



UNIVERSITATEA TEHNICĂ „GHEORGHE ASACHI” DIN IAȘI  
FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE  
Domeniul: Ingineria Sistemelor  
Programul de studii: Automatică și Informatică Aplicată



# PROIECT DE DIPLOMĂ

Coordonator științific:  
Conf. dr. ing. Anca Maxim

Absolvent:  
Gabriel Aganencei

Iași, 2025





UNIVERSITATEA TEHNICĂ „GHEORGHE ASACHI” DIN IAȘI  
FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE  
Domeniul: Ingineria Sistemelor  
Programul de studii: Automatică și Informatică Aplicată



# **Controlul temperaturii unui sistem de tip baie de apă**

**PROIECT DE DIPLOMĂ**

Coordonator științific:  
Conf. dr. ing. Anca Maxim

Absolvent:  
Gabriel Aganencei

**Iași, 2025**



## DECLARAȚIE DE ASUMARE A AUTENTICITĂȚII PROIECTULUI DE DIPLOMĂ

Subsemnatul \_\_\_\_\_ Agănenței Gabriel \_\_\_\_\_,  
legitimă cu CI seria XV nr. 711857, CNP 5020212336385 autorul  
lucrării “Controlul temperaturii unui sistem de tip baie de apă” elaborată în vederea  
susținerii examenului de finalizare a studiilor de licență, programul de studii  
Automatică și Informatică Aplicată organizat de către Facultatea de Automatică și  
Calculatoare din cadrul Universității Tehnice „Gheorghe Asachi” din Iași, sesiunea Iulie  
a anului universitar 2024-2025, luând în considerare conținutul Art. 34 din Codul de  
etică universitară al Universității Tehnice „Gheorghe Asachi” din Iași (Manualul  
Procedurilor, UTI.POM.02 - Funcționarea Comisiei de etică universitară), declar pe  
proprie răspundere, că această lucrare este rezultatul propriei activități intelectuale, nu  
conține porțiuni plagiate, iar sursele bibliografice au fost folosite cu respectarea  
legislației române (legea 8/1996) și a convențiilor internaționale privind drepturile de  
autor.

Data

28.06.2025

Semnătura





# Cuprins

<b>1</b>	<b>Introducere</b>	<b>1</b>
1.1	Clasificarea reguletoarelor . . . . .	1
<b>2</b>	<b>Regulatorul PID</b>	<b>4</b>
2.1	Prezentare generală a controlului proporțional-integral-derivativ (PID) . . . . .	4
2.2	Descrierea termenilor din regulatorul PID . . . . .	4
2.3	Metode de acordare a parametrilor PID . . . . .	5
2.3.1	Metoda alocării poli-zero-uri . . . . .	5
2.3.2	Metoda Haalman . . . . .	6
2.3.3	Metoda Lambda . . . . .	8
2.3.4	Metoda Zhuang Atherton . . . . .	9
2.3.5	Reglarea de tip Feedforward . . . . .	10
2.3.6	Reglare cu predictor Smith . . . . .	11
<b>3</b>	<b>Regulatorul MPC</b>	<b>12</b>
3.1	Prezentare generală a controlului predictiv bazat pe model (MPC) . . . . .	12
3.2	Formularea matematică a problemei QP în MPC . . . . .	13
3.2.1	Algoritmul KWIK – metodă active-set utilizată în MPC din MATLAB . . . .	14
<b>4</b>	<b>Prezentarea implementării soluțiilor</b>	<b>15</b>
4.1	Analiza comparativă a metodelor PID pe baza simulărilor . . . . .	16
4.1.1	Metoda Lambda (albastru închis) . . . . .	16
4.1.2	Metoda Haalman (roșu) . . . . .	16
4.1.3	Criterii integrale: ISE, ISTE, $IST^2E$ . . . . .	16
4.1.4	Predictorul Smith (galben punctat) . . . . .	17
4.1.5	Metoda de reglare utilizând structura Feedforward (turcoaz punctat) . . . .	17
4.1.6	Combinarea reguletoarelor PID cu structura Predictorului Smith . . . . .	18
4.1.7	Concluzii ale comparației . . . . .	19
4.2	Analiza reglării MPC . . . . .	22
4.2.1	Parametrii configurabili ai regulatorului MPC în MATLAB . . . . .	22
4.2.2	Selecția parametrilor pentru regulatorul MPC . . . . .	23
4.2.3	Impactul restricțiilor asupra comenzii în MPC . . . . .	25
4.3	Comparare MPC cu PID . . . . .	27
<b>5</b>	<b>Concluzii</b>	<b>30</b>
	<b>Bibliografie</b>	<b>31</b>
	<b>Anexe</b>	<b>33</b>
1	Cod MATLAB - reguletoare PID . . . . .	33
2	Cod MATLAB - regulator MPC utilizat . . . . .	40

3	Cod MATLAB - Generare MPC-uri . . . . .	41
4	Cod MATLAB - Comparare MPC-uri . . . . .	42
5	Cod MATLAB - MPC vs PID . . . . .	48



## Rezumat

Lucrarea de față propune o abordare practică a problemei controlului temperaturii într-un sistem de încălzire a apei, având ca scop reglarea eficientă a temperaturii într-o baie destinată bebelușilor. Sistemul analizat utilizează o rezistență electrică PTC de 11 kW, datorită comportamentului său autoreglant și a siguranței în aplicații casnice. Modelarea procesului termic s-a realizat printr-o funcție de transfer de ordinul întâi cu întârziere pură, dedusă din ecuațiile de echilibru energetic.

Pe baza acestui model, au fost implementate și comparate două clase de regulatori: regulatorul PID, configurat prin metode clasice de reglaj (Lambda, Haalman, ISE, ISTE, IST<sup>2</sup>E), și regulatorul predictiv MPC, dezvoltat în MATLAB cu ajutorul Model Predictive Control Toolbox, direct pe funcția de transfer, fără conversie în spațiul stărilor.

Pentru îmbunătățirea răspunsului sistemului, s-au introdus structuri adiționale precum Predictorul Smith și Controlul Feedforward, acestea fiind combinate cu regulatoarele PID pentru a reduce efectele întârzierii sistemului și a diminua suprareglarea.

Simulările realizate au inclus trei scenarii de testare: schimbarea valorii de referință, rejecția unei perturbații externe și comportamentul sistemului în prezența unei saturări pe semnalul de comandă. Evaluarea performanței a fost efectuată atât vizual (prin răspunsuri în timp), cât și cantitativ, folosind indicatori precum eroarea pătratică medie (MSE), timpul de reglare și deviația medie a temperaturii.

Concluziile subliniază avantajele metodei MPC în contexte cu constrângeri, dar și eficiența predictorului Smith în asociere cu metodele PID. Lucrarea oferă o bază solidă pentru implementări ulterioare, precum integrarea unor senzori reali sau extinderea sistemului pentru controlul simultan al mai multor variabile.



## Capitolul 1. Introducere

Încălzitoarele electrice sunt dispozitive ce transformă energia electrică în energie termică. În cadrul acestora, elementul de încălzire este reprezentat de un rezistor care produce căldură prin trecerea curentului electric. Încălzirea lichidelor este o formă de încălzire electrică prin care temperatura unui lichid este adusă la o valoare prestabilită, fiind utilizată în diverse domenii [1]. Controlul temperaturii este esențial pentru o varietate de aplicații, cum ar fi menținerea temperaturii optime a apei pentru diverse specii de pești, obținerea temperaturii ideale pentru cafea sau asigurarea unei temperaturi adecvate pentru apa de baie. Apa este una dintre cele mai utilizate substanțe în procesele de încălzire a lichidelor.

În cadrul controlului automat al proceselor dinamice, metodele de reglare pot fi clasificate în două mari categorii: metode clasice, precum controlul proporțional–integral–derivativ (PID), și metode moderne, cum ar fi controlul predictiv bazat pe model (Model Predictive Control – MPC). Alegerea unei strategii de control adecvate depinde de complexitatea sistemului, de prezența întârzierilor, de sensibilitatea la perturbații și de necesitatea respectării unor constrângeri asupra variabilelor de stare sau de control.

### 1.1. Clasificarea reguletoarelor

**Regulatorul PID** reprezintă una dintre cele mai răspândite și aplicate metode în industrie, datorită simplității conceptuale, ușurinței în implementare și eficienței în sisteme cu dinamică liniară sau cu întârziere redusă. Datorită flexibilității acestuia, putem defini mai multe metode de reglaj al parametrilor PID, fiecare având propriile avantaje și limitări. *Metoda Haalman*, derivată din abordarea clasică Ziegler–Nichols, permite determinarea rapidă a câștigurilor PID pornind de la o funcție de transfer simplificată a procesului. Este frecvent utilizată pentru obținerea unei reglări inițiale, însă sensibilitatea acesteia la modificările structurii sistemului limitează aplicabilitatea în medii variabile.

O altă metodă consacrată este *Lambda* (sau reglajul *IMC – Internal Model Control*), care oferă un echilibru între performanță și robustețe prin introducerea unui parametru de timp, notat  $\lambda$  (Lambda), ce dictează rapiditatea răspunsului sistemului. Această abordare este deosebit de utilă în cazul sistemelor cu întârzieri semnificative, unde stabilitatea are prioritate.

Pentru procesele cu întârziere pură, *predictorul Smith* reprezintă o soluție eficientă, prin compensarea întârzierii printr-un model predictiv. Astfel, regulatorul PID operează ca și cum întârzierile nu ar exista, ceea ce îmbunătățește semnificativ răspunsul sistemului. Complementar acestor metode, *controlul feedforward* poate fi integrat pentru a anticipa efectele unor perturbații cunoscute, îmbunătățind performanța în regim tranzitoriu.

Reglajul PID poate fi optimizat și prin intermediul unor criterii de performanță matematice, precum *ISE (Integral of Squared Error)*, *ISTE (Integral of Squared Time-weighted Error)* sau *IST<sup>2</sup>E (Integral of Squared Time-squared-weighted Error)*. Aceste criterii cuantifică eroarea sistemului în timp și sunt utilizate în procesul de ajustare fină a parametrilor PID pentru minimizarea deviațiilor față de referință.

Pe de altă parte, **controlul predictiv bazat pe model (MPC)** constituie o strategie modernă, adaptată sistemelor multivariabile și celor care impun respectarea unor constrângeri stricte. MPC utilizează un model matematic al procesului pentru a anticipa evoluția acestuia pe un orizont finit de timp și generează acțiuni de control prin optimizarea unei funcții de cost, la fiecare pas de eșantionare. Datorită acestei capacități de anticipare și adaptare, MPC este preferat în aplicațiile critice, unde controlul tradițional nu mai asigură performanțele necesare. Totuși, implementarea MPC presupune resurse computaționale semnificative și un model de proces precis, ceea ce poate limita aplicabilitatea în sistemele cu resurse reduse sau modele greu de identificat.

Această lucrare abordează cele două strategii de control utilizate pe scară largă în aplicațiile

industriale: *controlul predictiv bazat pe model (MPC)* și *controlul proporțional-integral-derivativ (PID)*. Obiectivul este de a vedea cum aceste două metode reușesc să gestioneze temperatura dintr-un bazin de apă, aplicație întâlnită în procesele de terapie din domeniul medical. Metodele vor fi examinate în termeni de stabilitate, timp de răspuns, robustețe la perturbări și precizie a controlului. Astfel, această cercetare își propune să evidențieze avantajele și dezavantajele controlului MPC și PID în sistemele de control al temperaturii.

Controlul sistemelor industriale are un rol esențial în monitorizarea și gestionarea parametrilor critici, asigurând funcționarea stabilă și eficientă în diverse domenii, precum procesarea chimică, producția alimentară, tratarea apei și generarea de energie. Obiectivul principal al controlului proceselor este menținerea variabilelor sistemului – precum presiunea, temperatura sau nivelul lichidelor – în limitele specificate, garantând astfel o operare sigură, productivă și conformă cu standardele de calitate. Evoluția rapidă a industriei moderne a generat o cerere tot mai mare pentru metode de control precise și adaptabile, favorizând implementarea unor strategii avansate de control. Sistemele de control joacă un rol esențial în sectoarele industriale în care menținerea unui parametru critic este vitală pentru asigurarea siguranței, eficienței și calității proceselor [1].

Numeroase lucrări științifice, precum [2], [3] și [4], abordează în detaliu problematica controlului temperaturii apei, oferind soluții teoretice și aplicative pentru procesele complexe.

În lucrarea [2], autorii propun și analizează o metodă precisă pentru controlul temperaturii unui flux de fluid cu debit ridicat, atât din punct de vedere teoretic, cât și experimental.

#### *Metodologie:*

Autorii propun un model teoretic care descrie modul în care sistemul reacționează la variațiile de temperatură ale fluidului în timp. Acest model permite o înțelegere detaliată a comportamentului termic al sistemului și oferă fundamentele necesare pentru proiectarea eficientă a acestuia. Modelul teoretic include criterii precise pentru ajustarea performanței, permițând optimizarea răspunsului sistemului în funcție de cerințele aplicației.

Un aspect esențial al modelului constă în faptul că atenuarea maximă a variațiilor de temperatură poate fi reglată prin modificarea rezistenței termice dintre fluxul de fluid și mediul de atenuare. În plus, gama de frecvențe care pot fi atenuate eficient depinde de produsul dintre rezistența termică și capacitatea termică a mediului, ceea ce oferă un instrument suplimentar de calibrare a performanței.

#### *Rezultate:*

Modelul propus conduce la o atenuare a variațiilor de temperatură care depinde de frecvență, cu performanțe superioare celor obținute prin metode clasice, cum ar fi amestecarea perfectă a fluidului. S-a demonstrat că atenuarea poate fi cu câteva ordine de mărime mai mare în comparație cu soluțiile convenționale. Pentru componentele spectrale ale temperaturii cu perioade mai mici decât timpul de echilibru termic al dispozitivului, atenuarea obținută depășește chiar și trei ordine de mărime. Aceste rezultate sunt în deplin acord cu predicțiile modelului teoretic, validând astfel corectitudinea și utilitatea acestuia.

#### *Metodologie:*

În lucrarea [3], este analizată o strategie de control a temperaturii apei dintr-un sistem de tip baie de apă. Modelarea procesului se bazează pe ecuații diferențiale care descriu disiparea căldurii și evoluția temperaturii în timp. Pentru a înțelege mai bine comportamentul sistemului, au fost realizate simulări în MATLAB, acestea permițând evaluarea distribuției spațiale și temporale a temperaturii, precum și pierderile de căldură.

Pentru controlul temperaturii s-a utilizat un regulator fuzzy PID, care îmbină principiile controlului clasic cu inteligența artificială. Acesta ajustează automat parametrii de control în funcție de erorile prezente și de dinamica sistemului, asigurând un răspuns rapid și stabil la perturbații.

#### *Rezultate:*

Rezultatele obținute arată că modelul este capabil să mențină temperatura apei constantă, cu deviații minime față de valoarea dorită. Designul optimizat al băii de apă contribuie la reducerea pierderilor termice, crescând astfel eficiența sistemului. De asemenea, regulatorul fuzzy PID a demonstrat

o capacitate ridicată de a menține stabilitatea temperaturii în timp, reducând semnificativ fluctuațiile nedorite.

#### *Metodologie:*

În lucrarea [4] este prezentată o strategie avansată de control al temperaturii apei la ieșire, utilizând un sistem de control combinat PID și feedforward. Sistemul este modelat matematic folosind legea conservării energiei, conducând la obținerea unor modele de transfer pentru rezervorul de apă și elementul de încălzire. Controlerul PID și sistemul feedforward sunt reprezentate prin funcții de transfer, fiind astfel integrate în structura generală a sistemului.

Pentru validarea performanței sistemului s-au realizat simulări în mediul Simulink. Parametrii controlerului PID ( $K_p = 2$ ,  $K_i = 0$ ,  $K_d = 10$ ) au fost ajustați utilizând metode clasice de reglaj. Simulările au permis evaluarea comportamentului sistemului în diferite condiții de funcționare și în prezența perturbațiilor.

#### *Rezultate:*

Simulările au demonstrat că sistemul PID controlează eficient temperatura apei prin ajustarea puterii încălzitorului, răspunzând prompt la abaterile de la valoarea de referință. Introducerea componentei feedforward contribuie la compensarea efectelor fluctuațiilor temperaturii de intrare, rezultând într-un sistem mai robust și cu răspuns îmbunătățit. Performanțele obținute confirmă validitatea și eficiența metodei propuse.

Controlul temperaturii reprezintă un aspect esențial în numeroase aplicații industriale și domestice, fiind direct legat de eficiența, siguranța și calitatea proceselor. În mod special, menținerea unei temperaturi constante în bazine de apă este întâlnită frecvent în domenii precum tratamentele medicale, procesele chimice sau prepararea alimentelor. În acest context, selectarea unei strategii de control eficiente este crucială. Dintre multiplele metode disponibile, regulatoarele PID și MPC sunt printre cele mai utilizate și studiate datorită aplicabilității largi și performanțelor demonstrate.

Alegerea temei „Controlul temperaturii unui sistem de tip baie de apă” este motivată de interesul de a înțelege și optimiza comportamentul dinamic al unui astfel de sistem. Lucrarea propune o analiză comparativă riguroasă între o metodă clasică (PID), recunoscută pentru simplitate și robustețe, și o metodă modernă (MPC), apreciată pentru capacitatea de anticipare și gestionare a constrângerilor.

Scopul principal al lucrării este analiza și compararea performanțelor celor două tipuri de regulatoare – PID și MPC – în contextul unui sistem de încălzire a apei. Obiectivele specifice includ:

- Modelarea matematică a sistemului de tip baie de apă;
- Implementarea și simularea regulatorului PID, utilizând mai multe metode de reglare (Lambda, Haalman, criterii integrale, feedforward, predictor Smith);
- Implementarea controlului predictiv bazat pe model (MPC), utilizând un model liniar discret în spațiul stărilor și rezolvarea unei probleme de optimizare pătratică (QP) la fiecare pas de timp.
- Compararea performanțelor între cele două strategii în ceea ce privește: timpul de răspuns, stabilitatea, robustețea și precizia controlului;
- Formularea unor concluzii privind aplicabilitatea fiecărei metode în funcție de cerințele sistemului.

Lucrarea este structurată în cinci capitole principale: capitolul 1, "Introducere" prezintă contextul, motivația și obiectivele cercetării, precum și relevanța temei. Capitolul "Regulatorul PID" oferă o prezentare generală a controlului PID, descrie metodele de reglare utilizate și avantajele fiecărei abordări; capitolul 3, „Regulatorul MPC”, introduce conceptele fundamentale ale controlului predictiv utilizat în această lucrare. Regulatorul este implementat cu ajutorul toolbox-ului MPC din MATLAB, folosind un model liniar discret și un algoritm de optimizare pătratică (QP) pentru calculul comenzilor optime în timp real.. Capitolul 4, "Prezentarea implementării soluțiilor" descrie implementarea practică a sistemului, modelarea sa matematică, simulările efectuate și analiza comparativă a rezultatelor. Capitolul 5 sintetizează concluziile obținute în urma comparației și evidențiază recomandările privind utilizarea celor două metode de control.

## Capitolul 2. Regulatorul PID

### 2.1. Prezentare generală a controlului proporțional-integral-derivativ (PID)

**Controlul Proporțional-Integral-Derivativ (PID)** este, fără îndoială, una dintre cele mai utilizate strategii de control în prezent. Se estimează că peste 90% dintre buclele de control utilizează regulatorii PID, adesea cu câștigul derivativ setat la zero, ceea ce echivalează cu utilizarea unui regulator PI. În ultimele decenii, un număr semnificativ de cercetări academice și industriale s-au concentrat pe îmbunătățirea performanței PID, în special în domeniile reglării parametrilor, tehnicilor de identificare și a metodelor de adaptare. În acest context, este important să analizăm stadiul actual al controlului PID și noile direcții de dezvoltare în acest domeniu.

Regulatorul PID (Proportional-Integral-Derivative) este descris de ecuația:

$$u(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(\tau) d\tau + K_d \cdot \frac{de(t)}{dt} \quad (2.1)$$

unde  $u(t)$  este semnalul de control (ieșirea regulatorului),  $e(t)$  este eroarea, definită ca  $e(t) = r(t) - y(t)$ , unde  $r(t)$  este valoarea de referință și  $y(t)$  este valoarea măsurată,  $K_p$  este factorul de proporționalitate,  $K_i$  este timpul de integrare,  $K_d$  este timpul de derivare,  $\int_0^t e(\tau) d\tau$  este integrala erorii (acumularea erorilor),  $\frac{de(t)}{dt}$  este derivata erorii (rata de schimbare a erorii).

### 2.2. Descrierea termenilor din regulatorul PID

**Termenul proporțional** ( $K_p \cdot e(t)$ ) al unui regulator PID este direct dependent de eroarea curentă dintre valoarea de referință și cea măsurată. Acesta acționează imediat și generează un semnal de control proporțional cu amplitudinea erorii. Cu cât eroarea este mai mare, cu atât comanda aplicată asupra sistemului va fi mai intensă; însă, un câștig proporțional  $K_p$  prea ridicat poate conduce la oscilații semnificative și instabilitate în răspunsul sistemului.

**Termenul integral** ( $K_i \cdot \int_0^t e(\tau) d\tau$ ) din structura regulatorului PID are rolul de a corecta eroarea acumulată în timp, contribuind semnificativ la eliminarea erorilor de regim staționar care nu pot fi compensate doar prin componenta proporțională. Prin integrarea erorii, regulatorul adaugă o acțiune corectivă continuă, care crește până când abaterea dispăre complet. Totuși, un câștig integral  $K_i$  ales prea mare poate duce la supracompensare și poate compromite stabilitatea generală a sistemului.

**Termenul derivativ** ( $K_d \cdot \frac{de(t)}{dt}$ ) este sensibil la variațiile rapide ale erorii și are rolul de a anticipa tendințele acesteia pe baza ratei de schimbare. Prin această acțiune, contribuie la amortizarea oscilațiilor și la îmbunătățirea stabilității sistemului, funcționând ca un mecanism de control predictiv. Totuși, un câștig derivativ  $K_d$  prea ridicat poate amplifica zgomotul de înaltă frecvență prezent în semnalele de măsurare sau poate introduce întârzieri nedorite în răspunsul sistemului.

În mod ideal, ajustarea corectă a parametrilor  $K_p$ ,  $K_i$  și  $K_d$  duce la un răspuns echilibrat: un timp de stabilizare rapid, fără oscilații mari și fără eroare de stare staționară [5].

Cele trei componente ale unui regulator PID răspund unor cerințe fundamentale ale majorității problemelor de control. Componenta integrală (**I**) elimină eroarea de regim staționar atunci când se urmărește un punct de referință constant, un rezultat explicat prin principiul modelului intern și demonstrat prin teorema valorii finale. Controlul integral permite, de asemenea, respingerea completă a perturbărilor constante. Deși acest tip de control filtrează zgomotul de frecvență înaltă provenit de la senzori, răspunde mai lent la erorile curente. În schimb, componenta proporțională (**P**) răspunde imediat la eroarea curentă, dar de obicei nu poate atinge o precizie acceptabilă a valorii de referință fără un câștig excesiv de mare.

În cazul sistemelor cu întârziere semnificativă, acțiunile de control anterioare sunt slab reprezentate în eroarea curentă, ceea ce poate conduce la erori tranzitorii mari atunci când se utilizează

control PI. Componenta derivativă (**D**) ajută la combaterea acestei probleme, prin prezicerea erorii viitoare pe baza istoricului acțiunilor anterioare. Cu toate acestea, componenta derivativă amplifică zgomotul de frecvență înaltă al senzorilor, iar pentru a reduce acest efect, este necesar un filtru al semnalului diferențiat. Acest tip de regulator PID, cu filtru derivativ, este denumit adesea regulator PIDF. Deși cele trei componente PID sunt suficiente pentru a controla cu succes multe tipuri de sisteme, numărul lor mic permite reglajul manual al parametrilor, iar acest lucru face ca PID să fie compatibil atât cu controlul adaptiv direct, cât și cu reglajul automat prin euristici.

Răspunsul regulatorului PID la impuls poate fi considerat un vector de bază pentru acțiunea globală de control. Combinând liniar aceste răspunsuri, se poate construi funcția de control globală. Provocarea în proiectarea și reglajul unui regulator PID constă în optimizarea performanței acestuia în subspațiul definit de aceste funcții de bază. În acest context, funcțiile alternative de bază pot fi utilizate pentru a obține aceleași funcționalități fundamentale ca PID, precum răspunsul rapid, modelul intern și predicția erorii.

Datorită simplității acestor componente, principiile de bază ale regulatorului sunt accesibile și utilizatorilor neexperimentați. Adaptarea, reglajul automat și programarea câștigului pot fi ușor integrate în controlul PID, iar specialiștii preferă să îmbunătățească performanța unei soluții deja cunoscute, în defavoarea unei soluții noi, nevalidate. Deși alternativele de control digital nu au înlocuit popularitatea PID, există excepții, cum ar fi controlul predictiv bazat pe model, care utilizează capacitățile de calcul ale procesorului pentru a calcula legea de control în timp real. Totuși, PID rămâne esențial pentru gestionarea buclelor de control de nivel inferior [6].

### 2.3. Metode de acordare a parametrilor PID

Pentru ca un regulator PID să funcționeze eficient, este esențial ca parametrii săi  $K_p$ ,  $K_i$ ,  $K_d$  să fie aleși corespunzător, în funcție de dinamica procesului controlat. Reglarea acestor parametri reprezintă o etapă crucială în proiectarea sistemului de control, influențând direct stabilitatea, timpul de răspuns și precizia sistemului. De-a lungul timpului, au fost dezvoltate mai multe metode de reglaj, atât empirice, cât și analitice, fiecare cu avantaje și limitări specifice. În continuare, vor fi prezentate principalele tehnici de reglare a parametrilor PID, utilizate în practică pentru optimizarea performanței sistemului.

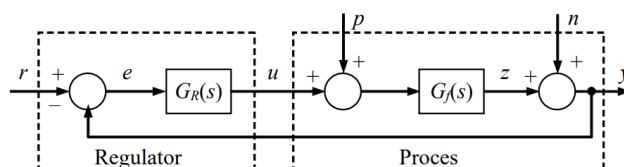


Figura 2.1. Structura de reglare convențională

#### 2.3.1. Metoda alocării poli-zeroouri

**Metoda alocării poli-zeroouri** reprezintă o tehnică de proiectare a reglatoarelor care are ca obiectiv principal obținerea unei funcții de transfer în circuit închis ce satisface un set prestabilit de cerințe de performanță. Această metodă se bazează pe plasarea deliberată a polilor și, opțional, a zerourilor funcției de transfer în buclă închisă, astfel încât sistemul de reglare automat să prezinte comportamente dinamice dorite, precum stabilitate, timp de răspuns scurt, suprareglare controlată sau eroare de regim staționar redusă.

Procesul de proiectare începe prin definirea performanțelor dorite – de exemplu, timpul de stabilizare, lățimea de bandă, eroarea la poziție sau eroarea la viteză. Pe baza acestor specificații, se determină o repartitie a polilor (și eventual a zerourilor) funcției de transfer în circuit închis. Ulterior, regulatorul este proiectat astfel încât să realizeze această repartitie, ceea ce se poate face printr-o combinație de metode analitice și numerice.

Pentru sistemele de ordinul I, performanțele dinamice sunt influențate în principal de poziția polului real negativ al funcției de transfer. Acest pol determină viteza de răspuns a sistemului, caracterizată de o constantă de timp asociată. Un pol mai apropiat de origine (în planul complex) implică un răspuns lent, în timp ce un pol situat mai departe accelerează tranziția către regimul staționar.

În cazul proiectării prin metoda poli-zero-uri, introducerea unui zero suplimentar poate îmbunătăți rapiditatea răspunsului prin compensarea parțială a efectului polului, însă poate conduce și la o creștere a suprareglării. De asemenea, adăugarea unui pol suplimentar — chiar într-un sistem de ordin inițial redus — tinde să degradeze performanțele de regim tranzitoriu, crescând timpul de răspuns și reducând stabilitatea, dar poate deveni necesară pentru asigurarea realizabilității fizice a regulatorului.

În cadrul metodei, procesul de proiectare debutează prin specificarea unor performanțe de comportare, precum suprareglarea relativă maxim admisibilă ( $\sigma$ ) și timpul de reglare ( $t_t$ ). Pe baza acestor cerințe, se determină parametrii caracteristici ai unui sistem de ordinul al doilea: coeficientul de amortizare  $\zeta$  și pulsația naturală  $\omega_n$ . Acestea sunt calculate utilizând relațiile:

$$\zeta = -\frac{\ln(\sigma)}{\sqrt{\pi^2 + \ln^2(\sigma)}} \quad (2.2)$$

Coeficientul de amortizare  $\zeta$ , determinat în funcție de suprareglarea relativă  $\sigma$ . Această relație este valabilă pentru sisteme de ordinul al doilea subamortizate ( $0 < \zeta < 1$ ).

$$\omega_n = \frac{4}{\zeta \cdot t_t} \quad (2.3)$$

Pulsația naturală  $\omega_n$ , calculată pe baza timpului de reglare  $t_t$  și a coeficientului de amortizare  $\zeta$ . Este utilizată pentru poziționarea polilor dominanți în planul complex.

Cu aceste valori, se definește o funcție de transfer dorită pentru sistemul în buclă închisă (de tip ordin II cu un pol în origine):

$$G_d(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s} \quad (2.4)$$

Ecuatia (2.4) reprezintă o structură de ordin al doilea cu un pol suplimentar în origine, care asigură eliminarea erorii statice la o comandă de treaptă. Alegerea acestei structuri reflectă intenția de a obține un sistem capabil să răspundă rapid, cu o suprareglare controlată și o eroare de regim staționar nulă.

Următorul pas în proiectarea regulatorului constă în determinarea funcției de transfer a acestuia. Având cunoscută funcția de transfer a procesului,  $G_m(s)$ , regulatorul se deduce conform relației:

$$G_r(s) = \frac{G_d(s)}{G_m(s)} \quad (2.5)$$

Această formulare decurge din condiția fundamentală a funcției de transfer în buclă închisă:

$$T(s) = \frac{G_r(s)G_m(s)}{1 + G_r(s)G_m(s)} = G_d(s) \quad (2.6)$$

În urma acestei operații, rezultă o funcție de transfer a regulatorului ce poate include termeni proporționali, integrali și derivați, în funcție de dinamica procesului și de ordinul acestuia. Forma obținută poate fi ulterior simplificată și adaptată în vederea implementării fizice, asigurându-se totodată respectarea condiției de realizabilitate fizică a regulatorului (exces pozitiv de poli față de zero-uri)[7].

### 2.3.2. Metoda Haalman

**Metoda Haalman** este o tehnică de proiectare analitică utilizată pentru acordarea reguletoarelor PID, având la bază modelarea simplificată a procesului ca un sistem de ordinul I cu timp mort.



Această metodă este aplicabilă în special în cazul proceselor lente, frecvent întâlnite în industrie, pentru care performanțele dinamice pot fi satisfăcător aproximative printr-un model de tip:

$$G_f(s) = \frac{K_f e^{-Ls}}{T_f s + 1} \quad (2.7)$$

unde:

- $K_f$  este câștigul procesului,
- $T_f$  este constanta de timp,
- $L$  este întârzierea pură (timpul mort).

Obiectivul metodei este de a obține o funcție de transfer în buclă deschisă astfel încât sistemul de reglare să atingă o performanță impusă, în mod uzual un factor de amplificare în frecvență  $M_s$  de aproximativ 1.9, corespunzător unei amortizări de aproximativ 25%.

Pentru un astfel de model, funcția de transfer în buclă deschisă este aleasă astfel:

$$G_d(s) = \frac{2e^{-Ls}}{3Ls} \quad (2.8)$$

iar regulatorul este obținut prin raportarea funcției dorite la modelul procesului:

$$G_R(s) = \frac{G_d(s)}{G_f(s)} = \frac{2}{3K_f} \cdot \frac{T_f s + 1}{T_f s} \quad (2.9)$$

Rezultă, astfel, un regulator de tip PI cu următorii parametri:

$$K_p = \frac{2T_f}{3K_f L}, \quad T_i = T_f \quad (2.10)$$

Acești parametri oferă un compromis rezonabil între rapiditatea răspunsului și stabilitate, fără a necesita o ajustare complexă. De asemenea, implementarea este simplă, ceea ce face metoda Haalman potrivită pentru sisteme cu resurse limitate sau în aplicații în care se dorește un reglaj inițial robust.

În cazul în care procesul este modelat printr-un sistem de ordinul al doilea cu timp mort, se poate aplica o versiune extinsă a metodei Haalman, ce conduce la un regulator PID. În acest caz, se pleacă de la o funcție de transfer de forma:

$$G_f(s) = \frac{K_f e^{-Ls}}{(T_1 s + 1)(T_2 s + 1)} \quad (2.11)$$

iar regulatorul proiectat va avea parametrii:

$$K_p = \frac{2T_1 T_2}{3K_f L}, \quad T_i = T_1 + T_2, \quad T_d = \frac{T_1 T_2}{T_1 + T_2} \quad (2.12)$$

Această extensie menține simplitatea metodei inițiale, dar permite o adaptare mai precisă la dinamica reală a procesului, crescând precizia controlului în regim tranzitoriu.

Metoda Haalman, prin simplitatea formulelor și aplicabilitatea sa largă, rămâne un punct de plecare valoros în proiectarea reglatoarelor PI și PID, mai ales în fazele incipiente de dezvoltare a unui sistem de control industrial[7].

### 2.3.3. Metoda Lambda

**Metoda Lambda**, cunoscută și sub denumirea de reglaj IMC (Internal Model Control), este o tehnică de acordare analitică a reglatoarelor, utilizată frecvent în controlul proceselor lente, caracterizate de întârziere semnificativă și comportament de ordinul I cu timp mort. Această metodă are la bază ideea formulării explicite a unei funcții de transfer dorite în buclă închisă, parametrizată printr-o constantă de timp de reglare  $\lambda$ , care permite ajustarea directă a performanței dinamice a sistemului controlat.

Modelul procesului este considerat de forma:

$$G_f(s) = \frac{K_f e^{-Ls}}{T_f s + 1} \quad (2.13)$$

unde:

- $K_f$  este câștigul procesului,
- $T_f$  este constanta de timp,
- $L$  este întârzierea pură (timpul mort).

Funcția de transfer dorită în buclă închisă este aleasă astfel:

$$G_0(s) = \frac{1}{\lambda s + 1} \quad (2.14)$$

unde:

- $\lambda$  este un parametru de proiectare ce influențează direct rapiditatea și robustețea sistemului,
- $\lambda$  mic conduce la un răspuns mai rapid, dar poate compromite robustețea,
- $\lambda$  mare atenuează oscilațiile, crescând stabilitatea.

Regulatorul este dedus din relația:

$$G_r(s) = \frac{G_0(s)}{1 - G_0(s)} \cdot \frac{1}{G_f(s)} = \frac{1}{K_f} \cdot \frac{T_f s + 1}{\lambda s + 1} \cdot e^{Ls} \quad (2.15)$$

Pentru implementare practică, termenul de întârziere  $e^{Ls}$  este în mod uzual aproximat printr-o dezvoltare Pade de ordinul I, conducând la un regulator de tip PI:

$$G_r(s) \approx \frac{K_p(1 + T_i s)}{T_i s} \quad (2.16)$$

unde:

$$K_p = \frac{T_f}{K_f(\lambda + L)}, \quad T_i = T_f \quad (2.17)$$

Acești parametri definesc un regulator PI ce asigură o reglare stabilă și robustă, ușor de implementat și ajustat. Prin alegerea corespunzătoare a lui  $\lambda$  se poate obține un compromis favorabil între performanță și robustețe, fără a necesita modele de ordin superior sau optimizări numerice complexe.

Astfel, regulatorul proiectat va avea o structură PI robustă, adaptabilă proceselor lente, permițând controlul eficient al temperaturii în aplicații industriale sensibile, cum este cazul tratamentelor termice sau sistemelor medicale[7].

### 2.3.4. Metoda Zhuang Atherton

**Metoda Zhuang–Atherton** reprezintă o tehnică modernă de acordare a reguletoarelor PID, care are la bază principiile optimizării pe criterii integrale de performanță. Scopul metodei este determinarea optimă a parametrilor regulatorului astfel încât sistemul controlat să satisfacă un anumit criteriu de performanță global, exprimat sub forma unei funcții cost definite pe întreaga durată a regimului tranzitoriu.

Metoda se bazează pe minimizarea uneia dintre următoarele funcționale integrale:

- ISE – Integral of Squared Error:

$$J_{ISE} = \int_0^{\infty} e^2(t) dt \quad (2.18)$$

- ISTE – Integral of Squared Time-weighted Error:

$$J_{ISTE} = \int_0^{\infty} t \cdot e^2(t) dt \quad (2.19)$$

- IST<sup>2</sup>E – Integral of Squared Time-squared-weighted Error:

$$J_{IST^2E} = \int_0^{\infty} t^2 \cdot e^2(t) dt \quad (2.20)$$

Aceste criterii cuantifică eroarea în timp, penalizând deviațiile în funcție de momentul apariției. De exemplu, IST<sup>2</sup>E penalizează mai sever erorile tardive decât cele inițiale, oferind un control mai ferm asupra stabilizării sistemului. Modelul procesului considerat este de tip FOPDT (First Order Plus Dead Time), descris prin trei parametri:

- $k_f$  - câștigul static,
- $T_f$  - constantă de timp,
- $L$  - întârziere (timpul mort).

Acordarea PID se face pe baza unei expresii generale, de forma:

$$K_p = \frac{a_1}{k_f} \left( \frac{L}{T_f} \right)^{b_1} \quad (2.21)$$

$$T_i = \frac{T_f}{a_2 + b_2 \cdot \frac{L}{T_f}} \quad (2.22)$$

$$T_d = a_3 \cdot T_f \left( \frac{L}{T_f} \right)^{b_3} \quad (2.23)$$

unde coeficienții  $a_i$  și  $b_i$  sunt extrași din tabele specifice fiecărui criteriu (ISE, ISTE, IST<sup>2</sup>E) și din domeniul valorilor raportului  $\frac{L}{T_f}$  [7]. Alegerea valorilor coeficienților depinde de intervalul în care se situează acest raport, tipic împărțit în două categorii:

- $0.1 \leq \frac{L}{T_f} \leq 1$ , pentru procese rapide,
- $1.1 \leq \frac{L}{T_f} \leq 2$ , pentru procese lente

### 2.3.5. Reglarea de tip Feedforward

**Reglarea de tip feedforward** reprezintă o extensie a strategiei de reglare clasică în buclă închisă (feedback), având scopul de a compensa în mod direct efectele unor perturbații cunoscute sau ale referinței asupra ieșirii sistemului. Spre deosebire de controlul feedback, care reacționează doar în urma apariției unei erori, structura feedforward acționează *proactiv*, aplicând o comandă calculată pe baza unui model matematic al procesului.

Această metodă presupune existența unui model suficient de precis al procesului și a unei cunoașteri prealabile a semnalului perturbator sau a referinței. Implementarea unei astfel de structuri conduce, în mod ideal, la eliminarea completă a erorii de regim tranzitoriu cauzate de perturbații lente sau schimbări ale valorii de referință.

Modelul procesului reglat este adesea exprimat printr-o funcție de transfer de forma:

$$G_p(s) = \frac{K_f}{T_f s + 1} \quad (2.24)$$

unde  $K_f$  reprezintă câștigul static al procesului, iar  $T_f$  este constanta de timp asociată procesului.

Pentru a compensa anticipativ comportamentul acestuia, se proiectează o funcție de transfer a regulatorului feedforward de forma:

$$G_{ff}(s) = -\frac{K_p(1 + T_i s)}{K_f(1 + T_p s)} \quad (2.25)$$

în care:

- $K_p$  este câștigul componentei proporționale a canalului anticipativ,
- $T_i$  este timpul de integrare specific unei componente I implicate,
- $T_p$  este constanta de timp a unei structuri de compensare a procesului.

Această structură reflectă o relație de inversare aproximativă a comportamentului procesului și este utilizată frecvent pentru a reduce erorile apărute în regim tranzitoriu. În practică, valorile parametrilor  $K_p$  și  $T_p$  se aleg în funcție de dinamica procesului și de tipul semnalelor care urmează a fi compensate.

Una dintre cele mai semnificative avantaje ale controlului anticipativ constă în capacitatea acestuia de a reduce considerabil erorile de regim tranzitoriu, prin compensarea directă a efectelor unor perturbații măsurabile sau a modificărilor semnalului de referință. Spre deosebire de controlul de tip feedback, care acționează reactiv, controlul feedforward furnizează o acțiune corectivă prealabilă, bazată pe un model matematic al procesului. Această abordare permite o atenuare timpurie a deviațiilor, conducând la un răspuns sistemic mai rapid și mai precis. De asemenea, integrarea unei componente anticipative contribuie la diminuarea solicitărilor aplicate asupra regulatorului de feedback, permițând acestuia să opereze într-un regim mai stabil și mai puțin exigent din punct de vedere dinamic.

Cu toate acestea, metoda prezintă și anumite limitări importante. Performanța controlului anticipativ este strâns dependentă de acuratețea modelului procesului utilizat în proiectare. În prezența unor incertitudini structurale sau a variației parametrilor în timp, eficiența acțiunii anticipative poate fi diminuată sau chiar compromisă. În plus, aplicabilitatea sa este restricționată la situațiile în care perturbațiile pot fi măsurate în mod direct și sunt disponibile în timp real. În lipsa acestor condiții, contribuția feedforward-ului devine marginală, iar sistemul trebuie să se bazeze exclusiv pe corecțiile generate de bucla de feedback. Din acest motiv, în implementările industriale, această strategie este rar utilizată în mod izolat, fiind aproape întotdeauna asociată unui control feedback clasic, cum ar fi un regulator PID, pentru a asigura robustețea și precizia necesare funcționării pe termen lung [7].

### 2.3.6. Reglare cu predictor Smith

**Predictorul Smith** reprezintă o metodă avansată de compensare a întârzierilor pure din sisteme de control, fiind utilizat în special în procese unde timpul mort este semnificativ comparativ cu dinamica internă a sistemului. Structura acestuia este mai complexă decât cea a unui regulator PID obișnuit, necesitând un model explicit al procesului pentru a anticipa comportamentul viitor al sistemului.

Conceptul fundamental al predictorului Smith se bazează pe utilizarea unui model intern al procesului pentru a estima ieșirea curentă a sistemului în absența întârzierii. Această estimare este apoi comparată cu ieșirea reală (care include întârzierea), permițând eliminarea efectului timpului mort din bucla de reglare. Prin urmare, regulatorul PID operează ca și cum întârzierea nu ar exista, îmbunătățind semnificativ stabilitatea și viteza de răspuns a sistemului.

Structura generală a sistemului cu predictor Smith implică două modele: unul fără întârziere, utilizat pentru control, și unul complet (cu întârziere), utilizat pentru reconstrucția ieșirii reale. Modelul procesului este exprimat ca:

$$G(s) = G_m(s) \cdot e^{-Ls} \quad (2.26)$$

unde:

- $G(s)$  este funcția de transfer a procesului real cu întârziere,
- $G_m(s)$  este modelul procesului fără întârziere,
- $L$  este timpul mort (întârzierea pură).

Funcția de transfer în buclă închisă a sistemului cu predictor Smith devine:

$$T(s) = \frac{G_r(s)G_m(s)}{1 + G_r(s)G_m(s)} \quad (2.27)$$

unde  $G_r(s)$  este regulatorul proiectat pe baza modelului fără întârziere  $G_s(s)$ . Avantajul principal al acestei metode constă în capacitatea de a compensa întârzierile fără a afecta negativ stabilitatea sistemului. Prin înlocuirea în buclă a procesului real cu modelul său predictiv, se elimină efectul negativ al întârzierii asupra controlerului. Astfel, regulatorul poate fi proiectat ca pentru un sistem fără întârziere, conducând la performanțe dinamice superioare.

Totuși, o limitare majoră o constituie dependența critică de acuratețea modelului  $G_m(s)$ . În cazul în care modelul nu reflectă fidel comportamentul procesului real, eficiența predictorului scade, putând apărea instabilități sau deviații mari față de referință. De asemenea, implementarea acestei structuri presupune resurse de calcul suplimentare și un efort semnificativ de identificare a parametrilor modelului, ceea ce o face mai puțin potrivită pentru sisteme cu arhitectură simplă sau în medii cu resurse computaționale limitate [7].

---

## Capitolul 3. Regulatorul MPC

### 3.1. Prezentare generală a controlului predictiv bazat pe model (MPC)

**Controlul predictiv bazat pe model (MPC)** este o metodă avansată de control utilizată pentru optimizarea comportamentului unui sistem pe un orizont de timp viitor. Această tehnică este folosită pe scară largă în industrie datorită abilității sale de a gestiona sisteme complexe și multivariabile, având în vedere și restricțiile impuse asupra variabilelor de control. MPC a fost dezvoltat pentru prima dată în anii 1970 și aplicat inițial în domenii precum industria chimică și rafinarea, unde procesele complexe necesită un control riguros și flexibil. Ulterior, aplicabilitatea MPC s-a extins și în alte domenii, cum ar fi automatizarea și electronica de putere.

MPC se bazează pe un model matematic care prezice comportamentul viitor al sistemului pe un orizont de predicție, folosind informațiile curente despre starea acestuia. La fiecare pas de eșantionare, controlerul MPC calculează o secvență de acțiuni de control care minimizează diferența față de punctul de referință dorit, respectând totodată restricțiile asupra variabilelor de control și stare.

Un aspect distinctiv al MPC este faptul că, deși se calculează o secvență completă de acțiuni de control, doar prima acțiune este aplicată efectiv în sistem. După aplicarea acesteia, modelul este actualizat cu noile informații, iar procesul de optimizare este reluat, asigurând astfel o adaptare dinamică la condițiile reale și schimbătoare ale procesului. Aceasta permite MPC să răspundă rapid la perturbațiile și variațiile condițiilor de operare, fiind extrem de util în aplicațiile industriale. Componentele fundamentale ale MPC-ului sunt reprezentate de:

1. *Modelul sistemului*: are rolul de a descrie dinamica internă a procesului controlat și stă la baza predicției comportamentului acestuia.
2. *Funcția de cost*: Funcția de cost are rolul de a optimiza performanța sistemului pe un orizont finit de predicție, penalizând deviațiile de la traiectoria de referință și variațiile comenzii de control. Această funcție poate fi descrisă de ecuația:

$$J = \sum_{i=0}^{N_p} \|y_{k+i} - r_{k+i}\|_Q^2 + \sum_{i=0}^{N_c-1} \|\Delta u_{k+i}\|_R^2 \quad (3.1)$$

unde  $N_p$  este orizontul de predicție,  $N_c$  orizontul de control cu  $N_c \leq N_p$ , iar  $Q$  și  $R$  matricele de ponderare pentru eroarea de urmărire și, respectiv, acțiunea de control.

3. *Orizontul de predicție ( $N_p$ )*: reprezintă numărul de pași în viitor pentru care sistemul își prezice comportamentul.
4. *Orizontul de control ( $N_c$ )*: determină numărul de pași pentru care sunt calculate acțiuni de control.
5. *Restricții*: impuse asupra stărilor și acțiunilor de control:

$$x_{min} \leq x_k \leq x_{max}, \quad u_{min} \leq u_k \leq u_{max} \quad (3.2)$$

6. *Algoritmul de optimizare*: rezolvă funcția de cost pentru a genera comenzi optime respectând restricțiile.

MPC integrează restricțiile direct în proiectarea controlului, ceea ce permite gestionarea eficientă a proceselor industriale complexe, precum centralele electrice sau rafinăriile. În plus, MPC

este robust față de perturbațiile și incertitudinile modelului de proces, fiind capabil să anticipateze comportamentele viitoare și să ajusteze proactiv acțiunile de control pentru a minimiza impactul acestora asupra sistemului.

În ciuda avantajelor sale, MPC poate întâmpina provocări în aplicarea practică, mai ales în sisteme cu dinamici rapide sau cu un număr mare de variabile de stare și control. În aceste cazuri, cerințele computaționale devin semnificative. Totuși, avansurile tehnologice recente, precum procesoarele de semnal digital (DSP) și FPGA-urile, au făcut posibilă implementarea MPC în timp real, chiar și în aplicațiile care necesită un control extrem de rapid, precum electronica de putere sau automatizarea proceselor industriale.

Eficiența MPC depinde de acuratețea modelului matematic, iar erorile de modelare pot reduce performanța sau pot provoca instabilitate. De asemenea, cerințele computaționale ale MPC, care presupun rezolvarea unei probleme de optimizare la fiecare pas de timp, pot limita aplicabilitatea acestuia în sisteme care necesită răspunsuri foarte rapide sau în cele cu resurse de calcul limitate. În aceste cazuri, metode de control mai simple pot fi preferate [8].

### 3.2. Formularea matematică a problemei QP în MPC

În cadrul acestei lucrări, regulatorul MPC utilizat este cel furnizat de toolbox-ul dedicat din MATLAB. Acesta rezolvă, la fiecare pas de eșantionare, o problemă de optimizare pătratică (QP), având la bază un model liniar al procesului exprimat sub formă de funcție de transfer. Toolbox-ul construiește automat matricele necesare pe baza modelului furnizat și a parametrilor impuși de utilizator (orizonturi, ponderi, constrângeri etc.).

$$\min_{\Delta \mathbf{u}} J(\Delta \mathbf{u}) = \frac{1}{2} \Delta \mathbf{u}^T H \Delta \mathbf{u} + f^T \Delta \mathbf{u} \quad (3.3)$$

unde:

- $\Delta \mathbf{u}$  reprezintă vectorul variațiilor de comandă pe orizontul de control;
- $H$  este matricea Hessian, ce include ponderile asociate deviației față de referință și efortului de control;
- $f$  este vectorul gradientului, calculat la fiecare pas de timp în funcție de eroarea față de referință și de modelul procesului.

Problema este rezolvată sub constrângeri liniare de tipul:

$$A_{ineq} \Delta \mathbf{u} \leq b_{ineq} \quad (3.4)$$

Aceste constrângeri sunt formulate automat în MATLAB, în funcție de:

- **Limite impuse pe comanda absolută** (de exemplu, puterea maximă/minimă disponibilă);
- **Limite pe variația comenzii** între două momente de timp succesive;
- **Limite pe ieșirea procesului** (dacă sunt specificate).

Toolbox-ul MPC permite specificarea acestor limite într-un mod intuitiv prin interfața grafică sau în mod programatic, iar modelul dinamic necesar este derivat dintr-o funcție de transfer discretizată. La fiecare pas de timp, doar prima variație de comandă din soluția optimă este aplicată:

$$u(k) = u(k-1) + \Delta u(k|k) \quad (3.5)$$

Ulterior, sistemul este actualizat, iar problema QP se reformulează pentru pasul următor. Rezolvarea acesteia se face intern prin algoritmi eficienți (precum metode de tip punct-interior sau algoritmi KWIK), fără ca utilizatorul să fie nevoit să gestioneze explicit aceste aspecte.

Această implementare oferă avantajul integrării directe a restricțiilor pe comandă, un aspect dificil de tratat cu alte metode de reglare, precum controlul PID clasic.

### 3.2.1. Algoritmul KWIK – metodă active-set utilizată în MPC din MATLAB

În implementarea regulatorului MPC din MATLAB, problema de programare pătratică (QP) este rezolvată la fiecare pas de timp folosind algoritmi eficienți, printre care și algoritmul KWIK (K-step Workload Invariant Kalman). Acesta aparține clasei metodelor *active-set* și este adaptat pentru aplicații de control în timp real.

Față de metodele generice de optimizare, KWIK profită de structura fixă a problemei MPC – matricea Hessian și matricea de constrângeri rămân neschimbate, iar doar vectorii de stare și referință variază. Acest lucru permite reutilizarea factorizărilor și actualizarea rapidă a soluției.

Problema standard rezolvată este:

Algoritmul KWIK rezolvă iterativ sistemul KKT:

$$\begin{bmatrix} H & A_{\mathcal{A}}^T \\ A_{\mathcal{A}} & 0 \end{bmatrix} \begin{bmatrix} \Delta \mathbf{u} \\ \lambda \end{bmatrix} = \begin{bmatrix} -f \\ b_{\mathcal{A}} \end{bmatrix} \quad (3.6)$$

unde  $A_{\mathcal{A}}$  reprezintă setul activ de constrângeri. La fiecare pas, se verifică multiplii Lagrange pentru a adăuga sau elimina constrângeri, accelerând convergența. MATLAB selectează automat metoda optimă de rezolvare în funcție de dimensiunea problemei și de constrângerile specificate.

Pentru aplicațiile prezentate, algoritmul KWIK oferă un timp de execuție suficient de mic pentru a permite reglarea în timp real, chiar și pentru orizonturi lungi de predicție [9].



## Capitolul 4. Prezentarea implementării soluțiilor

### Modelarea procesului termic:

În această lucrare a fost analizată problema reglării temperaturii apei dintr-o baie destinată bebelușilor [10], folosind o abordare practică bazată pe încălzirea electrică cu o rezistență PTC de 11 kW. Acest tip de rezistență este ales datorită comportamentului de reglare automata și a siguranței crescute în aplicații casnice.

Funcția de transfer utilizată pentru modelare:

$$H(s) = \frac{3,75}{420s + 1} e^{-90s} \quad (4.1)$$

unde:

- $K = 3,75$  [°C/kW] este câștigul static al sistemului;
- $T = 420$  [s] este constanta de timp;
- $\tau = 90$  [s] reprezintă întârzierea pură (timp mort).

Această funcție de transfer a fost adoptată ca model comun pentru compararea performanțelor a două tipuri de regulatori:

1. *Regulatorul PID* – implementat și testat folosind metode clasice de reglaj precum Haalman sau criterii integrale (ISE, ISTE);
2. *Regulatorul MPC (Model Predictive Control)* – configurat cu ajutorul "Model Predictive Control Toolbox" din MATLAB, utilizând direct funcția de transfer, fără conversie în spațiul stărilor.

Prin utilizarea aceluiași model matematic pentru ambele metode de control, compararea performanțelor devine echitabilă. Pentru a evidenția comportamentul sistemului în diferite condiții de operare, au fost efectuate simulări care au inclus trei scenarii principale: răspunsul sistemului la o modificare a valorii de referință, capacitatea de rejectare a perturbațiilor externe și evoluția temperaturii în prezența unei limitări impuse asupra valorii comenzii aplicate.

Rezultatele obținute au permis evaluarea avantajelor oferite de MPC în comparație cu regulatorul PID, în special în scenarii cu constrângeri stricte, datorită capabilității sale de a anticipa evoluția sistemului și de a respecta limitele impuse asupra acțiunii de control.

Graficul obținut pe baza funcției de transfer este:

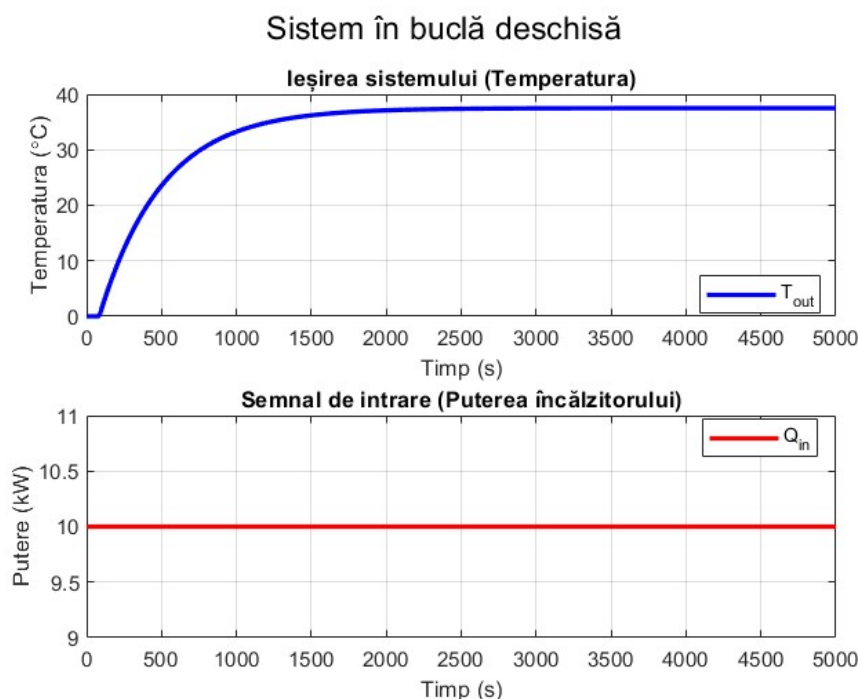


Figura 4.1. Sistem în buclă deschisă

#### 4.1. Analiza comparativă a metodelor PID pe baza simulărilor

##### 4.1.1. Metoda Lambda (albastru închis)

Metoda Lambda oferă un răspuns rapid și bine amortizat, sistemul atingând referința într-un timp scurt fără oscilații semnificative. Regulatorul PID obținut are parametrii:  $K_p = 0.3862$ ,  $K_i = 0.00091954$ . Acest reglaj oferă un compromis excelent între rapiditate și robustețe, fiind potrivit pentru aplicații industriale cu întârziere moderată.

##### 4.1.2. Metoda Haalman (roșu)

Răspunsul sistemului folosind metoda Haalman este mai lent comparativ cu Lambda, dar stabil și lipsit de oscilații mari. Parametrii regulatorului sunt:  $K_p = 0.8296$ ,  $T_i = 0.0019753$ . Această metodă este simplă și practică, fiind adecvată pentru aplicații în care se dorește un reglaj robust inițial fără cerințe stricte de performanță.

##### 4.1.3. Criterii integrale: ISE, ISTE, $IST^2E$

Aceste metode optimizează performanța regulatorului PID prin minimizarea unor funcționale integrale care penalizează diferit eroarea de reglaj.

- **ISE (turcoaz):**  $K_p = 1.1128$ ,  $K_i = 0.0030$ ,  $K_d = 58.1982$ . Oferă un răspuns relativ rapid, cu oscilații moderate.
- **ISTE (galben):**  $K_p = 1.1065$ ,  $K_i = 0.0024$ ,  $K_d = 57.8650$ . Răspuns bine echilibrat, cu supra-reglare redusă.
- **$IST^2E$  (albastru punctat):**  $K_p = 1.0390$ ,  $K_i = 0.0023$ ,  $K_d = 34.8986$ . Cel mai lent, dar cel mai stabil răspuns, fără oscilații notabile.

#### 4.1.4. Predictorul Smith (galben punctat)

Pentru această metodă s-a utilizat un regulator obținut prin alocarea poli-zero, ceea ce a permis compensarea întârzierii sistemului. Regulatorul are funcția de transfer:

$$G_r(s) = \frac{0.002485s + 5.917 \cdot 10^{-6}}{s^2 + 0.006667s} \quad (4.2)$$

Structura completă a sistemului cu Predictorul Smith este ilustrată în figura de mai jos:

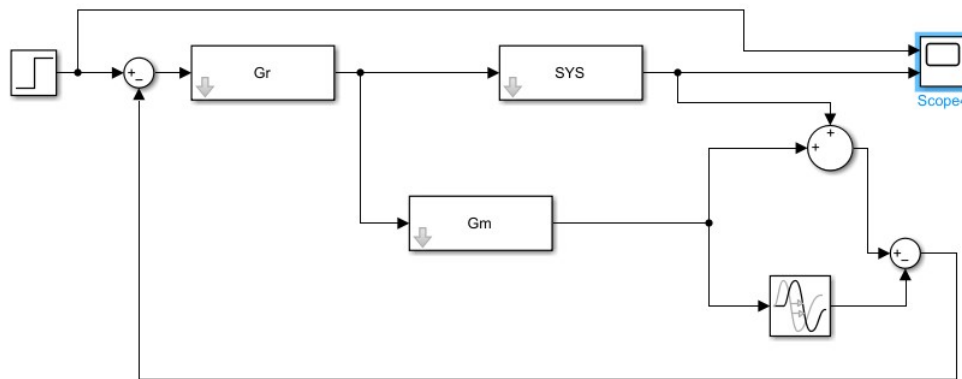


Figura 4.2. Structura bloc pentru control cu Predictorul Smith

Această abordare permite un răspuns rapid, dar este sensibilă la precizia modelului procesului. De asemenea, pentru metodele PID clasice (Haalman, ISE, ISTE, IST<sup>2</sup>E), implementarea a fost combinată cu această structură, întrucât Predictorul Smith contribuie la reducerea efectelor întârzierii pure și minimizează suprareglarea. Integrarea sa în sistem a dus la performanțe superioare în ceea ce privește stabilitatea și rapiditatea răspunsului, fără a compromite robustețea.

#### 4.1.5. Metoda de reglare utilizând structura Feedforward (turcoaz punctat)

În cazul controlului anticipativ (feedforward), s-a utilizat același regulator obținut prin metoda alocării poli-zero, ca la predictorul Smith. Deși introduce comenzi corective în avans, această metodă a condus la o stabilizare mai lentă și o deviere inițială mai mare. Eficiența sa este influențată direct de calitatea modelului procesului și de acuratețea cu care sunt cunoscute perturbațiile. Regulatorul Feedforward are următoarea formă:

$$G_{ff}(s) = \frac{-840s - 2}{1125s + 3.75} \quad (4.3)$$

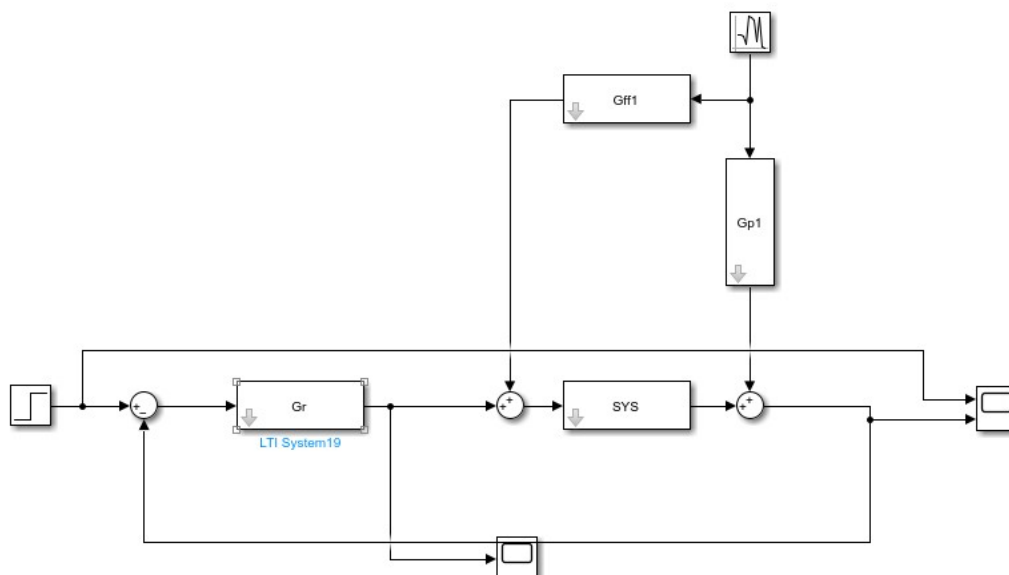


Figura 4.3. Structura bloc pentru control anticipativ (Feedforward)

Pentru a evidenția eficiența acestei structuri de reglaj, sistemul a fost testat atât cu, cât și fără componenta Feedforward, în prezența unui zgomot alb, pentru a putea observa impactul acesteia asupra performanței generale.

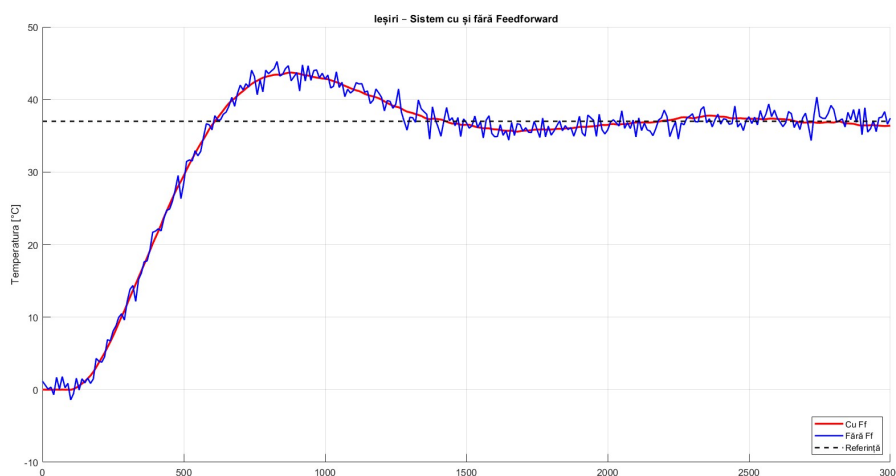


Figura 4.4. Structura bloc pentru control anticipativ (Feedforward)

#### 4.1.6. Combinarea reglatoarelor PID cu structura Predictorului Smith

În această lucrare, toate metodele de reglaj PID (Haalman, ISE, ISTE, IST<sup>2</sup>E și metoda Feed-Forward), cu excepția metodei Lambda, au fost combinate cu structura Predictorului Smith. Alegerea acestei combinații a avut ca scop principal compensarea întârzierii pure semnificative din sistemul analizat, reprezentată de termenul exponențial din funcția de transfer.

Predictorul Smith permite sistemului să reacționeze anticipativ, bazându-se pe un model al procesului fără întârziere, ceea ce duce la:

- *reducerea semnificativă a suprareglării*, mai ales în cazurile în care reglatoarele PID, aplicate individual, tindeau să provoace un răspuns oscilant;

- *îmbunătățirea stabilității sistemului* în faza de reglare, chiar și în prezența unei întârzieri de 90 de secunde;
- *scăderea timpului de stabilizare* față de varianta PID fără compensare;
- păstrarea simplității de implementare a reguletoarelor PID, dar cu un comportament semnificativ îmbunătățit.

Combinția regulator PID + Predictor Smith s-a dovedit eficientă în menținerea unui control robust și stabil, chiar și în condițiile întârzierii inerente procesului termic analizat. Acest lucru a permis obținerea unor performanțe comparabile cu cele ale metodei predictive MPC, dar cu o structură de control mai simplă.

#### 4.1.7. Concluzii ale comparației

În figura 4.5 am reprezentat cele 7 semnale:

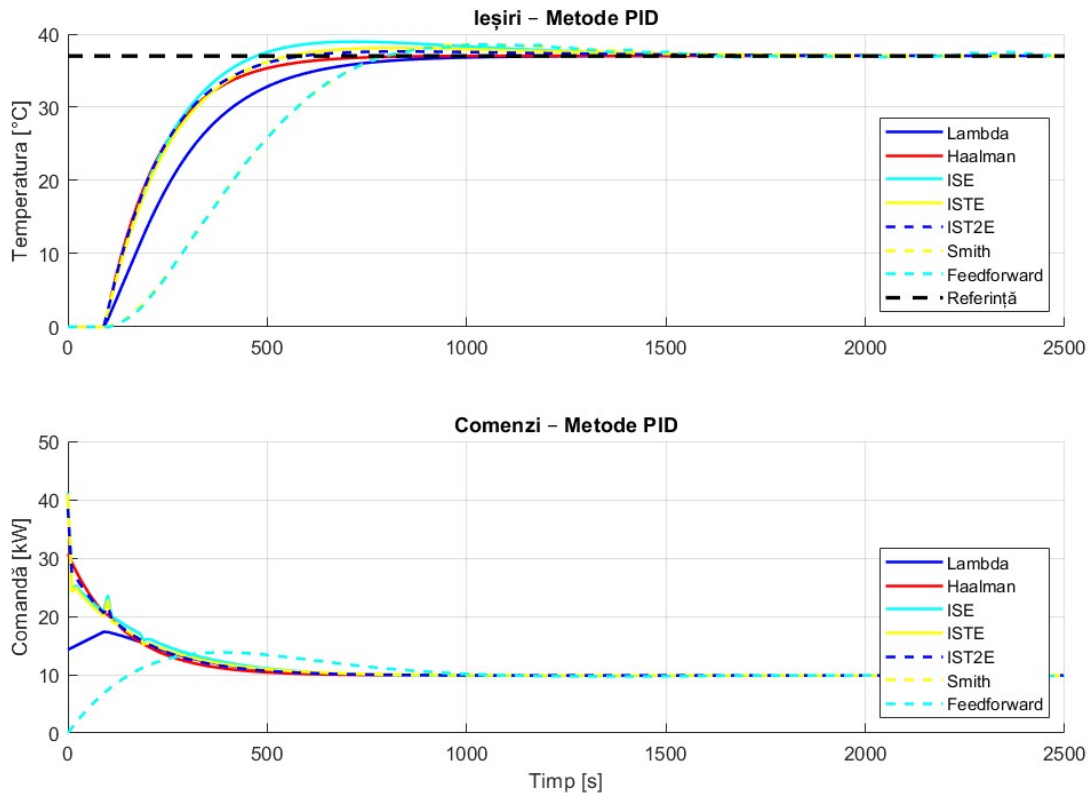


Figura 4.5. Reprezentarea grafică a ieșirilor și a comenzilor metodelor PID

În continuare, în figurile 4.4–4.7 sunt prezentate performanțele celor 7 metode de acordare PID, sub forma valorilor MSE (eroare medie pătratică), a supraraglării, a efortului de control și a duratei regimului tranzitoriu. Aceste rezultate permit o evaluare comparativă clară a eficienței fiecărei metode.

În figura 4.6 sunt prezentate valorile erorii medii pătratice (MSE) pentru fiecare metodă de acordare PID aplicată în controlul temperaturii apei. Acest indicator reflectă cât de bine reușește sistemul să urmărească temperatura de referință pe întreaga durată a simulării. Se observă că metodele Haalman, ISE, ISTE și IST2E au oferit o precizie ridicată, cu erori pătratice mici, în timp ce metodele Smith și Feedforward au avut cele mai slabe performanțe, semnalând abateri mai mari față de temperatura dorită.

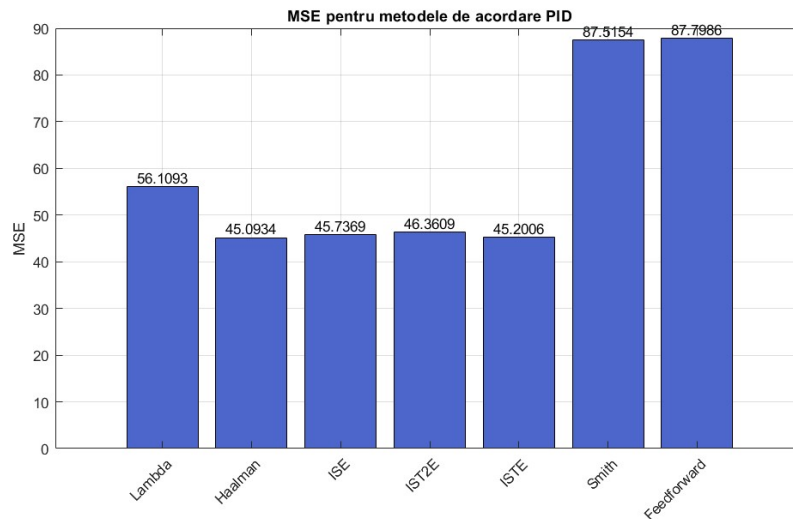


Figura 4.6. Reprezentarea grafică a eroarii medii patratice

Figura 4.7 evidențiază supraraglarea înregistrată de fiecare metodă în timpul reglării temperaturii apei. Supraraglarea indică cât de mult a depășit temperatura valoarea dorită imediat după aplicarea controlului. Metodele Haalman și Lambda se remarcă printr-o supraraglare aproape nulă, ceea ce înseamnă un control termic foarte stabil. În schimb, metoda ISE prezintă cea mai ridicată supraraglare, de peste 5%, ceea ce poate duce la un risc de supraîncălzire temporară a apei.

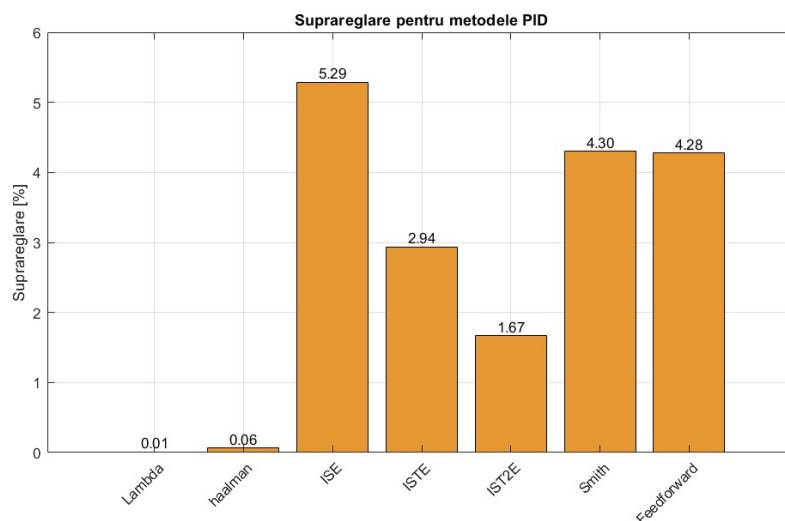


Figura 4.7. Reprezentarea grafică a mediei suprareglărilor

În figura 4.8 este prezentată media comenzii de control (puterea aplicată sistemului de încălzire) pentru fiecare metodă, exprimată în kW. Toate metodele au avut valori apropiate, în jur de 10.5 kW, însă metoda cu predictor Smith a înregistrat cel mai mic efort de control. Acest aspect este important în aplicațiile unde eficiența energetică contează, dar trebuie corelat și cu precizia controlului termic, întrucât metodele cu consum mai mic pot avea performanțe mai slabe.

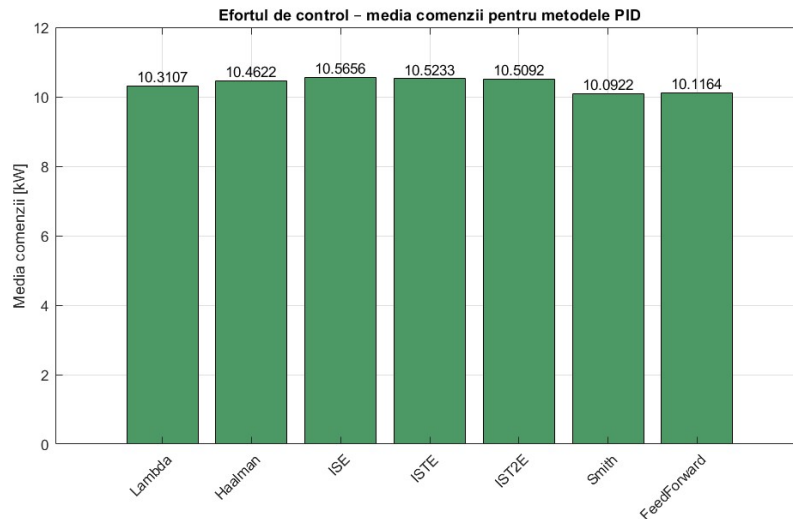


Figura 4.8. Reprezentarea grafică a mediei.

Figura 4.9 compară durata regimului tranzitoriu, adică timpul necesar pentru ca temperatura apei să se stabilizeze în jurul valorii dorite. Metodele Haalman și Lambda au condus la cele mai rapide stabilizări, sub 1400 s, ceea ce este avantajos în aplicații unde răspunsul termic trebuie să fie rapid. La polul opus, metodele ISE și IST2E au avut cele mai lungi perioade tranzitorii, indicând o reglare mai lentă a temperaturii.

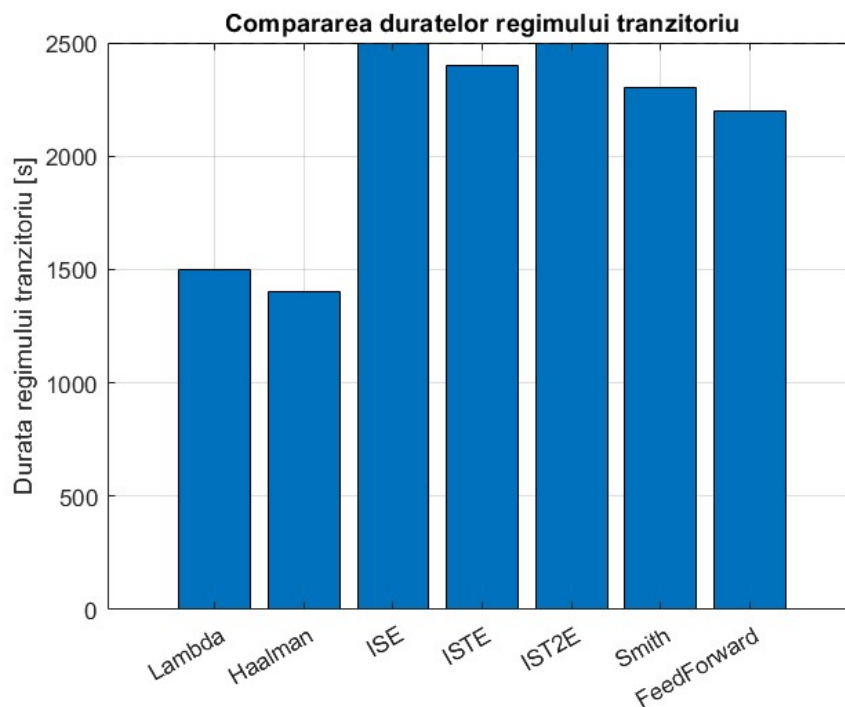


Figura 4.9. Reprezentarea grafică a duratei timpului tranzitoriu

Dintre metodele analizate, cea mai performantă este metoda Haalman. Aceasta are cel mai mic MSE (45.09), indicând o eroare medie pătratică redusă și o urmărire precisă a semnalului de referință. Supraraglarea este aproape nulă (0.06%), ceea ce înseamnă că răspunsul sistemului este stabil și nu depășește valoarea dorită. De asemenea, durata regimului tranzitoriu este mică (aproximativ 1600 s), comparabilă cu metoda Lambda, care are o valoare ușor mai scăzută dar un MSE mai mare. Efortul de control (media comenzii) în cazul metodei Haalman este moderat (10.46 kW), situându-se în apropierea celorlalte metode eficiente. Prin comparație, metodele Smith și Feedforward înregistrează cele mai slabe rezultate, cu MSE-uri mari (peste 87), suprarăglări de peste 4% și durate mari ale regimului tranzitoriu. Astfel, metoda Haalman oferă un echilibru optim între precizie, stabilitate și consum de energie, fiind considerată cea mai bună dintre metodele comparate.

## 4.2. Analiza reglării MPC

În cadrul acestei lucrări, a fost ales regulatorul MPC furnizat de toolbox-ul MATLAB datorită nivelului său ridicat de integrare, flexibilității în definirea restricțiilor și accesului direct la structuri de optimizare eficientă. Alegerea s-a bazat pe nevoia unui control anticipativ adaptat unui sistem termic, unde evoluția variabilei controlate se produce lent, dar este influențată de limitări fizice asupra acțiunii de control. Toolbox-ul oferă un cadru complet pentru configurarea, simularea și validarea regulatorului, precum și suport pentru implementare în timp real, fiind astfel o alegere robustă și eficientă pentru aplicația analizată.

### 4.2.1. Parametrii configurabili ai regulatorului MPC în MATLAB

Toolbox-ul MPC din MATLAB oferă un set extensiv de parametri de reglaj care permit adaptarea controlerului la particularitățile fiecărui proces. În această secțiune sunt prezentate principalele setări și rolul lor în comportamentul regulatorului:

- **Prediction Horizon ( $N_p$ ):** definește orizontul de timp pentru care regulatorul anticipează comportamentul viitor al sistemului. Un orizont mai lung oferă predicții mai precise, dar crește efortul computațional.
- **Control Horizon ( $N_c$ ):** stabilește numărul de pași pentru care comanda de control este optimizată activ. După acest interval, se presupune că acțiunea de control rămâne constantă. Alegerea  $N_c$  influențează flexibilitatea comenzii și complexitatea algoritmului.
- **Greutăți (Weights):** sunt folosite în funcția de cost a MPC pentru a echilibra între urmărirea traiectoriei și efortul de control. Se definesc mai multe tipuri:
  - *OutputVariables*: penalizează deviația dintre ieșirea sistemului și semnalul de referință.
  - *ManipulatedVariables*: penalizează magnitudinea absolută a comenzilor aplicate.
  - *ManipulatedVariablesRate*: penalizează viteza de variație a comenzilor (echivalent cu parametrul  $\lambda$  în literatura clasică MPC).
- **Restricții:** permit impunerea de limite fizice asupra variabilelor manipulate, ieșirilor și variației comenzilor. Acestea pot reflecta limitări de siguranță, eficiență energetică sau cerințe operaționale.
- **Metoda de optimizare:** MPC MATLAB utilizează algoritmi de programare pătratică (QP) pentru rezolvarea problemei de optimizare. În mod implicit, este folosit algoritmul *KWIK* (active-set rapid), dar sunt disponibile și alte metode precum *interior-point* sau *qpOASES*.
- **Blocarea comenzilor:** opțional, se pot configura mișcări blocate pentru reducerea gradului de libertate al problemei și accelerarea rezolvării, mai ales în sisteme lente sau cu resurse computaționale limitate.



Alegerea acestor parametri influențează direct performanța sistemului, compromisul între viteză de răspuns și stabilitate, precum și capacitatea de a respecta restricțiile fizice sau de siguranță impuse de aplicație. Reglajul fin al acestor valori este esențial pentru obținerea unui control predictiv robust și eficient.

#### 4.2.2. Selecția parametrilor pentru regulatorul MPC

Pentru identificarea configurației optime a regulatorului predictiv utilizat, a fost realizată o analiză sistematică asupra principalilor parametri de reglaj ai MPC: orizontul de predicție ( $H_p$ ), orizontul de control ( $H_c$ ) și factorul de ponderare  $\lambda$  asociat ratei variației comenzii. Strategia adoptată a presupus varierea individuală a fiecărui parametru, în timp ce ceilalți doi erau menținuți constanți, astfel încât să se poată evalua influența fiecărei componente asupra performanței sistemului de reglare.

În prima etapă, a fost variat orizontul de predicție  $H_p \in \{60, 80, 100, 120\}$ , păstrând constante valorile  $H_c = 30$  și  $\lambda = 1$ . Rezultatele, ilustrate în Figura 4.10, arată o îmbunătățire treptată a performanței sistemului pe măsură ce  $H_p$  crește. Inițial pare ca semnalele sunt suprapuse, de asta, în figura 4.11 a fost apropiată mai mult pentru a vedea evoluția semnalelor. Valoarea  $H_p = 120$  conduce la o traiectorie rapidă și stabilă, cu o abatere minimă față de referință și o comandă eficientă.

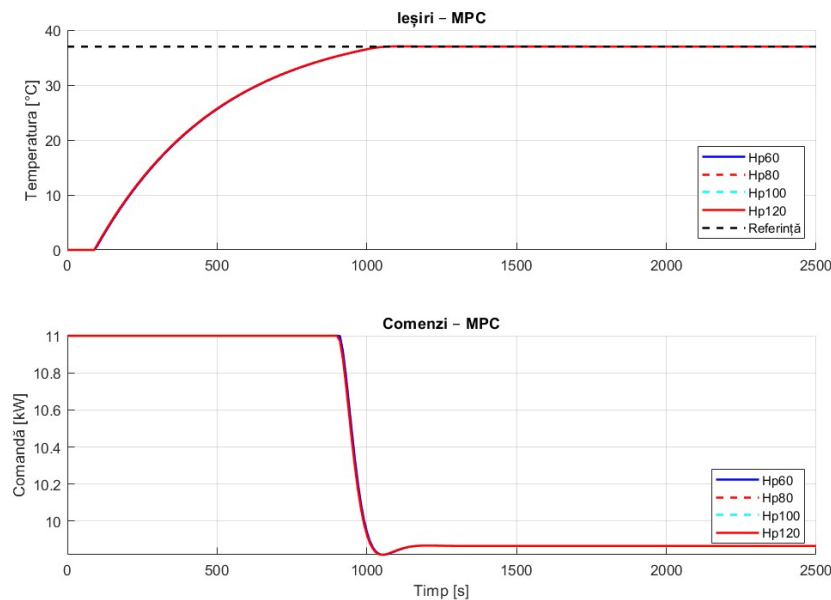


Figura 4.10. Reprezentarea grafică a orizonturilor de predicție alese

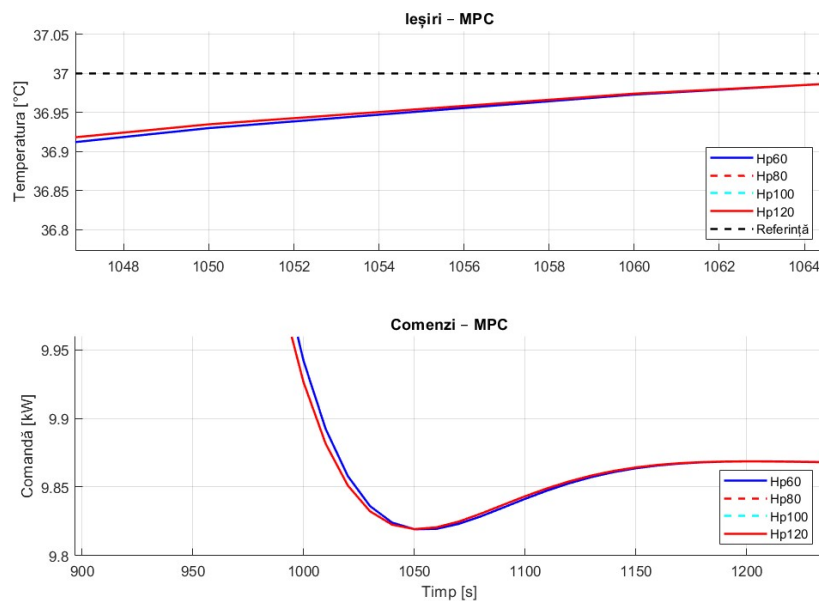


Figura 4.11. Reprezentarea grafica a orizonturilor de predicție alese

În etapa următoare, s-a variat orizontul de control  $H_c \in \{30, 40, 50, 60, 120\}$ , cu  $H_p = 120$  și  $\lambda = 1$  constante. Din analiza grafică (Figura 4.12) reiese că un  $H_c$  mai mare aduce o ameliorare a profilului comenzii, aici semnalele sunt suprapuse, rezultând faptul că poate fi aleasă orice valoare  $H_c$  din cele utilizate.

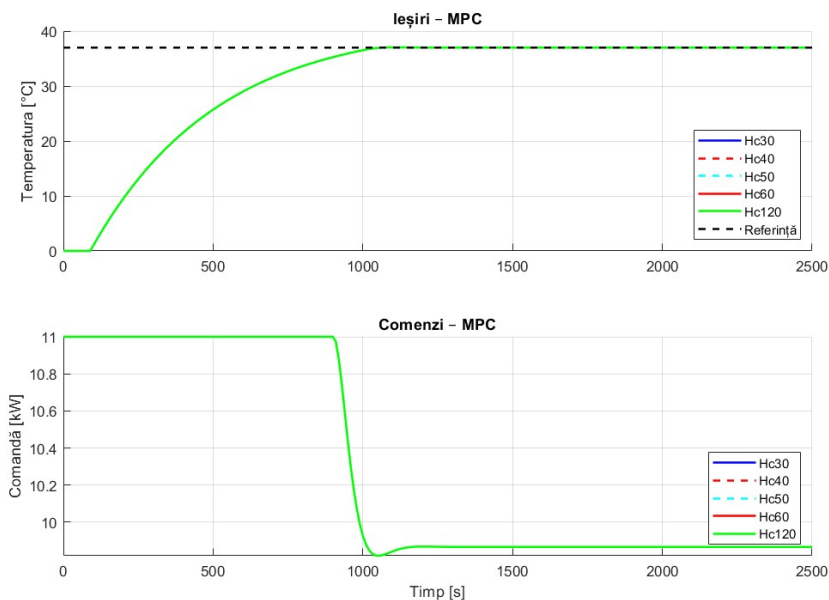


Figura 4.12. Reprezentarea grafica a orizonturilor de control alese

În ultima serie de simulări, a fost variată valoarea ponderii  $\lambda \in \{0.1, 1, 10\}$ , cu  $H_p = 120$  și  $H_c = 120$  constante. Figura 4.13 evidențiază faptul că o valoare mică a ponderii (respectiv  $\lambda = 0.1$ ) oferă o acțiune de control mai energică, permițând sistemului să atingă mai rapid referința, fără a induce instabilitate sau comportamente nedorite. În figura 4.14 se poate observa motivația alegerii valorii de 0.1 pentru acest parametru.

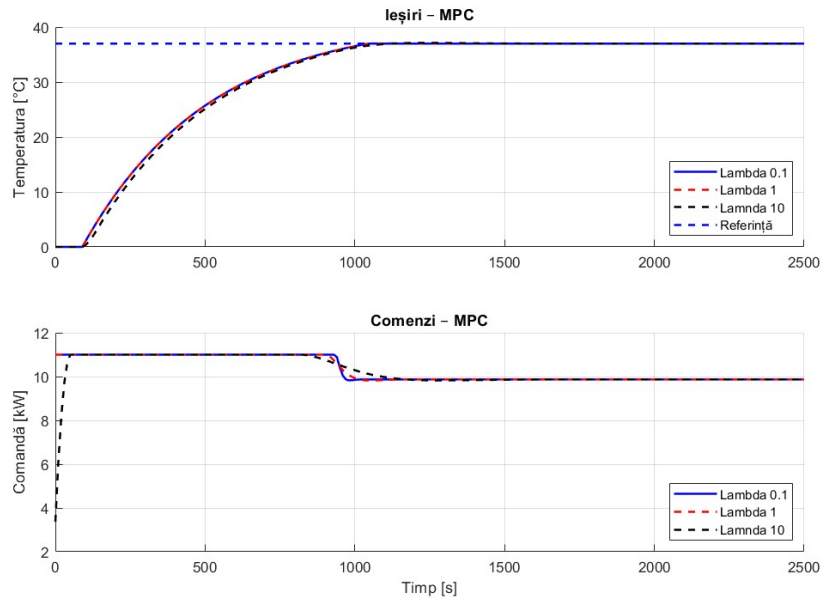


Figura 4.13. Reprezentarea grafica a parametrilor lambda aleși

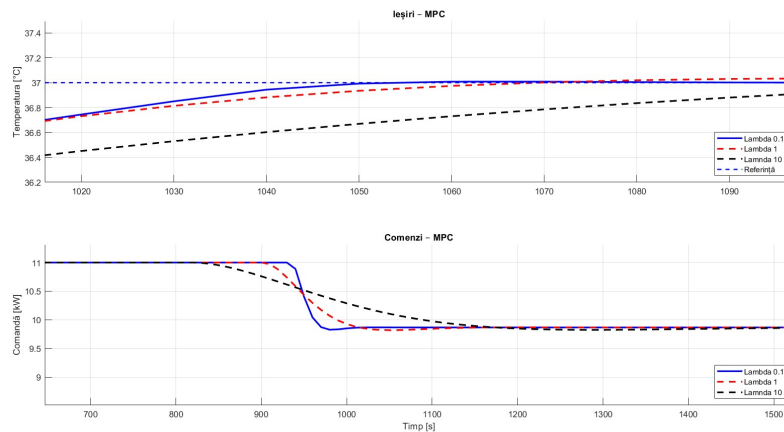


Figura 4.14. Reprezentarea grafica a parametrilor lambda aleși

În urma acestei analize, a fost aleasă configurația:

$$H_p = 120, \quad H_c = 120, \quad \lambda = 0,1$$

Acești parametri asigură cele mai bune performanțe din punct de vedere al timpului de răspuns, al stabilității și al eficienței comenzii, fiind considerați cei mai potriviți pentru implementarea finală a regulatorului MPC în contextul aplicației studiate.

#### 4.2.3. Impactul restricțiilor asupra comenzii în MPC

Pentru a înțelege mai bine influența restricțiilor asupra performanței regulatorului MPC, au fost analizate două scenarii: unul în care nu s-au impus limite asupra comenzii, și altul în care comanda a fost restricționată între valorile 0 kW și 11 kW.

##### 1. MPC fără restricție pe intrare

În primul caz, s-a permis comenzii să varieze liber, fără limitări, iar rezultatul este prezentat în figura 4.15. Se observă o creștere extrem de rapidă a temperaturii spre valoarea de referință

(37°C), realizată într-un timp foarte scurt. Totuși, acest lucru s-a realizat în detrimentul eficienței energetice: comanda generată de regulator a avut valori foarte mari, cu un vârf de peste 400 kW la inițializare. Un astfel de comportament este imposibil de implementat practic, întrucât poate conduce la suprasarcini sau deteriorarea echipamentelor de acționare.

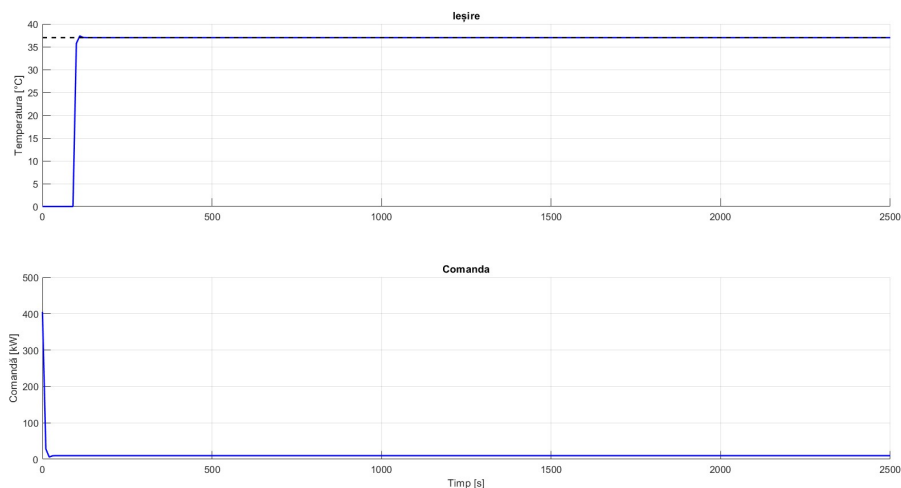


Figura 4.15. Răspunsul MPC fără restricție pe comanda de intrare

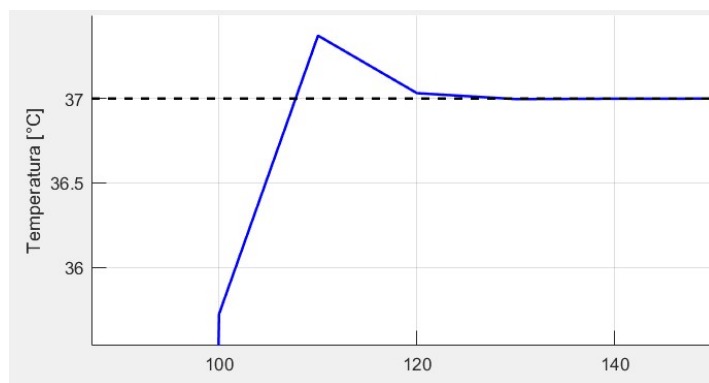


Figura 4.16. Răspunsul MPC fără restricție pe comanda de intrare (mărită)

## 2. MPC cu restricție pe intrare

În al doilea caz, a fost impusă o limită realistă asupra comenzii: 11 kW, corespunzătoare puterii maxime disponibile. Rezultatele sunt prezentate în figura 4.17. Se remarcă o creștere mai lentă a temperaturii, dar comanda este semnificativ mai redusă și constantă, situându-se în limitele impuse. Acest scenariu reflectă o implementare realistă și sustenabilă din punct de vedere energetic.

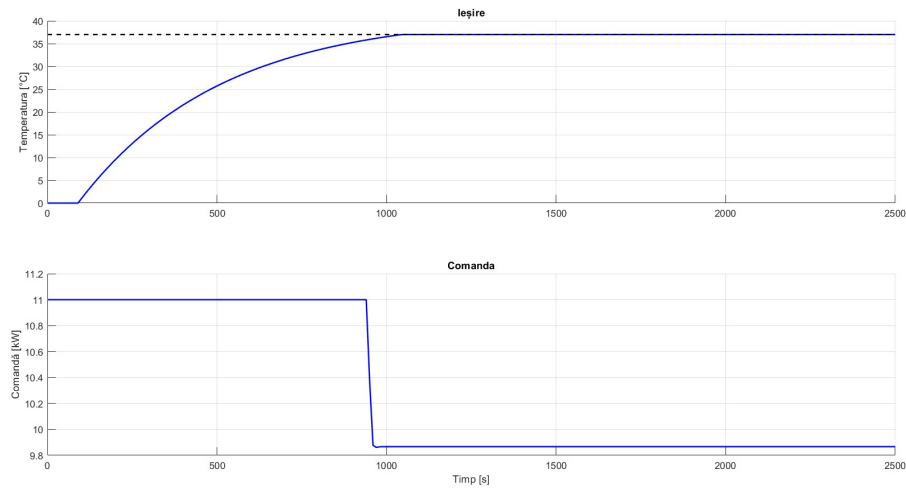


Figura 4.17. Răspunsul MPC cu restricție pe comanda de intrare (0–11 kW)

Compararea celor două cazuri evidențiază importanța impunerii de restricții în cadrul strategiei MPC. Deși eliminarea acestora poate conduce la performanță aparent îmbunătățită în ceea ce privește timpul de răspuns, soluțiile obținute sunt adesea impracticabile. Prin aplicarea limitelor, se obțin comenzi realizabile, cu o creștere graduală a temperaturii, asigurând atât siguranța, cât și eficiența energetică a sistemului.

### 4.3. Comparare MPC cu PID

#### 1. Compararea performanței la schimbarea referinței

În Figura 4.18, este prezentată reacția sistemului la o schimbare a valorii de referință, la 2000 s, de la 35 C la 37 C. Metoda Haalman reacționează mai rapid în faza inițială, atingând valoarea țintă într-un timp mai scurt. Totuși, regulatorul MPC, care a fost configurat cu o restricție activă pe comandă (limitare la 11 kW), oferă un răspuns mai controlat, cu o comandă net mai puțin agresivă. Această restricție, specifică și natural integrabilă în cadrul MPC, contribuie la protejarea echipamentelor și la un consum energetic mai eficient. Deși timpul de răspuns este ușor mai mare, stabilitatea și consumul energetic mai redus fac ca MPC să fie preferabil în aplicații sensibile.

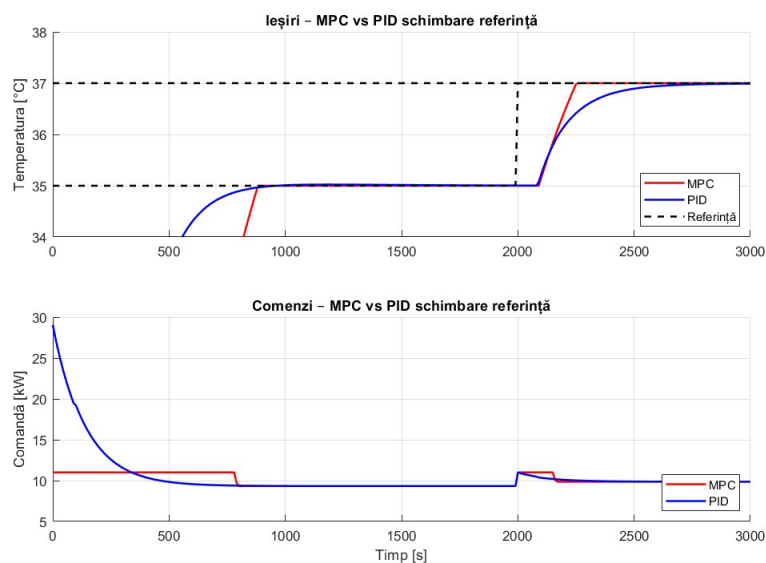


Figura 4.18. Compararea răspunsului MPC vs Haalman la schimbarea referinței

## 2. Compararea în prezența saturației/restricției

Figura 4.19 evidențiază avantajul net al MPC în fața metodei de reglare Haalman în situația în care există limitări impuse asupra comenzii. MPC respectă restricția de 11 kW și reușește să regleze sistemul într-un mod sigur, fără suprasarcini. În schimb, Haalman atinge limita superioară și continuă să aplice o comandă saturată, ceea ce conduce la un comportament oscilant și un control mai puțin eficient. Integrarea naturală a constrângerilor în MPC permite menținerea stabilității și a performanței într-un cadru sigur din punct de vedere energetic.

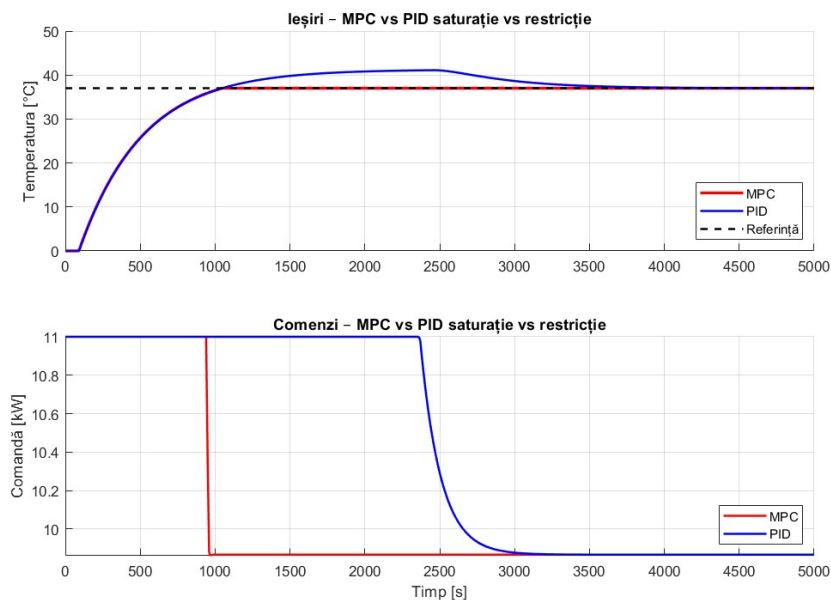


Figura 4.19. MPC vs Haalman în prezența restricțiilor și saturației

## 3. Compararea în cazul rejectării perturbației

În Figura 4.20, este analizată capacitatea de rejectare a unei perturbații bruște. La ieșirea sistemului a fost adăugat o treaptă la 2000 s cu valoarea -5. MPC reacționează rapid și eficient, reușind să readucă temperatura la valoarea de referință cu o comandă bine controlată, în limitele restricției impuse. Metoda de reglare Haalman, deși corectează în final deviația, o face într-un timp mai lung și cu un efort de control mai ridicat. Această comparație subliniază robustețea și eficiența MPC în fața perturbațiilor, mai ales când constrângerile fizice ale sistemului trebuie respectate.

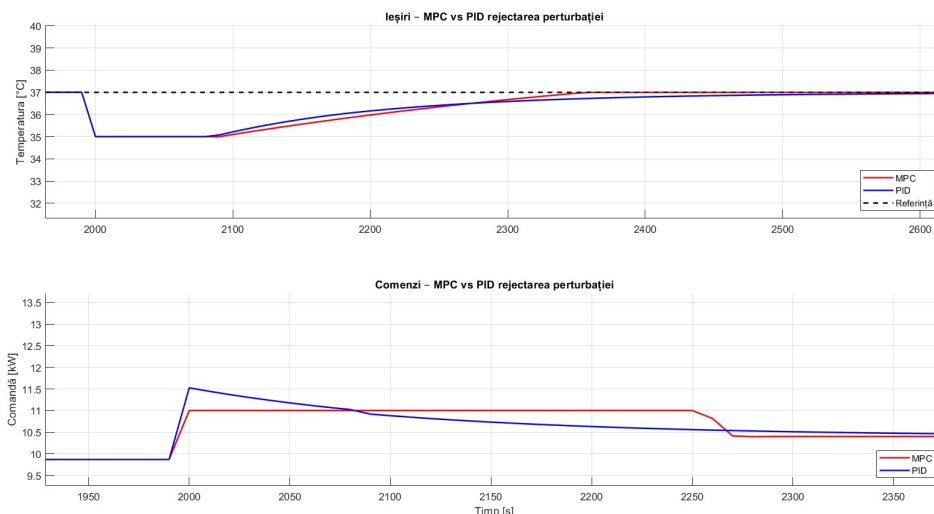


Figura 4.20. MPC vs Haalman – rejectarea unei perturbații

#### Compararea performanțelor între MPC și metoda Haalman

Pentru a evalua în mod obiectiv performanțele regulatorului predictiv (MPC) în comparație cu metoda clasică de reglare PID utilizând acordarea Haalman, au fost calculate și comparate mai multe criterii cantitative relevante: eroarea medie pătratică (MSE), suprareglarea, durata regimului tranzitoriu și media comenzii (efortul de control). Am utilizat MPC cu restricție (11kW) și Haalman fără saturație, iar referința de 37 C.

- **1. Eroarea medie pătratică (MSE):** Metoda Haalman a obținut un MSE mai mic (45.09) față de MPC (73.61). Acest rezultat poate fi explicat prin faptul că MPC-ul cu restricții reduce suprareglarea și efortul energetic, dar în detrimentul unui răspuns inițial mai lent, ceea ce duce la o eroare pătratică mai mare pe intervalul total analizat.
- **2. Suprareglarea:** MPC nu prezintă suprareglare, în timp ce metoda Haalman depășește referința cu aproximativ 0.06%. Lipsa suprareglării este un avantaj major al metodei MPC, indicând un control mai fin și mai robust, fără oscilații inutile.
- **3. Durata regimului tranzitoriu:** Regulatorul MPC aduce sistemul la regim permanent într-un timp mult mai scurt ( $\approx 1050s$ ), comparativ cu PID-ul acordat cu Haalman ( $\approx 1400s$ ), comparație ce a fost făcută pe MPC cu restricție la intrare, iar Haalman fără saturație. Această diferență subliniază eficiența MPC în atingerea rapidă a valorii dorite, chiar în prezența restricțiilor.
- **4. Efortul de control:** Media comenzii aplicate este mai mică în cazul MPC (10.08 kW) decât în cazul Haalman (10.46 kW), ceea ce înseamnă că MPC este mai economic din punct de vedere energetic, un aspect important în aplicațiile industriale.

În concluzie, deși metoda Haalman oferă o eroare pătratică mai mică în ansamblu, regulatorul MPC excelează în privința timpului de reglare, a eliminării suprareglării și a eficienței energetice. Utilizarea restricției pe comandă în cazul MPC nu doar că asigură protecția fizică a actuatorului, dar și evidențiază capacitatea sa de a obține performanțe ridicate fără oscilații sau suprasarcini, ceea ce nu poate fi realizat la fel de ușor cu un regulator PID clasic. Astfel, MPC reprezintă o soluție modernă și robustă pentru controlul precis al proceselor termice lente.

---

## Capitolul 5. Concluzii

În această lucrare au fost analizate două strategii de control aplicate unui proces termic: metoda clasică PID, unde au fost comparate mai multe metode de acordare precum: Haalman, Lambda, ISE, ISTE,  $IST^2E$ , alocare poli-zero-uri cu predictor Smith și, respectiv, Feedforward și regulatorul predictiv bazat pe model (MPC), implementat cu ajutorul toolboxului oferit de MATLAB. Contribuția personală a constat în explorarea parametrilor MPC – orizontul de predicție ( $H_p$ ), orizontul de control ( $H_c$ ) și ponderea pe variația comenzii ( $\lambda$ ) – pentru identificarea combinației care oferă cele mai bune performanțe în contextul aplicației. Totodată, a fost pus în evidență un avantaj major al MPC: posibilitatea de a impune restricții asupra comenzii, lucru care a fost utilizat în toate testele relevante pentru a reflecta condițiile reale de operare ale sistemului.

Rezultatele obținute în urma simulărilor arată că MPC este capabil să aducă sistemul la starea dorită într-un timp mai scurt, fără oscilații și cu o comandă mai economică energetic comparativ cu metoda Haalman. Deși PID-ul a avut o eroare pătratică totală (MSE) mai mică, acest rezultat este influențat de timpul mai lung de reglare și de amplitudinea mai mare a comenzii în cazul său. MPC a oferit un compromis superior între rapiditatea răspunsului, absența suprareglării și eficiența energetică.

Ca direcții de dezvoltare viitoare, se propune extinderea acestei lucrări prin integrarea unor metode de identificare online a modelului procesului, aplicarea controlului predictiv neliniar sau evaluarea performanței în prezența unor perturbații externe mai complexe. De asemenea, o testare pe un sistem fizic real ar reprezenta un pas important în validarea completă a soluției dezvoltate.



## Bibliografie

- [1] D. E. Seborg, T. F. Edgar, D. A. Mellichamp, and F. J. Doyle III, *Process dynamics and control*, 4th ed. John Wiley & Sons, 2011, vol. 4.
- [2] K. Lawton, S. Patterson, and R. Keanini, *Precision Temperature Control of High-Throughput Fluid Flows: Theoretical and Experimental Analysis*. Journal of Heat Transfer, 2001, vol. 123.
- [3] J. Zhang, M. Zhang, and J. Xu, *Research on Bathtub Water Temperature Control Strategy and Model*, 2021, vol. 5.
- [4] H. Liu, *A design of a household output water temperature control system based on the PID controller and feedforward control*, 2024, vol. 2711, no. 1.
- [5] Z. Instruments, “Principles of pid controllers,” *Zurich Instruments: Zürich, Switzerland*, 2023.
- [6] C. Knospe, “Pid control,” *IEEE Control Systems Magazine*, vol. 26, no. 1, pp. 30–31, 2006.
- [7] P. em. dr. ing. Corneliu Lazăr, *Ingineria reglării automate - note de curs 2025*. Facultatea de Automatică si Calculatoare, 2025.
- [8] E. F. Camacho and C. Bordons, *Constrained Model Predictive Control*. London: Springer London, 2007, pp. 177–216. [Online]. Available: [https://doi.org/10.1007/978-0-85729-398-5\\_7](https://doi.org/10.1007/978-0-85729-398-5_7)
- [9] The MathWorks, Inc., *Model Predictive Control Toolbox User's Guide*, MathWorks, 2024, <https://www.mathworks.com/help/mpc/>.
- [10] S. Limin and L. Ping, “Research on temperature control system of baby bath center based on fuzzy pid control,” *Revista Ibérica de Sistemas e Tecnologias de Informação*, no. 18A, p. 306, 2016.



## Anexe

### Anexa 1. Cod MATLAB - regulatoare PID

```

1  tih=420;
2  num=3.75;
3  den=[420 1];
4  Ts=10;
5  delay=90;
6  SYS=tf(num,den,'InputDelay',delay); %Funcția de transfer continua
7  hd=c2d(SYS,Ts); %Funcția de transfer discreta
8  step(SYS,hd)
9  Gm=tf(num,den);
10 Gz=c2d(Gm,Ts);
11 t = 0:10:5000; % Interval de timp (secunde)
12 % Aproximare Pade pentru întârziere (ordinul 1 recomandat pentru MPC)
13 G_pade = pade(SYS, 1);
14
15 % Conversie în model de stare
16 sys_ss = ss(G_pade);
17
18 % Afișare rezultate
19 disp('Modelul în spațiu de stare aproximat:');
20 sys_ss
21 disp('Funcția de transfer:');
22 disp(hd);
23
24 u = 11 * ones(size(t)); % kW
25
26 % Simularea răspunsului sistemului la intrare
27 [y, t_out] = lsim(SYS, u, t);
28
29 % Crearea figurii
30 figure;
31
32 % Subplotul 1: Ieșirea sistemului (Temperatura)
33 subplot(2,1,1);
34 plot(t_out, y, 'b', 'LineWidth', 2);
35 grid on;
36 title('Ieșirea sistemului (Temperatura)');
37 xlabel('Timp (s)');
38 ylabel('Temperatura (\circC)');
39 legend('T_{out}', 'Location', 'Best');
40
41 % Subplotul 2: Semnalul de intrare (Puterea)
42 subplot(2,1,2);
43 plot(t, u, 'r', 'LineWidth', 2);
44 grid on;
45 title('Semnal de intrare (Puterea încălzitorului)');
46 xlabel('Timp (s)');
47 ylabel('Putere (kW)');
48 legend('Q_{in}', 'Location', 'Best');



```

```
49
50  sgttitle('Sistem în buclă deschisă'); % Titlu general
51
52  %% proiectarea regulatorului
53  sigma=0.043;
54  tt=1200;
55  %rezulta
56  zeta=-log(sigma)/sqrt(pi^2+log(sigma)^2);
57  wn=4/(tt*zeta);
58  %calcularea Gr
59  Gd_num=wn^2;
60  Gd_den=[1 2*zeta*wn 0];
61  Gd=tf(Gd_num,Gd_den);
62
63  Gr=minreal(Gd/(Gm));
64  poli=abs(pole(Gm));
65  P1=2*zeta*wn;
66  GR=Gd*3.75;
67  %% Metoda lambda
68  kf=3.75;
69  lambda=200;
70  kpl=tih/(kf*(lambda+delay));
71
72
73  %% metoda Haalman
74  kf=3.75;
75  tih=420;
76  Gdh=tf(2,270,'InputDelay',90);
77  kph=2*420/(3*90*3.75);
78  %Zhuang-Atherton
79  R = delay / tih;
80  %% ISE
81  Kp_ISE = (1.048/kf)*((delay/tih)^(-0.897));
82  Ti_ISE = tih/(1.195+(-0.368*delay/tih));
83  Td_ISE = 0.489*tih*((delay/tih)^0.888);
84  kp_ISE = Kp_ISE;
85  ki_ISE = kp_ISE/Ti_ISE;
86  kd_ISE = kp_ISE*Td_ISE;
87
88  %% ISTE
89  Kp_ISTE = (1.042/kf)*((delay/tih)^(-0.897));
90  Ti_ISTE = tih/(0.987+(-0.368*delay/tih));
91  Td_ISTE = 0.489*tih*((delay/tih)^0.888);
92  kp_ISTE = Kp_ISTE;
93  ki_ISTE = kp_ISTE/Ti_ISTE;
94  kd_ISTE = kp_ISTE*Td_ISTE;
95
96  %% IST2E
97  Kp_IST2E = (0.968/kf)*((delay/tih)^(-0.904));
98  Ti_IST2E = tih/(0.977+(-0.253*delay/tih));
99  Td_IST2E = 0.316*tih*((delay/tih)^0.892);
100  kp_IST2E = Kp_IST2E;
101  ki_IST2E = kp_IST2E/Ti_IST2E;
```

```

102 kd_IST2E = kp_IST2E*Td_IST2E;
103
104 %% feedforward
105 kp=3;
106 Tp=50;
107 kp1=2;
108 Tp1=300;
109 kp2=5;
110 Tp2=200;
111 Gp=tf(kp,Tp);
112 Gp1=tf(kp1,Tp1);
113 Gp2=tf(kp2,Tp2);
114 s=tf('s');
115 Gff = - (kp * (1 + s*tih)) / (kf * (1 + s*Tp));
116 Gff1 = - (kp1 * (1 + s*tih)) / (kf * (1 + s*Tp1));
117 Gff2 = - (kp2 * (1 + s*tih)) / (kf * (1 + s*Tp2));
118
119
120
121 %% MSE
122 MSE_Lambda = mean((out.referinta_haalman - out.iesire_Lambda).^2);
123 MSE_Haalman = mean((out.referinta_haalman - out.iesire_haalman).^2);
124 MSE_ISE = mean((out.referinta_haalman - out.iesire_ISE).^2);
125 MSE_ISTE = mean((out.referinta_haalman - out.iesire_ISTE).^2);
126 MSE_IST2E = mean((out.referinta_haalman - out.iesire_IST2E).^2);
127 MSE_Smith =mean((out.referinta_haalman - out.iesire_Smith).^2);
128 MSE_ff =mean((out.referinta_haalman - out.iesire_feedforward).^2);
129
130 % Valori MSE pentru metodele testate
131 valori_MSE = [...
132     mean((out.referinta_haalman - out.iesire_Lambda).^2),...
133     mean((out.referinta_haalman - out.iesire_haalman).^2),...
134     mean((out.referinta_haalman - out.iesire_ISE).^2),...
135     mean((out.referinta_haalman - out.iesire_ISTE).^2),...
136     mean((out.referinta_haalman - out.iesire_IST2E).^2),...
137     mean((out.referinta_haalman - out.iesire_Smith).^2),...
138     mean((out.referinta_haalman - out.iesire_feedforward).^2)...
139 ];
140
141 % Etichetele corespunzătoare
142 etichete_MSE = {...
143     'Lambda', ...
144     'Haalman', ...
145     'ISE', ...
146     'IST2E', ...
147     'ISTE', ...
148     'Smith',...
149     'Feedforward',...
150     'MPC'...
151
152 };
153
154 % Creare grafic bară

```

```
155 figure;
156 b = bar(valori_MSE, 'FaceColor', [0.3 0.4 0.8]); % culoare albastru
    → deschis
157 text(1:length(valori_MSE), valori_MSE, num2str(valori_MSE, '%.4f'), ...
158     'HorizontalAlignment', 'center', 'VerticalAlignment', 'bottom', ...
159     'FontSize', 10);
160
161 % Personalizare axă X
162 set(gca, 'XTickLabel', etichete_MSE, 'XTickLabelRotation', 45,
    → 'FontSize', 10);
163
164 % Etichete și titlu
165 ylabel('MSE');
166 title('MSE pentru metodele de acordare PID');
167
168 % Activare grilă
169 grid on;
170
171 % Setează dimensiune figură
172 set(gcf, 'Position', [100 100 900 500]);
173 saveas(gcf, 'MSE.png')
174
175 %% ieseire si comanda
176 figure;
177
178 % Subplot 1  Ieșiri
179 subplot(2,1,1);
180 plot(out.tout, out.ieseire_Lambda, 'g', 'LineWidth', 2); hold on;
181 plot(out.tout, out.referinta_haalman, 'k--', 'LineWidth', 2); % după
    → ieșire
182 xlabel('Timp [s]');
183 ylabel('Temperatura [C]');
184 title('Ieșire sistem');
185 legend('Funcție de transfer', 'Referință');
186 grid on;
187
188 % Subplot 2  Comenzi
189 subplot(2,1,2);
190 plot(out.tout, out.comanda_Lambda, 'b', 'LineWidth', 2);
191 xlabel('Timp [s]');
192 ylabel('Comandă u [W]');
193 title('Comanda aplicată');
194 legend('Funcție de transfer');
195 grid on;
196
197 saveas(gcf, 'IST2E_comanda_raspuns.png')
198
199 %% DURATA REGIMULUI TRANZITORIU
200 figure;
201 metode = {'Lambda', 'Haalman', 'ISE', 'ISTE', 'IST2E', 'Smith',
    → 'FeedForward'};
202 t_tranzitor = [1500, 1400, 2500, 2400, 2500, 2300, 2200];
203 bar(t_tranzitor);
```

```

204 set(gca, 'xticklabel', metode);
205 yline(2500, '--k', '2500s', 'LabelHorizontalAlignment', 'right',
    → 'LabelVerticalAlignment', 'top');
206 ylabel('Durata regimului tranzitoriu [s]');
207 title('Compararea duratelor regimului tranzitoriu');
208 grid on;
209 saveas(gcf, 'durata_regimului_tranzitoriu_toate_metodele.png');
210
211 %% medie comanda
212
213 cLambda = mean(out.comanda_Lambda);
214 chaalman = mean(out.comanda_haalman);
215 cISE = mean(out.comanda_ISE);
216 cISTE = mean(out.comanda_ISTE);
217 cIST2E = mean(out.comanda_IST2E);
218 cSmith = mean(out.comanda_Smith);
219 cFF = mean(out.comanda_feedforward);
220
221 % Valori deja calculate ale mediei comenzilor
222 valori_medie_comanda_PID = [
223 mean(out.comanda_Lambda), ...
224 mean(out.comanda_haalman), ...
225 mean(out.comanda_ISE), ...
226 mean(out.comanda_ISTE), ...
227 mean(out.comanda_IST2E), ...
228 mean(out.comanda_Smith), ...
229 mean(out.comanda_feedforward)
230 ];
231
232 etichete_medie_PID = {'Lambda', 'Haalman', 'ISE', 'ISTE', 'IST2E',
    → 'Smith', 'FeedForward'};
233
234 i
235 figure;
236 b = bar(valori_medie_comanda_PID, 'FaceColor', [0.3 0.6 0.4]); %
    → verde-albăstrui
237
238 % Afișare valori numeric deasupra barelor
239 text(1:length(valori_medie_comanda_PID), valori_medie_comanda_PID, ...
240     num2str(valori_medie_comanda_PID, '%.4f'), ...
241     'HorizontalAlignment', 'center', 'VerticalAlignment', 'bottom', ...
242     'FontSize', 10);
243
244 % Personalizare axă X
245 set(gca, 'XTickLabel', etichete_medie_PID, 'XTickLabelRotation', 45,
    → 'FontSize', 10);
246
247 % Etichete și titlu
248 yline(12);
249 ylabel('Media comenzii [kW]');
250 title('Efortul de control - media comenzii pentru metodele PID');
251 grid on;
252

```

```
253 % Setează dimensiunea ferestrei
254 set(gcf, 'Position', [100 100 900 500]);
255
256 % Salvare imagine (opțional)
257 saveas(gcf, 'MediaComanda_MPC_manual.png');
258
259
260 %% Calcul și grafic pentru suprareglarea metodelor PID
261
262 y_ref = 37; % Valoarea de referință finală
263
264 % Etichetele pentru metodele PID
265 etichete_PID = {
266     'Lambda', ...
267     'haalman', ...
268     'ISE', ...
269     'ISTE', ...
270     'IST2E', ...
271     'Smith', ...
272     'Feedforward'
273 };
274
275 % Calcul suprareglare [%]
276 valori_suprareglare_PID = [ ...
277     (max(out.iesire_Lambda) - y_ref) / y_ref * 100, ...
278     (max(out.iesire_haalman) - y_ref) / y_ref * 100, ...
279     (max(out.iesire_ISE) - y_ref) / y_ref * 100, ...
280     (max(out.iesire_ISTE) - y_ref) / y_ref * 100, ...
281     (max(out.iesire_IST2E) - y_ref) / y_ref * 100, ...
282     (max(out.iesire_Smith) - y_ref) / y_ref * 100, ...
283     (max(out.iesire_feedforward) - y_ref) / y_ref * 100 ...
284 ];
285
286 % Creare grafic bară
287 figure;
288 bar(valori_suprareglare_PID, 'FaceColor', [0.9 0.6 0.2]);
289
290 % Afișare valori numeric deasupra barelor
291 text(1:length(valori_suprareglare_PID), valori_suprareglare_PID, ...
292     num2str(valori_suprareglare_PID, '%.2f'), ...
293     'HorizontalAlignment', 'center', 'VerticalAlignment', 'bottom', ...
294     'FontSize', 10);
295
296 % Personalizare axă X
297 set(gca, 'XTickLabel', etichete_PID, 'XTickLabelRotation', 45,
298     ↪ 'FontSize', 10);
299
300 % Etichete și titlu
301 ylabel('Suprareglare [%]');
302 title('Suprareglare pentru metodele PID');
303 grid on;
304
305 % Setare dimensiune fereastră
```



```

305 set(gcf, 'Position', [100 100 900 500]);
306
307 %% toate metodele PID
308 figure;
309 t = out.tout;
310
311 subplot(2,1,1);
312 hold on; grid on;
313 plot(t, out.iesire_Lambda, 'b', 'LineWidth', 1.5);
314 plot(t, out.iesire_haalman, 'r', 'LineWidth', 1.5);
315 plot(t, out.iesire_ISE, 'c', 'LineWidth', 1.5);
316 plot(t, out.iesire_ISTE, 'y', 'LineWidth', 1.5);
317 plot(t, out.iesire_IST2E, '--b', 'LineWidth', 1.5);
318 plot(t, out.iesire_Smith, '--y', 'LineWidth', 1.5);
319 plot(t, out.iesire_feedforward, '--c', 'LineWidth', 1.5);
320 plot(t, out.referinta_haalman, 'k--', 'LineWidth', 2); % Referință
321
322 ylabel('Temperatura [°C]');
323 title('Ieșiri - Metode PID');
324 legend({'Lambda', 'Haalman', 'ISE', 'ISTE', 'IST2E',
325        ↪ 'Smith', 'Feedforward', 'Referință'}, ...
326        'Location', 'southeast');
327
328 subplot(2,1,2);
329 hold on; grid on;
330 plot(t, out.comanda_Lambda, 'b', 'LineWidth', 1.5);
331 plot(t, out.comanda_haalman, 'r', 'LineWidth', 1.5);
332 plot(t, out.comanda_ISE, 'c', 'LineWidth', 1.5);
333 plot(t, out.comanda_ISTE, 'y', 'LineWidth', 1.5);
334 plot(t, out.comanda_IST2E, '--b', 'LineWidth', 1.5);
335 plot(t, out.comanda_Smith, '--y', 'LineWidth', 1.5);
336 plot(t, out.comanda_feedforward, '--c', 'LineWidth', 1.5);
337
338 xlabel('Timp [s]');
339 ylabel('Comandă [kW]');
340 title('Comenzi - Metode PID');
341 legend({'Lambda', 'Haalman', 'ISE', 'ISTE', 'IST2E',
342        ↪ 'Smith', 'Feedforward'}, ...
343        'Location', 'southeast');
344
345 set(gcf, 'Position', [100 100 900 600]);
346 saveas(gcf, 'rezultate_metode_PID_subplot.png');
347
348 save('rezultate_PID.mat', 'out'); % out.iesire_X, out.comanda_X,
349 ↪ out.referinta_haalman
350
351 %Aleg Lambda
352
353 %% ff
354 figure;
355 t = out.tout;

```

```
355 ref=37*ones(301,1);
356 hold on; grid on;
357 plot(t, out.iesire_ff, 'r', 'LineWidth', 2);
358 plot(t, out.iesire_poli, 'b', 'LineWidth', 1.5);
359 plot(t, out.referinta_haalman, 'k--', 'LineWidth', 1.5);
360 plot(t, ref, 'k--', 'LineWidth', 1.5);
361 ylabel('Temperatura [°C]');
362 title('Ieşiri - Sistem cu şi fără Feedforward');
363 legend({'Cu Ff', 'Fără Ff', 'Referinţă'}, ...
364        'Location', 'southeast');
```

Listing 1. Cod MATLAB- Regulateoare PID

### *Anexa 2. Cod MATLAB - regulator MPC utilizat*

```
1 %% Configurare MPC
2 tih=420;
3 num=3.75;
4 den=[420 1];
5 Ts=10;
6 delay=90;
7 SYS=tf(num,den,'InputDelay',delay); %Functia de transfer continua
8 hd=c2d(SYS,Ts);
9 mpcobj = mpc(hd, Ts);
10 mpcobj.PredictionHorizon = 120;%hp
11 mpcobj.ControlHorizon =120;%hc
12 mpcobj.Weights.OutputVariables = 10;
13 mpcobj.Weights.ManipulatedVariables = 5*1e-6; %agresivitate pe comanda
14 mpcobj.Weights.ManipulatedVariablesRate = 0.1; %lambda
15 mpcobj.MV.Min = 0;
16 mpcobj.MV.Max = 11;%kW
17 %Best Mpc hp=120,hc=120,lambda=0.1
18 %% figura MPC in_out
19
20 figure;
21 t = out.tout;
22 ref=37*ones(301,1);
23 subplot(2,1,1);
24 hold on; grid on;
25 plot(t, out.iesire_MPC_eu, 'b', 'LineWidth', 1.5);%best
26 plot(t, ref, 'k--', 'LineWidth', 1.5);
27 ylabel('Temperatura [°C]');
28 title('Ieşire');
29 subplot(2,1,2);
30 hold on; grid on;
31 plot(t, out.comanda_MPC_eu, 'b', 'LineWidth', 1.5);
32
33
34 xlabel('Timp [s]');
35 ylabel('Comandă [kW]');
36 title('Comanda');
37
38
39 set(gcf, 'Position', [100 100 900 600]);
```

## Listing 2. Cod MATLAB- regulator MPC utilizat

*Anexa 3. Cod MATLAB - Generare MPC-uri*

```

1  % Sistem discretizat
2  Ts = 10;
3  hd = c2d(tf(3.75, [420 1], 'InputDelay', 90), Ts);
4
5  % Inițializare
6  obiecte = {};
7
8  %% Variație Hp (cu Hc și constante)
9  valori_Hp = [60, 80, 100, 120];
10 for Hp = valori_Hp
11     mpc_tmp = mpc(hd, Ts);
12     mpc_tmp.PredictionHorizon = Hp;
13     mpc_tmp.ControlHorizon = 30;
14     mpc_tmp.Weights.OutputVariables = 1;
15     mpc_tmp.Weights.ManipulatedVariables = 5*1e-4;
16     mpc_tmp.Weights.ManipulatedVariablesRate = 1;
17     mpc_tmp.MV.Min = 0;
18     mpc_tmp.MV.Max = 11;
19
20     nume = sprintf('mpc_Hp%d', Hp);
21     eval([nume ' = mpc_tmp;']);
22     obiecte{end+1} = nume;
23 end
24
25 %% Variație Hc (cu Hp și constante)
26 valori_Hc = [30, 40, 50, 60, 120];
27 for Hc = valori_Hc
28     mpc_tmp = mpc(hd, Ts);
29     mpc_tmp.PredictionHorizon = 120;
30     mpc_tmp.ControlHorizon = Hc;
31     mpc_tmp.Weights.OutputVariables = 1;
32     mpc_tmp.Weights.ManipulatedVariables = 5*1e-4;
33     mpc_tmp.Weights.ManipulatedVariablesRate = 1;
34     mpc_tmp.MV.Min = 0;
35     mpc_tmp.MV.Max = 11;
36
37     nume = sprintf('mpc_Hc%d', Hc);
38     eval([nume ' = mpc_tmp;']);
39     obiecte{end+1} = nume;
40 end
41
42 %% Variație lambda (cu Hp și Hc constante)
43 valori_lambda = [0.1, 1, 10];
44 for lambda = valori_lambda
45     mpc_tmp = mpc(hd, Ts);
46     mpc_tmp.PredictionHorizon = 120;
47     mpc_tmp.ControlHorizon = 120;
48     mpc_tmp.Weights.OutputVariables = 1;

```

```

49     mpc_tmp.Weights.ManipulatedVariables = 5*1e-4;
50     mpc_tmp.Weights.ManipulatedVariablesRate = lambda;
51     mpc_tmp.MV.Min = 0;
52     mpc_tmp.MV.Max = 11;
53
54     lambda_str = strrep(num2str(lambda), '.', '_');
55     nume = sprintf('mpc_lambda_%s', lambda_str);
56     eval([nume ' = mpc_tmp;']);
57     obiecte{end+1} = nume;
58 end
59
60
61
62
63
64 %% Salvare toate obiectele într-un fișier .mat
65 save('obiecte_MPC.mat', obiecte{:});
66 disp('Obiectele MPC au fost generate și salvate.');
```

Listing 3. Cod MATLAB- Generare MPC-uri

#### *Anexa 4. Cod MATLAB - Comparare MPC-uri*

```

1  referinta=35*ones(501,1);
2  %% MSE
3  meu = mean((referinta - out.iesire_MPC_eu).^2);
4  mHc10 = mean((referinta - out.iesire_MPC_hc10).^2);
5  mHc20 = mean((referinta - out.iesire_MPC_hc20).^2);
6  mHc30 = mean((referinta - out.iesire_MPC_hc30).^2);
7  mHc5 = mean((referinta - out.iesire_MPC_hc5).^2);
8  mHp100 =mean((referinta - out.iesire_MPC_hp100).^2);
9  mHp20 =mean((referinta - out.iesire_MPC_hp20).^2);
10 mHp40 =mean((referinta - out.iesire_MPC_hp40).^2);
11 mHp60 =mean((referinta - out.iesire_MPC_hp60).^2);
12 mHp80 =mean((referinta - out.iesire_MPC_hp80).^2);
13 mlambda10 =mean((referinta - out.iesire_MPC_lambda10).^2);
14 mlambda0_1 =mean((referinta - out.iesire_MPC_lambda0_1).^2);
15 mlambda1 =mean((referinta - out.iesire_MPC_lambda1).^2);
16
17 % Valori MSE pentru metodele testate
18 valori_MSE = [...
19     72.4815,...      % eu
20     62.4289, ...    % Hc10
21     62.4293, ...    % Hc20
22     62.4293, ...    % Hc30
23     62.4286, ...    % Hc5
24     62.4289, ...    % hp100
25     62.4355, ...    % hp20
26     62.4297, ...    % hp40
27     62.4288, ...    % hp60
28     62.4288, ...    % hp80
29     66.3866, ...    % lambda10
30     62.4291, ...    % lambda0_1
31     62.4289, ...    % lambda1
```

```

32 ];
33
34 % Etichetele corespunzătoare
35 etichete_MSE = {...
36     'eu', ...
37     'Hc10', ...
38     'Hc20', ...
39     'Hc30', ...
40     'Hc5', ...
41     'hp100',...
42     'hp20',...
43     'hp40',...
44     'hp60',...
45     'hp80',...
46     'lambda10',...
47     'lambda0_1',...
48     'lambda1',...
49
50 };
51
52 % Creare grafic bară
53 figure;
54 b = bar(valori_MSE, 'FaceColor', [0.3 0.4 0.8]); % culoare albastru
    ↪ deschis
55
56 % Afișare valori numeric deasupra barelor
57 text(1:length(valori_MSE), valori_MSE, num2str(valori_MSE, '%.4f'), ...
58     'HorizontalAlignment','center', 'VerticalAlignment','bottom', ...
59     'FontSize', 10);
60
61 % Personalizare axă X
62 set(gca, 'XTickLabel', etichete_MSE, 'XTickLabelRotation', 45,
    ↪ 'FontSize', 10);
63
64 % Etichete și titlu
65 ylabel('MSE');
66 title('MSE pentru metodele de acordare PID');
67
68 % Activare grilă
69 grid on;
70
71 % Setează dimensiune figură
72 set(gcf, 'Position', [100 100 900 500]);
73 saveas(gcf, 'MSE.png')
74
75 %% Medie comanda
76
77 % Valori deja calculate ale mediei comenzilor
78 valori_medie_comanda = [
79     mean(out.comanda_MPC_eu),...
80     mean(out.comanda_MPC_hc30),...
81     mean(out.comanda_MPC_hc40),...
82     mean(out.comanda_MPC_hc50),...

```

```
83 mean(out.comanda_MPC_hc60),...
84 mean(out.comanda_MPC_hc120),...
85 mean(out.comanda_MPC_hp120),...
86 mean(out.comanda_MPC_hp100),...
87 mean(out.comanda_MPC_hp60),...
88 mean(out.comanda_MPC_hp80),...
89 mean(out.comanda_MPC_lambda10),...
90 mean(out.comanda_MPC_lambda_0_1),...
91 mean(out.comanda_MPC_lambda1)
92 ];
93
94 etichete_medie = {
95     'MPC_eu', ...
96     'Hc30', ...
97     'Hc40', ...
98     'Hc50', ...
99     'Hc60', ...
100    'Hc120',...
101    'Hp130', ...
102    'Hp100', ...
103    'Hp60', ...
104    'Hp80', ...
105    'lambda10', ...
106    'lambda0_1', ...
107    'lambda1'
108 };
109
110 % Creare grafic bară pentru media comenzii
111 figure;
112 b = bar(valori_medie_comanda, 'FaceColor', [0.3 0.6 0.4]); %
    ↪ verde-albăstrui
113
114 % Afișare valori numeric deasupra barelor
115 text(1:length(valori_medie_comanda), valori_medie_comanda, ...
116     num2str(valori_medie_comanda','%.2f'), ...
117     'HorizontalAlignment','center', 'VerticalAlignment','bottom', ...
118     'FontSize', 10);
119
120 % Personalizare axă X
121 set(gca, 'XTickLabel', etichete_medie, 'XTickLabelRotation', 45,
    ↪ 'FontSize', 10);
122
123 % Etichete și titlu
124 ylabel('Media comenzii [kW]');
125 title('Efortul de control - media comenzii pentru metodele MPC');
126 grid on;
127
128 % Setează dimensiunea ferestrei
129 set(gcf, 'Position', [100 100 900 500]);
130
131 % Salvare imagine
132 saveas(gcf, 'MediaComanda_MPC_manual.png');
133
```

```

134 %% 6. Calcul și grafic pentru suprareglare (overshoot)
135 % Referința finală
136 y_ref = 37;
137
138 % Lista etichetelor și vectorilor de ieșire
139 etichete_suprareglare = {
140     'MPC_eu', ...
141     'Hc10', ...
142     'Hc20', ...
143     'Hc30', ...
144     'Hc100', ...
145     'Hp100', ...
146     'Hp20', ...
147     'Hp40', ...
148     'Hp60', ...
149     'Hp80', ...
150     'lambda10', ...
151     'lambda0_1', ...
152     'lambda1'
153 };
154
155 % Extragere și calcul suprareglare din structura 'out'
156 valori_suprareglare = [ ...
157     (max(out.iesire_MPC_eu) - y_ref) / y_ref * 100, ...
158     (max(out.iesire_MPC_hc10) - y_ref) / y_ref * 100, ...
159     (max(out.iesire_MPC_hc20) - y_ref) / y_ref * 100, ...
160     (max(out.iesire_MPC_hc30) - y_ref) / y_ref * 100, ...
161     (max(out.iesire_MPC_hc100) - y_ref) / y_ref * 100, ...
162     (max(out.iesire_MPC_hp100) - y_ref) / y_ref * 100, ...
163     (max(out.iesire_MPC_hp20) - y_ref) / y_ref * 100, ...
164     (max(out.iesire_MPC_hp40) - y_ref) / y_ref * 100, ...
165     (max(out.iesire_MPC_hp60) - y_ref) / y_ref * 100, ...
166     (max(out.iesire_MPC_hp80) - y_ref) / y_ref * 100, ...
167     (max(out.iesire_MPC_lambda10) - y_ref) / y_ref * 100, ...
168     (max(out.iesire_MPC_lambda_0_1) - y_ref) / y_ref * 100, ...
169     (max(out.iesire_MPC_lambda1) - y_ref) / y_ref * 100 ...
170 ];
171
172 % Creare grafic bară pentru suprareglare
173 figure;
174 b = bar(valori_suprareglare, 'FaceColor', [0.8 0.3 0.3]); % roșu
    ↪ deschis
175
176 % Afișare valori numeric deasupra barelor
177 text(1:length(valori_suprareglare), valori_suprareglare, ...
178     num2str(valori_suprareglare, '%.2f'), ...
179     'HorizontalAlignment', 'center', 'VerticalAlignment', 'bottom', ...
180     'FontSize', 10);
181
182 % Personalizare axă X
183 set(gca, 'XTickLabel', etichete_suprareglare, 'XTickLabelRotation', 45,
    ↪ 'FontSize', 10);
184

```

```

185 % Etichete și titlu
186 ylabel('Suprareglare [%]');
187 title('Suprareglare pentru metodele MPC');
188 grid on;
189
190 % Setează dimensiune figură
191 set(gcf, 'Position', [100 100 900 500]);
192
193
194 %% Timpul tranzitoriu
195
196 figure;
197 metode =
198     ↳ {'MPC_eu', 'Hc10', 'Hc20', 'Hc30', 'Hc100', 'Hp100', 'Hp20', 'Hp40', 'Hp60', 'Hp80', 'Hp100', 'Hp120', 'lambda1'};
199 t_tranzitor = [1100, 1000, 1000, 980, 950, 1000, 1050, 1050,
200     ↳ 1080, 1000, 1150, 920, 1000];
201 bar(t_tranzitor);
202 set(gca, 'xticklabel', metode);
203 yline(1200, '--k', '1200s', 'LabelHorizontalAlignment', 'right',
204     ↳ 'LabelVerticalAlignment', 'top');
205 ylabel('Durata regimului tranzitoriu [s]');
206 title('Compararea duratelor regimului tranzitoriu');
207 grid on;
208 saveas(gcf, 'durata_regimului_tranzitoriu_toate_metodele.png');
209
210 %% hp grafice
211 figure;
212 t = out.tout;
213 ref=37*ones(251,1);
214 subplot(2,1,1);
215 hold on; grid on;
216 plot(t, out.iesire_MPC_hp60, 'b', 'LineWidth', 1.5);
217 plot(t, out.iesire_MPC_hp80, 'r--', 'LineWidth', 1.5);
218 plot(t, out.iesire_MPC_hp100, 'c--', 'LineWidth', 1.5);
219 plot(t, out.iesire_MPC_hp120, 'r', 'LineWidth', 1.5); %best
220 plot(t, ref, 'k--', 'LineWidth', 1.5);
221 ylabel('Temperatura [°C]');
222 title('Ieșiri - MPC');
223 legend({'Hp60', 'Hp80', 'Hp100', 'Hp120', 'Referință'}, ...
224     ↳ 'Location', 'southeast');
225 subplot(2,1,2);
226 hold on; grid on;
227 plot(t, out.comanda_MPC_hp60, 'b', 'LineWidth', 1.5);
228 plot(t, out.comanda_MPC_hp80, 'r--', 'LineWidth', 1.5);
229 plot(t, out.comanda_MPC_hp100, 'c--', 'LineWidth', 1.5);
230 plot(t, out.comanda_MPC_hp120, 'r', 'LineWidth', 1.5);
231
232 xlabel('Timp [s]');
233 ylabel('Comandă [kW]');
234 title('Comenzi - MPC');
235 legend({'Hp60', 'Hp80', 'Hp100', 'Hp120'}, ...
236     ↳ 'Location', 'southeast');

```



```

234
235 set(gcf, 'Position', [100 100 900 600]);
236
237 %% hc grafice
238 figure;
239 t = out.tout;
240 ref=37*ones(251,1);
241 subplot(2,1,1);
242 hold on; grid on;
243 plot(t, out.iesire_MPC_hc30, 'b', 'LineWidth', 1.5);
244 plot(t, out.iesire_MPC_hc40, 'r--', 'LineWidth', 1.5);
245 plot(t, out.iesire_MPC_hc50, 'c--', 'LineWidth', 1.5);
246 plot(t, out.iesire_MPC_hc60, 'r', 'LineWidth', 1.5);
247 plot(t, out.iesire_MPC_hc120, 'g', 'LineWidth', 1.5);%best
248 plot(t, ref, 'k--', 'LineWidth', 1.5);
249 ylabel('Temperatura [°C]');
250 title('Ieşiri - MPC');
251 legend({'Hc30', 'Hc40', 'Hc50', 'Hc60', 'Hc120', 'Referință'}, ...
252        'Location', 'southeast');
253 subplot(2,1,2);
254 hold on; grid on;
255 plot(t, out.comanda_MPC_hc30, 'b', 'LineWidth', 1.5);
256 plot(t, out.comanda_MPC_hc40, 'r--', 'LineWidth', 1.5);
257 plot(t, out.comanda_MPC_hc50, 'c--', 'LineWidth', 1.5);
258 plot(t, out.comanda_MPC_hc60, 'r', 'LineWidth', 1.5);
259 plot(t, out.comanda_MPC_hc120, 'g', 'LineWidth', 1.5);
260 xlabel('Timp [s]');
261 ylabel('Comandă [kW]');
262 title('Comenzi - MPC');
263 legend({'Hc30', 'Hc40', 'Hc50', 'Hc60', 'Hc120'}, ...
264        'Location', 'southeast');
265
266 set(gcf, 'Position', [100 100 900 600]);
267 %% lambda grafice
268 figure;
269 t = out.tout;
270 ref=37*ones(251,1);
271 subplot(2,1,1);
272 hold on; grid on;
273 plot(t, out.iesire_MPC_lambda_0_1, 'b', 'LineWidth', 2);%best
274 plot(t, out.iesire_MPC_lambda1, 'r--', 'LineWidth', 2);
275 plot(t, out.iesire_MPC_lambda10, 'k--', 'LineWidth', 2);
276 plot(t, ref, 'b--', 'LineWidth', 1.5);
277 ylabel('Temperatura [°C]');
278 title('Ieşiri - MPC');
279 legend({'Lambda 0.1', 'Lambda 1', 'Lamnda 10', 'Referință'}, ...
280        'Location', 'southeast');
281 subplot(2,1,2);
282 hold on; grid on;
283 plot(t, out.comanda_MPC_lambda_0_1, 'b', 'LineWidth', 2);
284 plot(t, out.comanda_MPC_lambda1, 'r--', 'LineWidth', 2);
285 plot(t, out.comanda_MPC_lambda10, 'k--', 'LineWidth', 2);
286

```

```
287 xlabel('Timp [s]');
288 ylabel('Comandă [kW]');
289 title('Comenzi - MPC');
290 legend({'Lambda 0.1', 'Lambda 1', 'Lambda 10'}, ...
291        'Location', 'southeast');
292
293 set(gcf, 'Position', [100 100 900 600]);
```

Listing 4. Cod MATLAB- Comparare MPC-uri

### **Anexa 5. Cod MATLAB - MPC vs PID**

```
1  %schimbare de referinta
2  figure;
3  t = out.tout;
4  ref=out.ref_MPCs;
5  subplot(2,1,1);
6  hold on; grid on;
7  plot(t, out.iesire_MPC_eu2, 'r', 'LineWidth', 1.5);
8  plot(t, out.iesire_haalmansceref, 'b', 'LineWidth', 1.5);
9  plot(t, out.referinta_haalman, 'k--', 'LineWidth', 1.5);
10 plot(t, ref, 'k--', 'LineWidth', 1.5);
11 ylabel('Temperatura [°C]');
12 ylim([34 38]);
13 title('Ieșiri - MPC vs PID schimbare referință');
14 legend({'MPC', 'PID', 'Referință'}, ...
15        'Location', 'southeast');
16 subplot(2,1,2);
17 hold on; grid on;
18 plot(t, out.comanda_MPC_eu2, 'r', 'LineWidth', 1.5);
19 plot(t, out.comanda_haalmansceref, 'b', 'LineWidth', 1.5);
20
21 xlabel('Timp [s]');
22 ylabel('Comandă [kW]');
23 title('Comenzi - MPC vs PID schimbare referință');
24 legend({'MPC', 'PID'}, ...
25        'Location', 'southeast');
26
27 set(gcf, 'Position', [100 100 900 600]);
28 %% rejectia perturbatiei
29 figure;
30 t = out.tout;
31 ref=37*ones(301,1);
32 subplot(2,1,1);
33 hold on; grid on;
34 plot(t, out.iesire_MPC_eu1, 'r', 'LineWidth', 1.5);
35 plot(t, out.iesire_haalmanpert, 'b', 'LineWidth', 1.5);
36 plot(t, out.referinta_haalman, 'k--', 'LineWidth', 1.5);
37 plot(t, ref, 'k--', 'LineWidth', 1.5);
38 ylabel('Temperatura [°C]');
39 title('Ieșiri - MPC vs PID rejectarea perturbației');
40 legend({'MPC', 'PID', 'Referință'}, ...
41        'Location', 'southeast');
42 subplot(2,1,2);
```

```

43 hold on; grid on;
44 plot(t, out.comanda_MPC_eu1, 'r', 'LineWidth', 1.5);
45 plot(t, out.comanda_haalmanpert, 'b', 'LineWidth', 1.5);
46
47 xlabel('Timp [s]');
48 ylabel('Comandă [kW]');
49 title('Comenzi - MPC vs PID rejectarea perturbației');
50 legend({'MPC', 'PID'}, ...
51        'Location', 'southeast');
52
53 set(gcf, 'Position', [100 100 900 600]);
54 %% saturatie vs restrictii
55 figure;
56 t = out.tout;
57 ref=37*ones(501,1);
58 subplot(2,1,1);
59 hold on; grid on;
60 plot(t, out.iesire_MPC_eu, 'r', 'LineWidth', 2);
61 plot(t, out.iesire_haalman11, 'b', 'LineWidth', 1.5);
62 plot(t, out.referinta_haalman, 'k--', 'LineWidth', 1.5);
63 plot(t, ref, 'k--', 'LineWidth', 1.5);
64 ylabel('Temperatura [°C]');
65 title('Ieșiri - MPC vs PID saturație vs restricție');
66 legend({'MPC', 'PID', 'Referință'}, ...
67        'Location', 'southeast');
68 subplot(2,1,2);
69 hold on; grid on;
70 plot(t, out.comanda_MPC_eu, 'r', 'LineWidth', 1.5);
71 plot(t, out.comanda_haalman11, 'b', 'LineWidth', 1.5);
72
73 xlabel('Timp [s]');
74 ylabel('Comandă [kW]');
75 title('Comenzi - MPC vs PID saturație vs restricție');
76 legend({'MPC', 'PID'}, ...
77        'Location', 'southeast');
78
79 set(gcf, 'Position', [100 100 900 600]);
80 %% regim tranzitoriu
81 figure;
82 metode = {'MPC', 'PID'};
83 t_tranzitor = [1050, 4200];
84 bar(t_tranzitor);
85 set(gca, 'xticklabel', metode);
86 yline(4500, '--k', '4500s', 'LabelHorizontalAlignment', 'right',
87      ↪ 'LabelVerticalAlignment', 'top');
87 ylabel('Durata regimului tranzitoriu [s]');
88 title('Compararea duratelor regimului tranzitoriu');
89 grid on;
90
91 %% Calcul și grafic pentru suprareglare (overshoot)
92 % Referința finală
93 y_ref = 37;
94

```

```
95 % Lista etichetelor și vectorilor de ieșire
96 etichete_suprareglare = {
97     'MPC', ...
98     'PID'
99 };
100
101 % Extragere și calcul suprareglare din structura 'out'
102 valori_suprareglare = [ ...
103     (max(out.iesire_MPC_eu) - y_ref) / y_ref * 100, ...
104     (max(out.iesire_haalman) - y_ref) / y_ref * 100
105
106 ]
107 % Creare grafic bară pentru suprareglare
108 figure;
109 b = bar(valori_suprareglare, 'FaceColor', [0.8 0.3 0.3]); % roșu
    ↪ deschis
110
111 % Afișare valori numeric deasupra barelor
112 text(1:length(valori_suprareglare), valori_suprareglare, ...
113     num2str(valori_suprareglare, '%.2f'), ...
114     'HorizontalAlignment', 'center', 'VerticalAlignment', 'bottom', ...
115     'FontSize', 10);
116
117 % Personalizare axă X
118 set(gca, 'XTickLabel', etichete_suprareglare, 'XTickLabelRotation', 45,
    ↪ 'FontSize', 10);
119
120 % Etichete și titlu
121 ylabel('Suprareglare [%]');
122 title('Suprareglare pentru metodele MPC');
123 grid on;
124
125 % Setează dimensiune figură
126 set(gcf, 'Position', [100 100 900 500]);
127
128 %% Medie comanda
129
130 % Valori deja calculate ale mediei comenzilor
131 valori_medie_comanda = [
132     mean(out.comanda_MPC_eu), ...
133     mean(out.comanda_haalman), ...
134
135 ];
136
137 etichete_medie = {
138     'MPC', ...
139     'PID' ...
140
141 };
142
143 % Creare grafic bară pentru media comenzii
144 figure;
145 b = bar(valori_medie_comanda, 'FaceColor', [0.3 0.6 0.4]); %
    ↪ verde-albăstrui
```

```

146
147 % Afişare valori numeric deasupra barelor
148 text(1:length(valori_medie_comanda), valori_medie_comanda, ...
149     num2str(valori_medie_comanda','%.2f'), ...
150     'HorizontalAlignment','center', 'VerticalAlignment','bottom', ...
151     'FontSize', 10);
152
153 % Personalizare axă X
154 set(gca, 'XTickLabel', etichete_medie, 'XTickLabelRotation', 45,
    ↪ 'FontSize', 10);
155
156 % Etichete şi titlu
157 ylabel('Media comenzii [kW]');
158 title('Efortul de control - media comenzii pentru metodele MPC');
159 grid on;
160
161 % Setează dimensiunea ferestrei
162 set(gcf, 'Position', [100 100 900 500]);
163
164 % Salvare imagine (opţional)
165 saveas(gcf, 'MediaComanda_MPC_manual.png');
166
167 %% MSE
168 referinta=37*ones(501,1);
169 % Valori MSE pentru metodele testate
170 valori_MSE = [...
171     mean((referinta - out.iesire_MPC_eu).^2),...
172     mean((referinta - out.iesire_haalman).^2)
173 ];
174
175 % Etichetele corespunzătoare
176 etichete_MSE = {...
177     'MPC', ...
178     'PID', ...
179
180
181     };
182
183 % Creare grafic bară
184 figure;
185 b = bar(valori_MSE, 'FaceColor', [0.3 0.4 0.8]); % culoare albastru
    ↪ deschis
186
187 % Afişare valori numeric deasupra barelor
188 text(1:length(valori_MSE), valori_MSE, num2str(valori_MSE','%.4f'), ...
189     'HorizontalAlignment','center', 'VerticalAlignment','bottom', ...
190     'FontSize', 10);
191
192 % Personalizare axă X
193 set(gca, 'XTickLabel', etichete_MSE, 'XTickLabelRotation', 45,
    ↪ 'FontSize', 10);
194
195 % Etichete şi titlu

```

```
196 ylabel('MSE');
197 title('MSE pentru metodele de acordare PID');
198
199 % Activare grilă
200 grid on;
201
202 % Setează dimensiune figură
203 set(gcf, 'Position', [100 100 900 500]);
```

Listing 5. Cod MATLAB- MPC vs PID