

**Proiect**

**Sisteme de conducere fuzzy**



Cosferent Roxana-Adelina

Îndrumător:

șef. lucr. dr. ing. CRISTIAN BOLDIȘOR

2021

**Cuprins:**

[1. Tema proiectului](#page3)……………………………………………………………………………………………………………………[3](#page3)

[2. Aspecte teoretice](#page3)……………………………………………………………………………………………….………………….[3](#page3)

[3. Realizarea implementării regulatorului cu ajutorul Matlab](#page6)………………………………………………………[6](#page6)

1. [Realizarea implementării regulatorului prin intermediul unui alt limbaj de programare](#page13)………….10

[5. Bibliografie](#page15)……………………………………………………………………………………………………………………………[11](#page15)

Anexa A……………………………………………………………………………………………………………………………………12

Anexa B……………………………………………………………………………………………………………………………………14

**1. Tema proiectului**

1. Se va proiecta un regulator fuzzy pentru controlul unui proces descris de funcția de transfer:

în care parametrii procesului și performanțele impuse sunt conform Tabelului 1.

B) Se va realiza o implementare a regulatorului proiectat, în limbajul C/C++ (sau alt limbaj cunoscut), care ulterior se va testa pentru diverse valori ale mărimilor de intrare ale sistemului de inferență prin comparație cu funcționarea aceluiași sistem de inferențe realizat în Matlab.

**Tabelul 1.**Date de proiectare

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Tema nr. | Student | Parametrii procesului | | | Performantele impuse | | |
|  |  |  |  |  |  |
| 34 | Cosferent Roxana-Adelina | 0.3 | 102 | 1 | 1% | 1% | 100 |

**2. Aspecte teoretice**

În cazul proiectelor cu regulatoare fuzzy există 2 metode de inferență des utilizate: Mamdani și Sugeno-Takagi. Inferența de tip Mamdani este metoda principală folosită la prezentarea noțiunilor de bază privind raționamentul fuzzy. Inferența de tip Mamdani pleacă de la o bază de reguli în care propozițiile consecință sunt propoziții fuzzy, forma generică a regulilor fiind:

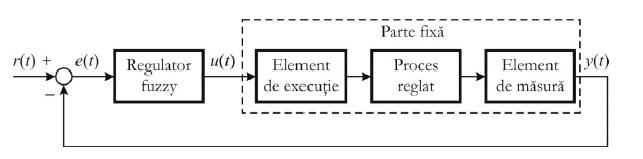
*Dacă x este X, atunci y este Y.*

Elementul care defineşte această metodă de inferență este utilizarea operatorilor: *min* pentru evaluarea implicației şi *max* pentru agregarea rezultatelor parțiale. Din acestmotiv, metoda se mai numeşte “inferență max-min” sau, datorită implicației *min*, “regula operatorului minimum a lui Mamdani”.

Pentru descrierea metodei de inferență de tip Mamdani, să considerăm o bază de reguli de tip Mamdani, cu două variabile de intrare – eroarea şi derivata acesteia – şi o variabilă de ieşire – mărimea de comandă, adică:

*Dacă e este E și de este DE, atunci u este U.*

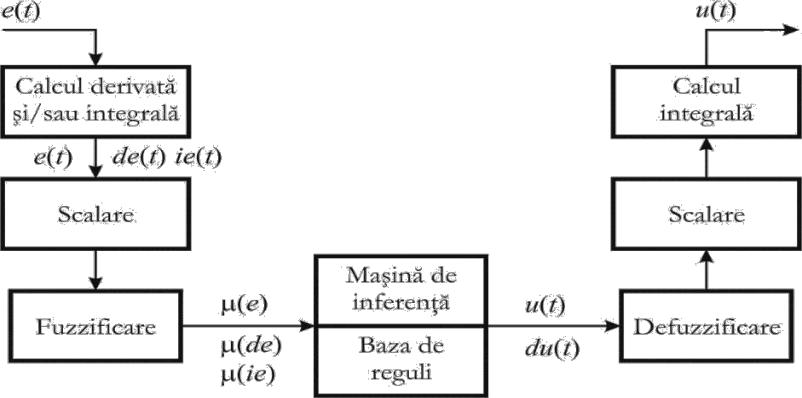
Pentru prezentarea noțiunilor de bază cu privire la regulatoarele fuzzy se consideră structura clasică de reglare din Figura 1, cu un singur regulator principal, aflat pe calea directă a sistemului.



**Figura 1.** Structură clasică de reglare

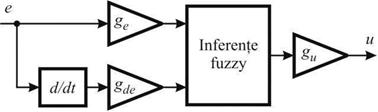
Structura de reglare convențională este un sistem cu reacție negativă, considerată unitară pentru o simplificare inițială. Pe calea de reacție este preluată mărimea de ieşire a părții fixe, y(t), care practic este un semnal achiziționat corespunzător mărimii supuse reglării. Mărimea de ieşire măsurată este comparată cu o mărime de referință, r(t), care defineşte evoluția dorită în timp a sistemului, iar diferența celor două reprezintă eroarea de reglare:

Un sistem de reglare cu regulator fuzzy are principalul avantaj al faptului că permite toleranță la incertitudine şi imprecizie, aspect demonstrat de numeroase aplicații practice.



**Figura 2.** Structura unui regulator fuzzy

Pentru regulatoarele fuzzy de tip PD, legea de reglare este:



**Figura 3.** Regulator fuzzy de tip PD

Variabilele lingvistice pentru un regulator fuzzy PD vor fi:

* eroarea;
* derivate erorii;
* comanda.

O regulă tipică din bază are acum formularea:

*Dacă e este E și este DE, atunci u este U.*

Realizarea unei proiectări corecte ține de respectarea unor reguli, cum ar fi:

* Pentru mulțimile de bază și termenii lingvistici:

Majoritatea aplicațiilor practice utilizează forme standard ale mulțimilor de bază ale variabilelor de intrare şi de ieşire. În general sunt două forme ale acestor mulțimi:

* 1. intervalul simetric în jurul lui zero - [−*L*; +*L*]
  2. intervalul asimetric [0; +*L*].

Variantele des întâlnite sunt [−1; +1], [−6; +6], [−100; +100] sau formele asimetrice [0; +1] , [0; +6] , [0; +100].

* Pentru setul de termini lingvistici:

Pentru fiecare variabilă de intrare se defineşte câte un set de atribute (termeni lingvistici, mulțimi fuzzy). Alegerea funcțiilor de apartenență ce descriu aceste atribute este din nou influențată de uşurința implementării şi rapiditatea calculelor. Astfel sunt preferate funcții de apartenență neliniare (triunghiulare sau trapezoidale), care presupun calcule simple ce pot fi efectuate foarte rapid de dispozitivele numerice. Lățimea acestor mulțimi fuzzy trebuie să asigure un anumit nivel de suprapunere a lor în universul de discurs, pentru a evita situații de incertitudine mare şi pentru a face regulatorul mai puțin sensibil la erori de măsurare. Este important să se evite existența în universul de discurs a unei variabile de intrare a unor zone neacoperite de nici un atribut. Uneori, în aplicațiile de control se acordă o atenție specială termenului lingvistic asociat valorii nule a erorii de reglare (de obicei, termenul „zero”). Acesta are o influență importantă asupra erorii staționare. În unele lucrări, pentru acest termen se adoptă o mulțime fuzzy mai îngustă, chiar cu nerespectarea unor condiții amintite anterior. Îngustarea acestui termen asigură deseori o îmbunătățire a erorii staționare.

* Pentru baza de reguli:

Alegerea unor mulțimi de bază standard, simbolice, în genul [−1; +1], permite stabilirea regulilor de control doar pe baza unor evaluări abstracte a erorii de reglare.

* Pentru factorii de scalare:

Alegerea factorilor de scalare este, în fapt, cea mai importantă etapă, dar şi cea mai dificilă. Aceştia au un rol determinant în funcționarea sistemului de reglare, iar alegerea eronată a lor înrăutățeşte performanțele, chiar până la instabilitate. Tehnicile de alegere a acestora, online sau offline, sunt subiectul multor lucrări ştiințifice, iar variantele sunt diverse. Varianta cea mai simplă propune ca valorile să fie alese în aşa fel încât valoarea maximă posibilă a fiecărei variabile în aplicația practică să corespundă valorii maxime din mulțimea de bază a acesteia. Dar se pot folosi și metode experimentale. Acestea constau în obținerea unui model aproximativ al procesului, prin înregistrarea răspunsului procesului supus reglării la un semnal de comandă de tip treaptă. Dacă răspunsul acestuia este în forma unui element de ordinul întâi, cu timp mort, atunci se folosesc relații de calcul a parametrilor regulatorului PID, din parametrii procesului.

**3. Realizarea implementării regulatorului cu ajutorul Matlab**

Conform datelor de proiect, funcția de transfer (1), devine:

De asemenea, valoarea nenulă a erorii oferă posibilitatea folosirii unui regulator PD, astfel se vor folosi atât eroarea, cât și derivata erorii, iar ieșirea regulatorului va fi mărimea de comandă. Din punct de vedere a inferenței folosite, aceasta este de tip Mamdani. Metoda de defuzificarea folosită este cea a mijlocului maxim (mom) pentru a facilita calculul în cazul verificării prin software a metodei folosite în Matlab.

Pentru definirea erorii (e), se vor folosi 4 funcții trapez și o funcție triunghiulară, definite în universul de discurs [-1; 1], astfel:

NB: funcție trapez cu parametrii [-1, -1, -0.4, -0.2];

NS: funcție trapez cu parametrii [-0.3, -0.15, -0.1, 0];

ZE: funcție triunghi cu parametrii [-0.1, 0, 0.1];

PS: funcție trapez cu parametrii [0, 0.1, 0.15, 0.25];

PB: funcție trapez cu parametrii [0.15, 0.25, 1, 1].

Pentru definirea derivatei erorii (de), se vor folosi 2 funcții trapez și o funcție triunghiulară definite în universul de discurs [-1; 1], astfel:

NE: funcție trapez cu parametrii [-1, -1, -0.9, 0];

ZE: funcție triunghi cu parametrii [-0.9, 0, 0.17];

PO: funcție trapez cu parametrii [0, 0.17, 1, 1].

Pentru definirea mărimii de comandă (u), se vor folosi 4 funcții trapez și o funcție triunghiulară definite în universul de discurs [-1; 1], astfel:

NB: funcție trapez cu parametrii [-1, -1, -0.25, -0.15];

NS: funcție trapez cu parametrii [-0.25, -0.15, -0.1, 0];

ZE: funcție triunghi cu parametrii [-0.1, 0, 0.2];

PS: funcție trapez cu parametrii [0, 0.2, 0.6, 0.9];

PB: funcție trapez cu parametrii [0.6, 0.9, 1, 1].

Setul de reguli este prezentat în următoarea tabelă de inferență:

**Tabelul 2.** Tabela deinferență

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| de/e | NB | NS | ZE | PS | PB |
| NE | NB | NB | PS | PB | PB |
| ZE | NB | NB | PS | PB | PB |
| PO | NB | NS | NS | PB | PB |

Folosind tool-ul fuzzy din Matlab obținem:

A screenshot of a computer

Description automatically generated

**Figura 4.** Reprezentarea mărimii de eroare cu ajutorul Matlab

A screenshot of a computer

Description automatically generated

**Figura 5.** Reprezentare derivatei erorii cu ajutorul Matlab

A screenshot of a computer

Description automatically generated

**Figura 7.** Reprezentarea bazei de reguli cu ajutorul Matlab

A screenshot of a computer

Description automatically generated

**Figura 8.** Modelde rezultat al mărimii de comandă cu ajutorul Matlab

În urma celor prezentate a fost proiectat un regulator fuzzy PD. Pentru a determina care sunt performanțele sistemului folosind acest regulator s-a realizat o simulare a sistemului de reglare folosind toolbox-ul Simulink. S-a considerat ca ge și gde să fie egal cu 0.875, iar gu egal cu 7.795.

A diagram of a computer program

Description automatically generated

**Figura 9.** Schema Matlab-Simulink a procesului în buclă închisă, cu reacție negativă și regulator fuzzy

Rezultatele obținute sunt satisfăcătoare deoarece se mențin în parametrii ceruți. Astfel, valorile obținute ale parametrilor sunt:

est = 0% < 1% (5)

M­v = 0% = 0% (6)

Ts = 105s < 140s (7)

­­­­A screen shot of a graph

Description automatically generated

**Figura 10.** Reprezentarea grafică a mărimii de ieșire

**4. Realizarea implementării regulatorului prin intermediul altui limbaj de programare**

Implementarea regulatorului obținut s-a realizat prin intermediul mediului de dezvoltare Visual Studio, folosind limbajul de programare C++. Procesul vizat are în vedere implementarea regulatorului proiectat folosind toolbox-ul din Matlab fuzzy. Programul conține atât funcțiile de reprezentare a mulțimiilor fuzzy și baza de reguli, cât și metodele de fuzificare, precum și defuzificare. Pentru procesul de defuzificare a fost implementată metoda centrelor de greutate. Rezultatele codului implementării regulatorului este prezentat în **Anexa A**.

În tabelul de mai jos sunt reprezentate diferențele dintre implementările sistemului fuzzy în limbajul C++, precum și cele folosind utilitarul din Matlab. (rezultatele experimentale ale implementării în Matlab sunt prezentate în **Anexa B**).

**Tabelul 2.** Diferențele dintre cele două implementări.

|  |  |  |  |
| --- | --- | --- | --- |
| Eroarea | Derivata erorii | Mărimea de comandă (C++) | Mărimea de comandă (Matlab) |
| -0.2 | 0.5 | -0.125 | -0.126 |
| 0.1 | 0.5 | 0.412 | 0.866 |
| -0.8 | 0.8 | -0.625 | -0.604 |
| -0.35 | 0.2 | -0.612 | -0.598 |
| 0.25 | 0.9 | 0.949 | 0.866 |

Putem observa o echivalență aproape perfectă între cele 2 metode. Având în vedere aproximările făcute în programul realizat cu ajutorul limbajului de programare C++, se poate spune că rezolvările sunt echivalente și corecte.

**5. Bibliografie**

1. Boldișor, C., *Sisteme de Conducere Fuzzy*, note de curs
2. Suciu, C., *Tehnici de inteligență artificială*, note de curs
3. [*https://www.mathworks.com/products/fuzzy-logic.html*,](https://www.mathworks.com/products/fuzzy-logic.html) *Fuzzy* Logic Toolbox

**Anexa A:** Rezultatele testării conform implementării în C++

A screenshot of a computer

Description automatically generated

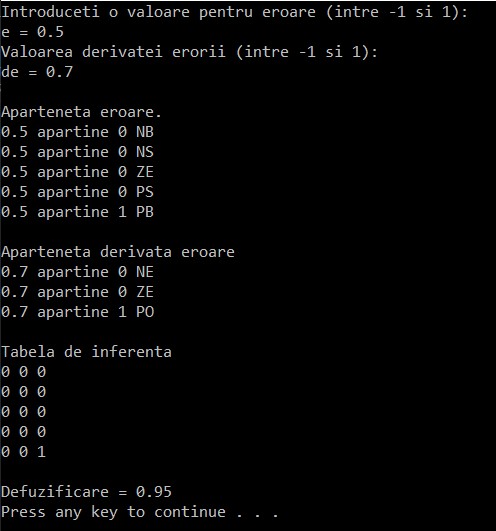
**Figura 11.** Rezultate testare Visual Studio pentru parametrii [e; de] = [-0.2; 0.5]

A screenshot of a white paper

Description automatically generated

A screenshot of a computer

Description automatically generated



**Anexa B:** Rezultatele testării conform implementării în Matlab

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated