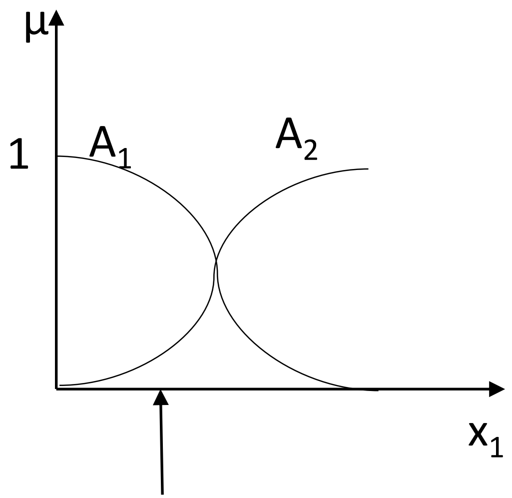
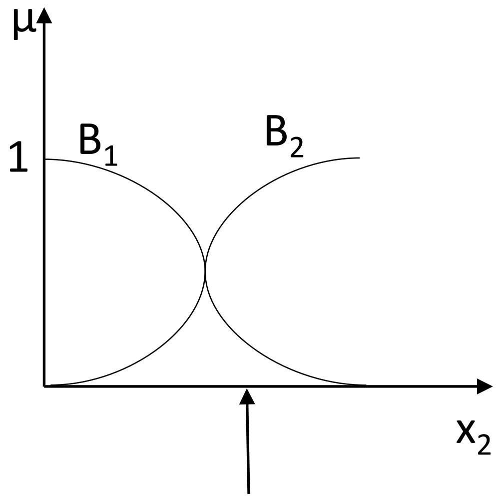
Neuro-Fuzzy Systems

- ✓ Neural Network can learn from input-output dataset, but appears as a black box to the human user i.e. lacks human like reasoning/interpretability
- ✓ Adaptivity and generalization of NN
- ✓ Fuzzy Systems follow human reasoning process, but they cannot learn
- ✓ However, both Fuzzy Systems and the Multilayer NN are universal approximators
- ✓ Combine learnability and adaptivity of NN with interpretability of FLS
- ✓ Suitable for on-line applications

Adaptive Neuro-Fuzzy Inference System (ANFIS):

- ✓ Sugeno Inference
- ✓ Algebraic Product T-norm
- ✓ Multilayer feedforward architecture
- ✓ Backpropagation learning
- ✓ Proposed in 1990’s





$$A_1(x_1) = \frac{1}{1 + \mathrm{e}^{b_1[x_1 - a_1]}}$$

$$A_2(x_1) = \frac{1}{1 + \mathrm{e}^{-b_1[x_1 - a_1]}}$$

$$B_1(x_2) = \frac{1}{1 + \mathrm{e}^{b_2[x_2 - a_2]}}$$

$$B_2(x_2) = \frac{1}{1 + \mathrm{e}^{-b_2[x_2 - a_2]}}$$

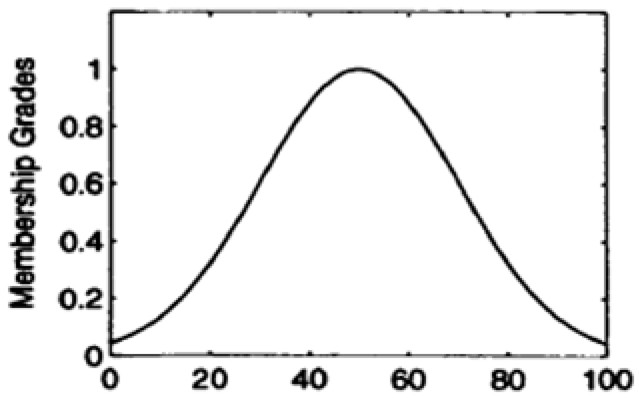
$x_1 \backslash x_2$	B_1	B_2
A_1	$f_1(\cdot,\cdot)$	$f_2(\cdot,\cdot)$
A_2	$f_3(\cdot,\cdot)$	$f_4(\cdot,\cdot)$

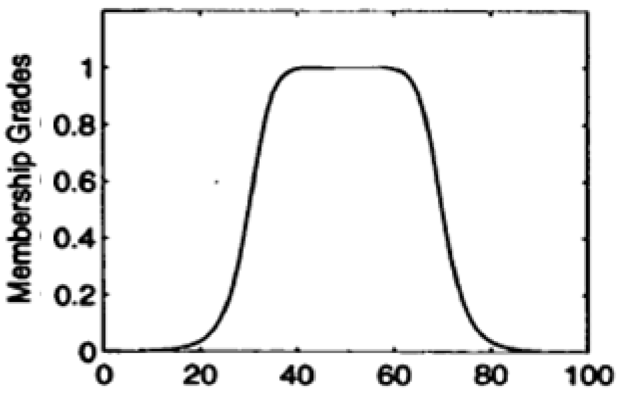
$$y_1 = f_1 = c_{11}x_1 + c_{12}x_2 + c_{13}$$

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

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

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





$$\text{gaussian}(x; c, \sigma) = e^{-\frac{1}{2}\left(\frac{x-c}{\sigma}\right)^2}.$$

$$\text{bell}(x; a, b, c) = \frac{1}{1 + \left|\frac{x-c}{a}\right|^{2b}}$$

weights of rules:

$$w_1 = A_1(x_1)\, B_1(x_2)$$

$$w_2 = A_1(x_1)\, B_2(x_2)$$

$$w_3 = A_2(x_1)\, B_1(x_2)$$

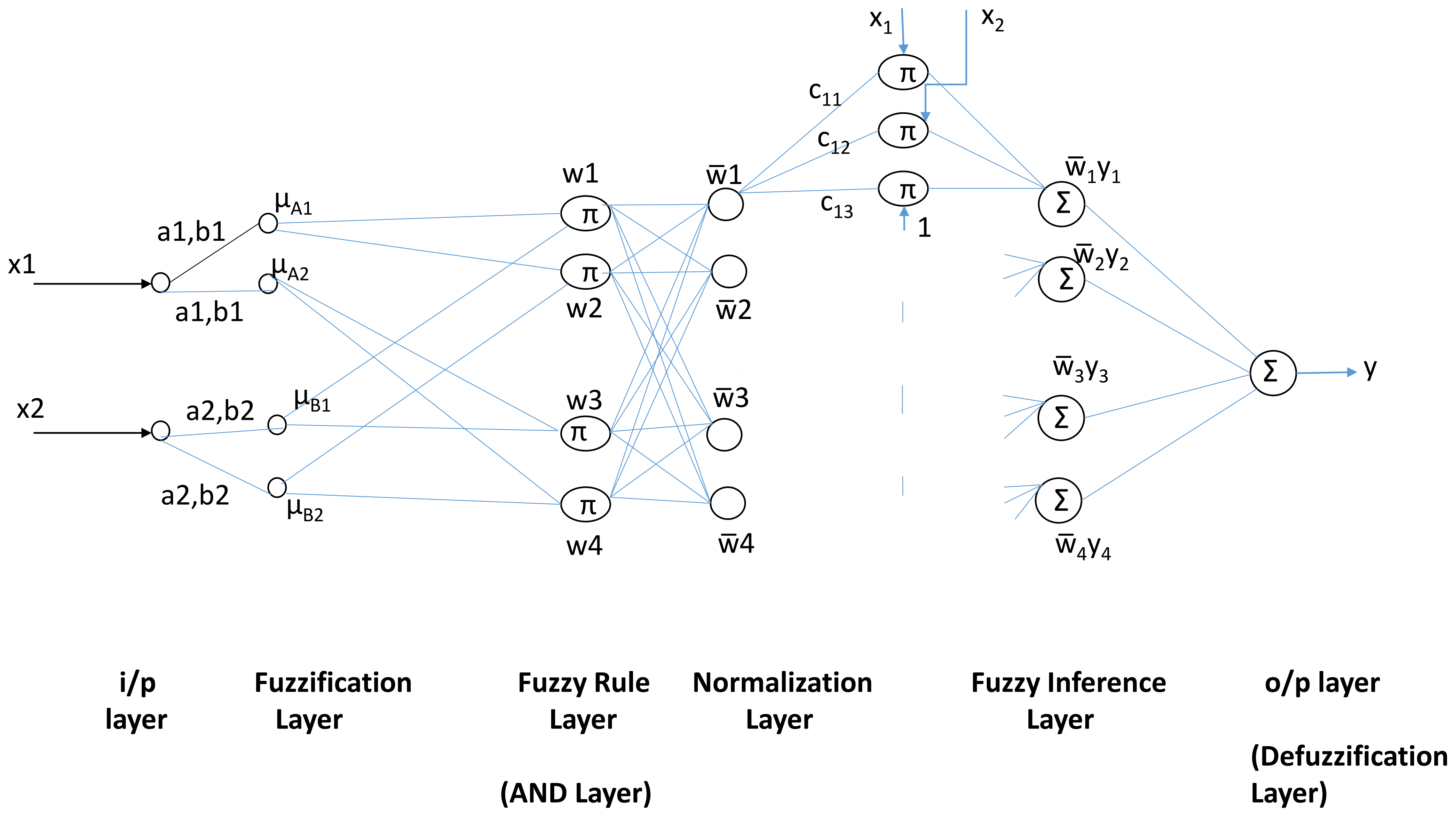
$$w_4 = A_2(x_1)\, B_2(x_2)$$

S.No.	x1	x2	T
1.	5.0	2.5	10.1
2.	3.5	-4.5	-8.2
⋮	⋮	⋮	⋮

$$y = \frac{w_1 y_1 + w_2 y_2 + w_3 y_3 + w_4 y_4}{w_1 + w_2 + w_3 + w_4} = \bar{w}_1 y_1 + \bar{w}_2 y_2 + \bar{w}_3 y_3 + \bar{w}_4 y_4$$

$$= w_1 y_1 + w_2 y_2 + w_3 y_3 + w_4 y_4$$

$$(\because w_1 + w_2 + w_3 + w_4 = A_1 B_1 + A_1 B_2 + A_2 B_1 + A_2 B_2 = 1)$$



x1
x2
a1,b1
a1,b1
a2,b2
a2,b2
μA1
μA2
μB1
μB2
w1
w2
w3
w4
π
w̄1
w̄2
w̄3
w̄4
x1
x2
c11
c12
c13
1
Σ
w̄1y1
w̄2y2
w̄3y3
w̄4y4
y
i/p layer
Fuzzification Layer
Fuzzy Rule Layer
(AND Layer)
Normalization Layer
Fuzzy Inference Layer
o/p layer
(Defuzzification Layer)

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

$$\Delta C_{11} = -\eta \frac{\partial E}{\partial C_{11}} = -\eta \frac{\partial E}{\partial y_1} \cdot \frac{\partial y_1}{\partial C_{11}} = -\eta \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial y_1} \cdot \frac{\partial y_1}{\partial C_{11}} = \eta(T - y) \cdot w_1 x_1 = \eta(T - y) \cdot A_1 B_1 x_1$$

$$\Delta C_{12} = -\eta \frac{\partial E}{\partial C_{12}} = -\eta \frac{\partial E}{\partial y_1} \cdot \frac{\partial y_1}{\partial C_{12}} = -\eta \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial y_1} \cdot \frac{\partial y_1}{\partial C_{12}} = \eta(T - y) \cdot w_1 x_2 = \eta(T - y) \cdot A_1 B_1 x_2$$

$$\Delta C_{13} = -\eta \frac{\partial E}{\partial C_{13}} = -\eta \frac{\partial E}{\partial y_1} \cdot \frac{\partial y_1}{\partial C_{13}} = -\eta \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial y_1} \cdot \frac{\partial y_1}{\partial C_{13}} = \eta(T - y) \cdot A_1 B_1$$

$$\Delta C_{21} = -\eta \frac{\partial E}{\partial C_{21}} = -\eta \frac{\partial E}{\partial y_1} \cdot \frac{\partial y_1}{\partial C_{21}} = -\eta \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial y_1} \cdot \frac{\partial y_1}{\partial C_{21}} = \eta(T - y) \cdot w_2 \cdot x_1 = \eta(T - y) \cdot A_1 B_2 \cdot x_1$$

…

Next tunable parameters are : $a_1,\ b_1,\ a_2,\ b_2$

$$A_1 + A_2 = 1$$

$$B_1 + B_2 = 1$$

$$\Delta a_1 = -\eta \frac{\partial E}{\partial a_1} = -\eta \cdot \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial A_1} \cdot \frac{\partial A_1}{\partial a_1}$$

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

$$y = w_1 y_1 + w_2 y_2 + w_3 y_3 + w_4 y_4$$

$$\therefore \frac{\partial y}{\partial A_1} = \frac{\partial}{\partial A_1}(A_1 B_1\ y_1 + A_1 B_2\ y_2 + A_2 B_1\ y_3 + A_2 B_2\ y_4)$$

$$= B_1 y_1 + B_2 y_2$$

$$\therefore \Delta a_1 = \eta(T-y)\cdot b_1\cdot A_1\cdot A_2\cdot(B_1y_1+B_2y_2)$$

Similarly,

$$\Delta a_2 = -\eta\frac{\partial E}{\partial a_2} = -\eta\cdot\frac{\partial E}{\partial y}\cdot\frac{\partial y}{\partial B_1}\cdot\frac{\partial B_1}{\partial a_2}$$

Now,

$$y = w_1y_1 + w_2y_2 + w_3y_3 + w_4y_4$$

$$\therefore \frac{\partial y}{\partial B_1} = \frac{\partial}{\partial B_1}(A_1B_1\ y_1 + A_1B_2\ y_2 + A_2B_1\ y_3 + A_2B_2\ y_4)$$

$$= A_1y_1 + A_2y_3$$

$$Now \qquad \frac{\partial B_1}{\partial a_2} = \frac{\partial}{\partial a_2}\left(\frac{1}{1+e^{b_2(x_2-a_2)}}\right) = b_2 \cdot B_1 \cdot (1 - B_1) = b_2 \cdot B_1 \cdot B_2$$

$$\therefore \Delta a_2 = \eta(T-y) \cdot b_2 \cdot B_1 \cdot B_2 \cdot (A_1 y_1 + A_2 y_3)$$

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

Similarly b_1 and b_2 are updated as

$$\Delta b_1 = -\eta(T-y) \cdot (x_1 - a_1) \cdot A_1 \cdot A_2 \cdot (B_1 y_1 + B_2 y_2)$$

$$\Delta b_2 = -\eta(T-y) \cdot (x_2 - a_2) \cdot B_1 \cdot B_2 \cdot (A_1 y_1 + A_2 y_3)$$

Pseudoinverse Method:

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

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

$$= \overline{\omega_1}(c_{11}x_1 + c_{12}x_2 + c_{13}) + \overline{\omega_2}(c_{21}x_1 + c_{22}x_2 + c_{23}) + \overline{\omega_3}(c_{31}x_1 + c_{32}x_2 + c_{33}) + \overline{\omega_4}(c_{41}x_1 + c_{42}x_2 + c_{43})$$

$$= \overline{\omega_1} c_{11} x_1 + \overline{\omega_1} c_{12} x_2 + \overline{\omega_1} c_{13} + \overline{\omega_2} c_{21} x_1 + \overline{\omega_2} c_{22} x_2 + \overline{\omega_2} c_{23}$$
$$+ \overline{\omega_3} c_{31} x_1 + \overline{\omega_3} c_{32} x_2 + \overline{\omega_3} c_{33} + \overline{\omega_4} c_{41} x_1 + \overline{\omega_4} c_{42} x_2 + \overline{\omega_4} c_{43})$$

$$= \begin{bmatrix} \overline{\omega_1} x_1 & \overline{\omega_1} x_2 & \overline{\omega_1} & \overline{\omega_2} x_1 & \overline{\omega_2} x_2 & \overline{\omega_2} & \overline{\omega_3} x_1 & \overline{\omega_3} x_2 & \overline{\omega_3} & \overline{\omega_4} x_1 & \overline{\omega_4} x_2 & \overline{\omega_4} \end{bmatrix} \begin{bmatrix} c_{11} \\ c_{12} \\ c_{13} \\ c_{21} \\ c_{22} \\ c_{23} \\ c_{31} \\ c_{32} \\ c_{32} \\ c_{41} \\ c_{42} \\ c_{43} \end{bmatrix}$$

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

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

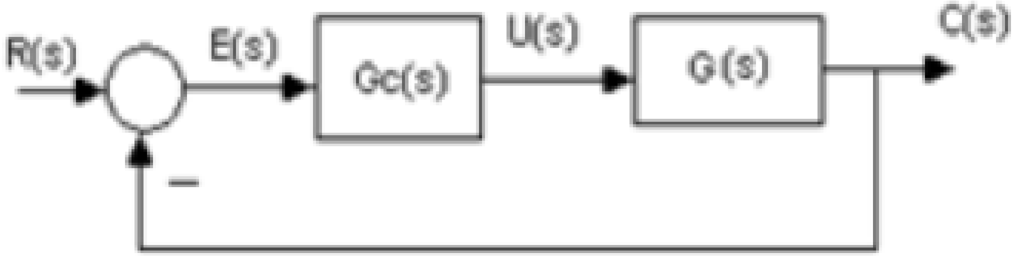
$$\underline{c} = pinv\left(\begin{bmatrix} p_1 \\ \vdots \\ p_N \end{bmatrix}\right) \begin{bmatrix} T_1 \\ \vdots \\ T_N \end{bmatrix}$$

: least square error solution

(over-determined system)

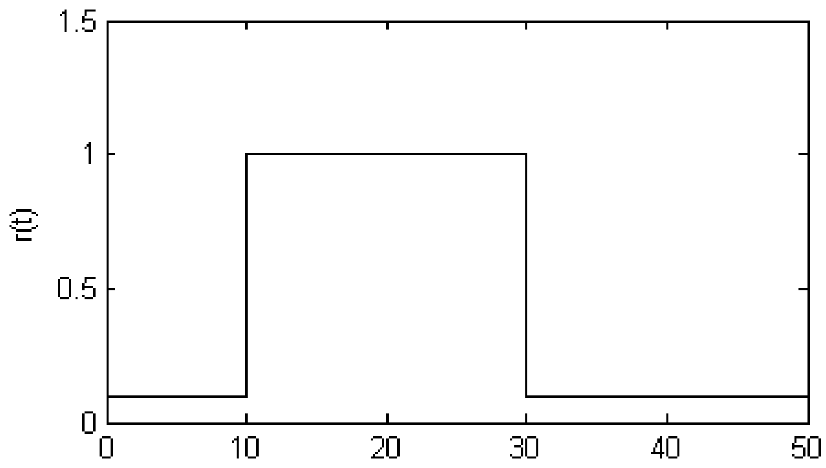
Fuzzy PID Control

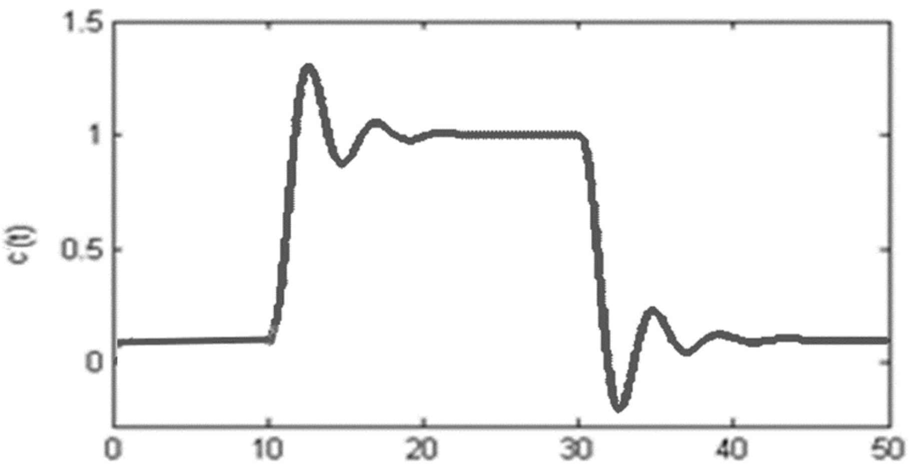
A Primer on PID Control:



- ✓ 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 disturbances
- Robustness to plant uncertainties
- 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

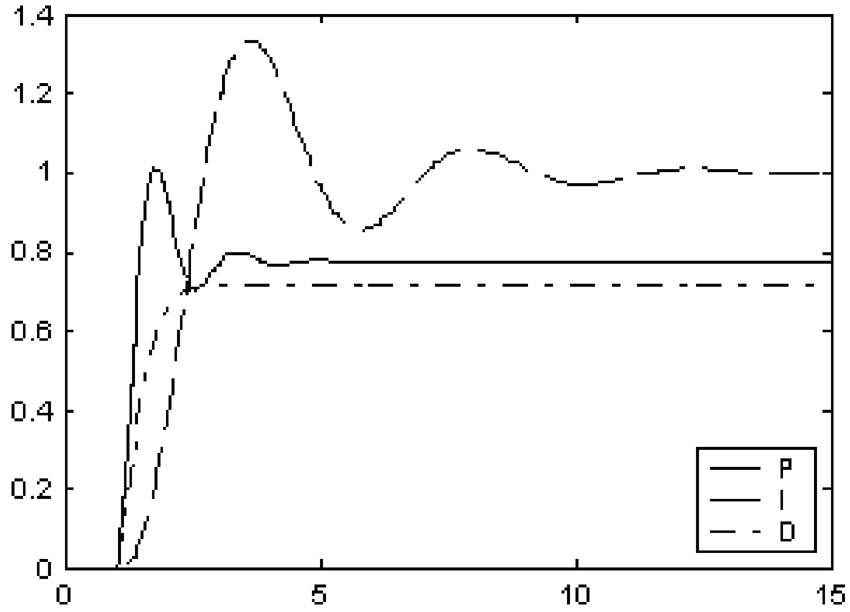
✓ I-term: $$u(t) = K_I \int_0^t e(t)dt$$

- 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

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

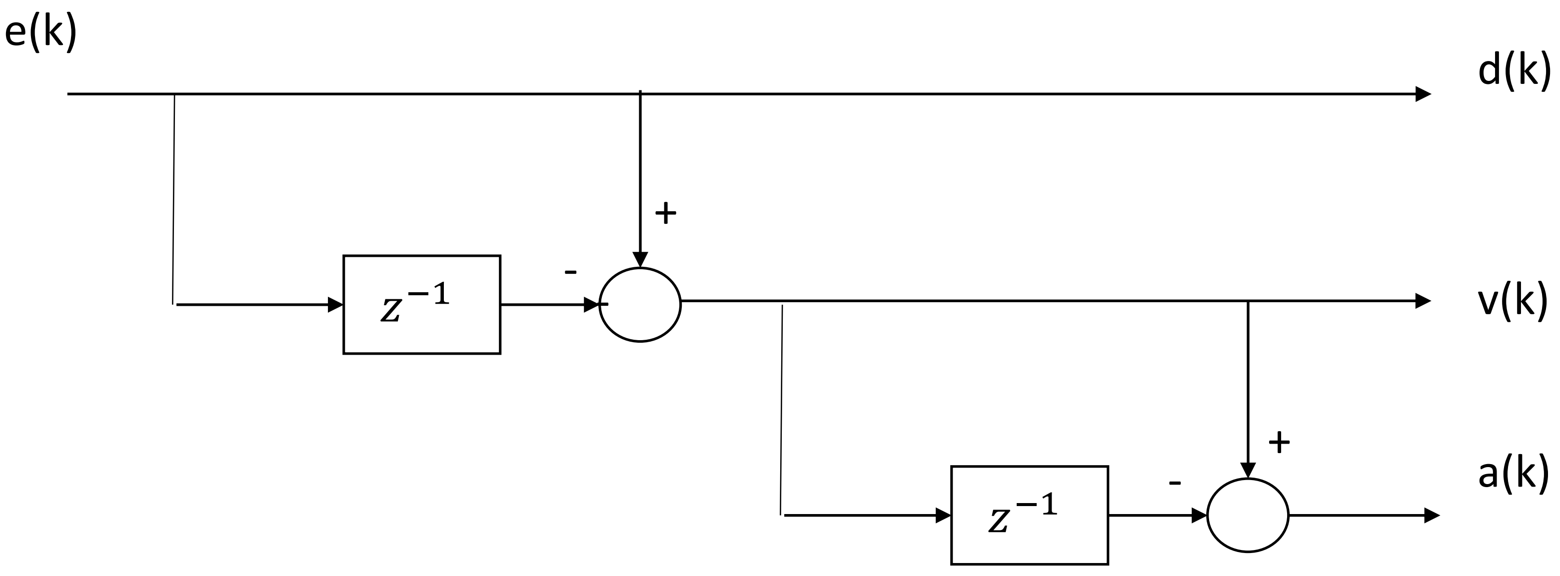
$$\frac{u(k) - u(k-1)}{T_s} = K_P\, v(k) + K_I\, d(k) + K_D\, a(k)$$
$$\Delta u(k) = K_I\, d(k) + K_P\, v(k) + K_D\, a(k)$$

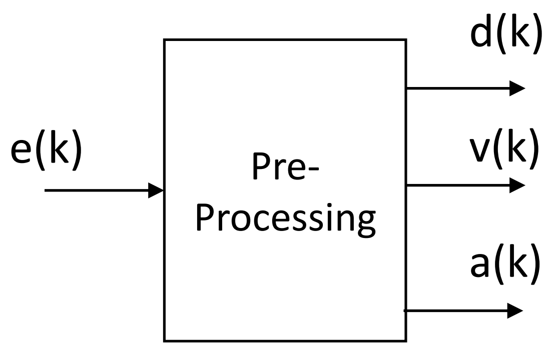
where $d(k) = error$

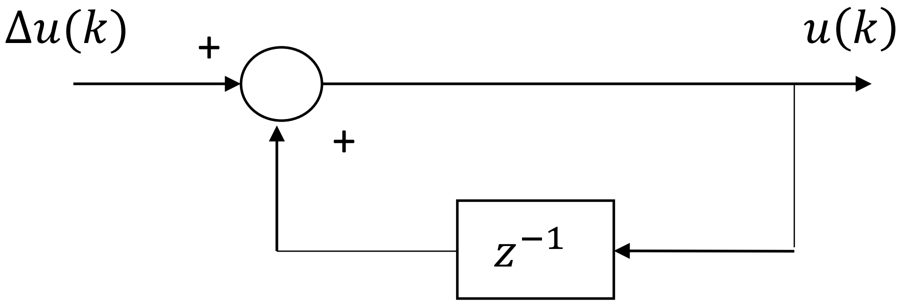
$v(k) = change\ in\ error$

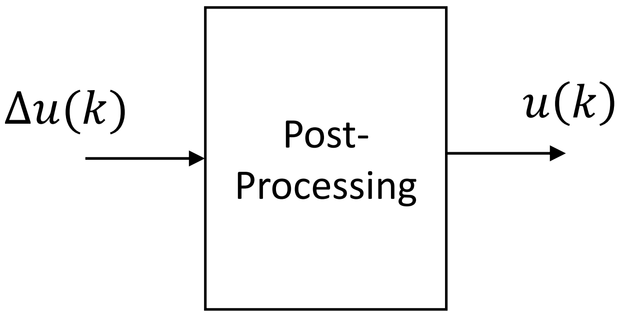
$a(k) = change\ in\ change\ in\ error$

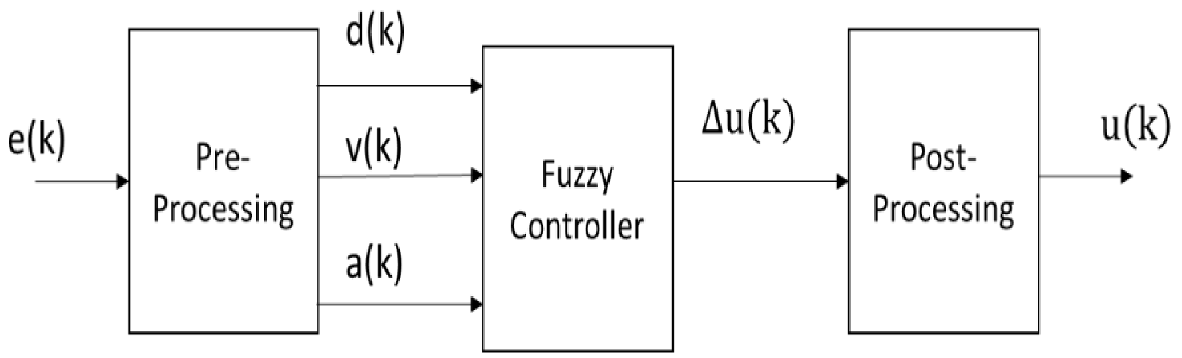
$$u(k) = u(k-1) + \Delta u(k)$$

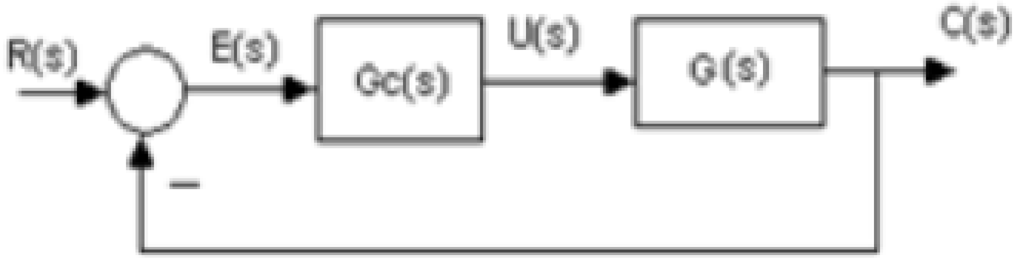












Advantages of Fuzzy PID

- ✓ 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

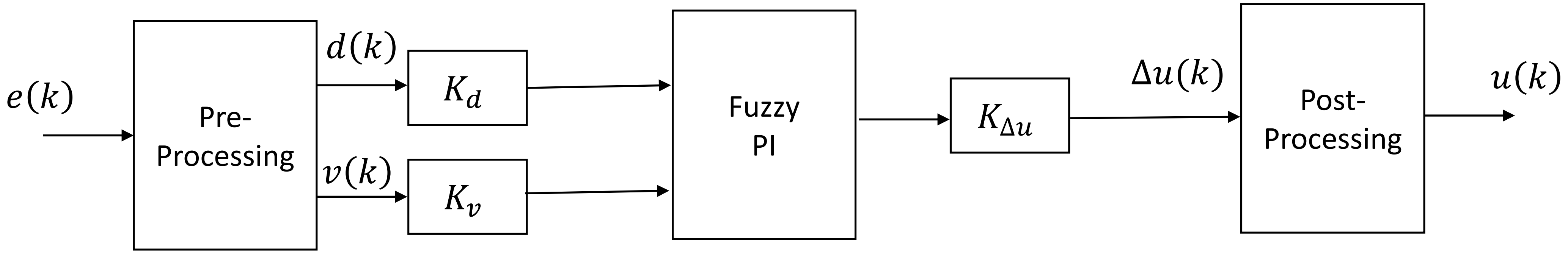
$$u(t) = K_P\, e(t) + K_I \int e(t)dt$$

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

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

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

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



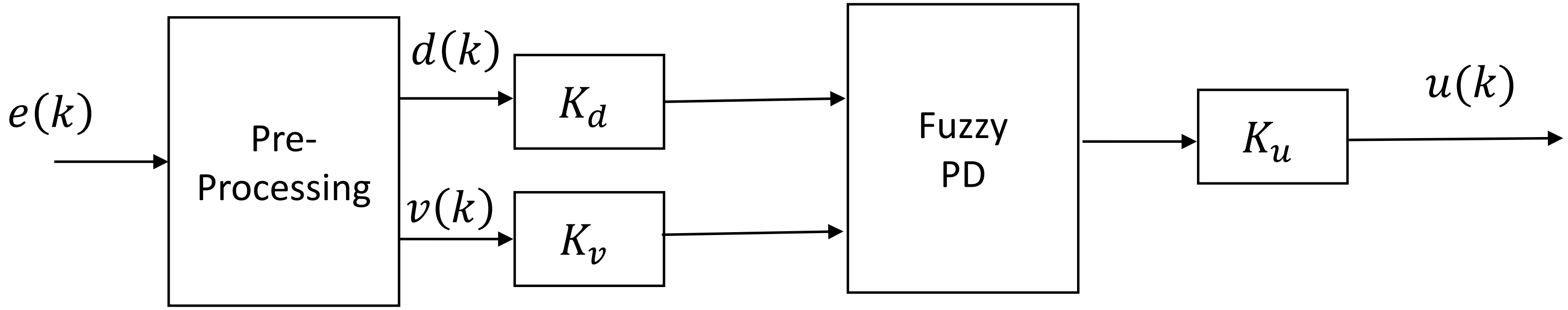
Fuzzy PD Controller

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

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

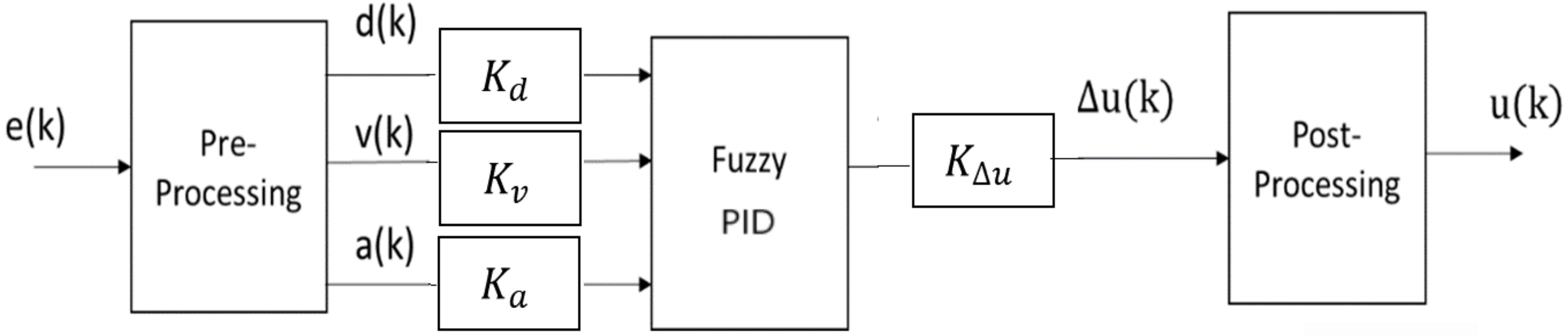
$$u(k) = K_P\, e(k) + K_D\, \Delta e(k)$$

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



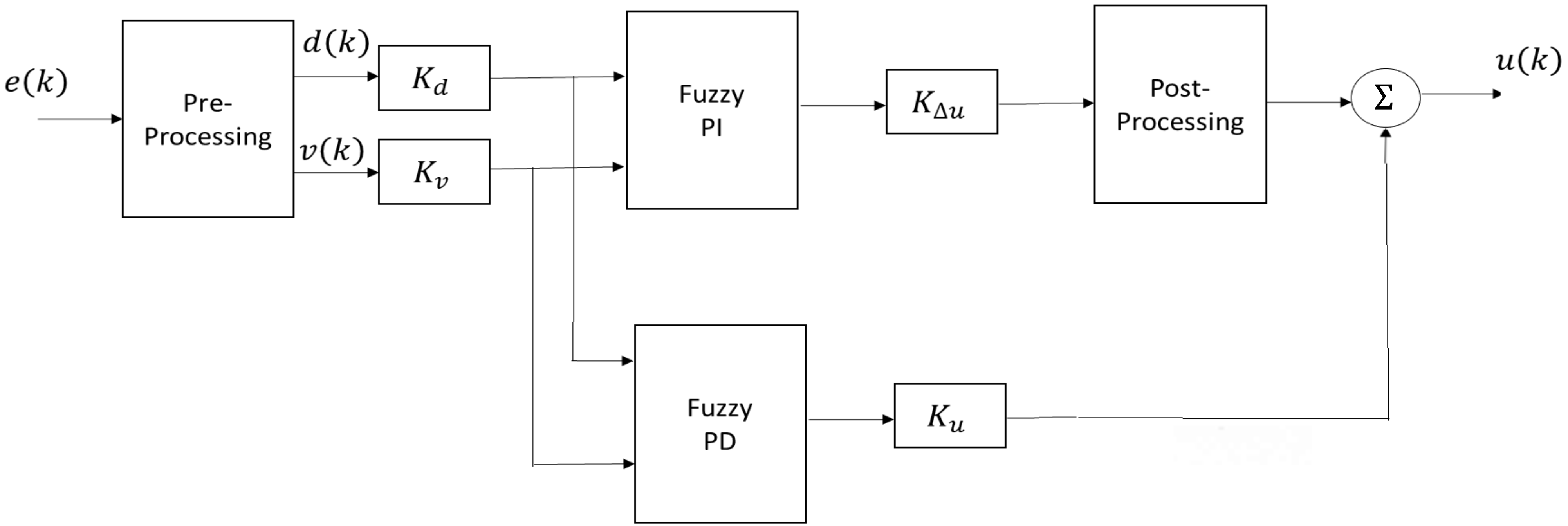
Three Term Fuzzy PID Controller

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

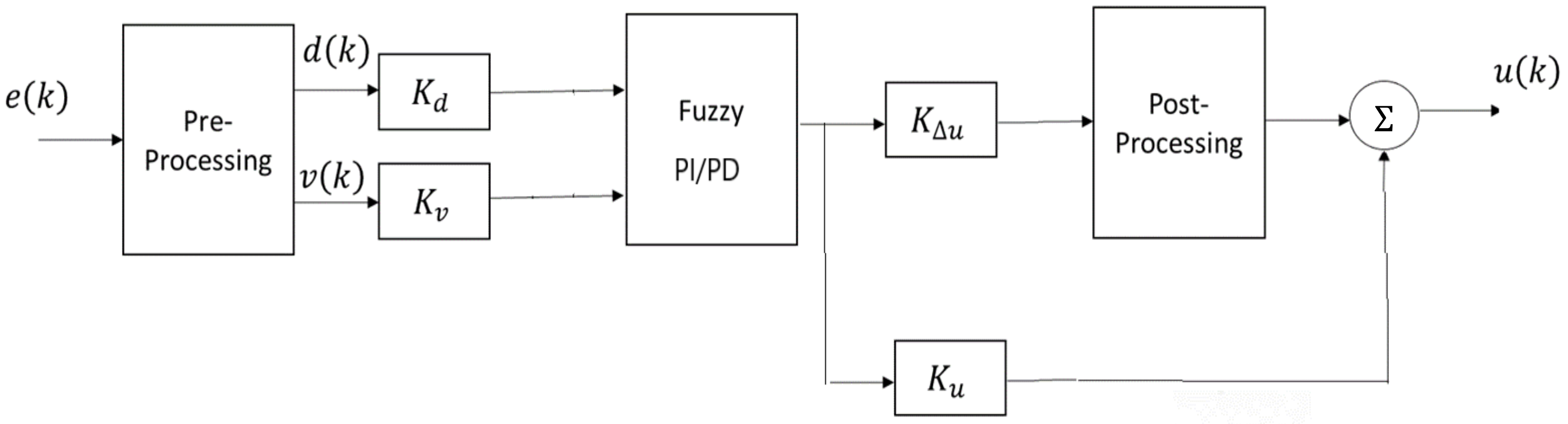


✓ Higher number of rules ($3^3 = 27$, $5^3 = 125$ etc.)

Two Term Fuzzy PID Controller



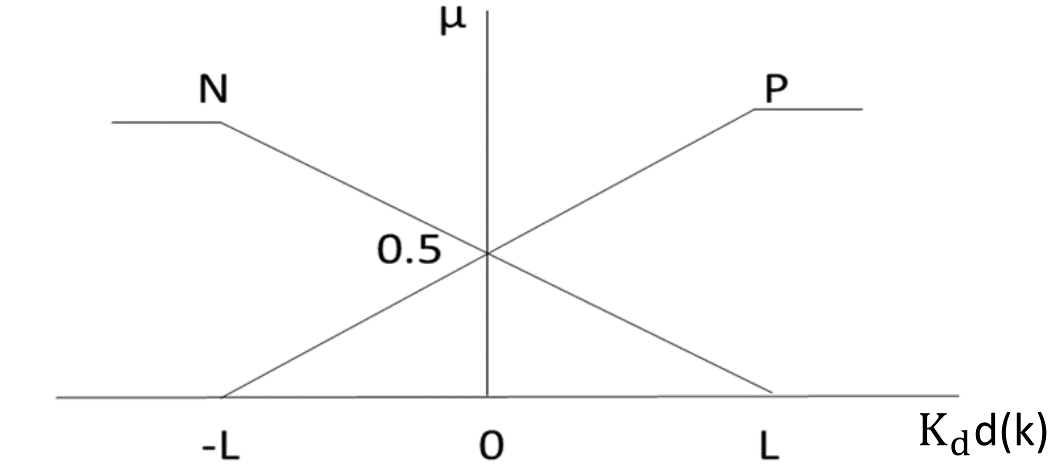
- ✓ Less number of rules ($2 \times 3^2 = 18$, $2 \times 5^2 = 50$ etc.)
- ✓ Scaling factors need to be properly tuned

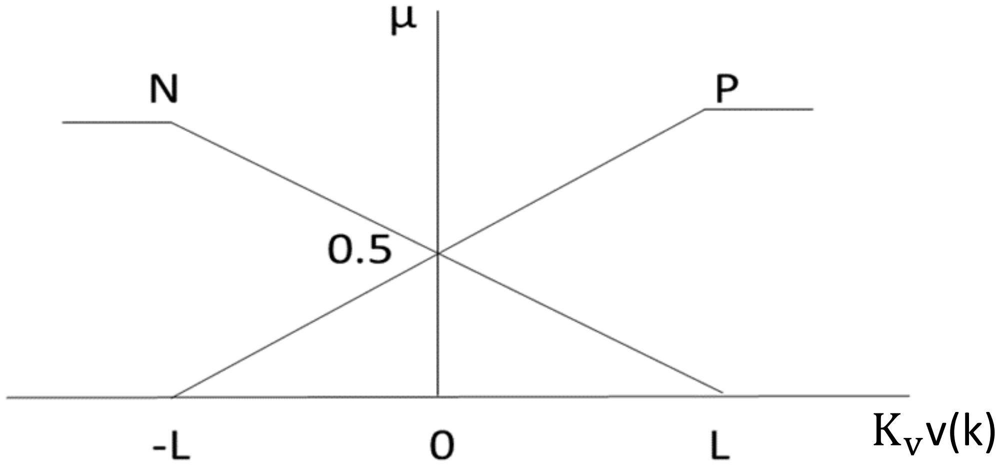


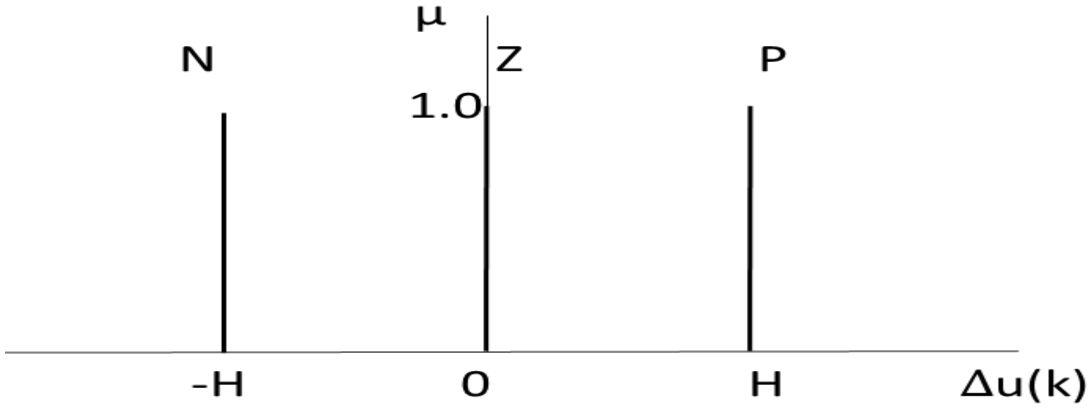
- ✓ Sometimes Fuzzy PI and Fuzzy PD may have identical rule base!
- ✓ No. of rules: 9, 25, …
- ✓ Many other structures are possible and proposed in the literature

  (E.g. FuzzyPD+I ; FuzzyPI+D etc.)

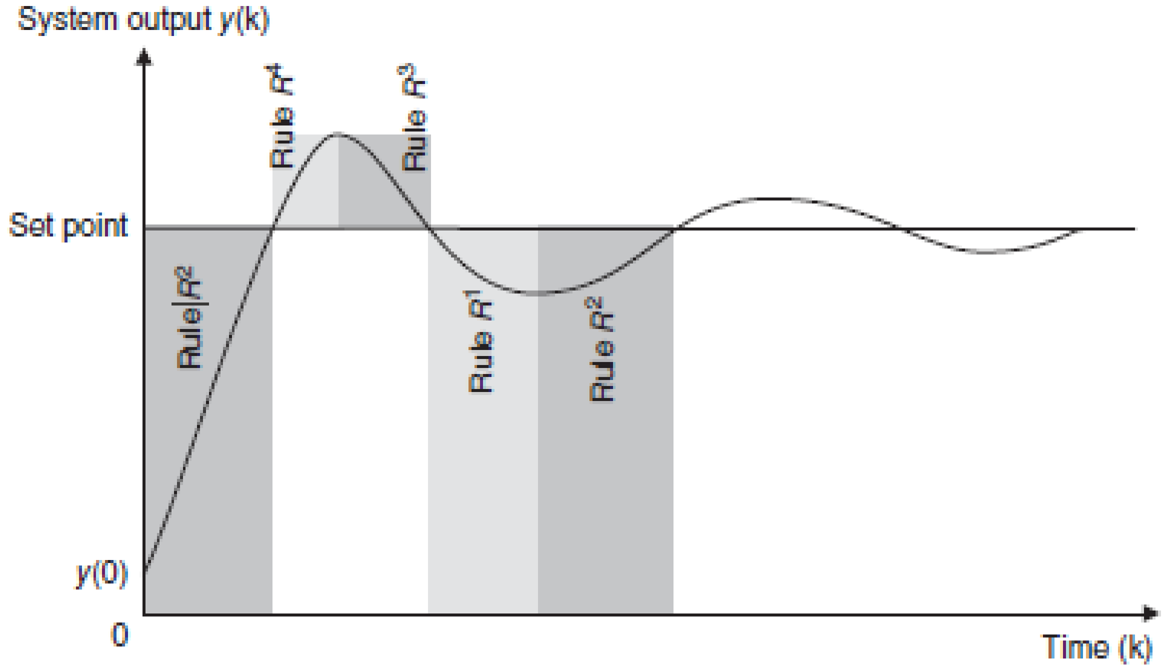
1) A Fuzzy PI Controller as a Conventional Linear PI Controller







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



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

✓ A two-term controller with four rules is generic enough!

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

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

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

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

- ✓ 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 = $\mu_N(d) * \mu_N(v)$

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

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

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

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

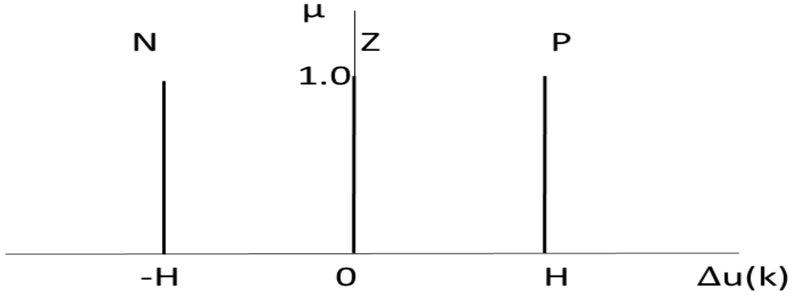
$$= K_{\Delta u}\left\{\frac{K_d\ d+L}{2L} * \frac{K_v\ v+L}{2L} * H\ - \frac{-K_d\ d+L}{2L} * \frac{-K_v\ v+L}{2L} * H\right\}$$

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

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

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

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



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

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

2) A Fuzzy PI Controller as a Nonlinear PI Controller

- ✓ Minimum T-norm

- ✓ Linear and symmetric membership functions

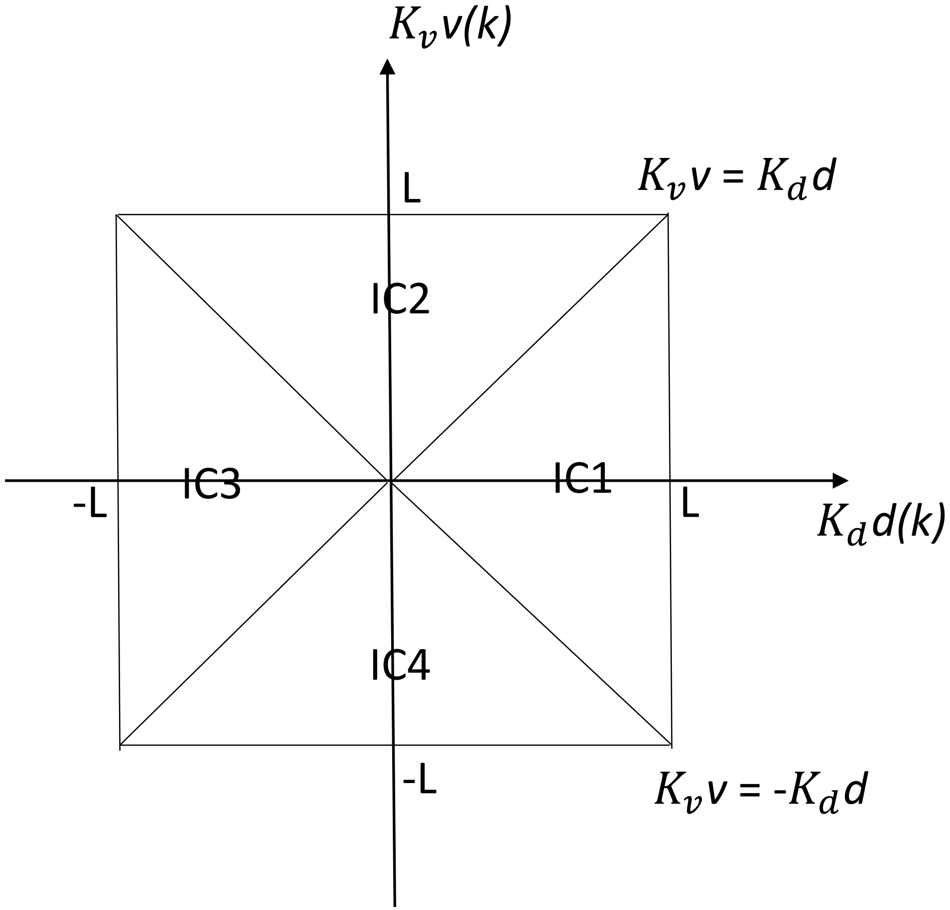
  for both d(k) and v(k)

- ✓ Zero order Sugeno Model

  (Mamdani with singleton o/p)

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

IC: Input Combination

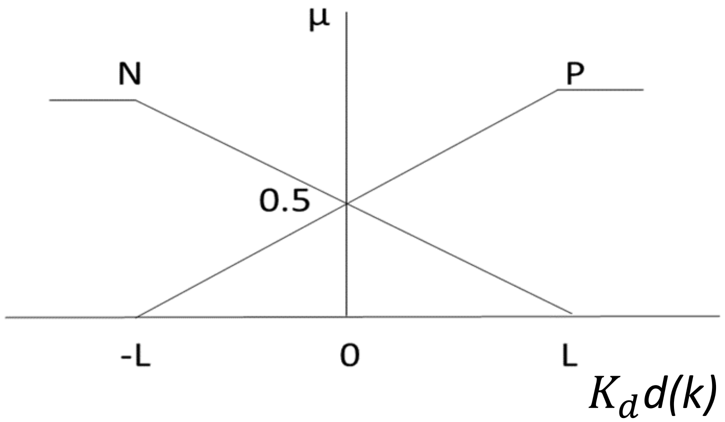


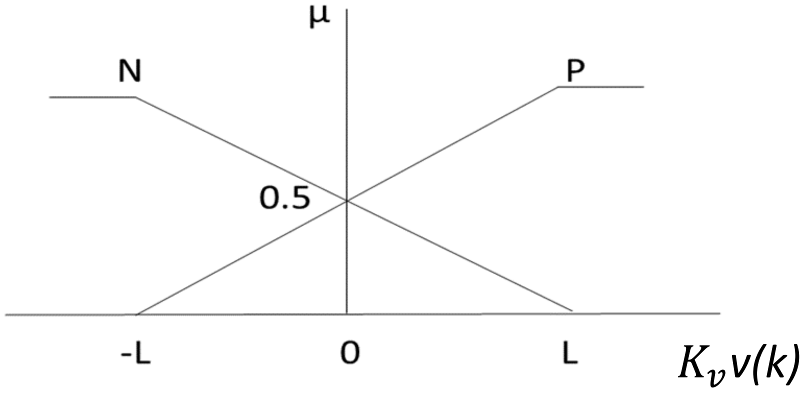
IC1: $0 \le K_d d(k) \le L$ and $-\mathrm{L} \le K_v v(k) \le L$

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

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

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





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

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

IC1: $0 \leq K_d d(k) \leq L$ and

$$-\mathrm{L} \leq K_v v(k) \leq L$$

$$|\, K_v \; v| \leq |\, K_d \; d|$$

Output for IC1 (and IC3):

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

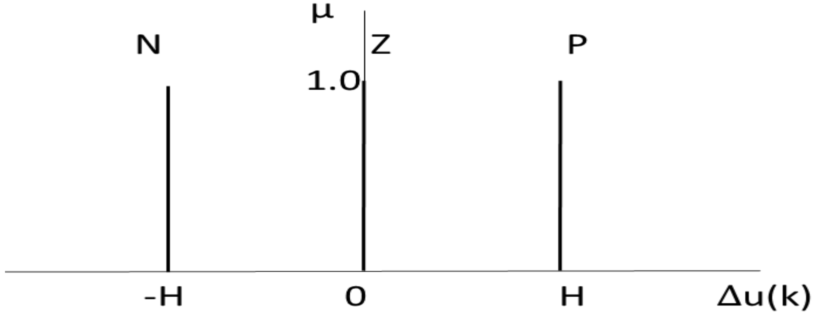
$$= K_{\Delta u} \frac{\mu_P(v) - \mu_N(d)}{1 + 2\, \mu_N(d)}\, H$$

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

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

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

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



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

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)