a_1} = \frac{\partial}{\partial a_1} \left(\frac{1}{1+e^{-b_1(x_1-a_1)}} \right) = -b_1 \cdot A_2 (1-A_2) = -b_1 A_1 A_2$.

Now, $$\frac{\partial E}{\partial A_1} = \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial A_1}$$

$$= -(t-y) \cdot \frac{\partial y}{\partial A_1}$$

$$y = \bar{w}_1 \ y_1 + \bar{w}_2 \ y_2 + \bar{w}_3 \ y_3 + \bar{w}_4 \ y_4$$

$$= \bar{w}_1(A_1) \ y_1 + \bar{w}_2(A_1) \ y_2 + \bar{w}_3(A_1) \ y_3 + \bar{w}_4(A_1) y_4$$

$$\therefore \quad \frac{\partial y}{\partial A_1} = \sum_{i=1}^{4} \frac{\partial \bar{w}_i}{\partial A_1} y_i \, .$$

where $$\frac{\partial \bar{\omega}_1}{\partial A_1} = \frac{\partial}{\partial A_1}\left[\frac{\omega_1}{\omega_1+\omega_2+\omega_3+\omega_4}\right] = \frac{\partial}{\partial A_1}\left[\frac{A_1 B_1}{A_1 B_1 + A_1 B_2 + A_2 B_1 + A_2 B_2}\right]$$

$$= \frac{B_1 (A_1 B_1 + A_1 B_2 + A_2 B_1 + A_2 B_2) - (B_1 + B_2) A_1 B_1}{(A_1 B_1 + A_1 B_2 + A_2 B_1 + A_2 B_2)^2}$$

$$-\eta \frac{\partial E}{\partial A_1} \cdot \frac{\partial A_1}{\partial a_1} = \eta (T-y) \left(\sum_{i=1}^{4} \frac{\partial \bar{\omega}_i}{\partial A_1} y_i\right) \cdot b_1 \cdot A_1 \cdot A_2$$

Similarly,

$$-\eta \frac{\partial E}{\partial A_2} \cdot \frac{\partial A_2}{\partial a_1} = -\eta (T-y) \left(\sum_{i=1}^{4} \frac{\partial \bar{\omega}_i}{\partial A_2} y_i\right) b_1 \cdot A_1 \cdot A_2$$

$$\therefore \quad \Delta a_1 = \eta (T-y) \cdot b_1 \cdot A_1 \cdot A_2 \left[\sum_{i=1}^{4} \left(\frac{\partial \bar{\omega}_i}{\partial A_1} - \frac{\partial \bar{\omega}_i}{\partial A_2} \right) y_i \right]$$

$$\Delta a_1 = \eta (T-y) \cdot b_1 \cdot A_1 \cdot A_2 \frac{B_1 y_1 + B_2 y_2 + B_1 y_3 + B_2 y_4}{A_1 B_1 + A_1 B_2 + A_2 B_1 + A_2 B_2}$$

Similarly, $\Delta b_1, \Delta a_2, \Delta b_2$ can be computed

Pseudoinverse Method

- Consequent parameters can be updated at one go in a batch mode

$$y = \frac{w_1 y_1 + w_2 y_2 + w_3 y_3 + w_4 y_4}{w_1 + w_2 + w_3 + w_4}$$

$$= \bar{w}_1 y_1 + \bar{w}_2 y_2 + \bar{w}_3 y_3 + \bar{w}_4 y_4$$

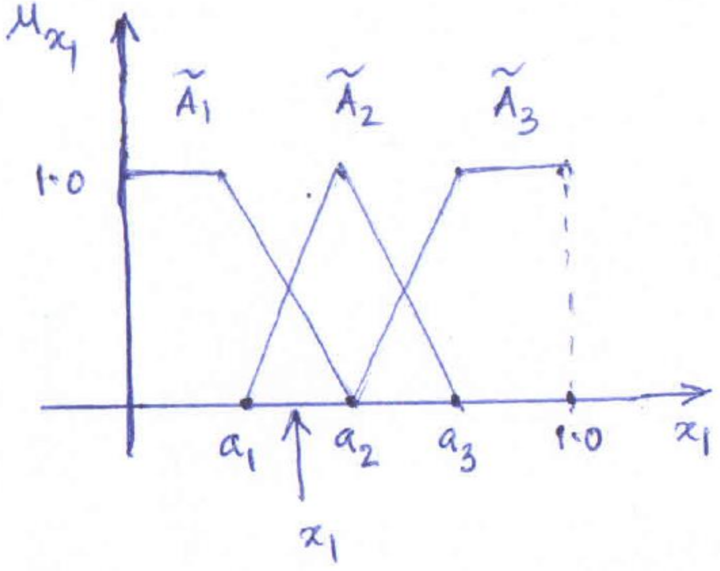
$$y = \begin{bmatrix} \bar{w}_1 x_1 & \bar{w}_1 x_2 & \bar{w}_1 & \bar{w}_2 x_1 & \bar{w}_2 x_2 & \bar{w}_2 & \bar{w}_3 x_1 & \bar{w}_3 x_2 & \bar{w}_3 & \bar{w}_4 x_1 & \bar{w}_4 x_2 & \bar{w}_4 \end{bmatrix} \begin{bmatrix} c_{11} \\ c_{12} \\ c_{13} \\ c_{21} \\ c_{22} \\ c_{23} \\ c_{31} \\ c_{32} \\ c_{33} \\ c_{41} \\ c_{42} \\ c_{43} \end{bmatrix}$$

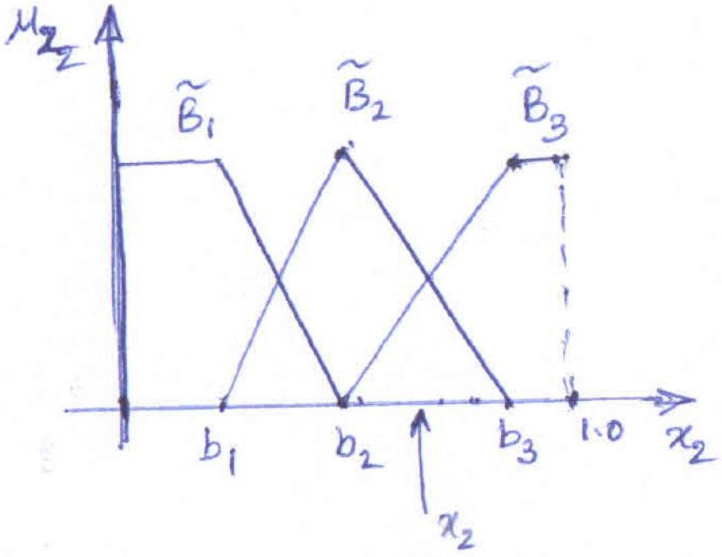
$$T_1 = p_1 \, \underline{c}$$

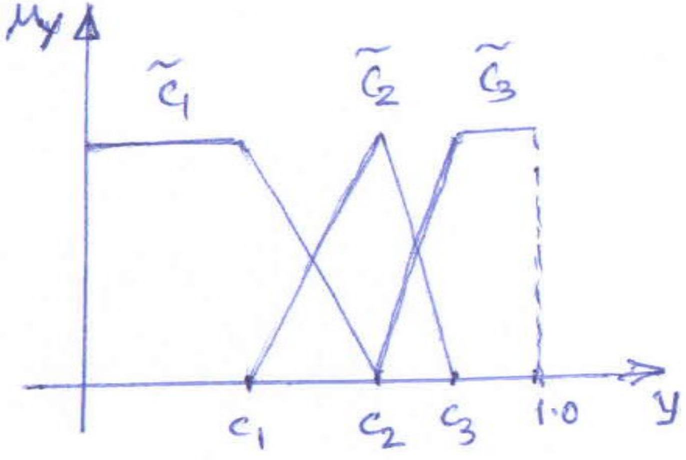
$$\begin{bmatrix} T_1 \\ \vdots \\ T_N \end{bmatrix} = \begin{bmatrix} p_1 \\ \vdots \\ p_N \end{bmatrix} \underline{c}$$

$$\underline{c} = pinv\left(\begin{bmatrix} p_1 \\ \vdots \\ p_N \end{bmatrix}\right) \begin{bmatrix} T_1 \\ \vdots \\ T_N \end{bmatrix}$$: least square error solution

Genetic Neuro-Fuzzy System Based on Mamdani Inference

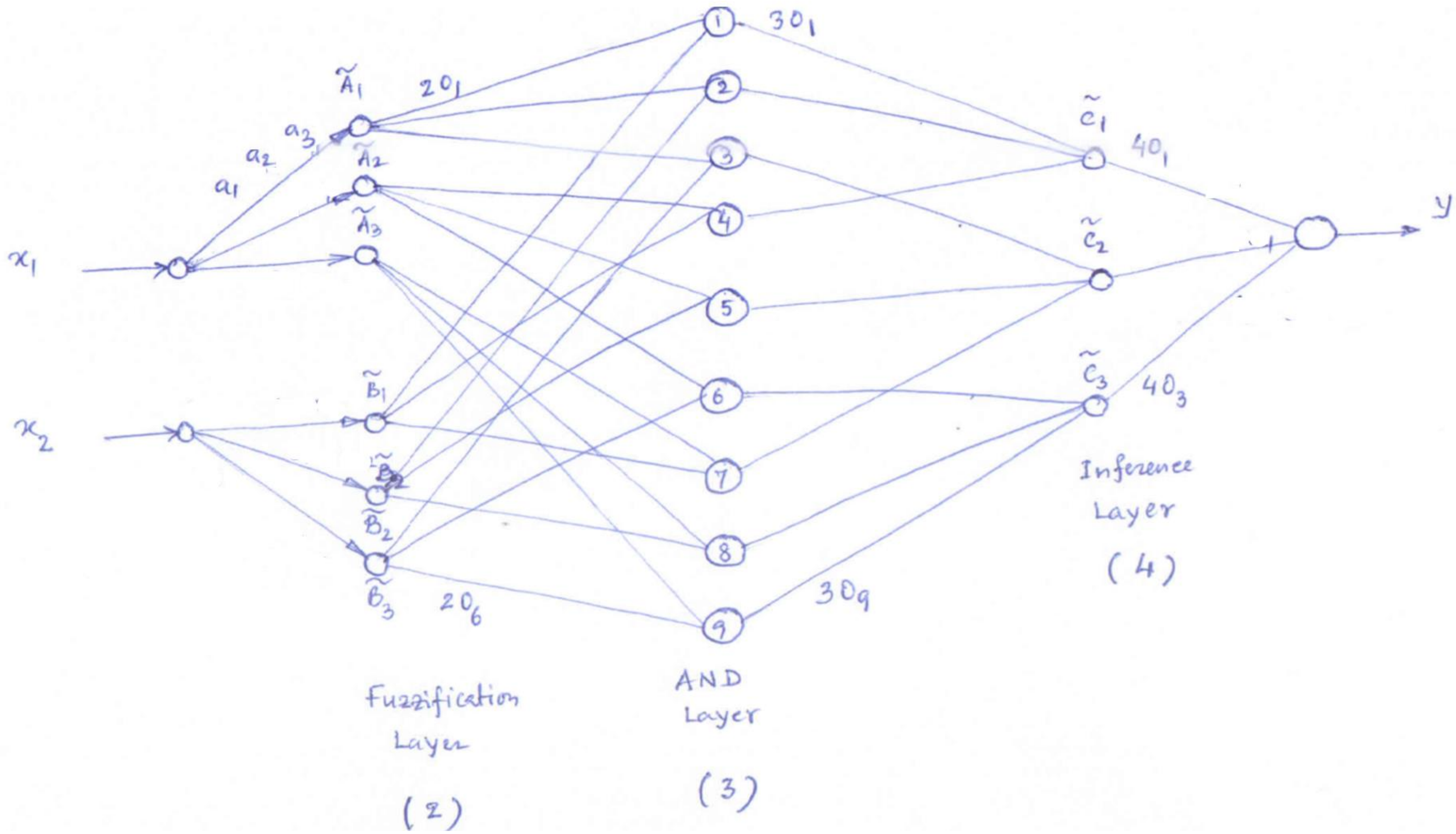






x_1 \ x_2	$\tilde{B}_1$	$\tilde{B}_2$	$\tilde{B}_3$		
$\tilde{A}_1$	$\tilde{C}_1$	$\tilde{C}_1$	$\tilde{C}_2$	R_2	R_3
$\tilde{A}_2$	$\tilde{C}_1$	$\tilde{C}_2$	$\tilde{C}_3$	R_5	R_6
$\tilde{A}_3$	$\tilde{C}_2$	$\tilde{C}_3$	$\tilde{C}_3$		

Sl. No.	x_1	x_2	~~y~~ T
1.	3.5	−4.5	−8.2
2.	⋮	⋮	⋮
⋮			
N_1			



x1
x2
a1
a2
a3
Ã1
Ã2
Ã3
B̃1
B̃2
B̃3
2O1
2O6
1
2
3
4
5
6
7
8
9
3O1
3O9
C̃1
C̃2
C̃3
4O1
4O3
y
Fuzzification Layer
(2)
AND Layer
(3)
Inference Layer
(4)

$$20_1 = \mu_{\tilde{A}_1}(a_1, a_2, a_3)$$

$$20_2 = \mu_{\tilde{A}_2}(a_1, a_2, a_3)$$

$$20_3 = \mu_{\tilde{A}_3}(a_1, a_2, a_3)$$

$$20_4 = \mu_{\tilde{B}_1}(b_1, b_2, b_3)$$

$$20_5 = \mu_{\tilde{B}_2}(b_1, b_2, b_3)$$

$$20_6 = \mu_{\tilde{B}_3}(b_1, b_2, b_3)$$

$$30_1 = \min(20_1, 20_4)$$

$$30_2 = \min(20_1, 20_5)$$

$$30_3 = \min(20_1, 20_6)$$

$$\vdots$$

$40_1 = A_1, y_1$ (both $f^{\underline{n}}$ of c_1, c_2, c_3)

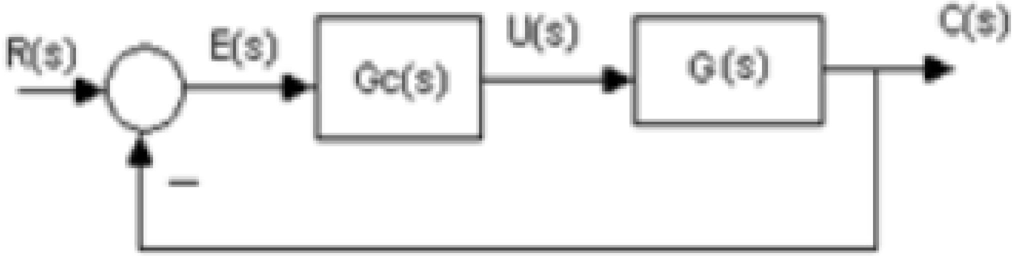
$$40_2 = A_2, y_2, A_3, y_3$$

$$40_3 = A_4, y_4$$

$$y = \frac{A_1 y_1 + A_2 y_2 + A_3 y_3 + A_4 y_4}{A_1 + A_2 + A_3 + A_4}.$$

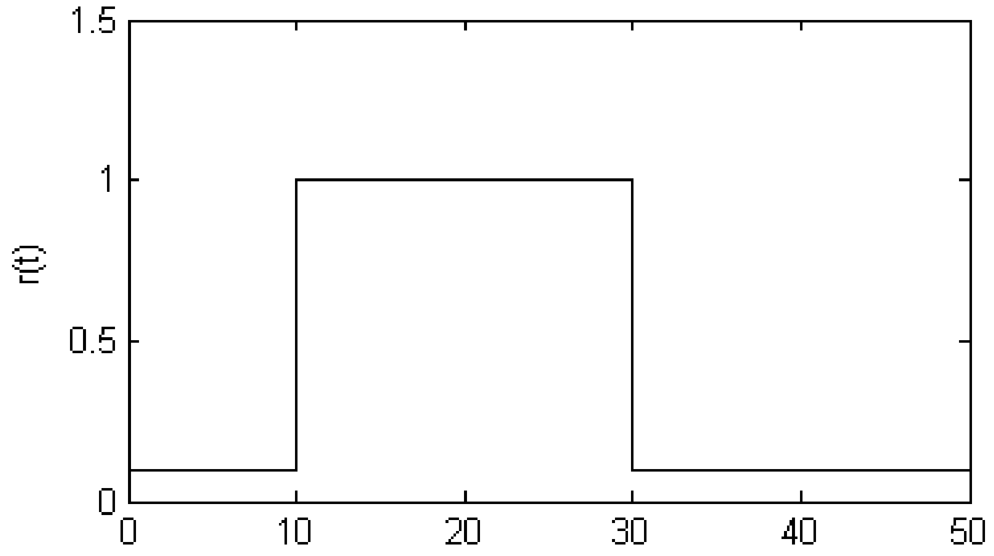
Fuzzy PID Control

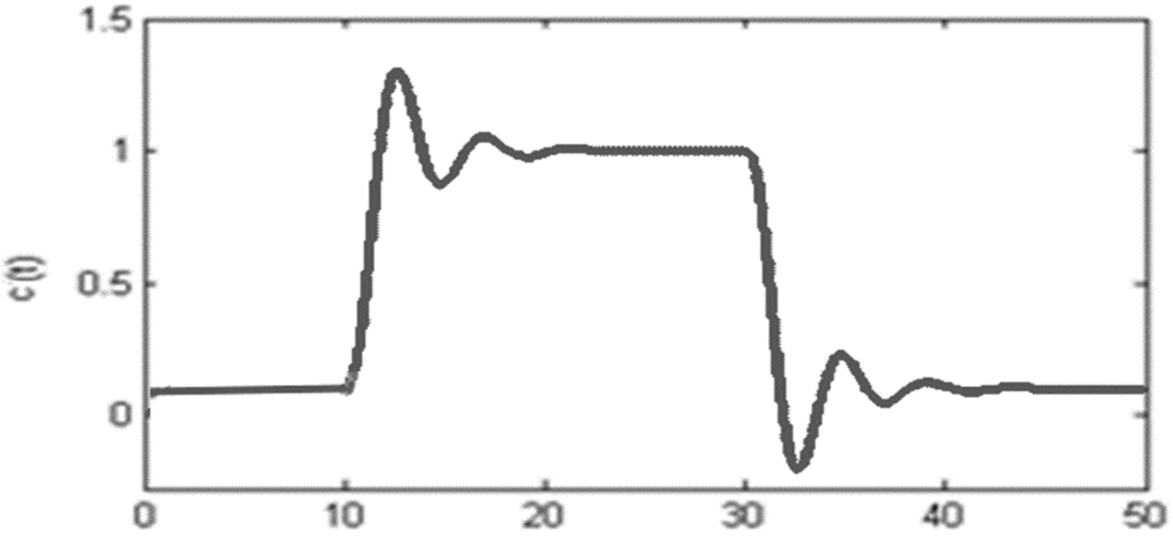
A Primer on PID Control



- ✓ Nicholas Minorsky in 1922
- ✓ Became well known from late 1930’s
- ✓ Intuitive
- ✓ No guarantee of stability
- ✓ Linear Controller
- ✓ Three design parameters
- ✓ They can be set analytically or experimentally or through trial and error

$$u(t) = K_P e(t) + K_I \int_0^t e(t)dt + K_D \frac{de(t)}{dt}$$





✓ Some Requirements:

- Fast response
- Low overshoot/undershoot
- Oscillations dying out fast
- Zero or low steady state error

✓ Other Requirements:

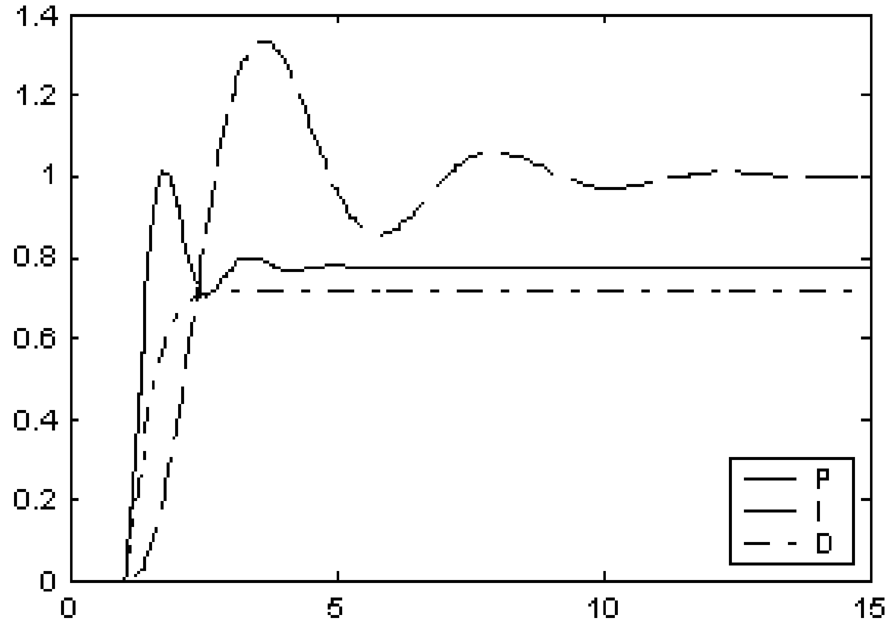
- Robustness to disturbance and plant uncertainties
- Robust stability
- Robust performance
- Handling nonlinearities, time delays

- ✓ P-term: $u(t) = K_P e(t)$
  - Control effort is proportional to instantaneous error
  - Makes the response faster
  - Usually leaves some steady state error
  - Higher K_P may reduce steady state error but at the cost of higher overshoot
  - Very high K_P may even lead to instability
- ✓ I-term: $u(t) = K_I \int_0^t e(t)dt$
  - Control effort is proportional to accumulation of error over time
  - Reduces steady state error to a great extent
  - Response becomes more oscillatory
  - Closed loop system becomes more prone to instability

✓ D-term: $$u(t) = K_D \frac{de(t)}{dt}$$

- Control action is proportional to inertia
- Reduces overshoot
- Takes no action if there is steady state error

  (always used along with a P-controller)



✓ PI, PD, PID Controllers

Discretization of a PID Controller

$$\dot{u}(t) = K_P \dot{e}(t) + K_I e(t) + K_D \ddot{e}(t)$$
$$= K_P\, v(t) + K_I\, d(t) + K_D\, a(t)$$

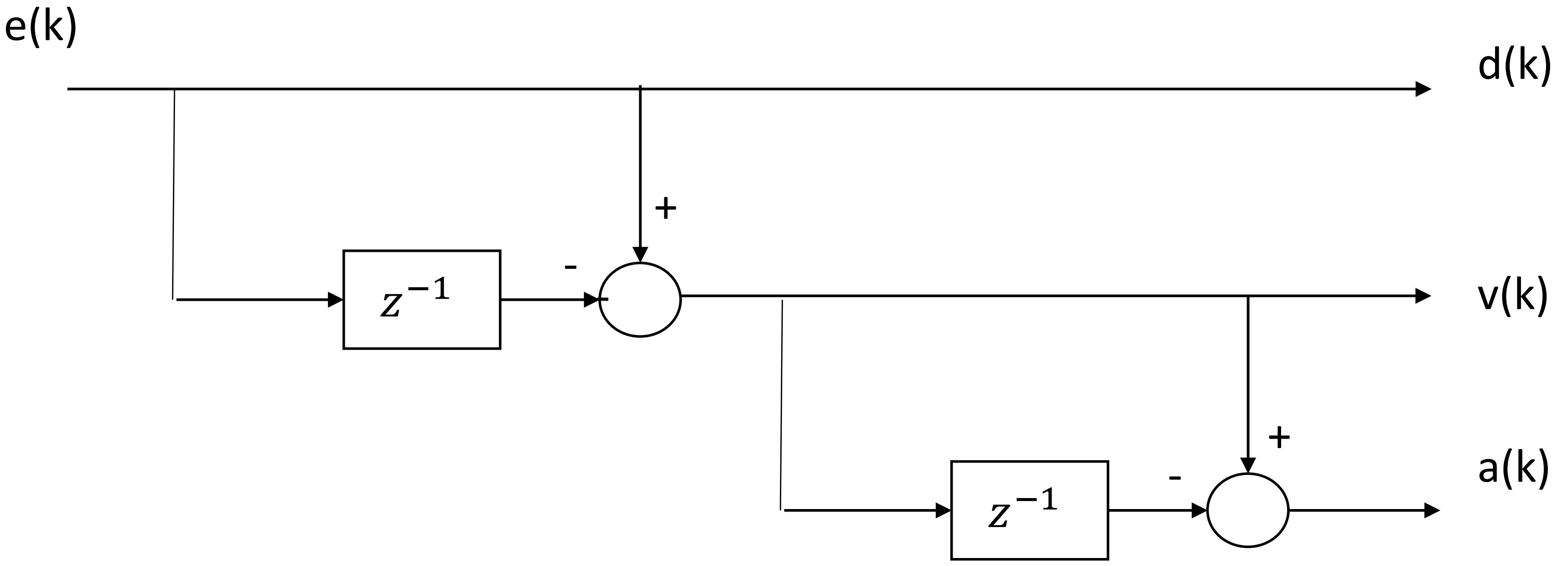
$$\frac{u(k) - u(k-1)}{T_s} = K_P\, v(k) + K_I\, d(k) + K_D\, a(k)$$
$$\Delta u(k) = K_I\, d(k) + K_P\, v(k) + K_D\, a(k)$$

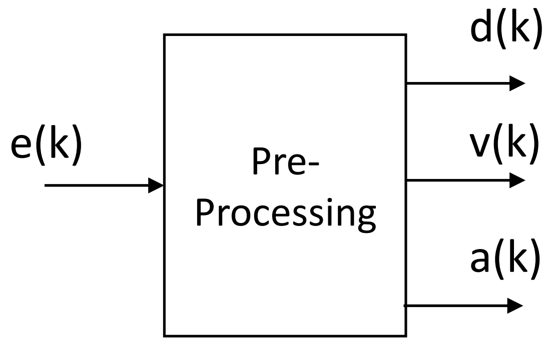
where $d(k) = error$

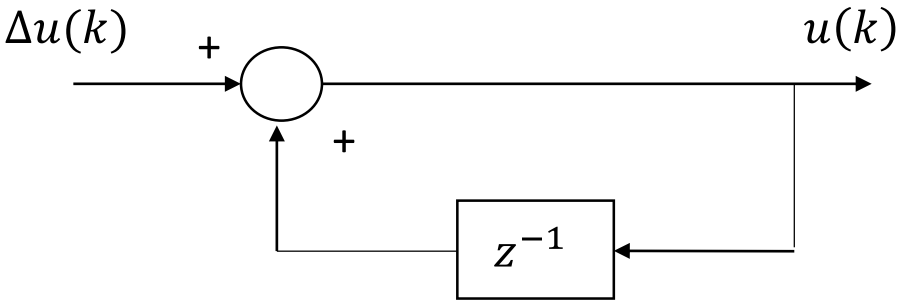
$v(k) = change\ in\ error$

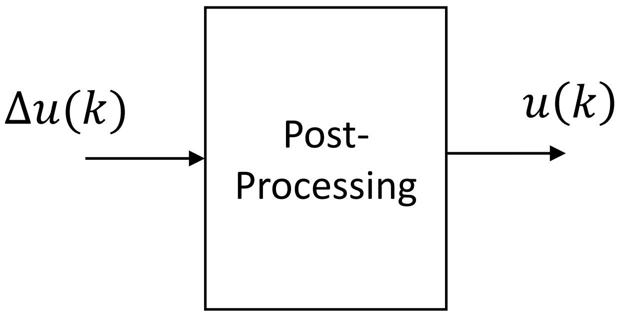
$a(k) = change\ in\ change\ in\ error$

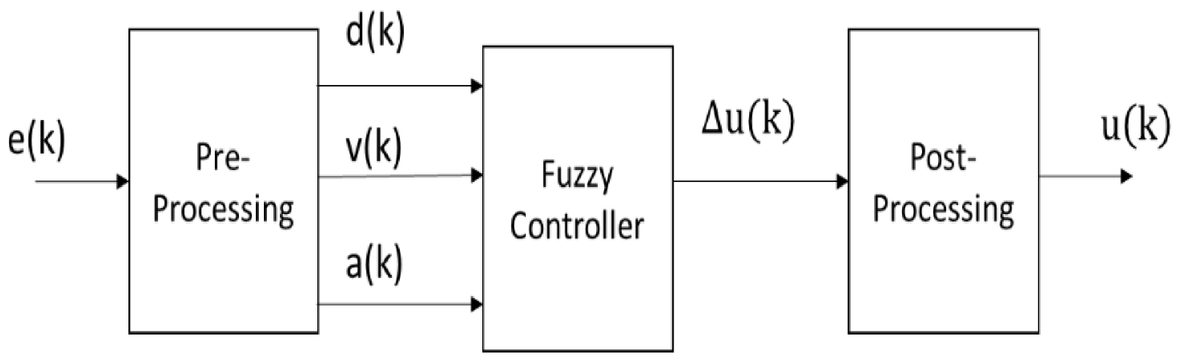
$$u(k) = u(k-1) + \Delta u(k)$$

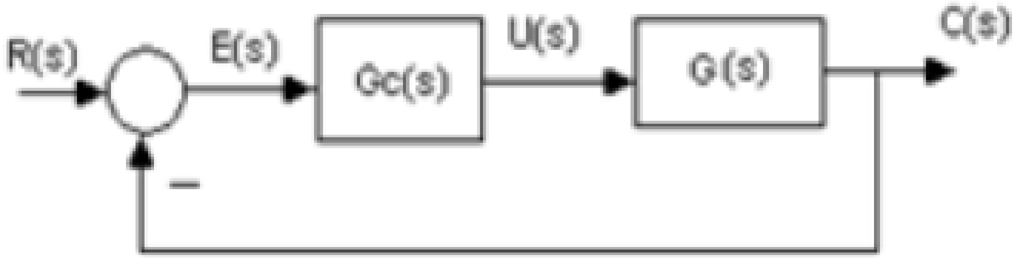












Advantages of Fuzzy PID

- ✓ A relatively easy way of designing nonlinear PID Control
- ✓ Bypassing the rigor of nonlinear control theory
- ✓ Without much dependence on the mathematical model of the plant

Fuzzy PI Controller

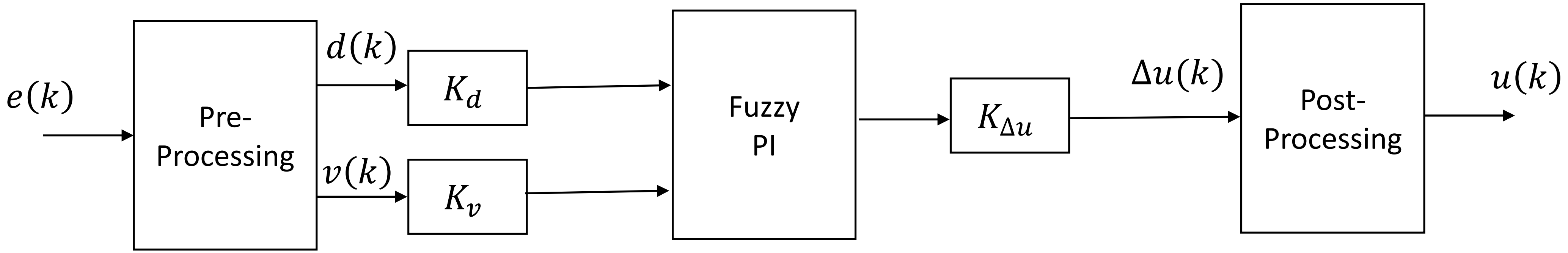
$$u(t) = K_P\, e(t) + K_I \int e(t)dt$$

$$\dot{u}(t) = K_P \dot{e}(t) + K_I e(t)$$

$$\frac{u(k)-u(k-1)}{T_S} = K_P \frac{e(k)-e(k-1)}{T_S} + K_I\, e(k)$$

$$\Delta u(k) = K_I\, e(k) + K_P \Delta e(k)$$

$$\Delta u(k) = f\{\, d(k), v(k)\}$$



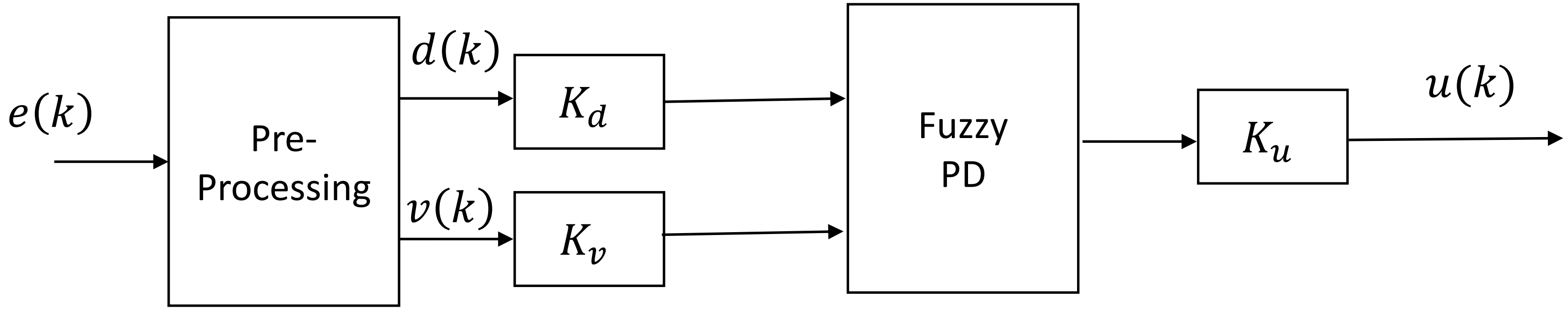
Fuzzy PD Controller

$$u(t) = K_P\, e(t) + K_D\, \dot{e}(t)$$

$$u(k) = K_P\, e(k) + K_D\, \frac{e(k)-e(k-1)}{T_S}$$

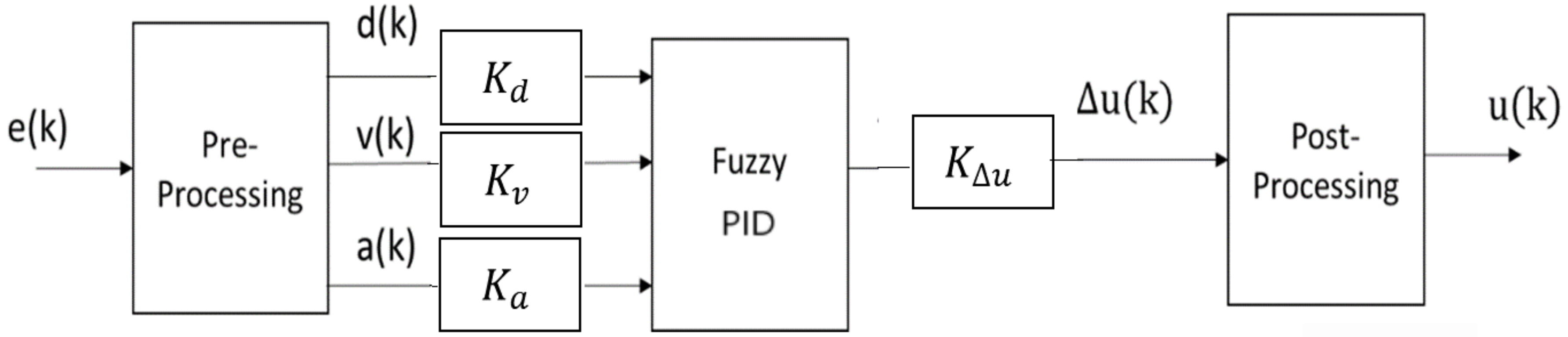
$$u(k) = K_P\, e(k) + K_D\, \Delta e(k)$$

$$u(k) = f\{\, d(k), v(k)\}$$



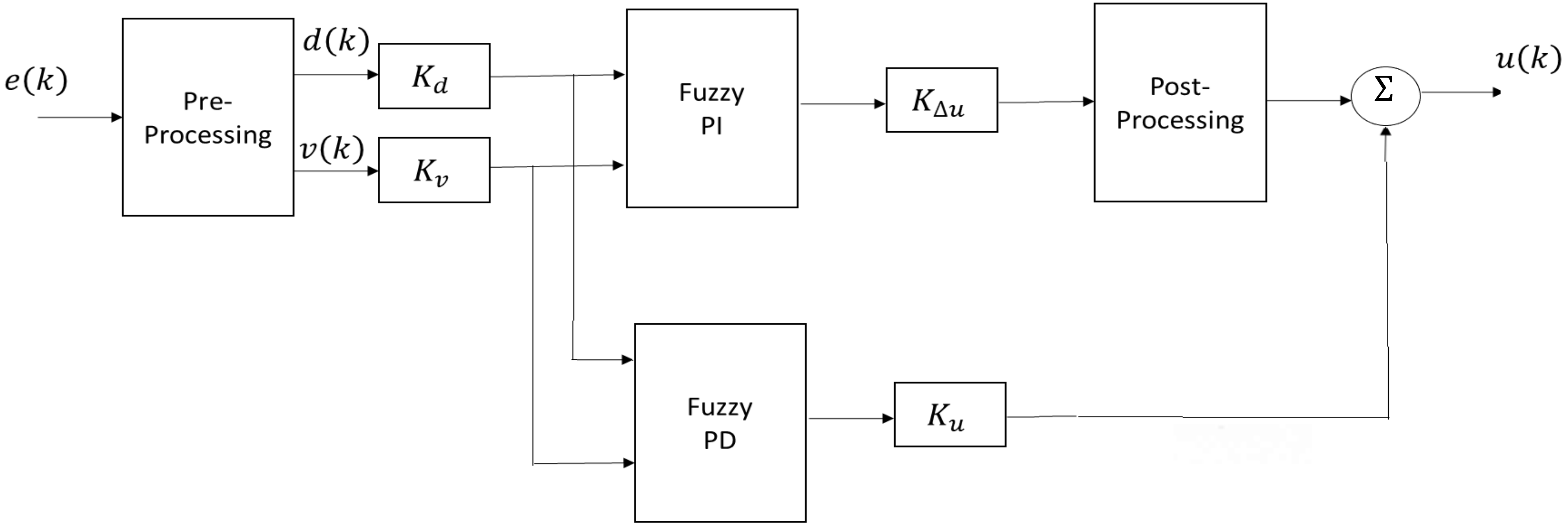
Three Term Fuzzy PID Controller

$$\Delta u(k) = f\{\, d(k), v(k), a(k)\}$$

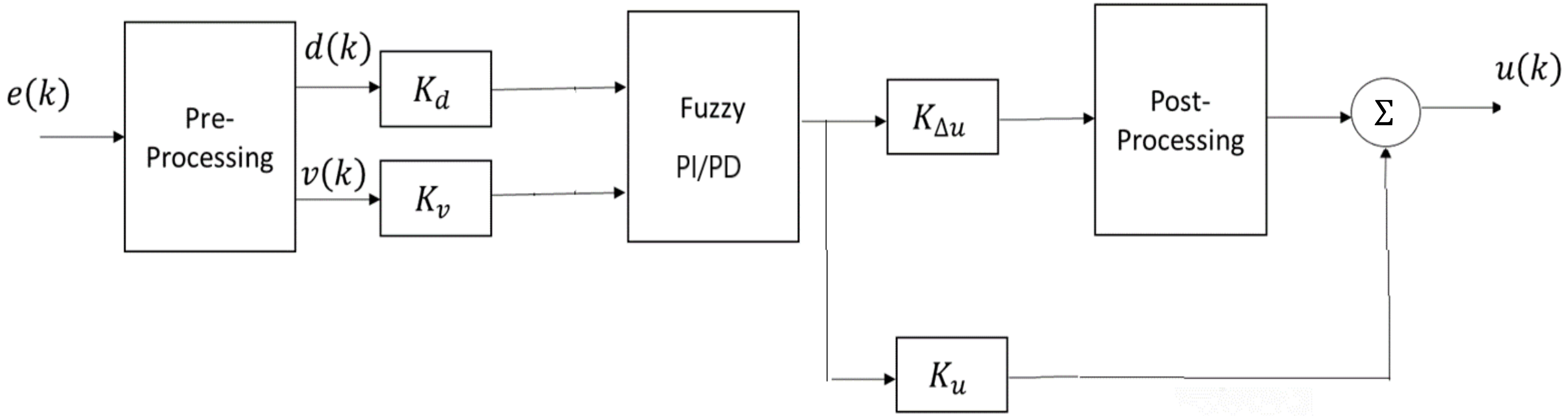


✓ Higher number of rules ($3^3 = 27$, $5^3 = 125$ etc.)

Two Term Fuzzy PID Controller

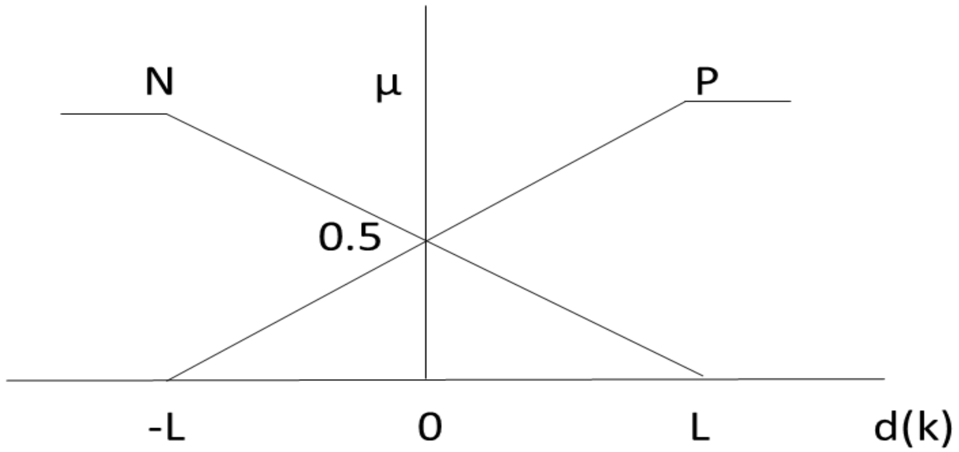


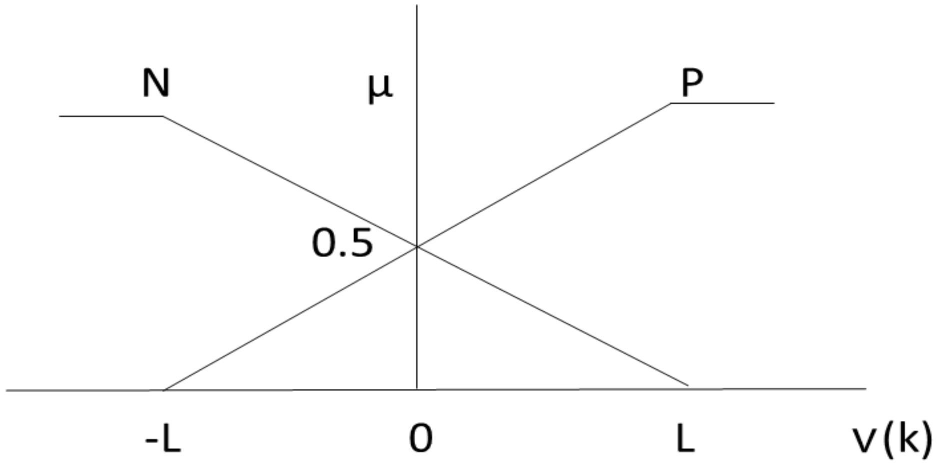
- ✓ Less number of rules ($2 \times 3^2 = 18$, $2 \times 5^2 = 50$ etc.)
- ✓ Scaling factors need to be properly tuned

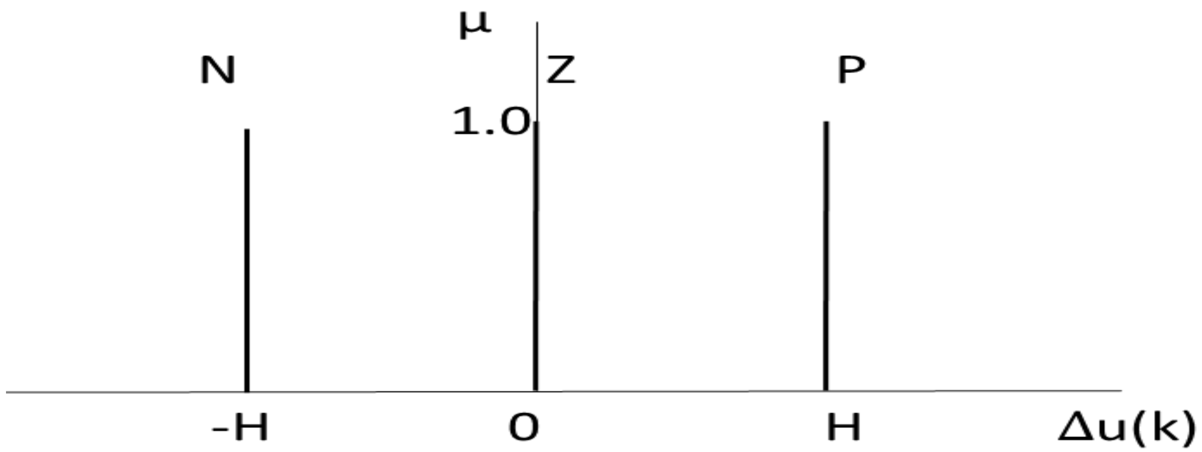


- ✓ Sometimes Fuzzy PI and Fuzzy PD may have identical rule base!
- ✓ No. of rules: 9, 25, …
- ✓ Many other structures are possible and proposed in the literature

A Fuzzy PI Controller as a Conventional Linear PI Controller







d(k) \ v(k)	P	N
P	P	Z
N	Z	N

$$\mu_P(d) = \begin{cases} 0 & ,\ d(k) < -L \\ \frac{K_d\ d+L}{2L} & ,\ -L <= d(k) <= L \\ 1 & ,\ d(k) > L \end{cases}$$

$$\mu_P(v) = \begin{cases} 0 & ,\ v(k) < -L \\ \frac{K_v\ v+L}{2L} & ,\ -L <= v(k) <= L \\ 1 & ,\ v(k) > L \end{cases}$$

$$\mu_N(d) = \begin{cases} 1 & ,\ d(k) < -L \\ \frac{-K_d\ d+L}{2L} & ,\ -L <= d(k) <= L \\ 0 & ,\ d(k) > L \end{cases}$$

$$\mu_N(v) = \begin{cases} 1 & ,\ v(k) < -L \\ \frac{-K_v\ v+L}{2L} & ,\ -L <= v(k) <= L \\ 0 & ,\ v(k) > L \end{cases}$$

- ✓ Mamdani model with singleton output sets (or zero order Sugeno model)
- ✓ Algebraic product t-norm

Weight of Rule1 = w1 = $\mu_P(d) * \mu_P(v)$

Weight of Rule2 = w2 = $\mu_P(d) * \mu_N(v)$

Weight of Rule3 = w3 = $\mu_N(d) * \mu_P(v)$

Weight of Rule4 = w4 = $\mu_N(d) * \mu_N(v)$

w1+w2+w3+w4 = $\mu_P(d) * \mu_P(v)+\mu_P(d) * \mu_N(v)+\mu_N(d) * \mu_P(v)+\mu_N(d) * \mu_N(v)$

= $\mu_P(d)\ \{\mu_P(v)+\mu_N(v)\} + \mu_N(d)\ \{\mu_P(v)+\mu_N(v)\}$

= $\mu_P(d)+\mu_N(d)$ =1

$$\Delta u(k) = K_{\Delta u} \frac{w1*H + w4*(-H)}{w1+w2+w3+w4}$$

$$= K_{\Delta u}\{\mu_P(d) * \mu_P(v) * H + \mu_N(d) * \mu_N(v) * (-H)\}$$

$$= K_{\Delta u}\left\{\frac{K_d\ d+L}{2L} * \frac{K_v\ v+L}{2L} * H\ - \frac{-K_d\ d+L}{2L} * \frac{-K_v\ v+L}{2L} * H\right\}$$

$$= K_{\Delta u}\{(K_d\ d + L) * (K_v\ v + L)\ - (-K_d\ d + L) * (-K_v\ v + L)\ \}\frac{H}{4L^2}$$

$$= \frac{K_{\Delta u}K_d\ H}{2L}\ d(k) + \frac{K_{\Delta u}K_v\ H}{2L}\ v(k)$$

$$[\ \Delta u(k) = K_I\ e(k) + K_P \Delta e(k)]$$

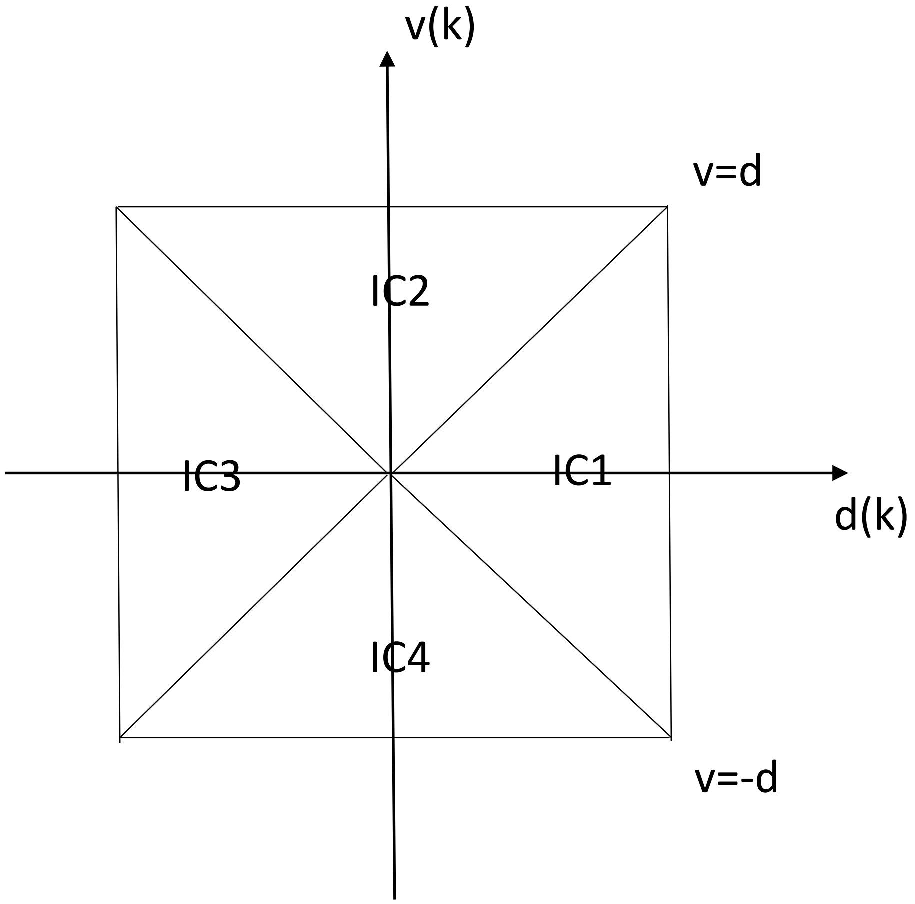
✓ Holds for Fuzzy PD Controller also (and therefore PID too)

A Fuzzy PI Controller as a Nonlinear PI Controller

- ✓ Minimum t-norm
- ✓ Singleton output sets
- ✓ Linear and symmetric membership functions for both d(k) and v(k)
- ✓ Mamdani/ Zero order Sugeno Inference

d(k) \ v(k)	P	N
P	P	Z
N	Z	N

IC: Input Combination

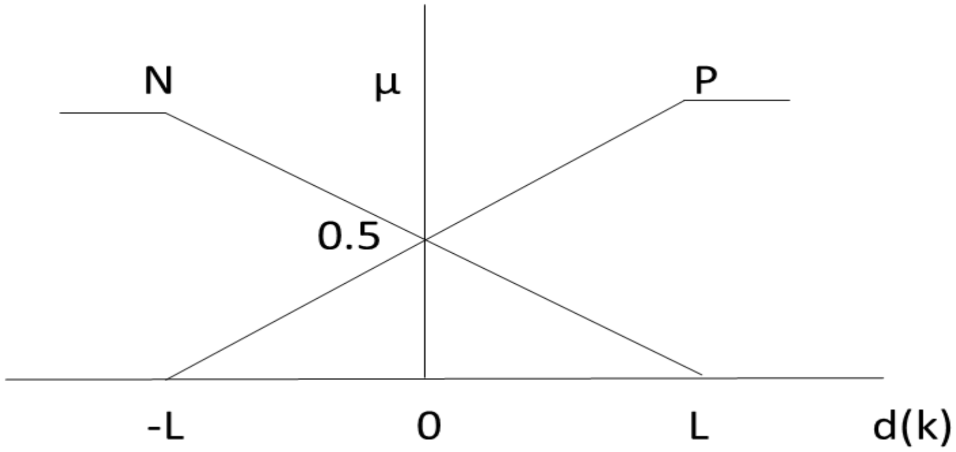


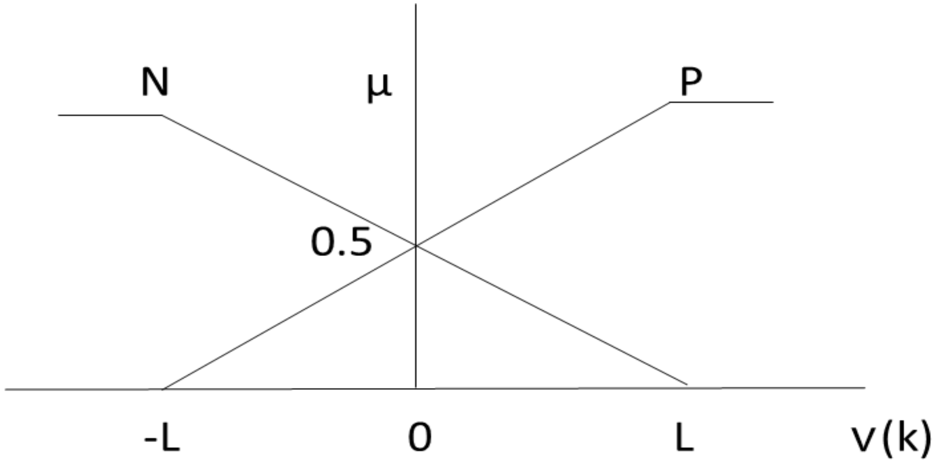
IC1: $0 \le d(k) \le L$ and $-\mathrm{L} \le v(k) \le L$

IC2: $-\mathrm{L} \le d(k) \le L$ and $0 \le v(k) \le L$

IC3: $-\mathrm{L} \le d(k) \le 0$ and $-\mathrm{L} \le v(k) \le L$

IC4: $-\mathrm{L} \le d(k) \le L$ and $-\mathrm{L} \le v(k) \le 0$





IC	Rule1	Rule2	Rule3	Rule4
IC1	$\mu_P(v)$	$\mu_N(v)$	$\mu_N(d)$	$\mu_N(d)$
IC2	$\mu_P(v)$	$\mu_N(v)$	$\mu_N(d)$	$\mu_N(d)$
IC3	$\mu_P(v)$	$\mu_N(v)$	$\mu_N(d)$	$\mu_N(d)$
IC4	$\mu_P(v)$	$\mu_N(v)$	$\mu_N(d)$	$\mu_N(d)$

Output for IC1 (and IC3):

$$\Delta u(k) = K_{\Delta u} \frac{\mu_P(v) * H + \mu_N(d) * (-H)}{\mu_P(v) + \mu_N(v) + \mu_N(d) + \mu_N(d)}$$

$$= K_{\Delta u} \frac{\mu_P(v) - \mu_N(d)}{1 + 2\, \mu_N(d)}\, H$$

$$= K_{\Delta u} \frac{\frac{K_v\, v + L}{2L} - \frac{-K_d\, d + L}{2L}}{1 + 2 \frac{-K_d\, d + L}{2L}} H$$

$$= K_{\Delta u} \frac{K_d\, d + K_v\, v}{2L + 2(-K_d\, d + L)}\, H$$

$$= \frac{K_{\Delta u}\, H}{4L - 2\, K_d\, d}\, (K_d\, d \;+\; K_v\, v)$$

$$= \frac{0.5\, K_{\Delta u}\, H}{2L - K_d\, d}\, (K_d\, d \;+\; K_v\, v)$$

Output for IC2 and IC4:

$$\Delta u(k) = \frac{0.5\, K_{\Delta u}\, H}{2L - K_v\, v}\, (K_d\, d + K_v\, v)$$

- ✓ Controller Gains are error or error rate dependent
- ✓ Higher gains for higher error or rate
- ✓ Faster convergence with less overshoot even for linear systems
- ✓ Gains vary smoothly across various regions (i.e. ICs)
- ✓ Larger no. of controller parameters
- ✓ Similar for Fuzzy PD Controller (and therefore PID controller)

Fuzzy Clustering

- **What is Clustering?**

  - It deals with finding a structure in a collection of unlabeled data

  - This is an unsupervised study where data of similar types are put into one cluster while data of another types are put into different cluster

  - A good clustering method will produce high quality clusters with

    - HIGH INTRA-CLASS SIMILARITY
      (Similar to one another within the same cluster)

    - LOW INTER-CLASS SIMILARITY
      (Dissimilar to the objects in other clusters)

- The <u>quality</u> of a clustering method is also measured by its ability to discover some or all of the <u>hidden</u> patterns
- Lot of applications in image processing, medical diagnosis etc.

Clustering can be classified as:

- Hard Clustering (or Exclusive Clustering)
- Soft Clustering (Overlapping Clustering)

- K-means is an important hard clustering technique
- Fuzzy C Means (FCM) is a very popular soft clustering technique

K-Means Clustering

- ✓ K means clustering cluster the entire dataset into K number of clusters where a data should belong to only one cluster

- ✓ Membership or partition matrix U is of the form:

$$U = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ . & . \\ . & . \\ 0 & 1 \end{bmatrix}$$

- ✓ One of the simplest Similarity measure is “Euclidean distance” between pairs of feature vectors in the feature space
  (Distance between the points in the same cluster will be considerably less than distance between points in different clusters)

- ✓ The first step is normalization of the data. This step is very important when dealing with parameters of different units and scales.

$$X_{normalized[0,1]} = \frac{X_{current} - X_{\min}}{X_{\max} - X_{\min}}$$

$$X_{normalized[-1,\,1]} = \frac{X_{current} - (X_{\max} + X_{\min})/2}{(X_{\max} - X_{\min})/2}$$

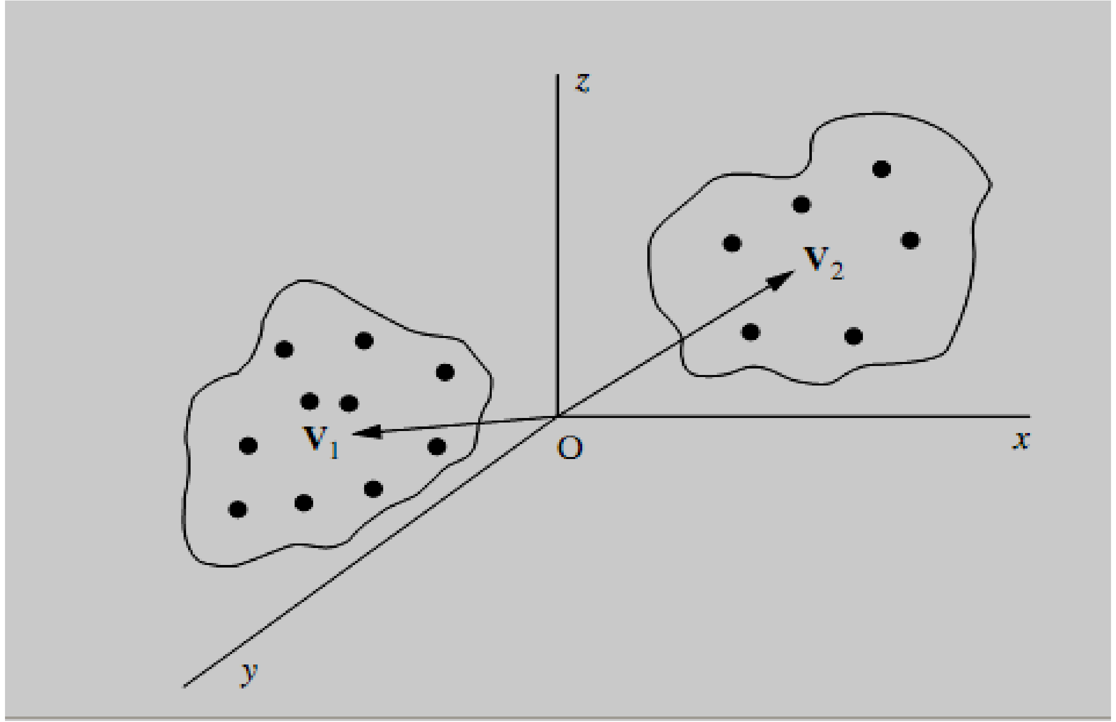
- ✓ In case there are outliers in the data, we prefer standardization (When using standardization, your new data aren't bounded unlike normalization)

$$x_{new} = \frac{x - \mu}{\sigma}$$

Objective function is developed to

1. Minimize Euclidian distance between each data points in a cluster and its cluster centre

2. Maximize the Euclidian distance between cluster centers (centroids)



Algorithmic Steps for K-Means Clustering:

1) Set K– choose a number of desired clusters, k

2) Initialization – choose k starting points which are used as initial estimates of the cluster centroids

3) Classification – Examine each point in the dataset and assign it to the cluster whose centroid is nearest to it.

4) Centroid calculation – When each point in the data set is assigned to a cluster,

it is needed to recalculate the new k centroids.

This is done by taking average of the members of the particular cluster

5) Convergence criteria – Steps 3 & 4 of to be repeated until no point changes

its cluster assignment or until the centroids no longer move

A Numerical Example: two dimensional feature space

(using K = No. of clusters = 2)

Individual	Variable 1	Variable 2
1	1	1
2	1.5	2
3	3	4
4	5	7
5	3.5	5
6	4.5	5
7	3.5	4.5

<u>Step 1:</u>

<u>Initialization</u>: Randomly choose following two centroids (k=2) for two clusters as

m1=(1.0,1.0) and m2=(5.0,7.0)

Individual	Variable 1	Variable 2
1	1	1
2	1.5	2
3	3	4
4	5	7
5	3.5	5
6	4.5	5
7	3.5	4.5

Individual	Distance from Centroid 1 (1,1)	Distance from Centroid 2 (5,7)
1(1.0,1.0)	0	7.21
2(1.5,2)	1.12	6.1
3(3,4)	3.51	3.51
4(5,7)	7.21	0
5(3.5,5)	4.72	2.5
6(4.5,5)	5.31	2.06
7(3.5,4.5)	4.3	2.92

$$d(m_1,2) = \sqrt{|1-1.5|^2 + |1-2|^2} = 1.12$$

$$d(m_2,2) = \sqrt{|5-1.5|^2 + |7-2|^2} = 6.1$$

Distance of point/individual 3 is same from both centroids

$$d(m_2,1) = \sqrt{(|5-1|^2 + |7-1|^2} = 7.21$$

$$d(m_1,6) = \sqrt{(|1-4.5|^2 + |1-5|^2} = 5.31$$

$$d(m_2,6) = \sqrt{|5-4.5|^2 + |7-5|^2} = 2.06$$

Individual	Distance from Centroid 1 (1,1)	Distance from Centroid 2 (5,7)
1(1.0,1.0)	0	7.21
2(1.5,2)	1.12	6.1
3(3,4)	3.51	3.51
4(5,7)	7.21	0
5(3.5,5)	4.72	2.5
6(4.5,5)	5.31	2.06
7(3.5,4.5)	4.3	2.92

Step 2: Thus, we obtain two clusters containing:

{1,2,3} and {4,5,6,7}. Their new centroids are:

$$m_1 = \frac{1}{3}(1.0 + 1.5 + 3.0), \frac{1}{3}(1.0 + 2.0 + 4.0) = (1.83, 2.33)$$

$$m_2 = \frac{1}{4}(5.0 + 3.5 + 4.5 + 3.5), \frac{1}{4}(7.0 + 5.0 + 5.0 + 4.5) = (4.12, 5.38)$$

Individual	Distance from Centroid 1 (1.83,2.33)	Distance from Centroid 2 (4.12,5.38)
1(1.0,1.0)	1.57	5.38
2(1.5,2)	0.47	4.28
3(3,4)	2.04	1.78
4(5,7)	5.64	1.84
5(3.5,5)	3.15	0.73
6(4.5,5)	3.78	0.54
7(3.5,4.5)	2.74	1.08

$$d(m_1,2) = \sqrt{|1.83-1.5|^2 + |2.33-2|^2} = 0.47$$

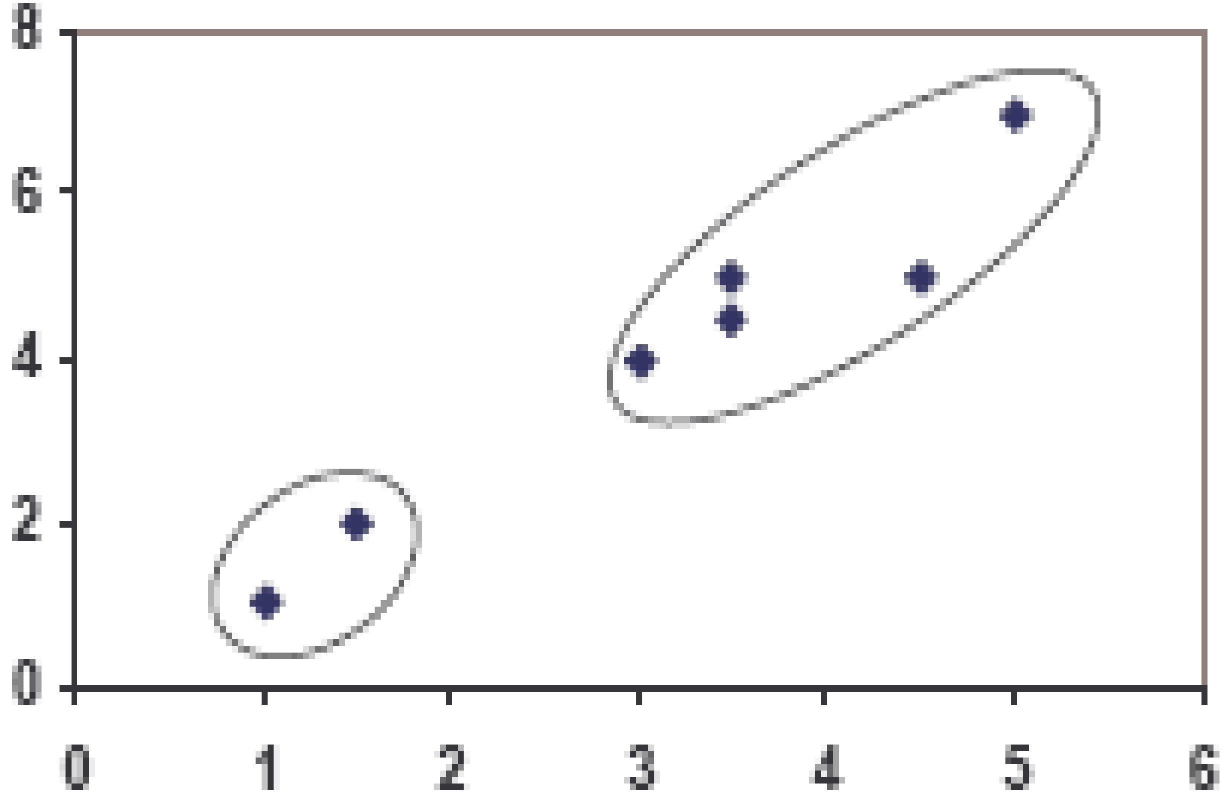
$$d(m_2,2) = \sqrt{|4.12-1.5|^2 + |5.38-2|^2} = 4.28$$

Individual	Distance from Centroid 1 (1.83,2.33)	Distance from Centroid 2 (4.12,5.38)
1(1.0,1.0)	1.57	5.38
2(1.5,2)	0.47	4.28
3(3,4)	2.04	1.78
4(5,7)	5.64	1.84
5(3.5,5)	3.15	0.73
6(4.5,5)	3.78	0.54
7(3.5,4.5)	2.74	1.08

Therefore, the new clusters are:{1,2} and {3,4,5,6,7}   Next centroids are:

$$m_1 = \frac{1}{2}(1.0+1.5), \frac{1}{2}(1.0+2.0) = (1.25,1.5)$$

$$m_1 = \frac{1}{5}(3+5.0+3.5+4.5+3.5), \frac{1}{5}(4+7.0+5.0+5.0+4.5) = (3.9,5.1)$$



8
6
4
2
0
0
1
2
3
4
5
6

(with K=3)

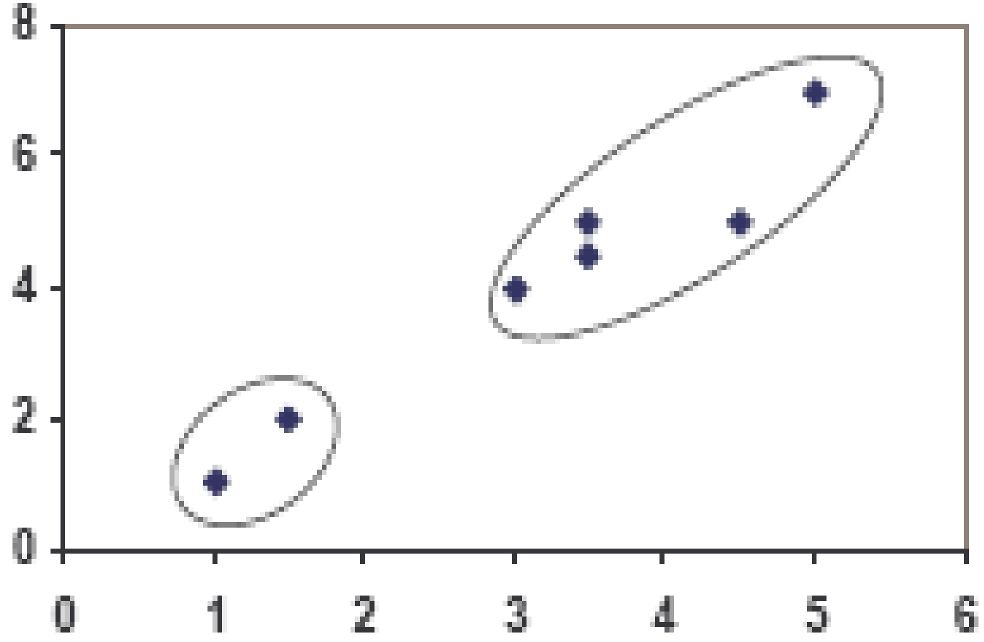
Individual	Variable 1	Variable 2
1	1	1
2	1.5	2
3	3	4
4	5	7
5	3.5	5
6	4.5	5
7	3.5	4.5

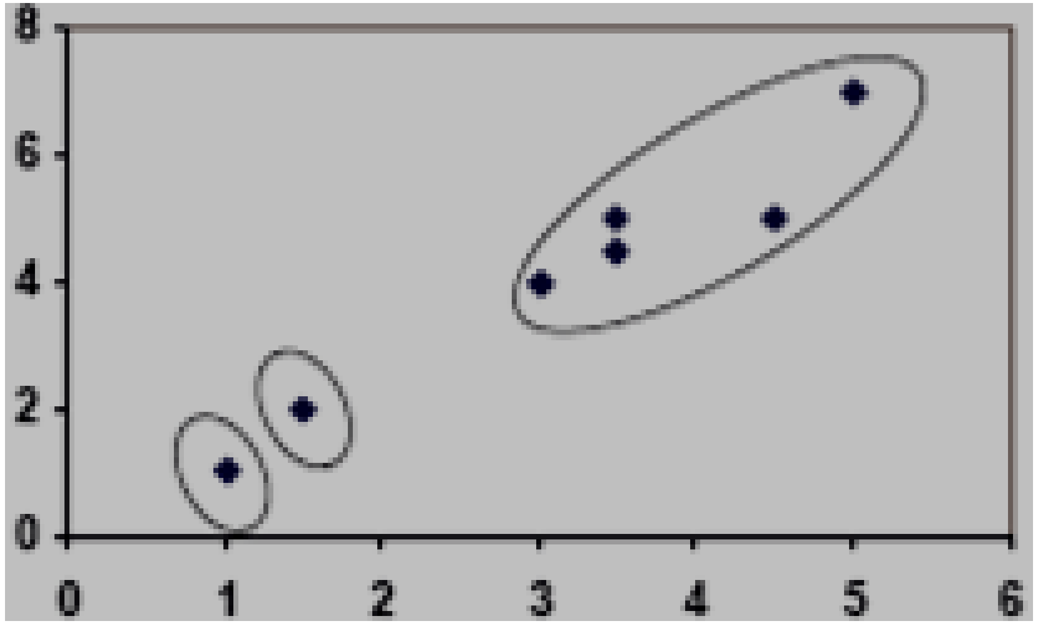
individual	Distance from cenroid 1(1,1)	Distance from cenroid 2(1.5,2)	Distance from cenroid 3(3,4)	Cluster
1(1,1)	0	1.11	3.61	1
2(1.5,2)	1.12	0	2.5	2
3(3,4)	3.61	2.5	0	3
4(5,7)	7.21	6.1	3.61	3
5(3.5,5)	4.72	3.61	1.12	3
6(4.5,5)	5.31	4.24	1.8	3
7(3.5,4.5)	4.3	3.2	0.71	3

New third centroid = (3+5+3.5+4.5+3.5)/5, (4+7+5+5+4.5)/5

= (3.9,5.1)

K=2





K=3

Some Features:

1) Apriori knowledge of number of clusters

   - may be problem specific

   - otherwise may be determined by trial and error

2) Sensitivity to initialization

   - suffers from local minima problem

   - initial centroids should be well spread out

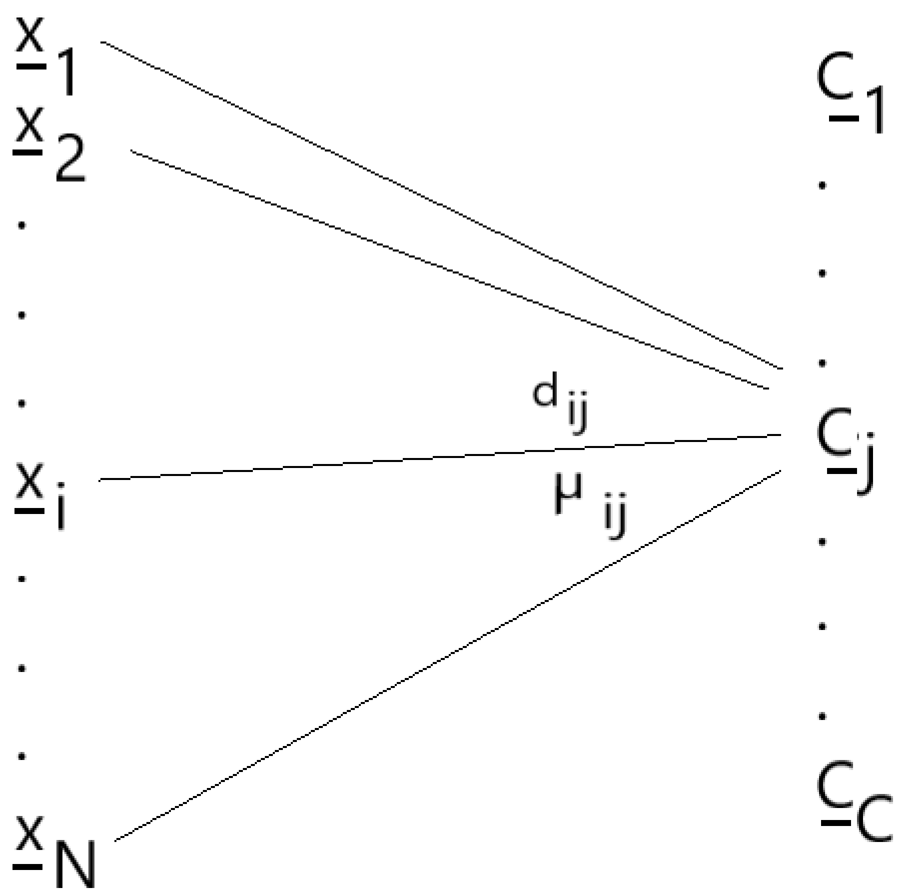
Fuzzy C-Means Clustering

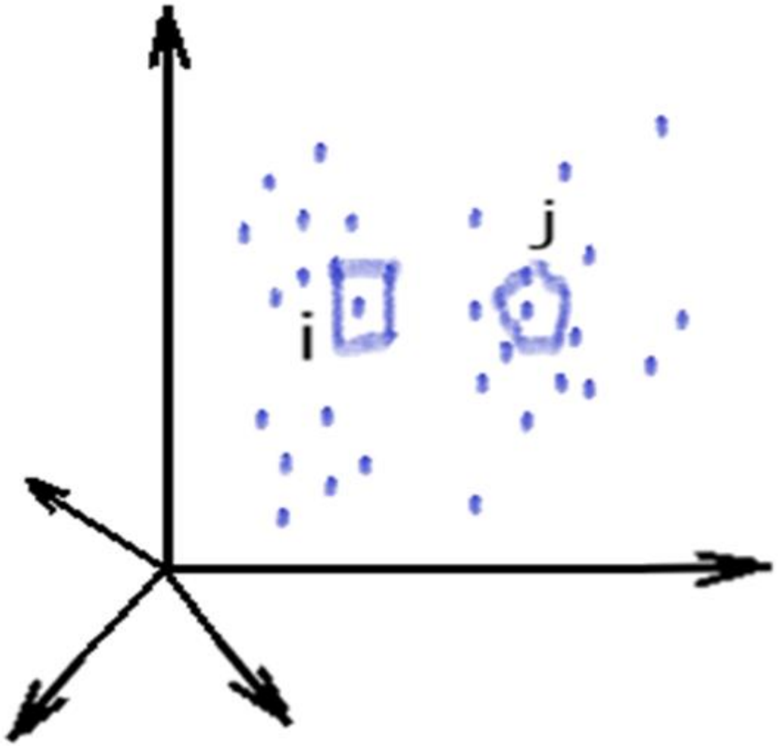
- ✓ Every point in the feature space belongs to all C clusters with varied membership grades

- ✓ Membership or partition matrix U takes the form:

$$U = \begin{bmatrix} 0.8 & 0.2 \\ 0.3 & 1.7 \\ 0.6 & 0.4 \\ . & . \\ . & . \\ 0.1 & 0.9 \end{bmatrix}$$

- ✓ Proposed by Bezdek in 1973





Dissimilarity, $F = \sum_{j=1}^{C} \sum_{i=1}^{N} \mu_{ij}^2 \, d_{ij}^2$ where $\sum_{j=1}^{C} \mu_{ij} = 1.0$

$$\mathcal{L} = \sum_{j=1}^{C} \sum_{i=1}^{N} \mu_{ij}^2 \left\| \underline{C}_j - \underline{x}_i \right\|^2 + \sum_{i=1}^{N} \lambda_i \left\{ \sum_{j=1}^{C} \mu_{ij} - 1.0 \right\}$$

$$\underline{C}_j = \frac{\sum_N \ \mu_{ij}^2 \ \underline{x}_i}{\sum_N \ \mu_{ij}^2}$$

$$\mu_{ij} = \frac{1}{\sum_C \left(\frac{d_{ij}}{d_{im}} \right)^2}$$ where, $$d_{ij} = \left\| \underline{C}_j - \underline{x}_i \right\|$$

Steps for FCM Algorithm:

1) Choose the number of clusters C

2) Choose a cluster fuzziness level (g > 1)

3) Initialize the membership or partition matrix U at random

(satisfying the constraint that row sum = 1)

4) Compute the cluster centers as

$$\underline{C}_j = \frac{\sum_N \mu_{ij}^{g} \underline{x}_i}{\sum_N \mu_{ij}^{2}}$$

5) Compute the Euclidean distance, $d_{ij} = \left\| \underline{C}_j - \underline{x}_i \right\|$

6) Update the membership matrix as

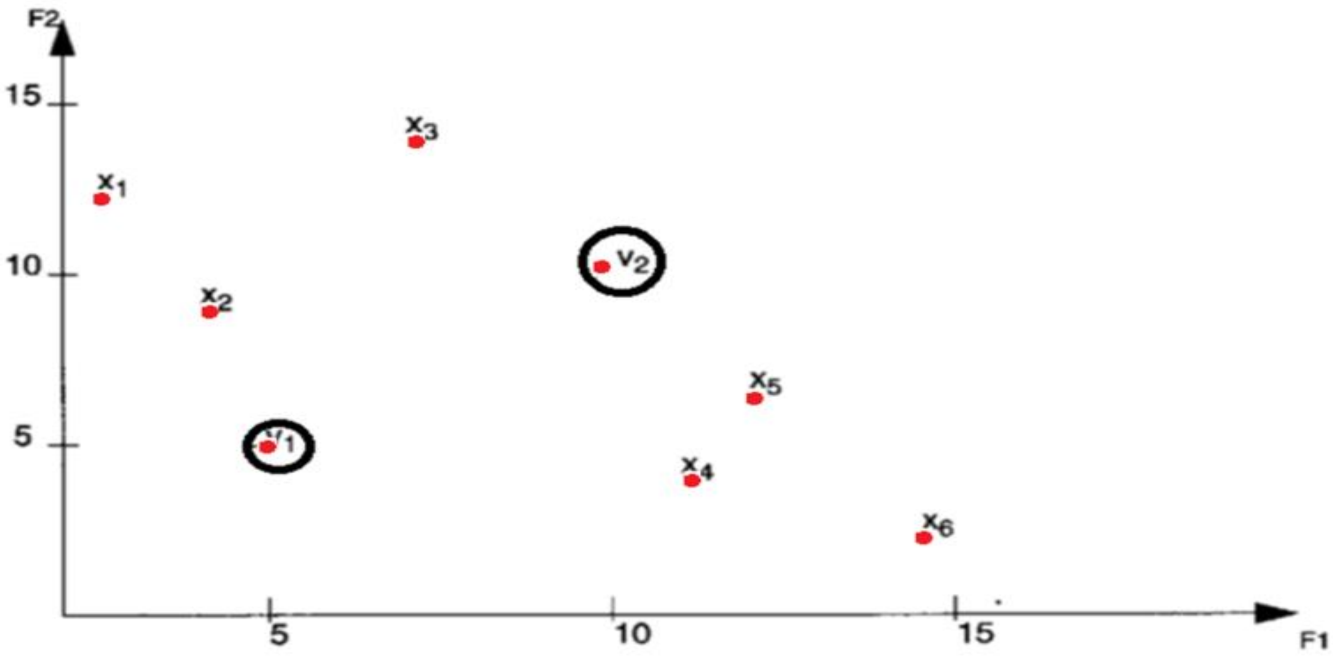
$$\mu_{ij} = \frac{1}{\sum_C \left(\frac{d_{ij}}{d_{im}} \right)^{2/(g-1)}}$$

if $d_{ij} = 0$, then $\mu_{ij} = 1$

7) Repeat steps 4, 5 and 6 till change in μ_{ij} values fall below a small value

Numerical Example:

Memebship value	X_1 (2,12)	X_2 (4,9)	X_3 (7,13)	X_4 (11,5)	X_5 (12,7)	X_6 (14,4)
C_1 (5,5)	0.5397	0.6852	0.2093	0.4194	0.197	0.3881
C_2 (10,10)	0.4603	0.3148	0.7907	0.5806	0.803	0.6119



$$
U = \begin{array}{cc} C1 & C2 \\ \end{array}
$$

$$
U = \begin{bmatrix} 0.5397 & 0.4603 \\ 0.6852 & 0.3148 \\ 0.2093 & 0.7907 \\ 0.4194 & 0.5806 \\ 0.1970 & 0.8030 \\ 0.3881 & 0.6119 \end{bmatrix}
$$

	X_1 (2,12)	X_2 (4,9)	X_3 (7,13)	X_4 (11,5)	X_5 (12,7)	X_6 (14,4)
C_1	**0.5397**	**0.6852**	**0.2093**	**0.4194**	**0.197**	**0.3881**
C_2	**0.4603**	**0.3148**	**0.7907**	**0.5806**	**0.803**	**0.6119**

Calculate coordinates of new centre of cluster 1

$$C_1 = \frac{\begin{array}{c}(0.5397)^2 \times (2,12) + (0.6852)^2 \times (4,9) + (0.2093)^2 \times (7,13) + \\ (0.4194)^2 \times (11,5) + (0.197)^2 \times (12,7) + (0.3881)^2 \times (14,4)\end{array}}{(0.5397)^2 + (0.6852)^2 + (0.2093)^2 + (0.4194)^2 + (0.197)^2 + (0.3881)^2}$$

$$= (6.6273, 9.1484)$$

	X_1 (2,12)	X_2 (4,9)	X_3 (7,13)	X_4 (11,5)	X_5 (12,7)	X_6 (14,4)
C_1	**0.5397**	**0.6852**	**0.2093**	**0.4194**	**0.197**	**0.3881**
C_2	**0.4603**	**0.3148**	**0.7907**	**0.5806**	**0.803**	**0.6119**

Calculate coordinates of new centre of cluster 2

$$C_2 = \frac{\begin{array}{c}(0.4603)^2 \times (2,12) + (0.3148)^2 \times (4,9) + (0.7907)^2 \times (7,13) + \\ (0.5806)^2 \times (11,5) + (0.803)^2 \times (12,7) + (0.6119)^2 \times (14,4)\end{array}}{(0.4603)^2 + (0.3148)^2 + (0.7907)^2 + (0.5806)^2 + (0.803)^2 + (0.6119)^2}$$

$$= (9.7374,\ 8.4887)$$

X_1 (2,12)	X_2 (4,9)	X_3 (7,13)	X_4 (11,5)	X_5 (12,7)	X_6 (14,4)

$$
\begin{array}{cc} & \begin{array}{cc} C1 & C2 \end{array} \\ U = & \begin{bmatrix} 0.5397 & 0.4603 \\ 0.6852 & 0.3148 \\ 0.2093 & 0.7907 \\ 0.4194 & 0.5806 \\ 0.1970 & 0.8030 \\ 0.3881 & 0.6119 \end{bmatrix} \end{array}
$$

$$\mu_{11} = \frac{1}{\sum_C \left(\frac{d_{ij}}{d_{im}}\right)^{2/(g-1)}} = \frac{1}{1 + \left(\frac{d_{11}}{d_{12}}\right)^2}$$

$$\mu_{21} = \frac{1}{\sum_C \left(\frac{d_{ij}}{d_{im}}\right)^{2/(g-1)}} = \frac{1}{1 + \left(\frac{d_{21}}{d_{22}}\right)^2}$$

$$\mu_{31} = \frac{1}{\sum_C \left(\frac{d_{ij}}{d_{im}}\right)^{2/(g-1)}} = \frac{1}{1 + \left(\frac{d_{31}}{d_{32}}\right)^2}$$

$$
U = \begin{array}{cc} C1 & C2 \\ \end{array}
$$

$$
U = \begin{bmatrix} 0.5397 & 0.4603 \\ 0.6852 & 0.3148 \\ 0.2093 & 0.7907 \\ 0.4194 & 0.5806 \\ 0.1970 & 0.8030 \\ 0.3881 & 0.6119 \end{bmatrix}
$$

$$
\mu_{41} = \frac{1}{\sum_C \left(\frac{d_{ij}}{d_{im}}\right)^{2/(g-1)}} = \frac{1}{1 + \left(\frac{d_{41}}{d_{42}}\right)^2}
$$

$$
\mu_{51} = \frac{1}{\sum_C \left(\frac{d_{ij}}{d_{im}}\right)^{2/(g-1)}} = \frac{1}{1 + \left(\frac{d_{51}}{d_{52}}\right)^2}
$$

$$
\mu_{61} = \frac{1}{\sum_C \left(\frac{d_{ij}}{d_{im}}\right)^{2/(g-1)}} = \frac{1}{1 + \left(\frac{d_{61}}{d_{62}}\right)^2}
$$

X_1 (2,12)	X_2 (4,9)	X_3 (7,13)	X_4 (11,5)	X_5 (12,7)	X_6 (14,4)

New centers :

c_1 = (6.6273,9.1484)
c_2 = (9.7344, 8.4887)

$$d_{ij} = \|\underline{C_j} - \underline{x_i}\|$$

$$d_{11}^2 = \|\underline{C_1} - \underline{x_1}\|^2 = \left\|\begin{bmatrix} 6.6273 - 2 \\ 9.1484 - 12 \end{bmatrix}\right\|^2 = 29.5435$$

$$d_{12}^2 = \|\underline{C_2} - \underline{x_1}\|^2 = \left\|\begin{bmatrix} 9.7344 - 2 \\ 8.4887 - 12 \end{bmatrix}\right\|^2 = 72.1502$$

$$\mu_{11} = \frac{1}{1 + \left(\frac{d_{11}}{d_{12}}\right)^2} = 0.7095$$

$$d_{21}^2 = \|\underline{C_1} - \underline{x_2}\|^2 = \left\|\begin{bmatrix} 6.6273 - 4 \\ 9.1484 - 9 \end{bmatrix}\right\|^2 = 6.9247$$

$$d_{22}^2 = \|\underline{C_2} - \underline{x_2}\|^2 = \left\|\begin{bmatrix} 9.7344 - 4 \\ 8.4887 - 9 \end{bmatrix}\right\|^2 = 33.1448$$

$$\mu_{21} = 0.8272$$

X_1 (2,12)	X_2 (4,9)	X_3 (7,13)	X_4 (11,5)	X_5 (12,7)	X_6 (14,4)

New centers :

c_1 = (6.6273,9.1484)
c_2 = (9.7344, 8.4887)

$$d_{ij} = \|\underline{C_j} - \underline{x_i}\|$$

$$d_{31}^2 = \|\underline{C_1} - \underline{x_3}\|^2 = \left\|\begin{bmatrix} 6.6273 - 7 \\ 9.1484 - 13 \end{bmatrix}\right\|^2 = 14.9737$$

$$d_{32}^2 = \|\underline{C_2} - \underline{x_3}\|^2 = \left\|\begin{bmatrix} 9.7344 - 7 \\ 8.4887 - 13 \end{bmatrix}\right\|^2 = 27.8288$$

$$\mu_{31} = 0.6502$$

$$d_{41}^2 = \|\underline{C_1} - \underline{x_4}\|^2 = \left\|\begin{bmatrix} 6.6273 - 11 \\ 9.1484 - 5 \end{bmatrix}\right\|^2 = 36.3297$$

$$d_{42}^2 = \|\underline{C_2} - \underline{x_4}\|^2 = \left\|\begin{bmatrix} 9.7344 - 11 \\ 8.4887 - 5 \end{bmatrix}\right\|^2 = 13.7728$$

$$\mu_{41} = 0.2749$$

X_1 (2,12)	X_2 (4,9)	X_3 (7,13)	X_4 (11,5)	X_5 (12,7)	X_6 (14,4)

New centers :

c_1 = (6.6273,9.1484)
c_2 = (9.7344, 8.4887)

$$d_{ij} = \left\| \underline{C}_j - \underline{x}_i \right\|$$

$$d_{51}^2 = \left\| \underline{C}_1 - \underline{x}_5 \right\|^2 = \left\| \begin{bmatrix} 6.6273 - 12 \\ 9.1484 - 7 \end{bmatrix} \right\|^2 = 33.4815$$

$$d_{52}^2 = \left\| \underline{C}_2 - \underline{x}_5 \right\|^2 = \left\| \begin{bmatrix} 9.7344 - 12 \\ 8.4887 - 7 \end{bmatrix} \right\|^2 = 7.3492$$

$$\mu_{51} = 0.1800$$

$$d_{61}^2 = \left\| \underline{C}_1 - \underline{x}_6 \right\|^2 = \left\| \begin{bmatrix} 6.6273 - 14 \\ 9.1484 - 4 \end{bmatrix} \right\|^2 = 80.8627$$

$$d_{62}^2 = \left\| \underline{C}_2 - \underline{x}_6 \right\|^2 = \left\| \begin{bmatrix} 9.7344 - 14 \\ 8.4887 - 4 \end{bmatrix} \right\|^2 = 38.3438$$

$$\mu_{61} = 0.3217$$

$$\begin{array}{cc} C1 & C2 \\ (6.6273, 9.1484) & (9.7344, 8.4887) \end{array}$$

$$U = \begin{bmatrix} 0.5397 & 0.4603 \\ 0.6852 & 0.3148 \\ 0.2093 & 0.7907 \\ 0.4194 & 0.5806 \\ 0.1970 & 0.8030 \\ 0.3881 & 0.6119 \end{bmatrix}$$

$$U = \begin{bmatrix} 0.7095 & 0.2905 \\ 0.8272 & 0.1728 \\ 0.6502 & 0.3498 \\ 0.2749 & 0.7251 \\ 0.1800 & 0.8200 \\ 0.3217 & 0.6783 \end{bmatrix}$$

Some Features:

1) Apriori knowledge of number of clusters

   - may be problem specific

   - may be determined by trial and error

   - some other simpler clustering algorithm may be used to have a rough idea about the number of clusters

2) Sensitivity to initialization

   - suffers from local minima problem

3) Higher ‘g’ value implies higher fuzziness; results in slower convergence