



Controlul diferențial a două motoare BLDC cu scopul înlăturării diferențialului clasic

LUCRARE DE LICENȚĂ

Absolvent: **Petrișor-Victor MARGIN**

Coordonator științific: **As.dr.ing. Vlad Mihai MIHALY**

2024



FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
DEPARTAMENTUL AUTOMATICĂ

DECAN,
Prof. dr. ing. Mihaela DÎNȘOREANU

DIRECTOR DEPARTAMENT,
Prof. dr. ing. Honoriu VĂLEAN

Absolvent: **Petrișor-Victor MARGIN**

Controlul diferențial a două motoare BLDC cu scopul înlăturării diferențialului clasic

1. **Enunțul temei:** *Controlul diferențial a două motoare de curent continuu fără perii (eng. Brushless DC Motors), folosind modelul geometric Ackermann, cu scopul înlăturării diferențialului clasic. Implementarea și simularea controlului cât și a funcției de direcție pe un sistem fizic. Extinderea controlului prin folosirea unui model în spațiul stărilor care are ca stări viteza unghiulară și curentul de armătură a unui motor și controlarea acestuia printr-un regulator cu impunere de poli, calculat folosind inegalități matriciale liniare (eng. Linear Matrix Inequations) și prin implementarea unui estimator care reușește să asigure estimarea vitezei unghiulare și a curentului de armătură pe baza ieșirii sistemului (viteza unghiulară măsurată cu ajutorul encoderelor de viteză realizate).*
2. **Conținutul lucrării:** *Introducere, Studiu Bibliografic, Analiză și fundamentare, Concluzii, Bibliografie*
3. **Locul documentării:** Universitatea Tehnică din Cluj-Napoca, Departamentul Automatică
4. **Consultanți:**
5. **Data emiterii temei:** 1 Noiembrie 2023
6. **Data predării:** 12 iulie 2024

Absolvent: _____

Coordonator științific: _____

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**
DEPARTAMENTUL AUTOMATICĂ**Declarație pe propria răspundere privind
autenticitatea lucrării de licență**

Subsemnatul(a) **Petrișor-Victor MARGIN**, legitimat(ă) cu **CI** seria **SX** nr. **565145** CNP **5010808314007**, autorul lucrării **Controlul diferențial a două motoare BLDC cu scopul înlăturării diferențialului clasic**, elaborată în vederea susținerii examenului de finalizare a studiilor de licență la Facultatea de Automatică și Calculatoare, Specializarea **Automatică și Informatică Aplicată** din cadrul Universității Tehnice din Cluj-Napoca, sesiunea **Iulie 2024** a anului universitar **2023-2024**, declar pe propria răspundere că această lucrare este rezultatul propriei activități intelectuale, pe baza cercetărilor mele și pe baza informațiilor obținute din surse care au fost citate, în textul lucrării și în bibliografie.

Declar că această lucrare nu conține porțiuni plagiate, iar sursele bibliografice au fost folosite cu respectarea legislației române și a convențiilor internaționale privind drepturile de autor.

Declar, de asemenea, că această lucrare nu a mai fost prezentată în fața unei alte comisii de examen de licență.

În cazul constatării ulterioare a unor declarații false, voi suporta sancțiunile administrative, respectiv, *anularea examenului de licență*.

Data

11 iulie 2024

Nume, Prenume

Margin Petrișor-Victor

Semnătura

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**
DEPARTAMENTUL AUTOMATICĂ**SINTEZA**

proiectului de diplomă cu titlul:

**Controlul diferențial a două motoare BLDC cu scopul înlăturării
diferențialului clasic**

Autor: **Petrișor-Victor MARGIN**

Conducător științific: **As.dr.ing. Vlad Mihai MIHALY**

1. **Cerințele temei:** Realizarea unui sistem fizic format din două motoare de curent continuu fără perii, având vitezele controlate. Studiarea metodelor de control și soluționarea neliniarităților. Cercetarea soluțiilor de implementare a funcționalităților de control și de calcul a vitezelor individuale ale motoarelor și rezolvarea problemelor introduse de interfațarea sistemului continuu cu cel discret.
2. **Soluții alese:** Realizarea controlului prin calcularea unui regulator cu impunere de poli, în două cazuri, unul care respectă o saturație impusă și respectiv unul care nu respectă această condiție. De asemenea, realizarea unui regulator cu impunere de poli calculat prin respectarea inecuațiilor liniare matriciale. Pentru funcționalitatea de calcul a vitezelor motoarelor, se folosește modelul geometric Ackermann, care descrie viteza roților în parcurgerea unui drum curbiliniu.
3. **Rezultate obținute:** Se obține diferența între folosirea regulatorului care respectă saturația și cel care nu o respectă. De asemenea, se prezintă o comparație cu sistemul necontrolat și se simulează un control avansat ce conține un estimator și regulatorul calculat cu ajutorul inegalităților.
4. **Testări și validări:** Setul de testări constă din testarea și validarea neliniarităților identificate, a reguletoarelor discrete calculate, cât și a sistemului cu control avansat.
5. **Contribuții personale:** Realizarea machetei de testare, implementarea funcționalităților pe microcontroller, proiectarea reguletoarelor pe baza unei funcții de transfer și respectiv pe baza unui model matematic în spațiul stărilor, proiectarea unui control avansat pe baza unui estimator, testarea și validarea fizică și software a proiectului, soluționarea problemelor de comunicare și de interpretare a mesajelor între elementele folosite.
6. **Surse de documentare:** Principalele resurse folosite în vederea realizării proiectului au fost: U. Celtekligil and M. Cavdar, "Modeling, simulation and analysis of bipolar



driven bldc motor,” 09 2021, pp. 334–339, C. Xiang, X. Wang, Y. Ma, and B. Xu, “Practical modeling and comprehensive system identification of a bldc motor,” Mathematical Problems in Engineering, vol. 2015, no. 1, p. 879581, 2015, P. Dobra, “Robust pid controller of bldc motor via state feedback design,” F. Celani and A. Isidori, “Feedback linearization,” Scholarpedia, vol. 4, p. 6517, 01 2009

Semnătura autorului:

Semnătura conducătorului științific:

Cuprins

Capitolul 1	Introducere	1
1.1	Contextul Temei	1
1.2	Lista obiectivelor propuse	1
1.3	Specificații	2
Capitolul 2	Studiu bibliografic	4
2.1	Modelarea și Identificarea Sistemelor	4
2.2	Corectarea Neliniarităților	5
2.3	Plasarea Polilor	6
2.4	Estimarea Stărilor	9
2.5	LQR și LMI (Linear Matrix Inequations)	11
Capitolul 3	Analiză și fundamentare teoretică	13
3.1	Interpretarea și analizarea problemei	13
3.1.1	Arhitectura proiectului	13
3.1.2	Componenta Fizică (Hardware)	14
3.1.3	Componenta Software	17
3.2	Proiectare și implementare	22
3.2.1	Proiectarea elementelor fizice	23
3.2.2	Proiectarea elementelor software și de simulare	24
3.3	Testare și Validare	34
3.3.1	Simularea sistemului identificat	34
3.3.2	Simularea efectului de liniarizare	36
3.3.3	Simularea reguletoarelor obținute	37
3.3.4	Testarea fizică a reguletoarelor	39
3.3.5	Testarea sistemului întreg	40
3.3.6	Simularea controlului avansat	41
3.4	Instalare și utilizare	43
Capitolul 4	Concluzii	45
4.1	Rezultate obținute	45
4.2	Direcții de dezvoltare	46
Bibliografie		47

Capitolul 1. Introducere

1.1. Contextul Temei

Tema acestui proiect se dezvoltă în jurul rezolvării unei probleme fizice cu ajutorul cunoștințelor elementelor de electronică, a cunoștințelor legate de identificarea, modelarea și controlul sistemelor, respectiv cunoștințe legate de metodele de programare a microcontrollerelor (μC). În sine, proiectul are ca scop controlul diferențial al motoarelor de curent continuu fără perii (eng. *Brushless DC Motor* – BLDC) cu scopul renunțării la diferențialul mecanic clasic. Una din principalele probleme ale mașinilor electrice este cea dată de greutatea mare a lor. Aceasta provine în principal din cauza acumulatorilor mari folosiți pentru încapsularea energiei electrice, care constituie mai mult de un sfert din greutatea mașinii, având o masă aproximativă de $700[kg]$ [1]. Alte componente care aduc greutate sunt motoarele și caroseria efectivă, însă acestea sunt destul de bine optimizate din punct de vedere al greutății. Astfel, pentru a reduce masa efectivă a autoturismului, trebuie ca elementele mecanice să devină din ce în ce mai ușoare fie, trebuie înlăturate, în cazul în care se poate găsi o soluție alternativă.

De asemenea, creșterea pieței de autovehicule electrice determină în cadrul producătorilor o cursă pentru obținerea performanțelor cele mai bune, cu un cost cât mai mic. Evoluția pieței de mașini electrice prezintă faptul că în anii 2010-2015 au fost realizate în jur de 4 milioane de mașini hibride/electrice, iar în anul 2023 au fost realizate 42 de milioane de autovehicule hibride/electrice [2]. Așadar, putem trage concluzia că piața mașinilor electrice a explodat în ultimii 10 ani, și reprezintă o ramură importantă pentru evoluția transportului modern, atât de persoane cât și de mărfuri.

Totodată, prin reducerea elementelor mecanice redundante, se reușește combaterea poluării de producție, eficientizarea utilizării de resurse naturale și respectiv asigurarea unui consum redus, fapt ce dermină o nevoie mai scăzută de energie obținută prin procese de ardere. Subiectul de ecologizare al proceselor din lume este unul dintre cele mai urmarite și importante probleme, fapt ce reiese din bugetul alocat de către Comisia Europeană (611 milioane euro) pentru soluționarea acestora [3].

Astfel, producătorii de mașini electrice sunt într-o continuă confruntare pentru obținerea performanțelor mai bune, cu un cost cât mai redus și care să se încadreze în noile dorinte de ecologizare. Această dorință de creștere a performanțelor aduce în prim plan tema acestei lucrări.

1.2. Lista obiectivelor propuse

Pentru realizarea atât fizică cât și în cadrul domeniului software, se urmăresc obiectivele definite în lista de mai jos:

- Procurarea tuturor elementelor care constituie sistemul dorit (senzori, motoare, etc) și asigurarea bunei funcționări ale acestora.
- Măsurarea și interpretarea datelor, precum viteza unghiulară a motoarelor, cu ajutorul senzorilor, cât și transmiterea informațiilor obținute prin cadrul comunicației seriale între plăcuța de dezvoltare și calculator.
- Studiarea metodelor de control în cazul respectării/nerespectării unei saturații im-

puse și a metodelor de înlocuire a diferențialului clasic prin calcularea vitezelor de referință a motoarelor pe baza geometriei unei mașini.

- Obținerea unui model matematic sub formă de funcție de transfer cu intrarea factor de umplere și ieșirea viteză unghiulară, respectiv a unui model extins în spațiul stărilor, care are ca intrare tensiunea de alimentare a motoarelor și sarcina atribuită sistemului, ca ieșire viteză unghiulară a motoarelor, iar ca stări viteză de rotație și curentul de armătură a motoarelor folosite.
- Realizarea unei machete care să conțină elementele folosite, care asigură funcționarea sigură a sistemului și totodată care permite observarea fiecărui element în parte și a conexiunilor dintre ele.
- Testarea mai multor metode de control, precum plasarea de poli folosind metoda regulatorului quadratic liniar (Linear Quadratic Regulator – LQR), cât și prin impunerea unor inegalități matriciale (Linear Matrix Inequation – LMI) cu scopul optimizării plasării polilor, respectând condițiile impuse, stările sistemului folosite pentru control fiind obținute prin estimarea acestora.
- Implementarea unui cod ce folosește resursele unui μC pentru realizarea funcționalităților necesare (citire senzori, generare de semnale, corecție semnale, etc.) și care reușește acționarea motoarelor diferențial în funcție de viteză impusă printr-un potențiomtru și respectiv în funcție de unghiul deplasării și direcția dată de encoderul de cuadratură.
- Asigurarea comunicării între elementul de comandă reprezentat de plăcuță de dezvoltare Arduino și invertoarele folosite cu scopul acționării motoarelor prin respectarea standardului acceptat de acest tip de drive.
- Soluționarea prin metode software a zgomotelor de citire a potențiometrului și a problemelor date de folosirea unor regiștrii cu lungime de zece biți a convertorilor analogici digitali pentru generarea semnalelor de tip PWM.
- Testarea și validarea metodelor de control și a funcționalității a diferențialului software, atât a sistemului fizic realizat, cât și a sistemului extins prin adăugarea regulatorului ce plasează polii corespunzător condițiilor impuse și prin implementarea estimatorului de stare.

Lucrarea aceasta se consideră finalizată atunci când toate obiectivele au fost atinse, astfel lucrarea va explica în detaliu fiecare din punctele menționate anterior.

1.3. Specificații

Această lucrare are o parte care este dezvoltată pe calculator în domeniul software și o parte care reprezintă realizarea acesteia cu ajutorul unor componente electrice. Astfel, în ceea ce privește procesul de dezvoltare, se respectă activitatea de lucru în stil model-în-bucă (eng. Model In The Loop – MIL), unde se testează modelul obținut, cu scopul de a vedea performanțele acestuia, software în buclă (eng. Software In The Loop – SIL), unde se simulează capacitățile reglatoarelor împreună cu modelul obținut și respectiv, hardware în buclă (eng. Hardware In The Loop – HIL) unde se testează comportamentul ansamblului obținut prin punerea în funcțiune a tuturor părților.

Așadar, pentru partea de simulare și modelare (majoritară software) s-au folosit următoarele elemente prezentate în lista de mai jos:

- Un calculator care reușește să asigure buna funcționare a programelor.
- Mediul de dezvoltare Matlab/Simulink, împreună cu toolbox-urile: Control System Toolbox, System Identification Toolbox, Robust Control Toolbox și Signal Processing Toolbox

- Mediul de dezvoltare Arduino, prin care se poate scrie pe placa de dezvoltare programul C/C++ ce conține funcționalitățile dorite.
- PuTTY, un software care asigură înregistrarea datelor trimise de către placa Arduino într-un fișier text.

Pentru partea de implementare fizică a sistemului (hardware), s-au folosit componentele prezentate în lista de mai jos:

- O placă de dezvoltare Arduino Uno R3, care deține un μC de tip Atmega328.
- Doi senzori LM393 (de mare viteză), folosiți pentru realizarea encoderelor de viteză cu scopul măsurării vitezelor celor două motoare.
- Doi senzori optici, care vor forma logica de detecție a direcției.
- Doi invertori 30A (curent maxim 30 amperi), care primesc un semnal de tip PWM, iar pe baza factorului de umplere al acestuia, motoarele sunt comandate să meargă mai tare sau mai încet.
- Două motoare BLDC A2122 2450Kv, reprezentând elementele principale ale sistemului care trebuie controlat diferențial.
- Două discuri printate 3D cu 8 fante, folosite pe post de disc de encoder, fiind așezate pe axul rotor al motoarelor.
- Un potențiomtru de 220 [k Ω] pentru setarea vitezei dorite de utilizator.
- Un disc de carton, care reprezintă discul de encoder de cuadratura folosit pentru calculul unghiului și a direcției.
- Un acumulator ABSIMA 5000 mAh 3s, este sursa de tensiune folosită pentru sistemul prezentat și care determină puterea sistemului.
- Elemente de fixare precum placă de PVC, șuruburi și piulițe.

Sistemul funcționează cu tensiuni de comandă cuprinse între 0-5 V, tensiuni de putere de până la 11.1 V pentru acționarea motoarelor, curenți de comandă cu valori mici și respectiv în cazul acționării motoarelor, aceștia pot atinge și 30 A pentru o perioadă scurtă de timp. În cazul folosirii altor elemente, se recomandă verificarea tensiunilor și curenților care pot apărea în cadrul circuitului.

Capitolul 2. Studiu bibliografic

2.1. Modelarea și Identificarea Sistemelor

Pentru a începe procesul de control al unui sistem trebuie să determinăm dinamica acestuia, fie pornind de la ecuațiile diferențiale date de legile fizicii, fie experimental, prin comandarea sistemului și analizarea comportamentului acestuia în funcție de evoluția semnalului folosit la intrarea sistemului în cazul în care acesta este realizat fizic.

Astfel pornind de la ecuațiile diferențiale detaliate în lucrările [4], [5] obținem următoarele ecuații care descriu dinamica unui motor de tip BLDC:

$$\frac{d\omega}{dt} = -\frac{K_F}{J} \cdot \omega + \frac{K_T}{J} \cdot I_a - \frac{1}{J} \cdot T_L, \quad (2.1)$$

$$\frac{dI_a}{dt} = -\frac{K_E}{L_a} \cdot \omega - \frac{R_a}{L_a} \cdot I_a + \frac{1}{L_a} \cdot u_a. \quad (2.2)$$

Se notează cu ω viteza unghiulară măsurată în [rad/s], I_a curentul din armătură motorului măsurată în [A], T_L reprezintă tracțiunea la care este supus motorul, K_F coeficientul de frecare, K_T constanta tracțiunii, K_E constanta tensiunii electromotoare inverse, R_a rezistența electrică a armăturii măsurată în [Ω], L_a inductanța armăturii măsurată în [H], J momentul de inerție al motorului măsurat în [$kg \cdot m^2$] și respectiv, u tensiunea la bornele motorului măsurată în [V]. De observat este faptul că, în lucrarea [4], se discută și despre problemele care apar din cauza modificării câmpului magnetic prin comutație, însă am considerat ecuațiile fără acest aspect.

Mai mult de atât, în lucrarea [6], se prezintă o problemă care impactează sistemul destul de tare. Această problemă este introdusă de faptul că pentru a transmite o tensiune către motorul de tip BLDC, se folosește un inverter, care primește ca intrare un semnal PWM, iar pe baza acestuia el generează trei faze astfel încât motorul să se învârtă. Așadar, aici intervine următoarea ecuație:

$$u_a = K_{PWM} \cdot u_c, \quad (2.3)$$

unde se notează u_a tensiunea de la capetele armăturii, u_c reprezintă tensiunea semnalului de comandă, iar K_{PWM} este un factor de proporționalitate dat de natura neliniară a inverterului. Acest factor (gain), se poate aproxima cu un factor liniar de proporționalitate, în cazul în care se lucrează cu frecvențe mici, însă pentru frecvențe mai mari, această aproximare nu reușește să descrie comportamentul fizic.

Astfel, folosind ecuațiile de mai sus putem să determinăm spațiul stărilor asociat sistemului sub forma prezentată mai jos:

$$\frac{d}{dt} \begin{pmatrix} \omega \\ I_a \end{pmatrix} = \begin{pmatrix} -\frac{K_F}{J} & \frac{K_T}{J} \\ -\frac{K_E}{L_a} & -\frac{R_a}{L_a} \end{pmatrix} \cdot \begin{pmatrix} \omega \\ I_a \end{pmatrix} + \begin{pmatrix} -\frac{1}{J} & 0 \\ 0 & \frac{1}{L_a} \end{pmatrix} \cdot \begin{pmatrix} T_L \\ u_a \end{pmatrix}. \quad (2.4)$$

O altă metodă de obținerea a unui sistem este identificarea comportamentului acestuia pe baza relației dintre semnalul de intrare și cel de ieșire, idee folosită și în lucrările [7], [8]. În acest sens, se știe că funcția de transfer de la tensiune la viteză a unui motor de tip BLDC este de ordinul I, așadar se poate considera forma următoare:

$$H_{u\omega}(s) = \frac{K}{T \cdot s + 1} = \frac{\omega(s)}{u(s)}. \quad (2.5)$$

Astfel, pentru a identifica dinamica sistemului, trebuie să determinăm coeficientii K , care reprezintă factorul de amplificare și respectiv T , constanta de timp a motorului. Pentru calculul acestora se consideră că sistemul există fizic și poate să fie comandat cu un semnal de tip PWM. Totodată, pentru obținerea semnalelor este nevoie de folosirea senzorilor adecvați, în sensul în care trebuie cunoscute anumite detalii despre motor, deoarece pentru achiziția de date, se folosesc în principal sisteme numerice, care lucrează cu anumite perioade de eșantionare.

În vederea calculării constantelor se folosesc următoarele formule:

$$K = \frac{y_{st} - y_0}{u_{st} - u_0}, \quad (2.6)$$

$$y(T) = y_{63\%} = y_0 + 0.63 \cdot (y_{st} - y_0), \quad (2.7)$$

unde y_{st} și y_0 sunt valorile măsurate în urma aplicării semnalelor de intrare u_{st} și respectiv u_0 . Pentru calculul constantei de timp T , se calculează valoarea de 63% și se determină momentul de timp (eșantionul) la care se atinge această valoare. În cadrul calcului constantei T se ține cont de perioada de eșantionare astfel:

$$T = T_e \cdot (t_{63\%} - t_0), \quad (2.8)$$

unde T_e este perioada de eșantionare folosită, $t_{63\%}$ reprezintă eșantionul cel mai apropiat de valoarea de 63% iar t_0 este momentul initial la care se aplica semnalul de comandă de tip treaptă. De asemenea, aceeași problemă apare ca în cazul modelării, aceea fiind comportamentul nelinier al invertorului, fapt ce produce efecte nedorite și care nu permit o analiză corectă a sistemului.

2.2. Corectarea Neliniarităților

Nelinariitățile într-un proces menit controlului fundamentează o problemă considerabilă. Pentru eliminarea lor există mai multe metode precum alegerea unui punct static de funcționare și liniarizarea în jurul aceluia punct folosind seria Taylor. Această metodă asigură aproximarea comportamentului unui proces într-o arie destul de restrânsă. Astfel, pentru procese care prezintă neliniarități profunde, soluția oferită de seriile Taylor nu este una viabilă, aceasta reușind aproximarea doar pe o porțiune locală.

O metodă care asigură liniarizarea pe întreg domeniul de funcționare este cea de liniarizare pe baza reacției (Feedback Linearization). În lucrările [9], [10] se prezintă sistemul sub forma ecuației prezentate mai jos:

$$\dot{x} = f(x) + g(x) \cdot u. \quad (2.9)$$

Idea acestei metode este de înlocuire a semnalului de intrare astfel [11]:

$$u = \alpha(x) + \beta(x) \cdot v. \quad (2.10)$$

Dacă se face înlocuirea în ecuația anterioară, se obține:

$$\dot{x} = f(x) + g(x) \cdot \alpha(x) + g(x) \cdot \beta(x) \cdot v. \quad (2.11)$$

Astfel, folosind această metodă, se poate impune ca neliniaritatea care acționează asupra semnalului de intrare să fie anulată prin condiția:

$$\beta(x) = g(x)^{-1}. \quad (2.12)$$

De asemenea, dacă apar neliniarități asupra stărilor în sine trebuie realizată o schimbare de variabilă, transformând stările în altele noi:

$$z = \theta(x). \quad (2.13)$$

Astfel încât sistemul nou să devină de forma:

$$\dot{z} = h(z) + v, \quad (2.14)$$

unde $h(z)$ este o funcție liniară în z iar v reprezintă noua intrare a sistemului liniarizat.

De observat este faptul că această metodă se poate folosi doar în anumite cazuri și doar pentru anumite sisteme, deoarece în cazul neliniarităților stărilor, transformarea acestora trebuie să rezolve problema aceasta dar totodată, trebuie să asigure și controlabilitatea sistemului. Mai mult de atât, funcțiile cu care se lucrează este indicat să fie continue și definite pe intervalul de lucru. În caz contrar, funcțiile se pot descompune pe intervale, și se pot folosi mai multe transformate care comută între ele pe regiuni.

Metoda Feedback Linearization este una de liniarizare exactă (*eng. exact linearization*), ceea ce înseamnă că poate liniariza dinamica sistemului pe un interval extins din punct de vedere al punctelor de funcționare. Însă necesită un efort de calcul sporit care trebuie asigurat într-un timp mai mic decât perioada de eșantionare, în sensul realizării discrete a regulatorului și nu este garantat faptul că se poate realiza o transformare astfel încât sistemul să fie controlabil.

2.3. Plasarea Polilor

Pentru sistemele care sunt în forma modelului matematic sub numele de spațiul stărilor, o metodă de calcul a regulatorului este plasarea de poli (*eng. Pole Placement*). Această metodă pornește de la valorile proprii ale matricei de stări. Este cunoscut faptul că aceste valori denumite în engleza 'eigenvalues' reprezintă poliul sistemului. Astfel pe baza acestora putem deduce anumite informații despre sistemul abordat. Așadar putem analiza regimul de funcționare al sistemului (oscilant întreținut, oscilant amortizat, etc.). De asemenea putem deduce anumite caracteristici ale regimului tranzitoriu precum timpul de răspuns, suprareglaj și coeficientul de roll-off.

În urma acestei analize, se determină care combinație de poli se dorește în scopul îmbunătățirii performanțelor, precum timpul de răspuns și suprareglajul, astfel ajungându-se la un polinom caracteristic nou, ce conține poliul impuși.

Pornind de la un sistem pus în spațiul stărilor precum este descris mai jos:

$$\dot{x} = A \cdot x + B \cdot u. \quad (2.15)$$

Se verifică în primul rând dacă sistemul este controlabil. Dacă acesta nu respectă condiția de controlabilitate, nu se poate aplica metoda plasării polilor, și este necesară aplicarea unei transformate care să asigure controlabilitatea sistemului. Pentru a verifica controlabilitatea, se calculează matricea de controlabilitate astfel:

$$Co(A, B) = (B \quad AB \quad A^2B \quad \dots \quad A^{n-1}B), \quad (2.16)$$

unde n reprezintă numărul de stări, A este matricea de stare iar B este matricea intrărilor. Se știe că un sistem este controlabil dacă rangul matricei de controlabilitate $Co(A, B)$ este egal cu numărul de stări, în caz contrar, sistemul este parțial controlabil.

Așadar, cum a fost menționat mai sus, prin impunerea polilor se determină un nou polinom caracteristic definit astfel:

$$P_0(s) = (s - p_1)(s - p_2) \dots (s - p_n). \quad (2.17)$$

Determinarea regulatorului K se face prin folosirea următoarei relații:

$$\det(s \cdot I - (A - B \cdot K)) = P_0(s). \quad (2.18)$$

În cazul sistemelor SISO (Single Input Single Output) regulatorul K are forma:

$$K = (k_1 \quad k_2 \quad k_3 \quad \dots \quad k_n). \quad (2.19)$$

Pentru sistemele de tip MIMO (Multiple Input Multiple Output) regulatorul K are forma:

$$K = \begin{pmatrix} k_{11} & k_{12} & k_{13} & \dots & k_{1n} \\ k_{21} & k_{22} & k_{23} & \dots & k_{2n} \\ k_{31} & k_{32} & k_{33} & \dots & k_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ k_{m1} & k_{m2} & k_{m3} & \dots & k_{mn} \end{pmatrix}. \quad (2.20)$$

Numărul de linii este dat de numărul de intrări al sistemului iar numărul de coloane este dat de numărul de stări în cadrul sistemului.

Astfel, formula 2.18 se poate folosi pentru sisteme care nu sunt complicate, însă pentru sisteme cu mai multe intrări și cu un număr mare de stări, trebuie folosită formula lui Ackermann:

$$K = e_n^T \cdot (Co(A, B))^{-1} \cdot P_0(A), \quad (2.21)$$

$$e_k^T = (0 \quad 0 \quad 0 \quad \dots \quad 1 \quad 0 \quad \dots), \quad (2.22)$$

$$P_0(A) = (A - p_1 \cdot I)(A - p_2 \cdot I)(A - p_3 \cdot I) \dots (A - p_n \cdot I). \quad (2.23)$$

În urma calculării regulatorului, se asigură performanțele impuse de polii aleși, însă eroarea staționară nu este garantată să fie 0. Astfel, se calculează o așa numită matrice de prefiltrare F cu următoarea formulă:

$$F = [C(A - B \cdot K)^{-1}B]^{-1}. \quad (2.24)$$

Această formulă provine din calculul erorii staționare la poziție ϵ_{stp} :

$$\epsilon_{stp} = 1 - F \cdot \lim_{s \rightarrow 0} H(s). \quad (2.25)$$

Pentru $\epsilon_{stp} = 0$ se obține:

$$H(s) = C \cdot (s \cdot I - (A - B \cdot K)^{-1}) \cdot B \quad (2.26)$$

$$F \cdot C \cdot (A - B \cdot K)^{-1} \cdot B = 1 \quad (2.27)$$

Din ecuația 2.27 se poate deduce ecuația 2.24 prin împărțirea la dreapta a produsului de matrici. În urma acestor calcule, se obține sistemul controlat și filtrat, care asigură atât performanțele regimului dinamic, cât și cele ale regimului staționar, sub forma prezentată în imaginea de mai jos:

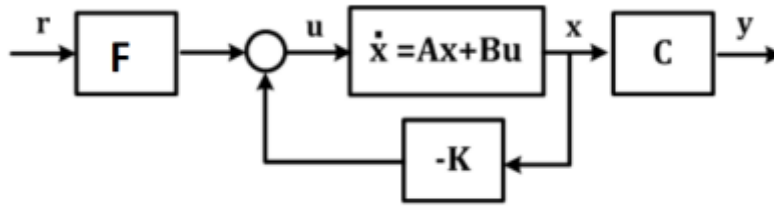


Figura 2.1: Plasare de poli și prefiltrare

Această metodă de control este una des întâlnită deoarece reușește să impună exact ce se dorește. Pentru sistemele mari, unde pentru fiecare buclă ar însemna un regulator, complexitatea și timpul de realizare al controlului crește exponențial. Așadar, sistemele mari nu pot beneficia de metode clasice de calcul al reguletoarelor PID pe baza funcției de transfer. Însă pe baza acestei metode și prin reprezentarea în spațiul stărilor a sistemului, se poate calcula eficient un regulator care să asigure toate cerințele impuse. Problema acestei metode apare atunci când nu avem acces fizic la fiecare stare. Adică, atunci când nu putem măsura toate stările nu putem aplica legea de control dată de acest regulator.

Totodată, există situația în care sistemul folosește stări care nu au o semnificație fizică măsurabilă, precum starea de sănătate a bateriei (aceasta fiind un calcul între anumite stări, dar nu poate fi direct măsurată) sau starea de sănătate a motorului care depinde de stări ca temperatura acestuia, grosimea stratului de izolator dintre firele ce alcătuiesc bobinajele acționate, etc.

2.4. Estimarea Stărilor

Nevoia de implementarea a unui estimator provine fie din punctul de vedere al imposibilității de măsurare a stărilor, fapt ce împiedică opțiunea de realizare a controlului prin impunere de poli, fie din motivul costurilor mari aduse de senzorii ce ar trebui să măsoare stările respective.

Astfel, în vederea estimării de stări, se pornește asemănător ca pentru metoda impunerii de poli, adică se pornește de la modelul matematic în spatiul stărilor al sistemului:

$$\dot{x} = A \cdot x + B \cdot u, \quad (2.28)$$

$$y = C \cdot x,$$

unde ne apare ieșirea sistemului y și matricea de ieșire C . Pentru a reuși să se dezvolte un estimator al stărilor pe baza ieșirii, trebuie ca sistemul să fie observabil. Astfel, se calculează matricea de observabilitate în funcție de matricea stărilor și matricea de ieșire folosind formula următoare:

$$Ob(A, C) = \begin{pmatrix} C \\ CA \\ CA^2 \\ \dots \\ CA^{n-1} \end{pmatrix}. \quad (2.29)$$

Condiția de observabilitate a unui sistem este ca rangul matricei $Ob(A, C)$ să fie egal cu numărul de stări prezente în sistem. În cazul în care sistemul respectă condiția de observabilitate, se poate descrie estimatorul sub forma următoare:

$$\dot{\hat{x}} = A \cdot \hat{x} + B \cdot u + L \cdot (y - \hat{y}), \quad (2.30)$$

$$\hat{y} = C \cdot \hat{x},$$

unde \hat{x} și \hat{y} reprezintă stările estimate și respectiv ieșirea estimată, iar L reprezintă estimatorul care trebuie calculat. De asemenea, se definește eroarea sub forma următoare:

$$e = x - \hat{x}. \quad (2.31)$$

Dacă se analizează dinamica erorii obținem:

$$\begin{aligned} \dot{e} = \dot{x} - \dot{\hat{x}} &= A \cdot x + B \cdot u - A \cdot \hat{x} - B \cdot u - L \cdot (y - \hat{y}) = \\ &= A \cdot (x - \hat{x}) - L \cdot (y - \hat{y}) = \\ &= A \cdot (x - \hat{x}) - LC \cdot (x - \hat{x}) = \\ &= (A - LC) \cdot e. \end{aligned} \quad (2.32)$$

Pentru ca estimatorul să fie unul care reușește să aproximeze stările sistemului, acesta trebuie ca să asigure convergența erorii spre zero. Astfel se impune ca matricea $A-LC$ să fie de tipul Hurwitz (toate valorile proprii să fie mai mici sau egale cu 0) pentru sistemele cu semnale continue, respectiv Schur (toate valorile proprii, în modul, să fie mai mici decât valoarea 1) pentru sistemele cu semnale discrete.

Se poate observa faptul că pentru calculul estimatorului, se folosește impunerea de poli, astfel încât eroarea să convergă cât mai rapid la 0. În general, poli estimatorului se impun să fie de 10 ori mai rapizi decât polii sistemului fizic. Pentru sistemele care nu necesită un calcul matricial complicat se poate impune următoarea condiție:

$$A - LC = \begin{pmatrix} p_1 & 0 & 0 & \dots & 0 \\ 0 & p_2 & 0 & \dots & 0 \\ 0 & 0 & p_3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & p_n \end{pmatrix}. \quad (2.33)$$

unde $p_1, p_1, p_1 \dots p_n$ sunt polii impuși pentru estimator. În cazul în care calculul matricial este prea complicat, se poate folosi formula lui Ackermann în felul următor:

$$L^T = Ob(A, C)^{-1} \cdot e_n^T \cdot P_0(A)^T. \quad (2.34)$$

Pentru sistemele SISO matricea L este de forma:

$$L = \begin{pmatrix} l_1 \\ l_2 \\ l_3 \\ \vdots \\ l_n \end{pmatrix}. \quad (2.35)$$

Pentru sistemele MIMO matricea L arată sub forma următoare:

$$L = \begin{pmatrix} l_{11} & l_{21} & l_{31} & \dots & l_{m1} \\ l_{12} & l_{22} & l_{32} & \dots & l_{m2} \\ l_{13} & l_{23} & l_{33} & \dots & l_{m3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{1n} & l_{2n} & l_{3n} & \dots & l_{mn} \end{pmatrix}, \quad (2.36)$$

unde m reprezintă numărul de ieșiri, iar n reprezintă numărul de stări din cadrul sistemului. Așadar, calculul unui estimator reprezintă un avantaj din punct de vedere al costului, cât și din punct de vedere al performanțelor deoarece reușește să elimine erorile de estimare în funcție de viteza stabilită de polii impuși, astfel putând să asigure nevoile unui regulator calculat prin metoda impunerii de poli.

Pe de altă parte, pentru a implementa un estimator fizic, este nevoie de un micro-controller care poate să asigure vitezele de calcul atribuite polilor plasați pentru viteze de reacție de 10 ori mai mari, și de asemenea trebuie să se asigure memoria necesară acestuia. De regulă estimatorul nu există singur, acesta fiind acompaniat de un regulator, care pe baza stărilor acestuia își desfășoară calcularea legii de comandă.

În diagrama de mai jos se evidențiază un sistem care apare în spațiul stărilor, și asupra căruia apar semnale de perturbație atât pe intrare cât și pe ieșire. Estimatorul este prezentat în partea de jos a imaginii, acesta nefiind afectat de perturbații.

Acest lucru constituie un avantaj deoarece, în sensul folosirii stărilor estimate, regulatorul poate să sesizeze mult mai ușor perturbatiile și astfel să le rejeteze.

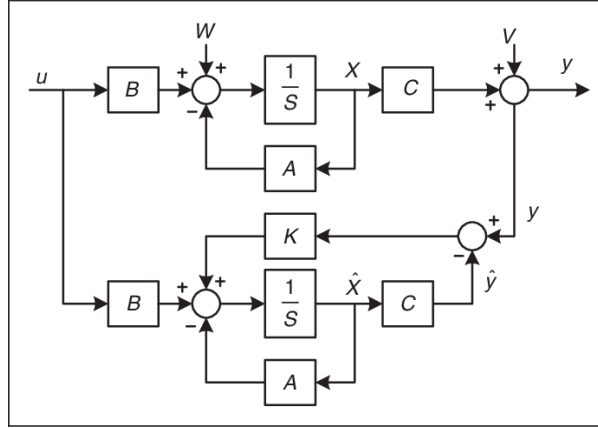


Figura 2.2: Estimator de stare

2.5. LQR și LMI (Linear Matrix Inequations)

Regulatorul Cuadratic Liniar (*eng. Linear Quadratic Regulator – LQR*) reprezintă un regulator menit sistemelor exprimate în spațiul stărilor, realizat prin impunere de poli, dar care optimizează funcția de cost [12] definită astfel:

$$J = \int_0^\infty (x^T Q x + u^T R u) dt, \quad (2.37)$$

unde matricea Q reprezintă matricea costurilor de performanțe ale stărilor, iar R reprezintă matricea de cost a intrărilor. Parametrii Q și R se modifică ținând cont de compromisul între performanțe și efortul de comandă, iar regulatorul se obține prin rezolvarea și optimizarea ecuației Riccati:

$$A^T S + S A - S B R^{-1} B^T S + Q = 0. \quad (2.38)$$

Dupa optimizarea rezultatului se obține S optim care duce la calcularea regulatorului K [13] folosind notația:

$$K = R^{-1} B^T S. \quad (2.39)$$

Această metodă reprezintă un caz particular al modului de lucru dat de inecuațiile liniare matriciale (*eng. Linear Matrix Inequations – LMI*). Acest concept nu reprezintă o metodă în sine de calcul a unui regulator, ci este un metodă de a analiza mai în detaliu un sistem, din punct de vedere matematic.

Astfel, pornind de la reprezentarea unui sistem în spațiul stărilor, se definește stabilitatea după criteriul lui Lyapunov:

$$A^T P + P A < 0. \quad (2.40)$$

Un sistem asimptotic stabil trebuie să îndeplinească această inegalitate, pentru un P pozitiv definit. În cazul impunerii de poli ecuația devine:

$$(A - BK)^T P + P(A - BK) < 0. \quad (2.41)$$

Se poate observa că la bază plasarea de poli defapt trebuie să respecte doar condiția de stabilitate. Astfel, dacă se doresc mai multe constrangeri în legătură cu timpul de răspuns, suprareglajul introdus sau pulsația naturală, se pot deduce inecuații în funcție de care să se realizeze optimizarea regulatorului. Precum este descris și în lucrarea [6], se poate observa o condiție care impactează timpul de răspuns. Aceasta este reprezentată de inecuația de mai jos:

$$A^T P + P A < -2\alpha P. \quad (2.42)$$

Prin considerarea acestei inegalități se asigură faptul că polii care se impun au partea reală mai mică decât $-\alpha$ ceea ce duce la scăderea timpului de răspuns. Totodată, pentru introducerea unei staturății pe semnalul de comandă (u_{max}) se poate folosi următoarea inegalitate prezentată mai jos:

$$\begin{pmatrix} u_{max}^2 I & Y \\ Y^T & Q \end{pmatrix} < 0. \quad (2.43)$$

Folosind imaginea de mai jos [14], se poate impune o zonă în care poli să fie optimizați pentru nevoile impuse.

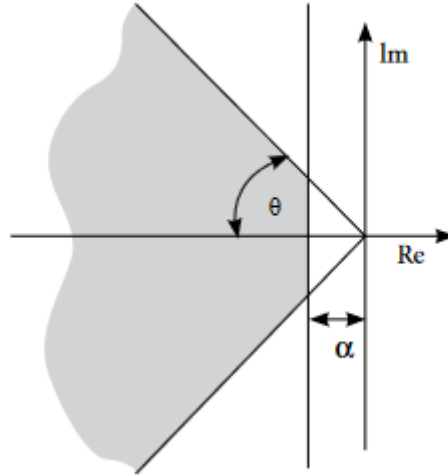


Figura 2.3: Locul rădăcinilor

La fel cum se poate impune partea reală a unui pol să fie într-o regiune, așa se poate impune un unghi maxim θ care să limiteze suprareglajul admis în vederea optimizării performanțelor. De asemenea, se pot impune mai multe condiții prin utilizarea altor inegalități de matrici, ce acționează asupra plasării polilor.

Capitolul 3. Analiză și fundamentare teoretică

3.1. Interpretarea și analizarea problemei

În cadrul acestei lucrări se pune problema controlului vitezei a doua motoare de tip BLDC astfel încât acestea să nu derapeze în scenariul în care o mașină se deplasează pe un o traiectorie curbilinie. Pentru realizarea acestui fapt, trebuie implementat un algoritm de control pentru vitezele motoarelor, iar mai apoi o funcție care reușește să comande o viteză de referință pentru fiecare motor în parte.

3.1.1. Arhitectura proiectului

Proiectul are la bază o parte care se concentrează pe implementarea hardware și o parte care ține de rezolvarea problemelor hardware prin ajutorul software-ului. Astfel, proiectul se descompune în cele doua părți, fapt care este evidențiat în diagrama de mai jos.

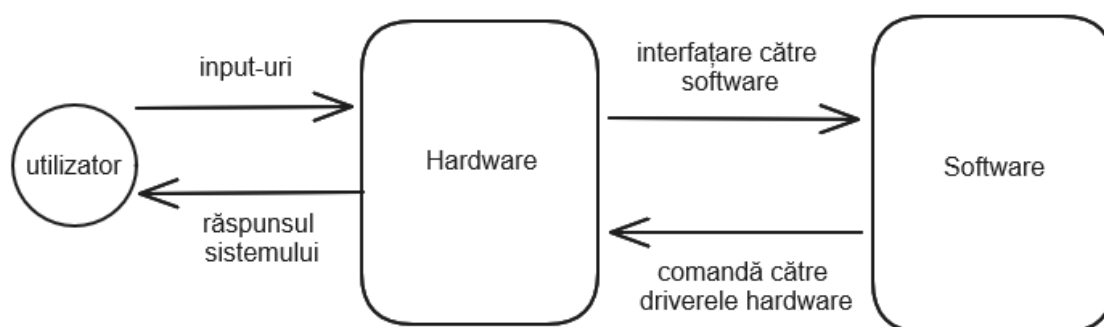


Figura 3.1: Diagrama Conceptuală

Ca mod de funcționare putem urmări cursul de comandă. Utilizatorul dorește modificarea anumiților parametrii, precum viteza de deplasare sau respectiv direcția de deplasare, acționând astfel asupra părților de detecție ale componentei hardware. Acești parametrii sunt mai apoi procesați de către componenta fizică și mapați astfel încât să poată fi recunoscuți de componenta software. După interfațarea inputurilor, software-ul realizează niște calcule astfel încât să asigure buna funcționare a elementelor de execuție și apoi trimite semnalele de comandă driverelor care se ocupă de conducerea proceselor. În final, utilizatorul poate observa răspunsul fizic al sistemului, prin observarea efectelor cerute, precum creșterea respectiv micșorarea vitezei motoarelor, și implicit a mașinii.

Mai mult de atât, componentele hardware și software se împart la randul lor pe categorii. Astfel, pentru componenta de hardware, recunoaștem elemente importante:

- circuite de comandă - aceste circuite se ocupă de comandarea elementelor de execuție

- elementele de execuție - acestea reprezintă elementul fizic care trebuie comandat și controlat în cadrul procesului
- circuitele de detecție - reprezintă senzorii care transmit date despre proces și respectiv analizează datele primite de la utilizator
- circuitul de alimentare - este elementul care asigură alimentarea tuturor componentelor electrice

În funcție de rolul fiecărei componente, partea de software se poate împărți în modul prezentat de lista de mai jos:

- funcția de control - aceasta se ocupă de realizarea logicii de control a motoarelor
- funcția de calcul a referințelor - realizează calculul referințelor motoarelor astfel încât acestea să nu derapeze pe parcursul unei traiectorii curbilinii
- funcția de interpretare a datelor - pe baza datelor primite din partea hardwareului (senzori), se interpretează rezultatele pentru a fi transmise mai departe spre celelalte funcționalități
- funcția de generare a semnalelor de comandă - se generează semnalul adecvat care să fie compatibil cu driver-ul elementelor de execuție, și care trimite informația menită să comande procesul.

3.1.2. Componenta Fizică (Hardware)

După cum s-a enumerat anterior, această parte este alcătuită din mai multe elemente care au relațiile stabilite precum descrie diagrama de mai jos.

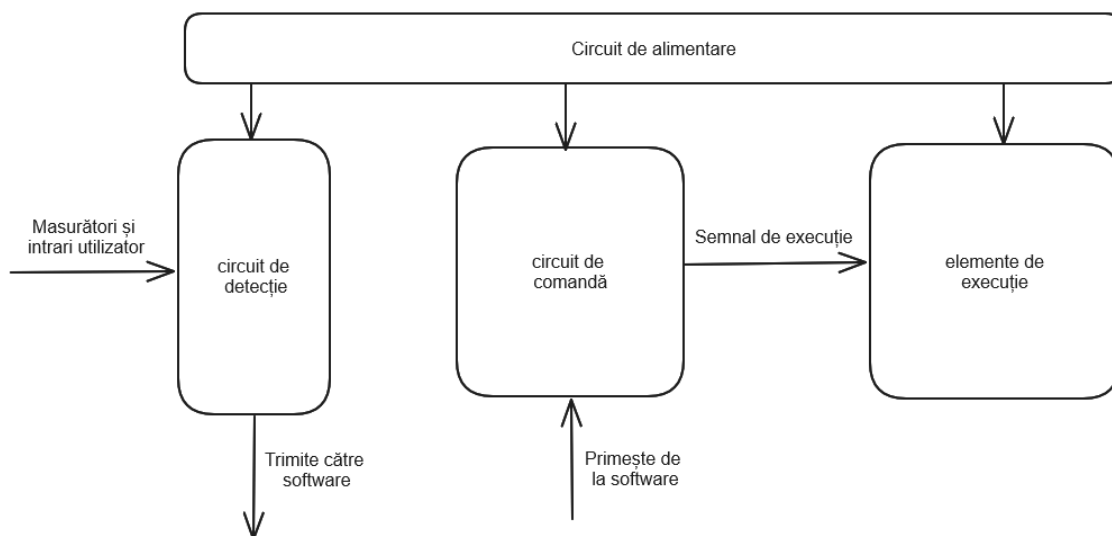


Figura 3.2: Componenta Hardware detaliată

Pe baza acestei diagrame putem observa rolul fiecărui circuit. Astfel, circuitul de detecție realizat prin utilizarea senzorilor obține informațiile necesare la fiecare moment de timp și le transmite prin intermediul semnalelor electrice către componentele software. Circuitul de comandă decodează semnalul primit din partea softwareului, iar pe baza acestuia crează mediul de funcționare dorit în scopul acționării elementului de execuție. În acest sens, motoarele de tip BLDC sunt comandate cu ajutorul invertoarelor, acestea

generând trei faze care să excite anumite bobinaje în cadrul motorului pentru a genera mișcarea de rotație. Circuitul de alimentare, reprezentat de un acumulator Lipo, asigură puterea necesară acționării motoarelor, cât și alimentarea celorlalte componente electrice, cum ar fi senzorii, driverele și respectiv microcontrollerul.

Funcționarea motoarelor BLDC

Numele întreg al acestui gen de motoare este Brushless Direct Current Motors. Pentru a înțelege modul de funcționare al acestora, pornim de la principiul motoarelor de curent continuu, des întâlnite în practică.

Acest gen de motoare prezintă o construcție formată dintr-un rotor (axul motorului care se învârt, generând o tracțiune) și un stator (partea fixă a motorului, în jurul căreia acesta se învârt). Astfel, statorul este format dintr-un bobinaj care este alimentat continuu cu scopul generării unui câmp magnetic, pe când rotorul este de asemenea un bobinaj dar este alimentat prin intermediul unor perii din grafit, aceasta fiind realizată prin apăsarea periilor pe axul rotorului. Mișcarea se realizează prin generarea unui câmp învârtitor (cel dat de stator), care are poziția polilor apropiată de câmpul învârtit (cel dat de rotor). Astfel, prin respingerea reciprocă a câmpurilor mișcarea este produsă, iar cu ajutorul periilor, rotorul își schimbă polaritatea câmpului, mișcarea de rotație putând continua fără comutația semnalului aplicat rotorului.

Așadar, reglarea turației acestor motoare se face ușor prin modificarea tensiunii de alimentare. Prin acest mod, controlul acestor motoare este ușor de realizat, însă problema majoră a acestor motoare este reprezentată de folosirea periilor de grafit, componente sensibile, care nu au o durată lungă de viață. Prin înlăturarea periilor din construcția motoarelor electrice, se ajunge la așa-numitele Brushless DC Motors.

Construcția motoarelor de curent continuu fără perii este asemănătoare cu cea a motoarelor de curent continuu. Acestea prezintă bobinaje pe stator, însă diferența majoră este dată de construcția rotorului. În cazul motoarelor fără perii, rotorul este alcătuit din magneți permanenți, în acest mod problema generării unui câmp magnetic învârtit nu mai există. Astfel, costul înlăturării periilor este reprezentat în creșterea complexității algoritmului de comandă ce trebuie asigurat.

Pentru acționarea motoarelor fără perii, câmpul statorului trebuie modificat continuu astfel încât câmpul magnetic al rotorului să fie mereu de aceeași polaritate și direcție. Prin construcția bobinajelor, motorul lucrează cu trei faze, viteza maximă fiind dată de numărul de poli prezenți pe o înfășurare, iar reglajul vitezei fiind realizat prin modificarea frecvenței semnalelor trifazice

Infasurarea A	Infasurarea B	Infasurarea C
+	-	0
+	0	-
0	+	-
-	+	0
-	0	+
0	-	+

Tabela 3.1: Secvența de comutație pentru rotirea motorului

Pentru secvența de mai sus avem mișcarea spre dreapta motorului, respectiv pentru mișcarea la stânga secvența se scrie invers. Prima problemă care apare, este cum putem

să detectăm următoarea combinație pentru a rotii motorul. Acesată întrebare are ca răspuns fie citirea tensiunii inverse care apare pe faza nulă (cea care nu este energizată) fie cu ajutorul senzorilor cu efect Hall. Acest control realizat de urmărirea secvențelor în funcție de tensiunea inversă reprezintă controlul trapezoidal, nume primit în urma evaluării curenților fazelor.

De notat este faptul că, în cazul acestui control, vectorul rezultat al câmpului magnetic se modifică, acesta scăzând și crescând în funcție de poziția rotorului și secvența curentă activă. Acest lucru impactează tracțiunea maximă care poate fi dezvoltată. Astfel se introduce conceptul de control sinusoidal dezvoltat pentru motoarele PSMS (Permanent Magnet Synchronous Motors), unde se asigură o rezultantă a câmpului magnetic perpendiculară pe vectorul curentului, ducând la o tracțiune maximă ce se păstrează constant în timpul acționării motorului.

Funcționarea invertoarelor

În cazul folosirii motoarelor fără perii, trebuie rezolvată problema generării celor trei faze. Așadar, inverterul joacă rolul driverului care acționează motorul prin generarea a trei faze, pe baza unui semnal dreptunghiular de o anumită frecvență și un anumit factor de umplere, caracteristici clasice ale semnalelor de tip treaptă.

Pornind de la construcția unui inverter, putem înțelege cum se generează cele trei faze. Astfel, circuitul este reprezentat de șase tranzistoare, legate în trei faze paralele, și grupate de câte două. Pentru fiecare fază se folosesc două tranzistoare legate în serie, iar ieșirea pentru fază se află între cele două componente. Prin acest circuit, se realizează defapt o punte de tranzistoare, care realizează cele trei faze prin comandarea a două tranzistoare concomitent, din grupuri diferite.

T_1	T_2	T_3	T_4	T_5	T_6
1	0	0	0	0	1
1	1	0	0	0	0
0	1	1	0	0	0
0	0	1	1	0	0
0	0	0	1	1	0
0	0	0	0	1	1

Tabela 3.2: Secvența de comutație a inverterului

Prin secvența de mai sus se prezintă modul în care se pot genera cele trei faze comutând tranzistoarele, în sensul în care cele trei grupuri paralele sunt formate din (T_1, T_4) , (T_2, T_5) și respectiv (T_3, T_6) . Tranzistoarele acestea sunt acționate de un microcontroller care are secvența de mai sus pusă sub formă de cod, și care reșește să decodeze un semnal de tip PWM, astfel încât să crească sau să scadă frecvența celor trei faze generate. Așadar, pe baza unui factor de umplere se poate modifica turația motorului de tip BLDC.

Funcționarea encoderelor

Având în vedere faptul că pentru controlul vitezei trebuie realizată o buclă de control, apare problema măsurării vitezelor motarelor. În practică există mai multe metode de măsurare a acestei mărimi, însă una dintre cele mai bune este folosirea unui encoder.

Acești senzori reușesc să calculeze viteza motorului și să transmită informația prin intermediul unor standarde precum CAN sau FlexRay. Pentru a înțelege cum un astfel de senzor poate realiza acest lucru trebuie să urmărim construcția acestuia.

În primul rând, un encoder este alcătuit dintr-un senzor optic, un disc cu un număr cunoscut de fante și un microcontroller. Astfel, discul este fixat pe un ax care se rotește cu aceeași viteză ca cea a rotorului motorului. Senzorul optic este plasat astfel încât să poată detecta golurile din discul care se rotește și pe baza cărora trimite impulsuri către microcontroller. În final, microcontrollerul numără câte impulsuri au fost citite într-o perioadă de timp, și raportat la numărul de fante pe care le are discul, se poate determina câte rotații complete au fost făcute în unitatea de timp. După obținerea vitezei, aceasta se transmite pe baza standardelor folosite în industria automotive, pentru a fi procesate mai departe de algoritmi de control.

Un alt motiv pentru folosirea encoderelor este cel de a detecta direcția unei mișcări. Aceste tipuri de encodere poartă numele de encodere de cuadratură. Diferența dintre cele două tipuri de encodere este generată atât de construcția acestora cât și de logica după care lucrează. Un encoder de cuadratură este format dintr-un disc cu un număr de fante, doi senzori optici și un microcontroller. Discul este fixat pe un ax mobil, iar senzorii optici sunt plasați de o parte și de alta a discului, astfel încât la rotirea spre dreapta senzorul optic din dreapta să fie primul care sesizează trecerea prin detectarea unei fante, iar în sens contrar, cel din stânga să fie cel care detectează primul fanta discului. Prin această construcție se asigură patru stări în care senzorii pot să fie și totodată tranziția dintre acestea să fie una clară.

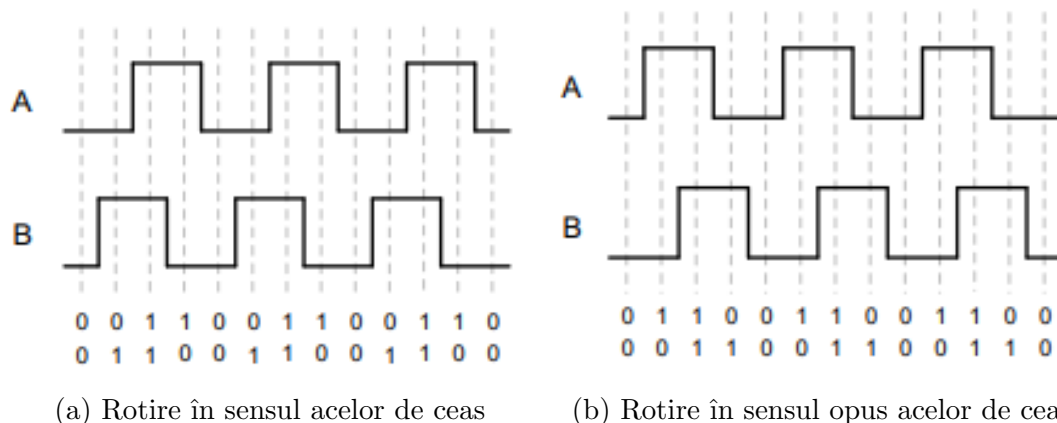


Figura 3.3: Secvențele Encoderului de Quadratură

Așadar, pe baza secvențelor prezentate mai sus putem observa o diferență între mișcarea în sens orar și respectiv cea în sens antiorar. Modul de parcurgere a stărilor este opus unul față de celălalt, astfel putând fi dedusă direcția de rotație. Mai mult de atât, în funcție de numărul de fante ale discului, putem vedea câte grade se deplasează într-o anumită direcție axul encoderului. Aceste informații sunt prelucrate de microcontroller, astfel acesta reușind să transmită direcția și unghiul de deplasare, mai departe pentru prelucrarea acestora în cadrul controlului.

3.1.3. Componenta Software

În cadrul acestei părți, comunicarea între funcționalități și structura abordată sunt prezentate în figura de mai jos.

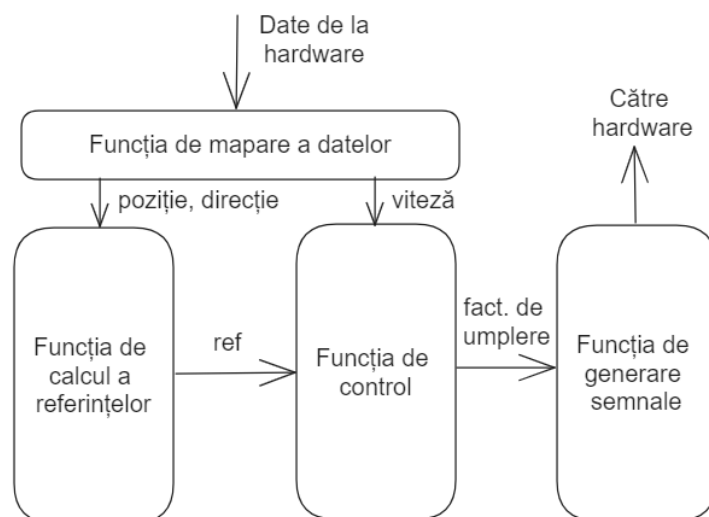


Figura 3.4: Componenta Software detaliată

După cum se poate vedea, funcția de mapare a datelor primește datele obținute de senzori, care se interfațează pentru a putea fi folosite de către celelalte funcții. Astfel, pentru calculul referinței este nevoie de poziția și direcția de deplasare, iar folosind modelul Ackermann, care descrie viteza fiecărui motor în cazul unei traiectorii curbilinii, se calculează referințele celor două motoare. Partea de control preia aceste viteze de referință și totodată vitezele curente ale motoarelor, pentru a decide pe baza unei legi de control, care este comanda pentru atingerea referințelor în cel mai scurt timp posibil. Funcția de generare a semnalelor primește factorul de umplere ce trebuie atribuit unui semnal de tip PWM pentru a transmite driverului ce trebuie să realizeze.

Maparea datelor

Această parte constă în convertirea informației primite de la senzori în informație utilă din punct de vedere al softwareului. Pentru a realiza acest lucru trebuie cunoscut tipul sensorului cu care se lucrează. Astfel, pentru determinarea vitezei, se folosesc senzori optici care detectează trecerile unui disc cu fante legat de elementul învârtitor (precum un encoder). În acest sens, microcontrollerul trebuie să interogheze pinul care este conectat la sensorul optic, iar la fiecare impuls primit, un registru care reține numărul de impulsuri este incrementat. După o perioadă de timp impusă, registrul se resetează, iar viteza se calculează cu ajutorul formulei:

$$\omega = \frac{imps}{nr.fante \cdot T}, \quad (3.1)$$

unde ω reprezintă viteza unghiulară măsurată pe parcursul unei perioade $T[s]$. De observat este faptul că pentru a obține informații corecte despre această mărime fizică, sensorul optic trebuie să asigure faptul că poate detecta tot spectrul de viteze pe care le poate lua motorul și totodată microcontrollerul la rândul lui trebuie să detecteze toate impulsurile date de sensorul optic. În cazul în care există posibilitatea atingerii unor viteze care nu respectă aceste condiții, acestea nu pot fi măsurate. Pentru această situație, microcontrollerul interpretează vitezele mai mari ca fiind valoarea maximă măsurabilă.

$$\omega_{max} = \frac{f_{detectare}}{nr.fante}. \quad (3.2)$$

După cum se poate vedea, folosind formula de mai sus, se poate deduce viteza maximă sesizabilă pe baza frecvenței de citire a impulsurilor și pe baza numărului de fante din cadrul discului de encoder de viteză.

O altă situație de mapare o reprezintă folosirea a doi senzori si un disc cu fante pentru realizarea unui encoder de cuadratură. După cum a fost descris anterior, se folosesc doi senzori optici care trimit impulsuri, iar pe baza acestora, se poate determina starea în care sunt. Bazat pe succesiunea stărilor, se poate afirma direcția și unghiul făcut. Astfel, poziția de zero este marcată la începutul programului, aceasta indicând direcția de mers drept înainte. În cazul unei succesiuni de secvențe, într-o direcție sau alta, un registru incrementează sau decrementează, în funcție de logica dorită. Astfel mișcarea care se face spre dreapta are un semn opus față de mișcarea la stânga. Pentru calculul unghiului realizat de encoderul de cuadratură se poate folosi următoarea relație:

$$\theta = \frac{|imps \cdot nr.fante|}{360^\circ}. \quad (3.3)$$

Ca în cazul anterior, se poate calcula o viteză maximă de variație a unghiului pe baza frecvenței de citire și a numărului de fante.

$$v_{\Delta\theta} = \frac{f_{detectare}}{nr.fante \cdot 4}. \quad (3.4)$$

În cazul folosirii unui potențiomtru pentru a face un reglaj, acesta se conectează la un convertor analogic-numeric, care are rolul de a transforma tensiunea într-o valoare întreagă, în funcție de lungimea registrului pe care îl folosește. Astfel, trebuie să fie asigurat faptul că tensiunea care pică pe potențiomtru este mai mică sau cel mult egală cu tensiunea maximă suportată de convertorul-analogic digital. Astfel, acest convertor realizează conversia tensiunii de la $[0 \text{ nV}]$, într-un număr cuprins între $[0 \cdot 2^{nr.biți} - 1]$, prin compararea tensiunii pe anumite nivele, date de precizia de lucru a acestui convertor.

$$p = \frac{\Delta u}{2^n - 1}. \quad (3.5)$$

Am notat cu p - precizia convertorului analogic digital, Δu reprezintă diferența maximă de tensiune care poate fi măsurată, iar n reprezintă numărul de biți al registrului care este folosit în cadrul conversiei. Se remarcă faptul că o tensiune care nu este exact multiplu al preciziei, este interpretată ca fiind cea mai apropiată tensiune mai mică decât valoarea adevărată, fiind o diferență fundamentală între domeniul continuu și cel discret.

Calcularea referințelor

Pentru această componentă, se pornește de la modelul sistemului geometric al direcției fundamentat de Ackermann. În cazul în care se consideră un autovehicul cu mărimile geometrice cunoscute (raza roților, lungime, lățime, etc.) și care parcurge un drum curbiliniu, se pot calcula vitezele fiecărei roți, astfel încât acest corp să nu alunece.

Pornind de la orientarea mașinii pe direcția dreaptă, vitezele roților trebuie să fie egale. La modificarea direcției, se consideră un punct ca fiind centrul cercului în jurul

căruia se continuă deplasarea curbilinie. Astfel, din cauza geometriei mașinii, o roată se află în spre interiorul drumului, iar una spre exteriorul acestuia. Din cauza faptului că roata exterioară are de parcurs un drum mai lung, pentru a menține pasul cu roata interioară, aceasta trebuie să se învârtă cu o viteză mai mare decât celaltă. În primul rând, pentru a calcula vitezele avem nevoie de raza fiecărui cerc corespunzător roților. Folosind formulele de mai jos, putem determina acești parametrii:

$$R_{int} = R - \frac{T}{2}, \quad (3.6)$$

$$R_{ext} = R + \frac{T}{2}, \quad (3.7)$$

unde T este lățimea autovehiculului, R_{int} este raza cercului interior, R_{ext} este raza cercului exterior, iar R este raza parcursă de centrul mașinii. Noi cunoaștem unghiul de deplasare, așadar se poate calcula R prin relația următoare:

$$\tan \theta = \frac{L}{R}. \quad (3.8)$$

În ecuația de mai sus, se notează cu L - lungimea mașinii. Astfel, folosind ecuația 3.8 se poate calcula unghiul interior și respectiv unghiul exterior:

$$\theta_{int} = \arctan \frac{L}{R_{int}}, \quad (3.9)$$

$$\theta_{ext} = \arctan \frac{L}{R_{ext}}. \quad (3.10)$$

În final, se pot calcula vitezele roților prin folosirea ecuațiilor:

$$v_{int} = v \cdot \cos \theta_{int}, \quad (3.11)$$

$$v_{ext} = v \cdot \cos \theta_{ext}. \quad (3.12)$$

Adițional, în cadrul formulelor de calcul ale vitezelor, se poate ține cont de un factor de alunecare (*eng. slip-ratio*), acesta fiind o caracteristică importantă a roților.

$$v_{int} = v \cdot \cos \theta_{int} \cdot (1 - \lambda_{int}), \quad (3.13)$$

$$v_{ext} = v \cdot \cos \theta_{ext} \cdot (1 - \lambda_{ext}). \quad (3.14)$$

Așadar, pe baza informațiilor de direcție, unghi și viteză de deplasare, am reușit să calculăm vitezele care se vor folosi pe post de referințe în partea de control.

Legea de control

Pentru ca efectul de alunecare să fie redus la minim, trebuie să putem controla vitezele motoarelor, astfel încât acestea să răspundă într-un timp cât mai mic și să prezinte o eroare la poziție mică (sau nulă). Astfel, folosind informațiile deprinse din capitolul doi al acestei lucrări, se pornește de la o funcție de transfer de ordinul întâi, cu un factor de proporționalitate variabil în funcție de viteză:

$$H(s) = \frac{K(\omega)}{Ts + 1}. \quad (3.15)$$

Problema neliniarității introduse de factorul de proporționalitate se poate rezolva prin indentificarea unei funcții care reșește să aproximeze valoarea acestuia. În urma descoperirii unei asemenea funcții, semnalul de comandă se poate înmulți cu inversa acesteia, iar mai apoi să fie folosit ca intrare pentru sistemul nostru. Astfel, modelul devine:

$$H(s) = \frac{1}{Ts + 1}. \quad (3.16)$$

Se poate observa că sistemul este unul liniar, însă costul liniarizării este un calcul al unei funcții la fiecare perioadă de eșantionare.

După liniarizarea sistemului, pentru a asigura un timp cât mai scurt se poate implementa un regulator cu metoda impunerii de poli. Pentru cazul în care considerăm ca stare doar viteza, spațiul stărilor este de felul următor:

$$A = -\frac{1}{T}, B = \frac{1}{T}, C = 1, D = 0. \quad (3.17)$$

După obținerea modelului în spațiul stărilor, se poate impune polul astfel încât sistemul să aibă un timp de răspuns mai mic decât inițial. Totodată, se consideră o limitare pe tensiunea de comandă cu scopul de a nu forța motorul, fapt ce duce la o balansare a performanțelor și a costului energiei de comandă. Pentru asta, această metodă se poate îmbunătăți prin utilizarea metodei LQR, care pe baza matricii de cost al performanțelor Q și a matricii de cost al energiei de comandă R , se poate calcula un regulator care să satisfacă ambele cerințe impuse.

Mai mult de atât, în practică curentul de armătură poate să fie considerat ca o stare. Astfel, pornind de la modelarea unui motor de curent continuu fără perii, obținem ecuațiile diferențiale descrise prin:

$$\frac{d\omega}{dt} = -\frac{K_F}{J} \cdot \omega + \frac{K_T}{J} \cdot I_a - \frac{1}{J} \cdot T_L, \quad (3.18)$$

$$\frac{dI_a}{dt} = -\frac{K_E}{L_a} \cdot \omega - \frac{R_a}{L_a} \cdot I_a + \frac{1}{L_a} \cdot u_a. \quad (3.19)$$

Spațiul stărilor obținut pe baza acestor ecuații este:

$$A = \begin{pmatrix} -\frac{K_F}{J} & \frac{K_T}{J} \\ -\frac{K_E}{L_a} & -\frac{R_a}{L_a} \end{pmatrix} B = \begin{pmatrix} -\frac{1}{J} & 0 \\ 0 & \frac{1}{L_a} \end{pmatrix} C = \begin{pmatrix} 1 & 0 \end{pmatrix}. \quad (3.20)$$

Se remarcă faptul că ieșirea este considerată viteza unghiulară, totodată aceasta fiind singura stare care este măsurată. Pentru implementarea unui regulator prin metoda plasării polilor, trebuie ca sistemul să fie în primul rând controlabil, și avem nevoie de a accesa fiecare stare în parte. Astfel, din cauza lipsei senzorilor de curent, se implementează mai întâi un estimator de stare. Pentru acesta trebuie verificată condiția de observabilitate iar polii estimatorului trebuie plasați astfel încât semnalul de eroare să converge cat mai rapid la zero. În urma calculării estimatorului, folosind inegalitățile de mai jos se pot impune anumite performanțe și limitări.

$$AP + P^T A^T + BZ + Z^T B^T + 2\alpha P < 0, \quad (3.21)$$

$$\begin{pmatrix} AP + P^T A^T + BZ + Z^T B^T & \frac{1}{\tan(\theta)}(AP - PA^T + BZ - Z^T B^T) \\ 0 & AP^T + P^T A^T + BZ + Z^T B^T \end{pmatrix} < 0, \quad (3.22)$$

$$\begin{pmatrix} -u_{max}^2 \cdot I & 0 \\ -Z^T & -Q \end{pmatrix} < 0, \quad (3.23)$$

$$P > 0. \quad (3.24)$$

Astfel, prin folosirea inegalităților de mai sus se impune ca polii sistemului să aibă partea reală mai mică decât $-\alpha$, suprareglajul maxim admis să fie θ , iar performanțele se impun prin matricea Q . Mai mult de atât, pentru viteza maximă admisă în matricea Q , se impune respectarea tensiunii de comandă maxime u_{max} . În final, trebuie ca matricea P să fie pozitiv definită pentru a se respecta condiția de stabilitate. După obținerea regulatorului, acesta se simulează și se observă performanțele și limitările impuse.

Generarea semnalelor PWM

În vederea generării de semnale se folosește convertorul digital analogic. Parametrii unui semnal de tip modulare în lățime a pulsurilor (eng. Pulse Width Modulation – PWM) sunt frecvența semnalului și factorul de umplere. Principiul după care se utilizează această funcționalitate este bazat pe un timer care numără până la o anumită valoare, aceasta reprezentând perioada semnalului. Pentru a impune un factor de umplere, același timer numără atâtea eșantioane cât se dorește să rămână semnalul ON sau OFF. După trecerea acelei valori semnalul comută și astfel se poate implementa conceptul de factor de umplere, care reprezintă în sine un mesaj.

Problema care poate apărea este dată de rezoluția registrilor folosiți și a frecvenței de procesare a microcontrollerului. Astfel, nu se pot atinge orice valori ale factorilor de umplere, fie din cauza perioadelor de eșantionare care nu sunt destul de mici, sau din cauza faptului că timerul numără până la un număr întreg, astfel toate opțiunile de numere fracționare nu beneficiază de un factor de umplere atribuit fizic.

$$rezolutie_{factor\ de\ umplere} = \frac{1}{2^{n-1}} \cdot 100, \quad (3.25)$$

$$F_{PWM} = \frac{F_{clk}}{N \cdot 2^{n-1}}. \quad (3.26)$$

Am notat n ca fiind numărul de biți ai registrului folosit, F_{clk} frecvența de ceas a microcontrollerului și N valoarea prescalerului folosit. Astfel, folosind cele două părți prezentate anterior se poate realiza implementarea proiectului și testarea acestuia.

3.2. Proiectare și implementare

În cadrul acestui capitol se conturează exact implementarea fizică și din punct de vedere software, cât și problemele care apar și soluțiile alese pentru contracararea acestora.

Astfel, pe baza informațiilor prezentate anterior se poate continua conturarea proiectului în categoriile de hardware și software. În cadrul secțiunii de hardware se vor enunța componentele folosite, compatibilitatea dintre ele, modul de funcționare și problemele care trebuie combătute. Pentru secțiunea de software, se analizează metoda de control în detaliu și funcționalitatea de calcul a direcției. De asemenea, se implementează funcționalitățile de encoder și encoder de cuadratură și, totodată, se evidențiază problemele ce pot apărea în integrarea sistemelor discrete cu cele continue.

3.2.1. Proiectarea elementelor fizice

Din punctul de vedere al componentelor, se folosește un microcontroller ATMEGA328p din cadrul unei plăcuțe de dezvoltare Arduino UNO R3. Aceasta asigură pini pentru intrări/ieșiri digitale și analogice, cât și canale de generare a semnalelor de tip PWM. Viteza de procesare de 16[MHz] asigurată de microcontroller este una destul de bună pentru a realiza controlul motoarelor A2122 2450Kv. Astfel, se pune problema comunicării cu motoarele de tip BLDC. Se știe faptul că pentru a acționa acest tip de motoare este nevoie de generarea unui semnal trifazic. Așadar, partea de generare este realizată de invertori 30A, pe baza unui semnal de tip PWM.

Comunicarea între Arduino și Invertor

Aceasta este prima problemă care apare din punct de vedere al componentelor fizice. Pentru rezolvarea acestei situații se apelează la fișa tehnică a invertorului. Astfel se determină faptul că invertorul are mai multe tipuri de intrări (una de putere și una de comandă). Intrarea de putere se referă la conectarea unei surse de alimentare, care este folosită pentru acționarea de putere a motoarelor, iar intrarea de comandă prezintă un pin de intrare, care așteaptă un semnal de tip PWM după următoarele cerințe:

- frecvența semnalului trebuie să fie cuprinsă între [50 60]Hz
- factorul de umplere este considerat 0% pentru nivel logic ON timp de 1[ms]
- factorul de umplere este considerat 100% pentru nivel logic ON timp de 2[ms]

Pe baza acestor cerințe, deducem faptul că pentru un reglaj bun, trebuie asigurată o frecvență mare pentru timerele care se ocupă de generarea semnalului deoarece, timpul util este de 1 [ms], iar pentru reglaj fin trebuie cât mai multe eșantioane de timp în acest interval acceptat de invertor.

Integrarea potentiometrului

Pentru a seta viteza, se folosește un potențiomtru de 220 [kΩ], care se alimentează cu o tensiune de 5 [V], asigurată de plăcuța de dezvoltare Arduino. Astfel, cursorul potențiometrului se leagă la o intrare analogică pentru a putea fi citită tensiunea ce pică pe porțiunea folosită din cadrul componentei. În acest sens, tensiunile de la [0 5][V] se pot folosi ca opțiuni de viteze. Totodată, apare problema enunțată anterior, aceasta fiind faptul că nu toate tensiunile pot să fie interpretate, din cauza rezoluției convertorului analogic-digital. Folosind formula rezoluției obținem în cazul nostru următorul rezultat:

$$p = \frac{\Delta u}{2^n - 1} = \frac{5}{1023} = 0.004887. \quad (3.27)$$

De notat este faptul că pentru o plăcuță Arduino UNO R3, convertorul analogic-digital lucrează cu un registru pe 10 biți, astfel rezoluția este cea obținută mai sus. Faptul indicat de această valoare este că pentru diferențe de tensiune mai mici decât 0.004887, nu sunt interpretate.

Comunicarea Senzorilor optici

Pentru ca viteza de rotație să fie măsurată, se folosește ansamblul format dintr-un disc de encoder cu 8 fante fixat pe rotorul motorului și un senzor optic LM393. Acest senzor este montat astfel încât discul se învârtă între fantele ce delimitează traiectoria fasciculului de lumină infraroșie. Prin acest montaj, senzorul trimite impulsuri la fiecare fantă depistată. Aceste impulsuri sunt preluate pe un canal de intrare digitală din cadrul microcontrollerului. Viteza maximă care poate să fie depistată este:

$$\omega_{max} = \frac{f_{detectare}}{nr.fante} = \frac{2 \cdot 10^5}{8} = 25 \cdot 10^3 [rps]. \quad (3.28)$$

În cadrul frecvenței se ține cont de timpul de adresare a registrului de intreruperi și a citirii efective a impulsurilor ($T = 5 [\mu s]$ sau o frecvență de $2 \cdot 10^5$).

În cazul detecției direcției și a unghiului de deplasare, se realizează montajul unui encoder de cuadratură. Doi senzori optici care se pun cap în cap iar prin deschizătura lor se montează un disc cu mai multe fante. Acest ansamblu este așa proiectat încât se respectă parcurgerea stărilor enunțate în capitolul anterior. Impulsurile senzorilor sunt preluate de intrări digitale ale microcontrollerului. De notat este faptul că acești senzori nu trebuie să fie de viteze mari, deoarece discul encoderului se consideră legat de volanul unui autovehicul, iar viteza de rotație a volanului este mult mai mică decât viteza de rotație a unui motor. Viteza maximă de variație a unghiului este:

$$v_{\Delta\theta} = \frac{f_{detectare}}{nr.fante \cdot 4} = \frac{2 \cdot 10^5}{32} = 6250 [rps]. \quad (3.29)$$

Alegerea sursei de alimentare

Pentru acționarea de putere a motoarelor s-a ales ABSIMA 5000 mAh 3s, un acumulator Li-Po care are destulă putere să acționeze pentru o perioadă două motoare BLDC a2122 2450Kv. Tehnologia acumulatorilor Li-Po este des întâlnită, însă trebuie respectate tensiunile de lucru. Tensiunile recomandate acumulatorilor cu 3 celule în serie sunt 11.12 și 12.6. Astfel, deoarece ambele motoare folosesc aceeași sursă de alimentare, se face o limitare de comandă pentru a nu supra-solicita acumulatorul.

3.2.2. Proiectarea elementelor software și de simulare

În această secțiune se descriu pașii făcuți pentru implementarea logicii de control, cea de interpretare a datelor și cea de calcul a vitezelor în funcție de modelul geometric Ackermann, ce ține cont de dimensiunile unei mașini.

Asigurarea controlului și performanțelor

Această categorie se împarte în două părți. Prima parte este reprezentată de identificarea unei funcții de transfer care are ca intrare factorul de umplere și ca ieșire viteza.

Pentru această funcție de transfer se calculează un regulator, iar acesta se implementează pe microcontroller cu scopul observării performanțelor sistemului. A doua parte este reprezentată de aprofundarea sistemului abordat și se pornește de la ecuațiile fizice ale motorului cu scopul îmbunătățirii performanțelor printr-un control mai avansat. Această parte este doar simulată, nu și implementată deoarece microcontrollerul nu poate realiza tehnica de control dorită.

Astfel, pornind pe primul plan, se assemblează sistemul astfel încât un motor să poată fi comandat, iar viteza acestuia să poată fi măsurată. Pentru obținerea datelor, plăcuta Arduino se leagă la un calculator și trimite serial viteza și factorul de umplere folosit pentru obținerea vitezei respective. Datele sunt luate pentru viteze diferite și pe fronturi diferite (front crescător - viteza să crească, front descrescător - viteza să scadă). Identificarea se face folosind vectorul de intrări $u = [0 \ 0.08 \ 0.04 \ 0.12 \ 0.1 \ 0.16 \ 0.08 \ 0]$. Pentru aceste intrări se obține vitezele (după stabilizare) $v = [0 \ 135 \ 98 \ 166 \ 151 \ 193 \ 135 \ 0]$. De menționat este faptul că datele au fost preluate cu o perioadă de eșantionare de 100 [ms], iar vitezele obținute sunt măsurate în [rps]. Totodată, dinamica motorului se poate observa în figura de mai jos.

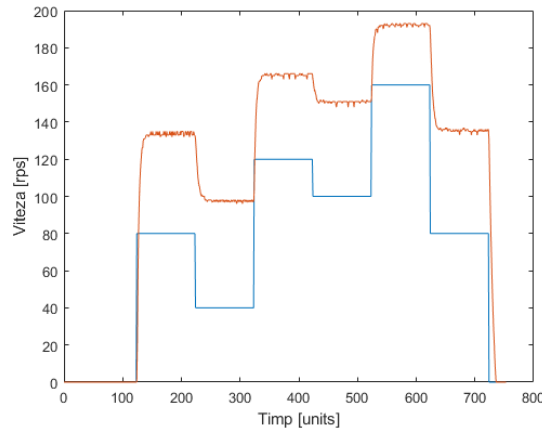


Figura 3.5: Rosu-viteza masurata, Albastru-factorul de umplere x 1000

Sistemul se poate considera unul de ordinul întâi, fiind de forma următoare:

$$H(s) = \frac{K}{Ts + 1}. \quad (3.30)$$

Pentru identificarea parametrului K, se poate utiliza formula:

$$K = \frac{\Delta v}{\Delta u} = \frac{v_{st} - v_0}{u_{st} - u_0}. \quad (3.31)$$

Pentru prima treaptă, se obține $K = 1650$. În cazul constantei de timp a sistemului T, se poate urmări faptul că sistemul răspunde după $T_s = 3.2$ [s] (32 de eșantioane). Astfel folosind formula de mai jos obținem parametrul căutat:

$$T_s = 4 \cdot T, \quad T = \frac{T_s}{4} = 0.8[s]. \quad (3.32)$$

Prin urmare, o primă funcție de transfer este obținută:

$$H(s) = \frac{1650}{0.8s + 1}. \quad (3.33)$$

Problema apare când se vizualizează suprapunerea acestui sistem identificat cu sistemul fizic. În figura de mai jos se observă diferențele valorilor staționare, fapt ce indică neliniaritatea sistemului. De asemenea, se observă că dinamica sistemului nu este afectată de neliniaritate. Acest lucru consolidează ideea că forma adevărată a funcției de transfer este, de fapt cea descrisă mai jos:

$$H(s) = \frac{K(\omega)}{0.8s + 1}. \quad (3.34)$$

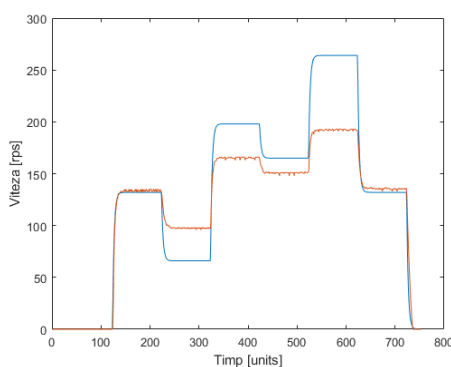


Figura 3.6: Evidențierea neliniarității

Astfel, pentru a rezolva problema neliniarității, se acționează motorul cu factori de umplere începând de la 4%, fiind incrementat cu 1% la fiecare 5[s] până la atingerea valorii de 20%. Datele citite serial sunt reprezentate de viteza motorului și factorul de proporționalitate (calculat de microcontroller folosind formula de mai sus). Din datele obținute se poate observa că evoluția factorului de proporționalitate se poate aproxima printr-o funcție de ordinul doi.

$$K(\omega) = 0.06328\omega^2 - 29.11\omega + 4417. \quad (3.35)$$

În imaginea de mai jos se observă evoluția gainului în raport cu viteza motorului.

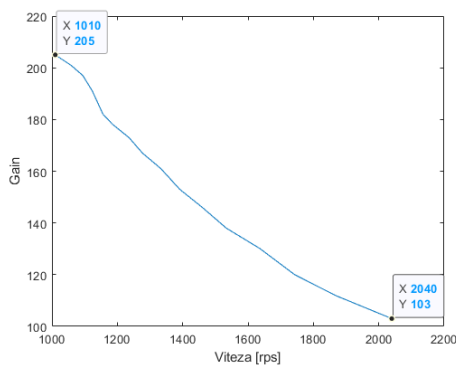


Figura 3.7: Evoluția gainului neliniar

Pentru înlăturarea acestui efect, se poate folosi conceptul de Feedback Linearization, care presupune liniarizarea pe baza stărilor. În cazul de față, se poate măsura viteza, iar comanda se poate trece printr-un bloc de calcul care execută următorul calcul:

$$u = \frac{v}{K(\omega)} = \frac{v}{0.06328\omega^2 - 29.11\omega + 4417}. \quad (3.36)$$

Noua noastră intrare se consideră v , iar sistemul liniarizat arată sub forma următoare:

$$H(s) = \frac{1}{Ts + 1} = \frac{1}{0.8s + 1}. \quad (3.37)$$

După rezolvarea problemei de liniarizare, se pune problema de control. Astfel, pentru implementarea unui regulator prin metoda impunerii de poli se face trecerea în spațiul stărilor, realizată astfel:

$$A = \frac{-1}{T} = -1.25, \quad B = \frac{1}{T} = 1.25, \quad C = 1, \quad D = 0. \quad (3.38)$$

Pentru calculul regulatorului, se poate folosi metoda LQR de plasare a polilor astfel încât costul performanțelor (dat de matricea Q) cât și costul acționării (dat de matricea R) să fie îndeplinite. În acest sens se calculează două tipuri de regulatoare, unul care respectă saturația impusă pe comandă (o saturație de 20% pentru a nu forța sursa de alimentare) și unul care este mai agresiv și nu respectă saturația impusă.

Așadar, pentru regulatorul care respectă saturația comenzii se impun valorile următoare ale matricilor (în acest caz, scalarilor):

$$Q = 7, \quad R = 1. \quad (3.39)$$

Pentru regulatorul agresiv, se impun valorile de mai jos:

$$Q = 7, \quad R = 0.1. \quad (3.40)$$

Se reamintește faptul că pentru a obține regulatorul, se optimizează funcția de cost:

$$J = \int_0^\infty x^T Q x + u^T R u. \quad (3.41)$$

Astfel, regulatoarele obținute sunt:

$$K_{sat} = 1.8284, \quad K_{agr} = 7.4261. \quad (3.42)$$

Este important de precizat că pentru această metodă de calcul a unui regulator nu se asigură direct eroarea staționară cât mai mică. Pentru acest lucru se poate calcula o matrice de prefiltrare după formula:

$$K_f = [C(A - B \cdot K)^{-1} B]^{-1}. \quad (3.43)$$

De altfel, în caz experimental, se poate calcula această matrice prin aplicarea unui semnal de tip treaptă unitară astfel obținând răspunsul sistemului, iar valoarea gainului de prefiltrare este calculat prin:

$$K_f = \frac{1}{y(0)}. \quad (3.44)$$

S-a notat cu $y(0)$ valoarea staționară a sistemului pentru un semnal de intrare de tip treaptă unitară. Astfel, pentru fiecare regulator se calculează această matrice de prefiltrare și se obțin următoarele valori:

$$K_f^{sat} = 2.8281, \quad K_f^{agr} = 8.4246. \quad (3.45)$$

În vederea realizării reguletoarelor în domeniul discret, se pornește de la discretizarea în sens Euler descrisă prin:

$$\dot{x} = \frac{x_k - x_{k-1}}{T_e}, \quad (3.46)$$

unde se notează cu x_k eșantionul curent al stării, x_{k-1} eșantionul trecut al stării și respectiv T_e perioada de eșantionare. Dacă se înlocuiește această notație în spațiul stărilor se obține:

$$x_k = (I + AT_e) \cdot x_{k-1} + BT_e \cdot u. \quad (3.47)$$

Pentru sistemul discret se aplică aceși condiție de optimizare a costului pe baza costului, cu aleași matrici pentru costul performanțelor și costul acționării, astfel reguletoarele obțin valorile de mai jos:

$$\begin{aligned} K_{sat} &= 1.5442, & K_{sat}^f &= 2.5439 \\ K_{agr} &= 4.7003, & K_{agr}^f &= 5.7013 \end{aligned} \quad (3.48)$$

A doua parte a analizei de control se bazează pe modelarea sistemului pornind de la ecuațiile diferențiale 2.1, 2.2:

$$\frac{d\omega}{dt} = -\frac{K_F}{J} \cdot \omega + \frac{K_T}{J} \cdot I_a - \frac{1}{J} \cdot T_L, \quad (3.49)$$

$$\frac{dI_a}{dt} = -\frac{K_E}{L_a} \cdot \omega - \frac{R_a}{L_a} \cdot I_a + \frac{1}{L_a} \cdot u_a. \quad (3.50)$$

Parametrii sunt identificați prin măsurare efectivă și prin parcurgerea datelor din fișa de specificații tehnice a motorului. În tabelul de mai jos sunt prezentate valorile acestora.

Numele parametrului	Simbol	Valoare
Constanta tensiunii inverse	K_E	0.0039 [V · s/rad]
Constanta tracțiunii	K_T	0.1331 [N · m/A]
Coeficient de frecare	K_F	0.00291 [N · m · s/rad]
Rezistența de armătură	R_a	0.05 [Ω]
Inductanța armăturii	L_a	0.1 [H]
Momentul de inerție al motorului	J	$5.99 \cdot 10^{-8}$ [kg · m ²]

Tabela 3.3: Parametrii motorului BLDC a2122 2450Kv

Sistemul scris în spațiul stărilor este reprezentat de matricile:

$$A = \begin{pmatrix} -\frac{K_F}{J} & \frac{K_T}{J} \cdot 2\pi \\ -\frac{K_E}{L_a \cdot 2\pi} & -\frac{R_a}{L_a} \end{pmatrix}, \quad B = \begin{pmatrix} -\frac{2\pi}{J} & 0 \\ 0 & \frac{1}{L_a} \end{pmatrix}. \quad (3.51)$$

Se observă faptul că apare 2π , pentru maparea vitezei din [rad/s] în [rps]. Asemănător cu prima parte, apare problema unui gain neliniar pe intrarea de tensiune a sistemului.

În aceeași manieră se indentifică funcția după care evoluează factorul proporțional în raport cu viteza unghiulară, obținându-se următoarea relație:

$$K(\omega) = 9.2289 \cdot 10^{-5} \cdot \omega^2 + 0.0371 \cdot \omega + 4.6870. \quad (3.52)$$

În aceeași manieră, acest gain neliniar se elimină și sistemul se poate analiza ca unul liniar. Este considerat faptul că ieșirea este viteza unghiulară a sistemului și doar această stare este măsurată. Astfel, matricile folosite în spațiul stărilor sunt:

$$A = \begin{pmatrix} -48580.9682 & 13961468.520 \\ -0.006207 & -0.5 \end{pmatrix}, \quad (3.53)$$

$$B = \begin{pmatrix} -104894579.418 & 0 \\ 0 & 10 \end{pmatrix},$$

$$C = (1 \ 0), \quad D = 0.$$

Pentru a calcula estimatorul de stare, se verifică condiția de observabilitate și se calculează polii sistemului. Matricea de observabilitate este :

$$Co(A, C) = \begin{pmatrix} C \\ CA \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ -48580.968 & 13961468.52 \end{pmatrix}. \quad (3.54)$$

Se observă faptul că rangul matricei este doi, ceea ce rezultă că sistemul este observabil. Pentru calcularea polilor se folosește ecuația de mai jos:

$$\det(sI - A) = 0. \quad (3.55)$$

Polii obținuți în urma rezolvării ecuației de mai sus sunt:

$$\hat{s} = \{-48579.1843, -2.2839\}. \quad (3.56)$$

Estimatorul se dorește să fie mai rapid decât procesul pentru ca eroarea să convergă repede la zero, astfel, polii estimatorului sunt aleși de zece ori mai rapizi, cu valorile de mai jos:

$$s_{est} = \{-485791.843, -22.839\}. \quad (3.57)$$

Pentru calculul estimatorului în sine, se folosește formula Ackermann descrisă în studiul bibliografic la capitolul estimatoarelor:

$$L^T = Ob(A, C)^{-1} \cdot e_n^T \cdot P_0(A)^T. \quad (3.58)$$

Astfel matricea obținută este:

$$L = \begin{pmatrix} 437233.214 \\ 0.7710 \end{pmatrix}. \quad (3.59)$$

Pentru calculul regulatorului prin impunere de poli, se verifică controlabilitatea sistemului prin calculul rangului matricii de controlabilitate:

$$Co(A, B) = (B \ AB) = \begin{pmatrix} -104.89 \cdot 10^6 & 0 & 50958.8 \cdot 10^6 & 139.61 \cdot 10^6 \\ 0 & 10 & 0.651 \cdot 10^6 & -5 \end{pmatrix}. \quad (3.60)$$

Rangul matricii de controlabilitate este egal cu numărul de stări ale sistemului, deci acesta este controlabil. În sensul calculării regulatorului, se impun inecuațiile descrise în capitolul anterior, unde $\alpha = 10$ (partea reală a polilor să fie mai mică decât -10 fapt ce reduce timpul de răspuns), $\theta = 0.1$ (suprareglajul nu are voie să depășească valoarea de 0.1%), $u_{max} = 1$, iar matricea $Q = \begin{pmatrix} 1/2 \cdot (\frac{1}{220})^2 & 0 \\ 0 & 1/2 \cdot (\frac{1}{20})^2 \end{pmatrix}$.

În urma impunerii condițiilor împreună cu parametrii specificați mai sus se obține următorul regulator care realizează impunerea de poli:

$$K = \begin{pmatrix} 0.000463 & -0.1331 \\ -0.00078 & 1.5295 \end{pmatrix}. \quad (3.61)$$

Unde prima linie a acestuia sunt factorii de proporționalitate pentru reglajul buclei de viteză iar a doua linie este pentru bucla de curent.

Parametrii modelului geometric Ackermann

După cum a fost realtat și în capitolul anterior, pentru modelul geometric introdus de Ackermann, avem nevoie de niște detalii fizice. Astfel, se consideră folosirea detaliilor unui Porsche Taycan, o mașină electrică care poate beneficia de problema studiată în această lucrare. Se deduce faptul că lungimea considerată în cadrul ecuațiilor (notată cu L) este de 4 [m], iar lățimea (notată cu T) este de 2 [m]. Totodată, pentru determinarea mapării vitezei de translație în viteză de rotație este nevoie de cunoașterea razei roților (notată cu R) folosite. Așadar această valoare este de 0.26 [m].

Mai mult de atât, se pot considera parametrii precum factorul de alunecare al roților (notat cu λ) care poate varia între 0.1 și 0.15. Acesta asigură o marjă de eroare pentru combateră imperfecțiunilor roților și totodată a diferențelor între acestea. În cazul aplicației folosite, nu s-a augmentat funcția de direcție, aceasta calculând vitezele pe baza ecuațiilor de mai jos:

$$v_{int} = v \cdot \cos \theta_{int}, \quad (3.62)$$

$$v_{ext} = v \cdot \cos \theta_{ext}. \quad (3.63)$$

Implementarea funcționalităților pe μC

În cadrul acestei părți se explică implementarea efectivă a funcționalităților pe microcontroller. Astfel, se recunosc funcțiile de citire a datelor și de interpretare ale acestora, funcția de calcul a referințelor pe baza modelului Ackermann și respectiv funcția de control a sistemului. Programul este scris cu ajutorul limbajului de programare C/C++, în mediul de dezvoltare Arduino IDE. Din cauza faptului că microcontrollerul are un singur core (unitate de procesare), trebuie realizată o structură care planifică acționarea fiecărei funcții, în raport cu importanța acesteia.

Astfel, se folosește o bibliotecă pe nume TaskScheduler, prin care se poate crea un obiect scheduler (planificator) care reține o listă de taskuri (proces) ce trebuie rulate, ordinea lor în listă dând de asemenea și importanța lor. Se știe faptul că în cazul în care două taskuri doresc executarea deodată, cel cu prioritatea mai mare este executat primul. De asemenea, apare conceptul preemptiv multitasking, care se referă la problema folosirii microcontrollerului. În cazul în care procesorul este folosit de un task și în timpul în

care acesta este folosit apare unul mai prioritar, un sistem preemptiv oprește executarea taskului inițial și permite executarea celui prioritar. După finalizarea acestuia, sistemul se întoarce la taskul inițial și permite desfășurarea acestuia din punctul opririi. Această librărie este una care asigură preemptivitatea programului.

Pentru cazul nostru, procesele sunt următoarele:

- sesizarea impulsurilor encoderelor de viteză - se execută o dată la fiecare 0.5 [ms]
- citirea direcției și a unghiului - se execută cu o perioadă de 1 [ms]
- citirea vitezei dată de utilizator - se execută o dată la fiecare 200[ms]
- calcularea referințelor - se acționează la fiecare 100[ms]
- calcularea comenzilor - se execută cu o perioadă de 100 [ms]

În scopul înțelegerii programului, se începe cu datele care se citesc de la utilizator. Se știe că viteza este setată prin intermediul unui potențiomtru care are cursorul legat la un convertor analogi-digital. Astfel prin rotirea potențiometrului se poate seta o anumită viteză. Din cauza zgomotelor care pot apărea, maparea potențiometrului este făcută în modul următor:

- $val_{ADC} \in [0, 233)$ - viteza impusă este de 0 [km/h]
- $val_{ADC} \in [233, 423)$ - viteza impusă este de 132 [km/h]
- $val_{ADC} \in [423, 623)$ - viteza impusă este de 152 [km/h]
- $val_{ADC} \in [623, 823)$ - viteza impusă este de 172 [km/h]
- $val_{ADC} \in [823, 1023]$ - viteza impusă este de 192 [km/h]

Se observă faptul că vitezele sunt puse în [km/h]. Acest lucru este făcut în această manieră deoarece un șofer dorește să acționeze viteza mașinii nu viteza de rotație a motorului.

Pentru direcție și unghiul descris de traiectoria curbilinie, se folosește encoderul de cuadratură, care are cei doi senzori legați la intrările digitale oferite de plăcuța de dezvoltare. De reținut este faptul că pinii nu pot să fie schimbați între ei, iar în cazul legării greșite programul interpretează mișcarea este interpretată pe dos. Funcționarea acestei componente este dată de urmărirea secvenței descrise în capitolul anterior. Astfel, se citește la fiecare milisecundă starea senzorilor, iar pe baza acesteia, se decide starea encoderului. La întoarcerea în starea inițială (aceasta se consideră pentru semnal high pentru ambii senzori) se decide incrementarea sau decrementarea unui counter care numără câte impulsuri s-au făcut într-o anumită direcție. Decizia se ia pe baza stării din care se vine în starea inițială. Dacă se vine din starea dată de semnal high pe senzorul stâng și low pe senzorul drept atunci counterul se incrementează. Dacă se vine din altă stare counterul se decrementează. Maparea direcției și a unghiului se face în felul următor:

- numărul de impulsuri numărate de counter reprezintă câte grade are unghiul deplasării ($1 \text{ imp} = 1^\circ$)
- semnul valorii din counter determină direcția (> 0 reprezintă deplasare spre stânga, < 0 reprezintă deplasare spre dreapta)

Se observă faptul că, deși fac în sine același lucru, funcția de citire a vitezei date de utilizator se execută mult mai încet decât cea de citire a direcției și a unghiului. Acest lucru este gândit așa deoarece se consideră mul mai importantă direcția de deplasare

decât viteza în sine. Astfel, în cazul unei urgențe, schimbarea direcției este sesizată iar viteza motoarelor se modifică în funcție de aceasta, păstrând în calcule viteza anterioară. Asta ajută împotriva schimbărilor bruște de viteză ce pot duce la accidente.

O altă funcție este cea de citirea a vitezelor efective ale motoarelor. Această citește la o perioadă de timp de 0.5[ms] senzorul optic. Citirea se face cu o viteză mare deoarece motorul se învâрте, iar dacă citirea se face mai lent decât viteza de rotire a motorului, măsurătoarea nu este viabilă. Pentru viteza de citire asignată, este posibilă citirea unei viteze maxime de 250[rps]. Maparea efectivă a impulsurilor se face în funcția de calculare a comenzilor, prin înmulțirea impulsurilor numărate cu 1.25 (se știe că taskul de comandă se execută o dată la 100 [ms], iar numărul de fante ale discului este de 8, iar pentru a face conversia de la impulsuri la [rps] se înmulțește cu $\frac{10}{8} = 1.25$).

În cadrul funcționalității de comandă, pașii făcuți în vederea realizării acesteia sunt următorii:

1. se realizează maparea vitezei citite și se resetează counterul encoderelor
2. se calculează comanda (factorul de umplere) care trebuie trimis către invertore
3. se generează semnalul PWM pe baza comenzii și a cerințelor driverelor

Calculul comenzii se face prin implementarea legii de control pe baza factorilor de proporționalitate obținuți prin metoda impunerii de poli. Pentru început eroarea, notată cu err , se calculează după formula ce apare mai jos:

$$err = K_f \cdot ref - K \cdot speed, \quad (3.64)$$

unde ref reprezintă referința impusă, K_f este matricea de prefiltrare, K este regulatorul care impune polii, iar $speed$ este viteza unghiulară citită. Pentru calcularea semnalului de comandă, notat cu com , se reamintește faptul că se aplică un calcul adițional în vederea liniarizării sistemului.

$$com = \frac{err}{0.06328 \cdot speed^2 - 29.11 \cdot speed + 4417}. \quad (3.65)$$

Mai mult de atât, comanda este limitată la o valoare maximă de 0.2 (20%) și o valoare minimă de 0.04 (4%). De asemenea, aici apare o problemă des întâlnită în practică. Din cauza faptului că există situații în care se dorește atingerea unei viteze care mapată în factor de umplere nu poate fi realizată din cauza rezoluției registrului convertorului digital analogic, comanda prezintă un efect de oscilație. În general, oscilațiile semnalului de comandă nu sunt dorite, cu atât mai mult în cazul sistemelor care răspund rapid. Astfel, pentru evitarea acestei probleme, comanda se aproximează la cel mai apropiat factor de proporționalitate care poate să fie atins.

Problema oscilației comenzii mai poate să fie dată și de zgomotul de măsurare a senzorilor optici. În cazul în care zgomotul este sesizabil, programul percepe că motorul merge mai tare sau mai încet, însă în realitate merge exact la fel. Pentru această problemă se calculează factorul de umplere la care trebuie să se stabilizeze comanda, folosind formula de mai sus doar că puțin schimbată:

$$com = \frac{ref}{0.06328 \cdot ref^2 - 29.11 \cdot ref + 4417}. \quad (3.66)$$

În urma calculării comenzii, trebuie generat un semnal PWM care să transmită comanda spre driver pe baza condițiilor impuse. Se știe din capitolul anterior faptul că semnalul PWM trebuie să aibă o frecvență de 60 Hz, cu factorul de 0% fiind reprezentat prin menținerea nivelului logic high pe o durată de 1 [ms], iar 100% fiind reprezentat prin menținerea nivelului logic high pe o durată de 2 [ms].

Astfel, se folosește timerul TIM1 cu următoarele configurări:

Numele registrului	Valoare
TCCR1A	0b10100010
TCCR1B	0b00011010
ICR1	40000
TCNT1	0
OCR1A	0
OCR1B	0

Tabela 3.4: Configurările timerului 1

Regiștrii TCCR1A și TCCR1B sunt folosiți pentru selectarea modului de operare, în cazul nostru, non-inverting PWM phase correct și prescalerul de valoare 8. Perioada semnalului PWM este dată de ICR1. Timerul numără fiecare eșantion, valoarea curentă fiind salvată în TCNT1. Când se ajunge la valoarea OCR1 semnalul trece din low în high, iar când se ajunge la valoarea ICR1, counterul se resetează. De remarcat este faptul că folosind același timer, se pot genera două semnale PWM care pot avea factori de umplere diferiți. Se observă de asemenea că pentru un prescaler de 8 și valoarea maximă de 40000, se obține un semnal cu o frecvență de $\frac{16\text{MHz}}{8 \cdot 40000} = 50\text{Hz}$.

Astfel, pentru îndeplinirea mapării cerute folosim următoarea formulă:

$$OCR1 = 2000 \cdot com + 2000. \quad (3.67)$$

Se remarcă faptul că pentru configurările făcute, 2000 de tacti înseamnă o perioadă de 1[ms], astfel dacă comanda este 0% atunci OCR1 stabilește un semnal PWM care ține nivelul de high timp de 1 [ms]. Totodată, prin faptul că se calculează un procentaj pentru comandă, acesta se poate adăuga peste valoarea de 0, astfel satisfăcând maparea cerută de invertore și asigurând comunicarea cu acesta.

Pentru funcția de calculare a direcției, se preia viteza măsurată în [km/h] (notată cu v) și se mapează în [rps] (notată cu ω) prin folosirea formulei:

$$\omega = \frac{v}{2\pi R}, \quad (3.68)$$

unde R este raza roții considerate. Astfel, pentru calcularea razei curbei este nevoie de o conversie în radiani a unghiului deoarece funcția $\tan()$ așteaptă valori date în radiani. Unghiul intern și cel extern se calculează prin:

$$\theta_{int} = \text{atan}\left(\frac{L}{R_c - T/2}\right), \quad (3.69)$$

$$\theta_{ext} = \text{atan}\left(\frac{L}{R_c + T/2}\right), \quad (3.70)$$

unde se definește de asemea variabila R_c , care reprezintă raza curburii, prin:

$$R_c = \frac{L}{\tan(\text{angle})}, \quad (3.71)$$

iar angle este valoarea în radiani a unghiului de deplasare. Diferențierea dintre cele două se face prin verificarea semnelor counterului, astfel ajungându-se la trei cazuri:

- dacă valoarea este pozitivă - motorul din stânga este cel din interior, iar cel din dreapta este cel din exterior
- dacă valoarea este negativă - motorul din stânga este cel din exterior, iar cel din dreapta este în interior
- dacă valoarea este zero - ambele motoare trebuie comandate în același mod deoarece urmează un drum drept

3.3. Testare și Validare

În cadrul acestui capitol se urmăresc metodele de testare și validare ale părții de control și ale rezultatelor fizice obținute. Totodată, se urmăresc rezultatele obținute în urma realizării simulării avansate a controlului prin intermediul mediului de dezvoltare MATLAB/Simulink. Astfel, acest capitol se poate împărți în următoarele părți:

- simularea sistemului identificat și compararea acestuia cu sistemul real
- simularea și evidențierea liniarizării obținute
- simularea celor două regulatoare (cel care respectă saturația și cel care nu)
- testarea fizică a reguletoarelor și compararea rezultatelor obținute
- testarea fizică a motorului împreună cu funcția de direcție
- simularea sistemului cu estimator de stare și regulator și interpretarea rezultatelor acestora

3.3.1. Simularea sistemului identificat

După cum s-a evidențiat în capitolul anterior, s-au identificat două sisteme care pretind să aproximeze dinamica sistemului. Primul sistem identificat este în forma funcției de transfer prezentată mai jos:

$$H(s) = \frac{K(\omega)}{Ts + 1} = \frac{0.06328\omega^2 - 29.11\omega + 4417}{0.8s + 1}. \quad (3.72)$$

Pentru validare, se construiește funcția de mai sus într-un bloc Simulink. Intrarea blocului este dată de vectorul folosit pentru identificarea sistemului prezentat anterior. Ieșirea blocului creat se compară cu datele citite de pe motor, cu o perioadă de eșantionare de 100 [ms]. Rezultatele se pot vedea acesând un bloc de tip scope.

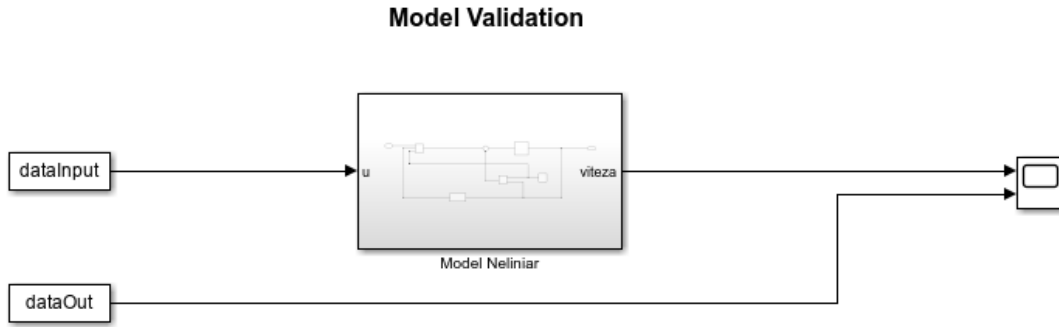


Figura 3.8: Simulink pentru validarea modelului (funcția de transfer)

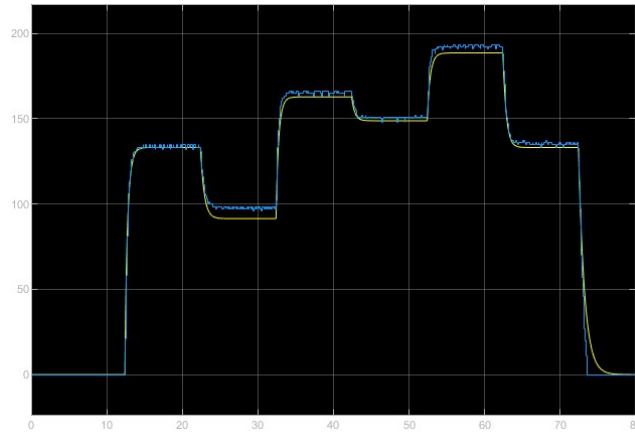


Figura 3.9: Rezultatul obținut pentru funcția de transfer identificată

Se poate observa că acest sistem identificat (reprezentat cu galben) reușește să aproximeze destul de bine sistemul real (reprezentat cu albastru), vitezele medii fiind cel mai bine aproximate, iar cele extreme (mari sau mici) sunt aproximate mai greu. Cu toate acestea, se poate considera o suprapunere destul de bună pentru continuarea procesului de control. În cazul procesului modelat cu ajutorul ecuațiilor diferențiale, se obține următoarea descriere a sistemului:

$$\dot{x} = A \cdot x + B \cdot u = \begin{pmatrix} -48580.9682 & 13961468.520 \\ -0.006207 & -0.5 \end{pmatrix} \cdot x + \begin{pmatrix} -104894579.418 & 0 \\ 0 & 10 \end{pmatrix} \cdot u, \quad (3.73)$$

unde stările sistemului și intrările acestuia sunt:

$$x = \begin{pmatrix} \omega \\ I_a \end{pmatrix}, \quad u = \begin{pmatrix} V \cdot K(\omega) \\ T_L \end{pmatrix}. \quad (3.74)$$

În acest context, V este factorul de umplere ce se trimite către drivere și T_L este tracțiunea care trebuie învinsă. Pentru afișarea datelor, singura tracțiune (torque) considerată este cea a motorului astfel pentru suprapunere $T_L = 0$. În acest caz, factorul de proporționalitate se calculează astfel: $K(\omega) = 9.2289 \cdot 10^{-5} \omega^2 + 0.0371 \omega + 4.6870$.

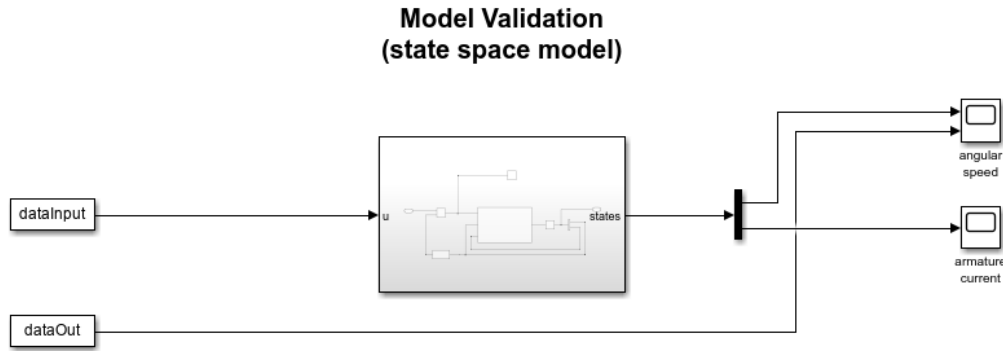


Figura 3.10: Simulink pentru validarea modelului (spațiul stărilor)

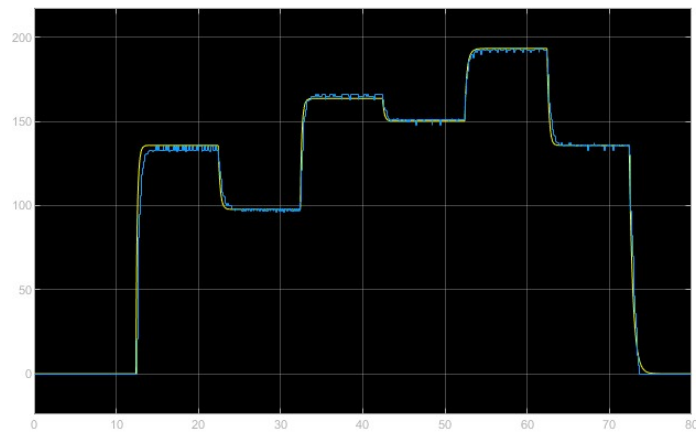


Figura 3.11: Suprapunerea vitezei (spațiul stărilor)

Din ce se poate observa, dinamica vitezei simulate (figurată cu galben) este aproape identică cu cea a vitezei reale (figurată cu albastru). Din suprapunerea rezultată sistemul acesta este mai performant decât cel bazat pe funcția de transfer, însă din cauza faptului că dinamica curentului nu este măsurată fizic, nu putem deduce faptul că acest sistem este exact ca cel adevărat. Însă din punct de vedere al vitezei, se poate continua cu implementarea controlului acestui sistem.

3.3.2. Simularea efectului de liniarizare

Prin faptul că suprapunerile obținute au fost un succes, putem să creăm un bloc Simulink care să preia semnalul de comandă și să îl înmulțească cu inversul funcției ce descrie gainul liniar. Poza de mai jos reprezintă blocul de liniarizare al sistemului, intrarea fiind o treaptă unitară din momentul la care se începe simularea până la secunda 5 unde se atribuie o treaptă de amplitudine 2. Prin efectul de liniarizare, forma sistemului trebuie să fie descrisă sub forma:

$$H(s) = \frac{1}{0.8s + 1}. \quad (3.75)$$

Asta înseamnă că eroarea staționară trebuie să fie nulă deoarece nu apare nicio amplificarea în funcția de transfer. Pentru verificarea se afișează pe scope semnalul de comandă și răspunsul obținut în urma aplicării acestuia.

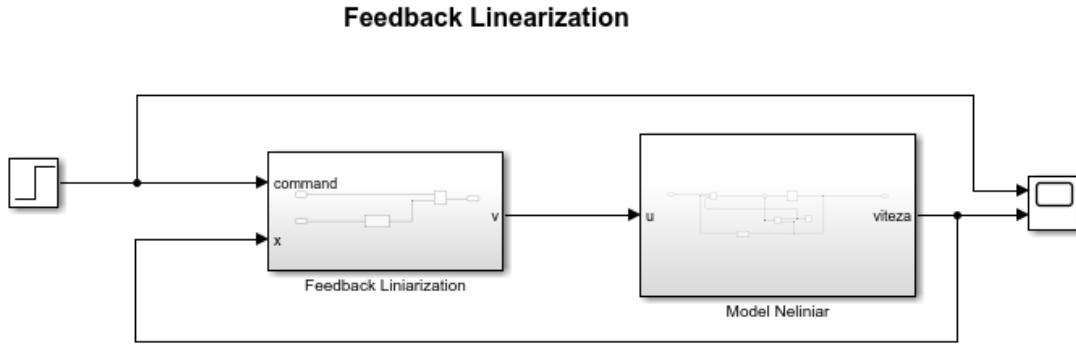


Figura 3.12: Sistemul liniarizat prin metoda Feedback Linearization

Astfel se realizează liniarizarea sistemului, iar aceasta se poate observa în figura de mai jos, urmărind rezultatele afișate de blocul de tip scope.

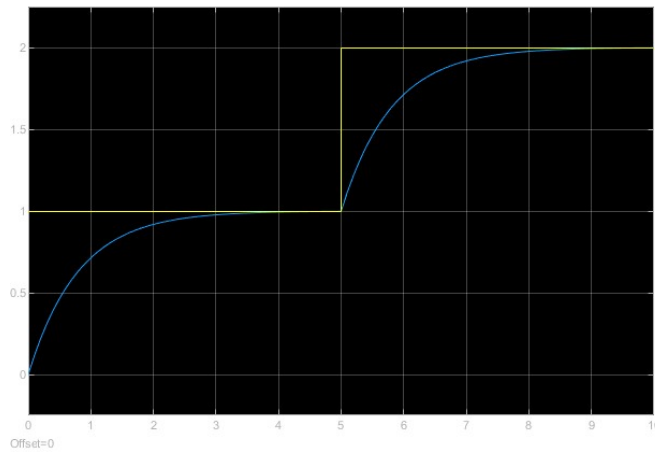


Figura 3.13: Evidențierea liniarizării prin metoda Feedback Linearization

Se observă faptul că eroarea staționară este zero și timpul de răspuns este în jur de 3.2 [s]. Mai mult de atât, evoluția comenzii este urmărită liniar de către sistem, fapt ce ne indică cu certitudine că sistemul este liniarizat. Pentru modelul obținut din ecuațiile diferențiale se procedează exact la fel, și se poate trage exact aceeași concluzie. În ambele cazuri, sistemul se bucură de o funcție ce descrie evoluția factorului de proporționalitate care este inversabilă pe tot domeniul de acționare. În cazul în care aceasta nu beneficia de această caracteristică, funcția trebuia împărțită în mai multe funcții care să fie inversabile pe intervale, sau să se identifice altă funcție.

3.3.3. Simularea regulatorilor obținute

Pentru simularea regulatorilor, se folosesc valorile obținute:

$$\begin{aligned} K_{sat} &= 1.8284, & K_{sat}^f &= 2.8281, \\ K_{agr} &= 7.4261, & K_{agr}^f &= 8.4246. \end{aligned}$$

Structura care se folosește pentru implementarea regulatorilor este prezentată mai jos:

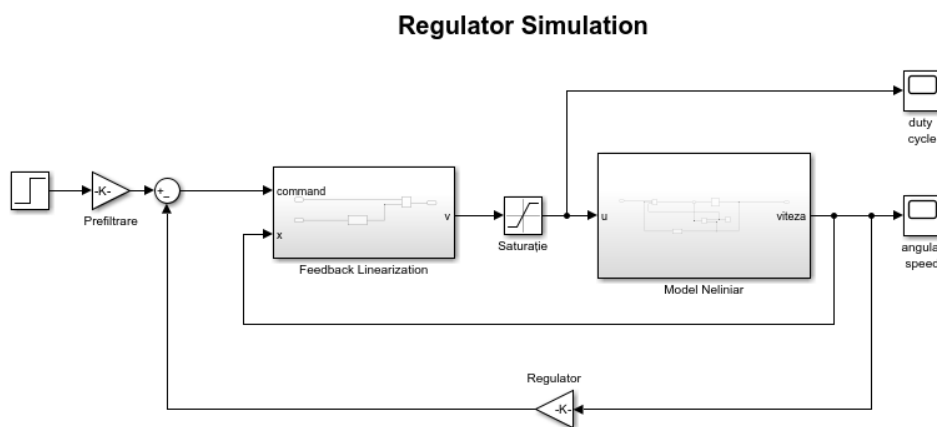


Figura 3.14: Structura de reglare

Pentru analizarea rezultatelor, se consideră viteza de referință de 160 [rps] dată la secunda 1 a simulării și o saturație pe semnalul care trebuie să ajungă la drivere, în valoare de 0.2 (20%). De asemenea, se analizează semnalul de comandă, pentru a observa diferențele între un regulator agresiv și unul care respectă saturațiile impuse.

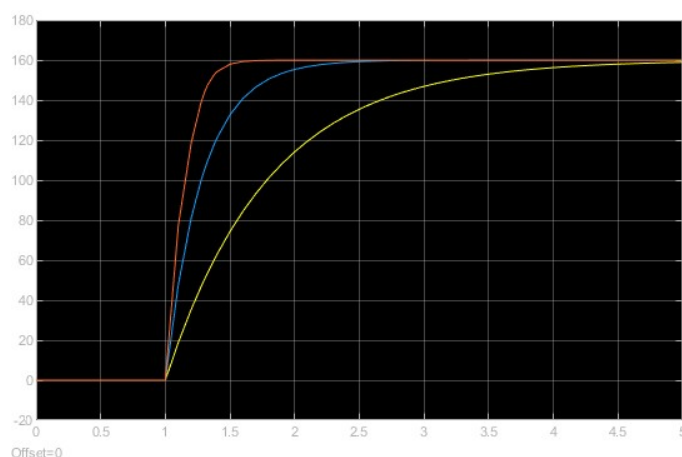


Figura 3.15: Rezultatul reguletoarelor

Se observă faptul că prin folosirea reguletoarelor, timpul de răspuns este mult mai mic comparativ cu dinamica sistemului fără regulator (reprezentat cu galben), iar eroarea staționară este nulă. De asemenea, regulatorul agresiv se observă că obține cel mai bun timp de răspuns (reprezentat cu roșu), iar regulatorul care respectă saturația (reprezentat cu albastru) ajunge puțin mai târziu decât cel agresiv. În vederea conceperii unui regulator, nu este de ajuns doar asigurarea performanțelor, astfel trebuie analizat și efortul de comandă pentru determinarea fezabilității acestuia.

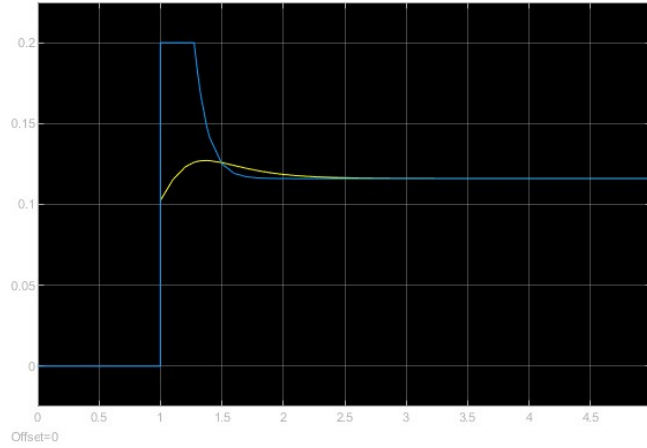


Figura 3.16: Comanda reguletoarelor

În figura de mai sus sunt reprezentate semnalele de comandă ale celor două reguletoare. Se observă faptul că pentru regulatorul agresiv (figurat cu albastru), comanda este blocată din cauza saturației, iar pentru regulatorul care ține cont de saturație (figurat cu galben), comanda evoluează frumos, fără să forțeze acționarea bruscă a motorului.

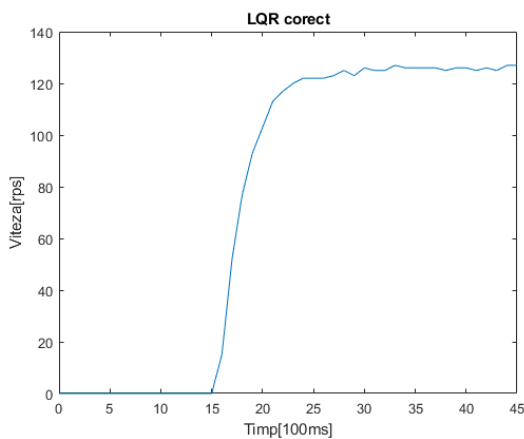
3.3.4. Testarea fizică a reguletoarelor

Pentru testarea fizică, se realizează montajul sistemului, iar legea de reglare scrisă pe microcontroller folosește pe rând parametrii discretizați, calculați anterior:

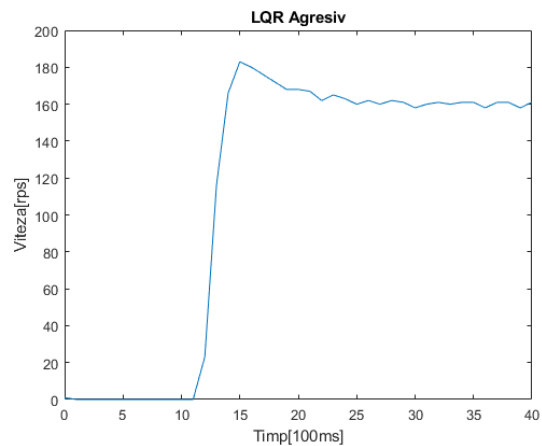
$$K_{sat} = 1.5442, \quad K_{sat}^f = 2.5439, \quad (3.76)$$

$$K_{agr} = 4.7003, \quad K_{agr}^f = 5.7013. \quad (3.77)$$

La fiecare 100 [ms] se transmite serial viteza măsurată în [rps], iar calculatorul înregistrează aceste valori folosind softwareul PuttY, care ascultă portul USB conectat la plăcuța Arduino. Astfel în imaginile de mai jos se observă rezultatele obținute pentru cele două tipuri de reguletoare.



(a) Regulatorul care respectă saturația



(b) Regulatorul agresiv

Figura 3.17: Rezultatul implementării fizice a reguletoarelor

La prima vedere, se observă ca pentru cazul în care se folosește regulatorul care respectă saturația, sistemul răspunde în aproximativ jumătate de secundă. În cazul în care se folosește regulatorul agresiv, se observă ca acesta introduce un suprareglaj de 11% și timpul său de răspuns este de aproximativ o secundă. Rezultatele fizice sunt foarte diferite de cele simulate, însă această diferență provine din faptul că sistemul fizic este unul discret, iar pentru acest sistem, regulatorul agresiv menține comanda saturată prea multe eșantioane de timp. Astfel, în cazul folosirii unui regulator agresiv, trebuie ca perioada de eșantionare să fie cât mai mică pentru evitarea suprareglajelor mari, însă efectul acesta o să persiste. Totodată putem concluziona că deși în cadrul simulărilor regulatorul care respectă saturația avea performanțe puțin mai slabe decât cel agresiv, în realitate acesta îmbunătățește performanțele sistemului, spre deosebire de celălalt, care înrăutățește răspunsul într-o manieră sesizabilă.

3.3.5. Testarea sistemului întreg

În cadrul acestei părți se testează funcționarea întreagă a sistemului ce cuprinde partea de control și partea de direcție, și se interpretează rezultatele.

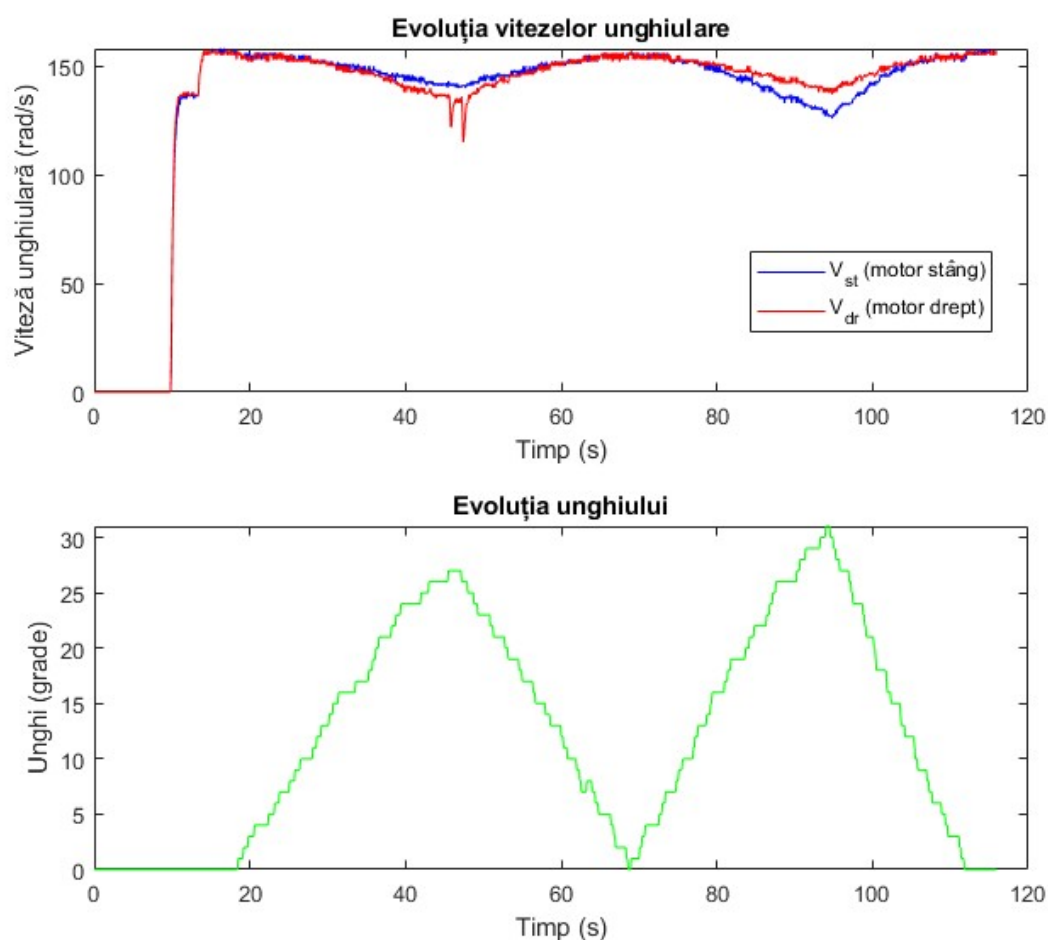


Figura 3.18: Viteza unghiulară și unghiul direcției

Din figura de mai sus se pot observa trei regiuni. Prima regiune este cea dată

de setarea unei viteze de 140 [rps] și unghiul deplasării este de 0 grade. Aici se vede că cele două motoare se învârt cu aceeași viteză. Pentru urătoarea treaptă de viteză (160 [rps]), la fel, acestea se învârt cu aceeași viteză. A doua regiune este reprezentată de creșterea unghiului, deplasarea făcându-se spre dreapta. Pe această porțiune se observă faptul că viteza motorului drept scade mai puternic decât cea a motorului stâng, deoarece motorul interior curbei este cel drept, iar cel stâng este cel exterior. A treia regiune este reprezentată de creșterea unghiului, în direcția opusă. Astfel, de data asta motorul stâng este cel care este interior, iar cel drept exterior.

De asemenea se poate observa faptul că deși motoarele și drive-urile sunt de același tip, comportamentul acestora este diferit. Pentru atingerea aceleiași viteze se respectă factorul de proporționalitate neliniar calculat astfel:

$$K_{dr}(\omega) = 0.06328\omega^2 - 29.11\omega + 5200. \quad (3.78)$$

Mai mult de atât, motorul drept nu se poate acționa astfel încât să se rotească mai încet de 140 [rps], fapt ce se revede din vârfurile ce apar în figură.

3.3.6. Simularea controlului avansat

Partea aceasta reprezintă o direcție de progresare a proiectului și pe baza principiilor prezentate anterior, se poate implementa fizic, desigur pe un microncontroller mai puternic și cu niste senzori mai calitativi.

Se continuă de la modelul în spațiul stărilor obținut prin folosirea ecuațiilor diferențiale și de la liniarizarea sistemului prin adăugarea blocului care anulează efectul factorului de amplificare neliniar. Astfel, prin respectarea condițiilor impuse ($\alpha = 10$, $\theta = 0.1$, $u_{max} = 1$ și matricea Q) se calculează un regulator prin impunere de poli. Din cauza faptului că sistemul nostru nu beneficiază de un senzor de curent, nu putem folosi regulatorul, decât dacă se implementează de asemenea un estimator de stare. Așadar structura sistemului controlat este cea din figura alăturată.

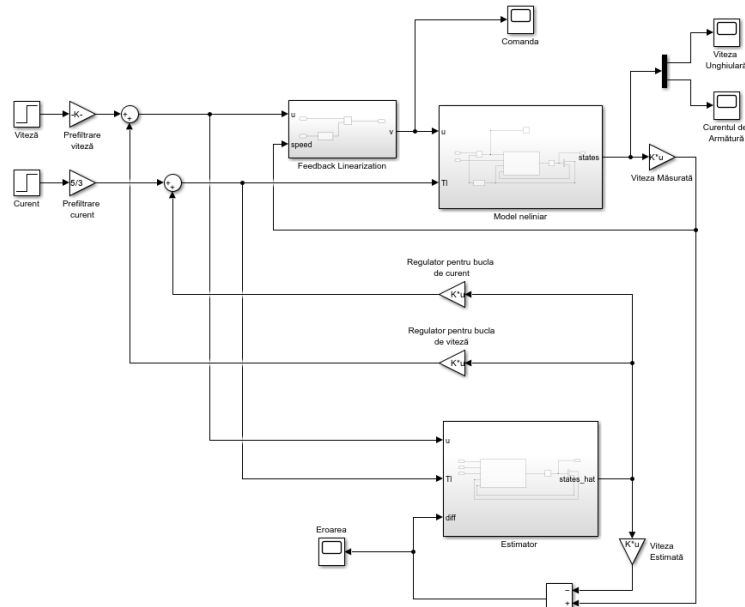


Figura 3.19: Structura avansată de control

După cum se poate vedea, estimatorul aproximează stările în funcție de intrările sistemului și pe baza diferenței de ieșiri (vitezelor). Aceste stări sunt luate și se aplică reglarea sistemului fizic, în două bucle, una de viteză și una de tracțiune. Rezultatele care se pot vedea sunt stările sistemului, viteza unghiulară a motorului și curentul de armătură, eroarea de estimare și comanda. Simularea se face cu setarea unui semnal de tip treaptă ca referință de viteză cu valoarea de 160 [rps], iar un alt semnal care este de același tip se folosește pentru cealaltă intrare cu valoarea de 0.05, dată la secunda 5 a simulării. În imaginile de mai jos se află comportamentul obținut în urma aplicării datelor enunțate anterior. Acestea sunt grupate în două, astfel primul set de imagini reprezintă performanțele modului de acționare (semnalul de comandă și semnalul de eroare de estimare), pe baza cărora se clădește controlul. Aici este de urmărit dacă eroarea converge la valoarea zero și dacă se respectă saturația comenzii de 100%.

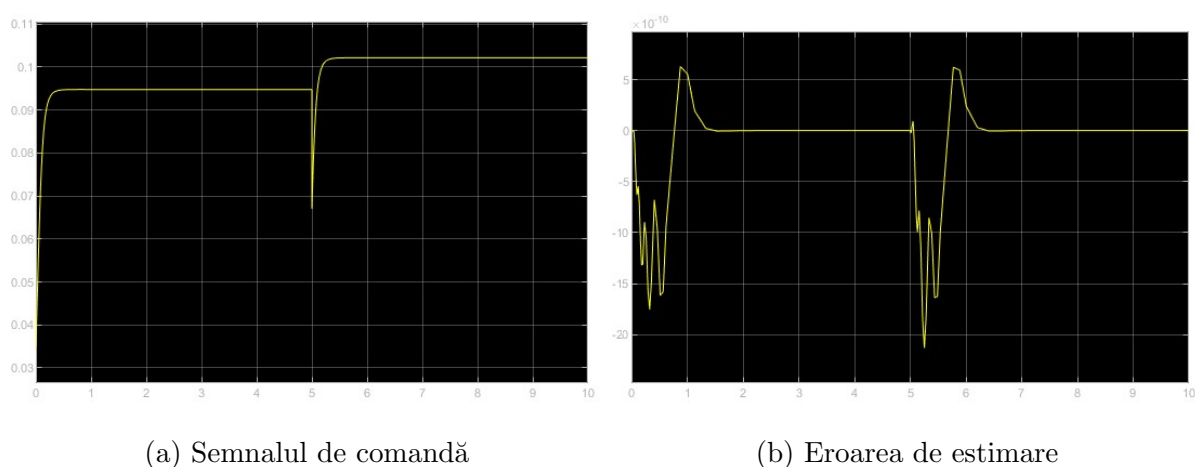


Figura 3.20: Performanțele părții de acționare

Din prima imagine a acestui grup, putem trage concluzia că saturația este respectată, iar semnalul de eroare converge la valoarea zero, într-un timp scurt și nu prezintă vârfuri mari. Acest lucru semnalează faptul că sistemul controlat este unul fezabil și se poate implementa.

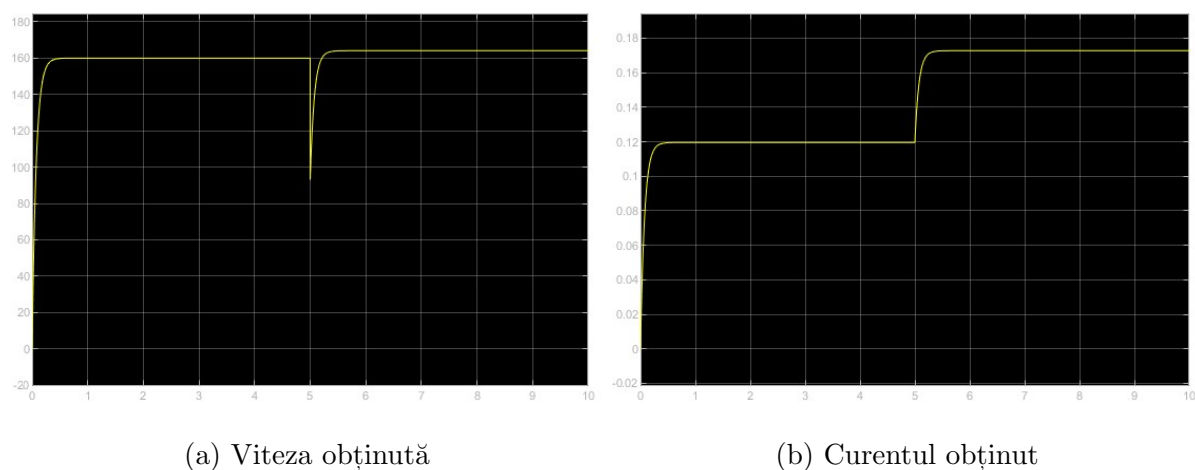


Figura 3.21: Performanțele răspunsului sistemului

În acest set de imagini, se poate vedea performanța mult îmbunătățită a sistemului și cum se comportă pentru încărcături (load) diferite. Astfel, pentru o schimbare de tracțiune, viteza scade iar curentul crește. După stabilizare, curentul are o valoare mai mare decât cea inițială, fapt ce era de așteptat, însă viteza sistemului este puțin mai mare decât cea văzută înainte de adăugarea tracțiunii suplimentare. Pentru înlăturarea acestui efect se poate apela la elemente de tip feedforward sau se poate analiza în detaliu acest efect în lucrări care au la bază aceași idee.

3.4. Instalare și utilizare

Pentru realizarea montajului fizic se urmează pașii în ordinea de mai jos:

1. se montează motoarele de placa PVC pentru a nu reprezenta un hazard pe timpul acționării motoarelor
2. se așează discurile de encoder pe axul fiecărui motor
3. se montează lângă fiecare motor o bucată de plastic care joacă rol de suport pentru senzorul optic.
4. se plasează senzorul optic astfel încât discul de pe motor să intre în deschizăturile senzorului.
5. se cuplează fiecare inverter la motoare prin conectorii de tip banană.
6. se realizează o gaură și se pune un surub pe care se montează un disc cu fante late.
7. se fixează discul cu ajutorul mai multor a patru piulițe
8. senzorii se plasează pe un suport de plastic, cap în cap, astfel încât discul cu fante late să fie plasat în deschizăturile senzorilor optici
9. se plasează convenabil plăcuța de dezvoltare Arduino
10. se folosește un bread-board pentru multiplicarea ieșirii de 5V și cea de GND
11. se leagă alimentarea senzorilor de bread-board astfel încât senzorii să funcționeze corect
12. senzorii care măsoară viteza motoarelor se leagă la pinul digital 7 (pentru motorul stâng) și pinul digital 4 (pentru motorul drept)
13. senzorii din cadrul encoderului de cuadratură se leagă la pinul digital 2 (pentru cel din dreapta) și la pinul digital 3 (pentru cel din stânga)
14. potențiometrul se alimentează cu 5V și cursorul se leagă pe canalul analogic 0
15. pe firul portocaliu al invertoarelor se leagă semnalul de comandă, pentru motorul stâng se folosește pinul digital 9 iar pentru motorul drept se folosește pinul digital 10.
16. firul maro al invertoarelor reprezintă firul de ground și acesta se conectează la groundul de pe bread-board (se face ground comun între toate elementele sistemului)
17. se conectează sursa de alimentare ABSIMA la un splitter, iar splitterul la cele două invertore prin conectorul în T.

Pentru transmiterea datelor fizice pe calculator se urmează pașii:

1. se realizează montajul fizic
2. se leagă plăcuța Arduino la calculator prin intermediul cablului de programare
3. se programează plăcuța cu codul C dezvoltat.
4. se deschide o comunicare serială prin programul PuTTY și se înregistrează datele
5. se salvează datele în format txt și se pot importa în Matlab pentru a fi plotate.

Pentru simularea sistemului se urmează instrucțiunile de mai jos:

1. se descarcă scriptul Matlab/Simulink
2. se deschide modelul Simulink
3. pentru vizualizarea identificării se încarcă workspaceul descărcat
4. se rulează modelul Simulink pe o perioadă de 80[s].
5. pentru vizualizarea celorlalte rezultate, se urmează descrierea sistemului făcută la capitolul Testare și Validare.

Observații:

1. invertoarele asigură și așa numitul circuit BEC (Battery eliminator circuit) cu scopul alimentării directe a plăcuței de dezvoltare.
2. se poate folosi circuitul BEC, însă doar de la un singur inverter, NU de la ambele deodată.
3. a nu se conecta firul roșu (firul cald) al inverterului la 5V în cazul alimentării plăcuței din alte surse de tensiune sau de la calculator.
4. a nu se lăsa splitterul neconectat la capetele dublate, deoarece se poate produce scurtcircuit prin lovirea capetelor.
5. se recomandă folosirea acumulatorului în intervalele de tensiune adecvate unei celule 3s Li-Po

Capitolul 4. Concluzii

4.1. Rezultate obținute

În urma acestui proiect au fost îndeplinite obiectivele propuse inițial:

- Procurarea tuturor elementelor care constituie sistemul dorit (senzori, motoare, etc) și asigurarea bunei funcționări ale acestora.
- Măsurarea și interpretarea datelor, precum viteza unghiulară a motoarelor, cu ajutorul senzorilor, cât și transmiterea informațiilor obținute prin cadrul comunicației seriale între plăcuța de dezvoltare și calculator.
- Studiarea metodelor de control în cazul respectării/nerespectării unei saturații impuse și a metodelor de înlocuire a diferențialului clasic prin calcularea vitezelor de referință a motoarelor pe baza geometriei unei mașini.
- Obținerea unui model matematic sub formă de funcție de transfer cu intrarea factor de umplere și ieșirea viteză unghiulară, respectiv a unui model extins în spațiul stărilor, care are ca intrare tensiunea de alimentare a motoarelor și sarcina atribuită sistemului, ca ieșire viteză unghiulară a motoarelor, iar ca stări viteză de rotație și curentul de armătură a motoarelor folosite.
- Realizarea unei machete care să conțină elementele folosite, care asigură funcționarea sigură a sistemului și totodată care permite observarea fiecărui element în parte și a conexiunilor dintre ele.
- Testarea mai multor metode de control, precum plasarea de poli folosind metoda regulatorului quadratic liniar (Linear Quadratic Regulator – LQR), cât și prin impunerea unor inegalități matriciale (Linear Matrix Inequation – LMI) cu scopul optimizării plasării polilor, respectând condițiile impuse, stările sistemului folosite pentru control fiind obținute prin estimarea acestora.
- Implementarea unui cod ce folosește resursele unui μC pentru realizarea funcționalităților necesare (citire senzori, generare de semnale, corecție semnale, etc.) și care reușește acționarea motoarelor diferențial în funcție de viteza impusă printr-un potențiomtru și respectiv în funcție de unghiul deplasării și direcția dată de encoderul de cuadratură.
- Asigurarea comunicării între elementul de comandă reprezentat de plăcuța de dezvoltare Ardunio și invertoarele folosite cu scopul acționării motoarelor prin respectarea standardului acceptat de acest tip de drivere.
- Soluționarea prin metode software a zgomotelor de citire a potențiometrului și a problemelor date de folosirea unor registrii cu lungime de zece biți a convertorilor analogici digitali pentru generarea semnalelor de tip PWM.
- Testarea și validarea metodelor de control și a funcționalității a diferențialului software, atât a sistemului fizic realizat, cât și a sistemului extins prin adăugarea regulatorului ce plasează polii corespunzător condițiilor impuse și prin implementarea estimatorului de stare.

Obiectivele legate de partea hardware s-au îndeplinit prin construirea machetei și prin

testarea elementelor fizice pentru a putea fi montate și folosite în cadrul proiectului. Totodată, folosind senzorii optici s-au realizat cu succes cele două tipuri de encodere, de viteză care reușește să citească viteza unghiulară și respectiv de cuadratură care reușește să interpreteze direcția și unghiul traiectoriei curbilinii simulate. De asemenea, interfațarea utilizatorului s-a realizat cu succes prin folosirea potențiometrului pentru impunerea unei viteze și prin folosirea encoderului de cuadratură enunțat anterior. Datele fizice citite de pe plăcuța de dezvoltare au fost transmise serial către un calculator, unde s-au dezvoltat metode de control și soluții pentru problemele apărute. Astfel, aceste obiective sunt subliniate de funcția de transfer obținută în cadrul identificării unui motor BLDC și liniarizată prin intermediul reacției. Aceasta reușește să fie controlată print-un regulator LQR, diferențele între regulatoarele obținute putând fi vizualizate în cadrul testării fizice a sistemului. Se remarcă faptul că nerespectarea saturației poate să ducă la apariția unor dinamici nedorite, precum suprareglajul observat anterior. De asemenea, folosind modelul geometric Ackermann, se reușește calcularea vitezelor de referință pentru fiecare motor. Sistemul fizic implementat reușește să asigure funcționarea corectă a logici abordată în acest proiect pentru anumite viteze și unghiuri de deplasare, care pot fi folosite și într-o situație concretă. Totodată, obținerea unui model matematic în spațiul stărilor și simularea controlului acestuia prin calcularea unui regulator ce impune polii într-un mod optimal în raport cu inecuațiile matriciale atribuite sunt prezente în această lucrare. În cadrul controlului avansat al sistemului extins, s-a prezentat și estimatorul care reușește să aducă o soluție pentru problema obținerii stărilor fără a folosi un senzor pentru a măsura valoarea fizică. Mai mult de atât, implementarea funcționalităților care trebuie îndeplinite de μC și soluționarea limitărilor fizice aduse de convertorul digital-analogic, analogic-digital și respectiv ale encoderilor de viteză au fost realizate astfel încât sistemul fizic să funcționeze corect pentru anumite intrări date de utilizator.

4.2. Direcții de dezvoltare

Când vine vorba de direcțiile de dezvoltare a acestui proiect putem pune în prim plan două idei. Prima direcție este orientată pe dezvoltarea hardware a sistemului. Limitările aduse în desfășurarea acestui sistem sunt evidente. Nevoia de regiștri mai mari pentru generarea semnalelor PWM și pentru conversia realizată de modulul convertorului analogic-digital este una crucială în controlul precis a celor două motoare. De asemenea, viteza de procesare și folosirea unui microcontroller/procesor cu mai multe coreuri care asigură funcționarea taskurilor real-time, este cu adevărat o soluție de dezvoltare. Mai mult de atât, asigurarea unor elemente de execuție (motoare, invertoare) și detecție (encoderi de viteză și de cuadratură) mai calitative este indicată.

A doua direcție de progresare tehnologică este reprezentată de algoritmul de control avansat. Prin continuarea acestuia, se pot controla mai multe procese, nu doar viteza motoarelor. Un avantaj ar fi modelarea individuală a fiecărui motor și de asemenea considerarea dinamicii invertoarelor folosite în comandarea acestora. Totodată se poate adăuga un sistem de ABS, care să controleze alunecările roților pe condiții nefavorabile. Pentru partea de programare a microcontrollerului, se pot folosi structuri de tip real-time, pentru asigurarea comunicării semnalelor sistemului într-o perioadă exactă de timp și gestionarea erorilor ce pot apărea. De asemenea, se pot implementa filtre software mai avansate pentru filtrarea semnalelor citite de la senzorii de viteză și de la encoderul de cuadratură, iar estimatorul se poate înlocui cu un algoritm ce antrenează un model de inteligență artificială care să poată face predicții veritabile pe baza măsurătorilor făcute.

Bibliografie

- [1] G. Belingardi and A. Scattina, “Battery pack and underbody: Integration in the structure design for battery electric vehicles—challenges and solutions,” *Vehicles*, vol. 5, no. 2, pp. 498–514, 2023. [Online]. Available: <https://www.mdpi.com/2624-8921/5/2/28>
- [2] International Energy Agency, “Global ev outlook 2024,” Paris, 2024, licence: CC BY 4.0. [Online]. Available: <https://www.iea.org/reports/global-ev-outlook-2024>
- [3] I. European Climate and E. E. Agency. (2023, Apr.) Life 2023 calls for proposals: €611 million available to bring green ideas to life. European Commission. [Online]. Available: https://cinea.ec.europa.eu/news-events/news/life-2023-calls-proposals-eu611-million-available-bring-green-ideas-life-2023-04-18_en
- [4] U. Celtekligil and M. Çavdar, “Modeling, simulation and analysis of bipolar driven bldc motor,” 09 2021, pp. 334–339.
- [5] C. Xiang, X. Wang, Y. Ma, and B. Xu, “Practical modeling and comprehensive system identification of a bldc motor,” *Mathematical Problems in Engineering*, vol. 2015, no. 1, p. 879581, 2015. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1155/2015/879581>
- [6] P. Dobra, “Robust pid controller of bldc motor via state feedback design.”
- [7] B. A. Md Zain, F. Anuar, and I. Latif, “Modeling and speed control for sensorless dc motor bldc based on real time experiment,” 12 2019.
- [8] I. Anshory, I. Robandi, and Wirawan, “Identification parameter system for mathematical modeling bldc motor using transfer function models,” *IOP Conference Series: Materials Science and Engineering*, vol. 821, p. 012023, 05 2020.
- [9] F. Celani and A. Isidori, “Feedback linearization,” *Scholarpedia*, vol. 4, p. 6517, 01 2009.
- [10] Y.-L. Kuo and P. Pongpanyaporn, “Continuous-time nonlinear model predictive tracking control with input constraints using feedback linearization,” *Applied Sciences*, vol. 12, p. 5016, 05 2022.
- [11] D. Beyene and D. Shiferaw, “Feedback linearization based adaptive stabilizing controller design coupled with fuzzy logic swing-up for pendulum on cart,” vol. 6, pp. 124–142, 03 2024.
- [12] J. Zou and X. Zuo, “Active suspension lqr control based on modified differential evolutionary algorithm optimization,” *Journal of Vibroengineering*, 05 2024.

- [13] H. Guo, J. Wu, Z. Yin, and W. Zhang, “Comparative study of furuta pendulum based on lqr and pid control,” *Journal of Physics: Conference Series*, vol. 2562, p. 012075, 08 2023.
- [14] —, “Comparative study of furuta pendulum based on lqr and pid control,” *Journal of Physics: Conference Series*, vol. 2562, p. 012075, 08 2023.