Sisteme Incorporate

Cursul 7

Control PID

Sisteme de Control Fuzzy

Recapitulare

- Control Proportional (P)
- Control Proportional Diferential (PD)
- Control Proportional Integral (PI)

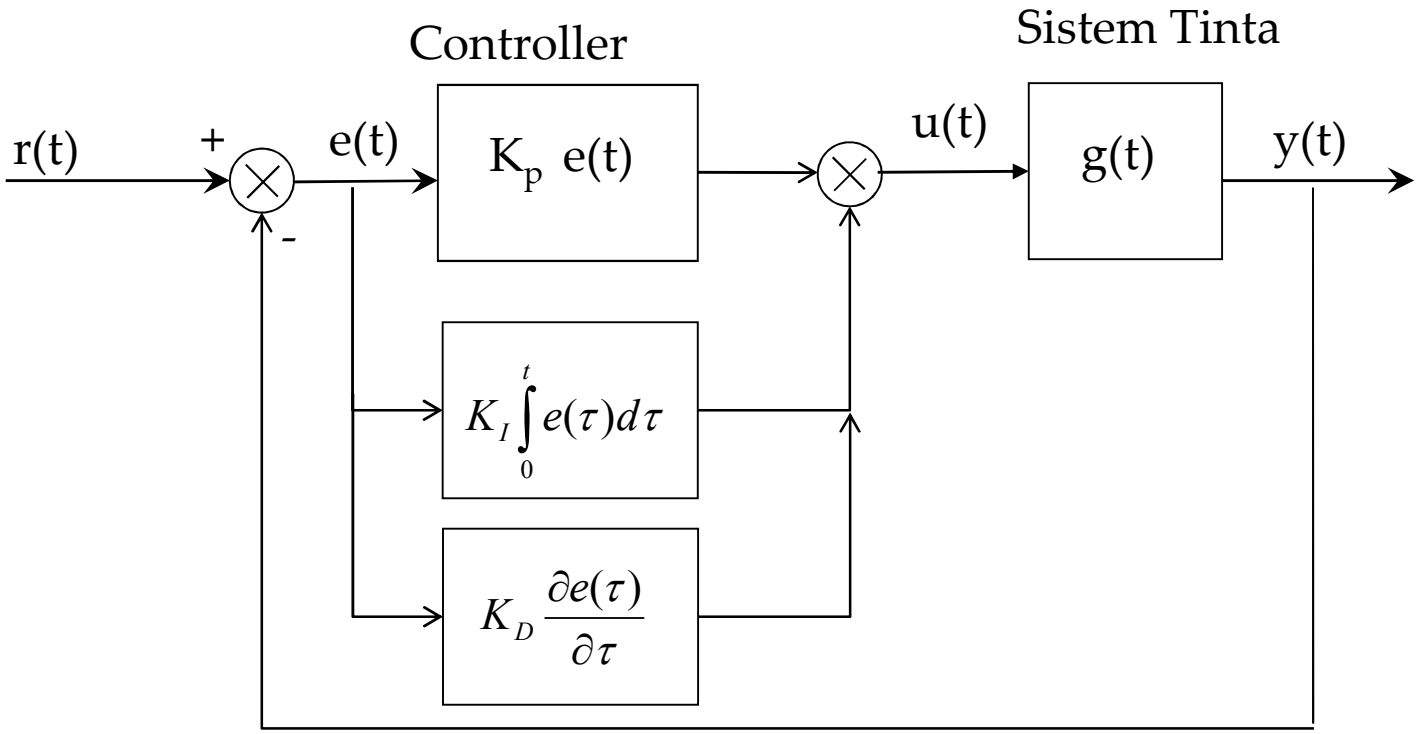
- Analiza din punct de vedere al stabilitatii si acuratetei raspunsului sistemului la o excitatie de tip treapta

Controlul PID

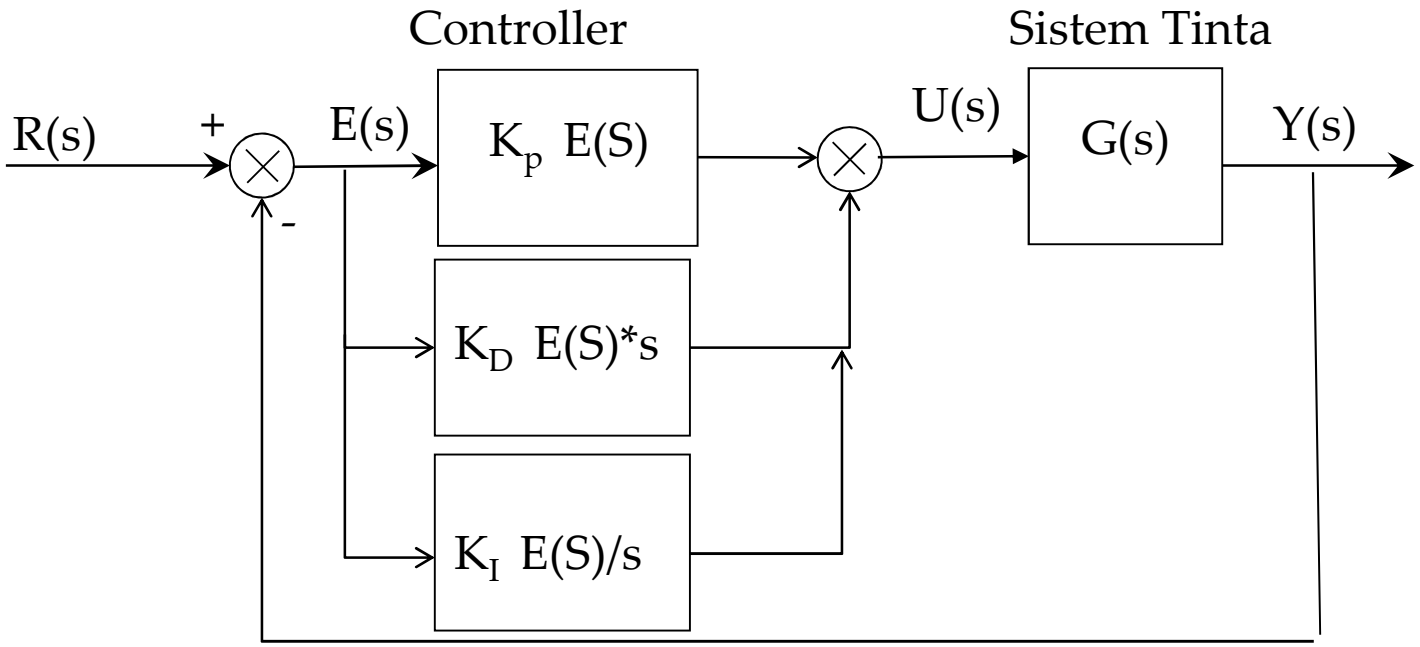
- Foloseste toate cele trei tipuri de control studiate anterior
- Este cel mai utilizat tip de control in industrie, datorita stabilitatii lui.
- Performante crescute din punctul de vedere al timpului de raspuns, stabilitatii si erorii de aproximare.

Controlul PID

$$u(t) = K_P e(t) + K_D \frac{\partial e(\tau)}{\partial \tau} + K_I \int_0^t e(\tau) d\tau$$



Functia de transfer PID



$$E(s) = R(s) - Y(s)$$

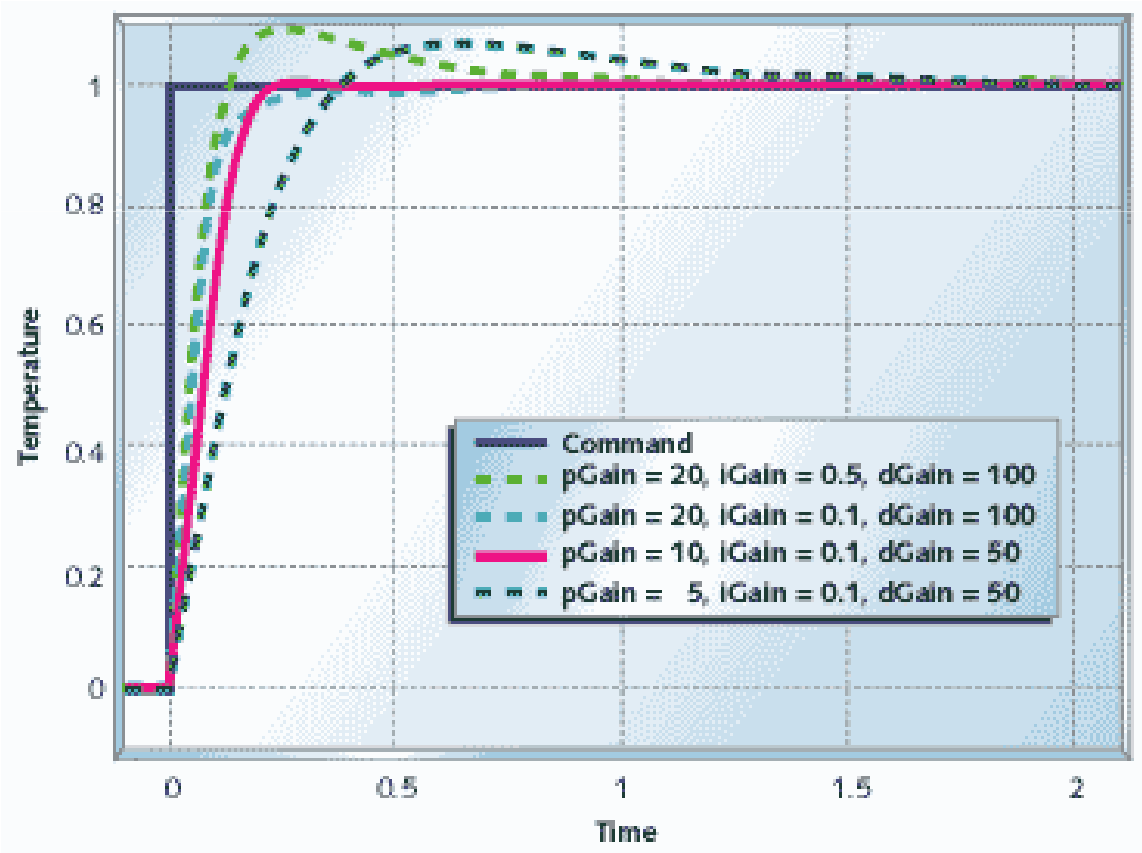
$$U(s) = K_P \cdot E(s) + K_D \cdot sE(s) + K_I \frac{E(s)}{s}$$

$$Y(s) = G(s)U(s)$$

$$H(s) = \frac{(K_P + sK_D + \frac{K_I}{s}) \cdot G(s)}{1 + \left(K_P + sK_D + \frac{K_I}{s} \right) \cdot G(s)}$$

Exemplu

- Pentru controllerul de temperatura:



www.embedded.com

Acuratete

Ecuatia caracteristica: $$1+\left(K_P+sK_D+\frac{K_I}{s}\right)\cdot G(s)=0$$

$$e_{ss} = \lim_{k->\infty} e(k)$$

$$e_{ss} = \lim_{s->0} \frac{1}{1+\left(K_P+sK_D+\frac{K_I}{s}\right)G(s)}$$

Estimarea coeficientilor (Loop Tuning)

- Daca parametrii controlului PID sunt alesi incorect, sistemul poate sa devina instabil si sa devieze de la raspunsul optim cu sau fara oscilatie.
- Mai multe metode de estimare:
  - Estimare Manuala
  - Ziegler-Nichols
  - Software Tuning
  - Cohen Coon

Manual Tuning

- Obtinerea matematica a K_P, K_I si K_D nu poate fi posibila
  - Sistemul este prea complex pentru a fi modelat matematic
  - Costurile de modelare sunt prea mari
- Metoda ad hoc pentru obtinerea unor valori rezonabile ale K_P, K_I, si K_D
  - Initial K_P foarte mic, $K_I = K_D = 0$
  - Mareste K_P pana cand apar oscilatii
    - Seteaza K_P la jumatate din valoarea obtinuta
  - Mareste K_I pana cand raspunsul sistemului este egal cu valoarea dorita
    - K_I mare poate induce oscilatii
  - Mareste K_D pana cand sistemul are un timp de raspuns suficient de mic pentru orice perturbatii

Metoda Ziegler-Nichols

- Produce rezultate acceptabile, dar nu optime
- Algoritm:
  - Initial K_P foarte mic, $K_I = K_D = 0$
  - Mareste K_P pana cand apar oscilatii (K_C – castigul critic)
    - Masoara perioada oscilatiilor sistemului (T_C)
  - Determina coeficientii folosind urmatoarele relatii:

	K_P	K_I	K_D
P	$0.5 \cdot K_c$	-	-
PI	$0.45 \cdot K_c$	$1.2K_p / T_c$	-
PID	$0.6 \cdot K_c$	$2K_p / T_c$	$K_p T_c / 8$

Implementare Software

- Bucla principala, cicleaza la infinit
  - Masoara variabila de iesire a sistemului
    - De cele mai multe ori convertor AD
  - Masoara valoarea de referinta curenta
  - Apeleaza PidUpdate pentru a calcula comanda efectorului
  - Seteaza valoarea curenta pentru efector
    - Convertor DA

```
void main()
{
    double sensor_value, actuator_value, error_current;
    PID_DATA pid_data;
    PidInitialize(&pid_data);
    while (1) {
        sensor_value = SensorGetValue();
        reference_value = ReferenceGetValue();
        actuator_value =
         PidUpdate(&pid_data,sensor_value,reference_value);
        ActuatorSetValue(actuator_value);
    }
```

Implementare Software

- Pgain, Dgain, Igain sunt constante stabilite cu un algoritm de tuning
- sensor_value_previous
  - Pentru controlul Diferential
- error_sum
  - Pentru controlul Integral

```
typedef struct PID_DATA {
    double Pgain, Dgain, Igain;
    double sensor_value_previous; // find the derivative
    double error_sum; // cumulative error
}
```

Calcularea coeficientilor

- $u_t=P*e_t+I*(e_0+e_1+...+e_t)+D*(e_t-e_{t-1})$

```
double PidUpdate(PID_DATA *pid_data, double sensor_value,
                 double reference_value)
{
    double Pterm, Iterm, Dterm;
    double error, difference;

    error = reference_value - sensor_value;
    Pterm = pid_data->Pgain * error; /* proportional term*/
    pid_data->error_sum += error; /* current + cumulative*/
    // the integral term
    Iterm = pid_data->Igain * pid_data->error_sum;
    difference = pid_data->sensor_value_previous -
                 sensor_value;
    // update for next iteration
    pid_data->sensor_value_previous = sensor_value;
    // the derivative term
    Dterm = pid_data->Dgain * difference;
    return (Pterm + Iterm + Dterm);
}
```

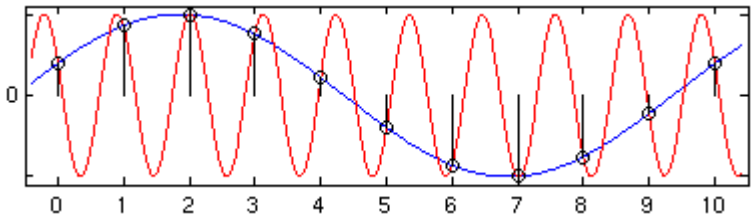
Deficiente in controlul numeric

- Erori de reprezentare
  - Nu toate valorile numerice pot fi reprezentate intern datorita limitarilor de memorie
- Overflow
  - Depasirea superioara sau inferioara a domeniului de reprezentare
- Aliasing
- Viteza de procesare

Aliasing

- Doua sau mai multe semnale periodice devin inditinctibile atunci cand sunt esantionate la o anumita frecventa.

- Exemplu:
- Esantionate la 2.5 Hz, urmatoarele semnale nu pot fi diferentiate
  - $y(t)=1.0*\sin(6\pi t)$, f = 3 Hz
  - $y(t)=1.0*\sin(\pi t)$, f = 0.5 Hz
- Conform teoremei Nyquist, frecventa minima de esantionare pentru primul semnal: 3/2 = 1.5 Hz



Viteza de procesare

- Produce intarzieri inevitabile
  - Actiunea calculata se petrece mai tarziu
- Intarzierile trebuiesc prevazute si estimate in implementarea oricarui sistem de control.
- Intarzierile hardware sunt de obicei usor de calculat
  - Structura sincrona a procesoarelor ajuta
- Intarzierile software sunt mai greu de prevazut
  - Codul trebuie organizat a.i. intarzierile sa fie minime
  - Software cu intarzieri predictibile
    - Rutine declansate la intervale regulate de timp
    - Limbaj de programare sincron (Esterel, ChucK)

Beneficiile controlului numeric

- Cost redus
  - Un control analogic este foarte scump mai ales daca este proiectat sa fie imun la perturbatii
    - Uzura, temperatura, erori de fabricatie
  - Controlul numeric inlocuieste circuitele analogice complexe cu programe complexe
- Programabilitate
  - Controlul poate fi “upgradat”
    - Schimbari la tipul controlului, castig sunt usor de facut
  - Se adapteaza la schimbarile sistemului
    - Datorate imbatranirii, temperaturii etc.
  - “future-proof”
    - Usor de adaptat la un alt standard

Sisteme de Control Fuzzy

Ce este Logica Fuzzy?

- Logica clasica:

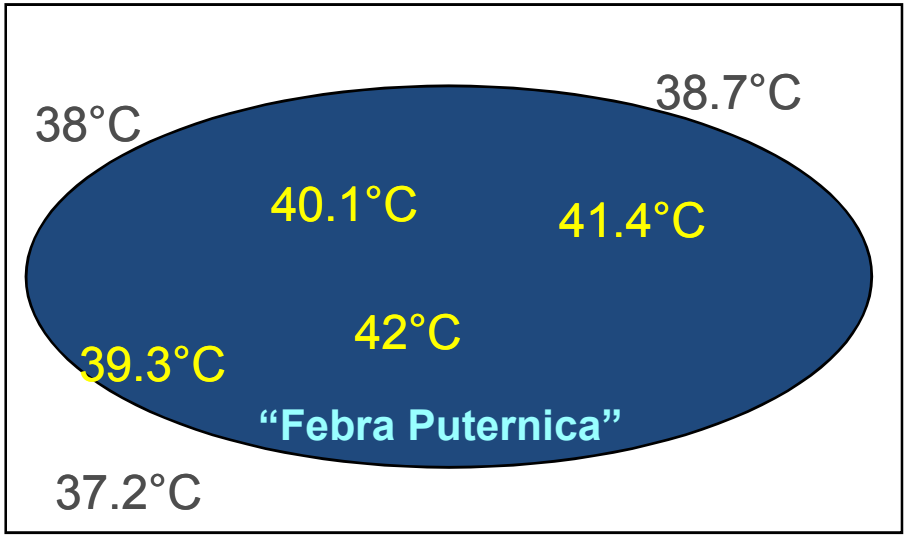
Un trandafir este Rosu…

sau nu este Rosu

Ce culoare are trandafirul acesta?

Ce este Logica Fuzzy?

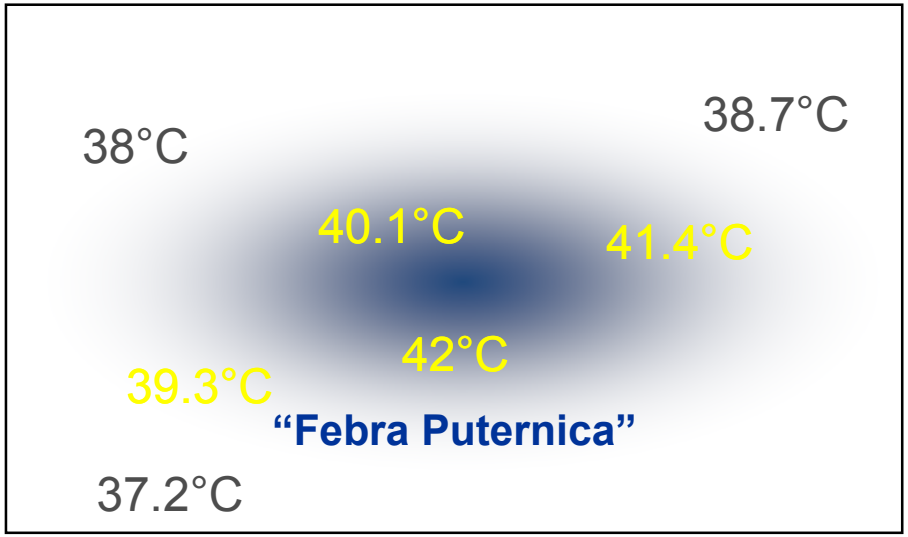
Logica conventionala (booleana):



“Mai mult sau mai putin” in locul

“ori asa – ori altfel” !

Teoria Fuzzy:



Ce este Logica Fuzzy?

- O extindere a logicii booleene care recunoaste mai multe valori de adevar decat adevarat si fals.
- In logica fuzzy propozitiile pot avea mai multe grade de adevar sau falsitate
  - Exemplu: “Afara este soare.”
    - 100% adevarat daca afara este senin
    - 80% adevarat daca este usor inorat
    - 50% adevarat daca este partial inorat
    - 0% adevarat daca ploua toata ziua
- Permite dispozitivelor de calcul sa “gandeasca” la fel ca oamenii.

Logica Fuzzy

- Este o teorie matematica ce permite lucrul cu informatii imprecise sau/si subiective
- Porneste de la o extensie a multimilor clasice, prin aceea ca in LF un element apartine *intr-un anumit grad* unei multimi

- Prezinta importanta practica deosebita inferenta fuzzy:
  - Se bazeaza pe variabile lingvistice (virsta, distanta, viteza, etc), care au termeni (tanar, batran sau viteza mica, medie, mare, etc)
  - Si pe un mecanism de inferenta fuzzy
    - Reguli de tip IF THEN, activate de fapte in diverse grade combinate conform unor relatii matematice

Istoric Fuzzy

- **1965 Lucrarea “Fuzzy Logic” scrisa de Prof. Lotfi Zadeh, Faculty in Electrical Engineering, U.C. Berkeley, pune fundatiile teoriei fuzzy**
- **1970 Prima aplicare a Controlului Fuzzy Logic (Europa)**
- **1975 Introducerea Fuzzy Logic in Japonia**
- **1980 Verificarea empirica a Fuzzy Logic in Europa**
- **1985 Aplicarea la scara larga a Fuzzy Logic in Japonia**
- **1990 Aplicarea la scara larga a Fuzzy Logic in Europa**
- **1995 Aplicarea la scara larga a Fuzzy Logic in U.S.A.**
- **2000 Fuzzy Logic devine o tehnologie standard si este aplicata in analiza de date si semnale. Aplicatii ale Fuzzy Logic in finante si business.**

Controllere Fuzzy

- Controllerele fuzzy sunt cele mai importante aplicatii ale logicii fuzzy si a teoriei din spatele ei
- Ele functioneaza foarte diferit de controllerele traditionale
  - In loc de ecuatii diferentiale, sistemul este modelat cu ajutorul cunostintelor dobandite in timp de catre experti
  - Aceste cunostinte sunt exprimate intr-un mod foarte natural folosind variabile lingvistice care sunt descrise de multimi fuzzy

Controllere Fuzzy - Aplicatii

- **Desi aplicarea fuzzy logic in rezolvarea si controlul sistemelor industriale a produs de foarte multe ori rezultate superioare controlului clasic, procedurile de design sunt limitate de regulile euristice ale sistemului.**

- **Aceasta constrangere implicita limiteaza numarul de aplicatii ale unui controller fuzzy**
  - **De cele mai multe ori, majoritatea controllerelor fuzzy au fost folosite in procese statice si bazate pe reguli derivate din cunostintele empirice ale unor operatori experimentati.**

Multimi Fuzzy

- Teoria multimilor fuzzy se ocupa de caracterizarea submultimilor unui domeniu de reprezentare U.
- O multime fuzzy $F \in U$ este o generalizare a conceptului de multime obisnuita si este identificata printr-o functie de apartenenta $\mu_F : U \to [0,1]$ in loc de valorile 0 si 1 ale unei multimi booleene obisnuite.

F este reprezentata de obicei ca o multime de perechi ordonate alcatuite din elementele u si gradul lor de apartenenta:

$$F = \{(u, \mu_F(u)) | u \in U\}$$

Daca U este un domeniu continuu, F poate fi exprimata dupa formula

$$F = \int_U \mu_F(u)/u$$

Daca U este discret, atunci:

$$F = \sum \mu_F(u_i)/u_i$$

Apartenenta la un set Fuzzy

Valori Discrete:

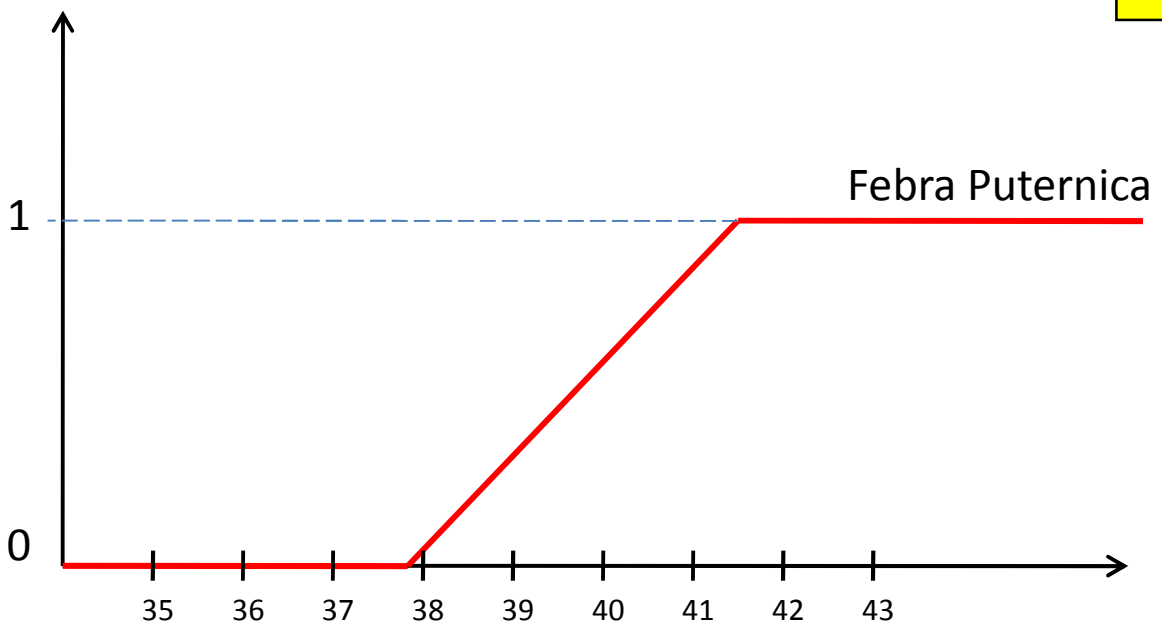
$\mu_{SF}(35°C) = 0$ $\mu_{SF}(38°C) = 0.1$ $\mu_{SF}(41°C) = 0.9$

$\mu_{SF}(36°C) = 0$ $\mu_{SF}(39°C) = 0.35$ $\mu_{SF}(42°C) = 1$

$\mu_{SF}(37°C) = 0$ $\mu_{SF}(40°C) = 0.65$ $\mu_{SF}(43°C) = 1$

Definitie Continua:

Fara praguri “abrupte”



Operatii pe seturi fuzzy

- Fie $A, B \in U$ doua multimi fuzzy cu functiile de apartenenta asociate u_A si u_B
- **Se pot defini urmatorii operatori:**
- Doua multimi A si B sunt egale daca si numai daca: $\mu_A(u) = \mu_B(u) \quad \forall u \in U$
- –Reuniunea celor doua multimi are functia de apartenenta

$$\mu_{A \cup B}(u) = \max(\mu_A(u), \mu_B(u)) = \mu_A(u) \vee \mu_B(u) \quad \forall u \in U.$$

- Intersectia celor doua multimi are functia de apartenenta:

$$\mu_{A \cap B}(u) = \min(\mu_A(u), \mu_B(u)) = \mu_A(u) \wedge \mu_B(u) \quad \forall u \in U.$$

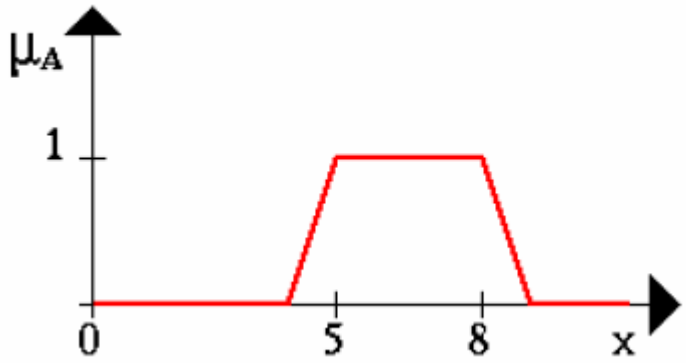
Operatii pe seturi fuzzy

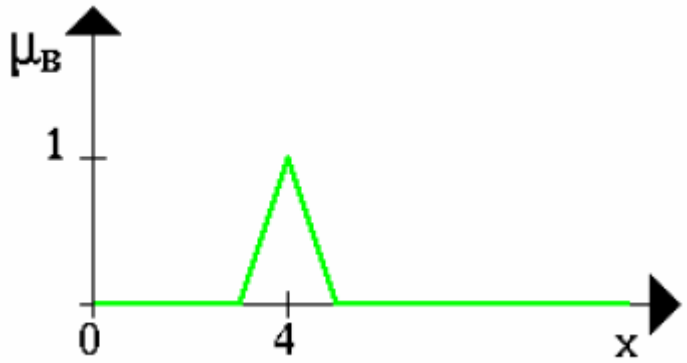
- Complementul lui A, A’ are functia de apartenenta:

$$\mu_{A'}(u) = 1 - \mu_A(u) \quad \forall u \in U.$$

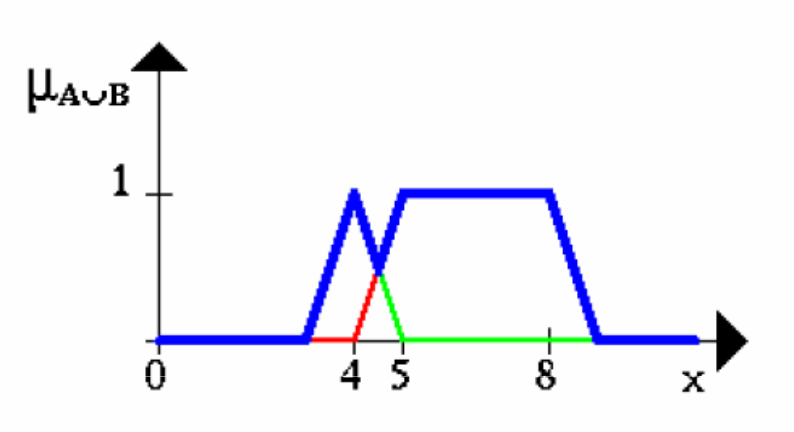
- **Exemplu:**

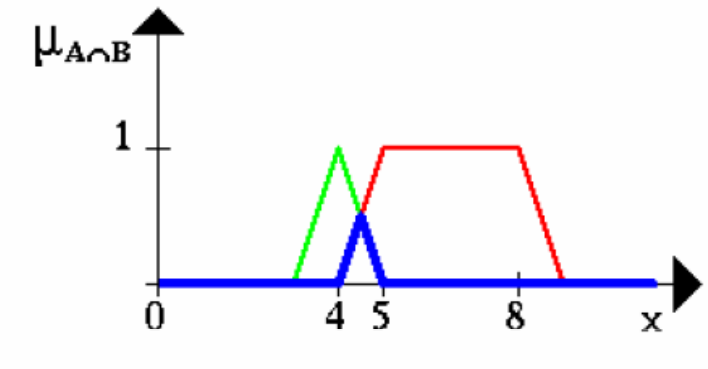
Fie A multimea “intre 5 si 8” si B “aproape 4”

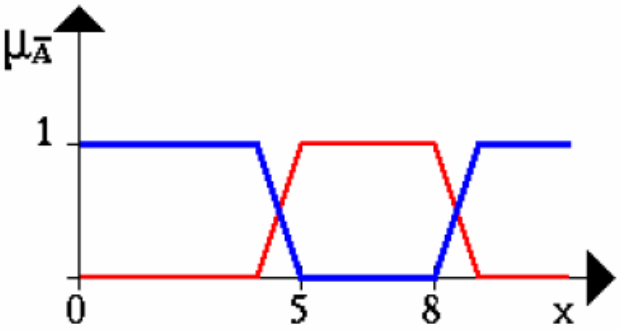




Operatii pe seturi fuzzy

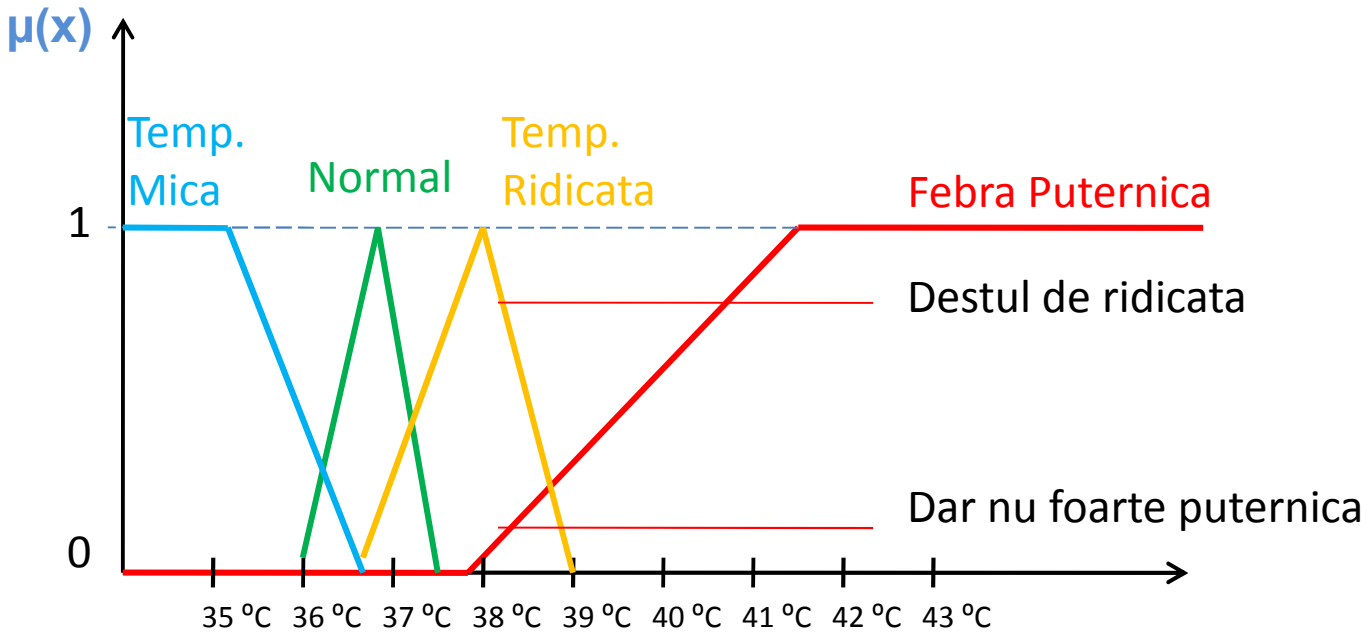






Variabile Lingvistice

- O variabila lingvistica defineste un concept din limbajul natural.

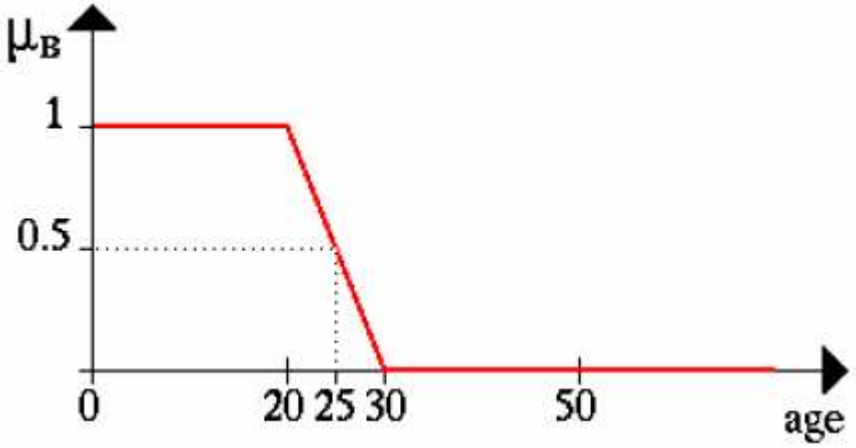


Variabile Lingvistice

- **O variabila lingvistica asociaza cuvinte sau propozitii cu o functie de apartenenta**
- Multimea de valori pe care fiecare variabila lingvistica le poate avea se cheama setul de termeni.
- Fiecare valoare din multime este o variabila fuzzy definita peste o variabila sa baza
- Variabilele de baza definesc domeniul de reprezentare pentru toate variabilele fuzzy din multimea de termeni
- **O variabila lingvistica este defapt un quintuplu [*X, T(X), U, G, M] unde***
  - *X este numele variabilei*
  - *T(X) este multimea de termeni (multimea numelor pentru valorile lingvistice ale lui X)*
  - *U domeniul de reprezentare,*
  - *Geste gramatica cu ajutorul careia se genereaza numele*
  - *M este un set de reguli semantice care asociaza fiecare X cu intelesul sau*
- Ierarhia: variabila lingvistica -> variabila fuzzy -> variabila de baza

Variabile Lingvistice

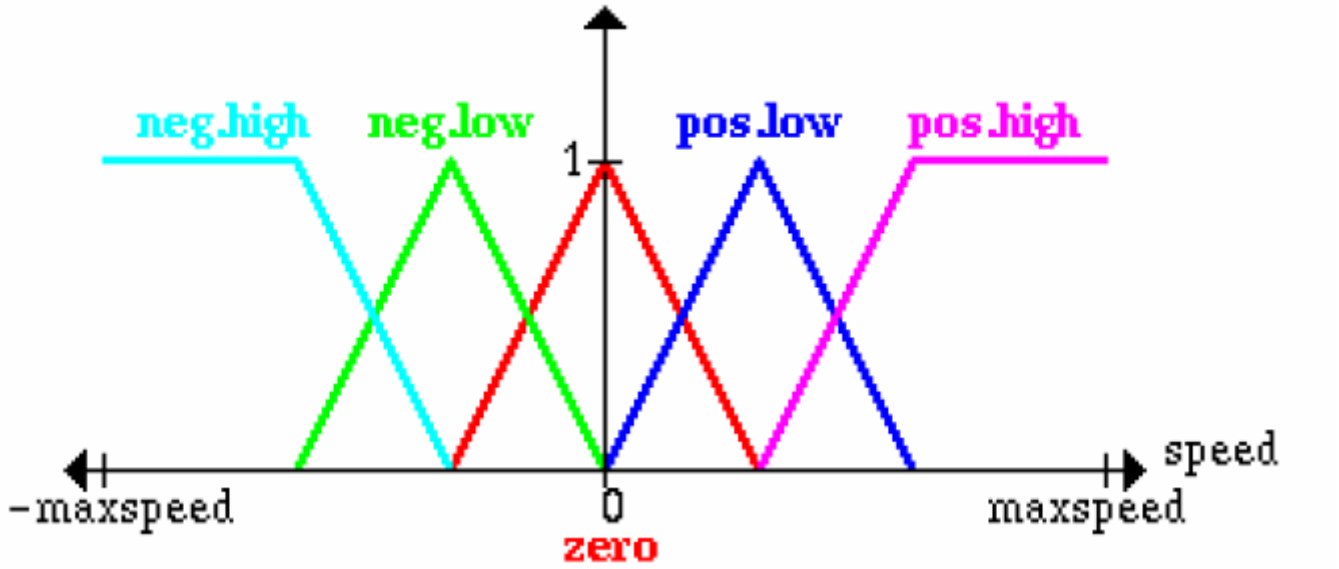
- **Fie x o variabila lingvistica denumita “Varsta”.**
- Termenii lui x, care sunt multimi fuzzy, pot fi: “batran”, “tanar” si “foarte tanar”din multimea de termeni **T= {Batran, *FoarteBatran, NuPreaBatran, MaiMultSauMaiPutinTanar, Tanar, FoarteTanar}***
- Fiecare termen este o variabila fuzzy definita peste variabila de baza, care poate fi o scala de la 0 la 100



Functia de apartenenta pentru x= “FoarteTanar”

Exemplul 2

- Fie x o variabila lingvistica numita “viteza”.
- Termenii lui x, care sunt multimi fuzzy, pot sa fie*”positive* low”, ”negative high” din multimea de termeni T= {*PositiveHigh, PositiveLow, NegativeLow, NegativeHigh, Zero}*
- Fiecare termen este o variabila fuzzy definita peste o variabila de baza care poate fi o scala cu toate vitezele relevante aplicatiei:



Controlul Fuzzy

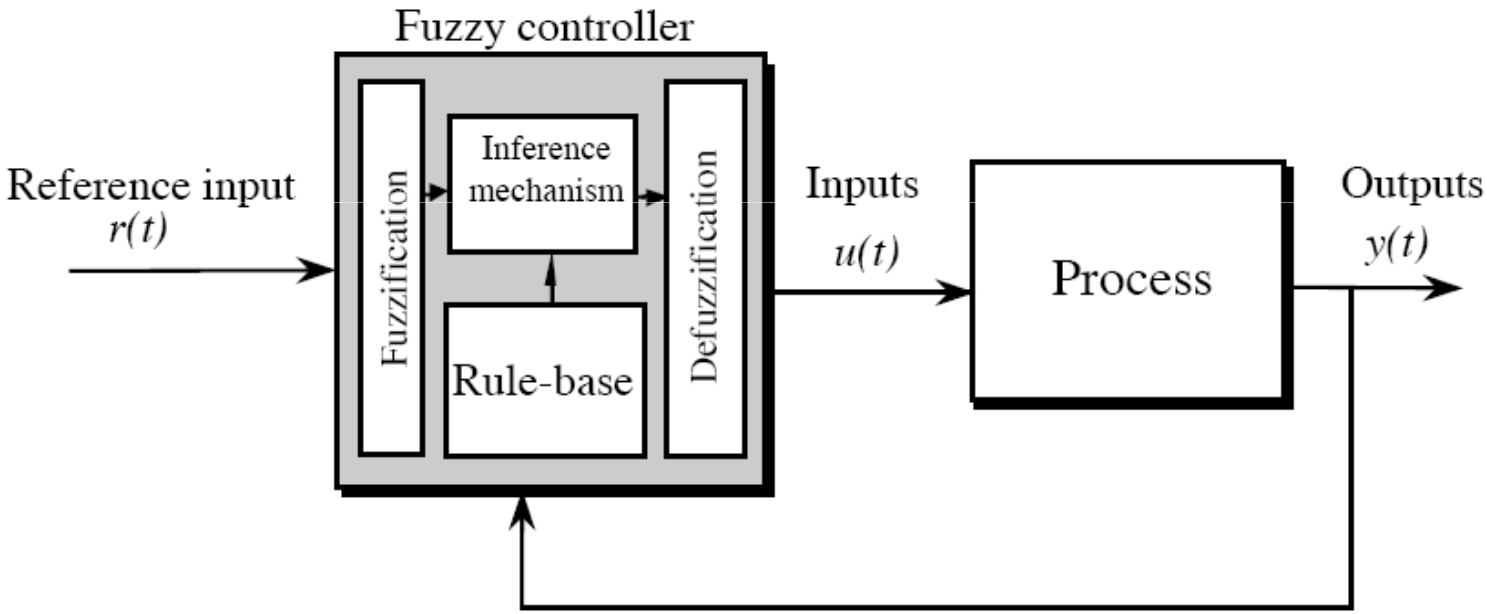
- Abordarea bazata pe fuzzy logic pentru rezolvarea problemelor de control este foarte indicata in sistemele foarte complexe, neliniare si care prezinta incertitudini pentru parametrii interni sau de intrare
- Un controller fuzzy poate fi vazut ca un sistem expert de timp real care foloseste logica fuzzy pentru a analiza raportul intrare/iesire al sistemului.
- Controllerele fuzzy pun la dispozitie o metoda pentru convertirea unei strategii de control empirice si specificate doar la nivel lingvistic (ex. “daca suna sirena atunci apasa butonul rosu”) intr-o strategie automata de control care poate sa furnizeze evolutia in timp a sistemului controlat si sa faca o estimare a performantelor acestuia

Controlul Fuzzy

- Elementele unui controller fuzzy:

1. Un *set de reguli* (reguli If-Then) ce reprezinta cuantificarea descrierii lingvistice a expertului despre cum se poate obtine un control bun asupra sistemului.
2. Un *mecanism de inferenta* (motor de inferenta, modul de inferenta fuzzy) care emuleaza procesul prin care expertul ia decizii prin interpretarea si aplicarea cunostintelor despre cum trebuie sa fie controlat sistemul.
3. O interfata de *fuzzificare* - converteste datele de la intrarile controllerului intr-o forma in care mecanismul de inferenta poate sa activeze si sa aplice anumite reguli.
4. O interfata de *defuzzificare* – converteste concluziile mecanismului de inferenta in comenzi si date de intrare pentru sistemul controlat.

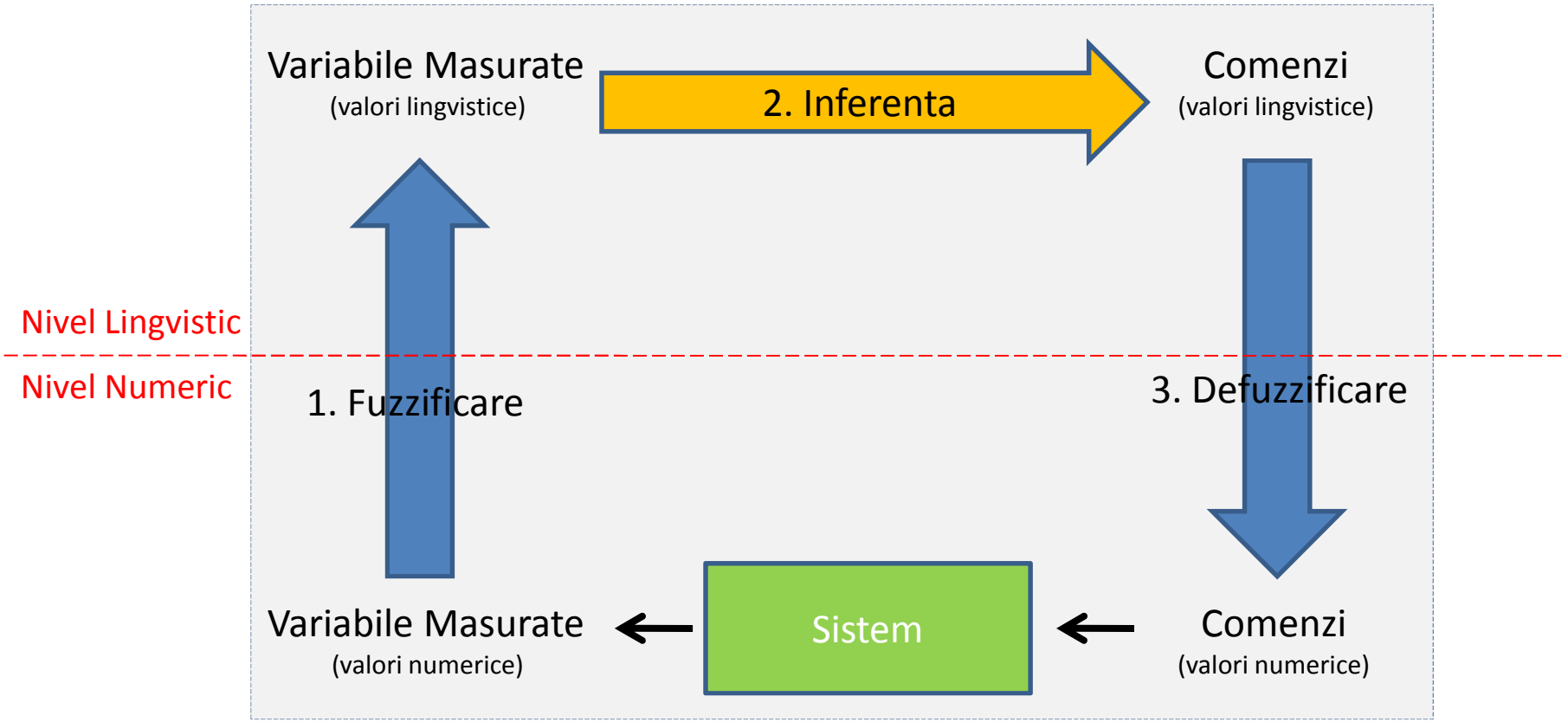
Controller-ul Fuzzy



Elementele de Baza ale unui Sistem de Control Fuzzy

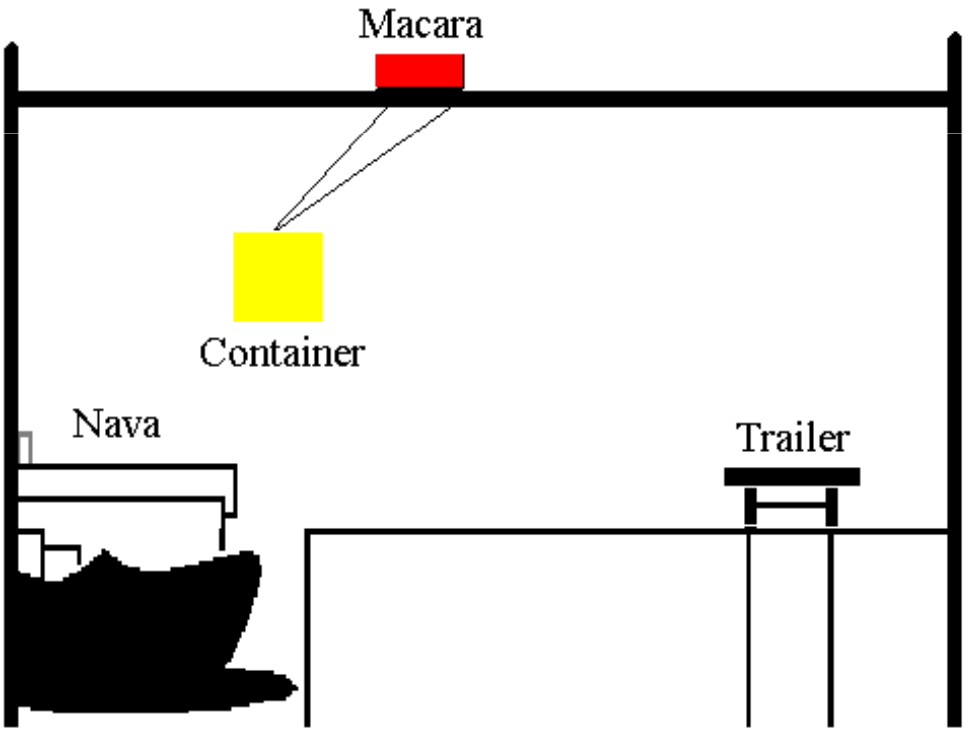
Fuzzificare, inferenta, defuzzificare:

Logica Fuzzy defineste strategia de control la nivel lingvistic!



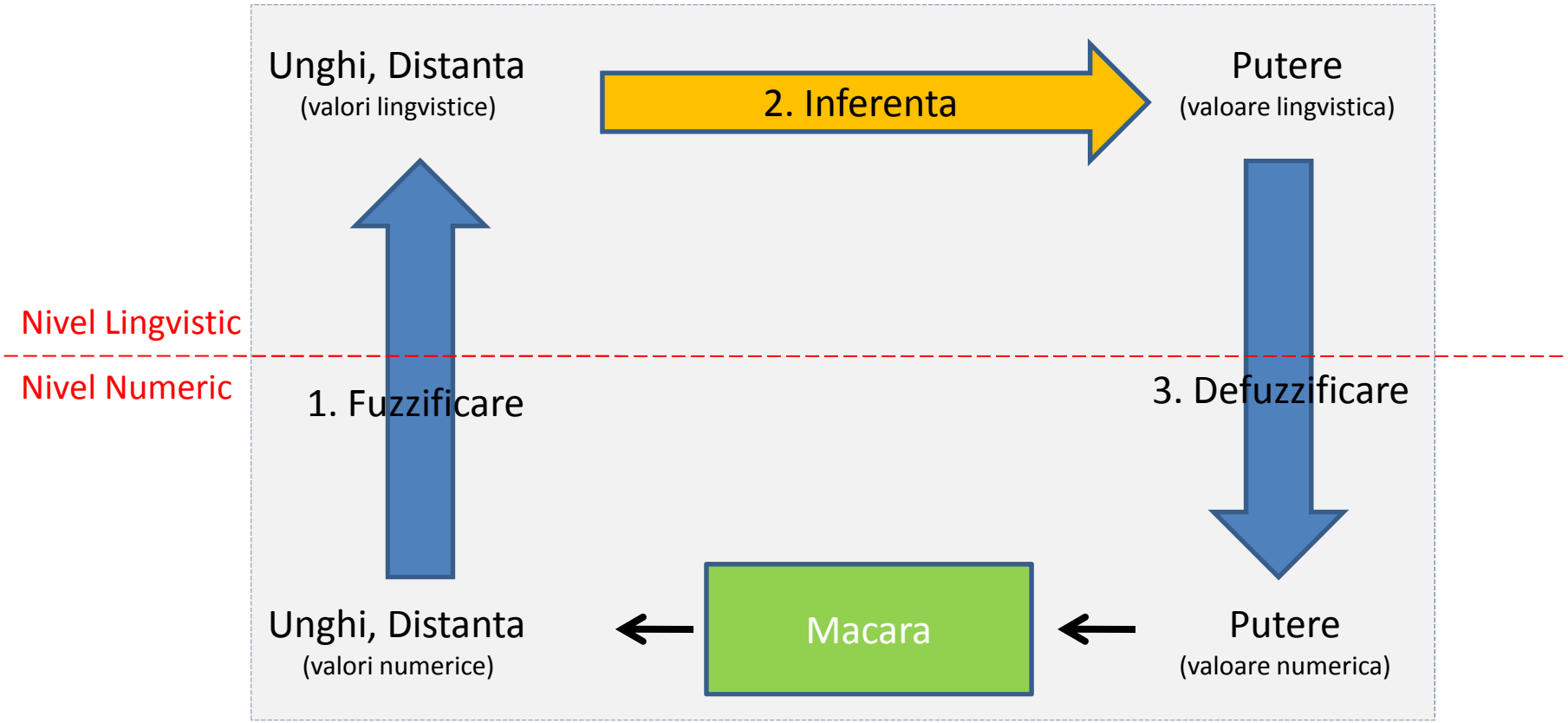
Exemplu: Controlul unei macarale

Doua variabile de masura si o variabila de comanda



Fuzzificarea Sistemului

Fuzzificare, inferenta, defuzzificare:

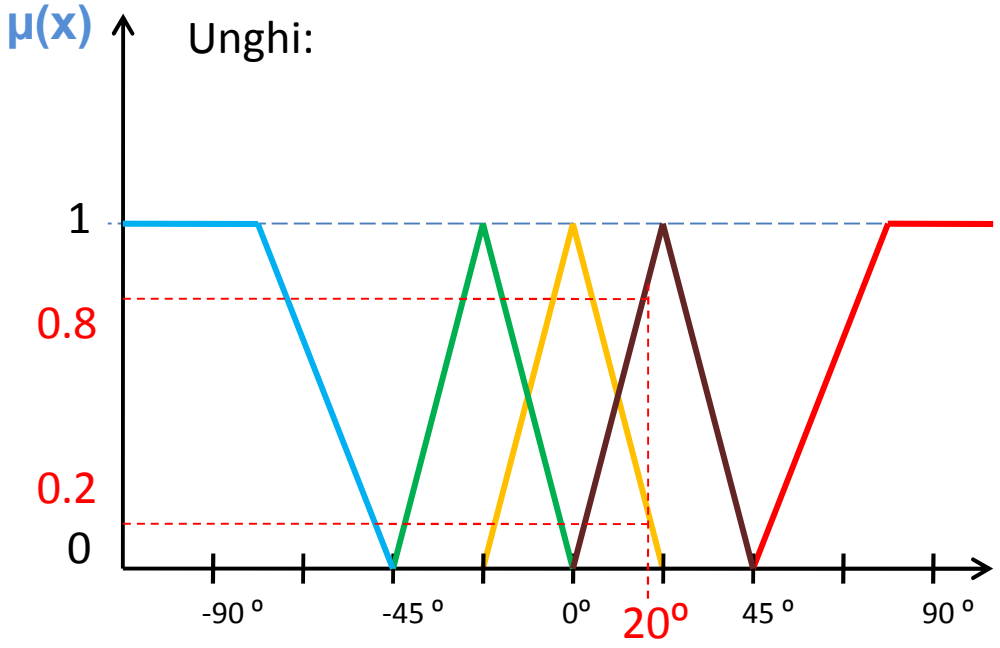


Variabile Lingvistice

Definitia Termenilor:

Variabilele lingvistice sunt “vocabularul” sistemului fuzzy.

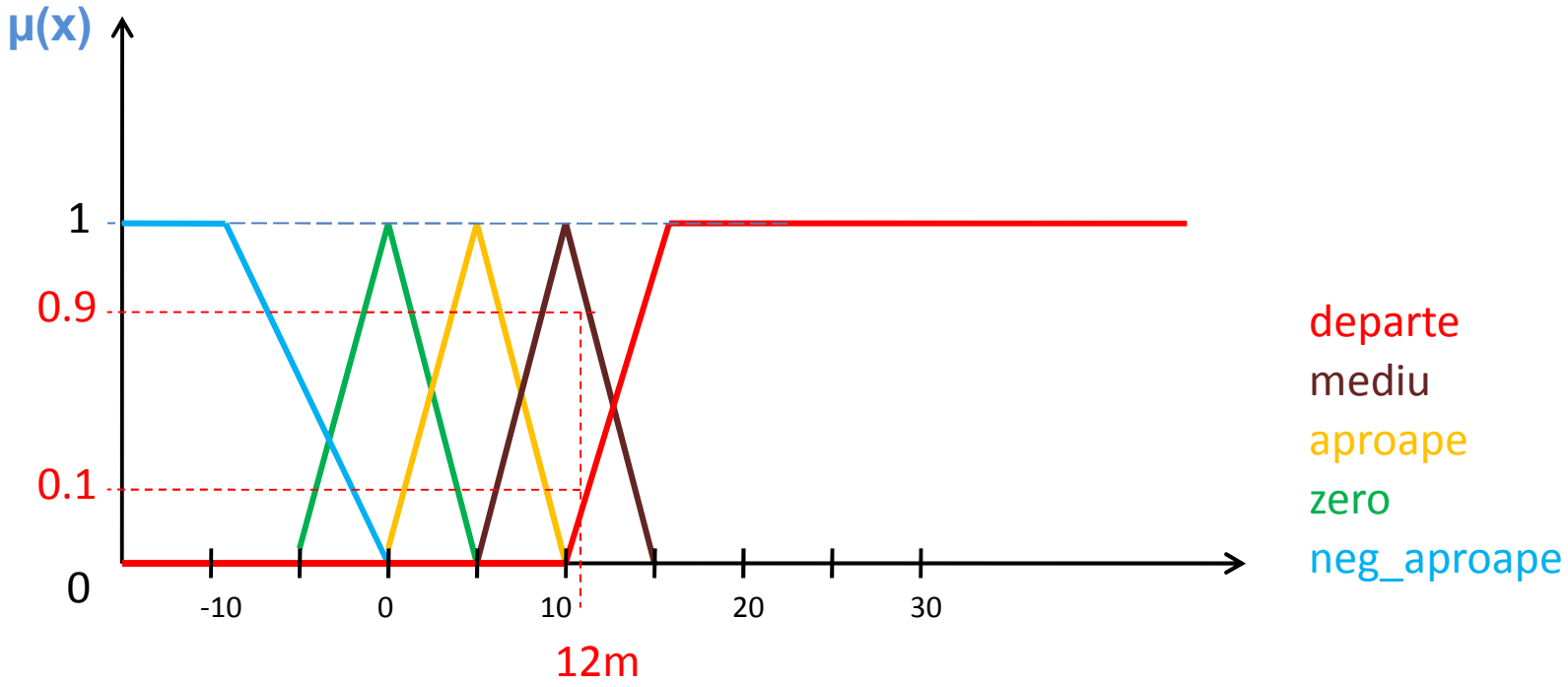
Distanta := {departe, mediu, aproape, zero, neg_aproape}
Unghi := {poz_mare, poz_mic, zero, neg_mic, neg_mare}
Putere := {poz_mare, poz_mediu, zero, neg_mediu, neg_mare}



poz_mare neg_mare
poz_mic neg_mic
zero

Variabile Lingvistice

Distanta:



Inferenta Fuzzy: Reguli IF-THEN

Stabilirea regulilor “IF-THEN”:

#1: IF Distanta = medie AND Unghi = poz_mic THEN Putere = poz_mediu

#2: IF Distanta = medie AND Unghi = zero THEN Putere = zero

#3: IF Distanta = departe AND Unghi = zero THEN Putere = poz_mediu

- **Agregare:** **Evaluarea partii “IF”**
- **Compozitie:** **Evaluarea partii “THEN”**

Regulile sistemului fuzzy sunt “legile” pe care acesta la executa.

Inferenta Fuzzy: Agregare

Logica Booleana defineste operatori doar pentru 0 si 1:

A	B	AvB
0	0	0
0	1	0
1	0	0
1	1	1

Logica Fuzzy ofera o extensie continua:

- **AND:** $\mu_{AvB} = \min\{ \mu_A; \mu_B \}$
- **OR:** $\mu_{A+B} = \max\{ \mu_A; \mu_B \}$
- **NOT:** $\mu_{-A} = 1 - \mu_A$

Agregarea partii "IF"

#1: min{ 0.9, 0.8 } = 0.8

#2: min{ 0.9, 0.2 } = 0.2

#3: min{ 0.1, 0.2 } = 0.1

Agregarea calculeaza cat de potrivita este fiecare regula pentru situatia curenta.

Inferenta Fuzzy: Compozitia

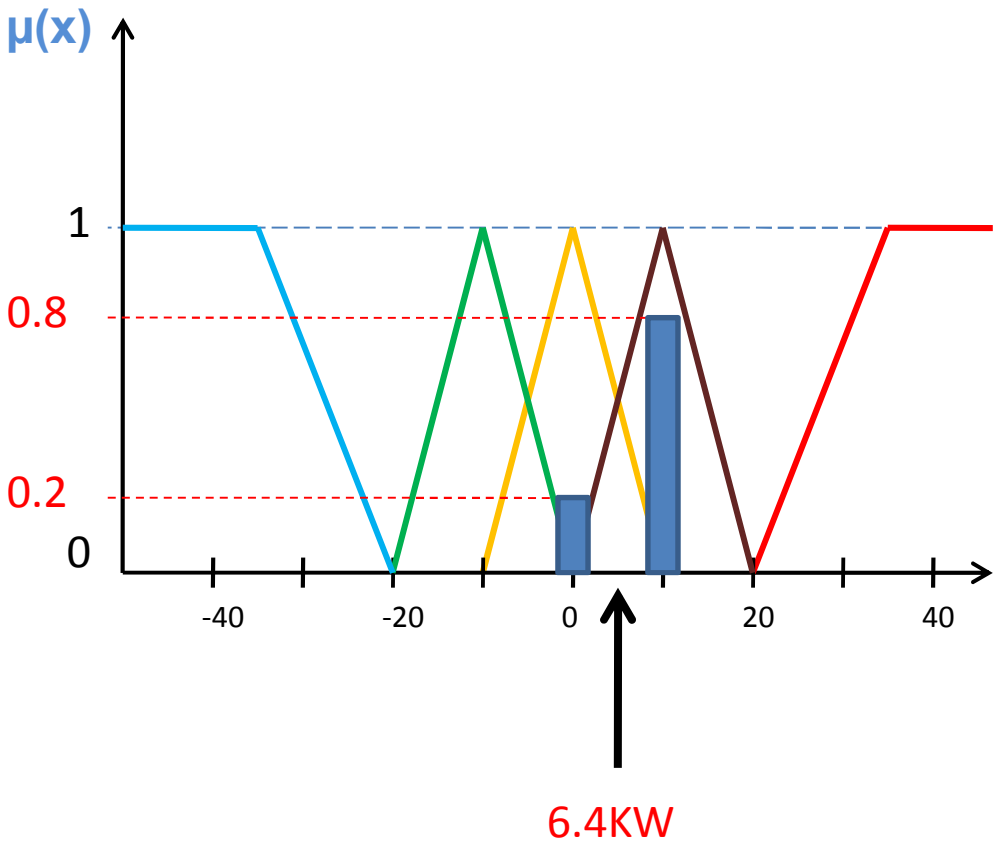
Rezultate pentru variabila lingvistica “Putere”:

poz_mare	**cu gradul 0.0**	
poz_mediu	**cu gradul 0.8**	**(= max{ 0.8, 0.1 })**
zero	**cu gradul 0.2**	
neg_mediu	**cu gradul 0.0**	
neg_mare	**cu gradul 0.0**	

Compozitia estimeaza cum fiecare regula modifica rezultatul de la iesire

Defuzzificare

Putere: Gaseste un compromis folosind o valoare medie ponderata



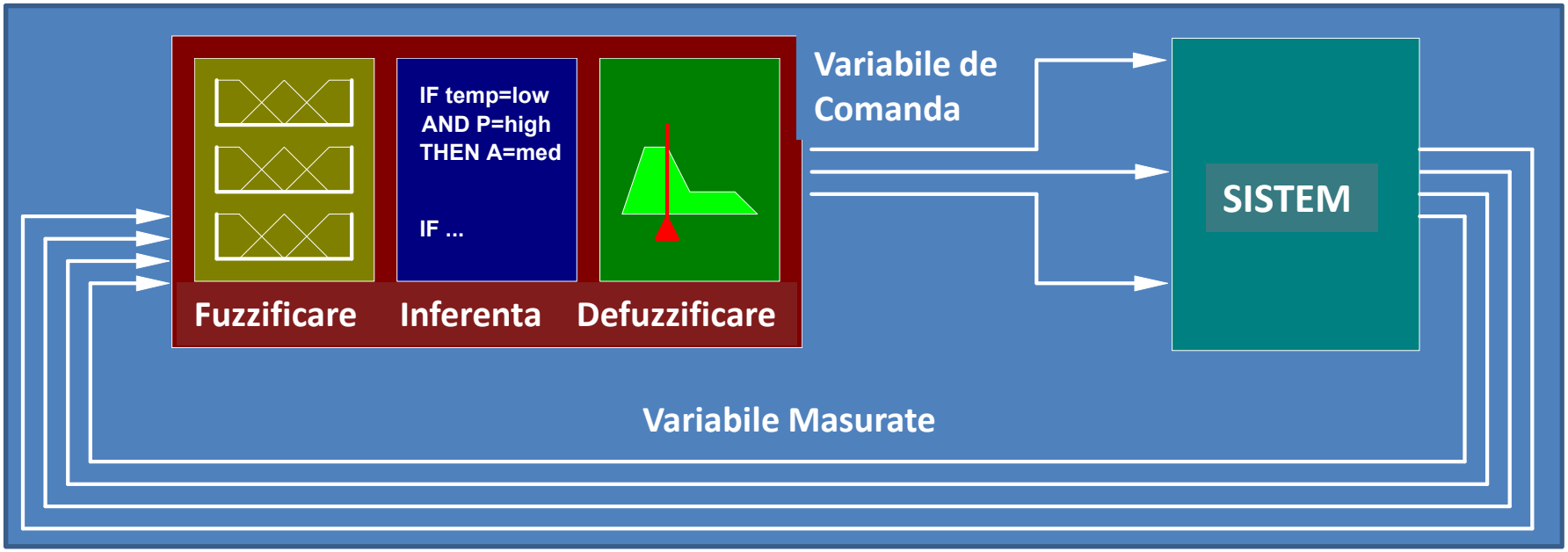
poz_mare neg_mare

poz_mediu neg_mediu

zero

Tipuri de Control Fuzzy: Controller Direct

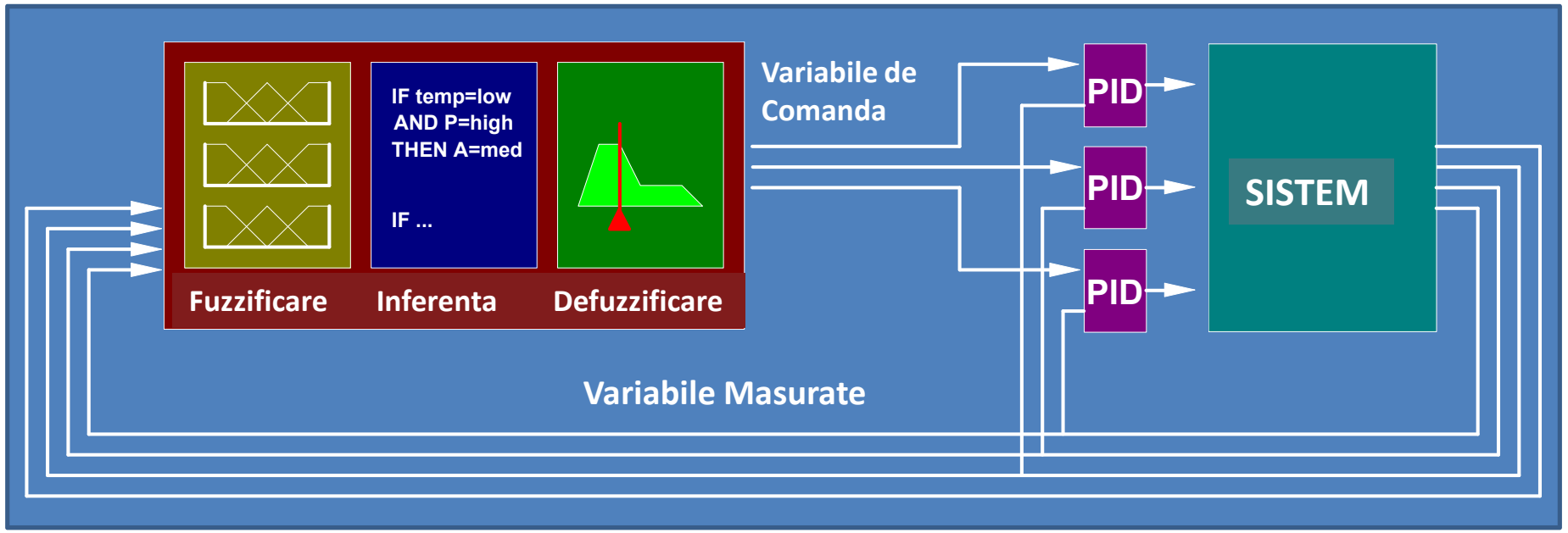
Iesirile de comanda ale sistemului fuzzy logic controleaza direct sistemul



Regulile Fuzzy produc valori exacte!

Tipuri de Control Fuzzy:
Control cu supervizare

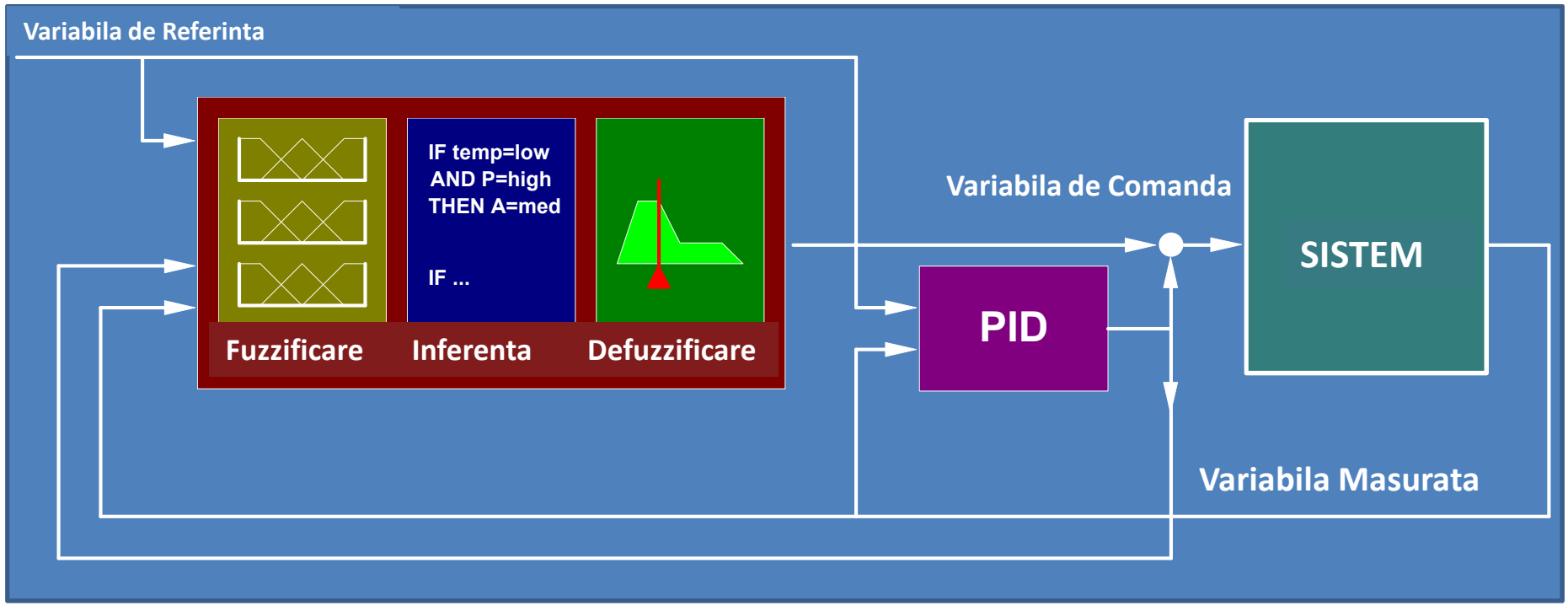
Controllerul Fuzzy Logic produce valori de setare pentru Controllerele PID:



Control de tipul "Operator Uman"

Tipuri de Control Fuzzy: Interventia Fuzzy

Controllerul Fuzzy Logic si Controllerul PID lucreaza in paralel:

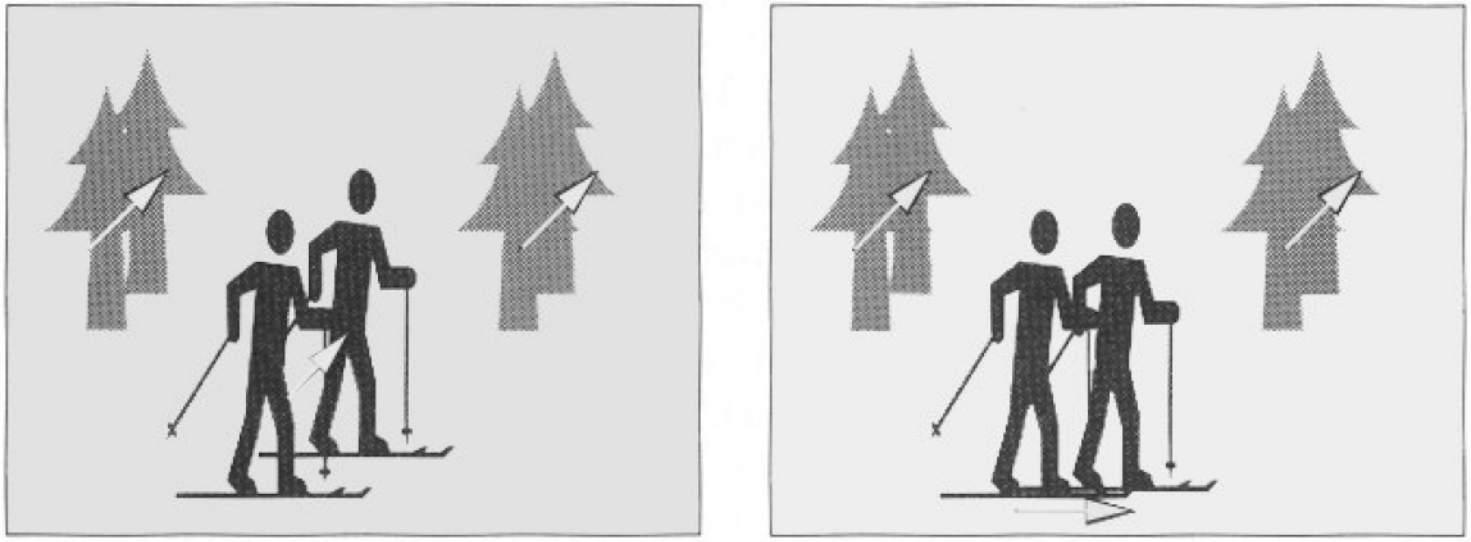


Interventia controlului fuzzy la perturbatii majore.

Aplicatii ale Controlului Fuzzy

- Stabilizarea imaginii:

  “Daca toti vectorii de miscare ai unei imagini sunt aproape paraleli si variatia lor in raport cu timpul este relativ mica, atunci este detectat tremurul mainii si directia de miscare a mainii este inversa directiei de miscare a vectorilor.



Fuzzy Logic: Aplicatii

- Business
  - Luarea de decizii
  - Sisteme de data mining
- Chimie
  - Dozarea substantelor in reactii
  - Reglarea conditiilor de reactie
- Comunicatii
  - QoS
  - Filtre adaptative
- Finante
  - Managementul fondurilor
  - Previziuni la bursa
- Robotica
  - Controlul efectoarelor
  - Determinarea pozitiei
- Armata
  - Determinarea tintei
  - Sisteme de ghidare
- Transporturi
  - Sisteme de transport fara pilot
  - Controlul sistemelor de trafic
- Medical
  - Controlul presiunii arteriale in timpul operatiei
  - Diagnosticarea cancerului, bolii Alzheimer, diabetului
- Electronica
  - Sisteme de climatizare
  - Sisteme de temporizare: cuptoare, masini de spalat
- Industrie
  - Controlul temperaturii in furnale
  - Controlul tratamentului apelor curate/uzate
  - Controlul calitatii