



Modelarea și Controlul Avansat al Vitezei a unui Sistem de Propulsie Hibriz

LUCRARE DE LICENȚĂ

Absolvent: **Beniamin-Ionică CIOBAN**

Coordonator științific: **As.Dr.Ing. Vlad Mihai MIHALY**

2024

Cuprins

Capitolul 1	Introducere	1
1.1	Context general	1
1.2	Obiective	2
1.2.1	Obiective generale	2
1.2.2	Obiective specifice	2
1.3	Specificații	3
1.3.1	Funcționalități	3
1.3.2	Performanțe	3
1.3.3	Limitări	3
Capitolul 2	Studiu bibliografic	4
2.1	Controlul Convertoarelor Conectate la Rețea.	5
2.1.1	Controlul <i>Virtual Synchronous Generator</i> (VSG)	6
2.1.2	Sinteza FCS-MPC	7
2.2	Regulator LFC Bazat pe o Rețea Neuronală Optimală	8
2.3	Controlul Sistemelor Neliniare	10
2.3.1	Control Lyapunov Bazat pe Rețele Neuronale	10
2.3.2	Sinteză Robustă de Structură Fixă pentru Sisteme Neliniare . . .	13
Capitolul 3	Analiză și Proiectare	15
3.1	Modelarea sistemului de propulsie	15
3.1.1	Modelarea Motorului Electric	15
3.1.2	Modelarea bateriei	17
3.1.3	Modelarea motorului pe combustie internă	19
3.1.4	Modelarea dinamicii	21
3.2	Sliding Mode Control	22
3.2.1	Modelul Sistemului și Proiectarea Suprafeței de Sliding Mode . .	23
3.2.2	Faza de <i>Atingere</i>	24
3.2.3	Legile de Atingere	25
3.2.4	Proiectarea Regulatorului Echivalent	25
3.2.5	Fenomenul de Chattering	26
3.2.6	Ajustarea Parametrului de Control	28
3.3	Proiectarea reguletoarelor bazate pe norma H_∞	29
3.3.1	Introducerea Normei H_∞	29
3.3.2	Proiectarea Bazată pe <i>Mixed Sensivity</i>	30
3.3.3	Concluzii și Specificații de Proiectare	32
3.4	Proiectarea Unui Regulator Neuronal pe Baza Modelului NARMA-L2 . .	35
3.4.1	Modelul NARMA	36
3.4.2	Modelul NARMA-L2	36
3.4.3	Îmbunătățirea Performanței Regulatorului	37
Capitolul 4	Implementare și Testare	39
4.1	Impelementarea Modelelor pentru Motoare	39
4.1.1	Motorul Electric	39
4.1.2	Motorul pe Combustie Internă	40

4.2	Implementarea Modelului Bateriei	40
4.3	Implementarea Modelului Dinamic	42
4.4	Controlul Sistemului	42
4.4.1	Liniarizarea Modelului	44
4.4.2	Implementarea SMC	45
4.4.3	Implementarea Regulatorului Robust pe Baza Normei H_∞	47
4.4.4	Implementarea Regulatorului Neuronal NARMA-L2	49
4.4.5	Compararea Rezultatelor Obținute	51
Capitolul 5 Concluzii		53
Bibliografie		55
Anexa A Secțiuni relevante din cod		58

Capitolul 1. Introducere

1.1. Context general

Expansiunea tehnologică accelerată și contextul general actual a adus la necesitatea folosirii metodelor avansate de control din ce în ce mai des în industria modernă pentru optimizarea performanței sistemelor și proceselor. Caracteristicile acestora includ modelarea proceselor și identificarea parametrilor, predicția comportamentului procesului folosind modelul sistemului și evaluarea criteriului de performanță sub constrângeri de proces. [1].

Câteva dintre tehnici ce merită a fi amintite sunt: control adaptiv, control multi-variabil, control predictiv, control fuzzy, control robust, control optimal și control bazat pe rețele neuronale.[1].

Toate aceste metode avansate de control vin de obicei în îmbunătățirea performanțelor proceselor complexe industriale unde metodele clasice de control nu pot fi folosite fie din cauza dinamicii neliniare sau procesul este multivariabil. De asemenea, soluționarea cerințelor de performanță înaltă se reduce de cele mai multe ori la o problemă de optimizare, de exemplu cum ar fi minimizarea timpului de răspuns sau reducerea variației și îmbunătățirea calității produsului, toate aceste cerințe sunt ilustrate ierarhic în figura 1.1.

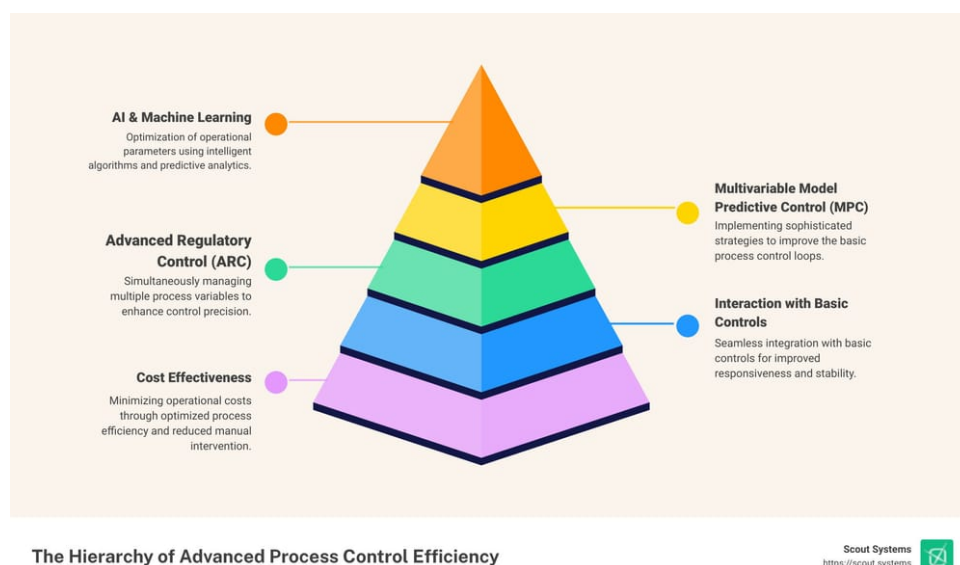


Figura 1.1: Ierarhia eficienței metodelor avansate de control[2].

Abordările de acest tip sunt cel mai des întâlnite în următoarele industrii unde măsurile de performanță trebuie îndeplinite cu strictețe. În industria chimică și petrochimică în care controlul avansat al reactoarelor este esențial pentru a maximiza randamentul și a minimiza produsele secundare nedorite [3]. O altă industrie este cea a energiei unde centralele electrice trebuie să răspundă rapid la fluctuațiile ce pot apărea pe rețea pentru a menține stabilitatea acesteia. De asemenea un alt element important din acest domeniu

este controlul turbinelor eoliene deoarece performanța acestora poate varia semnificativ în funcție de condițiile de vânt. Astfel utilizarea controlului avansat permite optimizarea funcționării turbinelor prin ajustarea parametrilor în timp real [4]. Ca un ultim exemplu ce merită a fi menționat pe lângă cel domeniul tehnologiei militare este industria auto cu sistemele avansate de asistență la conducere și funcțiile autonome ce devin din ce în ce mai importante. În acest caz controlul predictiv și integrarea tehnicilor de machine learning permite dezvoltarea funcțiilor de asistență și autonomie în vehicule astfel îmbunătățind siguranța și confortul conducătorilor auto [5]. Tot în acest domeniu, dar de data asta pe partea de producție, o problemă ce intervine în optimizarea liniilor de asamblare care trebuie să fie extrem de eficiente și adaptabile la variațiile ce apar în producție. Prin urmare utilizarea metodelor avansate de control devine aproape indispensabilă deoarece permite gestionarea în mod eficient a acestor probleme, iar controlul adaptiv de exemplu ajustează procesele de asamblare în timp real pentru a răspunde cerințelor de producție și pentru a minimiza timpii de oprire.

1.2. Obiective

Având în vedere contextul general prezentat, în continuare focusul temei proiectului ce urmează a fi prezentat în capitolele 3 și 4, în primă fază pe modelarea matematică a sistemului de propulsie a unui vehicul hibrid (motor DC și motor pe combustie internă) apoi controlul vitezei mașinii utilizând mai multe metode avansate de control printre ele și controlul bazat pe rețele neuronale, iar în ultimă instanță simularea și compararea rezultatelor obținute în urma aplicării tehnicilor de control.

1.2.1. Obiective generale

- a.) **Modelarea matematică și sinteza a unui sistem de propulsie pentru un vehicul hibrid:** se urmărește conceperea și modelarea sistemului de propulsie hibrid
- b.) **Conceperea și implementarea algoritmilor de control avansat:** acest obiectiv urmărește documentarea și implementarea algoritmilor de control ce vor fi adaptați pentru sistemul de propulsie și în final de a controla viteza mașinii.
- c.) **Simularea și compararea rezultatelor obținute în urma aplicării tehnicilor de control avansat pe sistemul propus:** aici se urmărește simularea interdependentă a fiecărui algoritm aplicat pe sistemul dat și interpretarea/analiza performanțelor obținute în urma tuturor simulărilor, compararea rezultatelor.

1.2.2. Obiective specifice

- a.) **Dezvoltarea modelelor pentru fiecare componentă în Simulink (Motor DC, Motor CI, Driver Motor DC, Baterie, Cinematică/Dinamică:** pe baza ecuațiilor fizice și principiilor de funcționare a fiecărei componente se concepe modelul în mediul MATLAB/Simulink.
- b.) **Identificarea sistemului și liniarizarea acestuia în jurul unui punct optim de funcționare:** deși majoritatea componentelor au o caracteristică liniară, totuși modelul bateriei și al dinamicii mașinii introduc neliniarități în model de aceea este nevoie de o identificare și o liniarizare.
- c.) **Proiectarea și sinteza unui regulator optimal utilizând norma H-infin**
- d.) **Proiectarea și sinteza unui regulator bazat pe SMC (engl. *Sliding Mode Controller*)**
- e.) **Proiectarea și sinteza unui regulator neuronal ce va funcționa în regim online:**

conceperea unui regulator NARMA-L2 ce este capabil să se antreneze online.

1.3. Specificații

Dezvoltarea modelului cât și proiectarea reguletoarelor menționate în subsecțiunea 1.2.2 necesită cunoașterea și stăpânirea unui aparat matematic sofisticat cât și o bună înțelegere a noțiunilor teoretice ce stau la baza conceperii unei rețele neuronale, antrenarea acestora și evaluare performanțelor obținute în urma antrenării, validării și testării. De asemenea un pas important este înțelegerea problemei de optimizare și a tehnicilor ce implică acest aspect, de aceea în această subsecțiune voi enunța pe scurt câteva dintre funcționalitățile și nivele de performanță a proiectului.

1.3.1. Funcționalități

Parametrizarea corectă a modelului sistemului de propulsie: Pentru a avea o caracteristică cât mai apropiată cu cea din realitate a unui astfel de sistem avem nevoie de o parametrizare a fiecărei componente în parte, parametrii care să aibă sens din punct de vedere fizic și ce respectă constrângerile ce pot apărea din dinamica mașinii, respectiv din legile fizicii.

Reglarea vitezei mașinii: se dorește îmbunătățirea performanțelor obținute față de controlul clasic al vitezei prin prisma metodelor avansate de control, respectiv gestionarea incertitudinilor ce pot apărea fie din cauza nivelului de descărcare a bateriei sau a dinamicii mașinii ce se poate schimba în funcție de mediu.

Control robust al sistemului: prin aplicarea tuturor metodelor de control se dorește a oferi un control robust pentru sistemul dat și a acoperii o clasă de reguletoare PID cât mai generală pentru a putea face față oricărei schimbări de comportament în sistem.

1.3.2. Performanțe

În urma aplicării tuturor metodelor de control menționate anterior se urmărește îmbunătățirea timpilor de răspuns, eliminarea suprareglajului și mărirea lățimii de bandă pentru sistem.

1.3.3. Limitări

Sistemul propus este unul simplificat astfel s-ar putea să nu se pună în evidență într-un mod just avantajele folosirii unor astfel de metode. O altă limitare ar fi implementarea fizică a unor astfel de reguletoare deoarece ar implica un cost de producție ridicat având în vedere structura complicată și ar necesita procesoare cu putere de calcul ridicat.

Capitolul 2. Studiu bibliografic

La începutul anilor 1960, apariția principiului de maxim al lui Pontryagin, cunoscut și sub numele de principiul maxim optim, programarea dinamică, filtrul Kalman și descrierea spațiului stărilor au marcat nașterea teoriei moderne de control. În ultimile trei decenii, metodele moderne de control au fost îmbunătățite și aplicate într-o serie de procese industriale. Din anii 1980, domeniile sistemelor de control neliniar, control bazat pe norma H_∞ , control adaptiv, sisteme de evenimente discrete și control bazat pe inteligență artificială au primit atenție și au fost dezvoltate aplicații [6].

Åström și Wittenmark menționează 12 martie 1959 ca fiind data în care s-a realizat primul control digital pentru optimizarea unui punct de operare economic la o rafinărie din Texaco, mai exact rafinăria Port Arthur [7].

Una dintre cele mai des folosite metode de control avansat este Controlul Predictiv pe Model (MPC) a cărei dezvoltare a început în industria proceselor mult înainte ca fundamentele teoretice să fie înțelese pe deplin. Aceasta metodă de control este cea mai des folosită în procesele din industria chimică și petrochimică încă din anii 1980, dar recent se folosește și în electronica de putere și pentru modele de echilibrare a sistemelor de putere. [8].

Primele generații ale MPC au fost dezvoltate în anii 1970 și 1980. Astfel au apărut algoritmi precum *Identification and Command* (IDCOM) și *Dynammic Matrix Control* (DMC) ce au fost dezvoltați pentru a optimiza procesele industriale aducând beneficii economice semnificative. Ulterior tehnologiile MPC au fost îmbunătățite pentru a gestiona probleme mai complexe precum manipularea constrângerilor și toleranța la defecte [9][10].

În anii 1990 piața MPC a cunoscut o creștere rapidă, cu fuziuni și achiziții majore. Astfel printre companiile ce s-au remarcat a fi lideri de piață se numără AspenTech și Honeywell ce au achiziționat tehnologii precum DMC și Setpoint. În această perioadă, multe alte tehnologii MPC au fost dezvoltate și aplicate pe scară largă, tehnologii ce sunt evidențiate în figura 2.1.. Astăzi, accentul în dezvoltarea metodelor avansate de control nu este doar pe îmbunătățirea algoritmilor, ci și pe simplificarea și accelerarea etapelor de dezvoltare. Se pune un mare accent pe crearea unor sisteme mai inteligente și pe reducerea timpului și resurselor necesare pentru dezvoltarea aplicațiilor [10].

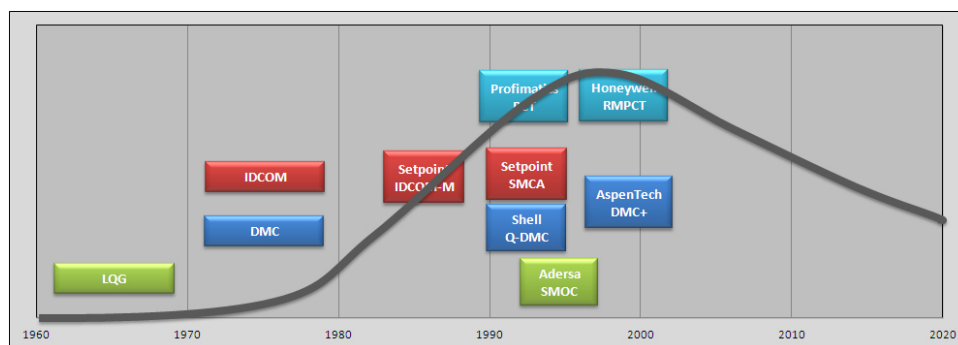


Figura 2.1: Evoluția tehnologiilor MPC.

În secțiunile următoare din acest capitol se va aduce în prim plan stadiul actual în care se află evoluția acestor metode de control în diverse industrii, dar și în cercetare. Toate aceste lucrări științifice ce vor fi prezentate ulterior stau la baza inspirației temei de proiect și oferă soluții pentru gestionarea sistemelor complexe de tipul celor din figura 2.2

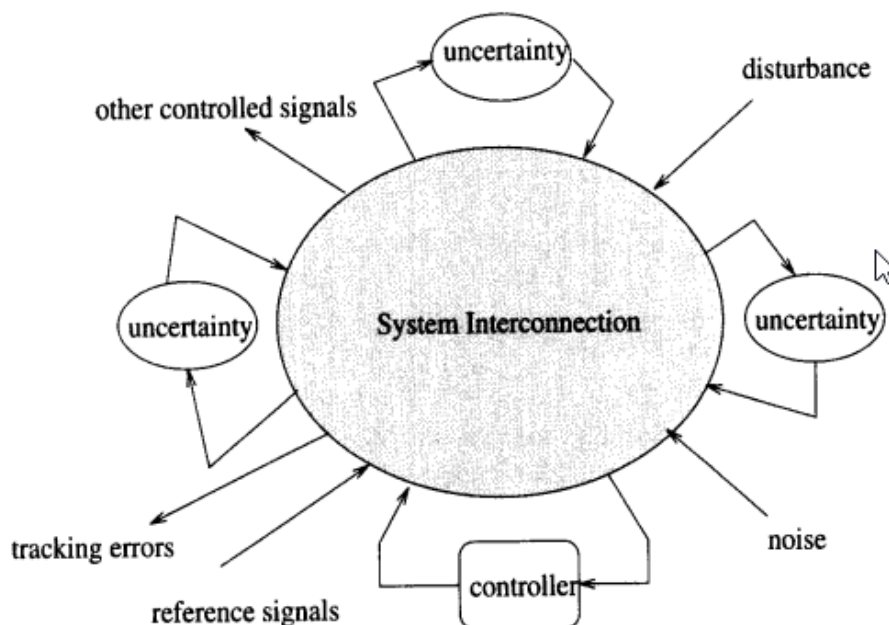


Figura 2.2: Schema generală de interconexiune a sistemelor cu incertitudini.

2.1. Controlul Convertoarelor Conectate la Rețea.

Convertoarele conectate la rețea (eng. *Grid Connected Converters - GCC*) joacă un rol esențial în sistemele de energie electrică moderne, în special în contextul energiei regenerabile și a distribuției acesteia. Aceste dispozitive permit interfațarea eficientă a surselor de energie regenerabilă, cum ar fi panourile solare sau turbinele eoliene, în rețelele electrice tradiționale [11].

Convertoarele sunt dispozitive electronice de putere care convertesc energia generată de sursele de energie regenerabilă în energie electrică compatibilă cu rețeaua publică și funcționează după următoarele principii: Conversia DC-AC, Sincronizarea cu Rețeaua, Reglarea Tensiunii și Curentului.

În capitolul 8.4.3 din [12] pentru a controla un astfel de convertor este propusă o metodă de control predictiv pe model bazat pe setul finit de stări (FCS-MPC) și pe VSG (engl. *Virtual Synchronous Generator*). Diagrama de control propusă este cea din figura 2.3 ce este compusă din: FCS-MPC, controlul VGS și părțile de comandă pe curent. FCS-MPC folosește modelul matematic al sistemului să prezică parametrii acestuia și să determine punctul optim în care convertorul trebuie să-și schimbe stările. Pentru a reazliza schme FCS-MPC cu atât controlul tensiunii, cât și al curentului, trebuie furnizată o comandă de curent, ce este generată din partea de comandă în curent.[13].

Un principal avantaj al acestei metode față de celelate tehnici bazate pe PWM îl constituie răspunsul dinamic rapid. În plus existența controlului buclei exterioare a

tensiunii în axele $\alpha\beta$ poate ajuta la tratarea neliniarităților și a distorsiunii în forma de undă a curentului și tensiunii sub efectul perturbațiilor tranzitorii [14].

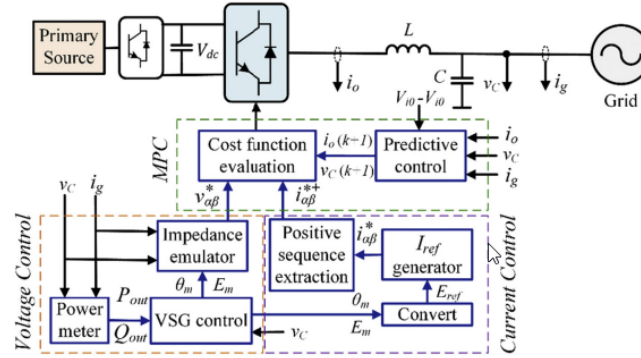


Figura 2.3: Schema de control FCS-MPC bazată pe control VSG [14].

2.1.1. Controlul *Virtual Synchronous Generator* (VSG)

Prin controlul unui generator sincron virtual (VSG) se dorește să se urmărească comanda de putere distribuită în timp ce oferă o inerție virtuală, compensare, ca să diminueze fluctuațiile/deviațiile ce pot apărea în frecvența rețelei. Asta se poate realiza cu o structură de control VSG, schemă ce este prezentată în figura 2.4 ce este compusa din *Swing equation*, *Governor model*, *Q droop*. Semnalele P_o , P_{out} , P_{gov} , Q_o , Q_{out} , V_{out} , ω_m , $\hat{\omega}_g$, E^* și θ^* reprezintă: comanda puterii active, ieșirea puterii active, comanda puterii reactive, puterea de pe arbore de la *Governor*, comanda puterii reactive, ieșirea puterii reactive, ieșirea în amplitudine a tensiunii, frecvența mecanică a rotorului, frecvența de ieșire măsurată de PLL (engl. *phase-locked loop*), magnitudinea de referință a tensiunii și referința unghiului de putere. Inertia virtuală este emulată în blocul *Swing equation* cu ajutorul formulei de balans: $P_{gov} - P_{out} + D(\omega_m - \hat{\omega}_g) = J\omega_m \frac{d\omega_m}{dt}$, unde J este momentul de inerție iar D este factorul de amortizare produs de amortizor [15]

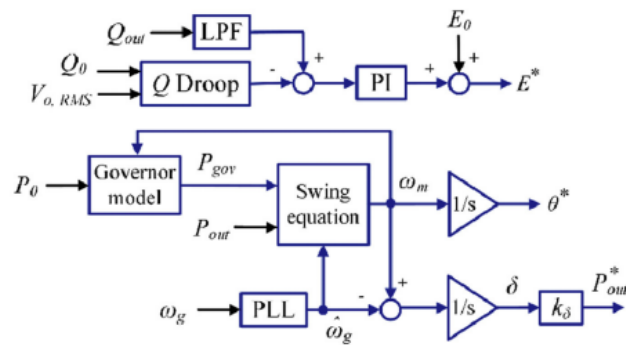


Figura 2.4: Schema de control VSG .

Întrucât valoarea K a controlului propus nu este limitat de valorile fizice ale sistemului, deoarece generatorul sincron este unul virtual, poate fi selectat în mod arbitrar pentru a optimiza performanța sistemului, P_{out} poate fi scris ca în ecuația 2.1

$$P_{\text{out}} = k_{\delta} \delta, \quad (2.1)$$

unde k_{δ} este constanta de sincronizare optimizată, iar δ poate fi determinată cu:

$$\delta = \int (\omega_m - \hat{\omega}_g) dt. \quad (2.2)$$

2.1.2. Sinteza FCS-MPC

Pentru sinteza unui astfel de regulator trebuie să se definească următoarele aspecte: un model matematic pentru convertor, definirea unei funcții de cost și construirea unui *Filter Resonance Damping*.

• Modelul convertorului conectat la rețea GCC

Pentru a realiza modelul matematic al GCC se consideră n circuit echivalent al GCC cu un inductor și condensator de filtrare al unui filtru LC este prezentat, acest circuit oferă un model pentru prezicerea tensiunii și a curentului de ieșire al sistemului, care poate fi descris în cadrul $\alpha\beta$ prin dinamica condensatorului și a inductanței așa cum reiese din ecuațiile (3.31)(2.4) .

$$C \frac{d\mathbf{v}_c}{dt} = \mathbf{i}_o - \mathbf{i}_g, \quad (2.3)$$

$$L \frac{d\mathbf{i}_g}{dt} = \mathbf{v}_c - \mathbf{v}_o, \quad (2.4)$$

unde $\mathbf{v}_c = [v_{c,\alpha} \ v_{c,\beta}]^T$, $\mathbf{v}_o = [v_{o,\alpha} \ v_{o,\beta}]^T$, $\mathbf{i}_o = [i_{o,\alpha} \ i_{o,\beta}]^T$ și $\mathbf{i}_g = [i_{g,\alpha} \ i_{g,\beta}]^T$ sunt tensiunea pe condensator, tensiunea de ieșire, curentul convertorului și curentul rețelei, respectiv.

• Definirea funcției de cost

Contrar schemelor de control clasice, prezența funcției de cost permite FCS-MPC să ia în considerare un număr de obiective de control și să controleze simultan diferite variabile de stare. Prin urmare, este foarte important să se definească funcția de cost în mod corespunzător. Pentru controlul GCC, trebuie luate în considerare variabile importante pentru a face ca întregul sistem să funcționeze corect. Sistemul de control GCC este setat să urmărească referințele de tensiune și curent simultan. Acest lucru poate fi realizat prin definirea unei funcții de cost după cum urmează:

$$g = k_v (v_{c,\alpha\beta,\text{pu}}(k+1) - v_{\alpha\beta,\text{pu}}^*)^2 + k_i (i_{o,\alpha\beta,\text{pu}}(k+1) - i_{\alpha\beta,\text{pu}}^*)^2 + g_m, \quad (2.5)$$

unde, $v_{\alpha\beta}^* = \begin{bmatrix} v_{\alpha}^* \\ v_{\beta}^* \end{bmatrix}$ este vectorul de referință al tensiunii pe condensator, $i_{\alpha\beta}^* = \begin{bmatrix} i_{\alpha}^* \\ i_{\beta}^* \end{bmatrix}$ este vectorul de referință al curentului prin inductor, $v_{c,\alpha\beta}(k+1) = \begin{bmatrix} v_{c,\alpha} \\ v_{c,\beta} \end{bmatrix}$ este tensiunea prezisă pe condensator și $i_{o,\alpha\beta}(k+1) = \begin{bmatrix} i_{f,\alpha} \\ i_{f,\beta} \end{bmatrix}$ este curentul prezis prin inductor, iar subscriptul pu indică valoarea per-unit. g_m este o constrângere inclusă în funcția de

cost pentru a se asigura că tensiunea pe partea de curent continuu nu devine niciodată negativă.

• Filtru

Pentru a asigura funcționarea corectă, GCC necesită un filtru de ieșire inductiv. Cu toate acestea, pentru a îmbunătăți calitatea formelor de undă, este preferat de obicei un filtru LC . Totuși, prezența frecvenței de rezonanță trebuie luată în considerare deoarece FCS-MPC are o frecvență de comutare variabilă. Aceasta ar putea genera armonici la frecvența de rezonanță a filtrului LC la un anumit punct de operare. Cu toate acestea, acest lucru va rezulta în rezonanță, dacă doar curentul de ieșire al invertorului $i_g(s)$ este controlat direct, în timp ce tensiunea pe condensator $v_c(s)$ este controlată indirect de curent și, prin urmare, este foarte dependentă de impedanțele filtrului.

Metoda propusă FCS-MPC utilizează un sistem multivariat pentru a controla simultan atât $v_c(s)$ cât și $i_g(s)$. Acest lucru este diferit de controlul clasic în cascadă al tensiunii și curentului, unde timpul de stabilizare al buclei interioare trebuie să fie semnificativ mai rapid decât timpul de stabilizare al buclei exterioare. Prin urmare, la frecvența de rezonanță unde $v_c(s)$ și $i_g(s)$ oscilează cu aceleași dinamici, controlul buclei exterioare nu este suficient de rapid pentru a oferi un efect de amortizare. Contrar, controlul multivariat al FCS-MPC este efectuat la aceeași lățime de bandă. Acest lucru asigură că reglarea $v_c(s)$ poate atenua distorsiunea cauzată de armonicile $i_g(s)$ și controlul $i_g(s)$ este capabil să reducă perturbarea cauzată de armonicile $v_c(s)$. Prin urmare, controlând tensiunea filtrului LC și curentul în același timp, răspunsul frecvenței filtrului nu mai este influențat doar de impedanța sa. Astfel, oscilația energetică rezonantă necontrolată între inductanțe și capacitate este efectiv amortizată.

2.2. Regulator LFC Bazat pe o Rețea Neuronală Optimală

În articolul [16] autorul propune o metodă LFC (*engl. Load Frequency Controller*) pentru a controla o structură PSN (*engl. Power System Network*) pe baza unei tehnici ANN (*engl. Artificial Neural Network*). Datele de antrenare pentru rețeaua neuronală propusă sunt simulate folosind mediul Simulink în care s-a dezvoltat un LFC așa cum se vede și în figura 2.5. Pentru a rezolva impedimentele majore ale unei astfel de rețele neuronale, autorul propune folosirea unui algoritm metaeuristic de optimizare, mai exact PSO (*engl. Particle Swarm Optimization*). Această optimizare este împărțită în două secțiuni:

1. Calcularea corectă a rețelei neuronale și optimizarea neuronilor inițiali a modelului, obținuți din regresia nodurilor rețelei.
2. Minimizarea erorii globale de antrenare a modelului.

În consecință răspunsul în frecvență al sistemului LFC ce utilizează rețeaua neuronală optimizată cunoaște o îmbunătățire semnificativă față de cel al metodelor convenționale pentru regulatoarele pe baza ANN și PID. Acest lucru devine fiabil datorită timpului rapid de procesare. În plus, este valabilă îmbunătățirea performanței PSN-ului cu mai multe arii supus unei stări de defect. Diagrama schematică ce ilustrează antrenarea modelului ANN bazat pe algoritmul PSO este prezentată în figura 2.6 în timp ce principalii parametri ai rețelei optime sunt detaliați în Tabelul.2.1. Prin urmare blocul de optimizare PSO este utilizat pentru găsirea celei mai bune topologii a rețelei neuronale, iar algoritmul testează per ansamblu numărul de neuroni din stratul ascuns progresiv.

Rezultatul algoritmului demonstrează că singurul strat al rețelei cu 23 de noduri,

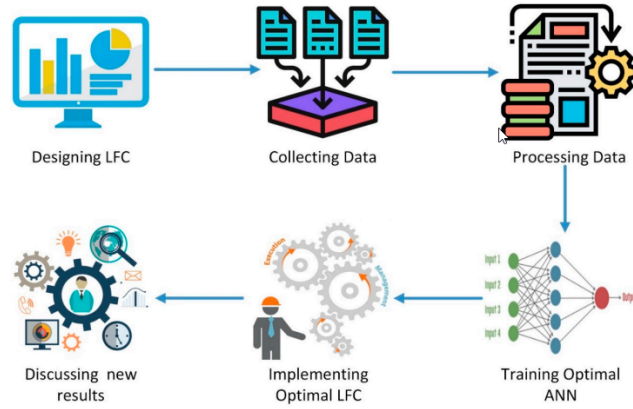


Figura 2.5: Procedura de implementare[16].

Tabela 2.1: Parametrii unei rețele neuronale artificiale (ANN) optime.

Tipuri	Parametri
Noduri în stratul de intrare al ANN	2
Noduri în stratul de ieșire al ANN	1
Noduri în stratul ascuns al ANN	23
Numărul de neuroni din ANN	93
Dimensiunea roiului de PSO	50
Ponderarea inerțială a PSO	0.75
Coeficientul cognitiv al PSO	1.15

care are două intrări și o ieșire, are cea mai mică eroare la antrenare, de asemenea având numărul optim de 93 de neuroni. O abordare hibridă bazată pe PSO și ANN este apoi dezvoltată în a doua modificare pentru a afla ponderile optime pentru modelul ANN odată ce a fost stabilită topologia rețelei. Când ponderile inițiale sunt corectate, s-a dovedit că acestea îmbunătățesc predicția de ieșire a modelului.

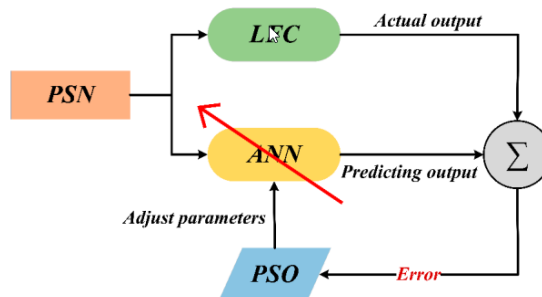


Figura 2.6: Procedura de implementare [16].

Din cauza varietății condițiilor de funcționare a rețelei de alimentare, în urma antrenării prin metode clasice a modelului ANN, acesta nu va fi capabil să prezică abaterile mari în nivelul frecvenței atunci când cererea de sarcină are o schimbare bruscă. Ca o posibilă soluție pentru problema enunțată autorii din [17] propun o metodă de control robust pentru a antrena ANN, mai exact folosesc tehnica de μ -*synthesis* pentru a obține o performanță bună cu o structură simplă de ANN. Regulatorul este simulat pentru un PSN cu două arii, ce este asigurat de două turbine cu abur pentru generarea de energie electrică.

Rezultatele demonstrează ca această metodă propusă are un timp de răspuns rapid pentru nivelul de frecvență cerut sub diverse condiții de operare și diferite solicitări de sarcină. Acest lucru vine în urma reducerii suprareglajului frecvenței.

Bazat pe aceeași idee autorii în articolul [18] propun un model de rețea neuronală dinamică pe funcții sigmoide, funcții standard fuzzy și wavelet pentru adaptarea LFC. La fel ca în cazurile precedente ei au simulat un sistem a unei turbine aplicat pe un PSN cu două arii pentru a testa algoritmul de control propus. Rezultatele bazate pe acest studiu comparativ demonstrează că controlerul ANN bazat pe dinamica wavelet este mai robust și ajustează deviațiile de la abaterea standard atunci când apar perturbații în rețea.

În contrast, autorii din [19] propun un ANN optimal pentru proiectarea robustă a regulatorului LFC folosind tehnica denumită (engl. *sliding mode*). Această metodă este folosită pentru a actualiza ponderile inițiale a rețelei neuronale, apoi ca și în celelalte cazuri regulatorul este aplicat pe un PSN, dar de data aceasta alimentat cu două surse generatoare diferite: turbină pe abur și una eoliană. Rezultatele ilustrează efectul adecvat al turbinei eoliene atunci când este conectată la rețeaua electrică, pe baza testelor de cerere de sarcină, și validează această metodă propusă sub diverse condiții de testare.

2.3. Controlul Sistemelor Neliniare

Teoria controlului sistemelor neliniare acoperă o clasă largă de sisteme ce nu se supun principiului superpoziției și se aplică sistemelor din lumea reală deoarece în mare parte acestea prezintă neliniarități. Astfel aparatul matematic devine riguros și tehnicile matematice sunt din ce în ce mai puțin generale, de obicei se aplică pe clase restrânse de sisteme. În continuare se va aduce în prim plan două lucrări de cercetare ce își propun să rezolve problemele controlului sistemelor ce prezintă neliniarități puternice (engl. *hard nonlinear control problems*).

2.3.1. Control Lyapunov Bazat pe Rețele Neuronale

În lucrarea [20] autorii noi metode de învățare a principiilor de control și a rețelei neuronale bazată pe funcții Lyapunov pentru probleme de control neliniar, cu garanție demonstrabilă de stabilitate. Algoritmul propus poate fi ilustrat schematic ca în figura 2.7 și constă:

- Definirea unui învățător (engl. *learner*) ai cărui sarcină este să găsească legea de control și parametrii funcției neuronale Lyapunov astfel încât să minimizeze riscul Lyapunov.
- Definirea unui falsificator (engl. *falsifier*) ce va prelua funcția de control învățată și funcția Lyapunov neuronală de la (*learner*) și verifică dacă există un vector de stare care încalcă condițiile Lyapunov. Mai simplu spus găsește contraexemple ca învățătorul (engl. *learner*) să găsească soluția cât mai rapid.

- Algoritmul se opreste atunci când nu se mai găsește niciun contraexemplu, caz în care sistemul neliniar controlat este demonstrabil stabil.

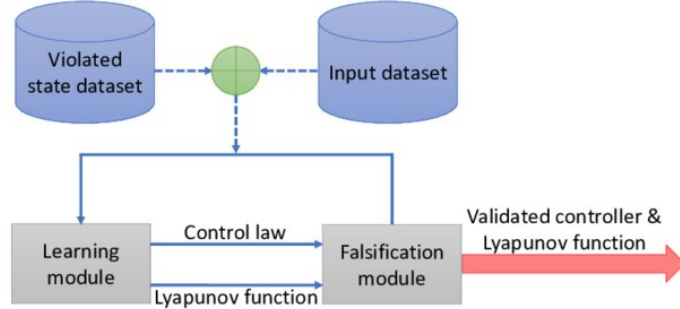


Figura 2.7: Învățarea neuronală a controlerului și a funcției Lyapunov.

Această abordare simplifică semnificativ procesul de sinteză a regulatorului Lyapunov și garantează o corectitudine integrală (*engl. end-to-end correctness*) și acoperă o gamă mai largă decât metodele deja existente cum ar fi LQR și SOS/SDP (*engl. Sum of square optimization*). Pentru a satisface toate constrângerile Lyapunov, să asigure stabilitate sistemului, autorii descriu în lucrare cum algoritmul să învețe atât funcția de control cât și cea Lyapunov împreună și propun algoritmul din Algorithm 1.

Definiția 1 (Constrângeri de Falsificare Lyapunov). Fie V o funcție candidat Lyapunov pentru un sistem dinamic definit de f_u în spațiul de stare D . Fie $\varepsilon \in \mathbb{Q}^+$ un parametru constant mic care limitează eroarea numerică tolerabilă. Constrângerea de falsificare Lyapunov este următoarea formulă logică de prim ordin pentru numere reale:

$$\Phi_\varepsilon(x) := \left(\sum_{i=1}^n x_i^2 \geq \varepsilon \right) \wedge (V(x) \leq 0 \vee \nabla_{f_u} V(x) \geq 0),$$

unde x este delimitat în spațiul de stare D al sistemului. Parametrul de eroare numerică ε este introdus explicit pentru a controla sensibilitatea numerică în apropierea originii. Aici ε este cu ordine de mărime mai mic decât intervalul variabilelor de stare, adică $\sqrt{\varepsilon} \ll \min(1, \|D\|_2)$.

Remarca 1. Parametrul de eroare numerică ε ne permite să evităm probleme patologice în algoritmi numerici precum (*engl. arithmetic underflow*). Valorile din această regiune restrânsă corespund unor perturbări care sunt fizic nesemnificative. Acest parametru este important pentru a elimina din cadrul nostru problemele de sensibilitate numerică frecvent întâlnite în metodele SOS/SDP. De asemenea, notați că mingea ε nu afectează proprietățile seturilor de nivel Lyapunov și regiunile de atracție din afara acestora (adică, $D \setminus B_\varepsilon$).

Funcția de falsificare calculează soluțiile constrângerii de falsificare $\Phi_\varepsilon(x)$. Soluționarea constrângerilor necesită minimizarea globală a unor funcții extrem de non-convexe (implicând derivatele Lie ale funcției Lyapunov neuronale), iar aceasta este o sarcină computațională costisitoare (NP-hard). Ne bazăm pe lucrări recente în rezolvarea constrângerilor neliniare în solve SMT cum ar fi dReal [21], care a fost folosit pentru probleme similare în proiectarea de control care nu implică rețele neuronale.

La baza acestui algoritm autorii au folosit rezultate din teoria de stabilitate Lyapunov, definiții ce se regăsesc în Anexa B.

Algorithm 1 Control Neural Lyapunov

```

1: function LEARNING( $X, f, q^{lqr}$ )
2:   Setează rata de învățare (0.01), dimensiunea intrării (număr de variabile de stare),
   dimensiunea ieșirii (1)
3:   Inițializează controlerul de feedback  $u$  la soluția LQR  $q^{lqr}$ 
4:   repeat
5:      $V_\theta(x), u(x) \leftarrow \text{NN}_{\theta_u}(x)$  ▷ Propagarea înainte a rețelei neuronale
6:      $\nabla_u V_\theta(x) \leftarrow \sum_{i=1}^n \frac{\partial V_\theta}{\partial x_i} [f_u]_i(x)$ 
7:     Calculează riscul Lyapunov  $L(\theta, u)$ 
8:      $\theta \leftarrow \theta + \alpha \nabla_\theta L(\theta, u)$ 
9:      $u \leftarrow u + \alpha \nabla_u L(\theta, u)$  ▷ Actualizează ponderilor folosind SGD
10:  until convergență
11:  return  $V_\theta, u$ 
12: end function
13: function FALSIFICATION( $f, u, V_\theta, \varepsilon, \delta$ )
14:   Codifică condițiile în Definiția 1
15:   Folosește SMT solver cu  $\delta$  pentru a verifica condițiile
16:   return satisfăcut
17: end function
18: function MAIN
19:   Input: sistem dinamic ( $f$ ), parametrii LQR ( $q^{lqr}$ ), raza ( $\varepsilon$ ), precizia ( $\delta$ ) și un set
   inițial de stări eșantionate aleatoriu în  $D$ 
20:   while Satisfăcut do
21:     Adaugă contraexemplele la  $X$ 
22:      $V_\theta, u \leftarrow \text{LEARNING}(X, f, q^{lqr})$ 
23:     CE= FALSIFICATION( $f, u, V_\theta, \varepsilon, \delta$ )
24:   end while
25: end function

```

2.3.2. Sinteza Robustă de Structură Fixă pentru Sisteme Neliniare

În teza de doctorat [22] autorul își propune să extindă aparatul matematic în teoria controlului sistemelor neliniare pentru sinteza reguletoarelor de structură fixă aplicabile sistemelor neliniare ce prezintă incertitudini. Pentru a cuprinde o clasă cât mai largă de sisteme neliniare, teza se focusează în mare parte pe găsirea unui mecanism intermediar care să transforme un astfel de sistem neliniar generic într-o structură liniară sau biliniară. Astfel autorul mizează pe două direcții:

1. Dezvoltarea reguletoarelor de structură fixă bazată pe teoria pasivității pentru sisteme biliniare.
2. Liniarizarea exactă prin reacție de la stare și cuprinderea tuturor neliniarităților reziduale ce apar pentru sisteme neliniare afine în raport cu intrarea ce prezintă incertitudini.

Pentru abordarea primei direcții menționate mai sus problema studierii pasivității Krasovskii a sistemelor neliniare cu intrări afine este abordată prin prisma unei probleme de convexitate și se reduce la definirea unui set finit de inegalități liniare matriciale pentru două cazuri particulare: cazul biliniar și cazul politopic, set de condiții care s-a dovedit a fi suficient pentru pasivitatea Krasovskii.

Structura de control propusă este ilustrată în Figura 2.8 și conține un regulator Krasovskii pentru bucla interioară, ce va asigura stabilitatea asimptotică a sistemului neliniar, iar în bucla exterioară un regulator robust pentru a impune un set de performanțe.

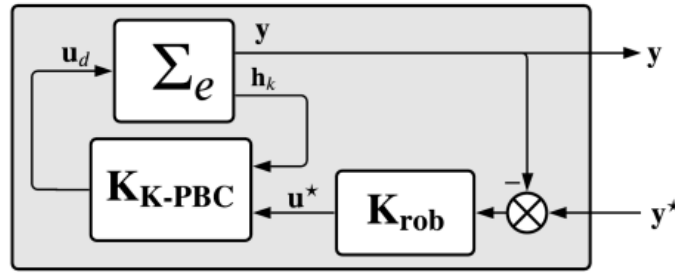


Figura 2.8: Structura de control bazată pe pasivitatea Krasovskii [22].

În cazul celei de a doua abordări teza extinde tehnicile matematice disponibile pentru liniarizarea cu reacție de la stare la cazul de liniarizare robustă de la stare. Astfel neliniaritățile reziduale obținute în urma aplicării unui semnal de intrare adecvat pentru a anula neliniaritatea sistemului după aplicarea difeomorfismului de liniarizare vor putea fi clasificate și încadrate de o incertitudine modelată ca un sistem impropriu ce are excesul zerourilor față de poli egal cu ordinul sistemului.

Prin urmare deoarece excesul de zerouri este egal cu ordinul sistemului, sistemul liniar cu incertitudini obținut este unul propriu, astfel este permisă sinteza unui regulator robust de structură fixă. Așadar liniarizarea robustă cu reacție de la stare a sistemelor neliniare afine în raport cu intrarea, sistem SISO, se rezumă la adăugarea unui singur regulator robust peste intrarea de liniarizare a sistemului dat, unde comanda de liniarizare este considerată de forma celei din Ecuația (2.6)

$$\mathbf{u} = \frac{\mathbf{v} - L_f^n h(\mathbf{x})}{L_g L_f^{n-1} h(\mathbf{x})} h. \quad (2.6)$$

Ca și principale dezavantaje ale acestor două metode prezentate din punct de

vedere al unei probleme de optimizare sunt convergența către optime locale, complexitatea computațională mare și lipsa garanțiilor teoretice.

Capitolul 3. Analiză și Proiectare

În acest capitol, se vor detalia aspectele matematice și metodele utilizate pentru dezvoltarea modelului al sistemului de propulsie propus. În primă fază se va concepe un model riguros al sistemului, având la bază principiile fizice fundamentale și relațiile dinamice ce descriu comportamentul fiecărui subsistem.

Ulterior, capitolul va trata fundamentele teoretice ale sintezei reguletoarelor aplicând tehnici avansate de control ce au fost menționate în 1.2. Accentul va fi pus pe tehnicile de stabilizare și urmărire a traiectoriei vitezei liniare a mașinii, având ca obiectiv obținerea unui comportament dinamic dorit în concordanță cu specificațiile de performanță.

3.1. Modelarea sistemului de propulsie

Sistemul de propulsie propus va avea o topologie de tipul celei din Figura 3.1 fiind o structură de tip paralel în care motorul este conectat prin roata dințată la transmisie, iar puterea mecanică este transmisă parțial sau integral mecanic către roți, iar partea care nu este transmisă către roți va fi convertită în energie electrică prin intermediul motorului electric pentru a încărca bateria. Motorul electric va funcționa în regim de motor atâta timp cât nivelul bateriei este menținut peste un anumit prag și va intra în regim generator în momentul în care nivelul de tensiune al bateriei scade sub acel prag.

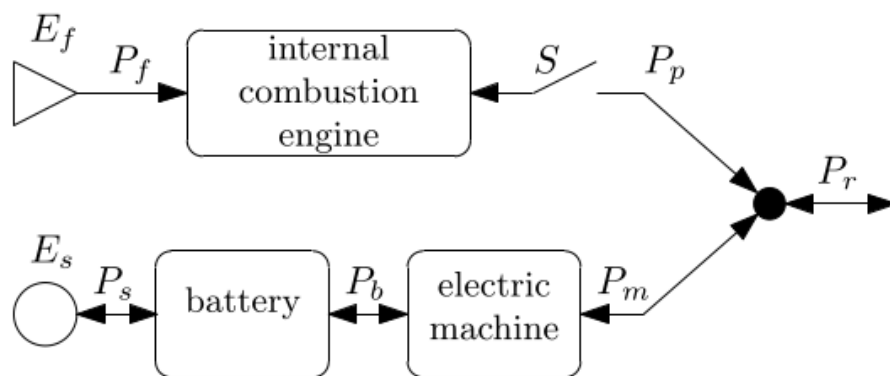


Figura 3.1: Topologia de tip paralel a sistemului hibrid de propulsie [23].

3.1.1. Modelarea Motorului Electric

În vederea stabilirii unui model matematic pentru motorul electric, se va porni de la schema circuitului echivalent al unui motor DC prezentată în Figura 4.1 și se vor aplica legile lui Kirchhoff pentru circuitul RL, din care se va deduce ecuația diferențială ce descrie comportamentul dinamic al părții electrice a unui motor. Pentru a obține ecuația diferențială ce descrie comportamentul mecanic, se va aplica legea a doua a lui Newton.

Astfel, după aplicarea celor două legi, obținem următorul model descris prin ecuațiile diferențiale, din care se va deduce ulterior o reprezentare în spațiul stărilor.

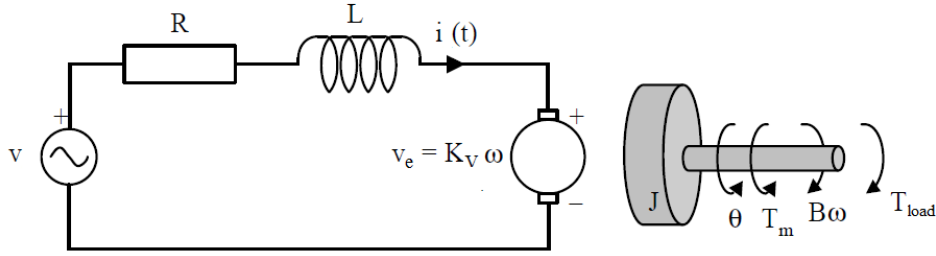


Figura 3.2: Schema circuitului echivalent al unui motor [24].

Ecuatiile de Bază

Dinamica unui motor de curent continuu poate fi descrisă prin următoarele ecuații diferențiale:

Ecuatia tensiunii electrice

Ecuatia tensiunii electrice descrie relația dintre tensiunea aplicată la bornele motorului, curentul care trece prin înfășurare și viteza unghiulară a rotorului. Aceasta poate fi dedusă aplicând legea lui Kirchhoff pentru tensiune în circuitul electric echivalent al motorului:

$$V(t) = L \frac{di(t)}{dt} + Ri(t) + K_b \omega(t), \quad (3.1)$$

unde în această ecuație simbolurile au următoare semnificație: $V(t)$ este tensiunea aplicată la bornele motorului $[V]$, $i(t)$ este curentul prin înfășurare $[A]$, L este inductanța înfășurării $[H]$, R este rezistența înfășurării $[\Omega]$, K_b este constanta de turație (tensiune electromotoare inversă) $[V \cdot s/rad]$, $\omega(t)$ este viteza unghiulară a rotorului $[rad/s]$.

Ecuatia cuplului mecanic

Ecuatia cuplului mecanic descrie dinamica rotorului motorului, ținând cont de momentele de forță care acționează asupra acestuia. Aceasta poate fi dedusă aplicând legea a doua a lui Newton pentru mișcarea de rotație:

$$T(t) = J \frac{d\omega(t)}{dt} + B\omega(t), \quad (3.2)$$

unde în această ecuație: $T(t)$ este cuplul mecanic generat de motor $[Nm]$, J este momentul de inerție al rotorului $[kg \cdot m^2]$, B este coeficientul de frecare $[Nm \cdot s/rad]$, $\omega(t)$ este viteza unghiulară a rotorului $[rad/s]$.

Această ecuație evidențiază faptul că cuplul mecanic $T(t)$ este echilibrat de suma momentului de inerție $J \frac{d\omega(t)}{dt}$ și a momentului de fricțiune $B\omega(t)$, ceea ce descrie comportamentul dinamic al rotorului în timpul rotației.

Reprezentarea în Spațiul Stărilor

Pentru a descrie sistemul dinamic al motorului într-o formă mai compactă și potrivită pentru analize ulterioare, vom introduce o reprezentare în spațiul stărilor. Definim vectorul de stare $\mathbf{x}(t)$, intrarea $\mathbf{u}(t)$ și ieșirea $\mathbf{y}(t)$ astfel:

$$\mathbf{x}(t) = \begin{bmatrix} i(t) \\ \omega(t) \end{bmatrix}, \quad \mathbf{u}(t) = V(t), \quad \mathbf{y}(t) = \omega(t).$$

Cu aceste definiții, ecuațiile de stare pot fi scrise în forma matriceală:

$$\frac{d}{dt} \begin{bmatrix} i(t) \\ \omega(t) \end{bmatrix} = \begin{bmatrix} -\frac{R}{L} & -\frac{K_b}{J} \\ \frac{K_t}{J} & -\frac{B}{J} \end{bmatrix} \begin{bmatrix} i(t) \\ \omega(t) \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} V(t). \quad (3.3)$$

Ieșirea sistemului poate fi exprimată astfel:

$$\mathbf{y}(t) = \omega(t) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} i(t) \\ \omega(t) \end{bmatrix}. \quad (3.4)$$

Funcția de Transfer

Funcția de transfer $G(s)$ a sistemului poate fi obținută aplicând transformata Laplace asupra ecuațiilor diferențiale și exprimând relația dintre ieșirea $\omega(s)$ și intrarea $V(s)$ în domeniul frecvențelor:

$$G(s) = \frac{\omega(s)}{V(s)} = \frac{K_t}{(Ls + R)(Js + B) + K_t K_b}. \quad (3.5)$$

3.1.2. Modelarea bateriei

Un ansamblu de baterii este compus din celule de baterie, celulă de tipul celei din Figura 3.3. În funcție de cerințele privind tensiunea de ieșire, puterea și capacitățile de energie pentru sistemul hibrid, un pachet de baterii este configurat prin conectarea mai multor celule în serie, în paralel sau mixt.

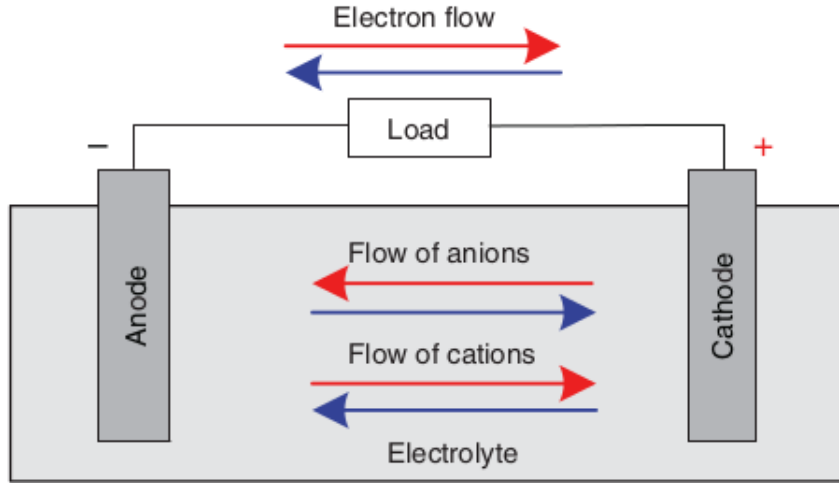


Figura 3.3: Celula de baterie [25].

Un circuit electric, precum cel din Figura 3.4, este utilizat în mod normal pentru a modela relația dintre curentul și tensiunea de la bornele bateriei. Starea de încărcare a sistemului de baterii poate fi calculată pe baza capacității în amperi-oră a bateriei, istoricului curentului, și eficienței de încărcare/descărcare. Parametrii modelului circuitului electric sunt de obicei identificați offline pe baza datelor de testare, cum ar fi prin metoda de caracterizare a puterii pulsului hibrid (HPPC).

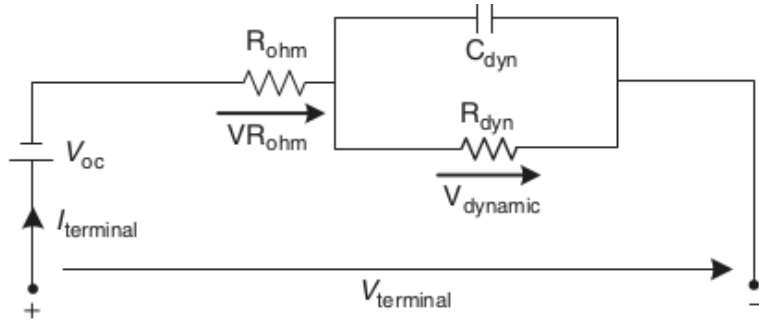


Figura 3.4: Circuitul electric echivalent al celulei [25].

Testul HPPC este conceput pentru a demonstra capacitățile de încărcare și descărcare ale unei baterii într-o perioadă scurtă de timp, de exemplu, 2 sau 10 secunde, la diferite niveluri de SOC (*engl. State of Charge*) sub diverse condiții de operare.

A. Modelarea Comportamentului Electric

Conform circuitului electric echivalent din 3.4, relația dintre tensiunea la borne și curentul măsurat la bornele bateriei poate fi exprimată prin următoarele ecuații:

Tensiunea la borne:

$$V_{terminal} = V_{oc} + V_{ohm} + V_{dynamic}, \quad (3.6)$$

această ecuație descrie tensiunea totală măsurată la bornele bateriei, $V_{terminal}$, ca sumă a tensiunii de circuit deschis V_{oc} , a tensiunii pe rezistența ohmică V_{ohm} , și a tensiunii dinamice $V_{dynamic}$.

Tensiunea circuitului deschis:

$$V_{oc} = f(SOC, T), \quad (3.7)$$

astfel tensiunea de circuit deschis, V_{oc} , depinde de starea de încărcare (SOC) a bateriei și de temperatura acesteia T . Aceasta reprezintă tensiunea măsurată la bornele bateriei atunci când nu există niciun curent care să circule prin circuit.

Tensiunea pe rezistența ohmică:

$$V_{ohm} = I \cdot R_{ohm} = I \cdot R_{ohm}(SOC, T), \quad (3.8)$$

astfel tensiunea pe rezistența, V_{ohm} , este proporțională cu curentul I care trece prin baterie și cu rezistența R_{ohm} . Rezistența poate varia în funcție de SOC și de temperatura T .

Curentul dinamic:

$$I_{R_{dyn}} = \frac{V_{dynamic}}{R_{dyn}(SOC, T)}, \quad (3.9)$$

astfel curentul prin componenta rezistivă dinamică $I_{R_{\text{dyn}}}$ este dat de tensiunea dinamică V_{dynamic} împărțită la rezistența dinamică R_{dyn} , care depinde de SOC și de temperatură.

$$I_{C_{\text{dyn}}} = C_{\text{dyn}}(\text{SOC}, T) \frac{dV_{\text{dynamic}}}{dt}, \quad (3.10)$$

astfel curentul prin componenta capacitivă dinamică $I_{C_{\text{dyn}}}$ este proporțional cu capacitatea dinamică C_{dyn} și derivata în timp a tensiunii dinamice V_{dynamic} .

Curentul total:

$$I = I_{R_{\text{dyn}}} + I_{C_{\text{dyn}}}, \quad (3.11)$$

de unde rezultă că curentul total I este suma curenților prin componentele rezistivă și capacitivă a modelului dinamic. Astfel, tensiunea dinamică poate fi descrisă de următoarea ecuație diferențială:

$$\frac{dV_{\text{dynamic}}}{dt} + \frac{V_{\text{dynamic}}}{R_{\text{dyn}}(\text{SOC}, T)C_{\text{dyn}}(\text{SOC}, T)} = \frac{I}{C_{\text{dyn}}(\text{SOC}, T)}, \quad (3.12)$$

această ecuație diferențială descrie comportamentul dinamic al tensiunii V_{dynamic} în timp, ținând cont de curentul total I , rezistența dinamică R_{dyn} și capacitatea dinamică C_{dyn} . Ecuația diferențială generală a circuitului electric este dată de:

$$\frac{dV_{\text{terminal}}}{dt} + \frac{V_{\text{terminal}}}{R_{\text{dyn}}C_{\text{dyn}}} = R_{\text{ohm}} \frac{dI}{dt} + \frac{R_{\text{dyn}} + R_{\text{ohm}}}{R_{\text{dyn}}C_{\text{dyn}}} I + \frac{V_{\text{oc}}}{R_{\text{dyn}}C_{\text{dyn}}}, \quad (3.13)$$

unde: I este curentul la bornele sistemului de baterii (A), V_{terminal} este tensiunea la bornele sistemului de baterii (V), T este temperatura bateriei, iar V_{oc} , R_{ohm} , R_{dyn} , și C_{dyn} sunt parametri ai modelului de circuit care depind de temperatura T și de SOC. Acești parametri pot fi deduși pe cale experimentală, cum ar fi testul HPPC.

B. Calculul SOC

Pentru analiza și simularea performanței sistemului de propulsie hibrid, precum și pentru proiectarea acestuia, calculul stării de încărcare (SOC) a bateriei poate fi realizat prin integrarea curentului în timp, conform următoarei relații:

$$\text{SOC}(t) = \text{SOC}(t_i) + \frac{1}{\text{Cap}_{\text{Ahr}} \cdot 3600} \int_{t_i}^t I(t) \eta_{\text{bat}}(\text{SOC}, T, \text{sign}[I(t)]) dt, \quad (3.14)$$

această ecuație calculează SOC la un moment dat t pornind de la SOC inițial, $\text{SOC}(t_i)$, ținând cont de capacitatea bateriei exprimată în amperi-oră Cap_{Ahr} și de eficiența Coulombică η_{bat} a bateriei. Eficiența Coulombică η_{bat} ia în considerare pierderile de energie în timpul ciclurilor de încărcare și descărcare, iar capacitatea în amperi-oră Cap_{Ahr} determină capacitatea totală de stocare a energiei a bateriei.

3.1.3. Modelarea motorului pe combustie internă

Când motorul este pornit și ambreiajul arborelui de transmisie este cuplat, motorul furnizează cuplul de propulsie. Acest cuplu poate fi descris prin următoarele ecuații ce vor fi expuse în continuare, ecuații ce iau în calcul mai multe aspecte fizice pentru a

determina cuplul total generat de motor.

Cuplul necesar pentru componentele mecanice

Cuplul necesar pentru a angrena componentele mecanice ale motorului se calculează folosind relația:

$$\tau_{\text{access}} = \frac{P_{\text{access}}}{\omega_{\text{eng}}}, \quad (3.15)$$

unde: τ_{access} este cuplul necesar pentru componentele mecanice, P_{access} este puterea consumată de componentele mecanice, ω_{eng} este viteza unghiulară a motorului.

Cuplul închiderii accelerației

Cuplul generat atunci când accelerația este închisă (de exemplu, în cazul frânării motorului) este descris prin:

$$\tau_{\text{cct}} = \alpha_1(T) \cdot d \cdot \delta(t) + \alpha_2(T) \operatorname{sgn}(\omega) + \alpha_3(T) \left(\frac{\omega}{\omega_{\text{max_eng}}} \right) + \alpha_4(T) \left(\frac{\omega}{\omega_{\text{max_eng}}} \right)^2, \quad (3.16)$$

unde: τ_{cct} este cuplul generat la închiderea accelerației, $\alpha_1(T)$, $\alpha_2(T)$, $\alpha_3(T)$, $\alpha_4(T)$ sunt coeficienți care depind de temperatura motorului T , d este deplasarea manetei de accelerație, $\delta(t)$ este funcția delta Dirac care reprezintă impactul instantaneu al închiderii accelerației, ω este viteza unghiulară a motorului, $\omega_{\text{max_eng}}$ este viteza unghiulară maximă a motorului.

Cuplul necesar pentru accelerație

Cuplul necesar pentru a accelera motorul, adică pentru a crește viteza unghiulară:

$$\tau_a = J_{\text{eng}} \frac{d\omega}{dt}, \quad (3.17)$$

unde: τ_a este cuplul necesar pentru accelerație, J_{eng} este momentul de inerție al motorului, $\frac{d\omega}{dt}$ este accelerația unghiulară a motorului.

Cuplul generat de motor

Cuplul total generat de motor, care include toate componentele menționate anterior, se calculează prin:

$$\tau_{\text{gen/d}} = \tau_{\text{demand}} + \tau_{\text{access}} + \tau_{\text{cct}} + \tau_a, \quad (3.18)$$

unde: $\tau_{\text{gen/d}}$ este cuplul total generat de motor, τ_{demand} este cuplul cerut de vehicul, τ_{access} este cuplul necesar pentru accesoriile mecanice, τ_{cct} este cuplul închiderii accelerației, τ_a este cuplul necesar pentru accelerație.

Limitarea cuplului maxim

Cuplul total generat de motor trebuie să fie limitat pentru a nu depăși capacitatea maximă fizică a motorului, conform relației:

$$\tau_{\text{gen/d}} \leq \max(\operatorname{trq}_{\text{eng}}) = f(\omega), \quad (3.19)$$

unde: $\max(\text{trq}_{\text{eng}})$ este cuplul maxim pe care motorul îl poate furniza, $f(\omega)$ este funcția care descrie această limitare în funcție de viteza unghiulară ω a motorului.

Funcția de Transfer a Cuplului Motorului cu Limitare Maximă

Funcția de transfer a sistemului care include limitarea cuplului maxim ca un câștig (gain) este exprimată astfel:

$$G(s) = \frac{\max(\text{trq}_{\text{eng}})}{\tau_{\text{demand}}(s)} \cdot \frac{1}{\tau J_{\text{eng}} s + 1}, \quad (3.20)$$

unde: $G(s)$ este funcția de transfer a sistemului, $\max(\text{trq}_{\text{eng}})$ este câștigul maxim (cuplul maxim pe care motorul îl poate furniza), τJ_{eng} este o constantă de timp asociată momentului de inerție al motorului, $\tau_{\text{demand}}(s)$ este cererea de cuplu din partea vehiculului.

Această funcție de transfer reflectă răspunsul dinamic al motorului la cererea de cuplu, având în vedere limitările impuse de capacitatea fizică a motorului.

3.1.4. Modelarea dinamicii

Dinamica sistemului poate fi segmentată în două componente: dinamica sistemului motoarelor și dinamica vehiculului. Sistemul motoarelor este interfațat cu sistemul mașinii hibride printr-o unitate de transmisie așa cum este prezentat în Figura 3.5. Într-o configurație standard, pedalele de accelerație/frână sunt folosite de către șofer pentru a transmite semnale de comandă de accelerare/deceleare către controller. Acest sistem este integrat cu sistemul de propulsie menționat și în esență se va modula viteza motorului DC pentru a controla întregul sistem hibrid.

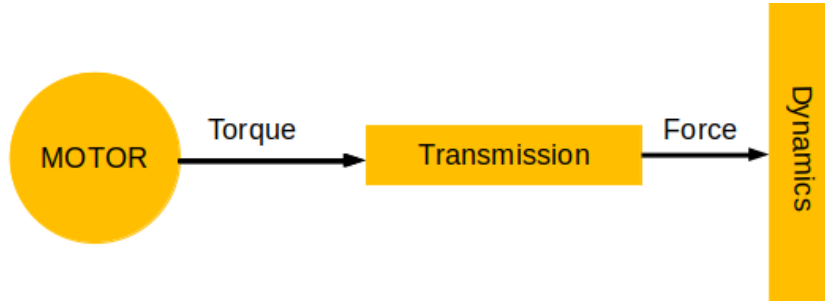


Figura 3.5: Configurație standard de interfațare între motor și mașină

Modelul dinamic al vehiculului este bazat pe suma tuturor forțelor ce acționează asupra sistemului fapt ce reiese din a doua lege a lui Newton.

Forța Rezultantă Totală și Dinamica Vehiculului

Sumarea tuturor forțelor active produce forța rezultantă totală, dată de ecuația:

$$F = Mgf_r \cos(\alpha) + \frac{1}{2}\rho A_f C_D V^2 + Mg \sin(\alpha) + M \frac{dV}{dt}. \quad (3.21)$$

Forțele individuale care acționează asupra vehiculului

1. **Forța gravitațională**, care acționează asupra unui vehicul în timpul traversării unei pante, este determinată de unghiul pantei și este proporțională cu masa vehiculului.

Aceasta este reprezentată prin:

$$F_g = Mg \sin(\alpha). \quad (3.22)$$

2. **Forța de rezistență aerodinamică**, creată de aerul care curge peste caroseria vehiculului în mișcare, este influențată de orientarea suprafeței frontale a vehiculului. Reprezentarea sa matematică este:

$$F_{\text{aero}} = \frac{1}{2} \rho A_f C_D V^2. \quad (3.23)$$

3. **Forța de rezistență la rulare**, provine în principal din frecarea dintre anvelopele vehiculului și suprafața drumului, opunându-se direcției de mișcare a vehiculului. Această forță este definită prin:

$$F_r = M g f_r \cos(\alpha). \quad (3.24)$$

Forța rezultantă F un cuplu care afectează turația motorului. Acest cuplu, numit forța de tracțiune, joacă un rol esențial în propulsia vehiculului și este exprimat astfel:

$$T_L = F \times \frac{R_{\text{wheel}}}{K_{\text{gear}}}. \quad (3.25)$$

Relația dintre viteza unghiulară a motorului și viteza liniară a vehiculului

Relația dintre viteza unghiulară a motorului, ω_{mot} , și viteza liniară a vehiculului, V , este explicată prin ecuația:

$$V = \frac{R_{\text{wheel}}}{K_{\text{gear}}} \omega_{\text{mot}}, \quad (3.26)$$

unde: T_L este cuplul generat de motorul de tracțiune, R_{wheel} este raza roții, ω_{mot} este viteza de rotație a motorului.

Cutia de viteze

Adăugarea unei cutii de viteze într-un vehicul electric (EV) are rolul de a spori eficiența motorului prin modularea relației dintre viteza motorului și viteza roților. Această modulare a cuplului, în concordanță cu condițiile de conducere, contribuie la o experiență de conducere îmbunătățită. Valoarea de conversie mecanică este obținută din ecuația:

$$\begin{aligned} T_{\text{gear}} &= K_{\text{gear}} \times T_{\text{mot}} \\ \Omega_{\text{mot}} &= K_{\text{gear}} \times \Omega_{\text{gear}}, \end{aligned} \quad (3.27)$$

unde: T_{gear} și Ω_{gear} sunt cuplul și viteza de rotație după reducerea din cutia de viteze, T_{mot} și Ω_{mot} sunt cuplul și viteza de rotație ale motorului, K_{gear} este coeficientul de reducere al cutiei de viteze.

3.2. Sliding Mode Control

În cadrul următoarelor secțiuni din acest capitol, se vor detalia aspectele teoretice și fundamentele matematice care stau la baza proiectării reguletoarelor, utilizând metode avansate de control. Aceste metode sunt esențiale în dezvoltarea de sisteme de control

robuste, care pot gestiona comportamente complexe și variabile în timp real. Proiectarea unui regulator nu se limitează la realizarea unei performanțe de bază, ci urmărește optimizarea răspunsului sistemului și îmbunătățirea stabilității, preciziei și eficienței acestuia.

În sistemele de control, controlul de tip *sliding mode* (SMC) este o metodă de control aplicată sistemelor neliniare care modifică dinamica unui astfel de sistem prin aplicarea unui semnal de comandă discontinuu la o frecvență foarte mare care forțează sistemul să "alunece" de-a lungul unei secțiuni transversale. Legea de control nu este o funcție continuă în timp. În schimb, aceasta poate comuta de la o stare discretă la una continuă în funcție de poziția curentă în spațiul stărilor. Prin urmare, controlul de tip *sliding mode* este o metodă de control cu structură variabilă. Multiplele structuri de control sunt proiectate astfel încât traiectoriile să se îndrepte întotdeauna către o regiune adiacentă cu o structură de control diferită, astfel încât traiectoria finală nu va exista în întregime într-o singură structură de control. În schimb, va aluneca de-a lungul marginilor structurilor de control. Mișcarea sistemului în timp ce alunecă de-a lungul acestor margini este denumită mod de alunecare, iar locul geometric format din aceste delimitări este denumită suprafața de alunecare (hipersuprafață). În contextul teoriei moderne a controlului, orice sistem cu structură variabilă, cum ar fi un sistem sub control SMC, poate fi considerat un caz special al unui sistem dinamic hibrid, deoarece sistemul parcurge atât un spațiu continuu al stărilor, cât și se deplasează prin diferite moduri de control discret.

Bazele tehnicii de tip sliding mode

Tehnica constă în două etape pentru proiectarea SMC (*engl. Sliding Mode Control*) ce sunt descrise după cum urmează:

- (i) O suprafață de alunecare este definită într-un mod care să asigure că dinamica dorită a sistemului este obținută în timpul modului de alunecare.
- (ii) Un regulator este proiectat astfel încât să conducă dinamica sistemului în buclă închisă pentru a atinge și a fi reținută pe suprafața de alunecare.

3.2.1. Modelul Sistemului și Proiectarea Suprafeței de Sliding Mode

Fără a pierde generalitatea, considerăm următorul sistem liniar invariant în timp (LTI):

$$\dot{x}(t) = Ax(t) + Bu(t), \quad (3.28)$$

unde $x \in \mathbb{R}^n$ este vectorul de stare, $u \in \mathbb{R}^m$ este vectorul intrării, $A \in \mathbb{R}^{n \times n}$ și $B \in \mathbb{R}^{n \times m}$ sunt matrice constante ale sistemului.

Remarca 3.1: Se presupune că $n > m$, B este de rang complet m , perechea (A, B) este complet controlabilă, adică, matricea de controlabilitate $[B \ AB \ A^2B \ \dots \ A^{n-1}B]$ are rang complet m .

Definim un vector variabil de *sliding* $s(t) \in \mathbb{R}^m$ care trece prin originea spațiului stărilor:

$$s(t) = Cx(t), \quad (3.29)$$

unde $C \in \mathbb{R}^{m \times n}$ este vectorul de parametri al suprafeței de sliding mode și $\|CB\| \neq 0$.

Sistemul (3.28) este considerat că atinge o suprafață de *sliding mode* atunci când vectorul variabil al stării ajunge și rămâne la intersecția celor m variabile de plan de comutare.

Metoda de control echivalent este o modalitate de a determina mișcarea sistemului restricționată la suprafața de sliding mode $s(x) = 0$. Pe suprafața de sliding mode, $s(x) = 0$ și $\dot{s}(x) = 0$, utilizând expresiile (3.28) și (3.29), avem:

$$\dot{s}(t) = C\dot{x}(t) = 0, \quad (3.30)$$

$$C(Ax(t) + Bu_{eq}(t)) = 0, \quad (3.31)$$

unde $u_{eq}(t)$ este privit ca un control echivalent. Din expresia (3.31), controlul echivalent poate fi exprimat ca:

$$u_{eq}(t) = -(CB)^{-1}CAx(t). \quad (3.32)$$

Substituind (3.32) în (3.28) obținem următoarea ecuație diferențială:

$$\dot{x}(t) = [I - B(CB)^{-1}C]Ax(t). \quad (3.33)$$

Sistemul (3.33) este numit sistem echivalent, care descrie mișcarea dinamică a sistemului (3.28) pe suprafața de *sliding mode*. Caracteristicile sistemului echivalent pot fi rezumate astfel:

Remarca 3.2: Comportamentul dinamic al sistemului echivalent este independent de intrarea de control. Astfel, determinarea matricei C poate fi completată fără a avea cunoștință în prealabil de forma comenzii. În general, parametrul C al suprafeței de *sliding* este proiectat astfel încât răspunsul sistemului restricționat pe suprafața de alunecare $s(x) = 0$ să aibă un comportament dorit, cum ar fi stabilitatea asimptotică și un răspuns tranzitoriu impus.

3.2.2. Faza de *Atingere*

Proiectarea SMC include faza de atingere și faza de alunecare. Faza de atingere este crucială în sensul că dinamica sistemului este garantată să atingă suprafața de *sliding* și să urmărească traiectoria de stare ulterior. Pentru un caz concret, ideea unui sliding mode al unui sistem de ordinul doi poate fi ilustrată în Figura 3.6.

Problema importantă este cum se proiectează un regulator pentru a garanta atingerea variabilei de *sliding* la suprafața de alunecare. Prin urmare, sarcina SMC-ului este de a conduce variabila de sliding, $s(t)$, să convergă la zero, iar apoi dinamica dorită a sistemului va fi obținută.

Condiția de stabilitate asimptotică

De fapt, condiția ca variabilele planului de comutare să atingă suprafața de sliding mode este o problemă de convergență. Prin urmare, metoda directă a lui Lyapunov a fost larg utilizată în proiectarea SMC ca o condiție de stabilitate pentru a asigura convergența variabilei de sliding mode pe suprafața de sliding în timpul fazei de atingere. Următoarea funcție candidată Lyapunov este utilizată:

$$V(t) = \frac{1}{2}s^T(t)s(t). \quad (3.34)$$

Pentru a garanta stabilitatea asimptotică a sistemului (3.28) în jurul punctului de echilibru $x(t) = 0$, următoarea condiție de atingere trebuie să fie satisfăcută:

$$\dot{V}(t) = s^T(t)\dot{s}(t) < 0 \text{ pentru } s(t) \neq 0. \quad (3.35)$$

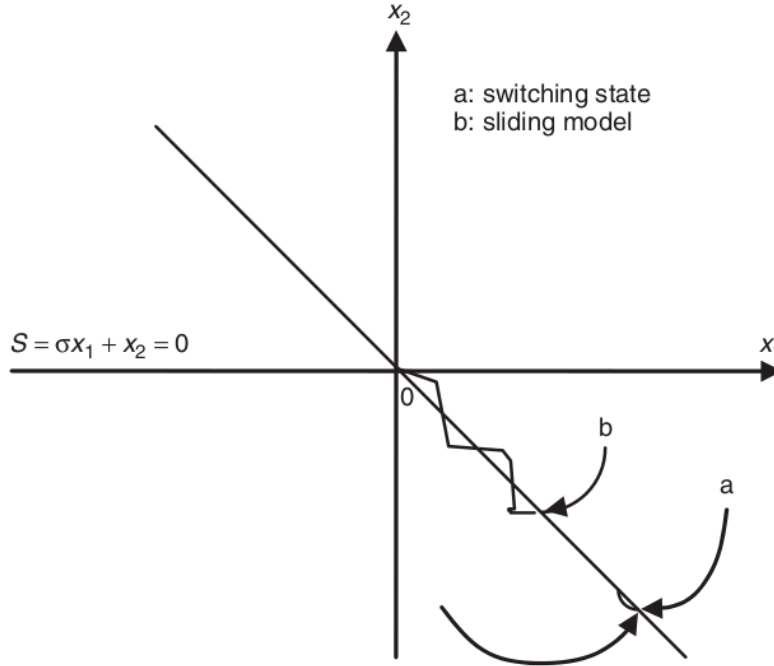


Figura 3.6: Modul de *sliding* și traiectoria de stare [26].

Remarca 3.3: Condiția (3.35) este suficientă pentru a asigura existența modului de *sliding*. Este de remarcat faptul că majoritatea reguletoarelor SMC sunt proiectate pe baza condiției de atingere din (3.35) pentru a asigura că legea de comandă va conduce ca $s(t)$ să convergă asimptotic la zero.

3.2.3. Legile de Atingere

SMC poate fi proiectat pe baza legilor de atingere pentru a garanta existența modului de *sliding*. Câteva tipuri posibile de legi de atingere sunt prezentate în [27]. În general, legea de atingere poate fi generalizată sub următoarea formă:

$$\dot{s} = -\varepsilon \text{sign}(s) - f(s), \quad \varepsilon > 0, \quad (3.36)$$

unde $f(0) = 0$ și $sf(s) > 0$ atunci când $s \neq 0$.

Una dintre legile de atingere utilizată în această lucrare este următoare și poate fi derivată din (3.36) astfel:

Legea de atingere cu rată constantă

$$\dot{s} = -\varepsilon \text{sign}(s), \quad \varepsilon > 0, \quad (3.37)$$

Această lege constrânge variabila de comutare să atingă varietatea de comutare la o rată constantă ε . Meritului acestei legi constă în simplitatea sa. Cu toate acestea, deoarece ε este prea mic, timpul de atingere va fi prea lung. Pe de altă parte, ε prea mare va cauza apariția fenomenului de *chattering*.

3.2.4. Proiectarea Regulatorului Echivalent

În cele mai multe scheme SMC, intrarea de control constă de obicei din două componente, după cum urmează:

$$u(t) = u_{eq}(t) + u_s(t), \quad (3.38)$$

unde componenta liniară $u_{eq}(t)$ este definită ca în (3.32), iar semnalul neliniar încorporează componenta discontinuă dată mai jos:

$$u_s(t) = -\eta(CB)^{-1}\text{sign}(s(t)), \quad (3.39)$$

unde $\eta > 0$ este un parametru de control constant.

Substituind (3.32) și (3.39) în (3.33) se obține:

$$\dot{V}(t) = s^T(t)CAx(t) + s^T(t)CBu(t) = -\eta|s(t)| < 0. \quad (3.40)$$

3.2.5. Fenomenul de Chattering

Un mod de alunecare ideal, așa cum este arătat în Figura 2.1, nu există în practică deoarece ar implica faptul că semnalul comenzii comută la o frecvență infinită. Datorită imperfecțiunilor în dispozitivele de comutare, SMC suferă de fenomenul de *chattering*, întreruperea în controlul cu reacție produce un comportament dinamic particular în vecinătatea suprafeței de sliding mode, așa cum este ilustrat în 3.7 [28].

În Figura 3.7, traiectoria sistemului în regiunea $s(t) > 0$ se îndreaptă spre suprafața de *sliding mode* $s(t) = 0$. Aceasta atinge pentru prima dată suprafața în punctul A. Într-un SMC ideal, traiectoria ar trebui să înceapă să alunece pe suprafață din punctul A. Cu toate acestea, din cauza unei întârzieri între momentul în care semnul lui $s(t)$ se schimbă și momentul în care controlul comută, traiectoria își inversează direcția și se îndreaptă din nou spre suprafață. Repetarea acestui proces creează "mișcarea zig-zag" care oscilează în jurul suprafeței de *sliding mode* definite.

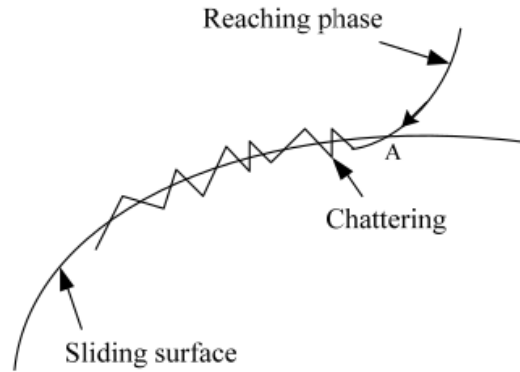


Figura 3.7: Fenomenul de chattering [26].

Prezența fenomenului de *chattering* scade performanțele controlului, aduce pierderi de căldură ridicate în circuitele de putere electrică și uzură intensă a pieselor mecanice în mișcare. Poate excita dinamici de frecvență înaltă nemodelate, ceea ce degradează performanța sistemului și poate chiar duce la instabilitate.

Tehnica Straturilor Limită

Au fost propuse diverse tehnici pentru a reduce sau elimina fenomenul de precum este descris și în *chattering*[29]. Una dintre metodele folosite este *Tehnica Straturilor*

Limită fiind o abordare comună pentru eliminarea fenomenului de *chattering*, dar are și dezavantaje.

Se observă că componenta discontinuă este descrisă în (3.39), iar această tehnică presupune înlocuirea funcției semn din (3.39) cu o funcție de saturație, ce este prezentată în Figura 3.8, după cum urmează:

$$u_s(t) = -\eta(CB)^{-1}\text{sat}(s(t)), \quad (3.41)$$

unde funcția de saturație este definită astfel:

$$\text{sat}(s) = \begin{cases} \frac{s}{\rho}, & \text{pentru } |s| \leq \rho \\ \text{sign}(s), & \text{pentru } |s| > \rho \end{cases}. \quad (3.42)$$

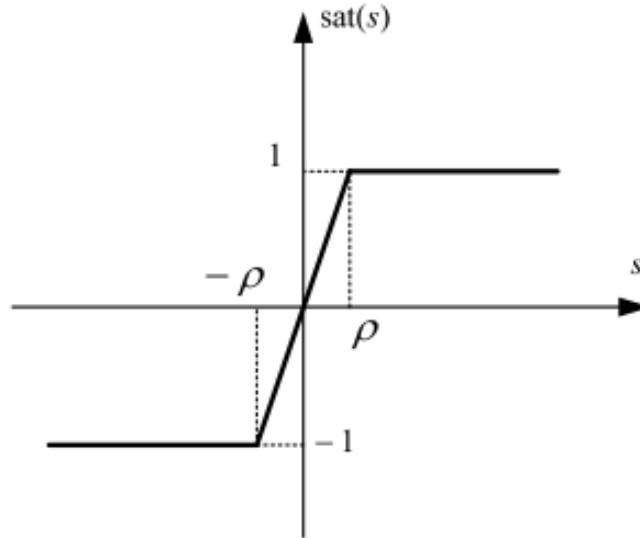


Figura 3.8: Funcția de saturație $\text{sat}(s)$ [26].

O constantă pozitivă $\rho > 0$ ar trebui aleasă prin simulare sau experimental pentru a garanta că fenomenul de *chattering* poate fi eliminat și că ar putea garanta o performanță rezonabilă de control. Cu toate acestea, deși fenomenul de *chattering* poate fi eliminat, robustețea modului de *sliding* este compromisă în timp. O astfel de abordare poate duce la pierderea stabilității asimptotice. Prin urmare, tehnica straturilor limită nu este o soluție perfectă pentru a elimina fenomenul de *chattering*.

O altă soluție pentru a face față fenomenului de *chattering* se bazează pe o metodă de aproximare continuă (denumită și metoda *pseudo-sliding mode* în literatură) în care funcția semn din (3.39) este înlocuită de o aproximare continuă

$$u_s(t) = -\eta(CB)^{-1} \left(\frac{s(t)}{|s(t)| + \rho} \right). \quad (3.43)$$

3.2.6. Ajustarea Parametrului de Control

Controlul echivalent pentru sistemele SISO poate fi obținut utilizând legea de control echivalent și legea de control comutat, după cum urmează:

Legea de control echivalentă: este dată prin setarea derivatei suprafeței de sliding mode la zero, astfel:

$$\dot{s}(t) = 0, \quad (3.44)$$

apoi se rezolvă pentru legea de control a intrării u , care va fi controlul echivalent.

Controlul în comutație este dat de următoarea ecuație:

$$u_s = \dot{s}(t) = -k_p \text{sgn}(s(t)), \quad (3.45)$$

Prin urmare, legea generală de control $u = u_{eq} + u_s$ este dată prin:

$$\frac{d}{dt} \left(\left(\frac{d}{dt} + k \right)^{r-1} e \right) = -k \text{sgn}(s(t)). \quad (3.46)$$

Forma suprafeței de sliding mode $s(t)$ ia următoarea formă:

$$s(t) = \sigma = \left(\frac{d}{dt} + k \right)^{r-1} e, \quad (3.47)$$

unde r este gradul relativ al sistemului, e este eroarea între vectorul de ieșire și intrarea de referință dorită; k este un parametru pozitiv care poate fi ales arbitrar.

Parametrii legii de control în sliding mode ar trebui să fie aleși pentru a îmbunătăți performanța sistemului controlat. Cu toate acestea, acești parametri sunt adesea aleși arbitrar folosind metoda încercare-eroare, ceea ce poate lua mult timp pentru a găsi valorile aproximativ exacte.

Vom folosi o metodă simplă care conduce la obținerea valorilor exacte prin identificarea cu o caracteristică a unui sistem de ordinul întâi sau al doilea cu (3.46). Să presupunem că avem un sistem neliniar cu un grad relativ $r = 2$; prin urmare, (3.46) va fi:

$$\ddot{e} + k\dot{e} = -k \text{sgn}(s(t)). \quad (3.48)$$

Deoarece $\text{sgn}(s) = \frac{s}{|s|}$, ecuația va deveni:

$$\ddot{e} + k\dot{e} = -k \frac{s}{|s|} \quad (3.49)$$

$$\ddot{e} + k\dot{e} = -k \frac{\dot{e} + ke}{|s|}, \quad (3.50)$$

observăm că $|s|$ este o funcție pozitivă scrisă în termeni ai erorii, care poate fi aleasă ca o valoare mică $|s| \ll 1$. Astfel, Ecuația (3.50) va deveni:

$$\ddot{e} + \left(k + \frac{k_p}{|s|} \right) \dot{e} + \frac{k}{|s|} e = 0, \quad (3.51)$$

astfel ecuația (3.51) ia forma unui sistem de ordinul al doilea:

$$s^2 + 2\zeta\omega s + \omega^2 = 0. \quad (3.52)$$

Prin identificarea Ecuației (3.52) cu Ecuația (3.50), obținem:

$$\left(k + \frac{k_p}{|s|}\right) = 2\zeta\omega, \quad (3.53)$$

$$\frac{k}{|s|} = \omega^2, \quad (3.54)$$

variabilele ω și ζ sunt alese pentru a impune performanțele dorite ale sistemului în buclă închisă.

3.3. Proiectarea reguletoarelor bazate pe norma H_∞

3.3.1. Introducerea Normei H_∞

Un sistem cu matricea de transfer $\mathbf{F}(s)$ poate fi privit ca o mapare între un semnal de intrare $u(t)$ și un semnal de ieșire $y(t)$. Dacă $u(t)$ are energie finită, atunci

$$\max_{u(t) \neq 0} \frac{\|y(t)\|_2}{\|u(t)\|_2}$$

reprezintă mărimea maximă a răspunsului sistemului în termeni de energie. Câștigul de energie (sau norma L_2) γ al unui sistem cu matricea de transfer $\mathbf{F}(s)$ este definit ca:

$$\gamma = \max_{0 < \|u(t)\|_2 < \infty} \frac{\|y(t)\|_2}{\|u(t)\|_2} = \max_{0 < \|u(t)\|_2 < \infty} \frac{\|\mathcal{L}^{-1}[\mathbf{F}(s)u(s)]\|_2}{\|u(t)\|_2}. \quad (3.56)$$

Alternativ, câștigul de energie (sau câștigul L_2) al unui sistem stabil poate fi caracterizat ca valoarea minimă a lui γ astfel încât:

$$\|y(t)\|_2 = \|\mathcal{L}^{-1}[\mathbf{F}(s)u(s)]\|_2 \leq \gamma \|u(t)\|_2 \text{ pentru toate intrările finite de energie } u(t). \quad (3.57)$$

Dacă sistemul cu matricea de transfer $\mathbf{F}(s)$ nu este stabil, atunci nu există un astfel de γ și $\gamma = \infty$. Pe de altă parte, dacă $\mathbf{F}(s)$ este stabil, atunci γ este finită. În consecință, o matrice de transfer $\mathbf{F}(s)$ este stabilă dacă și numai dacă toate semnalele de intrare finite $u(t) \in L_2$ sunt mapate de $\mathbf{F}(s)$ către semnale finite de energie $y(t) \in L_2$. Alternativ, matricea de transfer $\mathbf{F}(s)$ este stabilă dacă și numai dacă $\gamma < \infty$. În plus, dacă $\mathbf{F}(s)$ este o matrice de transfer stabilă, norma H_∞ este egală cu câștigul de energie al sistemului:

$$\|\mathbf{F}(s)\|_\infty = \gamma = \max_{\omega} \bar{\sigma}(\mathbf{F}(j\omega)), \quad (3.58)$$

unde $\bar{\sigma}(\mathbf{F}(j\omega))$ este astfel încât:

$$0 < \sigma(\mathbf{F}(j\omega)) \leq \sigma_i(\mathbf{F}(j\omega)) \leq \bar{\sigma}(\mathbf{F}(j\omega)) \quad \forall i, \omega. \quad (3.59)$$

Norma H_∞ a matricei de transfer $\mathbf{F}(s)$ este norma indusă obținută prin normele L_2 ale semnalelor vectoriale de intrare și ieșire:

$$\|\mathbf{F}(s)\|_\infty = \max_{u(t) \neq 0} \frac{\|y(t)\|_2}{\|u(t)\|_2} = \max_{\omega} \bar{\sigma}(\mathbf{F}(j\omega)), \quad (3.60)$$

această ecuație indică faptul că norma H_∞ este o măsură a câștigului maxim al unui sistem

stabil pe toate frecvențele ω ; este răspunsul în frecvență în cazul cel mai nefavorabil al sistemului.

Pentru un sistem SISO (Single Input Single Output) liniar stabil cu funcția de transfer $\mathbf{F}(s)$, norma H_∞ reprezintă valoarea maximă pe diagrama Bode a funcției $\mathbf{F}(j\omega)$:

$$\|\mathbf{F}(s)\|_\infty = \max_{\omega} |\mathbf{F}(j\omega)|. \quad (3.61)$$

Spre deosebire de norma H_2 , norma H_∞ nu poate fi calculată analitic. Doar soluții numerice pot fi obținute. Se poate demonstra că, dacă (A, B, C, D) reprezintă o realizare minimală în spațiul stărilor a matricei de transfer $\mathbf{F}(s)$, atunci norma H_∞ poate fi calculată astfel:

$$\|\mathbf{F}(s)\|_\infty = \sup_{\gamma \geq 0} \{\gamma : \mathbf{H} \text{ are valori proprii cu parte reală nulă}\}. \quad (3.62)$$

Matricea hamiltoniană \mathbf{H} este definită astfel:

$$\mathbf{H} = \begin{bmatrix} \mathbf{A} + \mathbf{B}\mathbf{R}_\gamma^{-1}\mathbf{D}^T\mathbf{C} & \mathbf{B}\mathbf{R}_\gamma^{-1}\mathbf{B}^T \\ -\mathbf{C}^T(\mathbf{I} + \mathbf{D}\mathbf{R}_\gamma^{-1}\mathbf{D}^T)\mathbf{C} & -(\mathbf{A} + \mathbf{B}\mathbf{R}_\gamma^{-1}\mathbf{D}^T\mathbf{C})^T \end{bmatrix} \quad (3.63)$$

$$= \begin{bmatrix} \mathbf{A} & \mathbf{B}\mathbf{B}^T \\ -\mathbf{C}^T\mathbf{C} & -\mathbf{A}^T \end{bmatrix} + \begin{bmatrix} \mathbf{B}\mathbf{R}_\gamma^{-1} & 0 \\ -\mathbf{C}^T\mathbf{D} & -\mathbf{C}^T \end{bmatrix} \mathbf{R}_\gamma^{-1} [\mathbf{D}^T\mathbf{C} \quad \mathbf{B}^T]^T, \quad (3.64)$$

unde $\mathbf{R}_\gamma = \mathbf{R}_\gamma^T > 0$ este definită ca:

$$\mathbf{R}_\gamma = \gamma^2\mathbf{I} - \mathbf{D}^T\mathbf{D} \quad (3.65)$$

În cazul obișnuit în care $\mathbf{D} = 0$, matricea hamiltoniană \mathbf{H} se reduce la:

$$\mathbf{H} = \begin{bmatrix} \mathbf{A} & \frac{1}{\gamma^2}\mathbf{B}\mathbf{B}^T \\ -\mathbf{C}^T\mathbf{C} & -\mathbf{A}^T \end{bmatrix}. \quad (3.66)$$

3.3.2. Proiectarea Bazată pe *Mixed Sensivity*

Mai specific, să considerăm Figura 3.9 și să definim vectorul de ieșire de performanță $\mathbf{z}(s)$ astfel:

$$\mathbf{z}(s) := \begin{bmatrix} z_1(s) \\ z_2(s) \end{bmatrix} = \begin{bmatrix} W_e(s)(r(s) - y_p(s)) \\ W_u(s)u(s) \end{bmatrix}. \quad (3.67)$$

Vectorul de intrare \mathbf{w} format din intrări exogene este definit astfel:

$$\mathbf{w} := r, \quad (3.68)$$

iar Figura 3.9 poate fi rearanjată, așa cum este prezentat în Figura 3.10.

Pentru calculul erorii de performanță $r(s) - y_p(s)$ din Figura 3.9 calculăm $y_p(s)$ pentru sistemul nominal $F_n(s)$ (presupunând că nu există incertitudini $\Delta(s) = 0$):

$$\Delta(s) = 0 \Rightarrow y_p(s) = F_n(s)C(s)(r(s) - y_p(s)) \quad (3.55)$$

$$\Leftrightarrow y_p(s) = (\mathbf{I} + F_n(s)C(s))^{-1} F_n(s)C(s)r(s), \quad (3.69)$$

recunoaștem în $(\mathbf{I} + F_n(s)C(s))^{-1} F_n(s)C(s)$ matricea de sensibilitate complementară $\mathbf{T}(s)$:

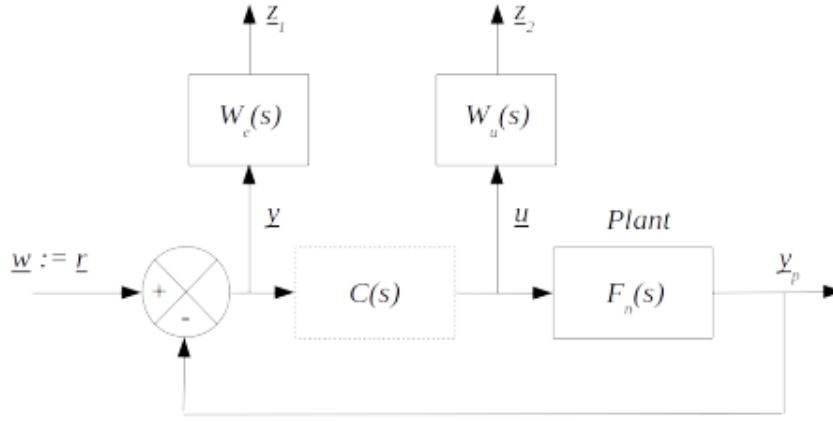


Figura 3.9: Structura de reglare cu un singur grad de libertate bazată pe filtre de ponderare [30].

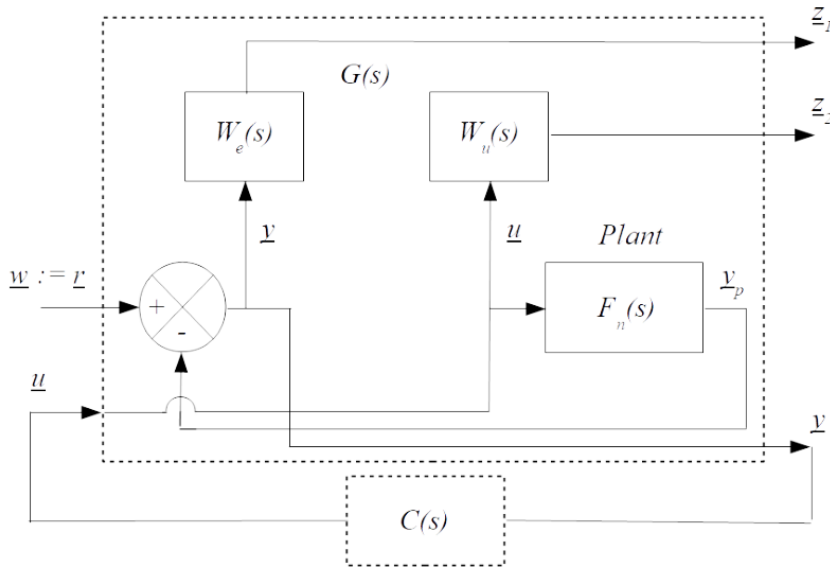


Figura 3.10: Augmentarea plant-ului nominal $F(s)$ [30].

$$\mathbf{T}(s) = (\mathbf{I} + F_n(s)C(s))^{-1} F_n(s)C(s) \quad (3.70)$$

$$y_p(s) = \mathbf{T}(s)r(s), \quad (3.71)$$

folosind relația $\mathbf{S}(s) + \mathbf{T}(s) = \mathbf{I}$ între matricea de sensibilitate $\mathbf{S}(s)$ și matricea de sensibilitate complementară $\mathbf{T}(s)$, obținem în final prima componentă $r(s) - y_p(s)$ a vectorului de ieșire $\mathbf{z}(s)$:

$$\mathbf{S}(s) + \mathbf{T}(s) = \mathbf{I} \Rightarrow r(s) - y_p(s) = (\mathbf{I} - \mathbf{T}(s))r(s) = \mathbf{S}(s)r(s). \quad (3.72)$$

În final, eroarea de performanță $r(s) - y_p(s)$ devine:

$$r(s) - y_p(s) = \mathbf{S}(s)r(s), \quad (3.73)$$

mai mult, din relațiile (3.71) și $\mathbf{S}(s) + \mathbf{T}(s) = \mathbf{I}$ avem:

$$u(s) = C(s) (r(s) - y_p(s)) \quad (3.56a)$$

$$= C(s) (r(s) - \mathbf{T}(s)r(s)) \quad (3.56b)$$

$$= C(s)\mathbf{S}(s)r(s), \quad (3.74)$$

astfel, relația (3.67) devine:

$$\mathbf{z}(s) := \begin{bmatrix} z_1(s) \\ z_2(s) \end{bmatrix} = \begin{bmatrix} W_e(s) (r(s) - y_p(s)) \\ W_u(s)u(s) \end{bmatrix} = \begin{bmatrix} W_e(s)\mathbf{S}(s) \\ W_u(s)C(s)\mathbf{S}(s) \end{bmatrix} r(s). \quad (3.75)$$

Relația precedentă este rescrisă astfel, unde matricea $T_{zw}(s)$ este matricea de transfer între vectorul de ieșire $\mathbf{z}(s)$ și vectorul de intrare $\mathbf{w} := r$ a sistemului nominal $F_n(s)$:

$$\mathbf{z}(s) = T_{zw}(s)\mathbf{r}(s) \text{ unde } T_{zw}(s) := \begin{bmatrix} W_e(s)\mathbf{S}(s) \\ W_u(s)C(s)\mathbf{S}(s) \end{bmatrix}. \quad (3.76)$$

3.3.3. Concluzii și Specificații de Proiectare

Ca o concluzie, specificațiile de proiectare ce presupun incertitudini aditive implică faptul că norma matricelor $\mathbf{S}(s)$ și $C(s)\mathbf{S}(s)$ (și posibil $\mathbf{T}(s)$) ar trebui să fie minimizată. Astfel, problema de control robust a unui sistem ce prezintă incertitudini aditive se rezumă la o problemă de minimizare a normei (fie H_2 sau H_∞) aplicată funcției de transfer $T_{zw}(s)$:

$$\mathbf{F}(s) = F_n(s) + W_u(s)\Delta(s) \Rightarrow T_{zw}(s) = \begin{bmatrix} W_e(s)\mathbf{S}(s) \\ W_u(s)C(s)\mathbf{S}(s) \end{bmatrix}. \quad (3.77)$$

Cu alte cuvinte, pentru a asigura un robustețe a regulatorului, următoarele afirmații trebuie să fie îndeplinite:

- Pentru stabilitatea sistemului incert în buclă închisă, norma produsului $W_u(s)C(s)\mathbf{S}(s)$ este necesar să fie cât mai mică presupunând că există *incertitudini aditive*.

- Pentru o bună urmărire a traiectoriei și respingerea perturbațiilor, norma matricei $W_e(s)\mathbf{S}(s)$ este necesar să fie mică.

Alte specificații de performanță, cum ar fi respingerea perturbațiilor și atenuarea zgomotului senzorilor, pot fi adăugate. Se poate demonstra că pentru a rămâne insensibil la zgomotul senzorilor, norma matricei de sensibilitate complementare $\mathbf{T}(s)$ trebuie să fie mică.

Filtrele de ponderare $W_e(s)$ și $W_u(s)$ sunt parametri de ajustare care ajută la impunerea mărimilor de performanță a sistemului în buclă închisă. Acestea sunt selectate pentru a reflecta dependența de frecvență a semnalelor și a specificațiilor de performanță. Sunt definite ca funcții de transfer minime, stabile și de fază minimă (i.e., fără poli sau zerouri în semiplanul drept). De obicei, $W_e(s)$ este ales ca un filtru trece-jos, permițând ca intrările de referință de frecvență joasă să fie luate în considerare pentru urmărire, în timp ce $W_u(s)$ este ales ca un filtru trece-sus, permițând dinamicii necomandate de frecvență înaltă să fie luate în considerare.

Un punct de plecare bun este să se aleagă

$$\begin{cases} W_e(s) = \frac{s/M_1 + \omega_1}{s + \omega_1} \\ W_u(s) = \frac{1}{M_2} \text{ sau } \frac{s + \omega_2/M_2}{\varepsilon s + \omega_2} \end{cases} \quad \text{unde} \quad \begin{cases} M_1 = M_2 := M \\ \omega_1 = \omega_2 := \omega_0 \\ \varepsilon_1 = \varepsilon_2 := \varepsilon \\ 0 < \varepsilon < 1 \end{cases}, \quad (3.78)$$

iar paramentru ε este cel mai mare offset admisibil față de punctul de echilibru a sistemului în buclă închisă, ω_0 este lățimea de bandă dorită, iar M este vârful sensibilității. $W_u^{-1}(s)$ este o limită superioară a funcției de sensibilitate dorită $\mathbf{S}(s)$, în timp ce $W_u^{-1}(s)$ va limita mărimea vectorului de control u .

- Pentru problema perturbației numărător-numitor a sistemului SISO o condiție suficientă pentru stabilitatea robustă este următoarea:

$$\left\| \begin{bmatrix} W_n(s) \frac{W_p(s)}{D_n(s)} C(s) \mathbf{S}(s) \\ W_d(s) \frac{W_p(s)}{D_n(s)} \mathbf{S}(s) \end{bmatrix} \right\|_\infty < 1. \quad (3.79)$$

Comparând relația precedentă cu (3.77), se poate observa că această problemă tinde să fie echivalentă cu problema de control robust cu sensibilitate mixtă, presupunând incertitudini aditive prin următoarele alegeri de filtre de ponderare $W_e(s)$ și $W_u(s)$:

$$\begin{cases} W_e(s) := \frac{W_d(s)W_p(s)}{D_n(s)} \\ W_u(s) := \frac{W_n(s)W_p(s)}{D_n(s)} \end{cases}, \quad (3.80)$$

unde $D_n(s)$ reprezintă numitorul funcției de transfer $F_n(s)$.

În cele din urmă, problema de control robust, constă în minimizarea normei H_∞ a următoarei funcții de transfer $T_{zw}(s)$:

$$\mathbf{F}(s) = (\mathbf{I} + \Delta(s)W_u(s))F_n(s) \Rightarrow T_{zw}(s) = \begin{bmatrix} W_e(s)\mathbf{S}(s) \\ W_u(s)T(s) \end{bmatrix}, \quad (3.82)$$

unde $T(s)$ este funcția de sensibilitate complementară:

$$T(s) = \mathbf{S}(s)F_n(s)C(s), \quad (3.83)$$

iar pentru această problemă de minimizare menționată mai sus se poate folosi un algoritm de tipul celui prezentat în 2.

Pentru a realiza proiectarea regulatorului care să minimizeze următoarea matrice de transfer:

$$T_{zw}(s) = \begin{bmatrix} W_e(s)\mathbf{S}(s) \\ W_u(s)C(s)\mathbf{S}(s) \end{bmatrix}, \quad (3.84)$$

unde $\mathbf{S}(s)$ este funcția de sensibilitate:

$$\mathbf{S}(s) = (\mathbf{I} + F_n(s)C(s))^{-1}, \quad (3.85)$$

se va realiza reprezentarea în spațiul stărilor a sistemului augmentat în buclă deschisă $\mathbf{G}(s)$

$$\begin{bmatrix} \mathbf{z}(s) \\ \mathbf{y}(s) \end{bmatrix} = \mathbf{G}(s) \begin{bmatrix} \mathbf{w}(s) \\ \mathbf{u}(s) \end{bmatrix} \quad (3.86)$$

Algorithm 2 Aproximarea limitei optime: Algoritmul de iterație- γ

```

1: function OPTIM_GAMMA( $A, B, C, \varepsilon$ )
2:    $\gamma \leftarrow 0$ 
3:    $\bar{\gamma} \leftarrow \|T_{zw}(s)\|_\infty$  obținut cu  $\gamma = \infty$  în Ecuația Riccati Algebrică
4:   while  $\bar{\gamma} - \gamma > \varepsilon$  do
5:     Calculează matricea Hamiltoniană:

```

$$\mathbf{H} = \begin{bmatrix} \mathbf{A} & \frac{1}{\gamma^2} \mathbf{B} \mathbf{B}^T \\ -\mathbf{C}^T \mathbf{C} & -\mathbf{A}^T \end{bmatrix}$$

```

6:     if  $\mathbf{H}$  nu are valori proprii pe axa imaginară then
7:        $\bar{\gamma} \leftarrow \frac{\bar{\gamma} + \gamma}{2}$ 
8:     else
9:        $\gamma \leftarrow \frac{\bar{\gamma} + \gamma}{2}$ 
10:    end if
11:  end while
12:  return  $\gamma$ 
13: end function

```

$$\begin{bmatrix} z_1(s) \\ z_2(s) \\ y(s) \end{bmatrix} = \begin{bmatrix} W_e(s) & -W_e(s)F_n(s) \\ 0 & W_u(s) \\ \mathbf{I} & -F_n(s) \end{bmatrix} \begin{bmatrix} r(s) \\ u(s) \end{bmatrix}; \quad (3.87)$$

Reprezentarea sub forma unei funcții de transfer a sistemului în buclă deschisă $\mathbf{G}(s)$ poate fi obținută astfel:

- În primul rând, se reprezintă face reprezentarea în spațiul stărilor al sistemului augmentat:

$$\begin{cases} \dot{x}_p(t) = A_p x_p(t) + B_p u(t) \\ y_p(t) = C_p x_p(t) \end{cases}; \quad (3.88)$$

- În al doilea rând, folosind relația $y(t) = r(t) - C_p x_p(t)$, reprezentarea în spațiul stărilor a filtrului $W_e(s)$ este următoarea:

$$\begin{cases} \dot{x}_e(t) = A_e x_e(t) + B_e(r(t) - C_p x_p(t)) \\ z_1(t) = C_e x_e(t) + D_e(r(t) - C_p x_p(t)) \end{cases} \quad (3.89)$$

- În al treilea rând, reprezentarea în spațiul stărilor a filtrului $W_u(s)$ este următoarea:

$$\begin{cases} \dot{x}_u(t) = A_u x_u(t) + B_u u(t) \\ z_2(t) = C_u x_u(t) + D_u u(t) \end{cases}, \quad (3.90)$$

astfel, vectorul de stare a plant-ului augmentat este următorul:

$$x(t) = \begin{bmatrix} x_p(t) \\ x_e(t) \\ x_u(t) \end{bmatrix} \quad (3.91)$$

Vectorul de intrare exogenă $w(t)$ este definit după cum urmează:

$$w(t) := r(t), \quad (3.92)$$

vectorul de ieșire $y(t)$ disponibil pentru regulator este următorul:

$$y(t) = r(t) - C_p x_p(t) = [r(t) + [-C_p \ 0 \ 0]x(t)], \quad (3.93)$$

iar vectorul de ieșire $z(t)$ este următorul:

$$z(t) = \begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix} = \begin{bmatrix} -D_e C_p & C_e & 0 \\ 0 & 0 & C_u \end{bmatrix} x(t) + \begin{bmatrix} D_e & 0 \\ 0 & D_u \end{bmatrix} \begin{bmatrix} r(t) \\ u(t) \end{bmatrix}. \quad (3.94)$$

În cele din urmă se obține:

$$\begin{cases} \dot{x}(t) = \mathbf{A}x(t) + \mathbf{B}_1 w(t) + \mathbf{B}_2 u(t) \\ z(t) = \mathbf{C}_1 x(t) + \mathbf{D}_{11} w(t) + \mathbf{D}_{12} u(t) \\ y(t) = \mathbf{C}_2 x(t) + \mathbf{D}_{21} u(t) \end{cases}, \quad (3.95)$$

unde matricile de pe prima linie au următoarea expresie:

$$\mathbf{A} = \begin{bmatrix} A_p & 0 & 0 \\ -B_e C_p & A_e & 0 \\ 0 & 0 & A_u \end{bmatrix} \quad \mathbf{B}_1 = \begin{bmatrix} 0 \\ B_e \\ 0 \end{bmatrix} \quad \mathbf{B}_2 = \begin{bmatrix} B_p \\ 0 \\ B_u \end{bmatrix}, \quad (3.96)$$

cele utilizate în a doua linie au forma:

$$\mathbf{C}_1 = \begin{bmatrix} -D_e C_p & C_e & 0 \\ 0 & 0 & C_u \end{bmatrix} \quad \mathbf{D}_{11} = \begin{bmatrix} D_e \\ 0 \end{bmatrix} \quad \mathbf{D}_{12} = \begin{bmatrix} 0 \\ D_u \end{bmatrix}, \quad (3.97)$$

iar matricile utilizate în a treia linie au următoarea expresie:

$$\mathbf{C}_2 = [-C_p \ 0 \ 0] \quad \mathbf{D}_{21} = \mathbf{I}. \quad (3.98)$$

3.4. Proiectarea Unui Regulator Neuronal pe Baza Modelului NARMA-L2

Un sistem de control discret poate fi reprezentat în spațiul stărilor astfel:

$$x(k+1) = f[x(k), u(k)], \quad (3.96)$$

$$y(k) = g[x(k)], \quad (3.97)$$

unde starea $x(k) \in \mathbb{R}^k$, intrarea $u(k) \in \mathbb{R}^m$, iar ieșirea $y(k) \in \mathbb{R}^n$. Funcțiile $f : \mathbb{R}^m \times \mathbb{R}^k \rightarrow \mathbb{R}^k$ și $g : \mathbb{R}^k \rightarrow \mathbb{R}^n$ sunt mapările de la starea curentă și intrare la starea următoare, respectiv de la starea curentă la ieșire, care sunt ambele necunoscute în acest scenariu.

Un sistem liniar discret poate fi reprezentat după cum urmează, reprezentare ce poate fi considerată o liniarizare a ecuației (3.96):

$$x(k+1) = Ax(k) + Bu(k), \quad (3.98)$$

$$y(k) = Cx(k). \quad (3.99)$$

3.4.1. Modelul NARMA

După cum este discutat în [31], un sistem SISO reprezentat ca în (3.96) poate fi rescris astfel:

$$y(k+1) = F_n[y(k), y(k-1), \dots, y(k-n+1), u(k), u(k-1), \dots, u(k-n+1)], \quad (3.100)$$

unde F_n este o mapare astfel încât $F_n : \mathbb{R}^{2n} \rightarrow \mathbb{R}$, iar (3.98) poate fi generalizată pentru a avea următoarea formă:

$$y(k+d) = F_n[y(k), y(k-1), \dots, y(k-n_y+1), u(k), u(k-1), \dots, u(k-n_u+1)]. \quad (3.101)$$

3.4.2. Modelul NARMA-L2

În teorie, modelul NARMA poate fi utilizat direct pentru identificarea sistemului folosind o rețea neuronală pentru a aproxima F_n . Această metodă poate obține o precizie mai mare în identificare, dar este dificil să se calculeze $u(k)$ dorit în proiectarea regulatorului adaptiv, deoarece conform Ecuației (3.101), controlul necesar pentru a obține referința dorită $y^*(k)$ este:

$$u(k) = H_n[y(k), y(k-1), \dots, y(k-n_y+1), y^*(k+d), u(k), u(k-1), \dots, u(k-n_u+1)], \quad (3.102)$$

unde $u(k)$ este o funcție neliniară și trebuie aproximat printr-o rețea neuronală mai degrabă decât prin calcul algebric. De aceea, un model fezabil, NARMA-L2, este propus în [32] pentru a simplifica proiectarea regulatorului și a accelera procesul de antrenare. Ecuația (3.102) poate fi derivată separând $u(k)$ și aplicând dezvoltarea în serie Taylor într-un punct de echilibru u_0 astfel:

$$y(k+d) = F_n[y(k), y(k-1), \dots, y(k-n_y+1), u(k), u(k-1), \dots, u(k-n_u+1)] = F_n[\phi(k), u(k)] \quad (3.103)$$

$$= G_n[\phi(k), u_0(k)] + \left. \frac{\partial G_n}{\partial u(k)} \right|_{\phi(k), u_0(k)} (u(k) - u_0(k)) + R_1(k) \quad (3.104)$$

$$\Rightarrow \{G_n[\phi(k), u_0(k)] + \left. \frac{\partial G_n}{\partial u(k)} \right|_{\phi(k), u_0(k)} u_0(k)\} + \left. \frac{\partial G_n}{\partial u(k)} \right|_{\phi(k), u_0(k)} u(k) \quad (3.105)$$

$$= f_0[\phi(k)] + g_0[\phi(k)]u(k), \quad (3.106)$$

unde f_0 și g_0 sunt două funcții neliniare care pot fi approximate de două rețele neuronale separate așa cum se vede și în Figura 3.11

Proiectarea regulatorului neuronal pe baza modelului NARMA-L2 poate fi realizată mult mai ușor deoarece poate fi reprezentată prin calcul algebric după cum urmează:

$$u(k) = \frac{y^*(k+d) - f_0[\phi(k)]}{g_0[\phi(k)]}, \quad (3.107)$$

unde $\phi(k) = [y(k), y(k-1), \dots, y(k-n_y+1), u(k-1), \dots, u(k-n_u+1)]$. Deoarece $u(k)$

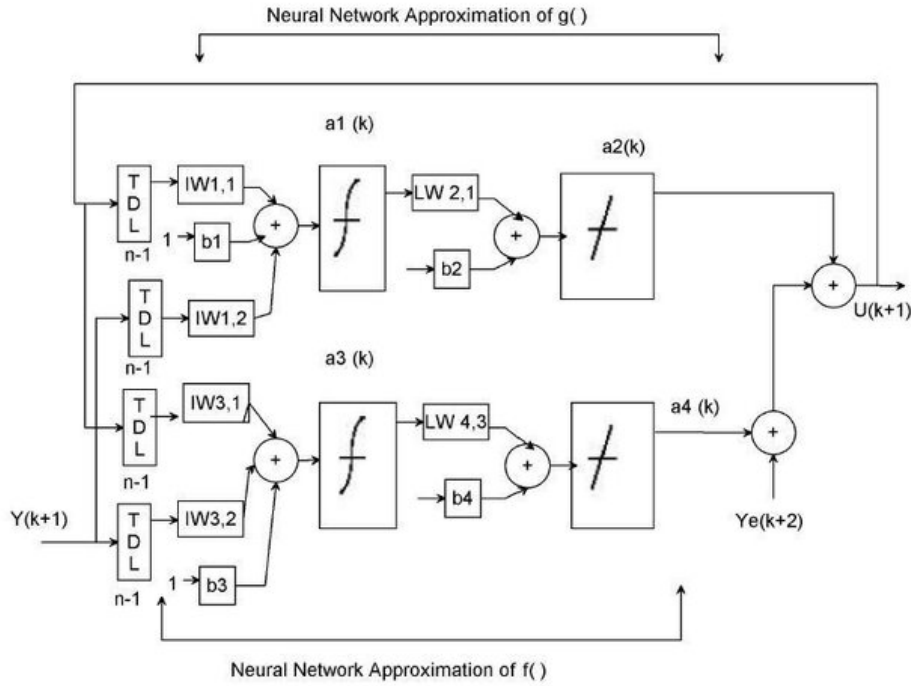


Figura 3.11: Modelul NARMA L2.

este determinat de $y(k)$, dar ieșirea în același timp, este dificil de implementat Ecuația (3.107) în realitate. Prin urmare, Ecuația (3.107) este modificată în următoarea formă:

$$u(k+1) = \frac{y^*(k+d) - f_0[\phi'(k)]}{g_0[\phi'(k)]}, \quad (3.108)$$

unde $\phi'(k) = [y(k), y(k-1), \dots, y(k-n_y+1), u(k-1), \dots, u(k-n_u+1)]$.

3.4.3. Îmbunătățirea Performanței Regulatorului

Regulatorul NARMA-L2 este capabil să genereze intrarea sistemului care va face sistemul de control să urmărească referința. Cu toate acestea, regulatorul poate garanta doar ieșirea sistemului, în timp ce intrarea de control nu este garantată. Intrarea de control generată poate avea fluctuații semnificative, ce sunt cauzate de schimbările bruște a intrării de referință.

Pentru a îmbunătăți performanța regulatorului, se adaugă întârzieri de timp și se face o liniarizare de la stare:

$$Y(z) = \frac{c_0}{D(z)} Y^*(z), \quad (3.102)$$

unde

$$D(z) = 1 + d_1 z^{-1} + d_2 z^{-2} + \dots + d_n z^{-n}. \quad (3.103)$$

Conform ecuației (3.102), eroarea de stare stabilă poate fi eliminată dacă:

$$c_0 = 1 + d_1 + d_2 + \dots + d_n. \quad (3.104)$$

În continuare voi prezenta cei doi algoritmi ce pot fi folosiți în proiectarea regu-

laturului, unul pentru identificarea sistemului și celălalt pentru antrenarea regulatorului neuronal.

Algorithm 3 Identificarea modelului NARMA-L2

```

1: Intrare: Întârzierea în timp a intrărilor și ieșirilor  $\phi(k) = [y(k), y(k-n+1), u(k), \dots, u(k-n+1)]$ 
2: Ieșire: Rețeaua neuronală reprezentată ca  $f$  și  $g$ 
3:  $N \leftarrow$  numărul de date,  $\text{epoch} \leftarrow$  durata unei iterații,  $k \leftarrow 0$ ,  $i \leftarrow 0$ ,  $\text{lr} \leftarrow$  rata de învățare
4:  $X(\text{strat/neuron}) \leftarrow$  ponderile rețelei neuronale într-un singur neuron într-un anumit strat.  $X$  reprezintă  $f$  sau  $g$ .
5: while  $k \leq N$  do
6:   while  $i < \text{epoch}$  do
7:      $yr(k) \leftarrow f[\phi(k)] + g[\phi(k)]u(k)$ 
8:      $e(k) \leftarrow$  Eroarea Medie Pătratică( $yr(k) - \text{plant}(u(k))$ )
9:     for  $n$  în straturile lui  $f$  do
10:      for  $l$  în  $1$  do
11:         $f(l)(n) \leftarrow f(l)(n) - \text{lr} \cdot \text{diff}(e(k), f(l)(n))$ 
12:      end for
13:    end for
14:    for  $n$  în straturile lui  $g$  do
15:      for  $l$  în  $1$  do
16:         $g(l)(n) \leftarrow g(l)(n) - \text{lr} \cdot \text{diff}(e(k), g(l)(n))$ 
17:      end for
18:    end for
19:     $i \leftarrow i + 1$ 
20:  end while
21:   $k \leftarrow k + 1$ 
22: end while
23: Returnează  $f, g$ 
    
```

Algorithm 4 Proiectarea regulatorului NARMA-L2

```

1: Intrare: Referința ieșirii  $y^*(k)$ ,  $f$  și  $g$ 
2: Ieșire: Controlul intrării  $u(k)$ , ieșirea rețelei  $y(k)$ 
3:  $N \leftarrow$  numărul de puncte de date,  $k \leftarrow 0$ ,  $[u(0), \dots, u(n)] \leftarrow 0$ ,  $[y(0), \dots, y(n)] \leftarrow 0$ 
4:  $c \leftarrow 1 + d_1 + d_2$ 
5: while  $k \leq N$  do
6:    $yr(k) \leftarrow \text{plant}(u(k))$ 
7:    $\phi(k) \leftarrow [yr(k), \dots, yr(k-n+1), u(k), \dots, u(k-n+1)]$ 
8:    $u(k+1) \leftarrow [c \cdot y^*(k+2) - f(\phi(k)) - d_1 \cdot y(k) - d_2 \cdot y(k-1)] / g(\phi(k))$ 
9:    $k \leftarrow k + 1$ 
10: end while
11: Returnează  $u, y$ 
    
```

Capitolul 4. Implementare și Testare

Implementarea modelului propus cât și a reguletoarelor descrise în capitolul 3 se va realiza în mediul MATLAB/Simulink. În acest capitol se va prezenta modul în care fiecare componentă descrisă anterior va fi implementată, respectiv cum se pot utiliza diverse librării specifice controlului robust, dar și cel neuronal pentru a putea realiza sinteza de reguletoare. Mai mult de atât acest capitol tratează și prezintă rezultatele și simulările obținute în urma aplicării tehnicilor avansate de control, rezultate ce vor fi prezentate prin diagrame și tabele, iar în final se va face o comparație între aceste metode pe baza rezultatelor obținute.

4.1. Impelementarea Modelelor pentru Motoare

4.1.1. Motorul Electric

La baza implementarii modelului stau ecuațiile diferențiale descrise în capitolul anterior, mai exact ecuațiile (3.1) și (3.2), respectiv realizarea în spațiul stărilor descrisă de (3.3) pe baza căreia folosind blocul *Integrator* din mediul simulink s-au implementat ecuațiile diferențiale asociate fiecărei componente dinamice din circuitul echivalent.

Există totuși o modificare față de modelul clasic al unui Motor Electric, pe lângă intrare de Tensiune al subsistemului, considerăm și turația motorului ca fiind intrare după cum se observă și în Figura 4.1, spre deosebire de modelul clasic în care turația este ieșire din model. Această adaptare a fost făcută deoarece luăm în considerare și dinamica sistemului final și atunci este mai ușor să calculăm turația motorului din modelul dinamic și să o transmitem către modelul motorului electric ca și intrare pentru reușii să captăm mai bine interacțiunea dintre motor și forțele exterioare care influențează turația acestuia.

Prin urmare, modelul propus permite o mai bună integrare cu dinamica sistemului global din care motorul electric face parte. De exemplu într-un vehicul, turația motorului poate fi afectată de panta drumului, greutatea vehiculului sau de forțele aerodinamice ce la viteze mai mari nu pot fi neglijate.

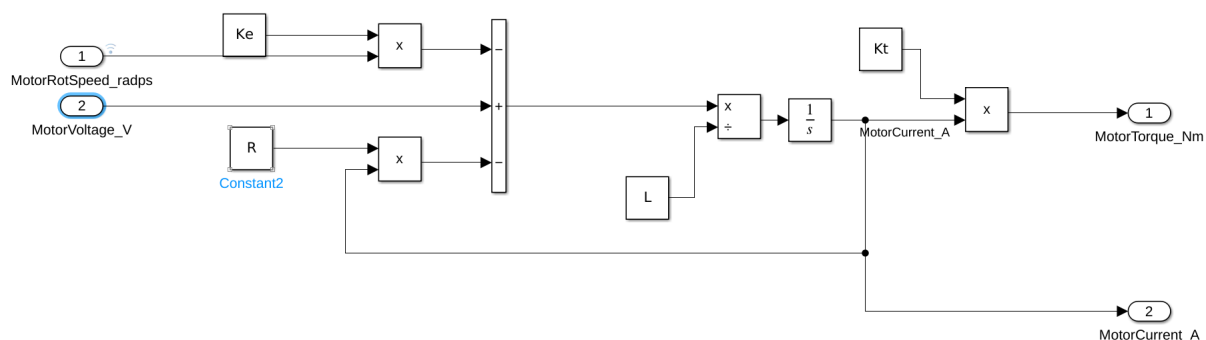


Figura 4.1: Modelul Motorului Electric.

Considerăm următoarea parametrizare a modelului prezentată în tabelul 4.1 și se aplică inițial o treaptă de 52V la intrarea modelui, urmând ca după 20 de secunde să se aplice o variație a comenzii cu +12V față de comanda inițială, în urma cărui test se obține răspunsul în Curent din Figura 4.2.

Parametru	Simbol	Valoare
Constanta cuplului motorului	K_t	0.668
Inductanța	L	0.01 H
Rezistența	R	0.141176 Ω
Constanta de tensiune inversă	K_e	0.1178

Tabela 4.1: Parametrii motorului electric

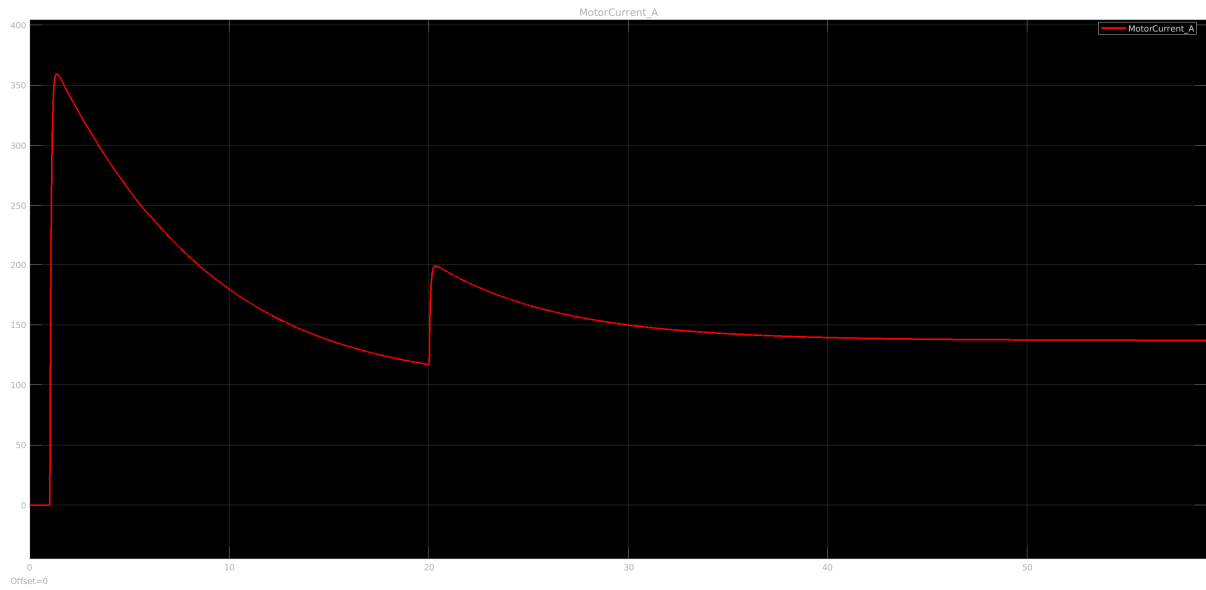


Figura 4.2: Variația Curentului Motorului Electric.

4.1.2. Motorul pe Combustie Internă

În ceea ce privește motorul pe combustie internă în această teză se va expune un model simplificat și va fi reprezentat printr-o funcție de transfer conform ecuației (3.20), deoarece pe întreg parcursul tezei vom considera că motorul funcționează doar pe modurile Stop/Start, ignorând etapele intermediare. Un astfel de model poate fi implementat tot cu ajutorul blocurilor integratoare folosindu-ne de forma canonică a modelului reprezentat printr-o funcție de transfer, acest lucru îl facem pentru a putea impune eventuale condiții inițiale ale modelului și implementarea acestuia este ilustrată în Figura 4.3.

4.2. Implementarea Modelului Bateriei

La baza implementării stă modelul de circuit echivalent Thevenin prezentat în Figura 3.4 cât și ecuațiile (3.13) și (3.14). Pentru caracteristica SOC-OCV s-a folosit un Look-Up table. Considerăm că ansamblu de baterii are 96V cu o capacitate de 83Wh, iar fiecare celulă având o tensiune nominală de 3.6V. În final pentru a obține 96V considerăm o configurație în serie cu 26 de celule. Un astfel de model este prezentat în Figura 4.4

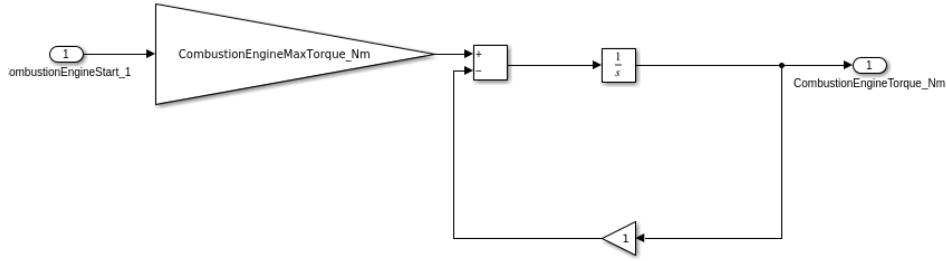


Figura 4.3: Implementarea Modelului Motorului pe Combustie Internă

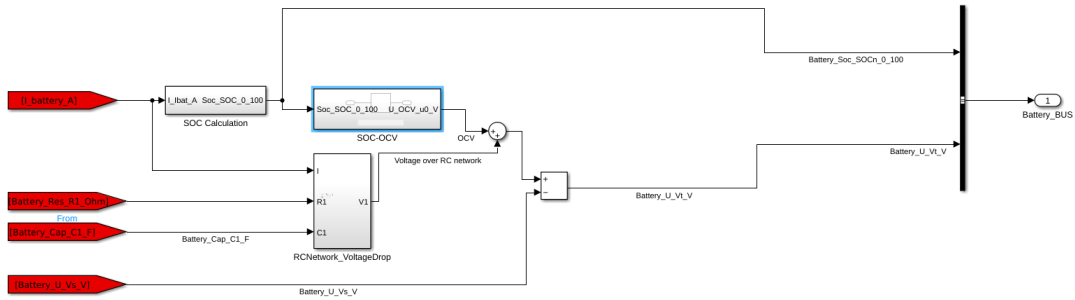


Figura 4.4: Modelul Thevenin de ordin 1 al bateriei

Totuși având în vedere că scopul final este de a controla dinamica sistemului din punct de vedere al vitezei, putem neglija pierderile de tensiune pe circuitul RC al modelului bateriei, deoarece necesită o parametrizare suplimentară, iar valorile pentru parametrii circuitului RC pentru a avea sens sunt determinați pe cale experimentală. Astfel în continuare vom folosi modelul Rint al bateriei care ține cont doar de calculul SOC și caracteristica SOC-OCV ce poate fi vizualizată în Figura 4.6, iar modelul simplificat este prezentat în Figura 4.5.

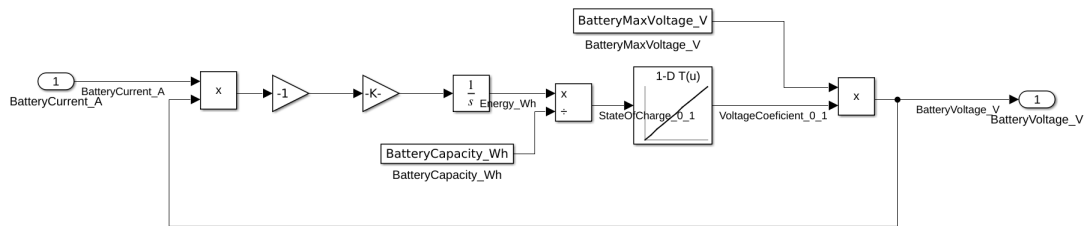


Figura 4.5: Modelul Rint al bateriei.

Pentru a testa și valida modul de funcționare al bateriei, s-a aplicat ciclic curenți de încărcare și descărcare și s-a monitorizat modul în care bateria se încarcă respectiv se descarcă în urma stimulării după cum se vede și în figura 4.7.

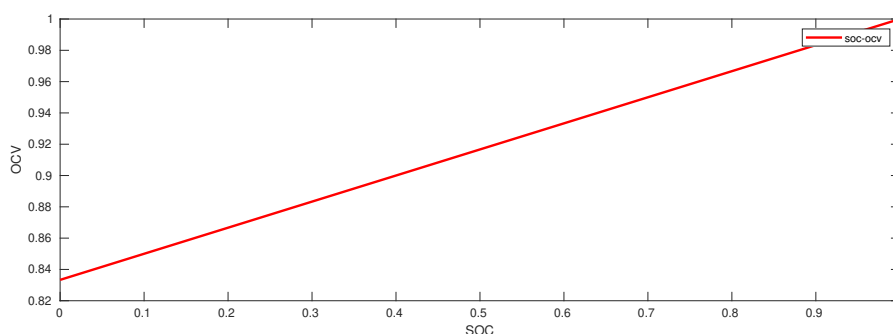


Figura 4.6: Caracteristica SOC-OCV.

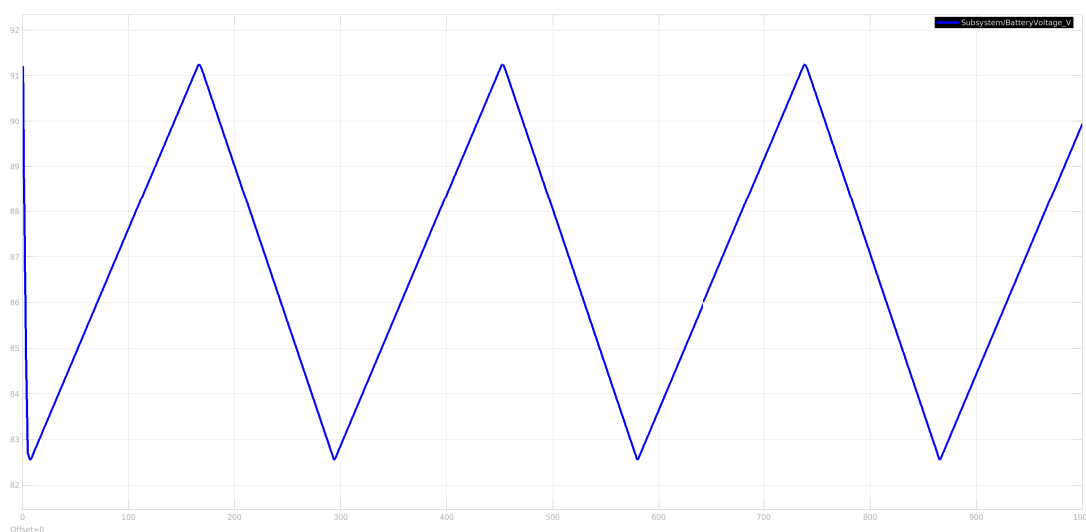


Figura 4.7: Ciclul de încărcare-descărcare a bateriei.

4.3. Implementarea Modelului Dinamic

Pentru modelul dinamic luăm în considerare mișcarea rectilinie și presupunem că deplasarea ansamblului se face pe plan drept, astfel nu vom lua în calcul forța de gravitație ce ar acționa dacă mișcarea s-ar realiza pe un plan înclinat, modelate fiind doar forțele descriese de ecuațiile (3.24), (3.25) fiind implementate în simulink așa cum se arată în Figura 4.8.

Pentru determinarea vitezei se folosește legea a doua a lui Newton, ecuație descrisă de (3.22), iar din (3.26) se va determina forța rezultantă din cuplul total, și egalând cele două ecuații putem deduce modelul prezentat în Figura 4.9 și la fel print-un bloc integrator implementăm ecuația diferențială obținută. Așa cum am menționat în 4.1.1 turația finală a motorului se va deduce din relația dintre viteza unghiulară a motorului și viteza liniară a vehiculului ce este exprimată de ecuația (3.27) și implementată în Simulink, iar parametrii modelului dinamic sunt expuși în tabelul 4.2

4.4. Controlul Sistemului

Scopul final este de a aplica tehnici de control avansat pentru a controla viteza, fapt ce la prima vedere pare a fi o problemă relativ simplă, dar fiind vorba de un sistem complex în esență ce își schimbă dinamica în timp și poate prezenta incertitudini aditive, are un caracter neliniar și prin prezența celor două motoare unul electric și unul mecanic, poate

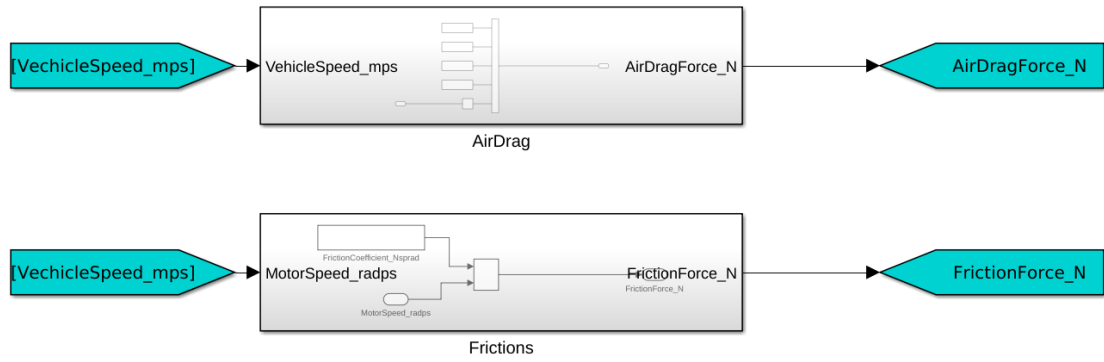


Figura 4.8: Modelul Forțelor Exterioare.

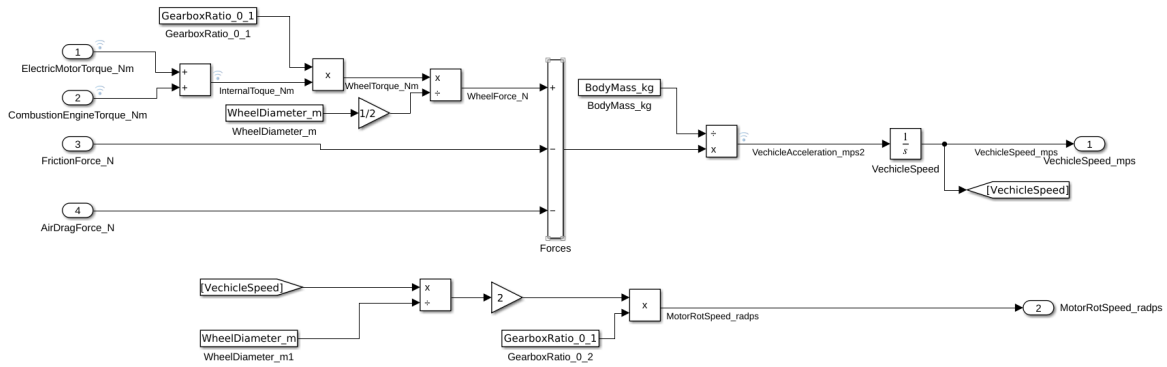


Figura 4.9: Modelul Dinamic

Parametru	Valoare	Unitate
Masa	900	kg
Raport Roată Dințată	3	-
Diametrul Roții Vehicului	0.5	m
Coeficientul Rezistenței la aer	0.4	-
Aria Rezistenței la Aer	4.5	m ²
Densitate Aer	1.225	kg/m ³
Coeficient de Frecare	0.01	N·s/rad

Tabela 4.2: Parametrii modelului dinamic.

fi privit ca un *switched system* (sisteme ce se schimbă între ele în timp și nu putem să garantăm tot timpul o stabilitate asimptotică globală), se dovedește că pentru a controla un astfel de sistem este o problemă complexă și trebuie să ținem cont de mai mulți factori.

Astfel în continuare voi detalia modul de implementare a celor trei metode de control propuse în capitolul anterior pentru a obține performanțele dorite și a asigura un control robust asupra sistemului indiferent de factorii ce acționează asupra lui.

Deoarece nu putem să controlăm direct prin tensiune un astfel de sistem a cărui implementare integrală este prezentată în Figura 4.10 vom avea nevoie de un driver de motor, o punte H, pentru a permite controlul prin setarea factorului de umplere a unui

semnal PWM. Astfel pe lângă celelalte subsisteme descrise deja s-a implementat o variantă simplificată a unei punți H în care nu se ține cont de pierderile de putere de pe mosfetiuri. Un model ce poate fi utilizat este prezentat în Figura 4.11 ce are ca intrări tensiunea bateriei, factorul de umplere, direcția de rotație a motorului respectiv curentul acestuia, iar ca ieșiri tensiunea de comandă pentru motorul electric și curentul ce va fi aplicat bateriei.

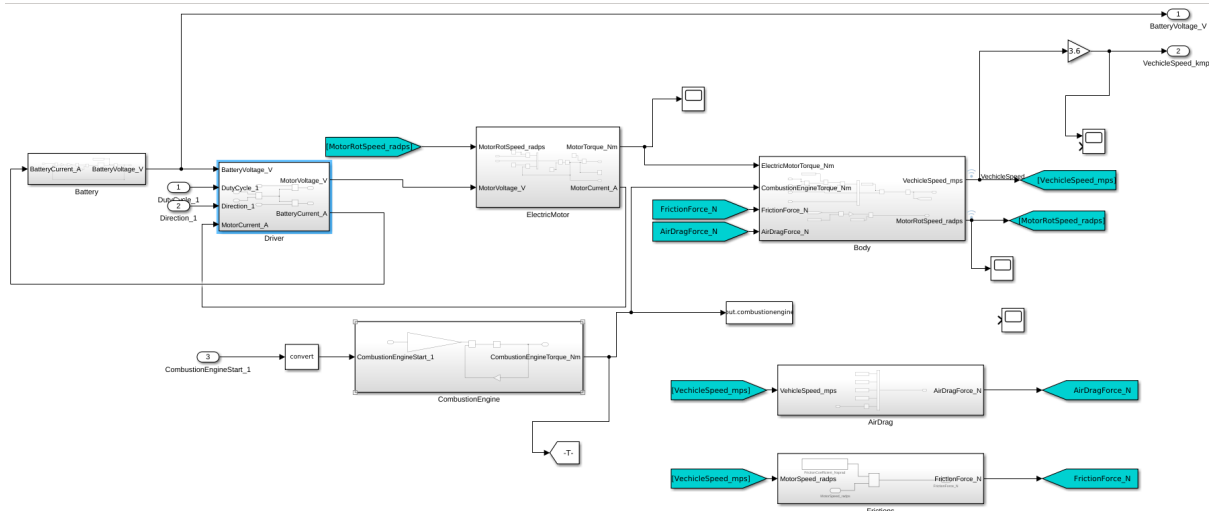


Figura 4.10: Implementarea Simulink a întregului sistem

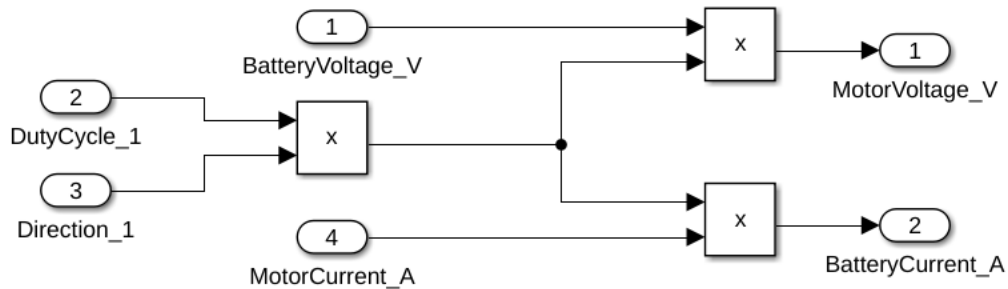


Figura 4.11: Model Driver

4.4.1. Liniarizarea Modelului

Fiind vorba de un sistem liniar și pentru a-l putea controla fie se realizează o liniarizare de la stare așa cum s-a descris în studiul bibliografic aplicând o comandă de liniarizare exprimată de ecuația (2.8) și în final în urma izolării neliniarităților obținem un set de integroare pe care putem să aplicăm tehnici clasice de control, fie se face liniarizarea în jurul unui punct de echilibru al sistemului prin dezvoltare în serie Taylor în jurul acestuia a ecuației ce introduce neliniarități, metodă pe care o vom aplica în continuare pentru a obține sistemul liniarizat.

Această liniarizare am realizat-o în Matlab folosind modelul deja implementat în simulink pe care l-am pus sub forma 4.12 și am utilizat comanda:

```
[A, B, C, D] = linmod('nume_model');
```

ce este utilizată pentru a liniariza un model Simulink în jurul unui punct de funcționare specificat. Aceasta extrage un model liniarizat sub forma unui sistem reprezentat în spațiul stărilor.

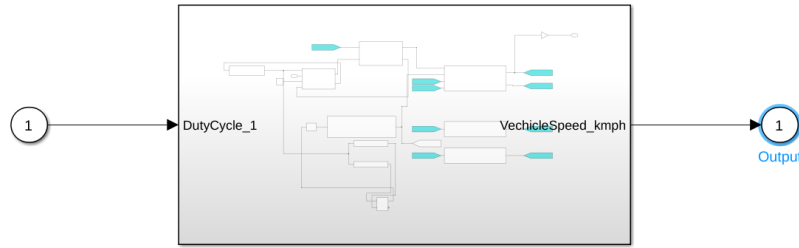


Figura 4.12: Model Utilizat Pentru Liniarizare

Se obține următorul model liniar ce are următoarea reprezentare în spațiul stărilor:

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} -1.111 \times 10^{-5} & 0.008907 \\ -141.4 & -14.12 \end{bmatrix}, \\ \mathbf{B} &= \begin{bmatrix} 0 \\ 9120 \end{bmatrix}, \\ \mathbf{C} &= [3.6 \ 0], \\ \mathbf{D} &= [0]. \end{aligned} \tag{4.1}$$

având funcția de transfer:

$$G(s) = \frac{292.4}{s^2 + 14.12s + 1.259}.$$

4.4.2. Implementarea SMC

Având deja modelul liniarizat din subsecțiunea anterioară putem proiecta un regulator pentru acest sistem folosindu-ne de ecuațiile (3.32) (3.29) (3.46) (3.48), iar rezultatele obținute vor fi implementate în simulink.

Folosind funcția de transfer descrisă anterior, cu viteza liniară ca variabilă controlată și factorul de umplere ca variabilă controlată așa cum reiese și din Figura 4.12 și aplicând transformata laplace inversă obținem următoarea ecuație diferențială în domeniul timp:

$$\ddot{v}(t) + 14.12\dot{v}(t) + 1.259v(t) = 292.4u(t), \tag{4.2}$$

facem acest lucru deoarece dorim o reprezentare sub forma canonică în spațiul stărilor a sistemului controlat

$$x_1 = v(t) \quad \text{și} \quad u = u(t). \quad (4.3)$$

Apoi, sistemul poate fi rescris în forma canonică astfel:

$$\dot{x}_1 = x_2, \quad (4.4)$$

$$\dot{x}_2 = -1.259 x_1 - 14.12 x_2 + 292.4 u, \quad (4.5)$$

$$y = x_1. \quad (4.6)$$

Acum în ecuația (3.48) dacă alegem $r=2$ obținem forma suprafeței de *sliding mode* astfel:

$$s(t) = \sigma = \lambda \dot{e} + e \quad \text{cu} \quad e = v_{\text{ref}}(t) - v(t) \quad (4.7)$$

Definim funcția Lyapunov ca

$$V = \frac{1}{2} s^2, \quad (4.8)$$

iar pentru a garanta condiția de stabilitate, derivata funcției Lyapunov din (4.8) trebuie ca $\dot{V} < 0$ de unde rezultă că:

$$s\dot{s} < 0. \quad (4.9)$$

Derivata suprafeței de *sliding mode* este:

$$\begin{aligned} \dot{s} &= c\dot{e} + \ddot{e} \\ &= c\dot{e} + v_{\text{ref}} - \dot{v} \\ &= \lambda\dot{e} + v_{\text{ref}} + 1.259\dot{v} + 14.12v - 292.4u, \end{aligned} \quad (4.9)$$

astfel, derivata funcției Lyapunov \dot{V} este:

$$s\dot{s} = s(\lambda\dot{e} + v_{\text{ref}} + 1.259\dot{v} + 14.12v - 292.4u). \quad (4.11)$$

Pentru a satisface condiția $s\dot{s} < 0$ și ținând cont de relația (3.44), alegem legea de control astfel :

$$u = \frac{1}{292.4} \left(14.12v + 1.259\dot{v} + v_{\text{ref}} + \lambda\dot{e} + \left(\frac{s}{|s| + \rho} \right) \right) \quad (4.12)$$

astfel putem identifica din (4.12):

$$u_{\text{eq}} = \frac{1}{292.4} (14.12v + 1.259\dot{v} + v_{\text{ref}} + \lambda\dot{e}) \quad (4.13)$$

$$u_{\text{sq}} = \frac{1}{292.4} \left(\frac{s}{|s| + \rho} \right) \quad (4.14)$$

Odată ce am obținut legea de control aceasta poate fi implementată în simulink astfel, ecuația (4.13) este calculată de subsistem EqCmd din 4.13, suprafața de sliding mode este implementată cu ajutorul unui bloc de matlab function, iar implementarea relației (4.14) poate fi văzută în colțul din dreapta sus a Figurii 4.13. Simularea sistemului neliniar în buclă închisă cu regulatorul SMC proiectat anterior poate fi observată în Figura 4.14. Astfel am aplicat inițial o treaptă de 63, urmată apoi la $t_1=100\text{s}$ de o treaptă de 138, iar la $t_2=150\text{s}$ o treaptă de 118. Se poate observa că nu există suprareglaj, timpul de răspuns este rapid aproximativ 4s și o urmărire bună a referinței cu o eroare staționară

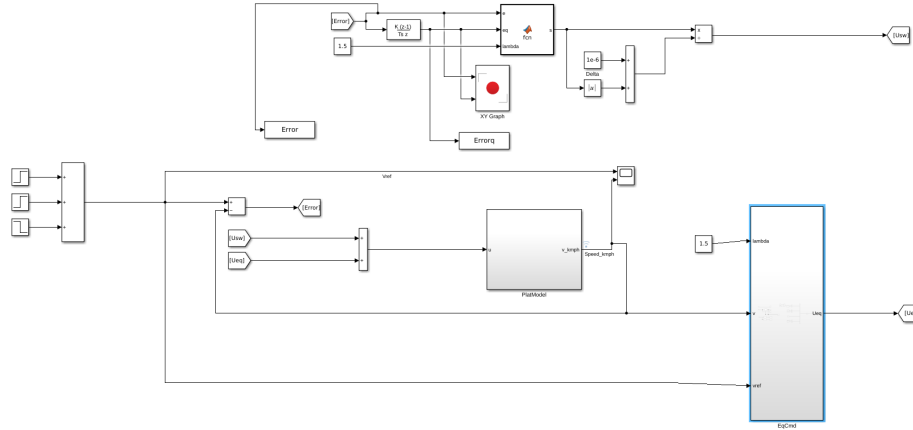


Figura 4.13: Implementarea SMC în Simulink

la poziție aproape 0.

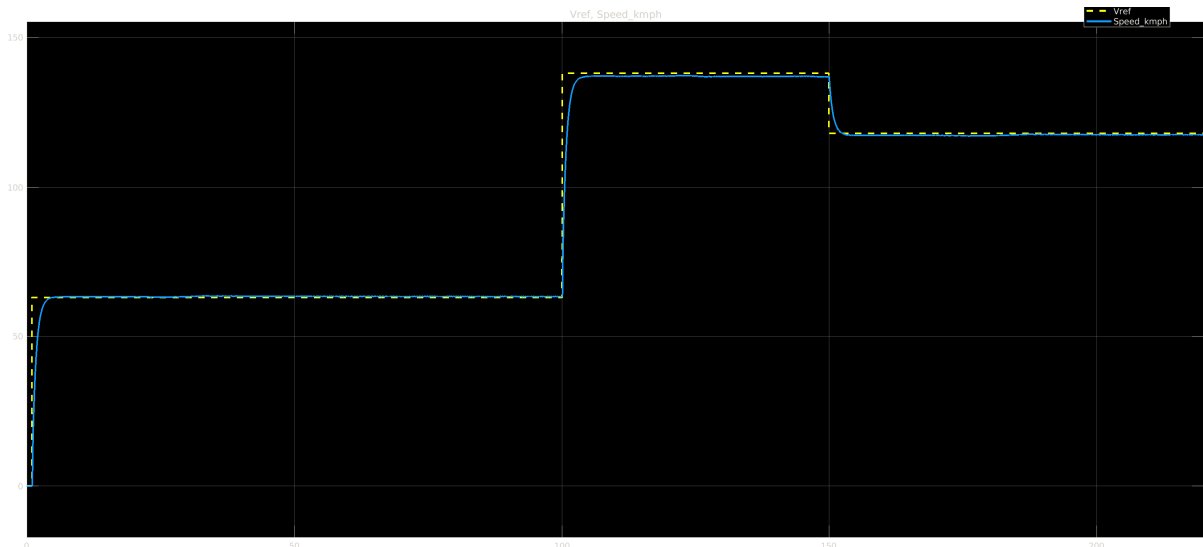


Figura 4.14: Simularea în buclă închisă a sistemului neliniar folosind SMC.

4.4.3. Implementarea Regulatorului Robust pe Baza Normei H_∞

Implementarea regulatorului robust pe baza sensibilității mixte se bazează pe conceptele teoretice expuse în 3.3.2 ce sunt integrate în toolbox-ul oferit de Matlab numit *Robust Control Toolbox*, astfel pentru obținerea rezultatelor ce vor fi expuse în această subsecțiune am utilizat script-ul din Anexa A pentru implementarea regulatorului și analiza performanțelor acestuia, iar testarea în buclă închisă a fost realizată implementând regulatorul în Simulink cu ajutorul blocului de State Space.

Pentru a impune performanțele dorite a sistemului în buclă închisă, așa cum am discutat în 3.3.3 trebuie să alegem funcțiile de ponderare a sensibilității și sensibilității complementare conform relațiilor din (3.78), astfel putem să impunem lățimea de bandă dorită, să limităm suprareglajul și să asigurăm o urmărire a traiectoriei cât mai bună prin impunerea unui offset maxim la eroarea staționară la poziție.

Astfel pentru sistemul liniarizat având funcția de transfer:

$$G(s) = \frac{292.4}{s^2 + 14.12s + 1.259},$$

am ales următoarele filtre de ponderare:

$$wS(s) = We(s) = \frac{0.6667s + 1}{s + 0.0001}, \quad (4.15)$$

$$wT(s) = Wu(s) = \frac{s + 10}{0.0001s + 15}, \quad (4.16)$$

ce vor asigura o banda de trecere de 1rad/s și un maxim de vârf al sensivității de 1,5 ceea ce va limita în final suprareglajul. Odată alese aceste două funcții de ponderare pentru a realiza *loopshaping-ul* funcțiilor de sensivitate putem folosi scriptul din Anexa A și obținem următorul regulator optimal cu următoarea reprezentare în spațiul stărilor:

$$A_K = \begin{bmatrix} -9.215 & -0.9149 & -10.17 \\ -359.6 & -74.51 & -395.5 \\ 1.231 & 4.225 & 1.229 \end{bmatrix},$$

$$B_K = \begin{bmatrix} 0.9697 \\ -0.02568 \\ -0.2429 \end{bmatrix},$$

$$C_K = [22.46 \quad 3.775 \quad 24.71],$$

$$D_K = [0],$$

iar $\gamma = 0.7894$ o valoare ce este mai mică decât 1 ceea ce indică că sistemul în buclă închisă îndeplinește cerințele impuse prin setarea funcțiilor de sensivitate de mai sus.

Analiza Performanței Regulatorului

Pentru analiza performanțelor am trasat graficul valorile singulare ale funcțiilor de sensivitate, astfel în figura 4.15 sunt prezentate valorile singulare ale lui S și T , precum și limitele de performanță și robustețe wS și $1/wT$. Se observă că valoarea singulară minimă a lui S^{-1} este sub răspunsul în magnitudine al lui wS și valoarea singulară maximă a lui T este sub răspunsul în magnitudine al lui $1/wT$. Acest lucru înseamnă că cerințele de performanță și robustețe specificate de funcțiile de ponderare wS și wU sunt îndeplinite. De asemenea în figura 4.16 sunt trasate separat valorile singulare ale sistemului în buclă deschisă, L , în raport cu limita de performanță wS și limita de robustețe $1/wT$. Astfel ca o reconfirmare al îndeplinirii cerințelor de performanță și robustețe impuse, din grafic rezultă că cea mai mică valoare singulară a lui L se află deasupra limitei lui wS în intervalul de frecvențe joase, iar cea mai mare valoare singulară a lui L este sub limita $1/wT$ în intervalul de frecvențe înalte.

Simularea Regulatorului

Simularea regulatorului în buclă închisă pe sistemul neliniar după cum am precizat a fost realizată în simulink, iar regulatorul a fost implementat printr-un block de state

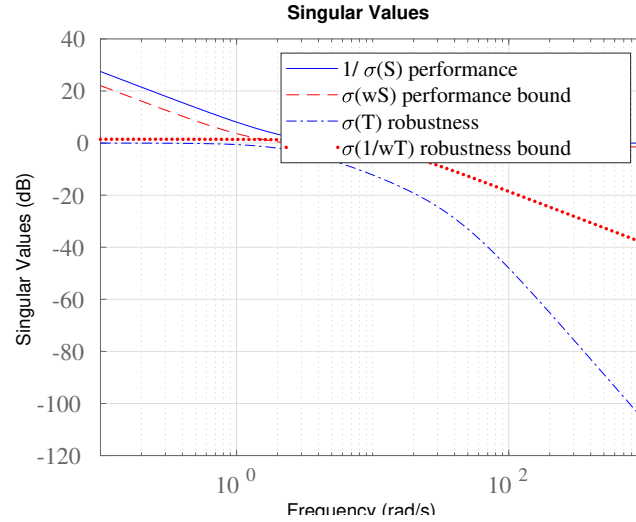


Figura 4.15: Performanțele obținute cu regulatorul bazat pe *Mixed Sensivity*.

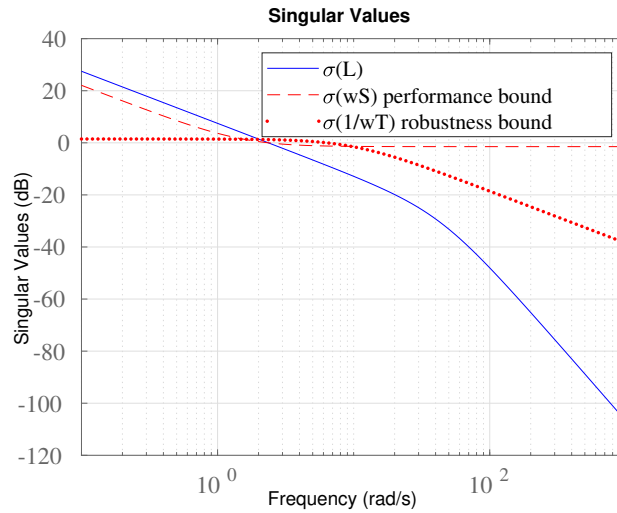


Figura 4.16: Valorile singulare ale sistemului în buclă deschisă.

space. Aplicând același set de trepte de referiță ca și în simularea regulatorului SMC am obținut următorul rezultat din Figura 4.17 în care se poate observa că sistemul urmărește traiectoria având eroare staționară la poziție nulă, există totuși un suprareglaj de aproximativ 2% ce este neglijabil, dar ulterior acesta nu mai apare, iar timpul de răspuns este unul rapid de aproximativ 10s.

4.4.4. Implementarea Regulatorului Neuronal NARMA-L2

Implementarea regulatorului s-a făcut folosind toolbox-ul *Deep Learning Toolbox* folosind blocul *NARMA-L2 Controller* ce implementează conceptele prezentate în 3.4.2. Primul pas în folosirea acestui tip de regulator este să identificăm sistemul folosind un model NARMA.

Apoi odată obținut setul de date prin configurarea rețelei neuronale, acest lucru se face prin setarea parametrilor blocului NARMA-L2 așa cum se vede în Figura 4.18 se poate face antrenarea regulatorului folosind datele obținute, iar rezultatele în urma

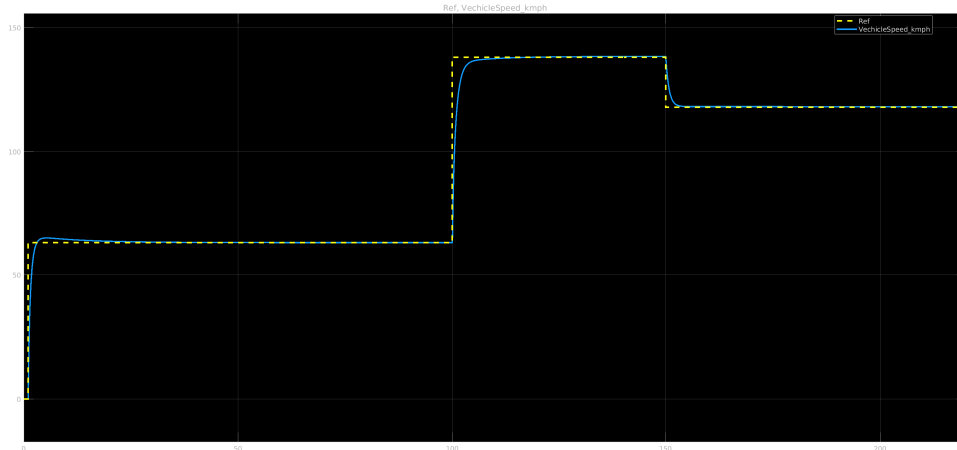


Figura 4.17: Simularea în buclă închisă a sistemului neliniar folosind regulatorul robust.

Figura 4.18: Configurarea Rețelei Neuronale.

antrenării sunt prezentate în Figurile 4.19 4.20 4.21 unde parametrii și performanțele obținute sunt: pentru antrenare s-a folosit algoritmul Levenberg-Marquardt, numărul de epoci folosit este 4000, dar antrenarea s-a oprit la iterația 243 acolo fiind atins minimul erorii medii pătratice de 0.736 cu un gradient de 2.39, date ce sunt validate și de Figura 4.22, în care pe validare cea mai mică eroare medie pătratică s-a obținut la iterația 237.

Simulare Regulatorului

Având configurațiile făcute anterior pentru simulare s-a folosit structura implementată în simulink din Figura 4.23 în urma cărei aplicând același set de trepte ca în cazurile precedente s-a obținut rezultatul din Figura 4.24 unde se poate observa că sistemul urmărește bine traiectoria, eroarea staționară la poziție este în banda acceptată de $\pm 3\%$ nu există suprareglaj, iar timpul de răspuns de aproximativ 6s.

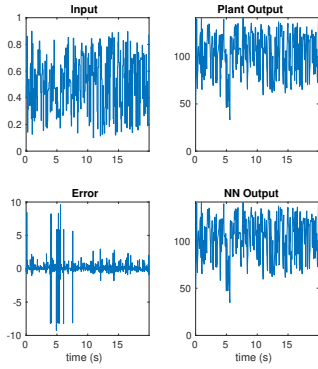


Figura 4.19: Antrenare

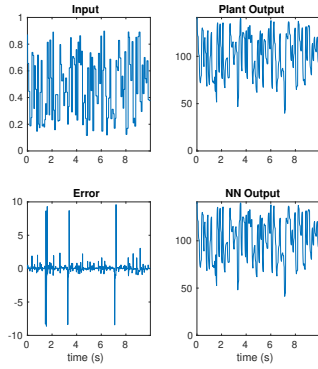


Figura 4.20: Validare

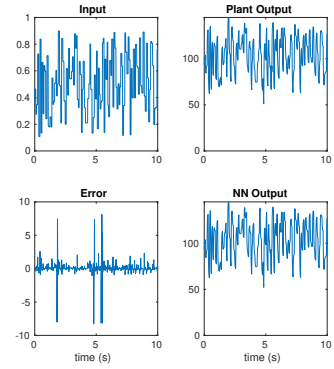


Figura 4.21: Antrenare

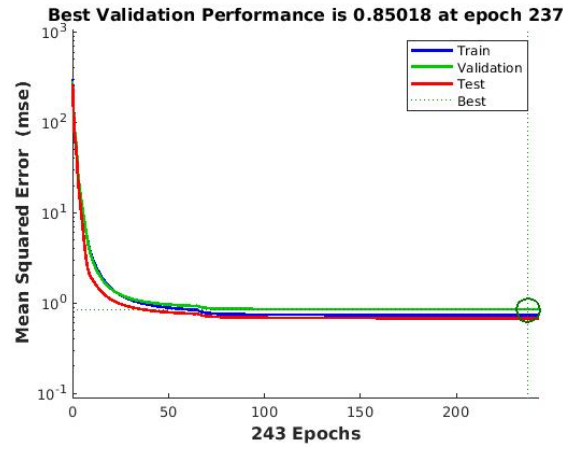


Figura 4.22: Performanțe obținute în urma validării.

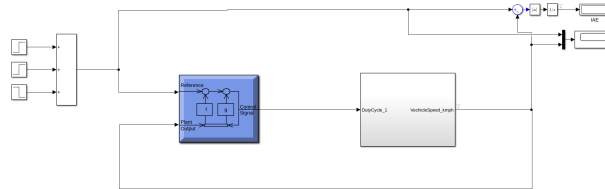


Figura 4.23: Implementarea pentru simulare a regulatorului.

4.4.5. Compararea Rezultatelor Obținute

Pe baza simulărilor obținute prezentate în Figura 4.25 s-a dedus tabelul 4.3 pe baza căruia putem trage următoarele concluzii:

Regulatorul H_∞ : prezintă un supragalaj de 2.85%, care este acceptabil, având totodată un timp de răspuns T_r de 10 secunde și o eroare staționară la poziție nulă. Deși acest regulator asigură o performanță bună în ceea ce privește eroarea staționară, timpul său de răspuns este cel mai mare dintre toate;

Regulatorul pe baza *sliding mode*: nu prezintă supragalaj, având un timp de răspuns mult mai mic, de 4.3 secunde, și o eroare staționară de 0.1. Aceasta sugerează că SMC oferă un compromis bun între timpul de răspuns și precizia în eliminarea erorii staționare;

Regulatorul NARMA-L2: similar cu SMC, nu prezintă supragalaj, dar are un timp de răspuns de 5.57 secunde, puțin mai mare decât SMC, și o eroare staționară mai mică, de

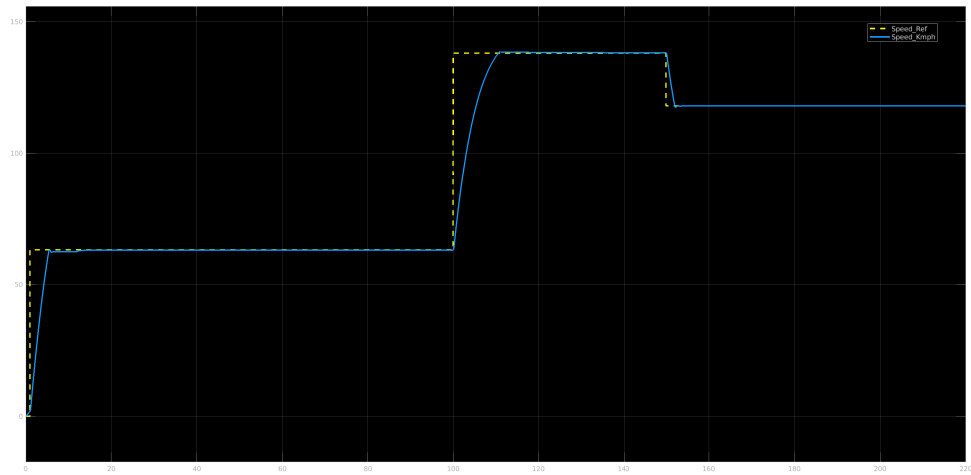


Figura 4.24: Simularea în buclă închisă a sistemului neliniar folosind regulatorul neuronal NARMA-L2.

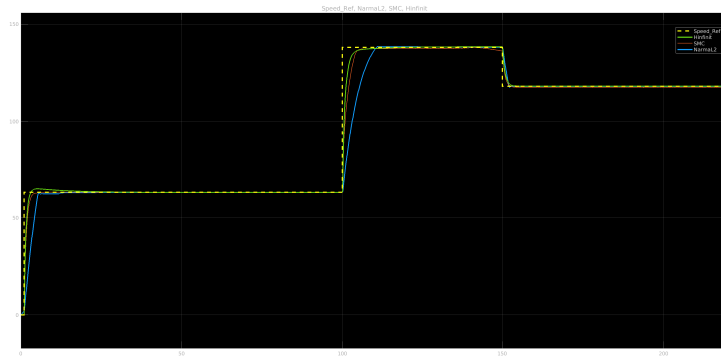


Figura 4.25: Simularea în buclă închisă a sistemului neliniar cu cele trei regulatoare.

0.02. Acest lucru indică faptul că NARMA-L2 oferă o precizie mai mare în comparație cu SMC, dar cu un compromis ușor în ceea ce privește timpul de răspuns.

	Suprareglaj	Tr	Eroarea Staționară
H_{∞}	2.85%	10	0
SMC	0	4.3	0.1
NARMA-L2	0	5.57	0.02

Tabela 4.3: Comparearea rezultatelor obținute în urma simulărilor.

Capitolul 5. Concluzii

În concluzie, pe baza cercetării efectuate și a analizei sistemelor de propulsie hibride și ceea ce implică controlul avansat al acestora am reușit să ating toate obiectivele propuse la începutul lucrării. De asemenea teza conține o documentare amplă a tuturor metodelor utilizate atât pentru modelarea sistemului fizic, dar și pentru controlul acestuia. Mai mult de atât am reușit să detaliez modul de implementare al întregului sistem în mediul MATLAB/Simulink cât și implementarea reguletoarelor propuse ce ar putea fi folosite și pentru alte tipuri de sisteme întrucât am tratat problema proiectării cât și a implementării dintr-un punct de vedere generalist.

Rezultate obținute

Așa cum am menționat și mai sus conceptele de bază de la care s-a plecat în partea de analiză au fost îndeplinite obținând în final un model complex complet funcțional ce simulează o dinamică aproape de realitate întrucât scopul final al lucrării a fost controlul unui astfel de sistem nu neapărat implementarea fidelă a tuturor caracteristicilor fizice și mecanice ce intervin în proiectarea unui astfel de sistem. Prin urmare odată ce am obținut un model funcțional am reușit să proiectez și să implementez pentru acesta trei tipuri de reguletoare bazate pe tehnici avansate de control, **Regulator bazat pe optimizarea normei H_∞** , **Regulator folosind tehnica de *sliding mode*** și un **Regulator Neuronal bazat pe modelul NARMA-L2**. Toate aceste reguletoare au adus îmbunătățiri semnificate din punct de vedere al performanței sistemului, timp de răspuns, suprareglaj, eroarea staționară la poziție ce este aproape nulă în fiecare caz deși sistemul este unul neliniar și proiectarea reguletoarelor a fost făcută în urma liniarizării acestuia. Astfel partea de "Modelare și control avansat" din titlul lucrării a fost atinsă cu succes, obținând rezultate satisfăcătoare.

Contribuții personale

Pe parcursul tezei, am efectuat o serie de contribuții personale pentru a reuși să duc la bun sfârșit soluționarea obiectivelor propuse. O primă contribuție personală după cum am menționat și la început a fost analiza și documentarea amănunțită în ceea ce privește sistemul propus pentru o mai bună înțelegere a principiilor de funcționare ca să pot să întocmesc modelul apoi să-l controlez și să obțin în final o optimizare a performanțelor acestuia în diferite condiții de funcționare.

Pe de altă parte pentru a putea efectua controlul unui astfel de sistem am fost nevoit să înțeleg aparatul matematic riguros ce stă în spatele proiectării unor astfel de reguletoare ca să pot să le adaptez la modelul implementat, iar în final am reușit să controlez un astfel de sistem complex.

Toate aceste aspecte menționate mai sus au fost implementate și integrate în mediul de dezvoltare MATLAB/Simulink și am realizat o testare și analiză a sistemului controlat pentru a putea evalua performanțele fiecărui regulator în parte și de evindetia eventualele îmbunătățiri sau dacă este fezabil sau nu folosirea unor astfel de reguletoare.

Direcții de dezvoltare

Din punct de vedere al direcțiilor de dezvoltare propun următoarele aspecte ce ar putea extinde proiectul actual. Una dintre ele ar fi întocmirea unui model mai fidel față de realitate, mai exact implementarea complexă a motorului pe combustie internă ținând

cont de toți ciclii de funcționare a acestuia și luând în considerare și partea de termodinamică, astfel s-ar putea realiza un control avansat în vederea optimizării consumului de combustibil prin alternarea eficientă între motorul electric și cel mecanic.

Un alt aspect ce l-am menționat și pe parcursul tezei, mai exact având în vedere integrarea celor două motoare ce au funcționalități diferite și funcționează fie în același timp, fie alternativ, sistemul final poate fi privit ca un *switched system* (sisteme care se schimbă între ele după o anumită lege de timp, sau pe baza unor evenimente discrete, care de cele mai multe ori nu sunt controlabile și nu putem intervenii, în cazul nostru nivelul bateriei influențează această interschimbare), se dovedește a fi un task greu pentru că deși cele două sisteme pot fi asimptotic stabile în condiții normale de funcționare, în momentul în care are loc schimbarea, acest aspect nu poate garanta o stabilitate asimptotică globală a întregului sistem, acesta din urmă poate intra în instabilitate. Astfel deși metodele clasice de control de cele mai multe ori reușesc să controleze un astfel de sistem prin anumite artificii, i.e introducerea de latențe în momentul când are loc schimbul, în ceea ce privește control avansat nu există un aparat matematic sau mecanic care să poată garanta stabilitatea asimptotică a unui astfel de sistem. Așadar o altă direcție de cercetare ar fi studiul dacă există sau nu o metodă prin care putem asigura stabilitatea asimptotică globală.

Bibliografie

- [1] P. Airikka, “Advanced control methods for industrial process control,” *Computing and Control Engineering*, vol. 15, no. 3, pp. 18–23, 2004.
- [2] Scout Systems, “Advanced process control systems,” 2024, accessed: 2024-09-01. [Online]. Available: <https://www.scout.systems/advanced-process-control-systems/>
- [3] S. Skogestad, “Plantwide control: The search for the self-optimizing control structure,” *Journal of process control*, vol. 10, no. 5, pp. 487–507, 2000.
- [4] R. Bishop and R. Dorf, *Modern control systems*, 2011.
- [5] H. Winner, S. Hakuli, F. Lotz, and C. Singer, *Handbook of driver assistance systems*. Springer International Publishing Amsterdam, The Netherlands:, 2014.
- [6] S.-L. Jämsä-Jounela, “Current status and future trends in the automation of mineral and metal processing,” *Control Engineering Practice*, vol. 9, no. 9, pp. 1021–1035, 2001.
- [7] J. Rifkin, “The end of work: The decline of the global labor force and the dawn of the post-market era,” *Journal of Leisure Research*, vol. 30, no. 1, p. 172, 1998.
- [8] M. Arnold and G. Andersson, “Model predictive control of energy storage including uncertain forecasts,” in *Power systems computation conference (PSCC), Stockholm, Sweden*, vol. 23. Citeseer, 2011, pp. 24–29.
- [9] S. Phuyal, D. Bista, and R. Bista, “Challenges, opportunities and future directions of smart manufacturing: a state of art review,” *Sustainable Futures*, vol. 2, p. 100023, 2020.
- [10] Simulate Live, “Advanced process control: A history overview,” 2024. [Online]. Available: <https://simulatelive.com/optimize/advanced-process-control/advanced-process-control-a-history-overview>
- [11] N. Jaalam, N. Rahim, A. Bakar, C. Tan, and A. M. Haidar, “A comprehensive review of synchronization methods for grid-connected converters of renewable energy source,” *Renewable and Sustainable Energy Reviews*, vol. 59, pp. 1471–1481, 2016.
- [12] H. Bevrani, T. Kato, T. Ise, and K. Inoue, *Grid connected converters: Modeling, stability and control*. Elsevier, 2022.
- [13] J. Jongudomkarn, J. Liu, and T. Ise, “Virtual synchronous generator control with reliable fault ride-through ability: A solution based on finite-set model predictive control,” *IEEE journal of emerging and selected topics in power electronics*, vol. 8, no. 4, pp. 3811–3824, 2019.

-
- [14] —, “Comparison of current-limiting strategies of virtual synchronous generator control during fault ride-through,” *IFAC-PapersOnLine*, vol. 51, no. 28, pp. 256–261, 2018.
 - [15] J. Liu, Y. Miura, H. Bevrani, and T. Ise, “Enhanced virtual synchronous generator control for parallel inverters in microgrids,” *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2268–2277, 2016.
 - [16] S. D. Al-Majidi, M. Kh. AL-Nussairi, A. J. Mohammed, A. M. Dakhil, M. F. Abbod, and H. S. Al-Raweshidy, “Design of a load frequency controller based on an optimal neural network,” *Energies*, vol. 15, no. 17, p. 6223, 2022.
 - [17] H. Shayeghi and H. Shayanfar, “Application of ann technique based on μ -synthesis to load frequency control of interconnected power system,” *International Journal of Electrical Power & Energy Systems*, vol. 28, no. 7, pp. 503–511, 2006.
 - [18] Y. Oysal, “A comparative study of adaptive load frequency controller designs in a power system with dynamic neural network models,” *Energy conversion and management*, vol. 46, no. 15-16, pp. 2656–2668, 2005.
 - [19] D. Qian, S. Tong, H. Liu, and X. Liu, “Load frequency control by neural-network-based integral sliding mode for nonlinear power systems with wind turbines,” *Neurocomputing*, vol. 173, pp. 875–885, 2016.
 - [20] Y.-C. Chang, N. Roohi, and S. Gao, “Neural lyapunov control,” *Advances in neural information processing systems*, vol. 32, 2019.
 - [21] S. Gao, J. Avigad, and E. M. Clarke, “ δ -complete decision procedures for satisfiability over the reals,” in *International Joint Conference on Automated Reasoning*. Springer, 2012, pp. 286–300.
 - [22] V. Mihaly, “Fixed structure robust synthesis for nonlinear systems,” Ph.D. dissertation, Technical University of Cluj-Napoca, 2024.
 - [23] B. De Jager, T. Van Keulen, and J. Kessels, *Optimal control of hybrid vehicles*. Springer, 2013.
 - [24] L. Miková, I. Virgala, and M. Kelemen, “Speed control of dc motor,” *American Journal of Mechanical Engineering*, vol. 4, no. 7, pp. 380–384, 2016.
 - [25] W. Liu, *Introduction to hybrid vehicle system modeling and control*. John Wiley & Sons, 2013.
 - [26] M. T. Do, “Sliding mode learning control and its applications,” *Ph. D. Thesis*, 2014.
 - [27] J. Y. Hung, W. Gao, and J. C. Hung, “Variable structure control: A survey,” *IEEE transactions on industrial electronics*, vol. 40, no. 1, pp. 2–22, 1993.
 - [28] V. Utkin and K. Yang, “Methods for construction of discontinuity planes in multidimensional variable structure systems,” *Avtomatika i Telemekhanika*, no. 10, pp. 72–77, 1978.

- [29] I. Boiko, L. Fridman, A. Pisano, and E. Usai, “Analysis of chattering in systems with second-order sliding modes,” *IEEE transactions on Automatic control*, vol. 52, no. 11, pp. 2085–2102, 2007.
- [30] T. Miquel, “Introduction to robust control lecture notes-draft,” Ph.D. dissertation, ENAC, 2022.
- [31] N. Hunter, “Analysis of nonlinear systems using arma models,” in *Proceedings of the 8th International Modal Analysis Conference*, 1990, pp. 341–347.
- [32] K. S. Narendra and S. Mukhopadhyay, “Adaptive control using neural networks and approximate models,” *IEEE Transactions on neural networks*, vol. 8, no. 3, pp. 475–485, 1997.

Anexa A. Secțiuni relevante din cod

```
numG = [0,0,292.4236800000000];
denG = [1,14.11761111111111,1.25920326222223];
G = tf(numG, denG);
s = zpk('s');
wS = createSweight(1,1e-4,1.5,1);
wT = createTweight(10,1e-4,1.5,1);
P = augw(G, wS, [], wT);
gmin = 0.1;
gmax = 10;
tol = 0.01;
[K, CL, gamma] = hinfsyn(P, 1, 1, [gmin gmax]);
gamma
sys = ss(K.A, K.B, K.C, K.D);
sys = minreal(tf(sys));

looptransfer = loopsens(G,K);
L = looptransfer.Lo;
T = looptransfer.To;
I = eye(size(L));

figure(1)
omega = logspace(-1,3,100);
sigma(I+L,'b-',wS/gamma,'r--',T,'b-.',gamma/wT,'r.',omega)
grid
legend('1/\sigma(S) performance', ...
'\sigma(wS) performance bound', ...
'\sigma(T) robustness', ...
'\sigma(1/wT) robustness bound')
set(gca,'FontSize',9,'Fontsize',14,'FontName','Times')
figure(2)
omega = logspace(-1,3,100);
sigma(L,'b-',wS/gamma,'r--',gamma/wT,'r.',omega)
grid
legend('\sigma(L)', '\sigma(wU) performance bound', ...
'\sigma(1/wT) robustness bound')
set(gca,'FontSize',9,'Fontsize',14,'FontName','Times')
```