

Sisteme fuzzy de control. Controller de temperatură.

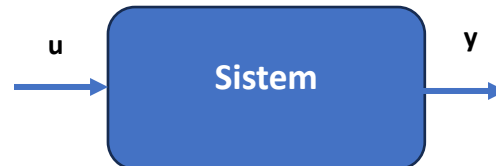
Obiective: înțelegerea conceptului de controller clasic, înțelegerea diferențelor dintre un controller clasic și un controller fuzzy, vizualizarea răspunsului unui controller fuzzy.

Observație: MATLAB/Simulink se accesează online (<https://matlab.mathworks.com/>), prin logare cu credențialele MS Teams (cele de tip nume.prenume@student.utcluj.ro).

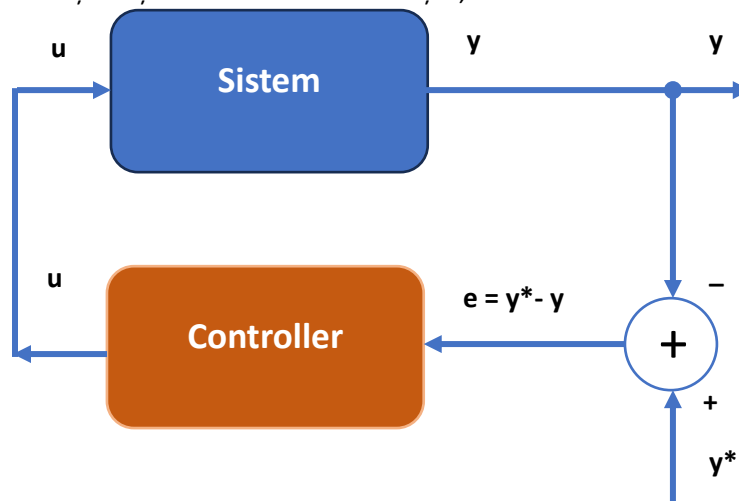
Termeni și acronime: *controller clasic, controller fuzzy, PID.*

○ Sisteme de control clasice

Fie un sistem în buclă deschisă, având o intrare de comandă u și o ieșire y :



Scopul sistemului este să asigure o anumită valoare dorită la ieșire, y^* . Sistemul este astfel proiectat încât, în absența perturbațiilor și a variațiilor parametrilor sistemului, $y = y^*$ pentru o anumită intrare $u = u^*$. Lipsa perturbațiilor nu apare niciodată în practică, adică y^* va fi diferit de y , dacă sistemul este lăsat să funcționeze în buclă deschisă, pentru intrarea de comandă $u = u^*$. Ca urmare, pentru a asigura $y = y^*$ în prezența perturbațiilor, este nevoie să se modifice u față de u^* , astfel încât să se compenseze acțiunea perturbațiilor. Modificarea u depinde de modificarea y față de y^* , și este realizată de un alt sistem, conectat între ieșirea și intrarea sistemului inițial, numit **controller**:



Noul sistem se numește *sistem în buclă închisă*, sau *sistem cu reacție*. Ieșirea u a controllerului reprezintă intrarea de comandă a sistemului și depinde, în general, de erorile la momentele de timp

anterioare: diferențele dintre ieșirea y și ieșirea dorită y^* , dar și de comenzile u la momentele de timp anterioare:

$$u(k) = f(e(k), e(k-1), \dots, e(k-t), u(k-1), \dots, u(k-t))$$

unde f definește legea de control, iar t este ordinul controllerului. Pentru $t > 0$, controllerul este cu *memorie*.

Eroarea e se definește ca:

$$e(k) = y^* - y$$

În general, legea de control f este neliniară. În teoria clasică a controlului, legea de control f este dedusă pe baza modelului matematic al procesului în buclă deschisă.

Legile clasice de control sunt:

a) legea de control proporțională (P):

$$u = K_p * e \Rightarrow u(k) = K_p * e(k)$$

b) legea de control integrală (I):

$$u = K_i * \int e dt$$

sau, în domeniul discret:

$$u(k) = K_i * \sum_{j=0}^{\tau} e(k-j)$$

c) legea de control derivativă (D):

$$u = K_d * \frac{d^\tau e}{dt^\tau}$$

d) combinații ale acestor legi - de exemplu ,controllerul PI:

$$u(k) = K_p * e(k) + K_i * \sum_{j=0}^{\tau} e(k-j)$$

○ Controllere fuzzy

Pentru un sistem fuzzy cu intrările $e(k), e(k-1), \dots, e(k-t), u(k-1), \dots, u(k-t)$, se poate formula o dependență lingvistică între ieșirea $u(k)$ și aceste intrări. Controllere fuzzy utilizate în practică sunt de ordin $t = 1$, astfel:

$$u(k) = f(e(k), e(k-1), \dots, u(k-1))$$

Controllerele fuzzy tipice au o formă și mai simplificată, în sensul că nu țin cont de ieșirea anterioară $u(k-1)$; în regulile fuzzy sunt luate în considerare doar $e(k)$ și $e(k-1)$, iar ieșirea controllerului este variația ieșirii u , definită ca:

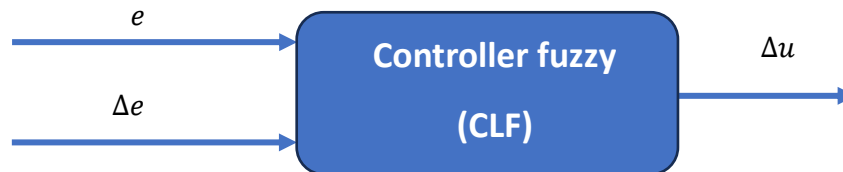
$$\Delta u(k) = u(k) - u(k-1) \Rightarrow u(k) = \Delta u(k) + u(k-1)$$

$$\Delta u(k) = F(e(k), e(k-1))$$

unde F este funcția de transfer a sistemului fuzzy, determinată de:

- mulțimile fuzzy definite peste intrări și ieșire
- baza de reguli
- mecanismul de inferență
- metoda de defuzzificare utilizată.

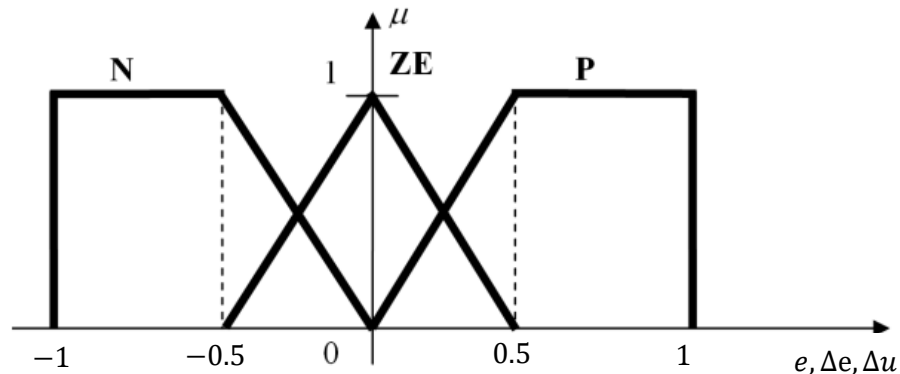
Tipic, intrările tranșante ale controllerului fuzzy sunt eroarea $e(k)$, derivata erorii $\Delta e(k) = e(k) - e(k - 1)$, iar ieșirea este $\Delta u(k) = F(e(k), e(k - 1))$.



Acest tip de controller fuzzy de ordin 1 ($t = 1$) a fost propus în 1975 de către Mamdani și Assilian, și se numește CLF de tip Mamdani.

○ Controller fuzzy de tip Mamdani - exemplu

Cel mai simplu model de controller fuzzy Mamdani prevede trei mulțimi fuzzy (**Negativ N**, **Zero ZE**, **Pozitiv P**), identic definite pentru cele două intrări, respectiv pentru ieșire:



Baza de reguli se deduce ținând cont de obiectivul dorit, $y = y^*$, adică $e = y^* - y = 0$. Privind nuanțat această valoare nulă a erorii, obiectivul dorit este "**e este ZE**". În plus, se consideră că ieșirea y a sistemului variază în același sens cu comanda u :

- dacă u crește, y crește;
- dacă u este constant, y este constant;
- dacă u scade, y scade;

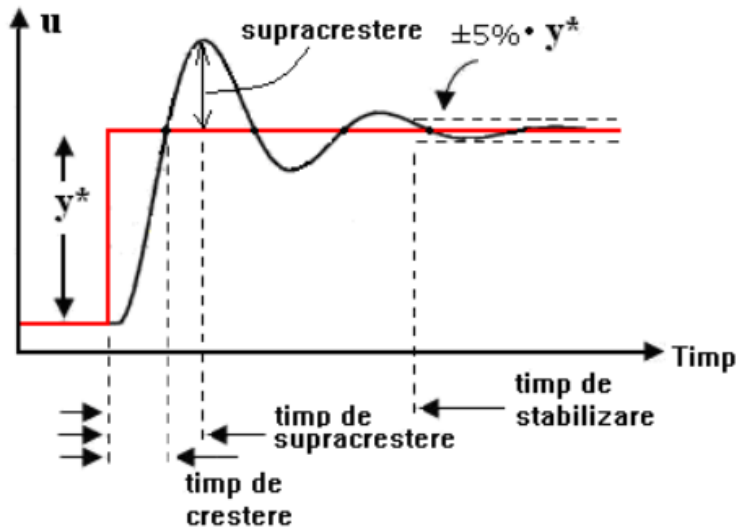
Baza de reguli conține toate combinațiile posibile ale valorilor lingvistice ale intrărilor (bază de reguli completă):

	e	N	ZE	P
Δe				
N		N	N	ZE
ZE		N	ZE	P
P		ZE	P	P

Mecanismul de inferență utilizat este Mamdani, adică inferență max-min, iar metoda de defuzzificare este COA (centroid).

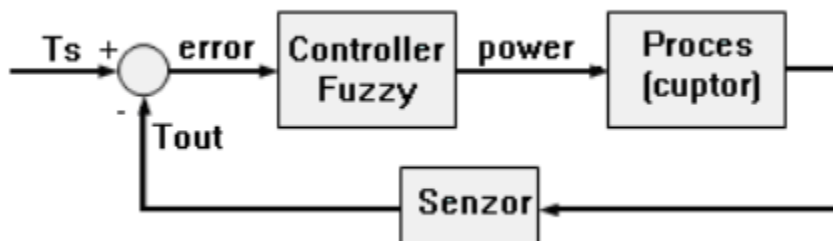
Răspunsul tipic al unui sistem automat de reglare la un semnal treaptă poate fi caracterizat de o serie de parametri:

- timpi de creștere/supracreștere/stabilizare
- supracreștere (mărimea primei supracreșteri peste valoarea stabilită).



o Controller fuzzy de temperatură

Sistemul de control al temperaturii se compune dintr-un proces (cuptorul) și controllerul fuzzy:

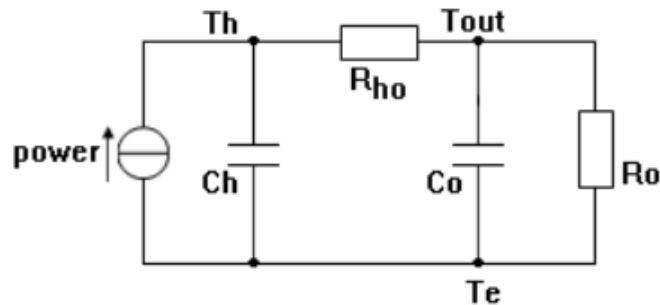


unde:

- T_s – temperatura dorită în incintă (*Set Point Temperature*)
- T_{out} – temperatura măsurată (temperatura provenită de la senzorul din incinta cuptorului)

- *error* – eroarea ($error = T_s - T_{out}$)
- *power* - puterea necesară încălzirii/răcirii (ieșirea controllerului de temperatură, respectiv intrarea de comandă în proces).

Rolul controllerului este de a menține temperaturii din cuptor egală cu temperatura dorită (T_s). Pentru modelarea generării și transferului de căldură, este folosit circuitul echivalent din figură:



Sursa de curent *power* (putere termică) reprezintă puterea furnizată elementului de încălzire/răcire. Sistemul cuprinde un element de încălzire/răcire electric, având capacitatea $Ch=500[J/°C]$ conectat printr-o rezistență $Rho=0.143[°C/W]$ la un cuptor cu capacitatea de încălzire $Co=1000[J/°C]$. Cuptorul degajă căldură în mediul exterior având temperatura Te , prin rezistența termică de transfer $Ro=0.1[°C/W]$. Controllerul de temperatură ajustează puterea disipată elementului de încălzire *power*, comparând temperatura cuptorului T_{out} cu temperatura de referință T_s .

Descărcați arhiva "*TempControl.zip*" și plasați conținutul acesteia (cu *drag-and-drop*) în directorul curent în care lucrează MATLAB. Dezarhivarea se face cu dublu click pe fișier.

<http://www.bel.utcluj.ro/dce/didactic/sf/lab/6ControlerTemperatura/TempControl.zip>

Schema bloc Simulink este prezentată în figură. Pentru a asigura "generalitatea" controllerului fuzzy, au fost utilizate două blocuri de conversie liniară la intrare (*Scale error*, *Scale delta_error*), și un bloc de conversie la ieșire (*Scale_power*). Pentru limitarea în interval $[-1, 1]$ a valorilor furnizate la intrarea controllerului, se utilizează blocurile de limitare dublă (*Saturation*, *Saturation1*).

Exercițiul 1

Implementați sistemul fuzzy Mamdani de control a temperaturii, conform specificațiilor definite anterior. Vizualizați suprafața de control. Salvați sistemul fuzzy creat cu numele „*TempControlM.fis*”. Citiți sistemul fuzzy creat într-o variabilă denumită *fls*, utilizând instrucțiunea *readfis*.

Exercițiul 2

Încărcați modelul Simulink al sistemului de control a temperaturii, „*TempControlM.mdl*”. Inițializați parametri cuptorului, prin rularea script-ului „*HeaterOven_params.m*”.

Porniți simularea și activați vizualizarea formelor de undă pe osciloscop (dublu-click pe blocul *Scope*). Măsurați parametrii sistemului de control: timpul de creștere, timpul de stabilizare, supracreșterea. Notați valorile obținute.

Exercițiul 3

Modificați factorii de scalare pentru intrări și ieșire, astfel încât performanțele sistemului să fie îmbunătățite.

Exercițiul 4

Reluați exercițiul 2, în ipoteza utilizării unui controller fuzzy de tip Takagi-Sugeno. Conversia între Mamdani și Takagi-Sugeno se face automat, cu opțiunea *Mamdani to Sugeno*. În acest caz, modelul Simulink utilizat va fi "TempControlTS.mdl". Care dintre cele două tipuri de sisteme de control are performanțe mai bune, pentru valorile inițiale ale factorilor de scalare?

