

**UNIVERSITATEA TEHNICĂ “GHEORGHE ASACHI” DIN IAȘI**

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**

***Domeniul: Ingineria Sistemelor***

***Specializarea: Automatică și Informatică Aplicată***

**PROIECT DE DIPLOMĂ**

***Implementarea pe DSP a sistemului de control  
al unei mașini electrice BLDC cu interpolarea  
poziției unghiulare prin filtrare Kalman***

*Coordonator științific,*

***Șef lucr. dr. ing. Mocanu Răzvan***

*Absolvent,*

***Antici Paul-Leonard***

**Iași**

**2024**

## DECLARAȚIE DE ASUMARE A AUTENTICITĂȚII PROIECTULUI DE DIPLOMĂ

Subsemnatul Antici Paul-Leonard, legitimat cu CI seria NZ nr. 254459, CNP 5010629270830 autorul lucrării: "Implementarea pe DSP a sistemului de control al unei mașini electrice BLDC cu interpolarea poziției unghiulare prin filtrare Kalman" elaborată în vederea susținerii examenului de finalizare a studiilor de licență, programul de studii AIA organizat de către Facultatea de Automatică și Calculatoare din cadrul Universității Tehnice „Gheorghe Asachi” din Iași, sesiunea iulie a anului universitar 2023-2024, luând în considerare conținutul Art. 34 din Codul de etică universitară al Universității Tehnice „Gheorghe Asachi” din Iași (Manualul Procedurilor, UTI.POM.02 - Funcționarea Comisiei de etică universitară), declar pe proprie răspundere, că această lucrare este rezultatul propriei activități intelectuale, nu conține porțiuni plagiate, iar sursele bibliografice au fost folosite cu respectarea legislației române (legea 8/1996) și a convențiilor internaționale privind drepturile de autor.

Data

30.06.2024

Semnătura

Antici

# Cuprins

<b>INTRODUCERE.....</b>	<b>4</b>
<b>CAPITOLUL 1: ECHIPAMENTE UTILIZATE .....</b>	<b>5</b>
1.1 Motorul BLDC – motorul Nanotec DF45L024048-A2.....	5
1.2 Placa de dezvoltare LAUNCHXL - F28379D .....	15
1.3 Invertorul BOOSTXL-DRV8305EVM.....	17
1.4 Montajul realizat.....	20
<b>CAPITOLUL 2: PROBLEME DE CONTROL ALE MODELELOR NELINIARE.....</b>	<b>22</b>
2.1 Sistemele neliniare.....	22
2.2 Sistemele de control cu feedback .....	24
<b>CAPITOLUL 3: STRATEGIILE DE CONTROL.....</b>	<b>26</b>
3.1 Controlerul proporțional (P).....	26
3.2 Controlerul integrator (I).....	27
3.3 Controlerul proportional integrator (PI).....	28
<b>CAPITOLUL 4: PROIECTAREA ÎN MATLAB/SIMULINK.....</b>	<b>33</b>
4.1 Modulul eCAP (Enhanced Capture).....	33
4.2 Determinarea poziției cu ajutorul senzorilor Hall.....	35
4.3 Determinarea vitezei.....	36
4.4 Subsistemul transmisiei seriale.....	37
4.5 Subsistemul recepției seriale.....	38
4.6 Schema Simulink PC_host_SCI_interface_PI.....	39
4.6.1 Modul de transmisie.....	40

4.6.2 Modul de recepție.....	40
4.7 Prelucrarea comenzii controllerelor .....	41
4.8 Filtrul Kalman.....	46
4.9 Regulatorul PI.....	51
<b>CAPITOLUL 5: REZULTATE EXPERIMENTALE.....</b>	<b>52</b>
5.1 Rezultate control PI – controller PI.....	53
5.2 Rezultatele filtrului Kalman extins.....	54
<b>6. CONCLUZII.....</b>	<b>56</b>
<b>7. BIBLIOGRAFIE.....</b>	<b>57</b>
<b>8. ANEXE.....</b>	<b>58</b>

# INTRODUCERE

Introducerea pe scară largă a automatizării și robotizării, precum și dezvoltarea de noi tipuri de mașini controlate electronic a creat nevoia de a utiliza cele mai rapide și mai fiabile sisteme de control.

Pentru că sunt capabile să mențină o viteză constantă și să aibă o eficiență energetică ridicată, motoarele sincron de curent continuu (DC) sunt foarte populare. Controlul precis al acestor motoare necesită soluții avansate de control care să poată gestiona complexitatea operațională și să optimizeze performanța în timp real.

Cu toate acestea, din cauza complexității modelului matematic implicat, aceste motoare prezintă multiple probleme de control. În acest context, utilizarea procesoarelor digitale de semnal (DSP) este o modalitate promițătoare de a îmbunătăți controlul și performanța acestor motoare.

Implementarea unui sistem de control pe un procesor de semnal digital (DSP) oferă multiple avantaje în acest context. DSP-urile sunt proiectate pentru a efectua calcule complexe rapid și eficient, ceea ce le face ideale pentru aplicații de control al motoarelor.

Acest proiect își propune să dezvolte și să implementeze un sistem de control pentru o mașină electrică sincronă de curent continuu, utilizând un DSP.

Motoarele de curent continuu fără perii (BDLC) reprezintă unul dintre cele mai populare tipuri de motoare, fiind utilizat pe scară largă într-o varietate de aplicații în automatizarea industrială și electrocasnice datorită eficienței lor superioare.

Cum pot fi îmbunătățite performanțele dinamice ale motorului, eficiența energetică și stabilitatea sistemului vom detalia etapele de implementare hardware și software ale sistemului de control propus.

# CAPITOLUL 1

## ECHIPAMENTE UTILIZATE

În capitolul 1 al acestei lucrari vor fi prezentate detalii despre motorul cu magneti permanenți fără perii în legătură cu structura, configurarea, acționarea, cât și o scurtă comparație a avantajelor și dezavantajelor pe care le are față de motorul cu magneti permanenți cu perii. În plus, în acest capitol vor fi descrise și celelalte echipamente utilizate pentru a realiza procesul și pentru a obține controlul dorit: placa de dezvoltare LAUNCHXL - F28379D și invertorul BOOSTXL-DRV8305EVM.

### 1.1 Motorul BLDC Nanotec DF45L024048-A2

Majoritatea echipamentelor electrice au adesea în componența lor cel puțin un motor folosit pentru a roti sau deplasa un obiect din poziția sa inițială. Motoarele convertesc energia electrică în energie mecanică folosind principii electromagnetice. Pe piață există o gamă largă de tipuri de motoare, inclusiv motoare de inducție, servomotoare și motoare DC, atât cu perii, cât și fără perii. Metoda de conversie a energiei este, în esență, aceeași pentru toate motoarele electrice.

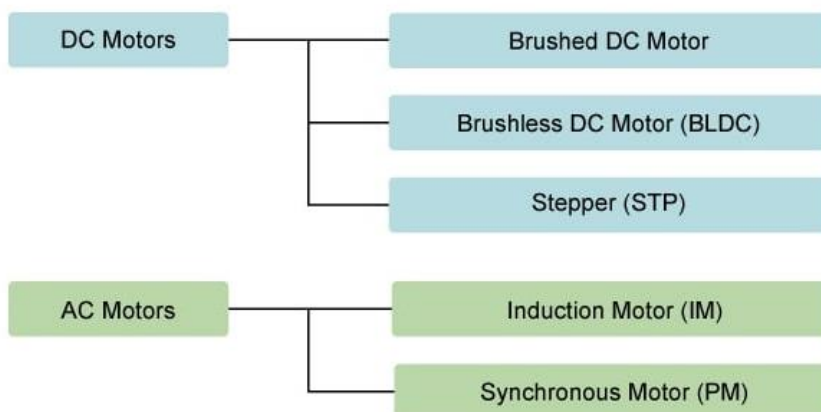


Fig.1.1.1 Clasificarea motoarelor electrice

Diferența principală dintre motoarele AC și DC este dată de tipul de putere aplicat armăturii. Din acest punct de vedere, un motor BLDC este de fapt un motor AC. Diferența dintre un motor asincron și unul sincron este dacă rotorul funcționează sau nu la aceeași frecvență cu câmpul magnetic al statorului.

Cel mai simplu tip de motor este motorul cu perii de curent continuu. În acest tip de motoare, curentul electric trece prin bobinele care sunt dispuse într-un câmp magnetic fix. Curentul generează câmpuri magnetice în bobine, iar acest lucru determină rotirea ansamblului de bobine, deoarece fiecare bobina este respinsă de polul de același semn și atrasă de polul de semn opus al câmpului fix.

În funcție de cerințele aplicației, se poate alege un anumit tip de motor. Cu toate acestea, în prezent, tot mai multe proiecte noi se bazează pe motoarele Brushless DC, cunoscute sub numele de motoare BLDC.



Fig. 1.1.2 Motorul BLDC Nanotec DF45L024048-A2

Un motor BLDC realizează comutația electronic folosind feedback-ul poziției rotorului pentru a determina momentul când trebuie să comute curentul. Feedback-ul implică de obicei un set de senzori Hall atașat sau un encoder rotativ.

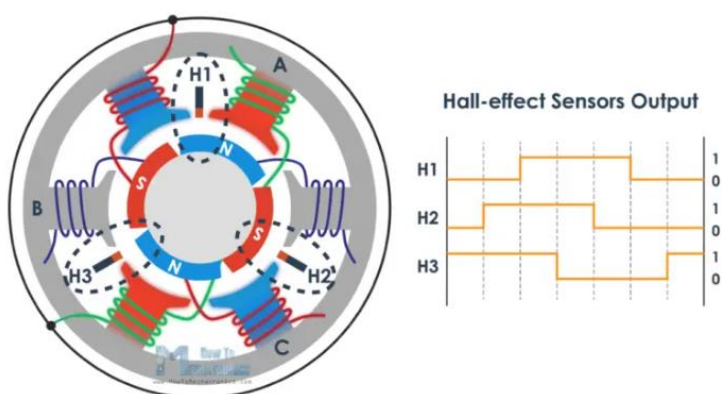


Fig. 1.1.3 Senzorii Hall în motorul BLDC

Înfășurările statorului funcționează împreună cu magneții permanenți de pe rotor pentru a genera o densitate a fluxului cât mai uniformă. Acest lucru permite ca bobinele statorului să fie alimentate de o tensiune continuă constantă (de aici și numele de DC fără perii), care comută de la o bobina la alta a statorului pentru a genera o formă de undă trapezoidală. Numarul de înfășurări utilizare într-un motor fără perii se numește număr de faze.

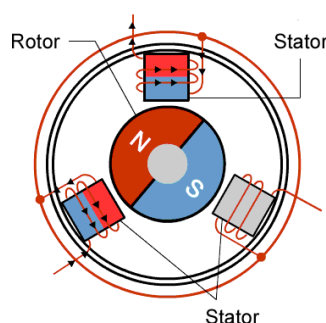


Fig.1.1.4 Motorul Brushless DC (BLDC)

Motoarele BLDC pot fi concepute în multiple variante fizice. În funcție de înfășurările statorului, acestea se clasifică în motoare monofazate, bifazate sau trifazate. Astfel, motoarele BLDC trifazate cu magneți permanenți sunt cele mai utilizate.

De obicei, există două tipuri diferite de construcție pentru acest tip de motor: configurația Inrunner și Outrunner. Configurația inrunner constă în trei înfășurări ale statorului situate în jurul rotorului, cu magneții permanenți ca parte a rotorului. Configurația Outrunner are o relație inversă între magneți și bobine. Magneții permanenți se rotesc în interiorul unui rotor suspendat care înconjoară miezul motorului Brushless DC.

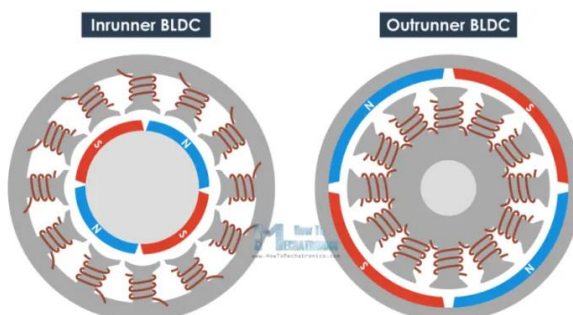


Fig.1.1.5 Configurațiile Inrunner și Outrunner ale unui motor BLDC



Intern, un motor trifazat poate fi configurat la o configurație “Wye” sau „Delta”. Avantajul principal al configurației „Wye”, cunoscută și sub denumirea de configurație Star, este că tensiunea de la fază la neutru este egală în toate cele trei brațe. Acest mod de dispunere constă dintr-un circuit paralel în formă de Y, în care toate înfășurările sunt conectate la un punct central, iar alimentarea este aplicată la capetele libere ale înfășurărilor. Configurația Delta crează un circuit în formă de triunghi, unde fiecare înfășurare este conectată în serie cu celelalte. În această configurație, puterea este aplicată la fiecare dintre varfurile triunghiului.

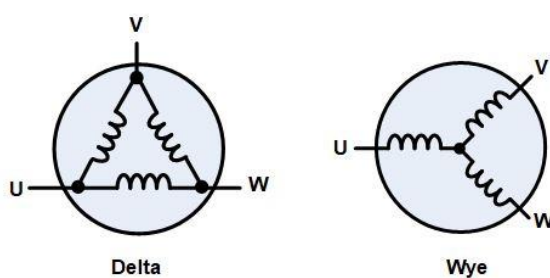


Fig.1.1.6 Configurațiile Delta si Star

Motoarele trifazate pot fi construite în diverse configurații magnetice, cunoscute sub denumirea de poli. Motoarele trifazate elementare au doar doi poli: rotorul are o singură pereche de poli magnetici (unul Nord și unul Sud). Motoarele pot fi construite și cu un număr mai mare de poli, ceea ce necesită mai multe secțiuni magnetice în rotor și mai multe înfășurări în stator.

Numărul de poli afectează performanța unui motor. Mai mulți poli pe rotor necesită cicluri de comutare mai frecvente, datorită necesității unui unghi de  $90^\circ$  pentru a oferi cuplu maxim între câmpul magnetic al rotorului și al statorului. Astfel, pe măsură ce numărul de poli crește, cuplul electromagnetic în motor va fi mai mare, iar viteza maximă va fi redusă.

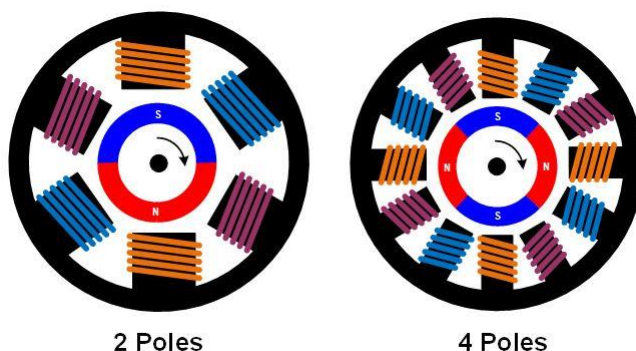


Fig.1.1.7 Numărul de poli al unui BLDC

Datorită faptului că bobinele sunt statice, nu este necesar niciun comutator mecanic pentru alimentarea înfășurărilor. În schimb, comutația se realizează electronic printr-un microcontroler și tranzistoare semiconductoare.

Comutația electronică constă într-o serie de pași în care curentul dintr-un circuit de comandă extern este livrat fiecărei bobine de fază într-o secvență controlată. Acest lucru permite motorului să se rotească corect datorită interacțiunii magnetice dintre rotor și stator.

Pentru a realiza acest lucru într-un motor cu trifazat, curentul este dirijat într-una dintre înfășurările de fază, trece printr-un nod comun și iese printr-o altă înfășurare, lăsând a treia înfășurare deschisă. Prin urmare, atunci când un pol al rotorului este aproape să se alinieze cu polul său electromagnetic corespunzător pe stator, curentul este întrerupt în acea fază și activat în următoarea. Acest proces asigură o mișcare de rotație continuă a motorului.

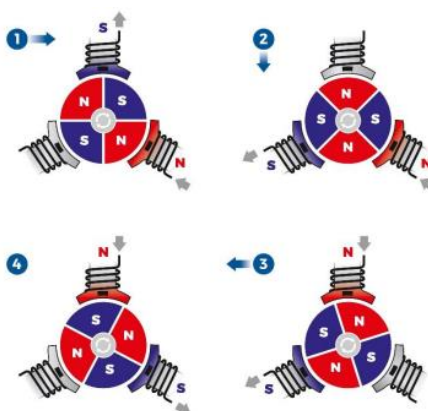


Fig.1.1.8 Comutarea electronică prin înfășurări

Există multe tehnologii de acționare disponibile pentru motorul trifazat fără perii. Cea mai simplă metodă este numită comutația trapezoidală, bloc sau de 120 de grade.

#### **Comutația trapezoidală:**

Comutația trapezoidală (alias în șase trepte) este comună în aplicațiile de mare viteză sau când este necesar un cuplu de pornire mai mare.

Comutația trapezoidală este, de asemenea, mai puțin costisitoare decât alte metode, datorită algoritmilor de control simpli. În majoritatea aplicațiilor, poziția rotorului este determinată de trei senzori cu efect Hall care sunt montați pe stator, la 120 de grade unul de celălalt. Când rotorul trece peste senzori, aceștia produc fie un semnal High, fie un semnal Low

pentru a indica ce pol de rotor (N sau S) trece. Schimbarea de la High la Low (sau Low la High) a celor trei senzori Hall oferă informații despre poziția rotorului la fiecare 60 de grade, ceea ce înseamnă că sunt necesari șase pași pentru a finaliza un ciclu electric – de aici termenul „comutație în șase trepte”.

Secvența corectă de comutare este determinată din combinația semnalelor senzorului Hall. În această schemă, una dintre cele trei faze este întotdeauna conectată la masa, una rămâne liberă, iar cealaltă este conectată la tensiunea de alimentare. Deoarece există doar șase stări ale semnalului pe ciclu electric, curba unghiului cuplului nu este întotdeauna la 90 de grade (cazul ideal), rezultând ondularea cuplului.

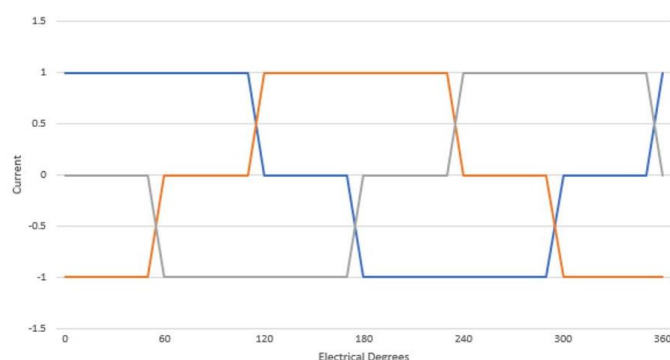


Fig.1.1.9 Comutarea trapezoidală

### Comutația sinusoidală:

Comutația sinusoidală direcționează curentul sub formă de undă sinusoidală prin toate cele trei faze ale motorului, sincronizată sinusoidal de către unitatea de control. Fiecare undă sinusoidală trebuie să fie la 120 de grade electrice una față de cealaltă. Comutarea sinusoidală se realizează utilizând un buclă de feedback cu encoder pentru a menține un unghi de cuplu de 90 de grade. Deoarece curentul este sinusoidal și nu există discontinuități în fluxul de curent, teoretic nu există ondulații de cuplu - deși imperfecțiunile algoritmului de comutare, uniformitatea bobinelor, variațiile în plasarea magnetului, lățimea și forța magnetului și tensiunile de offset ale unității de control non-zero pot contribui la apariția unor ondulații de cuplu.

În multe cazuri, comutarea trapezoidală cu senzori Hall este folosită la pornirea motorului, după care se trece la comutarea sinusoidală, folosind encoderul după prima schimbare de stare Hall. Acest proces evită erorile sau inexactitățile din algoritmul de comutare

al unității de control și profită de comutarea sinusoidală a encoderului odată ce motorul este în mișcare.

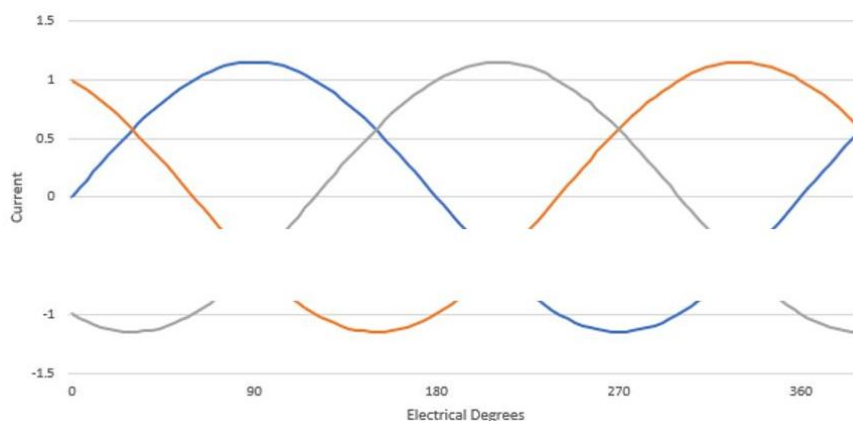


Fig. Comutatie sinusoidală

### **Motoare cu perii și motoare fără perii: Avantaje și dezavantaje:**

La motoarele cu perii, câmpul staționar (statorul) este creat de magneți permanenți care interacționează cu un câmp rotativ (rotorul) care conține înfășurările motorului. Motoarele fără perii sunt exact opusul - prin faptul că câmpul statorului este elementul bobinat, iar câmpul rotativ este magnetul permanent.

În ambele cazuri, interacțiunea acestor câmpuri produce un cuplu care rotește rotorul. Pe măsură ce rotorul se rotește, curentul din înfășurări este comutat pentru a produce un cuplu continuu.

Motoarele cu comutație prin perii folosesc de obicei perii din grafit care se deplasează pe bare de metal (comutatorul) care sunt conectate la bobinele rotorului. Pe măsură ce rotorul se rotește, perii transferă curentul de la un set de bobine la altul. Unitățile fără perii își realizează comutația prin utilizarea unui senzor de poziție a arborelui, care trimite un semnal către un circuit extern de comutare a înfășurărilor.

### **Principiul de funcționare al motorului DC cu perii:**

Motoarele de curent continuu cu perii au bobinele înfășurate în rotor, înconjurat de magneți fixați în stator. Capetele bobinelor sunt conectate la un comutator, care la rândul său se conectează la perii, permițând curentului electric să circule continuu prin perii și bobină cât timp acestea sunt în contact cu comutatorul.

Pe măsură ce bobina se rotește, ajunge într-o poziție unde contactul dintre perii și comutator se întrerupe, oprind curentul în bobină. Totuși, datorită impulsului, bobina continuă să se rotească până când contactul dintre perii și comutator se restabilește, permițând curentului să treacă printr-o bobină diferită.

Această comutare repetată a fluxului de curent menține motorul DC cu perii în mișcare continuă. Viteza acestor motoare poate fi controlată ușor prin ajustarea tensiunii aplicate.

### Principiul de funcționare al motorului BLDC:

Motorul BLDC operează pe un principiu similar cu cel al unui motor DC cu perii. Legea forței Lorentz afirmă că atunci când un conductor prin care parcurge curentul electric este plasat într-un câmp magnetic, asupra lui se exercită o forță. Ca urmare a reacției, magnetul va experimenta o forță egală și opusă. În motorul BLDC, conductorul parcurs de curentul electric este staționar, iar magnetul permanent se mișcă.

Când bobinele statorului primesc alimentare de la sursă, ele devin electromagneți și încep să producă un câmp uniform în fanta de aer. Deși sursa de alimentare este de curent continuu, comutarea face posibilă generarea unei forme de undă a tensiunii cu formă trapezoidală. Datorită interacțiunii dintre electromagnetul statorului și rotorul cu magnet permanent, rotorul rămâne în mișcare continuă.

Comutarea semnalelor dintre High și Low activează înfășurările corespunzătoare, energizându-le ca poli Nord și Sud. Rotorul, având poli Nord și Sud, se aliniază cu polii statorului, ceea ce determină rotația motorului.

	<b>Cu perii</b>	<b>Fără perii</b>
<b>Durată de viață</b>	Scurtă(se uzează periile)	Lungă(fără perii)
<b>Viteza și accelerația</b>	Medii	Ridicate
<b>Eficiență</b>	Medie	Ridicată
<b>Zgomot electric</b>	Ridicat	Scăzut
<b>Zgomot acustic și ondulație de cuplu</b>	Redus	Mediu(trapezoidal)
<b>Cost</b>	Redus	Mediu(electronice adăugate)

Tabel 1.1.1 Avantaje și dezavantaje motoare cu perii și fără perii

### **a. Durata de viață**

Așa cum s-a menționat anterior, unul dintre dezavantajele motoarelor cu perii este uzura mecanică a periilor și a comutatorului. Periile de cărbune se uzează și în multe motoare sunt proiectate pentru a fi înlocuite periodic, ca parte a unui program de întreținere. Cuprul moale al comutatorului se uzează, de asemenea, treptat din cauza contactului cu perii, ajungând în cele din urmă într-o stare în care motorul nu mai funcționează. Pe de altă parte, motoarele fără perii nu au contacte mobile, astfel încât nu suferă de această uzură.

### **b. Viteza și accelerația**

Motoarele cu perii funcționează cel mai bine în intervalul de viteză 1.000 RPM - 10.000 RPM. Vitezele mai mari sunt limitate practic la 10.000 RPM din cauza caracteristicilor de interfață dintre perii și comutator. Pe măsură ce rotorul accelerează, perii încep să plutească peste comutator, realizând un contact fizic și electric slab. Pe de altă parte, motoarele fără perii pot funcționa de obicei la viteze mai mari - acestea sunt limitate doar de integritatea mecanică a rotorului, pierderile legate de viteză și stabilitatea rulmenților utilizați.

### **c. Zgomot electric**

Periile și comutatorul formează un fel de comutator electric. Pe măsură ce motorul se rotește, comutatoarele se deschid și se închid, în timp ce un curent semnificativ curge prin înfășurările rotorului, care sunt inductive. Acest lucru duce la apariția unor scântei la contacte. Aceasta generează o cantitate mare de zgomot electric, care poate pătrunde în circuitele sensibile. Apariția scântei poate fi atenuată într-o anumită măsură prin adăugarea de condensatoare sau amortizoare RC peste perii, dar comutarea instantanee a comutatorului generează întotdeauna un anumit zgomot electric.

### **d. Zgomot acustic**

Motoarele cu perii sunt „comutate dur” – adică, curentul este mutat brusc de la o înfășurare la alta. Cuplul generat variază în timpul rotației rotorului pe măsură ce înfășurările sunt pornite și oprite. În cazul unui motor fără perii, este posibil să se controleze curenții din înfășurări într-un mod care să permită trecerea treptată a curentului de la o înfășurare la alta. Aceasta reduce ondularea cuplului, care este o pulsație mecanică de energie asupra rotorului. Ondularea cuplului cauzează vibrații și zgomot mecanic, în special la viteze mici ale rotorului.

### e. Cost

De multe ori, costurile totale ale ciclului de viață a două produse pot fi un factor decisiv. Un motor fără perii necesită o acționare electronică, în timp ce motorul cu perii nu. Costul suplimentar al acționării electronice face ca sistemele de motoare fără perii să fie mai scumpe decât motoarele cu perii. Ambele tipuri au nevoie de o sursă de alimentare. Motorul cu perii poate funcționa cu o sursă de alimentare directă. Când durata de funcționare devine un factor important, cum ar fi în aplicațiile cu ciclu de funcționare ridicat, costurile ciclului de viață aferente înlocuirii unui motor cu perie pot fi considerabile.

### f. Durata de viață

În general, motoarele fără perii au o durată de viață mai lungă decât motoarele cu perii. Caracteristicile principale care limitează durata de viață a unui motor cu perii sunt periile și comutatorul. O durată tipică de viață a periilor este de 2.000 până la 5.000 de ore de funcționare, dar aceasta nu trebuie considerată o garanție pentru toate aplicațiile. Motoarele fără perii depășesc de obicei 10.000 de ore și sunt în mod obișnuit limitate de durata de viață a rulmenților și de condițiile de mediu.

Atât motoarele cu perii, cât și cele fără perii sunt utilizabile pe piața actuală a sistemelor de control al mișcării. Alegerea între acestea depinde de modul în care factorii menționați anterior influențează obiectivele proiectantului.

Caracteristicile motorului utilizat - Nanotec DF45L024048-A2:

Curent nominal	3,26 A
Inductanță linie la linie	0,27 mH
Constanta de cuplu	3,69 Ncm/A
Conexiune motor	Cablu cu conector
Cuplul maxim	39 Ncm
Putere nominală	65 W
Viteza nominală	4840 rpm
mărimea	45 mm
Rezistență linie la linie	0,64 ohmi
Inerția rotorului	181 gcm <sup>2</sup>
Greutate	0,15 kg
Curentul de vârf	9,5 A
Lungimea "A"	27 mm
Cuplu nominal	13 Ncm
Tensiune nominală	24 V

Tabel 1.1.2 Caracteristicile motorului Nanotec DF45L024048-A2

## 1.2 Placa de dezvoltare LAUNCHXL - F28379D

Kitul de dezvoltare Texas Instruments (TI) F28379D LaunchPad constă dintr-o arhitectură de microcontroler dual-core de înaltă performanță pentru dezvoltarea sistemelor de control avansate. Permite implementarea algoritmilor de control în timp real pentru aplicații precum unități industriale, invertoare și controlul motoarelor. Pentru a valida corect algoritmi de control, poate fi utilizată o configurare de testare C-HIL, în care TI LaunchPad poate fi conectat la un simulator în timp real

LaunchPad-ul C2000™ Delfino™, LAUNCHXL-F28379D, este o placă completă de dezvoltare cu cost redus pentru dispozitivele Texas Instruments Delfino F2837xD. Kitul LAUNCHXL-F28379D include toate componentele hardware și software necesare pentru dezvoltarea aplicațiilor bazate pe microcontrolerele F2837xD. LaunchPad-ul este bazat pe dispozitivul F28379D superset și permite utilizatorilor să migreze ușor către dispozitivele F2837x cu un set de caracteristici mai redus sau cu un număr mai mic de pini, odată ce sunt cunoscute nevoile de proiectare. Placa oferă un instrument de emulare JTAG integrat care permite interfatare directă cu un PC pentru programare, depanare și evaluare ușoară. În plus față de emularea JTAG, interfața USB asigură o conexiune serială UART de la dispozitivul F28379D la PC-ul gazdă.

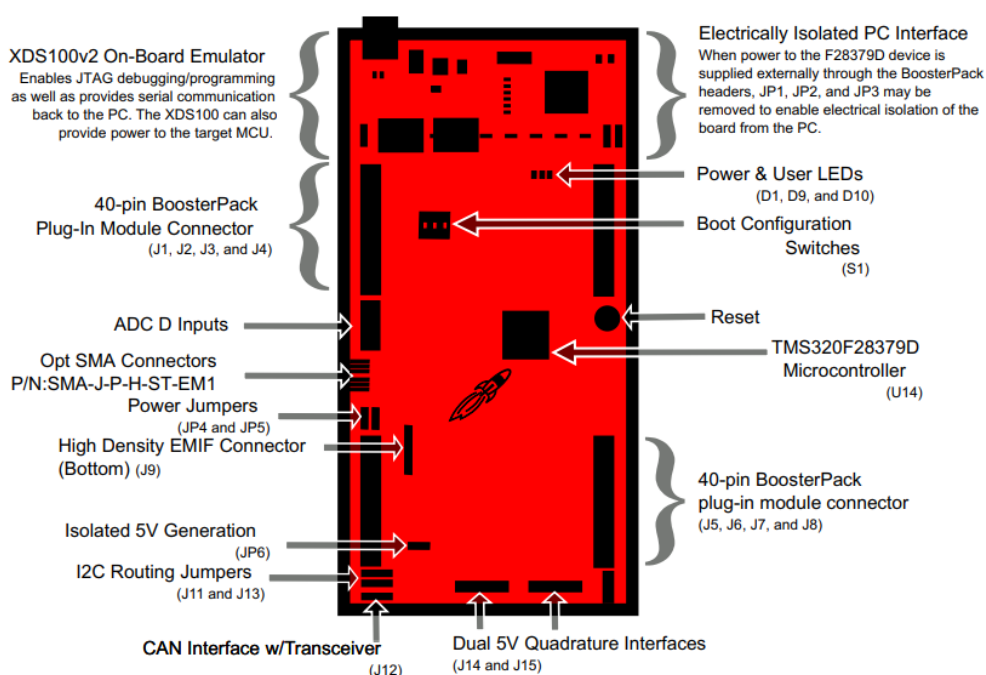


Fig.1.2.1 Caracteristicile hardware ale LAUNCHXL-F28379D



Conform Figurii 1, caracteristicile hardware generale ale LAUNCHXL-F28379D LaunchPad sunt:

- interfață de depanare și programare USB printr-o sondă de mare viteză XDS100v2 izolată galvanic și cu o conexiune USB/UART;
- Interfața de depanare JTAG XDS100v2 izolată conectată prin USB pentru depanare în timp real și programare flash
- 4x20 de pini pentru conexiuni
- Butoane și LED-uri programabile
- TMS320F28379D: 200 MHz dual C28xCPUș și CLA dual, 1 MB Flash, ADC-uri de 16 biți sau 12 biți, comparatoare, DAC-uri de 12 biți, filtre sinc delta-sigma, HRPWM, eCAP, eQEP, CAN-uri și altele
- MCU C2000 Delfino™, TMS320F28379D capabil să interfeteze cu codificatoare absolute, precum și cu rezoltoare și traductoare SINCOS
- Suportă două module suplimentare BoosterPack™
- Doi conectori de interfață pentru codificator
- Conector izolat pentru transceiver CAN

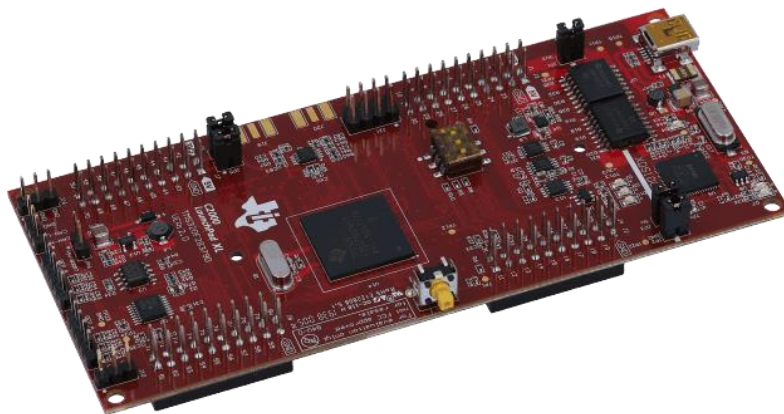


Fig. 1.2.2 Placa de dezvoltare LAUNCHXL - F28379D

Caracteristicile software:

- Descărcare gratuită a Code Composer Studio™ IDE
- Descărcare gratuită a C2000Ware pentru drivere de dispozitiv și exemple de proiecte
- Suport platformă DesignDRIVE

- Suport software powerSUITE
- Suport țintă MathWorks Embedded
- Suport solidThinking Embed

Pentru realizarea proiectului de diplomă a fost necesar add-on-ul “Embedded Coder Support Package for Texas Instruments C2000 processors” din cadrul MATLAB. Configurarea acestuia conduce spre descărcarea unor software-uri necesare pentru conversia schemelor utilizate în cod C și încărcarea codului pe placa de dezvoltare. Software-urile utilizate au fost: TI C2000 Code Generation Tools, TI controlSUITE, TI Code Composer Studio, TI C2000Ware și TI F28044 Headers.

### 1.3 Invertorul BOOSTXL-DRV8305EVM

Modulul de evaluare BoosterPack (EVM) Texas Instruments BOOSTXL-DRV8305EVM trifazic este ideal pentru evaluarea și dezvoltarea rapidă cu DRV8305. Este o unitate de conducere în 3 faze pentru motoare DC fără perii, cu o capacitate de 15A, bazată pe DRV8305 și CSD18540Q5B. DRV8305 este un driver pentru poarta motorului, iar CSD18540Q5B este un MOSFET de putere NexFET™.

Modulul are o magistrală DC individuală și un senzor de tensiune de fază. Modulul furnizează alimentarea de 3.3V pentru MCU cu regulatorul buck LMR16006, cu o capacitate de 0.6A. Etapa de conducere este complet protejată împotriva scurtcircuitelor, supraîncălzirii, fenomenului de supratensiune și subalimentării și poate fi configurată ușor prin intermediul unei interfețe SPI.

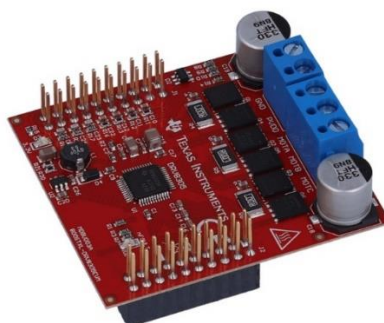


Fig. 1.3.1 Invertorul BOOSTXL-DRV8305EVM

#### Caracteristicile BOOSTXL-DRV8305EVM:

- driver trifazat complet într-un factor de formă compact (2,0 inch × 2,2 inch);
- Suportă alimentare cu tensiune de la 4,4 până la 45 V și curent de comandă de până la 15 A RMS (20 A de vârf)
- 6x MOSFET-uri de putere CSD18540Q5B N-Channel NexFET™ (1,8 mΩ)
  - detectare individuală a tensiunii magistralei DC și fazei motorului;
  - detectare curent de șunt pentru fiecare jumătate de punte;
  - driver complet protejat, inclusiv protecție la scurtcircuit, termică și la subtensiune;
- Intrare de tensiune extinsă LMR16006, regulator 0,6-A pentru alimentarea MCU
- posibilitatea integrării cu kiturile compatibile LaunchPad XL pentru a crea o platformă completă de control al motorului trifazat

Fazele A, B și C ale fiecărui circuit de ieșire jumătate de punte și de detectare a tensiunii de pe magistrala DC (PVDD) sunt caracteristici încorporate în designul BoosterPack. Circuitele prezentate în figura 1.3.2 includ un divizor de tensiune și un condensator de filtrare. Scopul acestor circuite este de a reduce zgomotul de înaltă frecvență care apare pe pinii ADC. Aceste circuite au fost scalate pentru a putea suporta o putere de la 4,4 până la 45 V.

Rezistoarele high-side pentru ieșirile de fază sunt situate lângă port-ul J4, în timp ce rezistențele low-side și condensatorii de filtrare sunt situați lângă pinii ADC de pe BoosterPack pe port-ul J1 pentru o reducere a zgomotului îmbunătățită.

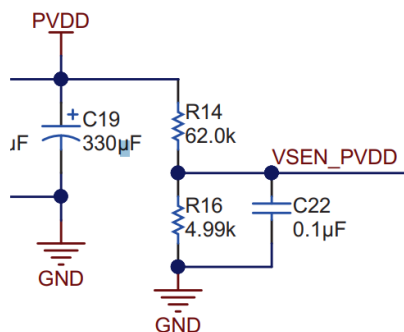


Fig. 1.3.2 Detectarea tensiunii de fază pe magistrala DC

Înlocuirea rezistențelor high-side cu altele cu o valoare mai mică poate fi efectuată pentru a obține o rezoluție mai mare a fluxului de tensiune.

Pentru fiecare jumătate de punte (fazele A, B și C), BoosterPack-ul are un senzor de curent de șunt la partea Low. Amplificatoarele de curent de derivație trifazate ale DRV8305 sunt utilizate în configurația detectării curentului.

Figura 1.3.3 arată configurația pentru sensul de jos. Amplificatorul diferențial detectează tensiunea printr-un rezistor de detectare a energiei cu conexiuni diferite de  $0.007\Omega$ . Pentru a permite detectarea de curent pozitiv și negativ, valoarea diferențială este amplificată cu 10 V/V și centrată la 1.65 V. Rezistorul de sens a fost ajustat pentru curenți de vârf de 0–20A.

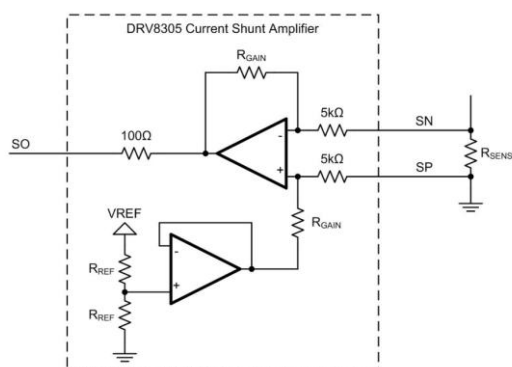


Fig. 1.3.3 Detectare de tip low-side a curentului de șunt

### Regiștri de stare și control ale DRV8305:

DRV8305 oferă raportare extinsă a erorilor și configurare a dispozitivului printr-o interfață SPI și regiștri interni. Există două categorii de regiștri: de stare și de control. Regiștri de stare oferă informații despre defecțiunile și avertismentele dispozitivului. Aceste informații pot include elemente care variază de la supratemperatura IC până la evenimente de supracurent MOSFET. Regiștri de control permit modificarea diversilor parametri ai dispozitivului pentru a se potrivi cerințelor sistemului. Acești parametri includ, dar nu se limitează la, curentul de acționare a porții, timpul mort, configurațiile amplificatoarelor de șunt de curent și modurile de raportare a erorilor.

Interfața grafică InstaSPIN permite accesul ușor la citirea și modificarea registrelor interne ale DRV8305. Acestea pot fi accesate în fila DRV8305 a interfeței grafice universale InstaSPIN.



Fig. 1.3.4 InstaSPIN GUI – activare sistem

Pentru a alimenta BoosterPack-ul și a permite citirile și scrierile SPI, sistemul trebuie să fie alimentat și opțiunea Enable System trebuie verificată. Fila SPI a DRV8305 afișează o hartă a regiștrilor interni ai DRV8305, incluzând atât regiștri de stare, cât și cei de control. Pentru a citi din regiștri, se selectează butonul Read, iar harta regiștrilor se va actualiza cu valorile curente. Pentru a scrie în regiștri, se fac modificările dorite în hartă și se apasă butonul Write. Regiștri individuali pot fi citați sau scriși manual folosind butoanele Manual Write și Manual Read (în format zecimal).

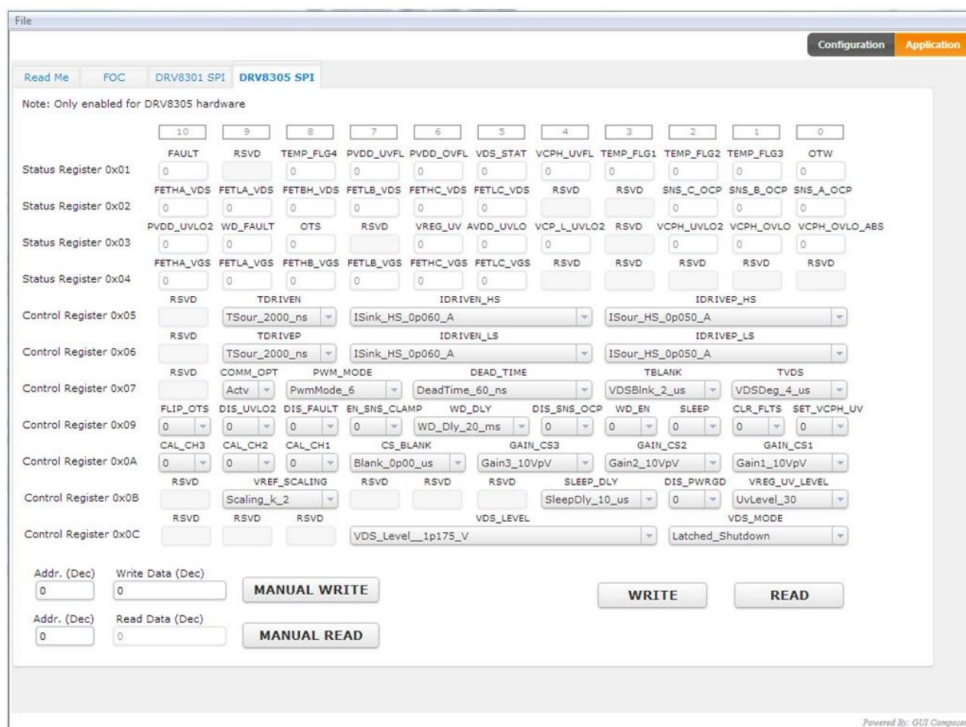


Fig. 1.3.5 InstaSPIN GUI – regiștri SPI

## 1.4 Montajul realizat

Pentru realizarea montajului, pe lângă echipamentele deja prezentate - motorul fără perii cu magneți permanenți BLDC, placa de dezvoltare și inverterul - au mai fost necesare o sursă de tensiune setată la 12V/3A și o parte de cablaj pentru realizarea conexiunilor. În figura 1.4.1 este ilustrat montajul final realizat.

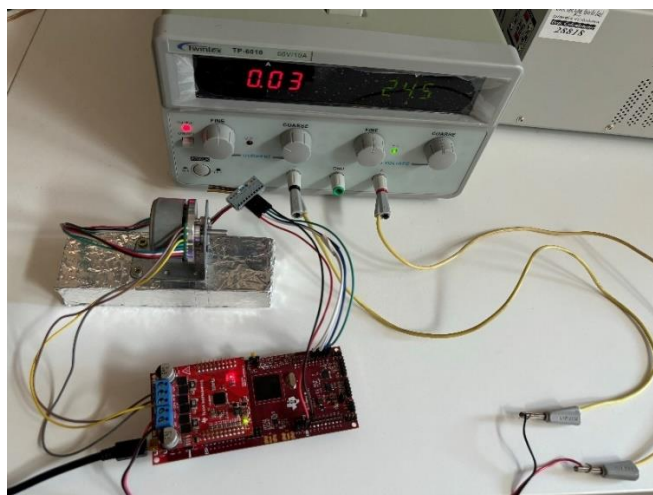


Fig. 1.4.1 Montajul realizat

Înainte de montarea inverterului pe placa DSP este necesară eliminarea jumperelor JP1(3.3V), JP2(GND) și JP3(5V) pentru a izola unitatea microcontroler (MCU) și emulatorul plăcii de dezvoltare. Aceasta se face pentru a preveni interferențele și pentru a asigura funcționarea corectă a sistemului. În plus, pentru a asigura funcționarea corectă, switch-ul S1 trebuie setat în poziția ON pentru toate cele trei poziții.

Sursa de tensiune alimentează motorul cu ajutorul inverterului prin cele două perechi de fire galben-rosu, reprezentand borna plus și galben-negru borna minus. Motorul este alimentat prin trei fire (maro, gri și galben) de la inverter, corespunzătoare fazelor U, V și W.

Senzorii Hall sunt alimentați de la placa de dezvoltare la +3.3V prin intermediul firului roșu situat în partea de jos a plăcii, în timp ce firul negru este conectat la masa (GND) a portului J7. Transmiterea semnalului de la senzori către placa de dezvoltare LAUNCHXL se realizează prin cele trei fire de culoare verde, albastru, respectiv alb, conectate la pinii 51, 52 și 53 ai portului J6. Starea senzorilor Hall 2 și Hall 3 este preluată și pentru modulul eCAP. Astfel, pentru Hall 2 și Hall 3 sunt folosite fire duble conectate pe pinii 54, respectiv 55 ai portului J6. Modulul eCAP, va fi detaliat într-un capitol următor.

# CAPITOLUL 2

## PROBLEME DE CONTROL ALE MODELELOR NELINIARE

Capitolul 2 al lucrării se concentrează pe aspectele teoretice ale controlului neliniar și ale modelelor neliniare, dat fiind că aceasta lucrare se ocupă de un astfel de model. La sfârșitul acestui capitol vor fi prezentate diferitele tipuri de comportament specifice acestor sisteme.

Modelarea matematică a sistemelor din viața reală este o metodologie foarte frecvent utilizată în știință și inginerie. Este folosită atât ca mijloc pentru obținerea unei înțelegeri mai profunde despre un sistem, cât și ca unealtă de inginerie, de exemplu, ca bază pentru simulări sau pentru proiectarea de controlere.

### 2.1 Sistemele neliniare

Sistemele fizice au o natură intrinsec neliniară. Prin urmare, toate sistemele de control sunt, într-o anumită măsură, neliniare. Sistemele de control neliniare pot fi descrise prin ecuații diferențiale neliniare. Cu toate acestea, dacă gama de funcționare a unui sistem de control este mică și dacă neliniitățile implicate sunt mici/netede, atunci sistemul de control poate fi aproximativ în mod rezonabil printr-un sistem liniarizat, ale cărui dinamici sunt descrise printr-un set de ecuații diferențiale liniare.

Forma generală a unui sistem liniar omogen de ecuații diferențiale de ordinul întâi este:

$$\begin{cases} \dot{x}_1(t) = a_{11}(t) x_1(t) + a_{12}(t) x_2(t) \\ \dot{x}_2(t) = a_{21}(t) x_1(t) + a_{22}(t) x_2(t) \end{cases} \quad (2.1)$$

unde  $a_{ij} \in \mathbb{C}[a; b]$ ,  $i, j \in 1, 2$ .

Dacă notăm cu:

$$X = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$$

obținem forma vectorială a sistemului (2.1):

$$X' = A * X \quad (2.2)$$

Așa cum am menționat anterior, un sistem dinamic este un sistem al cărui stări se schimbă în decursul timpului. Dinamica unui sistem se numește liniară, dacă relația ce leagă starea prezenta de starea viitoare este liniară, în caz contrar avem de-a face cu o dinamică neliniară.

Un sistem de control neliniar este, în esență, un sistem care nu urmează regulile simple ale sistemelor liniare discutate anterior, ele sunt mult mai complexe. Această complexitate înseamnă că reacțiile și interacțiunile în cadrul sistemului pot varia în moduri imprevizibile și neliniare, necesitând abordări de control mai avansate și adaptabile pentru a gestiona eficient comportamentele lor.

Nonlinearitățile pot fi clasificate ca fiind intrinseci (naturale) și intenționale (artificiale). Nonlinearitățile intrinseci sunt acelea care vin în mod natural cu hardware-ul. Exemple de nonlinearități intrinseci includ forțele centrifuge în mișcarea de rotație, caracteristicile tensiune-curent ale majorității sistemelor electronice, rezistența la un vehicul în mișcare și frecarea Coulomb între suprafețele aflate în contact. De obicei, astfel de nonlinearități au efecte nedorite, iar sistemele de control trebuie să le compenseze corespunzător.

Pe de altă parte, nonlinearitățile intenționale sunt introduse artificial de către proiectant. Legile de control neliniare, cum ar fi legile de control adaptativ și legile de control optimal bang-bang, sunt exemple tipice de nonlinearități intenționale.

<b>Sistem de control liniar</b>	<b>Sistem de control neliniar</b>
Sistemul de control liniar răspunde previzibil la modificarea intrărilor.	Nu răspunde în mod previzibil la modificările de intrare dincolo de un punct.
Modificările scalei de intrare afectează liniar ieșirea.	Nu prezintă scalabilitate liniară cu intrări.
Preziceți comportamentul în limitele specificate.	Comportamentul imprevizibil al sistemului de control neliniar, mai ales dincolo de anumite praguri.
Acest sistem respectă regulile specifice în mod constant.	Nu urmează modele simple previzibile pe parcursul operațiunii.

Tabel 2.1 Diferența dintre sistemul de control liniar și neliniar



Sistemele de control liniare urmează reguli specifice de omogenitate și aditivitate care prezintă răspunsuri previzibile la modificările de intrare și mențin o relație consistentă între intrări și ieșiri în limitele definite. Sistemele de control neliniare, opuse, nu respectă aceste reguli simple, acest lucru arată un comportament imprevizibil, mai ales dincolo de anumite praguri. În timp ce sistemele liniare aderă la modele consistente, sistemele neliniare prezintă comportamente complexe și neașteptate, deviind de la relațiile simple dintre intrări și ieșiri.

Distincția constă în modul în care acestea răspund la schimbări: sistemele liniare urmează o cale previzibilă, în timp ce sistemele neliniare prezintă comportamente complexe care depășesc tiparele simple și previzibile, mai ales atunci când sunt împinse dincolo de anumite limite, așa cum se vede în fenomene precum saturația în curbele de magnetizare a mașinilor de curent continuu.

## 2.2 Sistemele de control cu feedback

Un sistem de control cu feedback monitorizează continuu un proces (sau instalație) și influențează procesul în așa fel încât unul sau mai mulți parametri ai procesului (variabilele de ieșire) să rămână într-un interval prescris. Un sistem de control cu feedback necesită cel puțin următoarele componente:

**Un proces/instalație.** Procesul este responsabil pentru variabila de ieșire. În plus, procesul oferă un mijloc de a influența variabila de ieșire.

**Un senzor.** Senzorul măsoară continuu variabila de ieșire și convertește valoarea variabilei de ieșire într-un semnal care poate fi procesat în continuare, cum ar fi o tensiune (în sistemele electrice de control), o poziție (în sistemele mecanice) sau o presiune (în sistemele pneumatice).

**Un punct de referință** sau, mai precis, un mijloc de a regla un punct de referință. Valoarea de referință este legată de variabila de ieșire, dar are aceleași unități ca și ieșirea senzorului.

**Un controler.** Controlerul măsoară abaterea variabilei controlate de la valoarea de referință și creează o acțiune corectivă. Acțiunea corectivă este cuplată cu intrarea procesului și utilizată pentru a conduce variabila de ieșire către valoarea de referință.

O metodă pentru a asigura că ieșirea unui sistem urmează un semnal de referință dorit constă în compararea ieșirii actuale a sistemului cu cea dorită și de a oferi feedback pentru a ajusta ieșirea astfel încât să se apropie cât mai mult de valoarea de referință.

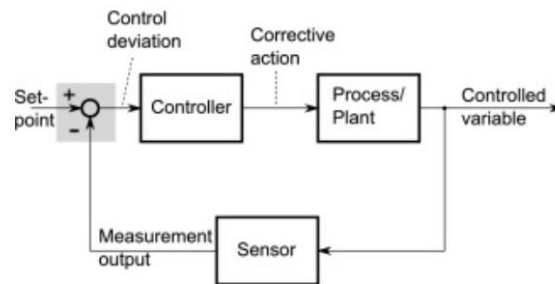


Fig. 2.1 Sistem de control cu feedback

## CAPITOLUL 3

### STRATEGIILE DE CONTROL

Capitolul al treilea cuprinde prezentarea părții teoretice a metodei de control abordate, respectiv controllerul de tip PI. Pentru această metodă a fost făcută trecerea din domeniul timp în continuu și apoi în discret pentru a lucra cu o anumită perioadă de eșantionare

#### 3.1 Controlerul proporțional (P)

Un controler proporțional (P-controller) este un tip de sistem de control prin feedback care ajustează semnalul de control al unui sistem  $u(t)$  proporțional cu eroarea  $e(t)$ , calculată între valoarea de referință și ieșirea reală  $y(t)$ . Acest tip de controler este utilizat în mod obișnuit în sistemele de control în buclă închisă pentru a se asigura că ieșirea sistemului rămâne cât mai aproape posibil de valoarea de referință.

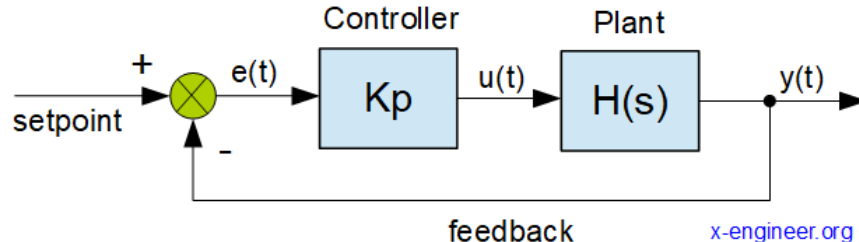


Fig. 3.1 Schemă controler proporțional

Ecuția de bază pentru un controler proporțional este următoarea:

$$Co(t) = K_p * e(t) \quad (3.1)$$

Unde:

$Co(t)$  – semnal de control (ieșirea controlerului)

$e(t)$  – semnal de eroare (intrare în controler)

$K_p$  – constrângeri de reglare pentru acțiune proporțională (câștig proporțional)

Câștigul proporțional și determină cât de repede răspunde controlerul la modificări ale punctului de referință sau ale ieșirii reale. Cu cât câștigul proporțional este mai mare, cu atât răspunsul controlerului este mai rapid.

Există câteva dezavantaje ale acestor controlere:

1. Datorită prezenței acestor controlere, obținem unele abateri în sistem.
2. Controlerele proporționale pot duce la creșterea overshoot-ului maxim, afectând stabilitatea sistemului.

### 3.2 Controlerul integrator (I)

După cum sugerează și numele, ieșirea (numită și semnal de acționare) este direct proporțională cu integrala semnalului de eroare. Acum să analizăm matematic controlerul integral.

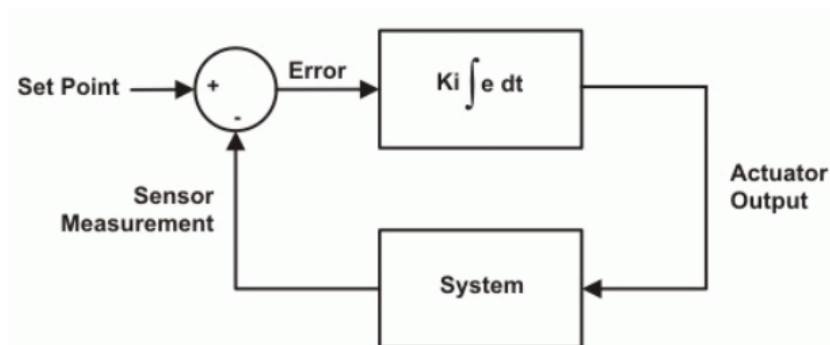


Fig. 3.2 Schemă controler integrator

După cum știm într-un controler integral, ieșirea este direct proporțională cu integrarea semnalului de eroare, scriind acest lucru matematic avem:

$$Co(t) = K_i * \int_0^t e(t)dt \quad (3.2)$$

Unde  $K_i$  este o constantă integrală cunoscută și sub denumirea de câștig al controlerului. Controlerul integral este cunoscut și sub numele de controler de resetare.

Dezavantajele controlerului integral este acela că are tendința de a face sistemul instabil deoarece răspunde lent la eroarea produsă.

### 3.3 Controlerul proportional integrator (PI)

Controlerul PI este o combinație de acțiune a controlerului proporțional și acțiunea controlerului integrator.

Controlerul proporțional-integrator, cunoscut în mod obișnuit ca controler PI este o parte esențială a sistemului de automatizare și control industrial. Este un mecanism de control al feedback-ului în buclă închisă care urmărește ajustarea variabilei de proces prin manipularea variabilei pe baza erorii dintre valoarea de referință și variabila de proces. Găsește un echilibru între răspunsul rapid la abateri și eliminarea erorilor pe termen lung.

Schema bloc generală a unui sistem cu regulator PI este prezentată în figura 3.1.1:

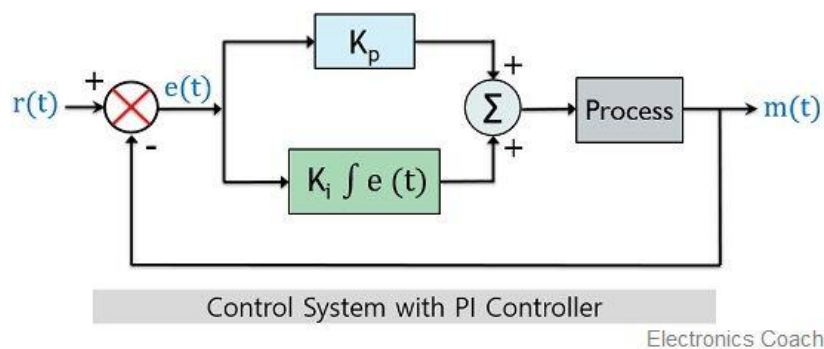


Fig. 3.1.1 Sistem de control cu controller PI

Semnalul de control arată proporționalitatea cu semnalul de eroare și integrala semnalului de eroare în acest caz. Ecuația (3.3) arată reprezentarea matematică a controllerului proporțional plus integrator.

Acum este timpul să înțelegem formula matematică din spatele ei. Putem reprezenta un controler PI în expresie matematică din definiție. Și din aceasta se face o diagramă bloc pentru o mai bună înțelegere a funcționării unui controler PI. Mai jos este elaborarea ecuației și diagramei unui controler PI.

Din ecuația numărul (3.1) și (3.2) putem scrie adunându-le pe ambele,

$$Co(t) = K_p * e(t) + K_i * \int_0^t e(t)dt \quad (3.3)$$

Această ecuație indică faptul că controlerul PI funcționează ca un controler PI simplificat fără o acțiune derivată.

Pentru a obține funcția de transfer a controller-ului, trebuie să aplicăm transformata Laplace a ecuației (3.3), astfel rezultă:

$$L\{Co(t)\} = Kp \cdot Le(t) + Ki \cdot L\{\int e(t)dt\} \quad (3.4)$$

Transformarea Laplace a termenului integral este exprimată folosind proprietatea  $L\{\int f(x)dx\} = \left(\frac{1}{s}\right) \cdot L\{f(x)\}$ , rezultând:

$$L\{Co(t)\} = Kp \cdot L\{e(t)\} + Ki \cdot \left(\frac{1}{s}\right) \cdot L\{e(t)\} \quad (3.5)$$

În domeniul Laplace,  $C(s)$  reprezintă ieșirea transformată,  $[L\{Co(t)\}]$ ,  $E(s)$  reprezintă semnalul de eroare transformat,  $[L\{e(t)\}]$ . Deci putem reforma ecuația de mai jos,

$$C(s) = Kp * E(s) + Ki * \left(\frac{1}{s}\right) * E(s) \quad (3.6)$$

$$C(s) = Kp * \left\{1 + \left(\frac{Ki}{s} * Kp\right)\right\} * E(s) \quad (3.7)$$

Putem simplifica relația, substituind:  $\frac{Kp}{Ki} = Ti$ , sau  $\frac{Ki}{Kp} = \frac{1}{Ti}$

$$C(s) = Kp * \left\{1 + \left(\frac{1}{s} * Ti\right)\right\} * E(s) \quad (3.8)$$

$$\frac{C(s)}{E(s)} = Kp * \left\{1 + \left(\frac{1}{s} * Ti\right)\right\} \quad (3.9)$$

Unde,  $Kp$  este câștigul proporțional și  $Ti$  este constanta de timp integrală.

Putem exprima  $C(s)$  ca  $M(s)$ . Acum ecuația va fi:

$$\frac{M(s)}{E(s)} = Kp * \left\{1 + \left(\frac{1}{s} * Ti\right)\right\} \quad (3.10)$$

Deci, schema bloc a unui controller PI ar arăta astfel:

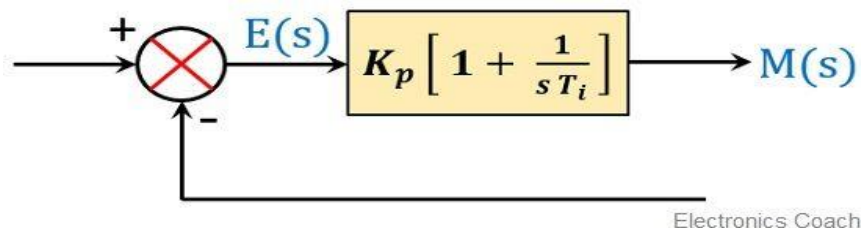


Fig. 3.1.2 Controller-ul PI cu parametrii  $K_p$  și  $K_i$

Un controler proporțional operează pas cu pas pentru a regla un sistem de control. Acesta începe prin calcularea erorii, care reprezintă diferența dintre valoarea de referință dorită și valoarea curentă a procesului. Partea proporțională (P) a controlerului înmulțește această eroare cu câștigul proporțional ( $K_p$ ) și generează o acțiune imediată, care este direct proporțională cu eroarea.

Partea Integral(I) calculează suma cumulativă a erorilor trecute prin integrarea câștigului ( $K_i$ ). Cele două componente P și I sunt apoi adunate împreună pentru a determina ieșirea de control, care este aplicată sistemului. Această ieșire ajustează ieșirea finală a sistemului reducând la minimum eroarea în timp și menținând valoarea de referință.

Un controler proporțional-integrator este o componentă a sistemului de control cu feedback, descris de ecuația  $C_o(t) = K_p \cdot e(t) + K_i \cdot \int e(t) dt$ , combinând acțiunea componentelor integratoare și proporționale. Componenta proporțională răspunde la eroarea curentă, în timp ce componenta integratoare acumulează erorile trecute pentru a elimina erorile la starea de echilibru.

Reglarea corectă a câștigurilor  $K_p$  și  $K_i$  este crucială pentru a obține un control stabil cu răspuns rapid și eroare minimă în diferite aplicații de control. Diagrama controlerului PI ilustrează funcționarea acestuia arătând cum intrarea erorii este procesată prin acțiune a celor două componente pentru a genera o ieșire care controlează procesul.

Pentru a înțelege efectele controller-ului PI, luăm în considerare controller-ul PI cu feedback negativ unitar:

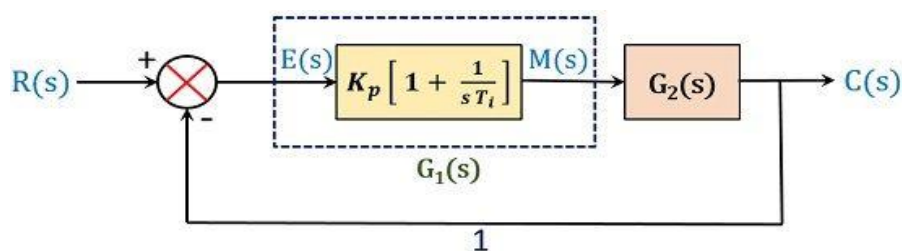


Fig. 3.1.3 Controller-ului PI cu feedback negativ unitar

Să presupunem că funcția de transfer a controller-ului este  $G_1(s)$ . Valoarea acesteia a fost evaluată recent ca:

$$G_1(s) = K_p * \left(1 + \frac{1}{T_i s}\right) \quad (3.11)$$

Iar funcția de transfer a sistemului în buclă deschisă  $G_2(s)$  :

$$G_2(s) = \frac{\omega_n^2}{s*(s+2*\xi*\omega_n)} \quad (3.12)$$

Stiind că:

$$G(s) = G_1(s) * G_2(s) \quad (3.13)$$

Dupa substituție:

$$G(s) = K_p * \left(\frac{1+T_i s}{T_i s}\right) * \frac{\omega_n^2}{(s^2+2*\xi*\omega_n*s)} \quad (3.14)$$

Știm că funcția de transfer a sistemului cu reacție este:

$$\frac{C(s)}{R(s)} = \frac{G(s)}{1+G(s)*H(s)} \quad (3.15)$$

Deoarece am considerat deja că stemul este unul cu reacție unitară,  $H(s) = 1$ , rezultând astfel funcția de transfer cu feedback/reacție unitară:

$$\frac{C(s)}{R(s)} = \frac{G(s)}{1+G(s)} \quad (3.16)$$

Înlocuind valorile lui  $G(s)$ , vom obține:

$$\frac{C(s)}{R(s)} = \frac{K_p * \left(\frac{1+T_i s}{T_i s}\right) * \frac{\omega_n^2}{(s^2+2*\xi*\omega_n*s)}}{1 + K_p * \left(\frac{1+T_i s}{T_i s}\right) * \frac{\omega_n^2}{(s^2+2*\xi*\omega_n*s)}} \quad (3.17)$$

Continuând prelucrarea ecuației (3.1.17), aceasta devine:

$$\frac{C(s)}{R(s)} = \frac{K_p * \left(\frac{1+T_i s}{T_i s}\right) * \frac{\omega_n^2}{(s^2+2*\xi*\omega_n*s)}}{1 + K_p * \left(\frac{1+T_i s}{T_i s}\right) * \frac{\omega_n^2}{(s^2+2*\xi*\omega_n*s)}} \quad (3.18)$$



$$\frac{C(s)}{R(s)} = \frac{K_p * (1 + T_i s) * \omega_n^2}{T_i s * (s^2 + 2 * \xi * \omega_n * s) + K_p * (1 + T_i s) * \omega_n^2} \quad (3.19)$$

$$\frac{C(s)}{R(s)} = \frac{K_p * (1 + T_i s) * \omega_n^2}{T_i s^3 + 2 * \xi * \omega_n * s^2 + T_i * K_p * \omega_n^2 + K_p * \omega_n^2 * T_i s} \quad (3.20)$$

Scoțând  $T_i$ , de la numitor din (3.20), rezultă:

$$\frac{C(s)}{R(s)} = \frac{K_p}{T_i} * \frac{(1 + T_i s) * \omega_n^2}{s^3 + 2 * \xi * \omega_n * s^2 + \frac{K_p * \omega_n^2}{T_i} + K_p * \omega_n^2 * s} \quad (3.21)$$

Ținând cont că:

$$T_i = \frac{K_p}{K_i} \rightarrow K_i = \frac{K_p}{T_i} \quad (3.22)$$

Substituind în (3.22) în (3.21), obținem funcția de transfer cu reacție unitară a controlerului PI:

$$\frac{C(s)}{R(s)} = K_i * \frac{(1 + T_i s) * \omega_n^2}{s^3 + 2 * \xi * \omega_n * s^2 + \frac{K_p * \omega_n^2}{T_i} + K_p * \omega_n^2 * s} \quad (3.23)$$

Deja am menționat că controllerele PI sunt concepute pentru a reduce eroarea în starea de echilibru. La proiectarea unui sistem de control, factorii de performanță: suprareglarea (overshoot) și timpul de stabilizare sunt cruciali. .

În ingineria sistemelor și teoria controlului, suprareglare, sau overshoot, este fenomenul în care răspunsul unui sistem dinamic depășește temporar valoarea de referință sau punctul de referință înainte de a se stabiliza la valoarea finală. Acesta este calculat prin diferența dintre valoarea maximă atinsă și valoarea de referință. Este de obicei exprimată ca procent din valoarea de referință.

Timpul de stabilizare reprezintă timpul necesar pentru ca răspunsul sistemului să ajungă și să rămână într-o anumită bandă de toleranță în jurul valorii de referință după ce a fost aplicată o perturbare.

## CAPITOLUL 4

# PROIECTAREA ÎN MATLAB/SIMULINK

În acest capitol voi descrie modul în care am implementat metodele de control în Matlab/Simulink. Voi începe cu achiziția datelor de la senzorii Hall ai motorului și voi continua cu subsistemele pentru determinarea poziției și a vitezei rotorului, realizarea transmisiei și a recepției seriale(SCI) și fiecare metodă de control folosită. Scriptul Matlab folosit este atașat în partea de anexe a acestei lucrări.

Pentru a realiza proiectul în MATLAB/Simulink au fost necesare: scriptul Matlab, o schema Simulink pentru partea de control și o schema Simulink pentru transmisia serială.

### 4.1 Modulul eCAP (Enhanced Capture)

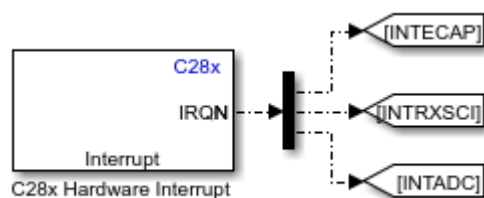


Fig. 4.1.1 Generare întreruperi pentru modulul eCAP, recepția serială și modulul ADC

Modulul eCAP (Enhanced Capture) este un component al unor microcontrolere. Este utilizat pentru măsurarea timpului cu o rezoluție ridicată și pentru capturarea temporizărilor precise ale evenimentelor externe.

În aceasta lucrare voi folosi modulul eCAP pentru a surprinde evoluția semnalelor senzorilor Hall utilizați pentru determinarea vitezei de rotație a motorului BLDC.

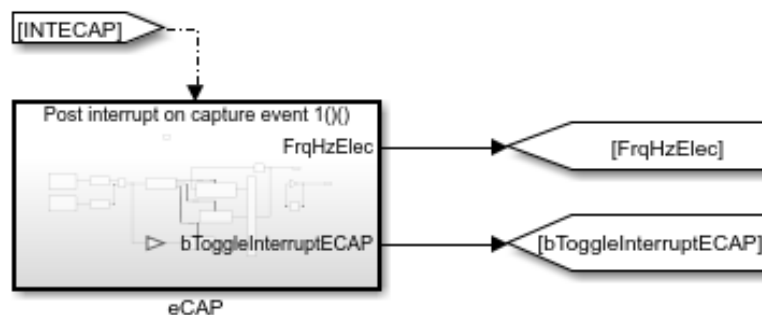


Fig. 4.1.2 Subsistemul modulului eCAP

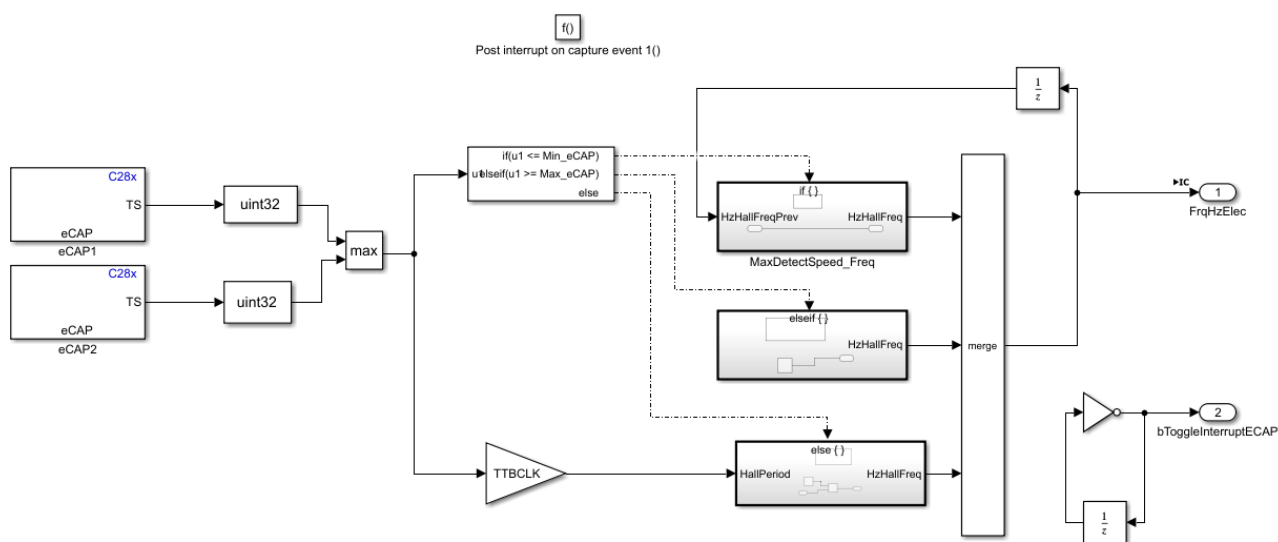


Fig. 4.1.3 Subsistemul modului eCAP detaliat

Subsistemul eCAP de tip C28x utilizează pinii GPIO63 pentru eCAP1, respectiv GPIO64 pentru eCAP2, aceștia fiind setați în configurarea hardware a plăcii de dezvoltare LAUNCHXL-F28379D. Aceștia corespund pinilor 54 și 55 ai portului J6, iar aceste semnale provin de la cei doi senzori Hall (Hall2 și Hall3 menționați în capitolul 1.4).

În structura acestui subsistem se configurează output-urile FrqHzElec(frecvența) și bToggleInterruptECAP(întreruperea la comutarea semnalelor senzorilor Hall) utilizate apoi în vederea determinării vitezei în subsistemul “position and speed calculation”.

## 4.2 Determinarea poziției cu ajutorul senzorilor Hall

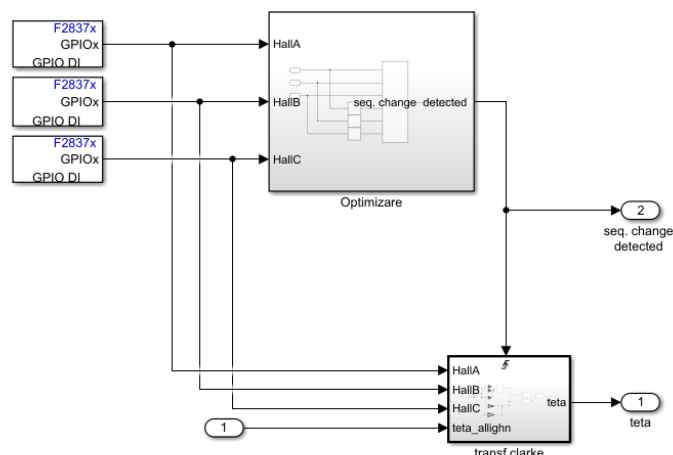


Fig. 4.2.1 Subsistemul de calcul al poziției

Pentru a controla precis un motor BLDC este necesar sa cunoaștem în orice moment poziția rotorului. Pentru a determina poziția, m-am folosit de cei trei senzori Hall conectați la pini 51, 52 și 53 ai portului J6(GPIO25, GPIO27, respectiv GPIO26). Perifericul GPIO oferă pini dedicați care pot fi folosiți fie ca intrare, fie ca ieseire. În cazul de față au fost configurați ca și pini de intrare, pentru a putea detecta starea senzorilor(GPIO DI-digital input).

Pentru a determina unghiul  $\theta$  (teta) al rotorului se utilizează Transformata Clarke. Aceasta transformă semnalele senzorilor Hall în sistemul de coordonate  $\alpha - \beta$ , iar apoi, prin calcularea arctangentei acestor componente se obține unghiul  $\theta$ , care reprezintă poziția unghiulară a rotorului în timp real.

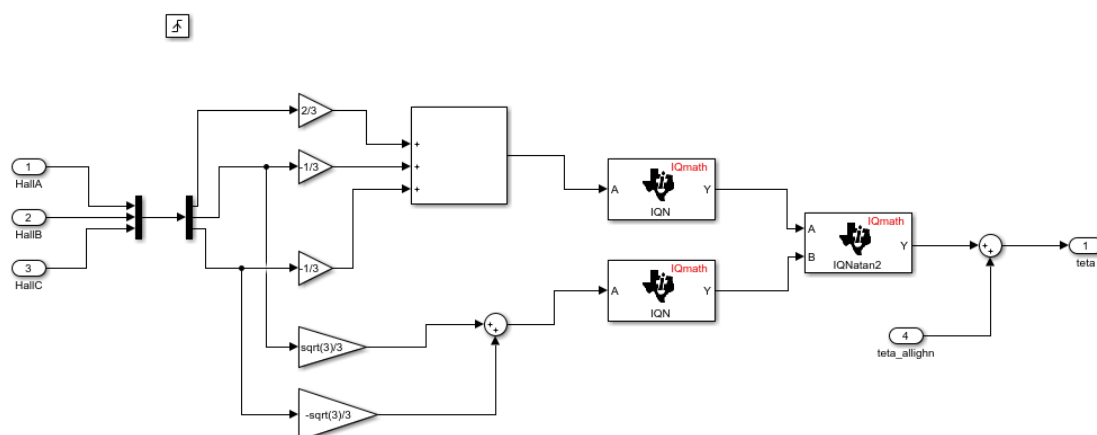


Fig. 4.2.2 Transformata Clarke și determinarea unghiului  $\theta$  (teta)

Pentru a spori eficiența proiectului, am implementat o optimizare care funcționează ca un Enable pentru Transformata Clarke. Această optimizare face ca Transformata Clarke să fie calculată doar când se detectează o schimbare a stării senzorilor Hall, astfel încât să se reducă utilizarea resurselor de calcul.

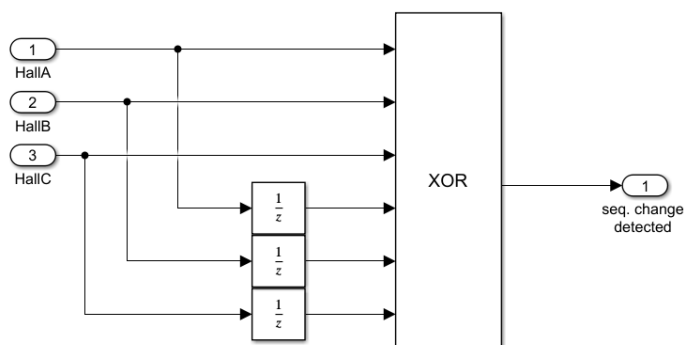


Fig.4.2.3 Detectarea schimbarii senzorilor Hall

## 4.3 Determinarea vitezei

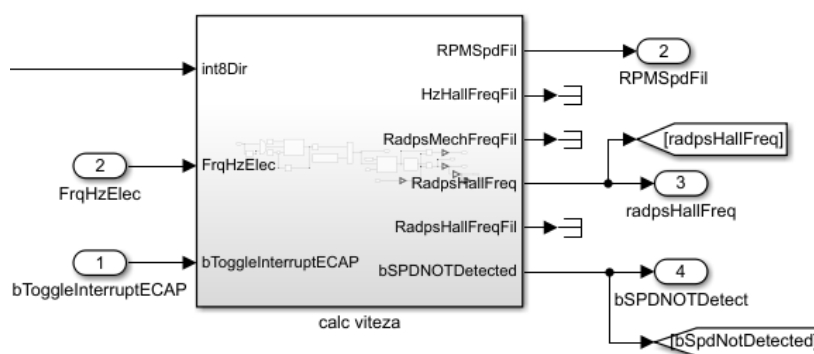


Fig 4.3.1 Subsistemul de calcul al vitezei

La ieșirea sistemului obținem viteza filtrată a motorului, “RPMSPdFil” exprimată în rotații pe minut. Acesta este rezultatul final al subsistemului, care va fi utilizat pentru implementarea și reglarea controlerelor prezentate ulterior în lucrare.

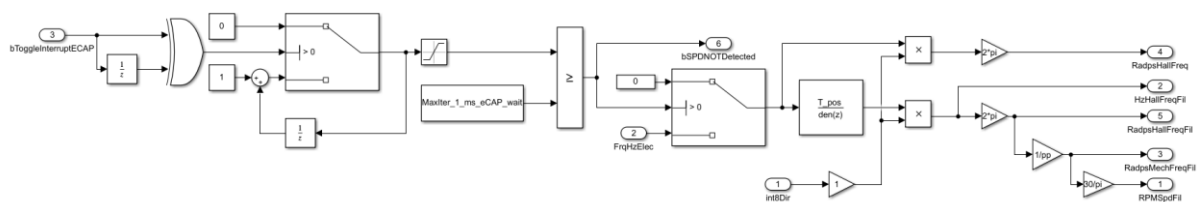


Fig 4.3.2 Subsistemul de calcul al vitezei detaliat

## 4.4 Subsistemul transmisiei seriale

Modulul SCI(Serial Communication Interface) este un subsistem hardware utilizat în microcontrolere și procesoare pentru a permite comunicarea serială bidirecțională între dispozitivul gazdă și alte dispozitive externe.

Blocul SCI XMT(SCI Transmit) transmite date scalare sau vectoriale utilizând modulul hardware specificat. Procesorul utilizat in aceasta lucrare are 4 astfel de module SCI(A, B, C și D).

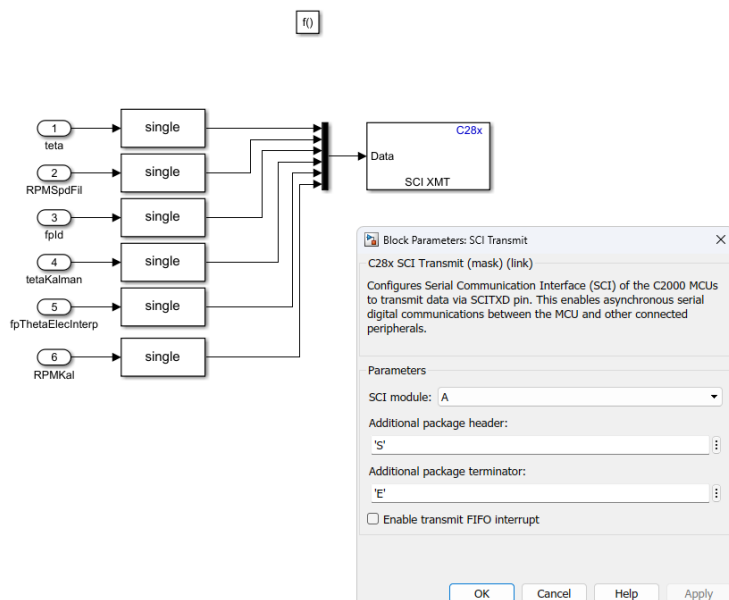


Fig. 4.4.1 Subsistemul transmisiei seriale

Pentru a realiza transmisa, am folosit un singur modul SCI Transmit, configurat pe modulul A al plăcii de dezvoltare, pe pinul GPIO42. Acesta transmite parametrii calculați în etapele anterioare(teta, RPMSpdFill, fpid).

În configurarea blocului SCI Transmit din Simulink, parametrii din „Additional package header” și „Additional package terminator” (‘S’, respectiv ‘E’), sunt setări suplimentare care permit adăugarea unor caractere specifice la începutul și la sfârșitul fiecărui pachet de date transmis. Acești parametri ajută la identificarea pachetelor și îmbunătățesc sincronizarea datelor transmise. Aceleași notații au fost respectate și în interfața “PC\_host\_SCI\_interface\_PI” detaliată în paginile următoare.

## 4.5 Subsistemul recepției seriale

Blocul C28x SCI RCV (SCI Receive) primește date scalare sau vectoriale printr-un canal de comunicare serială(UART) într-un model Simulink. În proiectul acesta, acest bloc utilizează tot modulul A (ca și în cazul transmisiei seriale). Un model poate conține doar un bloc SCI Receive pentru fiecare modul. Pinul alocat este GPIO43, iar, la fel ca la blocul SCI Transmit, întâlnim parametrii „Additional package header” și „Additional package terminator” (‘SS’, respectiv ‘EE’). După cum se poate observa si în figura 4.3.1, acest bloc necesită configurarea parametrului “Data length” pentru a face cunoscută lungimea secvenței de date(serial\_data\_length = 9;).

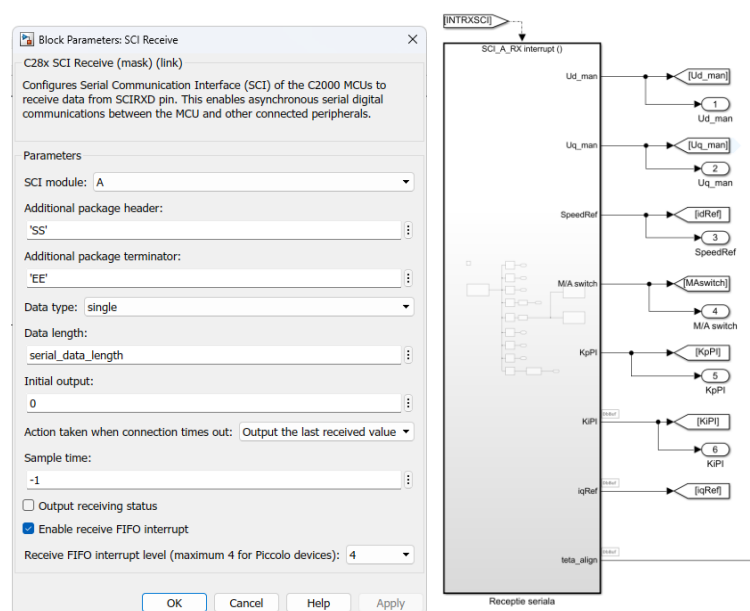


Fig. 4.5.1 Subsistemul recepției seriale

În cazul acesta blocul recepționează 9 parametri, conform figurii 4.5.2: Ud\_man, Uq\_man, SpeedRef, M/A switch, EN\_GATE Switch, KpPI, KiPI, iqRef și teta\_align.

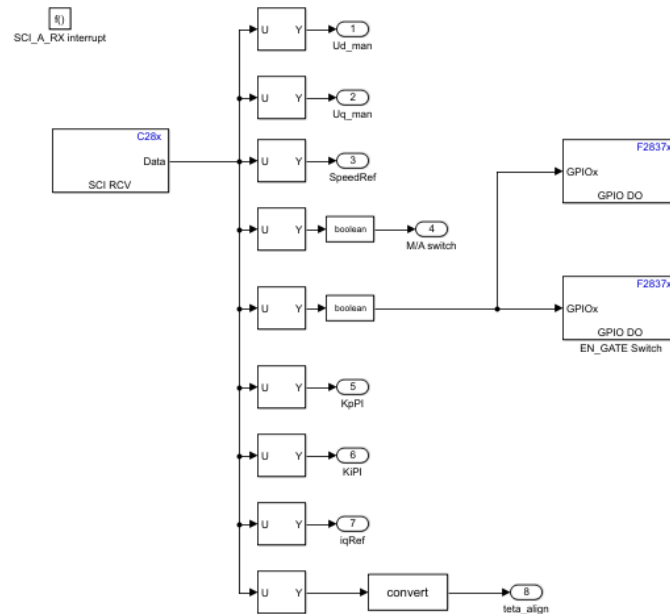


Fig. 4.5.2 Subsistemul recepției seriale detaliat

## 4.6 Schema Simulink PC\_host\_SCI\_interface\_PI

Pentru a realiza transmisia și recepția tuturor parametrilor necesari procesului, este utilizată o altă schemă Simulink denumită PC\_host\_SCI\_interface\_PI. Aceasta permite schimbarea parametrilor în timpul rulării și, astfel, obținerea acestora.

Această schemă este configurată pentru a gestiona comunicația între PC și placa de dezvoltare LAUNCHXL - F28379D. Ea conține blocuri pentru transmiterea și recepționarea datelor prin interfața de comunicație serială (SCI).

Componentele principale ale schemei:

- blocul de transmisie
- blocul de recepție



### 4.6.1 Modul de transmisie

Sistemul utilizează un canal de comunicare serială pentru a trimite date de la PC către un microcontroler. Aceasta se realizează prin blocuri specifice de transmisie în Simulink care codifică și trimite datele în format serial.

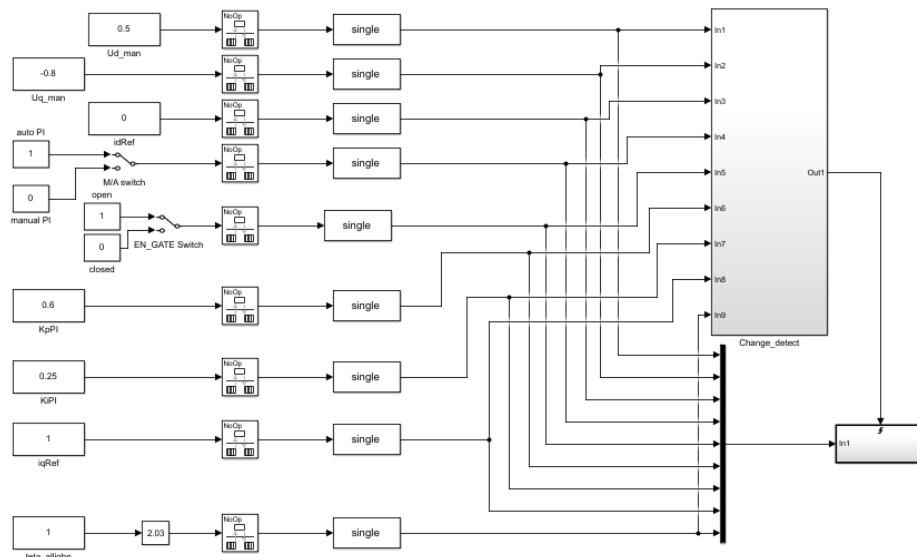


Fig. 4.6.1 Subsistemul de transmisie al PC\_host\_SCI\_interface\_PI

Schema permite utilizatorului să modifice parametrii sistemului în timp real prin intermediul interfeței. O modificare adusă unui parametru este transmisă imediat către dispozitivul embedded. Aceasta permite ajustarea comportamentului sistemului fără a opri și reporni instalația.

După cum se observă în figura 4.6.1, aplicația poate fi rulată atât în mod manual, cât și în mod automat. În modul manual, trebuie să se seteze tensiunea  $U_q$ , în timp ce în modul automat, comanda  $U$  este preluată de controlerul PI. De asemenea, referința poate fi modificată în timpul funcționării pentru a evalua eficiența controlerului.

### 4.6.2 Modul de recepție

În mod similar transmisiei, dispozitivul embedded poate trimite date înapoi către PC. Blocurile de recepție în Simulink decodează aceste date și le fac disponibile pentru monitorizarea în timp real sau pentru prelucrarea ulterioară în model.

Prin monitorizarea parametrilor în timp real, se potate evalua imediat impactul modificărilor asupra sistemului.

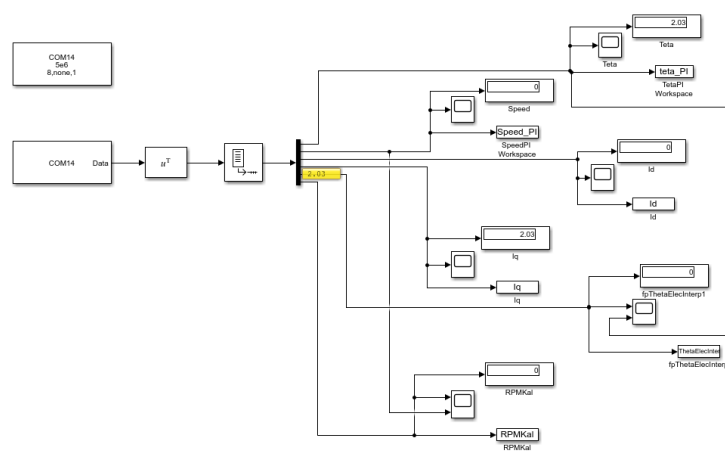


Fig.4.6.2 Subsistemul de recepție al PC\_host\_SCI\_interface\_PI

Pe partea de recepție, sunt preluate semnalele din schema anterioară, inclusiv teta, Id, Iq, viteza RPMSPdFil, unghiul tetaKalman estimat si viteza estimată RPMKal(după aplicarea filtrului Kalman). Prin utilizarea blocurilor To Workspace, aceste semnale sunt preluate, plotate și analizate pentru a evalua performanțele obținute. Acest proces va fi detaliat în capitolul următor, unde vor fi prezentate în detaliu rezultatele simulării.

## 4.7 Prelucrarea comenzii controllerelor

Figura 4.7.1 ilustrează schema Simulink a unui sistem de control pentru un motor electric, utilizând tehnici de control vectorial (Field Oriented Control - FOC). Schema a fost proiectată pentru a funcționa atât în mod automat, cât și în mod manual, oferind o mare flexibilitate sistemului. Aceasta conține un switch configurat astfel: 1 – automat, 0 – manual.

Tot la acest switch este conectat si un bloc enable care oprește funcționarea în fundal a controller-ului atunci când este selectat modul manual. Când se face trecerea de la modul manual la modul automat, este generată o comandă optimă pentru reglarea vitezei motorului.

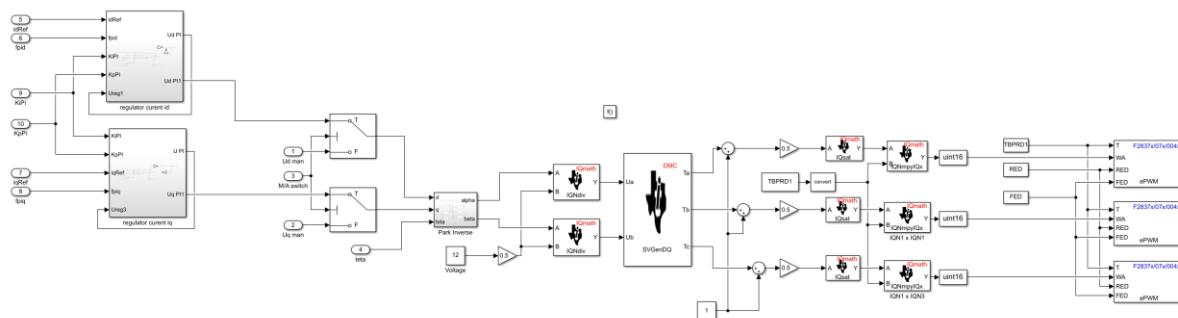


Fig. 4.7.1 Prelucrarea comenzii controllerelor

Blocurile IQmath din Simulink, dezvoltate de Texas Instruments, sunt create pentru operații matematice eficiente pe DSP-uri și microcontrolere TI. Ele sunt optimizate pentru aritmetica pe virgulă fixă și includ funcții matematice avansate. Aceste blocuri ajută la dezvoltarea și implementarea rapidă a algoritmilor matematici în aplicații embedded.

### Subsistemul pentru transformarea inversă Park.

Mărimile din cadrul de referință rotativ sunt transformate într-un cadru de referință staționar ortogonal cu două axe folosind transformarea inversă Park, așa cum este prezentat în Figura 4.7.2.

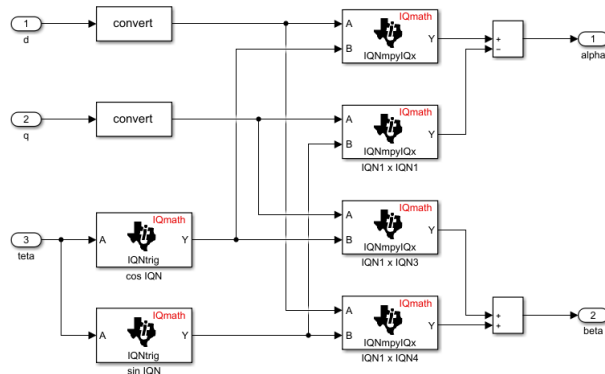


Fig. 4.7.2 Subsistemul pentru transformarea inversă Park.

Această transformare permite convertirea componentelor din cadrul de referință rotativ ( $d$  și  $q$ ) în componentele din cadrul de referință staționar ortogonal ( $\alpha$  și  $\beta$ ). Transformarea inversă Park este importantă pentru analiza și controlul mașinilor electrice, deoarece ajută la gestionarea mai eficientă a mărimilor care variază în timp.

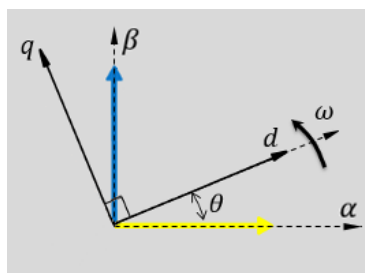


Fig. 4.7.3 Reprezentare combinată a vectorilor

Ecuatiile transformării inverse Park sunt date de următoarele relații:

$$V_{\alpha} = V_d * \cos(\theta) - V_q * \sin(\theta) \quad (4.7.1)$$

$$V_{\beta} = V_q * \cos(\theta) + V_d * \sin(\theta) \quad (4.7.2)$$

Unde,

$V_{\alpha}, V_{\beta}$  sunt mărimi din cadrul de referință staționar ortogonal,

$V_d, V_q$  sunt mărimi din cadrul de referință rotativ.

Valorile alpha și beta sunt convertite cu ajutorul blocurilor “IQN / IQN” din biblioteca “IQmath”, ca numere pe 32 de biți în virgulă fixă, având același factor Q (puterea la care este ridicat numărul 2 pentru poziționarea virgulei în reprezentarea numerelor în virgulă fixă). Astfel, aceste valori în virgulă fixă vor putea fi utilizate de blocul Space Vector Generator (SVGenDQ).

Blocul Space Vector Generator este utilizat pentru a genera semnale de comandă pentru motorul BLDC. Principala funcție a acestui bloc este să convertească semnalele de comandă date în semnale trifazate necesare pentru a controla modulația PWM. Această funcție este esențială în cadrul acestui proiect pentru a crea formele de undă de curent necesare pentru a controla eficient motorul BLDC prin intermediul unui invertor trifazat.

Invertorul trifazat funcționează prin comutarea rapidă a tranzistoarelor MOSFET (denumite A+, A-, B+, B-, C+ și C-) pentru a genera tensiuni alternative trifazate la ieșire. Aceste comutatoare sunt controlate de un circuit de comandă care reglează momentul și durata de comutare a fiecărui comutator în funcție de semnalele de comandă primite. Schema electrică a invertorului este prezentată în figura 4.7.4. Fiecare pereche de comutatoare (de exemplu, A+ și A-) controlează o fază a tensiunii de ieșire și sunt conectate într-o configurație tipică de punte H pentru fiecare fază.

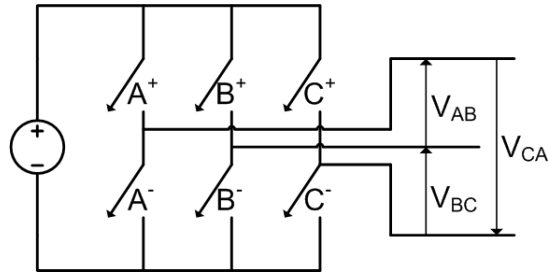


Fig. 4.7.4 Schema intertorului trifazat

Pentru a genera tensiunea alternativă trifazată la ieșire, invertorul comută rapid între stările ON și OFF ale fiecărui comutator într-o secvență controlată. De exemplu, pentru a genera o fază A cu tensiune pozitivă, comutatorul A+ este în starea ON (conducție) și comutatorul A- este în starea OFF (neconducție), permițând curentului să curgă prin comutatorul A+ și astfel creând o tensiune pozitivă în faza A. Aceasta conduce la opt vectori de comutare posibili pentru invertor, de la V0 la V7, dintre care șase sunt vectori de comutare activi și doi sunt vectori zero. Prin controlul secvenței de comutare a comutatoarelor, invertorul creează o tensiune trifazată alternativă la ieșire.

Vector	A+	B+	C+	A-	B-	C-	$V_{AB}$	$V_{BC}$	$V_{CA}$	
$V_0 = \{000\}$	OFF	OFF	OFF	ON	ON	ON	0	0	0	Zero vector
$V_1 = \{100\}$	ON	OFF	OFF	OFF	ON	ON	$+V_{dc}$	0	$-V_{dc}$	Active vector
$V_2 = \{110\}$	ON	ON	OFF	OFF	OFF	ON	0	$+V_{dc}$	$-V_{dc}$	Active vector
$V_3 = \{010\}$	OFF	ON	OFF	ON	OFF	ON	$-V_{dc}$	$+V_{dc}$	0	Active vector
$V_4 = \{011\}$	OFF	ON	ON	ON	OFF	OFF	$-V_{dc}$	0	$+V_{dc}$	Active vector
$V_5 = \{001\}$	OFF	OFF	ON	ON	ON	OFF	0	$-V_{dc}$	$+V_{dc}$	Active vector
$V_6 = \{101\}$	ON	OFF	ON	OFF	ON	OFF	$+V_{dc}$	$-V_{dc}$	0	Active vector
$V_7 = \{111\}$	ON	ON	ON	OFF	OFF	OFF	0	0	0	Zero vector

Tabel 4.7.1 Vectorii de comutare ai invertorului

Pentru implementare, un semnal de tensiune de referință este eșantionat cu o frecvență  $F_{PWM}$  ( $T_{PWM} = 1/F_{PWM}$ ). Vectorul de referință este apoi compus folosind o combinație a celor doi vectori de comutare activi adiacenți și unul sau ambii vectori zero. Ulterior, sunt generate semnale PWM prin intermediul blocurilor ePWM, utilizând perioada TBPRD1,

semnalul eșantionat anterior și registrele RED și FED, respectiv întârzierea la creșterea semnalului (rising edge delay) și întârzierea la descreșterea semnalului (falling edge delay).

Astfel, cu RED, ieșirea urmează valoarea maximă a intrării după perioada de întârziere și rămâne ridicată până când intrarea scade, iar cu FED, ieșirea urmează valoarea minimă a intrării după perioada de întârziere și rămâne scăzută până când intrarea crește. Blocurile ePWM utilizează PWM1A, 2A și 3A, iar conform configurației plăcii, folosesc GPIO0, GPIO2 și GPIO4.

Toți cei opt vectori de comutare posibili pentru un invertor trifazat care utilizează modulația vectorială spațială sunt prezentați în figura 4.7.5. Un exemplu de tensiune de referință  $V_{ref\_MAX}$  este prezentat în primul sector,  $V_{ref\_MAX}$  fiind amplitudinea maximă a  $V_{ref}$  înainte de a se atinge supramodularea neliniară.

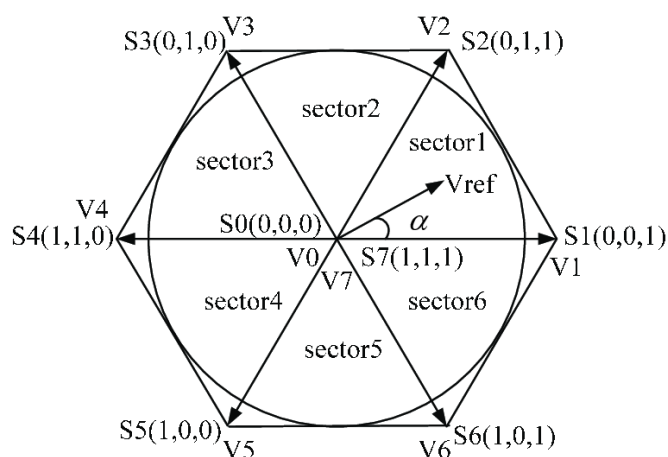


Fig. 4.7.5 Vectorii de comutare ai invertorului reprezentați grafic

Cei șase vectori non-zero ( $v_1$  până la  $v_6$ ) formează axele unui hexagon regulat cu șase sectoare egale (fiecare dintre doi vectori non-zero adiacenți fiind separați de un unghi egal cu  $60^\circ$ ) și doi vectori zero ( $v_0$  și  $v_7$ ) situați în centrul hexagonului. Notăția binară de pe axele hexagonului indică starea de comutare a brațelor invertorului. În acest caz, 1 logic înseamnă că întrerupătorul superior ( $A+$ ,  $B+$  sau  $C+$ ) este pornit, iar 0 logic înseamnă că întrerupătorul inferior ( $A-$ ,  $B-$  sau  $C-$ ) al brațului este pornit. Bitul cel mai semnificativ în notația logică este pentru brațul A, bitul din mijloc este pentru brațul B, iar cel mai puțin semnificativ este pentru brațul C.

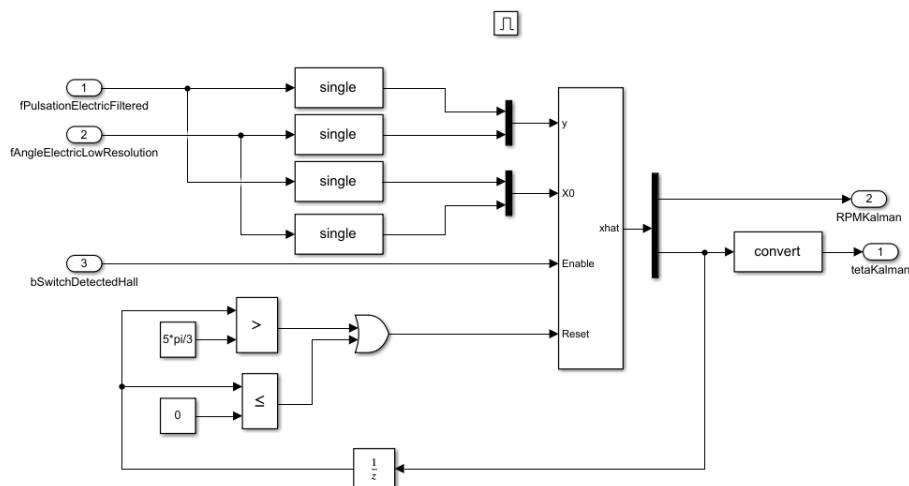


Fig.4.7.6 Subsistemul Filtrului Kalman

## 4.8 Filtrul Kalman

În trecut, au fost propuse multe operațiuni de control fără senzori pentru a reduce costurile ridicate și slaba detectare a poziției pentru motoarele BLDC, inclusiv detectarea tensiunii electromotoare inversă (back-EMF), integrarea back-EMF, estimarea fluxului și tehnicile de modificare a motorului. Din păcate, niciuna dintre aceste tehnici nu a funcționat bine la diferite viteze de operare și există probleme de fiabilitate și complexitate asociate cu ele.

Aceste metode nu sunt capabile să ofere o estimare continuă a poziției rotorului, așa că pentru o precizie ridicată a estimării este necesară informația continuă a poziției rotorului, nu doar la fiecare 60 de grade electrice. Metodele descrise au multe beneficii și pot fi aplicate pentru estimarea poziției rotorului. Cu toate acestea, toate aceste metode prezintă unele probleme: performanța inefficientă la viteze și există o altă deficiență legată de variabilitatea intervalelor de viteză care necesită îmbunătățiri. Majoritatea acestor metode prezintă neliniarități și perturbații în obiectele de control mecanic. Din cauza acestor limitări, aceste metode nu sunt aplicate frecvent în prezent, având unele dezavantaje asociate. Pentru a depăși toate aceste limitări, ar trebui aplicate alte metode. Una dintre aceste metode, care va fi descrisă în detaliu în această cercetare, este Filtrul Kalman Extins.

Această tehnică va fi implementată în timp real. Variabilele pot fi estimate utilizând tensiunile de linie ale statorului și curenții de fază. Parametrii considerați pentru algoritmul de estimare au fost descriși, iar ceilalți parametri care urmează să fie estimați sunt rezistența

statorului, care poate fi aproximată simultan prin variabilele de stare ale motorului. Filtrul Kalman Extins este o tehnică iterativă pentru sistemele neliniare perturbate de zgomot. Acesta utilizează un algoritm în două etape: în prima etapă se prezice estimarea stării și estimarea covarianței erorii pentru a obține o estimare precisă a stării la momentul prezent. În etapa următoare, estimarea precisă a stării este corectată utilizând măsurătorile recente ale procesului pentru a genera o estimare actualizată a stării. Această tehnică este o metodă de estimare pentru a aproxima viteza și poziția rotorului.

Estimarea poate fi realizată prin aplicarea Filtrului Kalman Extins. În primul rând, în timpul aproximării, semnalele de curent și tensiune nu vor fi filtrate, așa cum se întâmplă în alte metode, unde acestea sunt de obicei filtrate. Parametrii care necesită aproximare sunt, în esență, de două tipuri: unul este viteza și poziția rotorului, care sunt estimate împreună cu parametrii motorului, iar rezistența statorului poate fi aproximată utilizând variabilele de stare ale motorului. Din rezultatele prezentate la sfârșitul cercetării, va fi destul de convenabil pentru motorul BLDC să aproximeze poziția și viteza rotorului cu mare precizie în ambele stări, atât în regim staționar, cât și în regim dinamic.

Prin utilizarea rezistenței statorului, precizia aproximării va fi mai mare, în special la viteze mici. Această tehnică realizează estimarea optimă a stării pentru sisteme neliniare dinamice prin aplicarea semnalelor care sunt perturbate de zgomot. Tehnica Filtrului Kalman Extins este utilizată pentru aproximarea parametrilor necunoscuți, cum ar fi rezistența statorului  $R_r$  și estimarea parametrilor. Observatorul care este determinist și nu are zgomot este observatorul Luenberger și poate fi aplicat sistemelor liniare invariabile în timp

În mod normal, Filtrul Kalman este aplicat sistemelor liniare, iar pentru sistemele neliniare se aplică Filtrul Kalman Extins. Modelul mașinii este prezentat mai jos.

$$\frac{dx}{dt} = Ax + BV_s \quad (4.8.1)$$

$$x = [i_a \ i_b \ i_c \ \omega \ \theta \ R_s]^T \quad (4.8.2)$$

$$V_s = [V_{ab} \ V_{bc} \ V_{ca}]^T \quad (4.8.3)$$

Unde:

-  $i_a \ i_b \ i_c$  = Curenții de fază

-  $V_{ab} \ V_{bc} \ V_{ca}$  = Tensiuni linie-la-linie

-  $V_s$  = Vectorul de intrare



$$A = \begin{bmatrix} 1 - \frac{R_s T}{L_s} & 0 & \frac{L_m R_r}{\theta L_r L_s^2} & \frac{L_m \omega_r}{\theta L_r L_r} & 0 & 0 \\ 0 & 1 - \frac{R_s T}{L_s} & \frac{-L_m R_r}{\theta L_r L_r} & \frac{L_m R_r}{\theta L_r L_s^2} & 0 & 0 \\ \frac{L_m R_r}{L_r L_r} & 0 & 1 - \frac{R_s T}{L_s} & -\omega_r & 0 & 0 \end{bmatrix} \quad (4.8.4)$$

$$B = \begin{bmatrix} \frac{T}{3L_s} & \frac{-T}{3L_s} & 0 \\ 0 & \frac{T}{3L_s} & \frac{-T}{3L_s} \\ \frac{T}{3L_s} & 0 & \frac{-T}{3L_s} \end{bmatrix} \quad (4.8.5)$$

Descrierea Variabilelor:

- $L_m$  = Inductanța de magnetizare
- $R_r$  = Rezistența rotorului
- $R_s$  = Rezistența statorului
- $\theta$  = Poziția rotorului
- $\omega_r$  = Viteza rotorului
- $L_r$  = Inductanța rotorului

### Modelul matematic al filtrului kalman extins

Filtrul Kalman utilizează un model dinamic al mașinii în care viteza poate fi considerată atât o stare, cât și un parametru.

$$Y = CX \quad (4.8.6)$$

$$x = [i_a \ i_b \ i_c \ \omega \ \theta \ R_s]^T \quad (4.8.7)$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (4.8.8)$$

$$Y = [i_a \ i_b \ i_c]^T \quad (4.8.9)$$

## Implementarea digitală a filtrului kalman extins

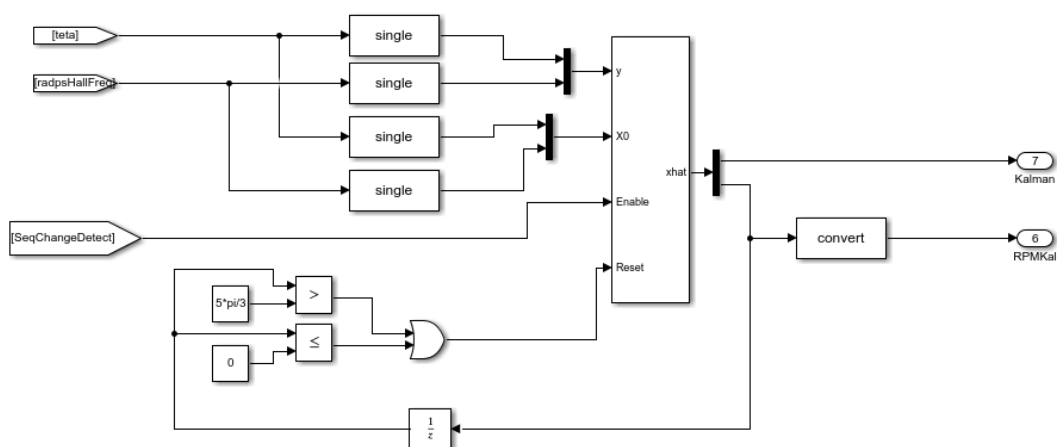


Fig. 4.8.1 Subisitemul filtrului Kalman

Pentru a implementa filtrul Kalman în mod digital, modelul trebuie discretizat, iar în model există și câteva surse de zgomot, notate cu  $V$  și  $W$ , astfel încât modelul în formă discretă poate fi prezentat ca:

$$X(k+1) = A_d X(k) + B_d U(k) + V(k) \quad (4.8.10)$$

$$Y(k) = C_d X(k) + W(k) \quad (4.8.11)$$

Variabilele pot fi considerate independente una de cealaltă. Zgomotul și statisticile măsurătorilor pot fi descrise prin matricele de covarianță  $P$ ,  $Q$  și  $R$ . Proiectarea EKF implică unele expresii care sunt computaționale.

Include două etape: predicție și corecție. În prima etapă, valorile prezise  $(k+1)$  sunt găsite prin modelul mașinii și ultimele valori ale stărilor care sunt estimate. Etapa de filtrare începe de la vectorul  $Q$ , în care starea estimată  $X'(K+1)$  poate fi găsită din ultima stare de predicție în care poate exista o adăugare a termenului care este  $e$ , unde  $e = Y(K+1) - Y'(K+1)$  și câștigul Kalman. Acest câștig poate fi utilizat pentru a optimiza erorile în estimarea stării. Calculele EKF sunt realizate în așa fel încât termenul de corecție  $e$  să se apropie de zero. Un rezumat complet al proiectării EKF cuprinde următorii șase pași.

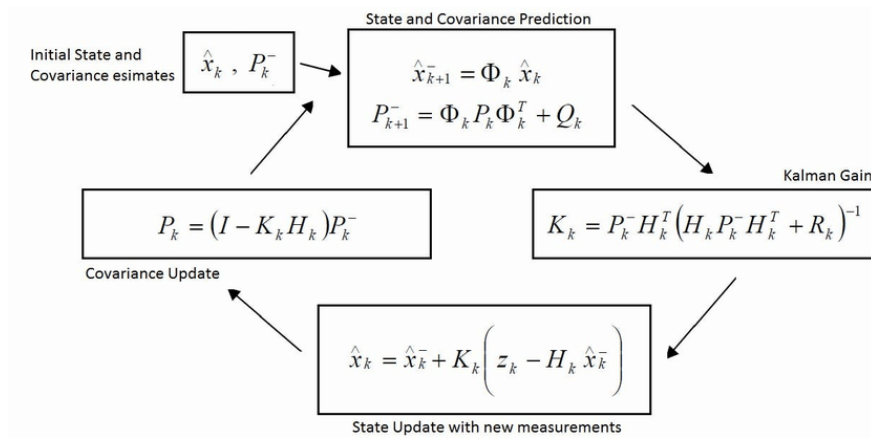


Fig. 4.8.2 Ecuatiile Recursive ale Filtrului Kalman

Prima etapă este reprezentată de inițializarea vectorului de stare și a altor matrici

$$X(0), Q_0, R_0, P_0$$

Se calculează predicția pentru starea sistemului la momentul k+1 folosind modelul de proces:

$$X(k+1, k) = X(k+1) = A_d X(k) + B_d U(k) \quad (4.8.11)$$

Urmează calcularea matricei de covarianță a erorii de predicție:

$$P(k+1) = f(k+1)P(k)f^T(k+1) + Q \quad (4.8.12)$$

Câștigul Kalman, care ajută la actualizarea estimării stării:

$$K(k+1) = P(k+1)h(k+1)[h(k+1)P(k+1)h^T(k+1) + R]^{-1} \quad (4.8.13)$$

Se actualizează estimarea stării folosind câștigul Kalman și măsurătorile:

$$X'(K+1, K) = X(k+1, k) + K(k+1)[Y(k+1)] \quad (4.8.14)$$

Se actualizează matricea de covarianță a erorii de estimare:

$$P^{(K+1)} = P(K+1) - K(k+1)h(k+1)P(k+1) \quad (4.8.15)$$

Pentru a obține modelul motorului BLDC prin Filtrul Kalman Extins, simularea este realizată în Simulink. Pe lângă funcționarea normală a controlului vitezei motorului BLDC, fluxul generat în stator și valorile aproximative ale stării mecanice sunt obținute prin Filtrul Kalman Extins. Filtrul Kalman Extins va fi folosit pentru a obține o estimare a valorilor fluxului

în stator, a vitezei rotorului și a poziției fluxului statorului prin valorile calculate ale curentului și tensiunii motorului, care sunt apoi date înapoi la porțile logice de comutare.

Principalul avantaj al filtrului Kalman este că este un observator de stare care poate oferi cea mai bună estimare prin minimizarea erorii pătratice pentru sistemul unei variabile de stare, care poate fi aplicat atât la ieșirea, cât și la intrarea perturbațiilor aleatorii. Modelul mașinii pentru a obține o estimare a vitezei motorului fără utilizarea Filtrului Kalman Extins este prezentat mai jos.

## 4.9 Regulatorul PI

Pentru controlul PI (Proportional-Integral) și proiectarea unui controler, se pornește de la ecuația erorii, care este diferența dintre valoarea dorită și valoarea măsurată:

$$U(s) = K_p * E(s) + K_i * \frac{1}{s} * E(s) \quad (4.9.1)$$

Scotând factor comun  $E(s)$ , rezultă

$$U(s) = E(s) * \left( K_p + K_i * \frac{1}{s} \right) \quad (4.9.2)$$

Transferând termenul  $E(s)$  în partea stângă a ecuației (4.9.2), se obține:

$$\frac{U(s)}{E(s)} = \left( K_p + K_i * \frac{1}{s} \right) \quad (4.9.3)$$

În urma aplicării metodei dreptunghiului înapoi  $\left( s \leftarrow \frac{1-z^{-1}}{T_s} \right)$ , ecuația devine

$$\frac{U(z^{-1})}{E(z^{-1})} = \left( K_p + K_i * \frac{T_s}{1-z^{-1}} \right) \quad (4.9.4)$$

Aducem la numitor comun membrul drept

$$\frac{U(z^{-1})}{E(z^{-1})} = \left( \frac{K_p - K_p z^{-1} + K_i T_s}{1-z^{-1}} \right) \quad (4.9.5)$$

Dupa ce facem produsul pe diagonale, obținem ecuația

$$U(z^{-1}) * (1 - z^{-1}) = E(z^{-1}) * (K_p - K_p z^{-1} + K_i T_s) \quad (4.9.6)$$

Din care va rezulta o ecuație cu diferențe de forma

$$U(k) - U(k-1) = E(k) * K_p - E(k-1) * K_p + E(k) * K_i T_s \quad (4.9.7)$$

După modelarea ecuației, obținem comanda regulatorului PI în formă discretă

$$U(k) = E(k) * (K_p + K_i T_s) - E(k-1) * K_p - U(k-1) \quad (4.9.8)$$

Aceasta a fost integrată în Simulink într-un subsistem dedicat controlorului PI, după cum poate fi observat în figura 4.9.1.

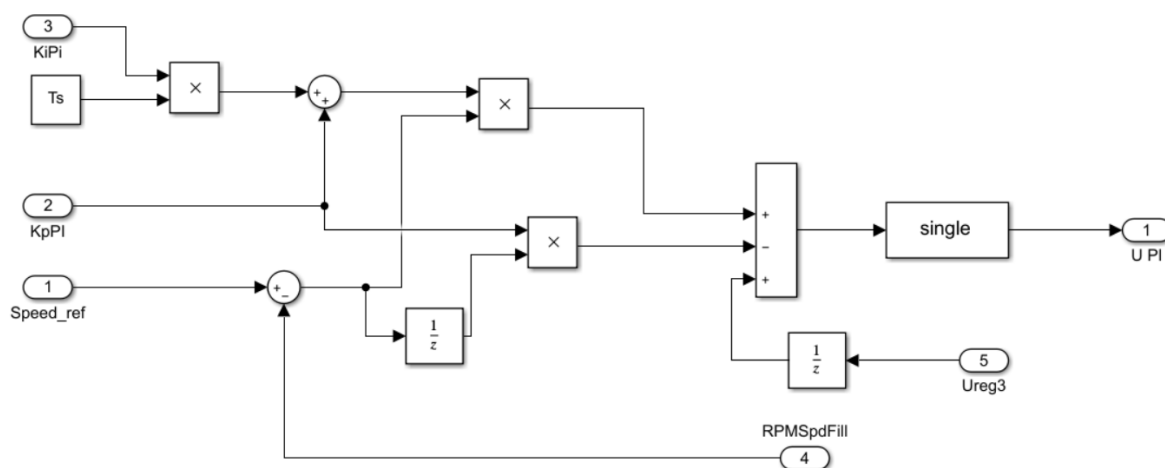


Fig 4.9.1 Subsistemul regulatorului PI

## CAPITOLUL 5

### REZULTATE EXPERIMENTALE

În cadrul acestui capitol, vom prezenta și analiza rezultatele experimentale obținute în urma implementării și controlului motorului BLDC (Brushless DC Motor). Scopul principal al acestui capitol este de a evalua performanța sistemului de control dezvoltat și de a verifica dacă obiectivele inițiale ale proiectului au fost atinse.

Motorul BLDC a fost ales datorită avantajelor sale semnificative, precum eficiența ridicată, durabilitatea și capacitatea de a furniza un cuplu constant la diverse viteze. Implementarea controlului unui astfel de motor implică utilizarea unor tehnici avansate de control și a unor algoritmi specifici care să asigure o funcționare optimă.

Pentru obținerea parametrilor optimi ai controllerului a folosit modelul Simulink Controller\_PI. Acest model conține o funcție de transfer de ordin I a procesului, obținută prin

identificare. Modelul este creat pentru a simplifica și facilita ajustarea controlerelor, oferind o bază pentru optimizarea parametrilor sistemului. Aceasta permite o calibrare mai eficientă și precisă a controlerelor.

În procesul descris, a fost generat un model de funcție de transfer utilizând date de la sistem în modul manual, de unde a fost extrasă comanda manuală  $U_q$  și ieșirea procesului (viteza motorului) prin interfața SCI și salvată în MATLAB folosind blocuri To Workspace fiind apoi prelucrate pentru a le folosi în toolbox-ul System Identification. După ce datele au fost convertite la tipul double (pentru a putea fi aplicate în toolbox), din interfață s-a selectat Transfer Function Model, rezultând funcția de transfer:

$$Tf = \frac{148.5}{0.2s+1} \quad (5.1)$$

## 5.1 Rezultate control PI – controller PI

Pentru a calibra controller-ul PI, a fost utilizată o schemă de test denumită "test\_PI". În acest context, s-a folosit funcția de transfer definită în relația (5.1) și structura de control PI descrisă în subcapitolul anterior. Simularea a fost efectuată cu o perioadă de eșantionare  $T_s = \frac{1}{40000}$  secunde, ceea ce corespunde unei frecvențe de eșantionare de 40 kHz.

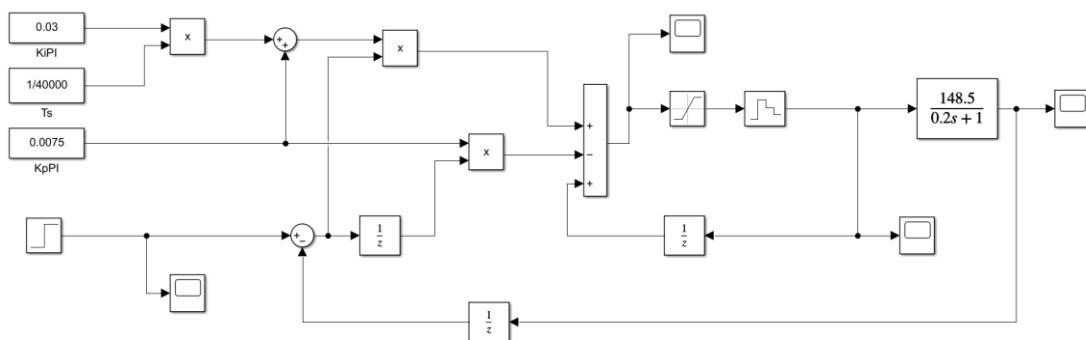


Fig.5.1.1 Acordare regulatorului PI

După efectuarea unui număr semnificativ de simulări atât pe baza schemei din figura 5.1.1, cât și pe procesul real, s-au stabilit parametrii finali:  $K_p = 0.0075$  și  $K_i = 0.03$ . În timpul experimentelor pe procesul real, acești parametri au fost transmiși în timp real printr-o

conexiune serială. Acest lucru a permis monitorizarea directă a comportamentului motorului, observându-se cum acesta respectă viteza impusă și cum se manifestă oscilațiile. Rezultatele acestor observații sunt prezentate în figura 5.1.2.

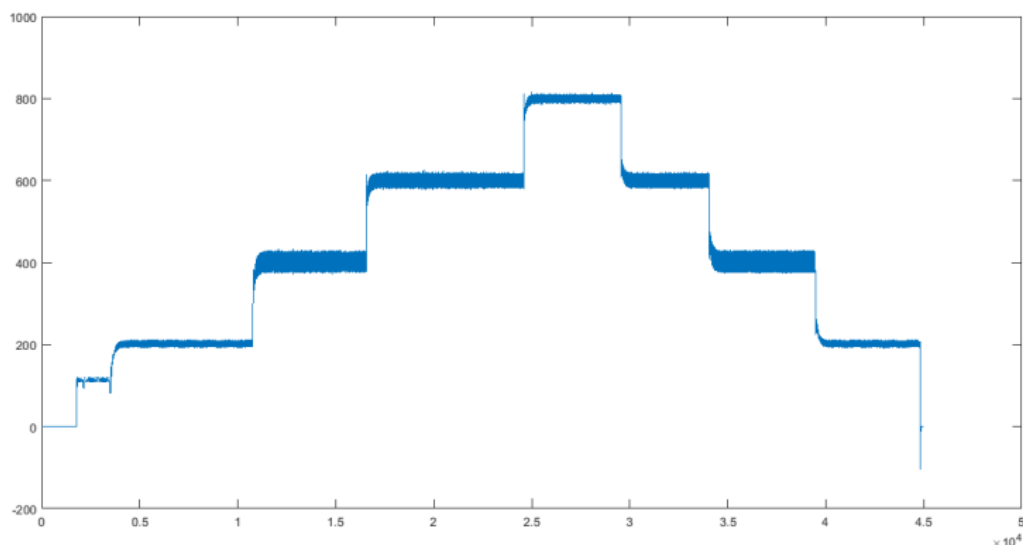


Fig. 5.1.2 Răspunsul în viteză al motorului BLDC cu control PI

Inițial, motorul BLDC a fost setat să atingă viteze de 200, 400, 600 și 800 RPM succesiv, urmând ca apoi să fie scăzute treptat în ordinea inversă a setării inițiale. Deși controlerul PI a reușit să urmărească aceste valori de referință, au fost observate oscilații notabile în răspunsul motorului. Pentru a îmbunătăți performanța sistemului și a reduce aceste oscilații, în subcapitolul 5.2 va fi explorată o metodă avansată de control bazată pe filtrul Kalman.

## 5.2 Rezultatele filtrului Kalman extins

Pentru a valida eficacitatea filtrului Kalman extins (EKF) în controlul motorului BLDC (Brushless DC), au fost efectuate o serie de experimente practice și simulări. Rezultatele obținute demonstrează capacitatea EKF de a îmbunătăți precizia estimărilor de stare ale sistemului și de a reduce zgomotul în măsurători.



Fig. 5.2.2 Viteza estimată

Filtrul Kalman extins a fost implementat pentru estimarea stărilor critice ale motorului BLDC, inclusiv poziția rotorului și viteza unghiulară. Pentru a evalua performanța EKF, estimările generate au fost comparate cu valorile reale obținute din senzorii de poziție



Fig. 5.2.2 Unghiul  $\theta$  estimat

Rezultatele au arătat că EKF oferă estimări cu o precizie ridicată, având o eroare medie pătratică (RMSE) mult redusă în comparație cu metodele tradiționale de filtrare.

Rezultatele experimentale și de simulare confirmă faptul că filtrul Kalman extins este o metodă eficientă pentru controlul motorului BLDC. Prin îmbunătățirea preciziei estimărilor de stare și reducerea zgomotului de măsurare, EKF contribuie la o performanță superioară a sistemului de control.



## 6. Concluzii

În cadrul acestei lucrări de licență, am investigat și implementat controlul unui motor cu curent continuu fără perii (BLDC Nanotec DF45L024048-A2) utilizând Matlab/Simulink, un procesor digital de semnal (DSP TMS320F28379D) și a inverterului BOOSTXL-DRV8305EVM. Obiectivul principal a fost de a proiecta și valida o metodologie eficientă pentru controlul vitezei și poziției motorului BLDC, folosind senzori Hall pentru feedback și algoritmi de control avansați, inclusiv reglatoare PI și filtrul Kalman.

În cadrul primului capitol au fost prezentate noțiunile teoretice esențiale legate de motoarele fără perii (BLDC), placa de dezvoltare, inverter și montajul realizat. Capitolul al doilea a fost dedicat studiului sistemelor neliniare, în timp ce capitolele trei și patru au detaliat diversele metode de control aplicate. Acest ansamblu de informații asigură o înțelegere clară a aplicației dezvoltate și facilitează implementarea ei practică.

Structura logică și progresivă a lucrării permite cititorului să acumuleze cunoștințele necesare pentru a înțelege și controla eficient un motor BLDC. Lucrarea începe cu fundamentele teoretice și continuă cu analiza sistemelor neliniare, oferind astfel o imagine clară asupra complexității acestor sisteme.

Utilizarea Filtrului Kalman Extins pentru a modela un motor BLDC în Simulink permite obținerea unor estimări precise ale fluxului statorului și vitezei rotorului. Aceste estimări sunt esențiale pentru controlul eficient și precis al motorului BLDC, îmbunătățind performanțele generale ale sistemului de control al vitezei motorului.

Studiul poate fi extins prin integrarea algoritmilor de control adaptiv și robust, care să îmbunătățească și mai mult performanța în prezența perturbărilor și variațiilor parametrilor de sistem. De asemenea, utilizarea unor senzori mai avansați și tehnici de estimare a stării ar putea duce la o și mai mare precizie și fiabilitate.

În concluzie, această lucrare demonstrează eficiența utilizării reglatoarelor PI și a filtrului Kalman pentru controlul motoarelor BLDC, deschizând calea pentru inovații viitoare în controlul motoarelor electrice.

## 7. Bibliografie

1. Brushless DC Motor Fundamentals Application Note-Prepared by Jian Zhao/Yangwei Yu July 2011
2. Commutation of a BLDC Motor with Various Pole-Pairs Can Be Simplified By Using Absolute Encoder Feedback White Paper Author Sithambaram Palaniappan Avago Technologies, Malaysia
3. ELECTRONICALLY COMMUTATED MOTORS (ECMS): How are they are different from regu... Arhivat în 10 mai 2011, la Wayback Machine.
4. Bequette, B. Wayne (2003). Process Control: Modeling, Design, and Simulation. Upper Saddle River, New Jersey: Prentice Hall.
5. Motor Unit Conversions – What They Mean and How They are Used REV210209
6. <https://www.nanotec.com/eu/en/products/1787-df451024048-a2>
7. LAUNCHXL-F28379D Overview User's Guide Literature Number: SPRUI77 August 2016
8. User's Guide SLVUAI8A–August 2015–Revised June 2017 BOOSTXL-DRV8305EVM
9. Analytical Design of PI Controller for First Order Transfer Function Plus Time Delay: Stability Triangle Approach
10. Cooper, Douglas. "PI Control of the Heat Exchanger". Practical Process Control by Control Guru. Retrieved 2014-02-27.
11. Hills, Richard L (1996), Power From the Wind, Cambridge University Press
12. W. C. Duesterhoeft; Max W. Schulz; Edith Clarke (July 1951). "Determination of Instantaneous Currents and Voltages by Means of Alpha, Beta, and Zero Components". Transactions of the American Institute of Electrical Engineers.
13. Park, Inverse Park and Clarke, Inverse Clarke Transformations MSS Software Implementation User Guide -Microsemi
14. Chui, Charles K.; Chen, Guanrong (2009). Kalman Filtering with Real-Time Applications. Springer Series in Information Sciences. Vol. 17 (4th ed.). New York
15. Kalman, R.E.; Bucy, R.S. (1961). "New Results in Linear Filtering and Prediction Theory". Journal of Basic Engineering. 83: 95–108.

## 8. Anexe

### Codul principal Matlab

```
clc
clear all
close all

F = 200*10^6; %[Hz] system clock
TTBCLK = 1/F;
TiSamplePhaCrt = 10^-4;
Tf_spd = 0.01;

% Inverter PWM config.
F_PWM = 20000; %[Hz]

% the time-base clock (TBCLK) is a prescaled version of the system
% clock (SYSCLKOUT ---- For Up and Down Count ---- TPWM = 2*TBPRD*TTBCLK;
TPWM = 1/F_PWM;
TBPRD1 = uint16(TPWM/(2*TTBCLK));
RED = uint16(50); %rising edge delay
FED = uint16(50); %falling edge delay

% Tasks sampling time
T_pos = 5*10^-4;
Ts_crt_control = 1/30000;
%Tf_crt_control= 1/10000;
Ts_spd_control = 1/10000;
Ts = Ts_crt_control;

% SCI config
serial_data_length = 10;
h = 0.01;
Tf = 0.015;

% Interrupt table config
ECAP1_CPU = 4;
```

```

ECAP1_PIE = 1;
SCIA_RX_CPU = 9;
SCIA_RX_PIE = 1;
INT_CPU = [ECAP1_CPU, SCIA_RX_CPU];
INT_PIE = [ ECAP1_PIE, SCIA_RX_PIE];

% Motor control config.
pp = 16/2; %perechi poli
MinDetectSpeed_RPM = 40; %RPM
MaxDetectSpeed_RPM = 2000; %RPM
MinDetectSpeed_radps = pp*MinDetectSpeed_RPM*pi/30;
MaxDetectSpeed_radps = pp*MaxDetectSpeed_RPM*pi/30;
MinDetectSpeed_period = 1/(MinDetectSpeed_radps/(2*pi));
MaxDetectSpeed_period = 1/(MaxDetectSpeed_radps/(2*pi));
MaxDetectSpeed_Freq = 1/MaxDetectSpeed_period;
MinDetectSpeed_Freq = 1/MinDetectSpeed_period;
Min_eCAP = (MaxDetectSpeed_period/TTBCLK);
Max_eCAP = (MinDetectSpeed_period/TTBCLK);
MaxIter_1_ms_eCAP_wait = MinDetectSpeed_period/T_pos;

vref=3.3;
xT=1.65;
% 12bits ADC=4095
%vref.....4095
%xT.....offset_ADC

offset_ADC = xT*4095/vref;

Rshunt=0.007;

%Isen=10*Ushunt;
%Ishunt=Ushunt/Rshunt;
%Ishunt=(reg_adc_i*vref/4095)/10/Rshunt;

factor_conversie=(1/Rshunt)*(1/10)*(vref/4095);

%Kalman
TiSamplePos = T_pos;

```