

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1977

**ESTIMACIJA KUTA ROTORA MOTORA KORIŠTENJEM
HALLOVOG SENZORA**

Domagoj Ciković

Zagreb, lipanj 2020.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1977

**ESTIMACIJA KUTA ROTORA MOTORA KORIŠTENJEM
HALLOVOG SENZORA**

Domagoj Ciković

Zagreb, lipanj 2020.

Zagreb, 13. ožujka 2020.

DIPLOMSKI ZADATAK br. 1977

Pristupnik: **Domagoj Ciković (0036487757)**
Studij: Elektrotehnika i informacijska tehnologija
Profil: Elektroničko i računalno inženjerstvo
Mentor: prof. dr. sc. Damir Seršić

Zadatak: **Estimacija kuta rotora motora korištenjem Hallovoog senzora**

Opis zadatka:

U okviru diplomskog zadatka potrebno je realizirati pouzdanu estimaciju kuta rotora elektromotora korištenjem Hallovoog senzora. Za realizaciju vektorskog upravljanja sinkronim motorom potrebna je informacija o trenutnom kutu rotora. Ako se radi o upravljanju usklađenim sa smjerom magnetskog polja, onda se pretpostavlja konstantni odnos istog s mehaničkom pozicijom rotora. Hallovi senzori osiguravaju detekciju pozicije rotora u 6 dijelova punog kuta (sekstanata). Koristiti Kalmanov filter za optimalnu estimaciju kuta u maksimalnoj rezoluciji. Potrebno je kalibrirati sustav kako bi se kompenzirale sistemske pogreške u sekstantima, kao i realizirati robusnu varijantu estimacije da se u uvjetima promjena smjera rotacije minimizira maksimalno odstupanje. Algoritme simulirati u programskom jeziku MATLAB, te realizirati u stvarnom vremenu na odabranom ugradbenom računalu u programskom jeziku C/C++.

Rok za predaju rada: 30. lipnja 2020.

SADRŽAJ

1. Uvod	1
2. Upravljanje sinkronim motorima	2
2.1. Sinkroni motori	2
2.2. Vektorsko upravljanje motorima - FOC	7
2.3. Detekcija pozicije rotora motora pomoću Hallovih senzora	15
3. Kalmanov filter	20
3.1. Uvod u Kalmanov filter	20
3.2. Realizacija Kalmanovog filtra	21
3.3. Primjena Kalmanovog filtra	23
4. MATLAB analiza Kalmanovog filtra i metode kalibracije	26
4.1. Kalman filter	26
4.2. Kalibracija	32
5. Implementacija algoritma na STM32fxx	34
6. Zaključak	43
Literatura	45

1. Uvod

Pojava silicijevog karbida (SiC) donijela je velike promjene u industriji, a posebno na području učinkovite elektronike. SiC MOSFET tranzistore moguće je preklapati velikim brzinama te zadržavaju konstantni unutarnji otpor na visokim temperaturama. Također, SiC MOSFET podržava visoke napone (reda veličine kV) zbog čega se primjenjuje u trofaznim pretvaračima za pogon trofaznih elektromotora. Jedan od načina upravljanja elektromotorima je vektorsko upravljanje. Vektorsko upravljanje temelji se na podešavanju vektora struje kako bi se osigurao maksimalan moment motora, odnosno kut od 90° između statora i rotora. Da bi se postigao maksimalan moment sinkronog motora potrebno je poznavati trenutni kut rotora. Ako se radi o upravljanju motorom usklađenim sa smjerom magnetskog polja, tada je moguće odrediti mehaničku poziciju rotora, odnosno njegov kut u odnosu na stator. Postavljanjem Hallovih senzora na motor, razmaknutih za 120° , moguće je odrediti poziciju rotora u 6 dijelova punog kruga (seksanta), mjereći magnetsko polje koje stvara rotor. Za veću točnost mjerenja kuta rotora potrebno je više Hallovih senzora, no poziciju rotora između dva senzora moguće je predvidjeti koristeći već postojeće algoritme. Kalmanov filter omogućuje nam optimalnu estimaciju kuta na temelju prethodnih mjerenja i diferencijalnih jednadžbi koje predstavljaju sustav. Algoritam Kalmanovog filtra moguće je implementirati na raznim mikrokontrolerima koji povezuju senzore i omogućuju obradu njihovih izlaznih signala. Uspješnost algoritma ovisi o mnogim parametrima, kao što su kovarijancijska matrica procesnog šuma i modelu vektora stanja koji se koristi. Izmjereni kut rotora se koristi u vektorskom upravljanju kako bi se podesio vektor struja koje se koriste u pulsnoj modulaciji i ostvario željeni puni moment. Kako bi se kompenzirale sistemske pogreške potrebno je kalibrirati sustav i tako dovesti odstupanja na minimum. Vektorsko upravljanje motorima, zbog svojih kvaliteta, pronalazi široku primjenu u automobilskoj industriji kako bi se postiglo optimalno upravljanje motorom u svim režimima rada, dobra energetska bilanca motora te postigla dobra dinamička svojstva upravljanja u uvjetima velike akceleracija ili retardacije.

2. Upravljanje sinkronim motorima

2.1. Sinkroni motori

Rad svih električnih strojeva može se analizirati pomoću tri osnovna zakona elektrotehnike [7]. Prvi zakon je zakon elektromagnetske indukcije (Faradejev zakon):

$$e = -\frac{d\Phi(t)}{dt}. \quad (2.1)$$

Drugi zakon je zakon protjecanja (Amperov zakon protjecanja):

$$\Theta = \oint \vec{H} d\vec{l}. \quad (2.2)$$

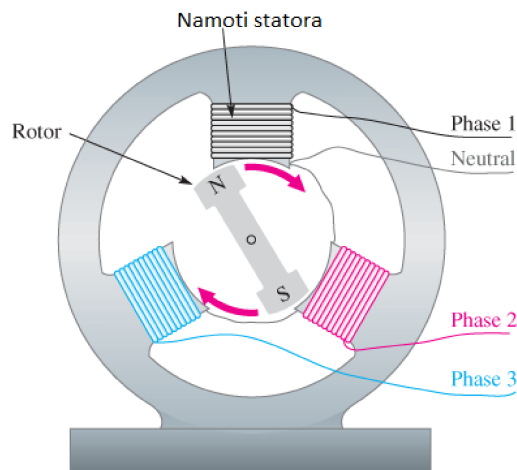
Treći zakon je zakon sile u magnetskom polju:

$$\vec{F} = I(\vec{l} \times \vec{B}). \quad (2.3)$$

Za pretvorbu energije u električnom stroju potrebno je imati vodič, koji se giba relativno, i njime protječe električna struja prema magnetskom polju. Stoga svaki električni stroj ima pomični dio koji se naziva rotor i statični dio koji miruje, odnosno stator. Jezgre rotora i statora provode magnetski tok te su izrađene od feromagnetskih materijala koji imaju dobru magnetsku vodljivost. Za rad na mrežama izmjeničnog napona koriste se sinkroni i asinkroni strojevi. Bitna karakteristika sinkronih strojeva je da im je brzina vrtnje rotora ω jednaka brzini vrtnje okretnog magnetskog polja kojeg stvaraju struje kroz namote na statoru. Ta se brzina naziva sinkronom brzinom stroja ω_s i određena je brojem parova polova stroja p i frekvencijom napona napajanja f prema jednadžbi:

$$\omega = \omega_s = \frac{60f}{p}. \quad (2.4)$$

Frekvencija napona napajanja stroja, f , jednaka je frekvenciji napona gradske mreže ili općenito, frekvenciji napona i struja u statoru, f_s . Kod asinkronih strojeva brzina vrtnje ovisi o opterećenju stroja i u stacionarnom pogonu različita je od sinkrone ($n \neq n_s$).



Slika 2.1: Slika prikazuje namote statora te odnos rotora i statora. Preuzeto s [poveznice](#).

Na statoru sinkronog stroja je u većini slučajeva smješten armaturni namot u kojem se pod utjecajem promjene magnetskog toka inducira napon. U vodičima statorskog namota inducira se elektromotorna sila te kada je stroj pod opterećenjem kroz namote protoku struje koje stvaraju okretno protjecanje koje se vrti jednakom brzinom kao i rotor. Dakle, možemo reći da su rotacija rotora i frekvencija struja, koje prolaze kroz namote, sinkronizirane [23].

Sinkroni strojevi se najčešće izvode kao trofazni generatori, s dva ili više polova. Postoje razne izvedbe sinkronih motora. Jedna od njih je sinkroni motor s permanentnim magnetom (PMSM) [1]. PMSM se sastoji od namota na svaku od tri faze na statoru i permanentnim magnetom u rotoru. Za razliku od uobičajenih sinkronih strojeva, koji zahtijevaju napajanje izmjeničnom i istosmjernom strujom, PMSM za svoj rad zahtijeva samo izmjeničnu struju. To je ujedno i njegova najveća prednost nad uobičajenim sinkronim motorima. Postoje razne izvedbe PMSM-a. Tako ih možemo podijeliti na radijalne i aksijalne. Kod radijalnog polja, smjer magnetskog toka je uzduž polumjera stroja. Motori s permanentnim magnetima s radijalnim poljem se najčešće koriste. Kada je u pitanju aksijalno polje, smjer magnetskog toka paralelan je s osovinom rotora. Takvi motori se koriste u raznim primjenama gdje je potrebna veća gustoća snage i naglo ubrzanje. Razlikujemo i sinkrone motore prema načinu na koji se postavljaju magneti na rotor motora. Magneti se mogu postaviti na površinu rotora ili u njega.

Za teorijska razmatranja dovoljno je promatrati samo dva pola, odnosno jedan par polova, budući da je fizikalna slika jednaka za svaki par polova. Iako se analiza može provesti koristeći samo jedan par polova, u stvarnosti treba uzeti u obzir sve polove [7].

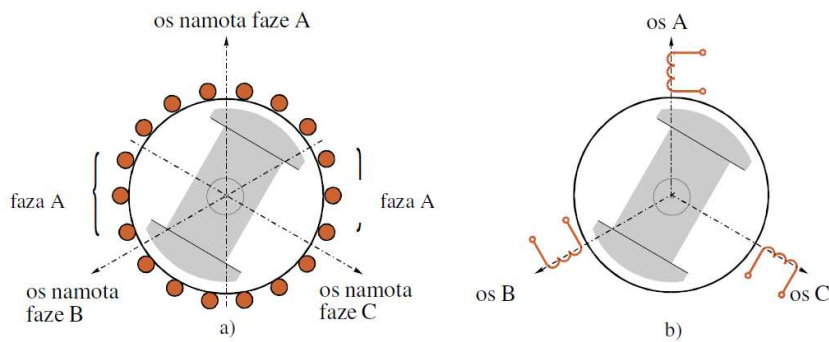
Povezanost između mehaničkog kuta, Θ_m , i električnog kuta, Θ_e , dana je sljedećim izrazom [14]:

$$\Theta_e = p\Theta_m. \quad (2.5)$$

Isto vrijedi i za kutnu brzinu rotora.

$$\omega_e = p\omega_m \quad (2.6)$$

Statorski namot je simetričan te os svakog namota u njegovoj simetrali. Osi namota sve tri faze razmaknute su za 120 električnih stupnjeva.



Slika 2.2: Shematski prikaz 3-faznog statorskog namota [7].

a) smještaj vodiča b) osi namota

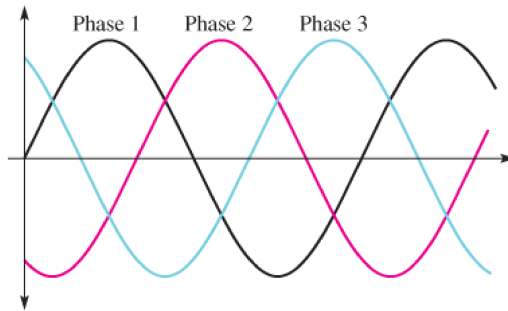
Kada se rotor vrti konstantnom brzinom n , tada magnetsko polje rotora inducira u namotima pojedinih faza elektromotorne sile koje se u vremenu mijenjaju frekvencijom:

$$f = f_s = \frac{np}{60}. \quad (2.7)$$

Kružnu frekvenciju možemo izraziti kao:

$$\omega = 2\pi f. \quad (2.8)$$

Kako su i namoti pomaknuti za 120 stupnjeva, inducirane elektromotorne sile vremenski su pomaknute za 120 električnih stupnjeva.



Slika 2.3: Inducirane elektromotorne sile u pojedinim fazama 3-faznog sinkronog stroja. Preuzeto s [poveznice](#).

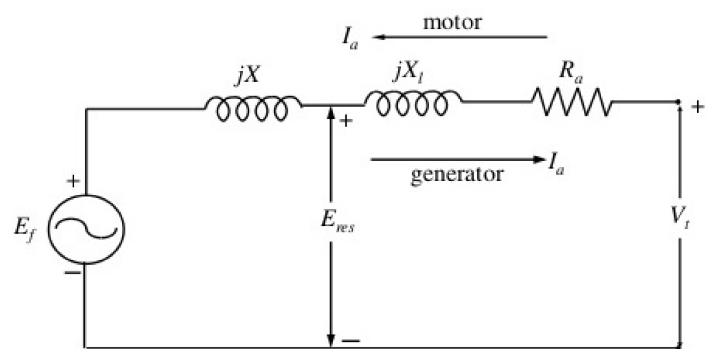
Linijski naponi i struje se mogu prikazati pomoću sinusnih funkcija pomaknutih za $\frac{2\pi}{3}$.

$$V_a = \frac{E}{2} \cos(\omega t) \quad (2.9)$$

$$V_b = \frac{E}{2} \cos(\omega t - \frac{2\pi}{3}) \quad (2.10)$$

$$V_c = \frac{E}{2} \cos(\omega t + \frac{2\pi}{3}) \quad (2.11)$$

Sinkroni motor se može prikazati pojednostavljenom shemom koja pokazuje unutarnji otpor i induktivitet motora. Ako struja pojedine faze protječe u negativnu stranu govorimo o motoru. U suprotnom slučaju radi se o generatoru.



Slika 2.4: Pojednostavljena shema sinkronog motora. Preuzeto s [poveznice](#).

Uzimajući u obzir unutarnji otpor i induktivitet, napon se može prikazati kao:

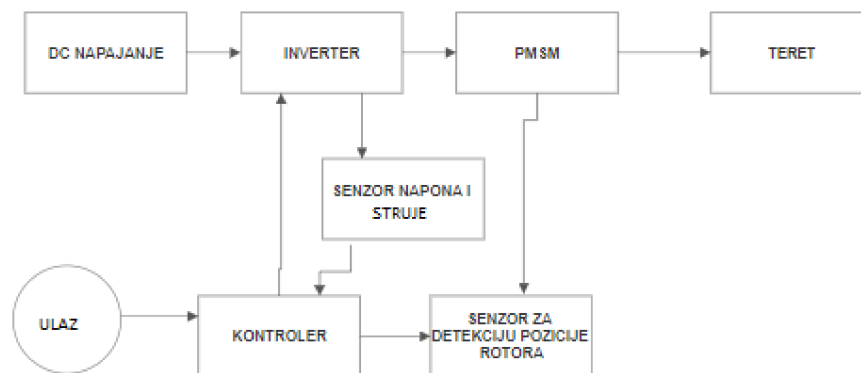
$$V_{a,b,c} = Ri_{a,b,c} + \frac{d\psi}{dt} = Ri_{a,b,c} + \frac{d[Li_{a,b,c} + \Psi_m(\Phi)]}{dt}, \quad (2.12)$$

gdje je Ψ_m prirodni magnetski tok permanentnog magneta.

Za kraj je važno istaknuti još jednu bitnu veličinu za sinkrone motore, a to je okretni moment. Okretni moment je sila okretanja koja omogućuje rad motora, odnosno daje kutno ubrzanje. Okretni moment sinkronih motora je konstantan i proporcionalan sa strujom I_s .

$$T = \frac{3}{2} I_s \Psi_m \quad (2.13)$$

Upravljanje elektromotorima sastoji se od četiri glavne jedinice, a to su PMSM, inverter, kontrolna jedinica i senzori struje i pozicije.

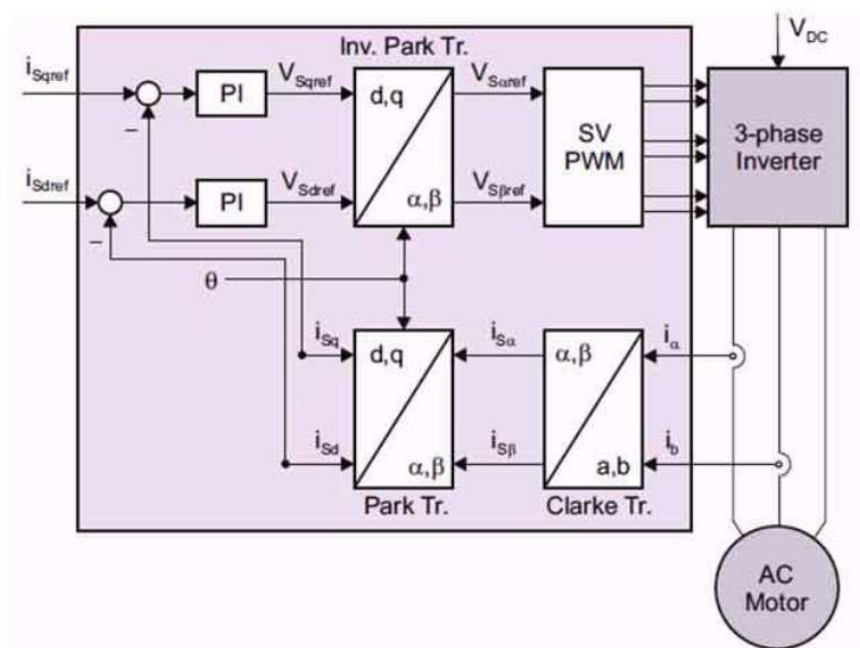


Slika 2.5: Jedinice za upravljanje motorima.

Postoje razne tehnike kontrole motora, a neke od njih su skalarno upravljanje motorom i vektorsko upravljanje motorom. Skalarno upravljanje izmjeničnim motorima koristi se u aplikacijama u kojima podaci o kutnoj brzini ne moraju biti poznati. Tipično se koristi za pogon crpki ili ventilatora. Veličina napona napajanja mijenja se u skladu s frekvencijom u konstantnom omjeru. Glavna prednost ovog režima je što nisu potrebni senzori za mjerenje kutne brzine ili položaja rotora, ali su bitno smanjene dinamičke performanse. Kada je u pitanju vektorsko upravljanje motorima razlikuje Direct Torque Control (DTC) i Field Orientated Control (FOC). Izravna kontrola okretnog momenta se temelji na procjeni magnetskog toka i zakretnog momenta na temelju izmjerenog napona i struje motora. Magnetski tok se procjenjuje integriranjem napona statora, a zakretni moment se procjenjuje kao produkt procijenjenog magnetskog toka statora i struje motora. Procijenjene vrijednosti se uspoređuju s referentnima i u ovisnosti o rezultatu, regulira se zakretni moment [19]. Više o FOC-u i njegovim dijelovima nalazi se u nastavku.

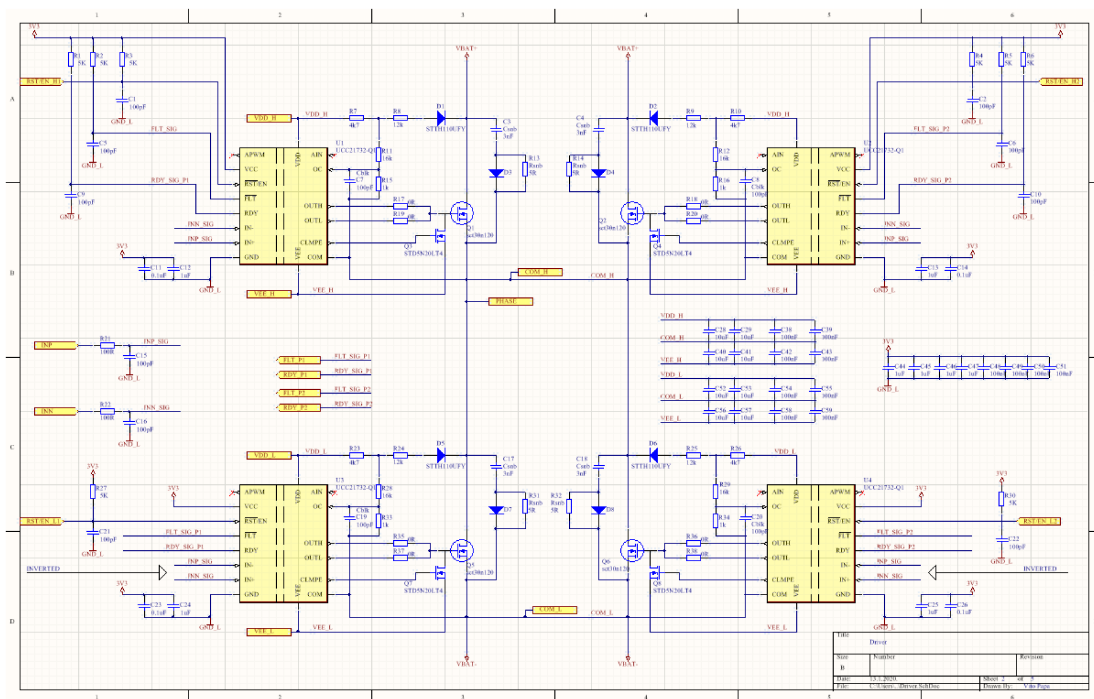
2.2. Vektorsko upravljanje motorima - FOC

Kao što smo u prethodnom poglavlju opisali, za upravljanje sinkronim motorima potrebna nam je kontrolna jedinica, inverter i razni senzori. Postoje razne izvedbe vektorskog upravljanja motorima, a u ovom poglavlju bazirat ćemo se na Field orientated control (FOC) tehnici. FOC se koristi za kontrolu sinkronih motora s permanentnim magnetima s ciljem da se motor ponaša kao istosmjerni. Namoti statora motora se napajaju preko invertera koji stvara promjenjivi napon s promjenjivom frekvencijom. Umjesto da se kontrolira frekvencija rada invertera, frekvencija i faza izlaznog vala se reguliraju preko pozicije kuta rotora. FOC pokazuje da se sinkronim motorom može upravljati kao istosmjernim motorom pomoću orijentacije statora ili vektora struja u odnosu na magnetski tok rotora. Da bi se motor ponašao kao istosmjerni motor, potrebno je znati položaj rotora motora u odnosu na stator [1].



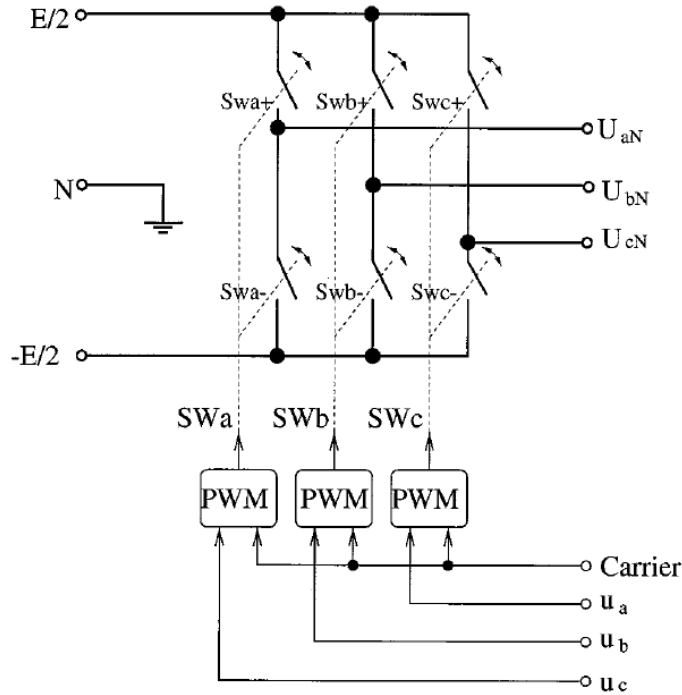
Slika 2.6: Shematski prikaz FOC-a [14].

Za početak ćemo se dotaknuti invertera koji upravlja radom motora. Shematski prikaz invertera nalazi se na slici 2.7.



Slika 2.7: Shematski prikaz invertera. Dvije grane polumosta.

Inverter je izveden kao trofazni polu-most kojemu je svaka grana spojena na jednu od faza motora. U svakoj grani mosta nalaze se po dvije prekidačke sklopke preko kojih se upravlja motorom. Prekidačke sklopke su SiC MOSFET-i, koji se zbog svojih visokih performansi koriste u visokonaponskim aplikacijama. SiC MOSFET-i omogućuju rad na visokim temperaturama, zadržavajući konstantan iznos otpora i na temperaturama do 200°C. Za napone od 1200 V, taj unutarnji otpor iznosi oko 100 mV [17]. SiC MOSFET-i također omogućuju velike brzine preklapanja reda veličine ns, zbog čega se koriste u automobilske industriji. Za upravljanje sklopkama koriste se driveri koji omogućuju brzo preklapanje i smanjuju gubitke usred preklapanja. Veliku pozornost mora se obratiti na to da obje sklopke jedne grane mosta ne smiju biti istovremeno otvorene ili zatvorene. Driveri moraju imati galvansku izolaciju zbog velike razlike u naponima na ulaznoj i izlaznoj strani. Na ulaznoj strani nalazi se procesor koji generira PWM. Procesor radi na niskim naponskim razinama, za razliku od izlaznog dijela koji radi na naponskim razinama veličine kV. Inverter otvaranjem i zatvaranjem sklopki pretvara konstantnu razinu istosmjernog napona u promjenjivi izmjenični napon. Istosmjerni napon nalazi se u gornjoj i donjoj grani mosta izveden tako da u gornjoj grani se nalazi napon $\frac{E}{2}$, a u donjoj grani napon $-\frac{E}{2}$.



Slika 2.8: Trofazni PWM inverter [14].

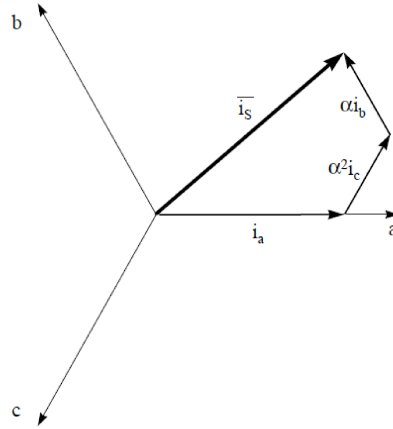
FOC se temelji na projekciji koja transformira trofazni vremenski ovisan sustav u sustav s dva vektora neovisnima o vremenu. Takve projekcije vode do strukture koja je slična upravljanju istosmjernim motorima. Ulazne reference u FOC su dvije konstantne komponente, komponenta okretnog momenta i komponenta magnetskog toka statorskih struja. Komponenta okretnog momenta je poravnata s q-osi, a magnetskog toka poravnata je s d-osi [4].

Trofazne struje i naponi se mogu prikazati kao kompleksni prostorni vektori, gdje se i_a , i_b , i_c odnosi na struje pojedine faze. Kompleksni vektor struja statora možemo prikazati kao:

$$\vec{i}_s = \alpha^0 i_a + \alpha^1 i_b + \alpha^2 i_c \quad (2.14)$$

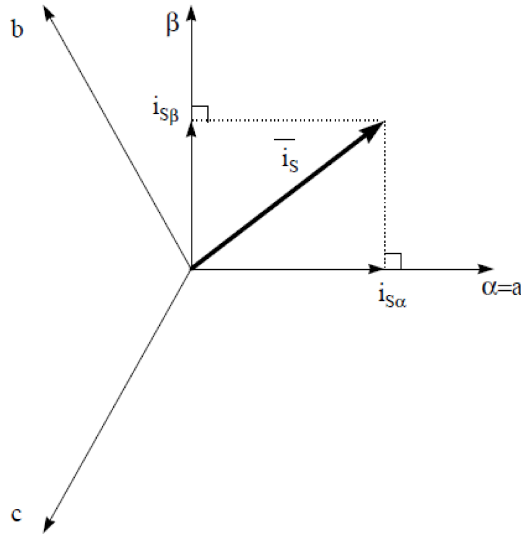
gdje je $\alpha^0 = 1$, $\alpha^1 = e^{j\frac{2\pi}{3}}$, $\alpha^2 = e^{j\frac{4\pi}{3}}$.

Na slici 2.9 prikazan je vektorski dijagram statorskih struja. Na grafu su osi označene s (a,b,c) i odnose se na faze sustava koje je potrebno prebaciti u dvodimenzionalni koordinatni sustav. Transformacije se provode pomoću Clarkeove i Parkove transformacije.



Slika 2.9: Vektorski prikaz statoskih struja [4].

Clarkeova transformacija projektira trodimenzionalni prostor (a,b,c) u vektorski prostor s dvije ortogonalne osi (α, β). Uzimajući u obzir da su a-os i α -os u istom smjeru, vektorski prostor možemo prikazati kao na slici 2.10.



Slika 2.10: Vektorski prikaz statoskih struja u dvodimenzionalnom sustavu (α, β) [4].

Struje $i_{s\alpha}$ i $i_{s\beta}$ se mogu prikazati kao:

$$i_{s\alpha} = i_a \quad (2.15)$$

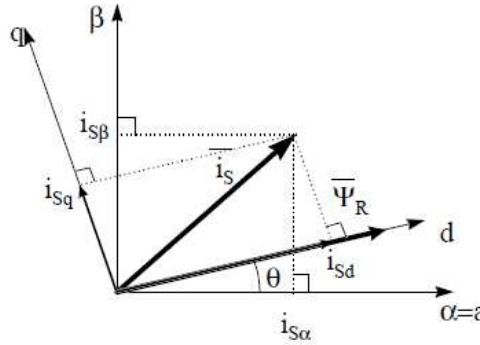
$$i_{s\beta} = \frac{1}{\sqrt{3}}i_a + \frac{2}{\sqrt{3}}i_b. \quad (2.16)$$

Vektorski prikaz struja statora tada izgleda:

$$\vec{i}_s = 1i_{s\alpha} + ji_{s\beta}. \quad (2.17)$$

Struju i_c možemo lako izračunati ako znamo preostale dvije struje ($i_a + i_b + i_c = 0$).

Iako smo dobili sustav s dvije komponente, on je i dalje ovisan o vremenu i brzini te nam je potreban rotirajući koordinatni sustav. Parkova transformacija mijenja dvodimenzionalni ortogonalni sustav u rotirajući. To je pogodno jer komponente vektora i_{sq} i i_{sd} miruju u rotirajućem koordinatnom sustavu. Ako uzmemo u obzir da je d-os poravnata s magnetskim tokom rotora, vektorski prostor se može prikazati kao na slici 2.11.



Slika 2.11: Rotirajući koordinatni sustav [4].

Kut Θ nam govori trenutnu poziciju rotora. Parkovu transformaciju možemo prikazati kao:

$$\begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} = \begin{bmatrix} \cos\Theta & \sin\Theta \\ -\sin\Theta & \cos\Theta \end{bmatrix} \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix}. \quad (2.18)$$

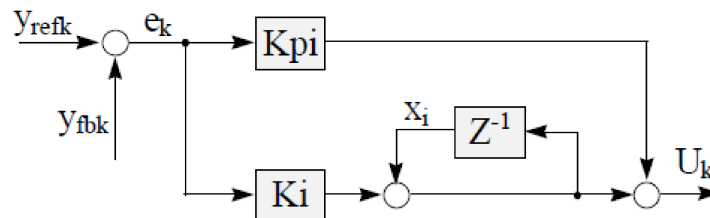
Komponente i_{sd} i i_{sq} ovise o vektorima struja (α, β) i poziciji rotora. Kada znamo poziciju kuta rotora, tada komponente d,q postaju konstantne. Vektorskim upravljanjem želimo postići da je razlika između d-osi i q-osi jednaka 90° i konstantna. Sukladno Parkovoj i Clarkeovoj transformaciji potrebno je pronaći i njihov inverz. Inverzne transformacije koristimo kada želimo statorske struje izraziti u vektorskom prostoru (a,b,c). Inverzna Parkova matrica dana je izrazom:

$$\begin{bmatrix} \cos\Theta & -\sin\Theta \\ \sin\Theta & \cos\Theta \end{bmatrix}.$$

Kod sinkronih motora s permanentnim magnetom vrijednost komponente i_{sd} trebala bi biti jednaka 0. Opcionalno se struja i_{sd} može koristiti za istiskivanje magnetskog toka magneta čime se povećava brzina motora te smanjuje moment [14]. Kao što vidimo, izrazi za Parkovu i inverznu Parkovu transformaciju trebaju informaciju o poziciji rotora i iznose struja. Struje mjerimo pomoću A/D pretvarača i uvodimo u sustav, a poziciju rotora pomoću senzora pozicije, što će detaljnije biti opisano u sljedećem poglavlju.

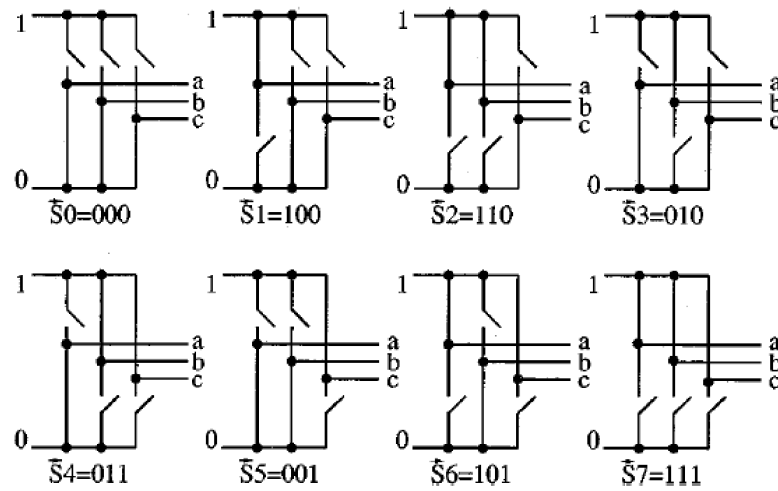
Za kraj nam preostaje objasniti ulogu PI (Proportional and Integral) regulatora prikazanog na slici 2.12. PI regulator je vrlo prikladan za regulaciju povratne veze okretnog momenta i magnetskog toka. Postavljanjem pravih vrijednosti pojačanja K_{pi} i K_i može se postići konstantna referenca. PI regulator kontinuirano uspoređuje pravu, mjerenu, vrijednost struja s referentnom, željenom, vrijednosti i primjenjuje korekcije temeljene na proporcionalnim i integralnim izrazima [21]. Proporcionalni izraz postavlja izlaznu vrijednost koja je proporcionalna trenutnoj vrijednosti pogreške pomnoženoj s konstantom K_{pi} , koja se naziva konstantom proporcionalnog dobitka. Za previsoku vrijednost konstante K_{pi} sustav može postati nestabilan, a premala vrijednost može regulator učiniti manje osjetljivim na ulaznu grešku. Doprinos integralnog izraza proporcionalan je i veličini pogreške i trajanju pogreške. Integral u PI regulatoru je zbroj trenutnih pogrešaka tijekom vremena i daje akumulirani pomak koji je trebao biti prethodno ispravljen. Akumulirana pogreška množi se s pojačanjem K_i . Integralni izraz ubrzava kretanje procesa prema željenoj vrijednosti i eliminira zaostalu pogrešku u mirovanju koja se javlja s čistim proporcionalnim regulatorom. PI regulator u diskretnom sustavu može se prikazati jednadžbom:

$$U_k = K_{pi}e_k + K_i e_k + \sum_{n=0}^{k-1} e_n. \quad (2.19)$$



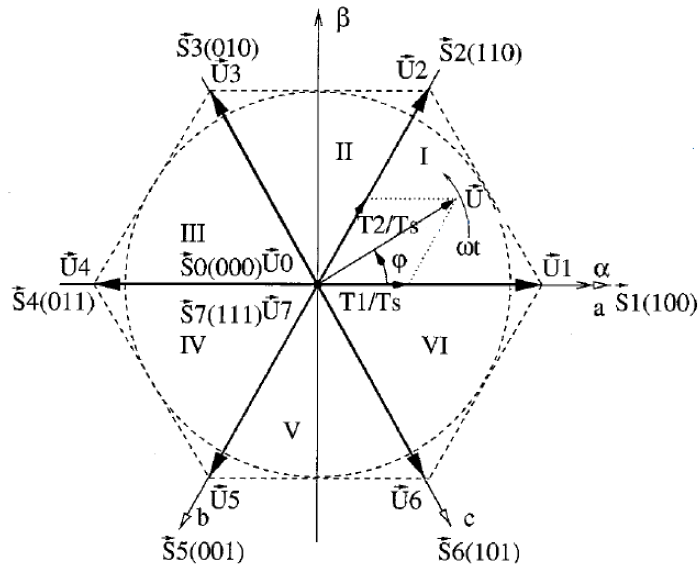
Slika 2.12: PI regulator [4].

Nakon što se pomoću PI regulatora osigura željena vrijednost signala, potrebno je napraviti inverznu Parkovu transformaciju kako bismo dobili ulaze za SVPWM. SVPWM (eng. *Space Vector Pulse Width Modulation*) je tehnika koja se koristi kao posljednji korak FOC-a za generiranje PWM-a signala za upravljanje sklopkama invertera [3]. Napon koji se dovodi na motor moduliran je pomoću trofaznog invertera koji sadrži po dva tranzistora u svakoj grani polu-mosta. U svakoj grani polu-mosta može biti jedan tranzistor otvoren, dok je drugi zatvoren, što nam ukupno daje osam (2^3) mogućih izlaznih stanja. Tih 8 stanja odnosi se na bazne vektore. Moguće kombinacije prikazane su na slici 2.13.



Slika 2.13: Stanja sklopki trofaznog invertera u SVPWM modulaciji [15].

Vektor s 8 baza je prikazan pomoću vektorskog dijagrama na slici 2.14. Svaki vektor čini jedan krak zvijezde gdje su dva susjedna kraka razmaknuta za 60° . Dva vektora ($S_0 = 000$ i $S_7 = 111$) se prikazuju kao nul-vektori položeni u središtu zvijezde.



Slika 2.14: Bazni vektori koji čine šesterokraku zvijezdu sa vektorima S_0 i S_7 kao nul-vektorima u središtu [15].

Cilj SVPWM-a je generirati vektor za PWM perioda (T_{PWM}) koji je jednak željenoj vrijednosti naponskog vektora $\vec{U}_{out}$. Smjer i orijentacija vektora $\vec{U}_{out}$ određena je prema vrijednostima baznih vektora, zajedno s nul-vektorima. To je moguće tako da se određeno stanje sklopki zadrži neko određeno vrijeme T . Kao primjer možemo uzeti bazni vektor $\vec{U}_1$ koji se primjenjuje kroz vrijeme T_1 i bazni vektor $\vec{U}_3$ s vremenom

trajanja T_3 i nul-vektori trajanja T_0 . Vrijednosti trajanja T_1 , T_3 i T_0 se mogu izraziti kao:

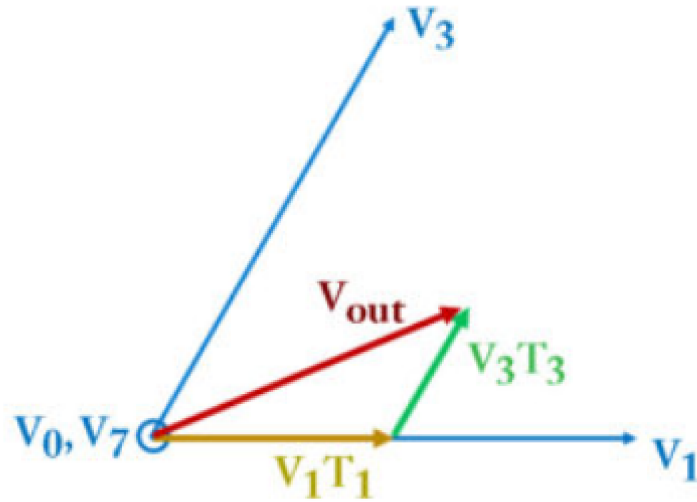
$$T_1 = \left| \frac{\vec{U}_{out}}{\vec{U}_1} \right| T_{PWM} \quad (2.20)$$

$$T_3 = \left| \frac{\vec{U}_{out}}{\vec{U}_3} \right| T_{PWM} \quad (2.21)$$

$$T_0 = T_{PWM} - (T_1 + T_3). \quad (2.22)$$

Konačno, vrijednost vektora $\vec{U}_{out}$ možemo izraziti kao:

$$\vec{U}_{out} = \vec{U}_1 T_1 + \vec{U}_3 T_3 + \vec{U}_0 T_0. \quad (2.23)$$



Slika 2.15: Primjer računanja željenog vektora napona [3].

2.3. Detekcija pozicije rotora motora pomoću Hallovih senzora

Elementi temeljeni na Hallovom efektu načinjeni su od tankog sloja vodljivog materijala s izlaznim vodovima okomitim na smjer toka struje. Kada se na takav element primjeni magnetsko polje, javlja se izlazni napon koji je proporcionalan jačini narinutog magnetskog polja [18]. U poluvodičima, koji se koriste u Hallovim senzorima, nosioci naboja su elektroni i šupljine. Kada se na poluvodič primjeni magnetsko polje, na nosioce naboja poluvodiča djeluje sila okomita na magnetsko polje i na smjer struje te se na rubovima poluvodiča javlja napon. Dakle, Hallov efekt je posljedica Lorentzove sile $\vec{F}$ na naboj q koji se giba brzinom $\vec{v}$ u magnetskom polju indukcije $\vec{B}$ [20].

$$\vec{F} = -q(\vec{v} \times \vec{B}) \quad (2.24)$$

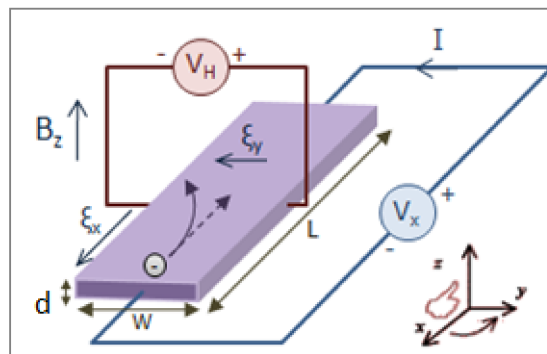
U poluvodičima doprinose vođenju donose i elektroni i šupljine koji mogu biti prisutni u različitim koncentracijama i imati različite pokretljivosti. Hallov koeficijent, R_H , se kod poluvodičkih elemenata prikazuje kao:

$$R_H = \frac{p\mu_h^2 - n\mu_e^2}{e(p\mu_h + n\mu_e)} \quad (2.25)$$

gdje je n koncentracija elektrona, p koncentracija šupljina, μ_e pokretljivost elektrona, μ_h pokretljivost šupljina i e predstavlja jedinični naboj. Napon koji se javlja se može izraziti kao:

$$V_H = R_H \frac{I}{d} B_z \quad (2.26)$$

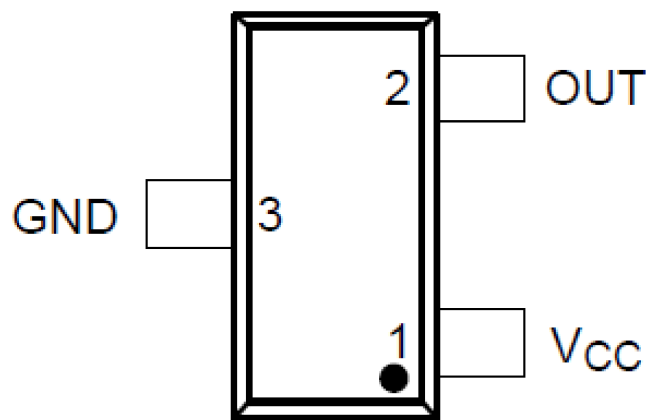
gdje je I konstantna struja duž poluvodiča, d debljina poluvodiča i B_z magnetsko polje okomito na poluvodič.



Slika 2.16: Mjerenje Hallovog napona [20].

Iako Hallovi senzori rade na principu detekcije magnetskog polja, svejedno imaju široku primjenu. Hallovi senzori se mogu koristiti za mjerenje struja, temperature, tlaka, pozicije, brzine i slično. Prednosti Hall senzora su robusnost, dugi životni vijek, rad na velikim brzinama, rad na temperaturama od -40°C do $+150^{\circ}\text{C}$ i ostalo. Postoje razne izvedbe Hallovih senzora, a u nastavku ćemo se bazirati na Hallovim sensorima s digitalnim izlazom [6].

Digitalni Hallovi senzori koriste se za mjerenje pozicije rotora motora. Senzori sadrže tri glavna voda, a to su: uzemljenje (GND), napajanje (V_{CC}) i digitalni izlaz (OUT).



Slika 2.17: Kućište SOT-23 digitalnog Hallovog senzora [8].

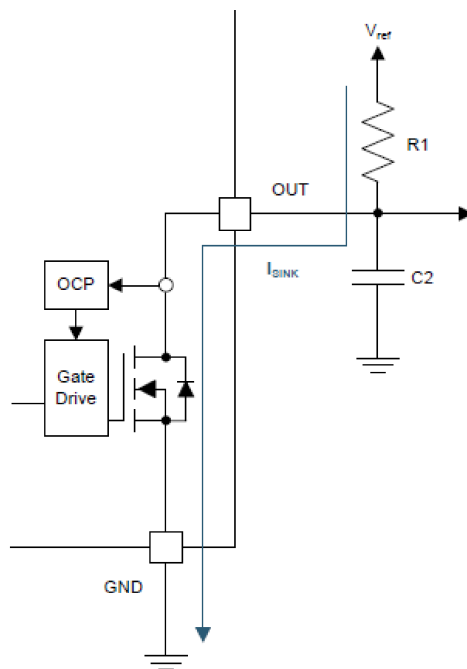
Kako želimo da signal koji mjerimo nema smetnji, moramo odabrati ispravne vrijednosti otpornika i kondenzatora koji se spajaju na senzor. Drugim riječima, potrebno je na izlaz staviti niskopropusni filter koji će prigušiti šum. Vrijednosti elemenata za Hallov senzor DRV5013 određuju se prema jednadžbama:

$$\frac{V_{refmax}}{I_{sinkmax}} \leq R_1 \leq \frac{V_{refmin}}{I_{sinkmin}} \quad (2.27)$$

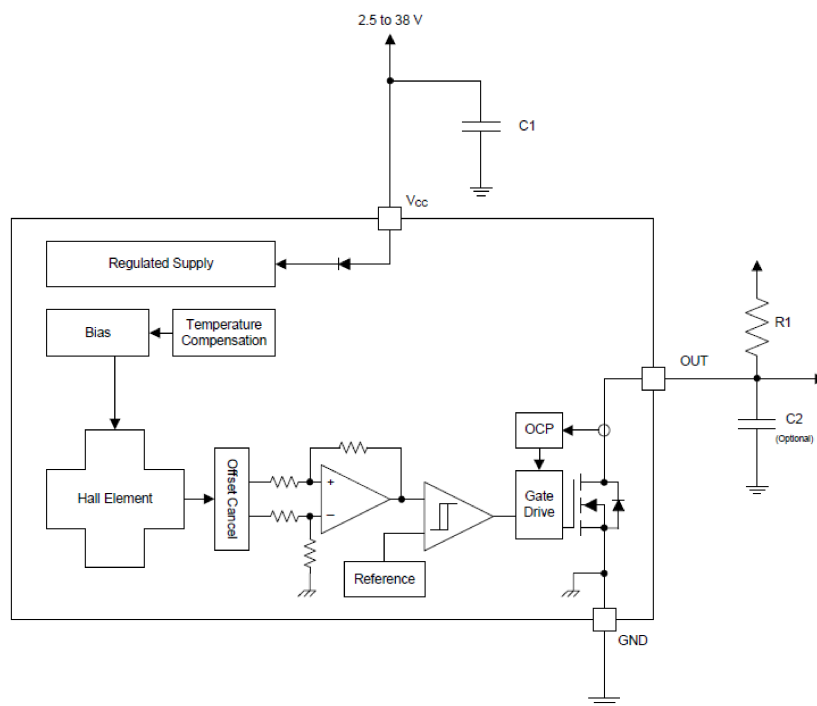
i jednadžbi za definiranje niskopropusnog filtra:

$$2f_{BW}(\text{Hz}) < \frac{1}{2R_1C_2} \quad (2.28)$$

gdje je f_{BW} širina pojasa. Kondenzator C_1 je blokadni kondenzator koji se spaja uz napajanje [8].

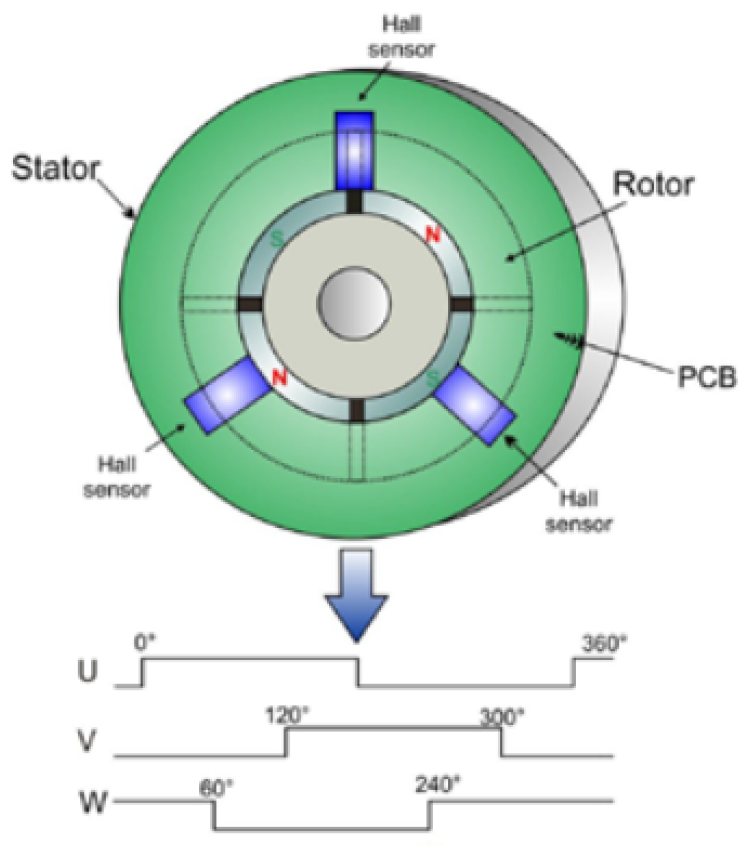


Slika 2.18: Shema za odabir vrijednosti otpornika R_1 [8].



Slika 2.19: Funkcijski dijagram Hallovog senzora [8].

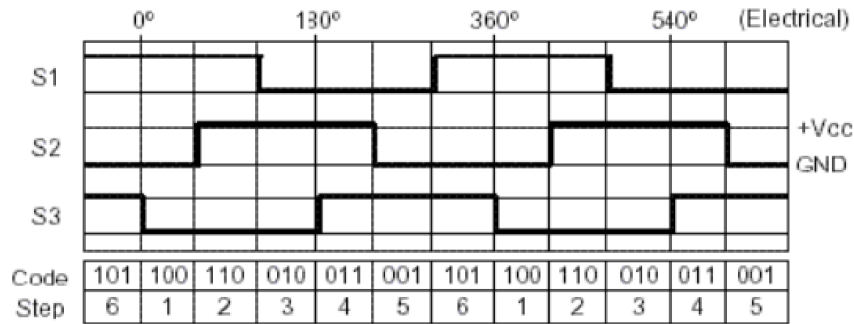
Halovi senzori se koriste za detekciju pozicije rotora, odnosno kuta, koja se koristi za vektorsko upravljanje motorima. Tri senzora se postave u razmaku od 120° na sinkroni motor. Pozicije senzora su slijedno: 60° , 180° i 300° . U realnom svijetu pozicije Hallovih senzora nikada nisu idealne i postoji odstupanje koje unosi pogrešku u proračunima brzine okretanja motora potrebnima za FOC. Odstupanje idealnih vrijednosti pozicija Hallovih senzora uklanja se kalibracijom. Kao što smo ranije naveli rotor sadrži permanentne magnete koji stvaraju magnetsko polje i svakim prolaskom rotora uz Hall sensor generira se signal. Pozicije i generirani signal možemo vidjeti na slici 2.20.



Slika 2.20: Raspored Hallovih senzora na elektromotoru i njihovi izlazni signali. Preuzeto s [poveznice](#).

Analizom izlaznih signala možemo detektirati trenutnu poziciju rotora motora. Ovisno, radi li se o rastućem ili padajućem bridu signala i znamo li na kojoj je poziciji Hallov senzor smješten, možemo generirati binarni kod s pripadajućim kutevima. Razine izlaznih signala govore nam koji je senzor aktivan. Ako je razina signala jednaka "1", na Hallov senzor primijenjeno je magnetsko polje. Kao primjer možemo uzeti kut rotora od 0° koji odgovara padajućem bridu Hallovog senzora "S3". Prikaz signala i pripadnih

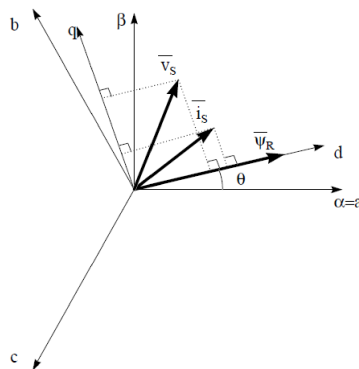
binarnih kodova možemo vidjeti na slici 2.21 [5]. Tako možemo puni krug podijeliti na šest dijelova, odnosno na šest sekstanta. Svaki sekstant odgovara određenom binarnom kodu. Hallovi senzori nemaju mogućnost mjeriti apsolutnu poziciju rotora, stoga je potreban Kalmanov algoritam koji će estimirati vrijednosti unutar jednog sekstanta.



Slika 2.21: Izlazni signali Hallovih senzora i pripadajući binarni kod pojedinog kuta [5].

Prepoznavanjem rastućeg ili padajućeg brida signala možemo mjeriti vrijeme potrebno da rotor prijeđe između dva senzora te pomoću Kalmanovog filtra procijeniti poziciju rotora između dva senzora.

Na kraju treba napomenuti da je poznavanje kuta rotora jedan od glavnih elemenata za FOC. Netočna vrijednost kuta rotora rezultirat će pogrešnom vrijednosti struja, odnosno napona kojima reguliramo željeni PWM. Kod sinkronih motora kutna brzina rotora proporcionalna je magnetskom toku rotora, stoga je potrebno koristiti Hallove senzore za detekciju kuta rotora. Na slici 2.22 možemo vidjeti konačni odnos prostornih vektora struja statora (a, b, c), (α, β), (d, q) i pozicije rotora [4].



Slika 2.22: Prostorni vektori struja, napona i magnetskog toka rotora u rotirajućem koordinatnom sustavu (d, q) i njegov odnos s (a, b, c) i (α, β) koordinatnim sustavom [4].

3. Kalmanov filter

3.1. Uvod u Kalmanov filter

Mnogi signali se mogu opisati kao [10]:

$$y_k = a_k x_k + n_k \quad (3.1)$$

gdje je y_k vremenski ovisan signal, a_k predstavlja pojačanje signala, x_k je signal koji sadrži informaciju i n_k je aditivni šum. Kada procjenjujemo signal x_k postoji razlika između procijenjene vrijednosti $\hat{x}_k$ i x_k , koja predstavlja pogrešku.

$$f(e_k) = f(x_k - \hat{x}_k). \quad (3.2)$$

Funkcija $f(e_k)$ mora biti pozitivna i monotonno rastuća, stoga je funkcija pogreške kvadratna funkcija koja zadovoljava zadane uvjete.

$$f(e_k) = f(x_k - \hat{x}_k)^2. \quad (3.3)$$

Kako je važno uzeti u obzir da određeni filtri imaju mogućnost predviđanja stanja kroz vrijeme potrebna je očekivana vrijednost kvadratne pogreške.

$$E = E(f(e_k)). \quad (3.4)$$

Na kraju, očekivanu kvadratnu pogrešku možemo zapisati kao:

$$e(t) = E(e_k^2). \quad (3.5)$$

Kalmanov filter je linearni vremenski promjenjivi rekurzivni stohastički estimator koji daje optimalnu procjenu (u smislu najmanje očekivane kvadratne greške) neke slučajne veličine iz mjerenja koja sadrže aditivni šum. Kalmanov filter koristi opis sustava u prostoru stanja (eng. *state-space*) gdje slučajni proces kojeg estimiramo odgovara stanju sustava. Stanje sustava sadrži sve informacije o sustavu te temeljem poznatog stanja sustava uz poznatu pobudu sustava možemo izračunati buduća stanja i izlaze sustava [13]. Vremenski diskretni linearni sustav se može prikazati u sustavu stanja.

Vektor $\mathbf{u}[n]$ predstavlja vektor pobude sustava u koraku n .

Vektor $\mathbf{x}[n]$ predstavlja vektor stanja u koraku n .

Vektor $\mathbf{y}[n]$ predstavlja vektor odziva u koraku n .

Matrice $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$ sadrže koeficijente diferencijskih jednadžbi koje opisuju promatrani sustav. Za sustav drugog reda s dva ulaza i dva izlaza pripadajuće matrice su:

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}, \mathbf{B} = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}, \mathbf{C} = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix}, \mathbf{D} = \begin{pmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{pmatrix}.$$

Stanje sustava se može opisati pomoću dvije matrične jednadžbe:

$$\mathbf{x}[n + 1] = \mathbf{A}\mathbf{x}[n] + \mathbf{B}\mathbf{u}[n], \quad (3.6)$$

$$\mathbf{y}[n] = \mathbf{C}\mathbf{x}[n] + \mathbf{D}\mathbf{u}[n]. \quad (3.7)$$

Prva jednadžba prikazuje jednadžbu stanja, a druga jednadžba izlaznu jednadžbu. Neka je N dimenzija vektora stanja, $\mathbf{x}$, odnosno red sustava, neka je M dimenzija vektora ulaza, $\mathbf{u}$, odnosno broj ulaza, i neka je K dimenzija vektora izlaza, $\mathbf{y}$, odnosno broj izlaza. Tada se dimenzije matrica prikazuju kao: $\mathbf{A}$ je kvadratna matrica $N \times N$, $\mathbf{B}$ je $N \times M$, $\mathbf{C}$ je $K \times N$ i $\mathbf{D}$ je $K \times M$. Matrica $\mathbf{A}$ je uvijek kvadratna dimenzija $N \times N$, gdje je N red sustava. Problem nastaje kada imamo zadani slučajni proces $X(t)$ i šum $N(t)$, a možemo mjeriti sam aditivnu kombinaciju $Y(t) = X(t) + N(t)$. Pretpostavimo da smo izmjerili n vrijednosti $Y(t_1), Y(t_2), \dots, Y(t_n)$ i da vrijedi $t_i < t_j$ za $i < j$. Potrebno je pronaći vrijednost $X(t_0)$ za neki trenutak t_0 .

1. Ako je $t_0 < t_n$ problem se naziva interpolacija.
2. Ako je $t_0 = t_n$ problem se naziva filtracija.
3. Ako je $t_0 > t_n$ problem se naziva predikcija.

Procjenu slučajnog procesa $X(t)$ označavamo kao $\hat{X}(t)$.

3.2. Realizacija Kalmanovog filtra

U elektrotehnici se najčešće koristi kauzalni model sustava opisan izrazima:

$$\mathbf{X}[n] = \mathbf{A}\mathbf{X}[n - 1] + \mathbf{B}\mathbf{u}[n] + \mathbf{W}[n], \quad (3.8)$$

$$\mathbf{Y}[n] = \mathbf{H}\mathbf{X}[n] + \mathbf{V}[n]. \quad (3.9)$$

Vektori $\mathbf{W}[n]$ i $\mathbf{V}[n]$ predstavljaju procesni i mjerni šum.

Kalmanov filter se implementira preko dva skupa jednadžbi [13]:

1. Jednadžbe predikcije:

$$\hat{\mathbf{X}}^-[n] = \mathbf{A}\hat{\mathbf{X}}[n-1] + \mathbf{B}\mathbf{u}[n] \quad (3.10)$$

$$\mathbf{P}^-[n] = \mathbf{A}\mathbf{P}[n-1]\mathbf{A}^T + \mathbf{R}_{WW}[n] \quad (3.11)$$

2. Jednadžbe korekcije:

$$\mathbf{K}[n] = \mathbf{P}^-[n]\mathbf{H}^T(\mathbf{H}\mathbf{P}^-[n]\mathbf{H}^T + \mathbf{R}_{VV}[n])^{-1} \quad (3.12)$$

$$\hat{\mathbf{X}}[n] = \hat{\mathbf{X}}^-[n] + \mathbf{K}[n](\mathbf{Y}[n] - \mathbf{H}\hat{\mathbf{X}}^-[n]) \quad (3.13)$$

$$\mathbf{P}[n] = (\mathbf{I} - \mathbf{K}[n]\mathbf{H})\mathbf{P}^-[n] \quad (3.14)$$

U jednadžbi (3.10), $\hat{\mathbf{X}}[n-1]$ predstavlja vektor stanja pomaknut za jedan korak nazad tako da je model sustava postao kauzalan. Matrice $\mathbf{A}$ i $\mathbf{B}$ su matrice sustava koje su neovisne o koraku n pa se za sustav može reći da je postao vremenski nepromjenjiv. Vektor $\mathbf{u}[n]$ predstavlja pobudu sustava. Vektor $\hat{\mathbf{X}}^-[n]$ predstavlja estimaciju željenog vektora stanja. Jednadžba (3.11) predstavlja estimator kovarijancijske matrice $\mathbf{P}^-[n]$. $\mathbf{P}[n]$ predstavlja korelacijsku matricu greške procjene, $\mathbf{X}[n] - \hat{\mathbf{X}}[n]$, koja je ujedno i kovarijancijska matrica procjene, $\text{cov}(\mathbf{X}[n], \mathbf{X}[n])$. Šumovi $\mathbf{W}$ i $\mathbf{V}$ su međusobno nezavisni Gaussovi procesi očekivanja nula te poznatih korelacijskih funkcija $\mathbf{R}_{WW}[n]$ i $\mathbf{R}_{VV}[n]$. Jednadžba (3.12) predstavlja jednadžbu za računanje Kalmanovog pojačanja $\mathbf{K}[n]$. Kalmanovo pojačanje nam govori za koliko želimo promijeniti estimaciju vektora $\hat{\mathbf{X}}^-[n]$ u odnosu na mjerenje $\mathbf{Y}[n]$ [2]. Kalmanovo pojačanje se kreće između vrijednosti 0 i 1, odnosno:

$$0 \leq \mathbf{K}[n] \leq 1.$$

Kada je mjerna nesigurnost $\mathbf{R}_{VV}[n]$ jako velika i procjena nesigurnosti $\mathbf{P}[n]$ jako mala, Kalmanovo pojačanje se bliži nuli. Tada Kalmanovo pojačanje slabo utječe na mjerenje $\mathbf{Y}[n]$, a u puno većoj mjeri utječe na estimaciju. S druge strane, kada je mjerna nesigurnost mala, a procjena nesigurnosti velika, Kalmanovo pojačanje se približava vrijednosti 1. Kada je Kalmanovo pojačanje približno 1, tada ima veći utjecaj na mjerenju vrijednost. U jednadžbama (3.13) i (3.14) se ažuriraju vrijednosti estimacije vektora stanja i estimacije kovarijancijske matrice za računanje novih vrijednosti.

Filtriranje pomoću Kalmanovog filtra može se provesti u nekoliko koraka, a započinje u nekom trenutku n_0 s početnim stanjem sustava $\mathbf{X}[n_0]$ i $\mathbf{P}[n_0]$ te mjerenjem $\mathbf{Y}[n_0]$ [13]. Prvo se pomoću jednadžbe sustava u prostoru (3.10) određuje *a priori* procjena $\hat{\mathbf{X}}^-[n_0+1]$. U drugom koraku računamo vrijednost Kalmanovog pojačanja te pomoću jednadžbe (3.13) korigiramo *a priori* vrijednost procjene kako bi dobili *a posteriori* procjenu stanja $\hat{\mathbf{X}}[n+1]$ i kovarijancijsku matricu $\mathbf{P}[n+1]$. Postupak se povalja iterativno za svako sljedeće mjerenje $\mathbf{Y}[n]$, $n_0 < n$.

3.3. Primjena Kalmanovog filtra

Kalmanov filter ima brojne primjene u tehnologiji. Uobičajena primjena je za navigaciju i kontrolu vozila, posebno zrakoplova. Također se koristi na području upravljanja robotskim kretanjem te za optimizaciju puta [22]. U konkretnom slučaju u ovom radu, Kalmanov filter se koristi za estimaciju kuta rotora sinkronog motora. Gibanje rotora se može prikazati pomoću diferencijalnih jednadžbi drugog reda u ovisnosti o vremenu.

$$\Theta(t) = \Theta_0 + \omega_0 t + \alpha \frac{t^2}{2} \quad (3.15)$$

$$\omega(t) = \omega_0 + \alpha t \quad (3.16)$$

Gdje je:

θ - električki kut koji zatvaraju rotor i stator motora,

ω - kutna brzina motora i

α - kutna akceleracija motora.

Vektor stanja prikazan je matricom $\mathbf{X}[n]$:

$$\mathbf{X}[n] = \begin{bmatrix} \Theta \\ \omega \\ \alpha \end{bmatrix}.$$

Pomoću jednadžbi (3.15) i (3.16) možemo konstruirati matricu prijelaza stanja $\mathbf{A}$.

$$\mathbf{A} = \begin{bmatrix} 1 & \Delta t & \frac{\Delta t^2}{2} \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix}$$

Ako uzmemo u obzir da je vektor pobude $\mathbf{u}[n]$ jednak 0, jednadžbu estimacije vektora stanja $\hat{\mathbf{X}}^- [n]$ možemo prikazati kao:

$$\hat{\mathbf{X}}^- [n] = \begin{bmatrix} 1 & \Delta t & \frac{\Delta t^2}{2} \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{\Theta} \\ \hat{\omega} \\ \hat{\alpha} \end{bmatrix}. \quad (3.17)$$

Ako zanemarimo utjecaj kutne akceleracije na sustav možemo pojednostavniti izraz (3.17).

$$\hat{\mathbf{X}}^- [n] = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{\Theta} \\ \hat{\omega} \end{bmatrix} \quad (3.18)$$

Nakon što smo odredili vektor stanja i prijelaznu matricu $\mathbf{A}$, potrebno je odrediti početne uvjete sustava u trenutku t_0 i vrijednosti kovarijancijske matrice. Ako ne znamo

koji su početni uvjeti, možemo dodijeliti vektoru stanja vrijednosti 0 te će nakon svega nekoliko iteracija filter ispravno računati procjenu sljedećeg stanja. Kovarijancijska matrica je oblika:

$$\mathbf{P}[n] = \begin{bmatrix} \sigma_{\Theta}^2 & \sigma_{\Theta\omega}^2 \\ \sigma_{\omega\Theta}^2 & \sigma_{\omega}^2 \end{bmatrix}. \quad (3.19)$$

Važno je napomenuti kako vrijednosti glavne dijagonale kovarijancijske matrice ne smiju biti jednake 0, odnosno možemo pretpostaviti da su vrijednosti varijanci σ_{Θ}^2 i σ_{ω}^2 u početku jednake 1, dok su ostale vrijednosti matrice jednake 0.

$$\mathbf{P}[n_0] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (3.20)$$

Kako na mjereni signal utječe procesni šum potrebno je odrediti matricu $\mathbf{R}_{WW}[n]$. Procesni šum može biti neovisan između različitih varijabli vektora stanja i tada je matrica $\mathbf{R}_{WW}[n]$ dijagonalna matrica [2].

$$\mathbf{R}_{WW}[n] = \begin{bmatrix} q_{11} & 0 & \cdots & 0 \\ 0 & q_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & q_{kk} \end{bmatrix}$$

No u većini slučajeva varijable vektora stanja ovise jedna o drugoj. Iako model pretpostavlja da je kutna akceleracija α jednaka nuli, svaki šum varijance σ_{α}^2 će utjecati na vrijednosti kutne brzine i pozicije rotora. Kod definiranja varijance procesnog šuma važno je odrediti njenu pravu vrijednost kako bi točnost filtra bila što veća. Kovarijancijska matrica procesnog šuma računa se na sljedeći način.

$$\mathbf{R}_{WW}[n] = \begin{bmatrix} V(\Theta) & COV(\Theta, \omega) \\ COV(\omega, \Theta) & V(\omega) \end{bmatrix}$$

$$V(\omega) = \sigma_{\omega}^2 = E(\omega^2) - \mu_{\omega}^2 = E((\alpha \Delta t)^2) - (\mu_{\alpha} \Delta t)^2 = \Delta t^2 \sigma_{\alpha}^2 \quad (3.21)$$

$$V(\Theta) = \sigma_{\Theta}^2 = E(\Theta^2) - \mu_{\Theta}^2 = E((\omega \Delta t)^2) - (\mu_{\omega} \Delta t)^2 = \frac{\Delta t^4}{4} \sigma_{\omega}^2 \quad (3.22)$$

$$COV(\Theta, \omega) = COV(\omega, \Theta) = E(\Theta \omega) - \mu_{\Theta} \mu_{\omega} = \frac{\Delta t^3}{2} \sigma_{\alpha}^2 \quad (3.23)$$

Tada kovarijancijsku matricu procesnog šuma možemo zapisati kao:

$$\mathbf{R}_{WW}[n] = \begin{bmatrix} \frac{\Delta t^4}{4} & \frac{\Delta t^3}{2} \\ \frac{\Delta t^3}{2} & \Delta t^2 \end{bmatrix} \sigma_{\alpha}^2.$$

Kada je Δt jako mali koristi se navedena matrica. Kada je Δt velik potrebno je integrirati matricu $\mathbf{R}_{WW}[n]$ da bi se dobio kontinuirani model.

$$\mathbf{R}_{CWW} = \int_0^{\Delta t} \mathbf{R}_{WW} dt = \int_0^{\Delta t} \sigma_\alpha^2 \begin{bmatrix} \frac{t^4}{4} & \frac{t^3}{2} \\ \frac{t^3}{2} & t^2 \end{bmatrix} dt = \sigma_\alpha^2 \begin{bmatrix} \frac{\Delta t^5}{20} & \frac{\Delta t^4}{8} \\ \frac{\Delta t^4}{8} & \frac{\Delta t^3}{3} \end{bmatrix}$$

Preostalo je odrediti matricu $\mathbf{H}$, mjernu nesigurnost $\mathbf{R}_{VV}[n]$ i definirati $\mathbf{Y}[n]$. Matricu $\mathbf{H}$ određujemo na temelju toga koju varijablu mjerimo. Kod određivanja kuta motora, mjeri se razlika vremena prolaska rotora između dva susjedna Hallova senzora. Na temelju izmjerenog vremena može se odrediti kutna brzina rotora koje se primjenjuje na Kalmanov filter. Zaključujemo da je $\mathbf{Y}[n]$ mjerena kutna brzina motora rotora. Mjerna nesigurnost, $\mathbf{R}_{VV}[n]$, je pogreška koja se javlja prilikom mjerenja razlike vremena pomoću Hallovog senzora. Tada je matrica $\mathbf{H}$ dimenzija 1×2 .

$$\mathbf{H} = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

Kod estimacije kuta rotora, filter na temelju izmjerene brzine i razlike vremena određuje kut rotora između dva Hallova senzora. Jednadžbe predikcije i korekcije rade u različitom režimu. Predikcija kuta rotora se odvija za vrijeme PWM ciklusa, a korekcija estimirane vrijednosti se koristi prilikom svakog novog mjerenje Hallovog senzora.

4. MATLAB analiza Kalmanovog filtra i metode kalibracije

4.1. Kalman filter

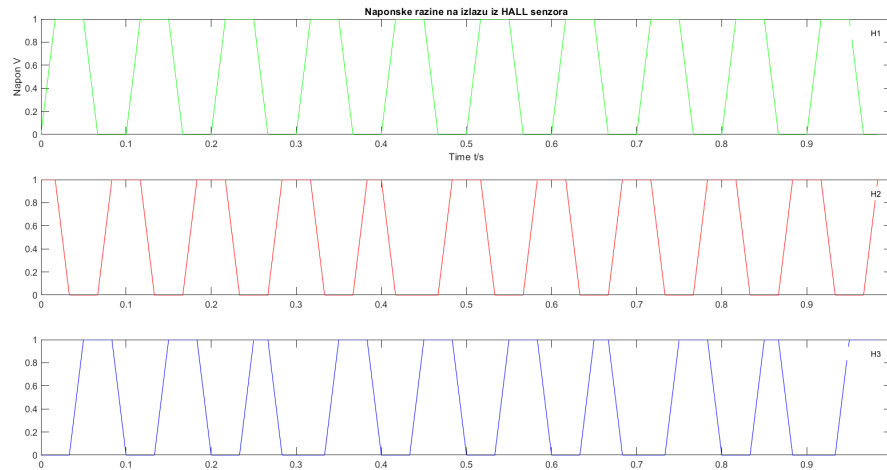
Algoritam Kalmanovog filtra proveden je u MATLAB okruženju. U nastavku će biti navedeni i grafički objašnjeni ključni dijelovi algoritma.

U MATLAB algoritmu na svaki novi pojavljeni brid, jednog od tri signala iz senzora, ažurira se prijelazna matrica stanja $\mathbf{A}$ i na temelju razlike vremena računa kovarijancijska matrica procesnog šuma, R_{WW} . Zatim se izračunavaju kutevi dok se ne pojavi novi brid, nakon čega se korigira nova vrijednost procijene. MATLAB algoritam Kalmanovog filtra izgleda ovako:

```
1                                     #Kalmanov filter
2
3  for i = 1:1:length(dt)
4      A = [ 1 delta_t ; 0 1];
5      Rww = [0.25*delta_t^4 0.5*delta_t^3;
6             0.5*delta_t^3 delta_t^2]*sigma_W^2; #Kovarijancijska
7      matrica
8      for j = 1:1:60
9          # Jednadzbe predikcije
10         X = A*X;
11         P = A*P*transpose(A) + Rww;
12     end
13     # Jednadzbe korekcije
14     Z = wr(i); #mjerena vrijednost
15     K = P*H' / (H*P*H' + Rvv);
16     X = X + K*(Z-H*X);
17     P = (eye(size(A))-K*H)*P;
18 end
```

Listing 4.1: Kalmanov filter algoritam

Izlaz Hallovih senzora, pomaknutih za 120° , prikazan je na slici 4.1. Na temelju izlaza iz senzora i prema binarnom kodu, koji je prikazan u prošlom poglavlju, možemo detektirati električni kut rotora. Binarni kod pojedinih kuteva dan je u tablici 4.1.



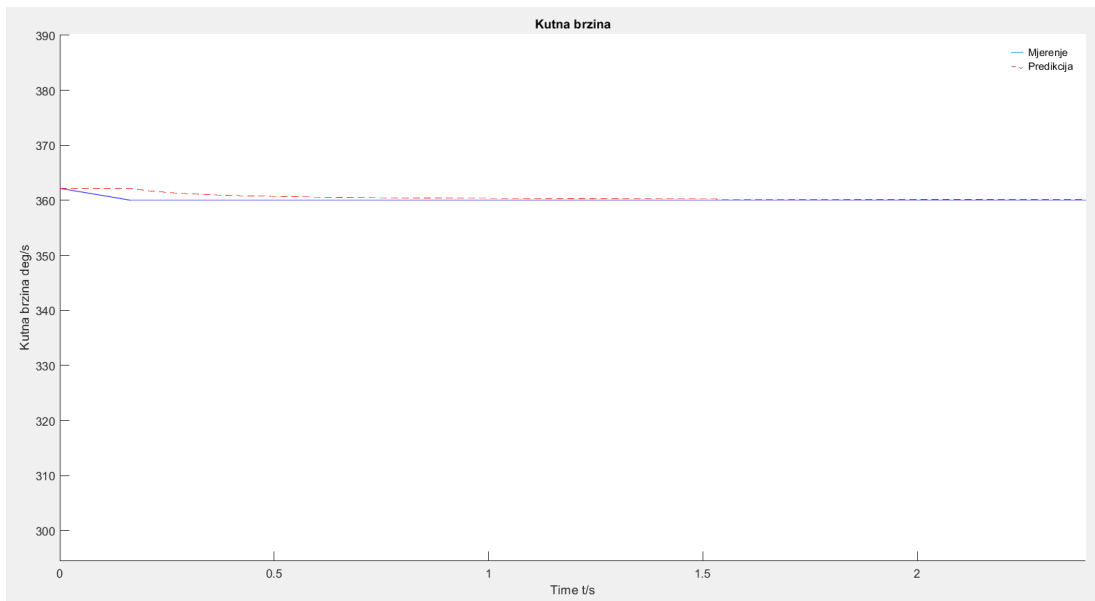
Slika 4.1: Izlazni signali Hallovih senzora pomaknuti za 120° .

$0^\circ - 60^\circ$	100
$60^\circ - 120^\circ$	110
$120^\circ - 180^\circ$	010
$180^\circ - 240^\circ$	011
$240^\circ - 300^\circ$	001
$300^\circ - 360^\circ$	101

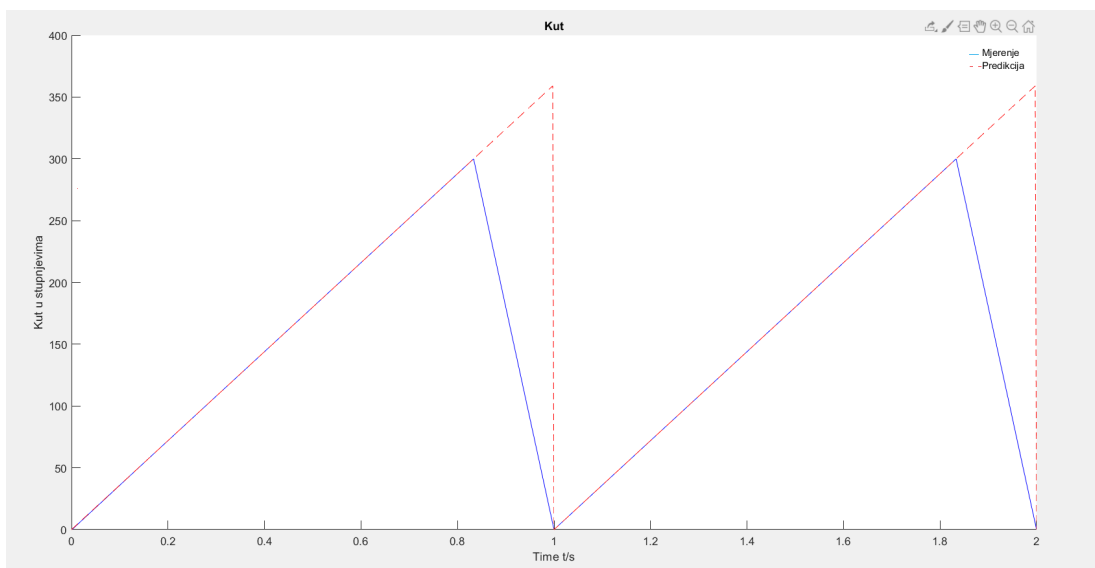
Tablica 4.1: Binarni kod izlaznih signala Hallovih senzora za određeni kuta rotora

U sljedećim primjerima uzet ćemo u obzir idealno stanje i provjeriti što se događa s predviđanjem Kalmanovog filtra u ovisnosti o kovarijancijskoj matrici procesnog šuma.

Prvi primjer prikazuje kada na rotor ne djeluje nikakav šum i na slikama 4.2 i 4.3 možemo vidjeti kako filter idealno estimira sljedeću vrijednost. Na grafovima možemo vidjeti da estimator u početku ima pogrešku, ali s vremenom se filter prilagođava i prima pravu vrijednost. Hall senzori ne mogu mjeriti kuteve između dva senzora, stoga ima prijelaz s 300° na 360° , odnosno ponovo kreće od 0° : Estimator može procijeniti vrijednost između dva senzora.

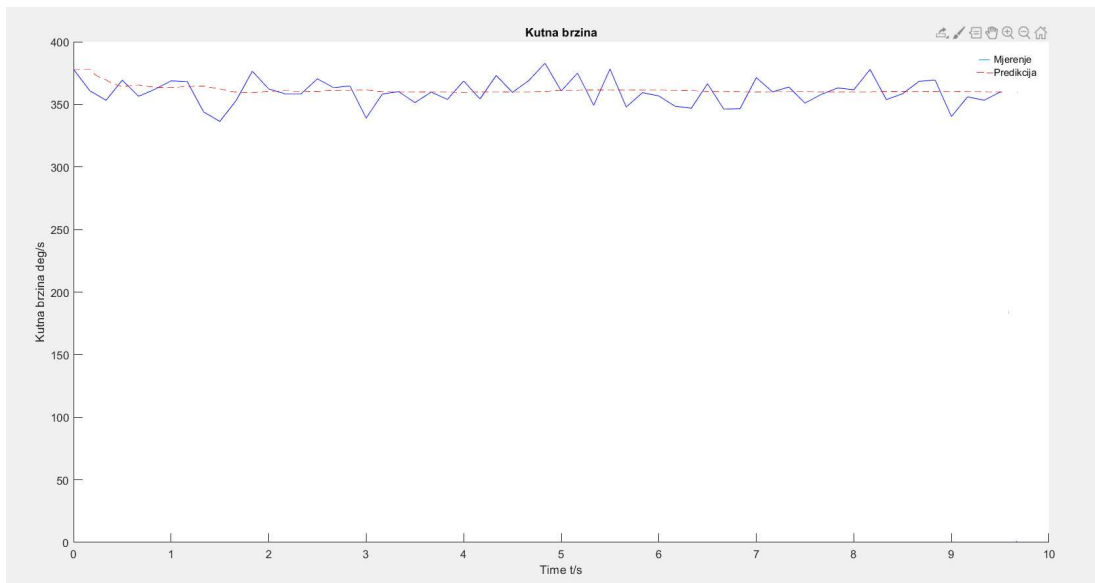


Slika 4.2: Estimacija brzine bez procesnog šuma.



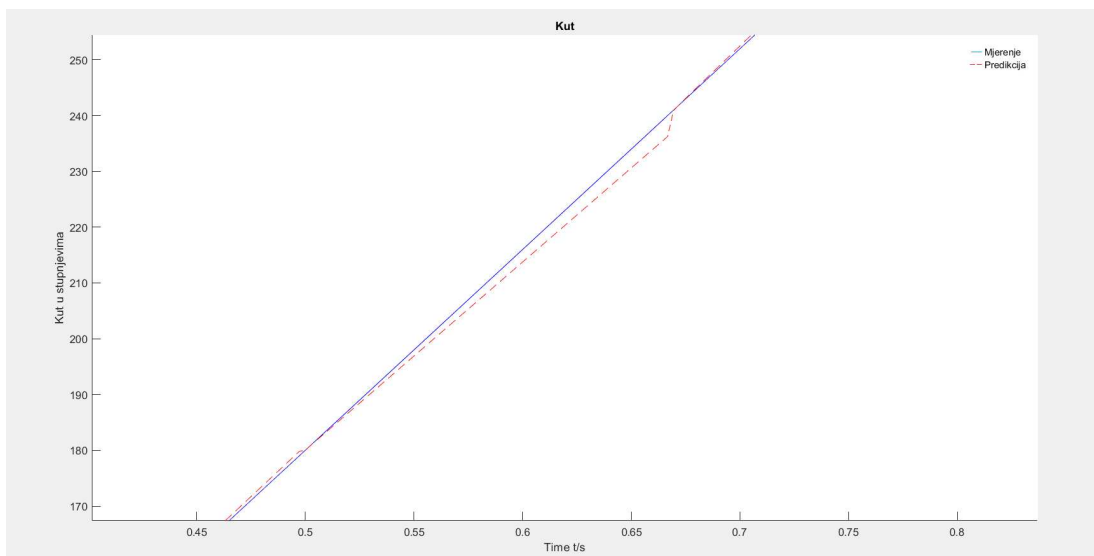
Slika 4.3: Estimacija kuta bez šuma.

Kada na mjereni signal primijenimo šum, a ne koristimo kovarijancijsku matricu procesnog šuma, estimator kuta neće pravilno računati sljedeća stanja. Na slici 4.4 prikazano je kako se Kalmanov filter ne prilagođava stvarnom signalu ako ne koristimo matricu R_{WW} . Filter će procijeniti prvu vrijednost koristeći prijelaznu matricu A , ali se neće moći prilagoditi na šum.

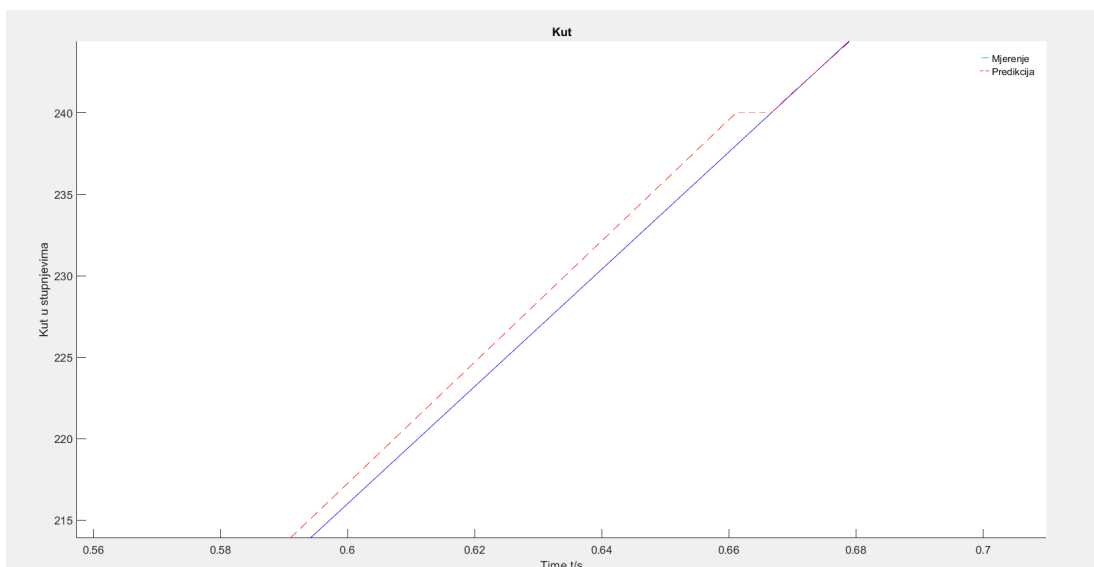


Slika 4.4: Estimacija brzine kada je na signal primijenjen šum, a zanemari se kovarijancijska matrica procesnog šuma u Kalmanovom filtru.

Jednako vrijedi i ako uzmemo u obzir premalu vrijednost varijance procesnog šuma, σ_W^2 . Druga stvar na koju je potrebno obratiti pozornost je izračun kuta rotora. Ako estimiramo krive vrijednosti mjerene brzine okretanja rotora, Kalmanov filter će računati krive vrijednosti kuta u vremenu. Postoje dva scenarija koja je potrebno izbjeći. Prvi scenarij je kada je procijenjena brzina okretanja rotora manja od očekivane. Tada će estimator računati krive vrijednosti kuta u trenutku t_n . Prilikom pojave novog brida jednog od tri Hallova senzora estimirana vrijednost će biti manja od očekivane i potrebno je ažurirati novo početno stanje ovisno o tome koji senzor je aktivan. Primjer takvog scenarija vidimo na slici 4.5. Drugi scenarij se odnosi na vrijednost veću od očekivane. Tada estimator prebrzo računa vrijednosti kuta rotora i dolazi do prekoračenja opsega. Pod pojmom opseg misli se na to da estimator prelazi vrijednost sljedećeg predviđenog kuta na kojem se nalazi Hallov senzor, odnosno želi prijeći u novi sektor. Tada je potrebno čekati i ne dopustiti estimaciju sljedećeg stanja sve dok se ne pojavi novi brid signala sa senzora. Primjer takvog scenarija nalazi se na slici 4.6.

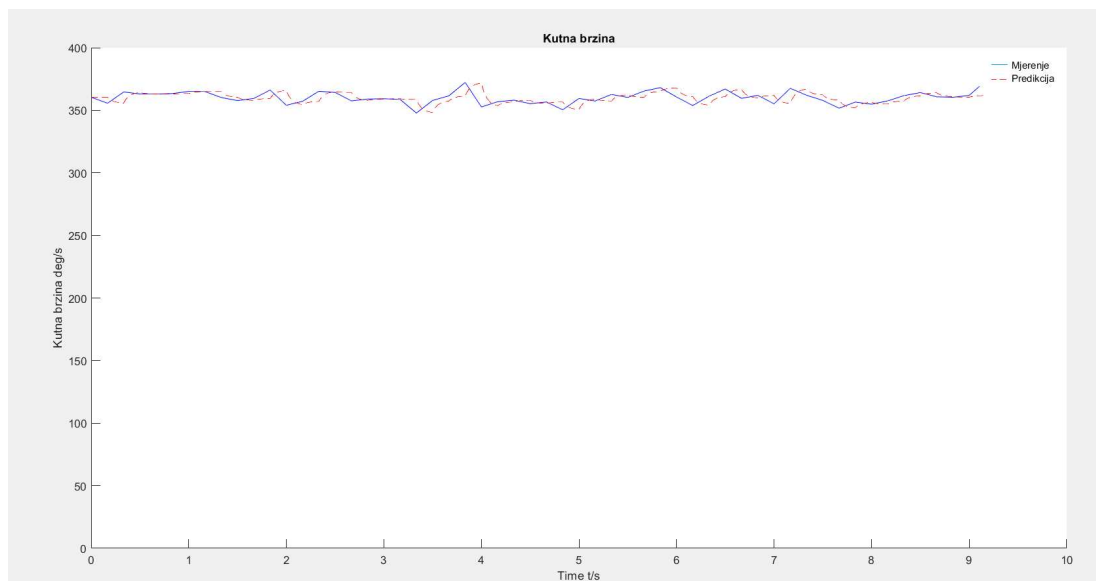


Slika 4.5: Estimirana vrijednost je manja od trenutnog kuta i potrebno je ažurirati početno stanje s pozicijom trenutnog Hallovog senzora.

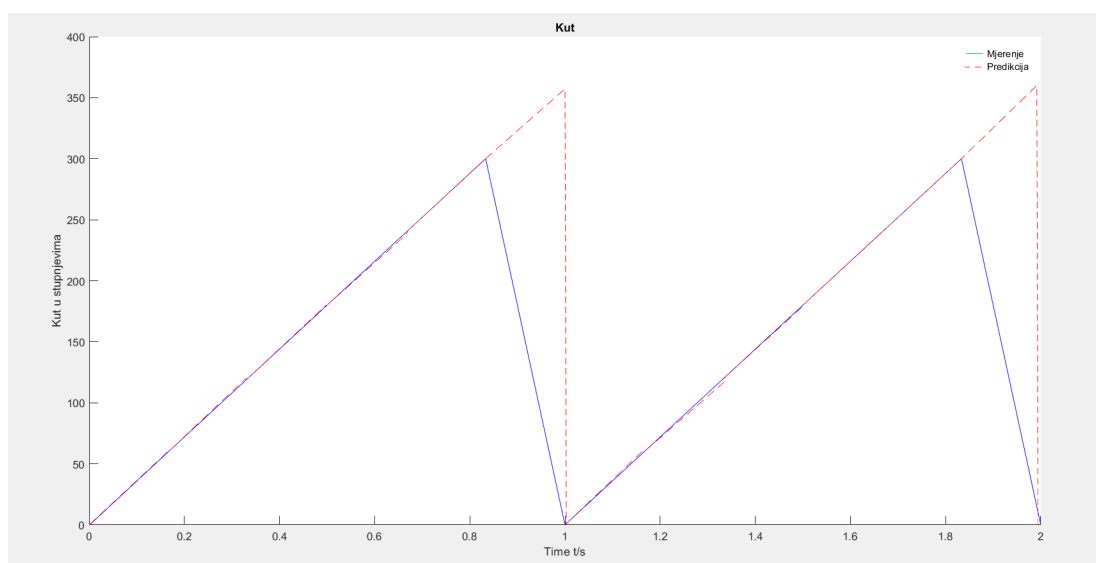


Slika 4.6: Estimirani kut želi prijeći u novi sektant.

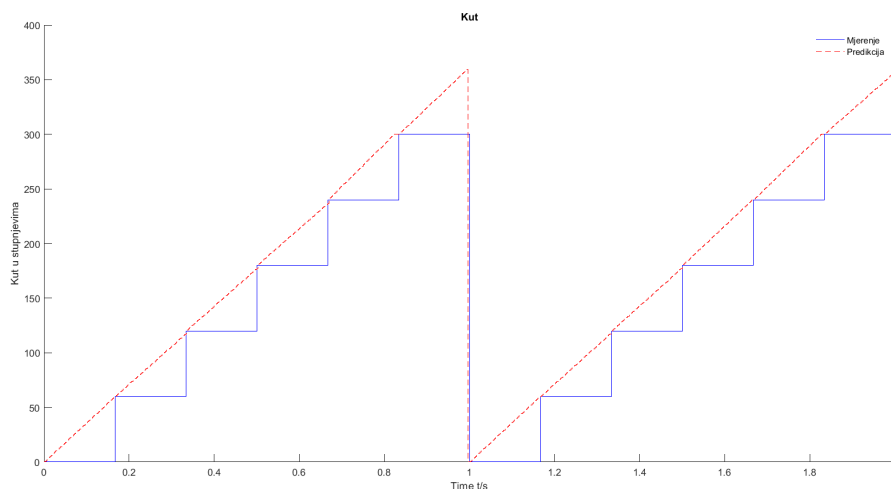
Možemo zaključiti da odabirom pravih vrijednosti kovarijancijske matrice procesnog šuma i dobrim algoritmom možemo ispraviti pogreške. Iako su nagle promjene učestale u stvarnim procesima, Kalmanov filter može se brzo prilagoditi na osnovu prošlih stanja i točno predvidjeti novo stanje. Filter koji se prilagođava sljedećim stanjima bez obzira na šum prikazan je na slici 4.7 i 4.8. Bolji prikaz odnosa estimatora kuta i mjerenja Hallovih senzora možemo vidjeti na slici 4.9. Koristeći funkciju *stairs()* možemo vidjeti da Hallovi senzori ne mogu mjeriti kutove unutar sektanta, stoga je potrebno koristiti estimator.



Slika 4.7: Estimacija brzine okretanja rotora koji je pod utjecajem procesnog šuma.



Slika 4.8: Estimacija kuta rotora.



Slika 4.9: Usporedba mjerenja Hallovih senzora i estimatora kuta rotora.

4.2. Kalibracija

U stvarnom svijetu možemo pretpostaviti da Hallovi senzori nisu smješteni točno onako kako smo zamislili te postoje odstupanja od idealnih vrijednosti. Na primjer Hallov senzor koji se nalazi na poziciji kuta od 60° može biti pomaknut za koji stupanj ulijevo ili udesno te će iznositi npr. 63° . Ovakve nesavršenosti dovode do pogreške u mjerenju stvarne pozicije rotora i brzine okretanja. Da bismo izbjegli te pogreške potrebno je kalibrirati sustav. Kalibracija sustava je provedena tako da se na sinkroni motor dovede napon poznatog iznosa i čeka se dok sustav ne postane stabilan. Kada sustav postane stabilan mjeri se vrijeme potrebno da rotor prijeđe puni električni krug. Vrijednosti vremena koje je potrebno rotoru da dođe do jednog Hallovog senzora sprema se u tablicu. Nakon nekoliko okreta rotora potrebno je usrednjiti vrijednosti vremena za pojedini Hall senzor. Nakon što znamo koliko je vremena potrebno da rotor dođe do svakog od senzora u pojedinom sekstantu može se izračunati prava vrijednost pozicije senzora.

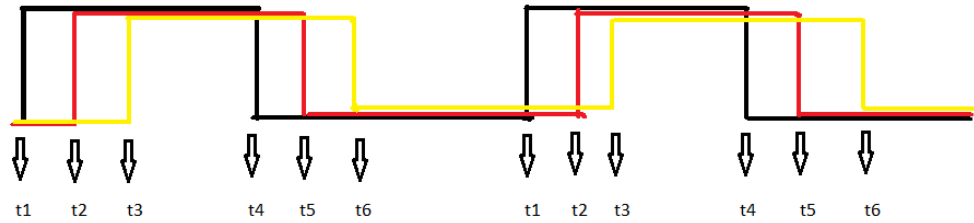
Srednja vrijednost vremena pojave pojedinog brida može se računati prema:

$$\overline{t_n} = \sum_{i=1}^N t_n, n = 1, \dots, 6. \quad (4.1)$$

Period T je vrijeme potrebno da rotor napravi puni električni okretaj. Poznavajući vrijednosti vremena potrebnog da rotor puni krug i vrijeme potrebno do pojedinog senzora možemo izračunati stvarnu poziciju senzora. jednačba kojom se može računati

stvarna pozicija dana je izrazom:

$$\alpha_n = \frac{\overline{t_n}}{T} 360^\circ, n = 1, \dots, 6. \quad (4.2)$$



Slika 4.10: Vrijeme pojave pojedinog brida.

5. Implementacija algoritma na STM32fxx

Kao što smo u prethodnim poglavljima objasnili, poziciju rotora motora moguće je detektirati pomoću Hallovih senzora. Hallovi senzori imaju digitalni izlaz zbog čega se može direktno koristiti za ulaz na ugradbena računala. Kao ugradbeni računalni sustav koristi se STM32F407VG razvojna pločica. U nastavku će biti detaljno objašnjeno kako se postaviti razvojnu pločicu i povezati je s Hallovim sensorima. Također, na ugradbenom računalu moguće je implementirati algoritam Kalmanovog filtra i kalibracije koristeći programski jezik C.

Za početak je potrebno konfigurirati razvojnu pločicu tako da obrađuje izlazne signale s Hallovih senzora za daljnje korištenje. Jedan od načina obrade signala je koristeći *timer*. *Timer* omogućuje mjerenje vremena koje prođe između pojave dva susjedna brida signala Hall senzora. Prije nego što se konfigurira *timer* potrebno je postaviti periferiju za ulaze i postaviti ulazne pinove za potrebe mjerenja. U tablici 7. u [16] nalazi se popis pinova i funkcije koje podržavaju. Važno je da određeni pin podržava sučelje za *timer*. Sljedeći dio programskog koda opisuje način postavljanja ulaza za senzore.

```
1 // enable pin clock , GPIOx pin 12,13,14 for timer x
2 RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOx , ENABLE);
3
4 GPIO_InitStruct.GPIO_Pin = GPIO_Pin_12 | GPIO_Pin_13 |
5                             GPIO_Pin_14; //Hall sensors input
6 GPIO_InitStruct.GPIO_Mode = GPIO_Mode_AF; // Alternate function
7 GPIO_InitStruct.GPIO_Speed = GPIO_Speed_50MHz;
8 GPIO_InitStruct.GPIO_OType = GPIO_OType_PP;
9 GPIO_InitStruct.GPIO_PuPd = GPIO_PuPd_UP;
10 GPIO_Init(GPIOx, &GPIO_InitStruct);
11
12 // Connect TIM4 pin (Px12,13,14) to AF2
13 GPIO_PinAFConfig(GPIOx, GPIO_PinSource12, GPIO_AF_TIMx);
14 GPIO_PinAFConfig(GPIOx, GPIO_PinSource13, GPIO_AF_TIMx);
```

```
14 GPIO_PinAFConfig(GPIOx, GPIO_PinSource14, GPIO_AF_TIMx);
```

Listing 5.1: Inicijalizacija ulazno izlaznog sklopovlja.

Funkcijom `RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOx, ENABLE)` omogućili smo rad periferije. Pinove koje želimo koristiti omogućujemo dodjeljivanjem vrijednosti `GPIO_InitStruct.GPIO_Pin = GPIO_Pin_x`. Važno je napomenuti da kada koristimo *timer*, pinove je potrebno postaviti kao alternativnu funkciju. U suprotnom odabrani pinovi neće biti povezani sa sučeljem za *timer*, nego će djelovati kao ulazni pinovi (`GPIO_Mode_AF`). Uz to je dodatnom funkcijom potrebno povezati pinove s *timerom* `GPIO_PinAFConfig(GPIO, GPIO_PinSource, GPIO_AF_TIM)`. Zatim je potrebno omogućiti *clock* za periferiju *timera*. Bitnu ulogu ima frekvencija kojom radi određena periferija. Prema frekvenciji *clocka* možemo odrediti parametre kao što su *prescaler* i *period timera*. Iz tablice 4. u [16] možemo vidjeti da `TIMx` radi na maksimalnoj frekvenciji od 84 MHz. Tada vrijednosti *prescalera* i *perioda* možemo računati prema [12]:

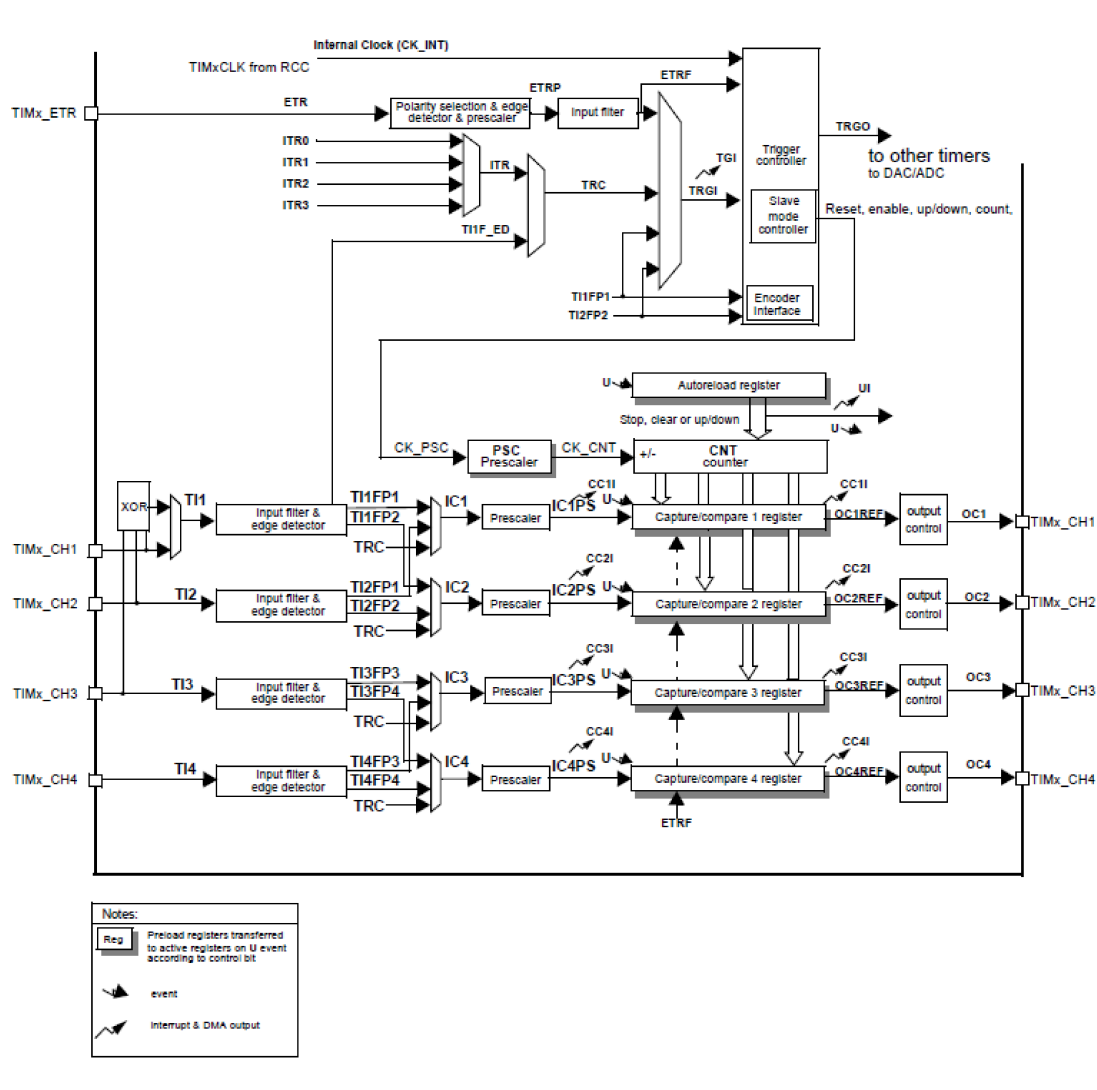
$$f = \frac{84MHz}{(PSC + 1)(ARR + 1)}, \quad (5.1)$$

gdje je f frekvencija kojom želimo da broji brojač *timera*. *ARR* je željeni period čije vrijednosti brojač poprima, brojeći od 0 do *ARR*. Kada *ARR* dosegne maksimalnu vrijednost, počinje brojati od 0. *PSC* je vrijednost *prescalera*. *Timer* postavljamo da broji prema rastućoj vrijednosti. Odabir parametara prikazan je djelom programskog koda:

```
1 // enable timer clock, TIMEx
2 RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIMx, ENABLE);
3
4 // initialization timerx
5 TIM_TimeBaseStructure.TIM_Prescaler = PSC;
6 TIM_TimeBaseStructure.TIM_CounterMode = TIM_CounterMode_Up;
7 TIM_TimeBaseStructure.TIM_Period = ARR;
8 TIM_TimeBaseStructure.TIM_ClockDivision = 0;
9 TIM_TimeBaseStructure.TIM_RepetitionCounter = 0;
10 TIM_TimeBaseInit(TIM4, &TIM_TimeBaseStructure);
```

Listing 5.2: Inicijalizacija *timera*.

Timer omogućuje rad u *Input capture and compare* načinu. Na slici 5.1 nalazi se blok shema sučelja za `TIMx`.



Slika 5.1: Blok shema sučelja za timer [11].

Na slici možemo vidjeti da TIMx sadrži četiri kanala koje možemo koristiti kao ulaze. Prva tri kanala povezana su preko XOR funkcije što je pogodno za mjerenja pomoću tri Hallovog senzora. Na pojavu svakog brida, jednog od Hallovog senzora, zastavica TI1F_ED će se aktivirati. Sučelje za Hallove senzore možemo aktivirati funkcijom `TIM_SelectHallSensor(TIMx, ENABLE)`. Kako kao ulaz u timer želimo da bude pojava svakog novog brida. To možemo učiniti osluškujući zastavicu TI1F_ED i funkcijom `TIM_SelectInputTrigger(TIMx, TIM_TS_TI1F_ED)`. Kada se pojavi svaki novi brid timer će aktivirati prekid, stoga je timer potrebno konfigurirati u *Input capture* načinu rada. Prekid će se aktivirati na pojavu oba brida, padajući i rastući. Dio programskog koda je sljedeći:

```
1 TIM_ICInitStructure.TIM_Channel = TIM_Channel_1;
2 TIM_ICInitStructure.TIM_ICPolarity = TIM_ICPolarity_BothEdge;
```



```

3 TIM_ICInitStructure.TIM_ICSelection = TIM_ICSelection_TRC;
4 TIM_ICInitStructure.TIM_ICPrescaler = TIM_ICPSC_DIV1;
5 TIM_ICInitStructure.TIM_ICFilter = 0x0;
6 TIM_ITConfig(TIMx, TIM_IT_CC1, ENABLE);

```

Listing 5.3: Konfiguracija *Input capture* načina rada.

Nakon konfiguracije *timera* potrebno je konfigurirati prekid s odgovarajućim prioritetima istiskivanja kao:

```

1 NVIC_InitStructure.NVIC_IRQChannel = TIMx_IRQn;
2 NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0;
3 NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0;
4 NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
5 NVIC_Init(&NVIC_InitStructure);

```

Listing 5.4: Konfiguracija prekida.

U prekidnoj funkciji obrađuje se podatak koji je izazvao prekid. Na pojavu svakog brida vrijednost brojača `TIMx->CNT` sprema se u pomoćnu varijablu. Vrijednost varijable `TIMx->CNT` mijenja se od 0 do `ARR`, nakon čega se resetira na 0. Kako bi znali na kojoj se točno poziciji nalazi rotor motora potrebno je očitati vrijednosti ulaza u *timer*. Ulazna stanja na pojedino pinu možemo očitati pomoću funkcije za čitanje stanja ulaznog bita `GPIO_ReadInputDataBit(GPIOx, GPIO_Pin_x)`. Prema tablici 4.1 i očitanim stanjima ulaza možemo provjeriti na kojoj se poziciji nalazi rotor. Kao najbitnija stavka prekidne funkcije je razlika vremena između pojave dva susjedna brida signala. Uzimajući u obzir da brojač broji do vrijednosti `ARR`, nekada može doći do preljeva. Preljev označuje da je vrijednost brojača resetirana na 0 i počinje ponovo brojati, što ponekad može predstavljati pogrešku. No programskim kodom možemo ispraviti pogrešku. Ako je vrijednost prethodnog vremena veća od trenutnog vremena, tada jednostavno pribrojimo sadašnjem vremenu vrijednost punog perioda, `ARR`. Razliku vremena potrebno je pretvoriti u sekunde jer se vrijednosti varijabli `ARR` i `TIMx->CNT` izražavaju u Hz. Poznavajući vrijednost frekvencije kojom brojač broji možemo izračunati stvarnu vrijednost. Kada izračunavamo vrijednost brzine motora moramo uzet u obzir da su pozicije Hallovih senzora određene unaprijed. Može doći do pogreške u računu kada motor prelazi iz zadnjeg sekstanta u prvi zbog premotavanja faze. Potrebno je osigurati da niti u jednom trenutku razlika kuteva ne bude veća od dva sekstanta, odnosno 120° . Izračunatu razliku vremena moramo podijeliti sa 60 kako bi mogli računati predikciju u vrijednosti od 1° . Za potrebe kalibracije u posebnu varijablu sprema se razlika vremena od `N` uzoraka, između pojave dva brida. Podaci se u obrađuju u posebnom zadatku u glavnom programu. Prekidna funkcija dana je

programskim kodom:

```
1 void TIMx_IRQHandler(void) {
2
3     Hall_time = TIMx->CNT;
4     Hall1 = GPIO_ReadInputDataBit(GPIOx, GPIO_Pin_12);
5     Hall2 = GPIO_ReadInputDataBit(GPIOx, GPIO_Pin_13);
6     Hall3 = GPIO_ReadInputDataBit(GPIOx, GPIO_Pin_14);
7
8     if (TIM_GetITStatus(TIMx, TIM_IT_CC1) != RESET ) {
9         TIM_ClearITPendingBit(TIMx, TIM_IT_CC1);
10        NVIC_ClearPendingIRQ(TIMx_IRQn);
11
12        if (Hall1 == 1 && Hall2 == 0 && Hall3 == 1 )
13            fi = theta300;
14        else if (Hall1 == 0 && Hall2 == 1 && Hall3 == 0 )
15            fi = theta120;
16        else if (Hall1 == 1 && Hall2 == 1 && Hall3 == 0 )
17            fi = theta60;
18        else if (Hall1 == 0 && Hall2 == 0 && Hall3 == 1 )
19            fi = theta240;
20        else if (Hall1 == 0 && Hall2 == 1 && Hall3 == 1 )
21            fi = theta180;
22        else
23            fi = theta0;
24    }
25
26    if( Hall.time >= t_p) dt = Hall.time - t_p;
27    else dt = ARR + Hall.time - t_p;
28    delta_t = (double)dt/f;
29
30    if(k < N){
31        t_cal[k] = dt;
32        k++;
33    }
34
35    if ( delta_t != 0){
36        if ( (fi_p == theta300 || fi_p == theta240) && (fi ==
37            theta0 || fi == theta60))
38            speed_mea = (double)(fi + 360 - fi_p)/delta_t;
39        else if ( (fi_p == theta0 || fi_p == theta60) && (fi ==
40            theta300 || fi == theta240))
41            speed_mea = (double)(fi - (fi_p + 360))/delta_t;
42        else
```

```

41         speed_mea = (double)(fi - fi_p) / delta_t;
42     }
43
44     delta_t = delta_t / 60;
45     t_p = Hall_time;
46     fi_p = fi;
47 }

```

Listing 5.5: Prekidna funkcija.

Algoritam Kalmanovog filtra izvodi se u glavnom programu. Algoritam je razdvojen na dvije funkcije jer funkcija predikcije i korekcije ne rade u istom režimu rada. Na svaku pojavu brida potrebno je ažurirati prijelaznu matricu stanja novom razlikom vremena. Korekcija se izvodi na pojavu nove mjerene vrijednosti brzine. U algoritmu Kalmanovog filtra potrebno je voditi brigu da estimirana vrijednost ne prijeđe trenutni sekstant u kojem se rotor nalazi. Programskim kodom je potrebno osigurati da se to ne dogodi, stoga na pojavu novog brida potrebno je definirati novu početnu vrijednost od koje započinje predikcija i krajnju vrijednost koju algoritam ne smije prijeći. Ako estimator želi prijeći zadanu krajnju vrijednost potrebno je fiksirati izlaz filtra u spomenutoj točki. Programski kod je implementiran raspisivanjem matričnih jednadžbi iz poglavlja 3.

```

1 void kalman_predict(double dt) {
2     p_kalman.P[0][0] += p_kalman.P[0][1]*dt + p_kalman.P[1][0]*dt +
                        p_kalman.P[1][1]*dt*dt +
                        0.25*dt*dt*dt*dt*p_kalman.sigma_w;
3     p_kalman.P[0][1] += p_kalman.P[1][1]*dt +
                        0.5*dt*dt*dt*p_kalman.sigma_w;
4     p_kalman.P[1][0] += p_kalman.P[1][1]*dt +
                        0.5*dt*dt*dt*p_kalman.sigma_w;
5     p_kalman.P[1][1] += dt*dt*p_kalman.sigma_w;
6 }

```

Listing 5.6: Implementacija matričnih jednadžbi Kalmanovog filtra u programskom jeziku C. Jednadžbe predikcije.

```

1 void kalman_update(double speed_mea) {
2     S = p_kalman.P[1][1] + p_kalman.sigma_v;
3     K[0] = p_kalman.P[0][1]/S;
4     K[1] = p_kalman.P[1][1]/S;
5     p_kalman.angle = p_kalman.angle + K[0]*(speed_mea -
                        p_kalman.speed);
6     p_kalman.speed = p_kalman.speed + K[1]*(speed_mea -
                        p_kalman.speed);

```

```

7     p_kalman.P[0][0] -= K[0]*p_kalman.P[1][0];
8     p_kalman.P[0][1] -= K[0]*p_kalman.P[1][1];
9     p_kalman.P[1][0] -= K[1]*p_kalman.P[1][0];
10    p_kalman.P[1][1] -= K[1]*p_kalman.P[1][1];
11 }

```

Listing 5.7: Implementacija matričnih jednadžbi Kalmanovog filtra u programskom jeziku C. Jednadžbe korekcije.

Fiksiranje estimatora u krajnju točku možemo djelom programskog koda:

```

1     if (p_1kalman.angle > cilj) p_kalman.angle = cilj;

```

Listing 5.8: Kada estimirana vrijednost želi prijeći trenutni sekstant - ispitivanje vrijednosti.

Kao što smo rekli, funkcije predikcije i korekcije se izvode u glavnom programu. Koristeći FreeRTOS možemo izvoditi zadatke u stvarnom vremenu uzimajući u obzir istiskivanje zadataka većeg prioriteta. Estimacija kuta se koristi u FOC-u za reguliranje odnosa d i q osi, stoga je potrebno odraditi zadatak u što kraćem periodu kako ne bi došlo do blokiranja sustava. Za raspoređivanje zadataka koristi se *Task scheduler*. Primjer definiranja zadataka u glavnom programu dan je sljedećim programskim odjeljkom:

```

1     if (xBinarySemaphore != NULL){
2         xTaskCreate(vKalmanTask, "Kalman filter",
3             configMINIMAL_STACK_SIZE, NULL, 3, NULL);
4         vTaskStartScheduler();
5     }

```

Listing 5.9: Definiranje zadatka u FreeRTOS-u [9].

Binarni semafori onemogućuju izvođenje zadataka višeg prioriteta i istiskivanje zadataka nižeg, kako ne bi došlo do izgladnjivanja. Kada zadatak zauzme semafor, zadatak višeg prioriteta može krenuti s izvođenjem tek kada je semafor otpušten.

Kalmanov filter omogućuje estimaciju kuta rotora motora. Prikazani programski kodovi omogućuju estimaciju kuta kada se motor okreće u pozitivnom smjeru. Kada bi se motor okretao u negativnom smjeru potrebno bi bilo uzeti u obzir i smjer kretanja, provjeravajući je li vrijednost brzine pozitivna ili negativna. Potrebno je u obzir uzeti i nesavršenosti sustava te provesti kalibraciju koristeći jednadžbe iz prethodnog poglavlja, 4.1 i 4.2. Kalibracijom je moguće ukloniti pogreške nastale zbog odstupanja položaja Hallovih senzora od stvarnih vrijednosti. Svakako treba voditi računa o definiranju vrijednosti ARR registra i *prescalera* kako bi se mjerile ispravne vrijednosti razlike vremena. Smanjenjem pogrešaka može se povećati efikasnost vektorskog upravljanja motorima.

Za potrebe kalibracije je potrebno kreirati poseban zadatak u glavom programu. U tom zadatku će se obrađivati podaci spremljeni u varijablu `t_cal[N]`. U početku je potrebno rotor motora postaviti u početni položaj 0. Kada se pokrene rad motora, rotor će izazvati pojavu signala na Hallovim senzorima i svakom pojavom brida pojavit će se prekid. U prekidu se u polje spremaju vrijednosti razlike vremena svakog puta kada se prekid pojavi. Nakon što se prikupi N podataka, zadatak je spreman za obradu. Zadatak u glavom programu čeka određeni period vremena dok se ne ispuni polje podacima. Kada smo sigurni da je polje ispunjeno, može se računati prosječna vrijednost vremena potrebnog da rotor stigne do određenog senzora i napravi puni kut. Kada znamo da puni krug mora imati 6 ulaza u prekid, tada prosječno vrijeme može se računati kao:

```

1 void avarage(int t[N]) {
2
3     for(int k = 0; k < N; k = k + 6){
4         t_n[0] += t[k];
5     }
6     for(int k = 1; k < N; k = k + 6){
7         t_n[1] += t[k] + t[k-1];
8     }
9     for(int k = 2; k < N; k = k + 6){
10        t_n[2] += t[k] + t[k-1] + t[k-2];
11    }
12    for(int k = 3; k < N; k = k + 6){
13        t_n[3] += t[k] + t[k-1] + t[k-2] + t[k-3];
14    }
15    for(int k = 4; k < N; k = k + 6){
16        t_n[4] += t[k] + t[k-1] + t[k-2] + t[k-3] + t[k-4];
17    }
18    for(int k = 5; k < N; k = k + 6){
19        t_n[5] += t[k] + t[k-1] + t[k-2] + t[k-3] + t[k-4] + t[k-5];
20    }
21    for(int k = 0; k < N; k++){
22        T += t[k];
23    }
24    t_n[0] = t_n[0]/3;
25    t_n[1] = t_n[1]/3;
26    t_n[2] = t_n[2]/3;
27    t_n[3] = t_n[3]/3;
28    t_n[4] = t_n[4]/3;
29    t_n[5] = t_n[5]/3;
30    T = T/3;

```

```
31  
32     theta[0] = t_n[0]/T*360;  
33     theta[1] = t_n[1]/T*360;  
34     theta[2] = t_n[2]/T*360;  
35     theta[3] = t_n[3]/T*360;  
36     theta[4] = t_n[4]/T*360;  
37     theta[5] = t_n[5]/T*360;  
38 }
```

Listing 5.10: Kalibracija Hallovih senzora.

Kalibraciju je potrebno provesti na stvarnom sustavu i ustvrditi stvarna odstupanja Hallovih senzora od idealnih vrijednosti kuteva.

6. Zaključak

Svojstvo sinkronih motora je sinkroniziranost frekvencije struja kroz namote statora i frekvencije rotacije rotora. Kao posebna vrsta sinkronih motora je sinkroni motor s permanentnim magnetima. Permanentni magneti nalaze se u rotoru motora te motor može sadržavati više parova polova magneta, ovisno o primjeni. U vektorskom upravljanju motorima potrebno je promatrati električni kut između statora i rotora, no za druge potrebe potrebno je uzeti u obzir mehanički kut rotora. Mehanički kut rotora se izražava kao omjer električnog kuta rotora i broj parova polova magneta. Struje kroz statore motora pomaknute su međusobno za 120° . Upravljanje motorima sastoji se od nekoliko glavnih jedinica. To su inverter, sinkroni motor, kontrolna jedinica te senzori. Preklapanjem sklopki invertera možemo generirati napone za upravljanje izmjeničnim motorima. Određenom duljinom trajanja PWM ciklusa moguće je stvoriti željeni napon. Field Orientated Control (FOC) se temelji na projekciji koja transformira trodimenzionalni ovisan o vremenu u dvodimenzionalan, vremenski neovisan sustav. Za transformaciju iz jednog sustava u drugi koriste se Clarkeina transformacija koja projektira sustav (a, b, c) u sustav (α , β). Parkova transformacija sustav (α , β) transformira u rotirajući koordinatni sustav (d, q). Za potrebe Parkove transformacije moramo znati vrijednosti trenutne pozicije rotora, odnosno njegov električni kut, te iznose statorskih struja. Za detekciju kuta rotora koriste se Hallovi senzori koji na temelju Hallovog senzora detektiraju magnetsko polje uzrokovano permanentnim magnetima u rotoru. Prolaskom rotora pored senzora generira se digitalni izlazni signal. Hallovi senzori smješteni su na poznatim pozicijama te kada se rotor izazove promjenu stanja Hallovih senzora, možemo odrediti trenutnu poziciju rotora. Problem korištenja Hallovih senzora je to što ne mogu mjeriti kuteve između dva senzora. Zbog toga se koristi Kalmanov filtar kao estimator kuta. Kalmanov filtar na temelju vektora stanja i matrica koje predstavljaju sustav može predvidjeti buduće stanje kuta rotora. Estimator se može implementirati na ugradbenim računalima. Korištenje filtra može poboljšati odnos q i d komponente i držati ih na razmaku od 90° . Algoritam u radu zanemaruje komponentu kutne akceleracije te uslijed naglih ubrzanja motora neće moći dati dobru

estimaciju kuta. Kako Hallovim senzorima mjerimo razliku vremena čak i najmanje pogreške u mjerenju imaju utjecaj na sustav. Korištenjem komponente kutne akceleracije model bi bio osjetljiviji na pogrešku zbog kvadratne funkcije vremena. Iako je korišten model sustava prvog reda, može se reći da estimator daje dobru procjenu sljedećeg stanja. Kod definiranja kovarijancijske matrice procesnog šuma zadana je konstantna, pretpostavljena vrijednost šuma. Postoji mogućnost poboljšanja algoritma tako da se na temelju prošlih mjerenja izračuna nova kovarijancijska matrica svakog puta kada se pojavi nova vrijednost razlike vremena. Prilagodba na šum povećala bi točnost filtra. Također prilikom mjerenja moramo uzeti u obzir okretanje motora u oba smjera. Kao što smo naveli u prošlim poglavljima, pozicije smještaja Hallovih senzora odstupaju od idealnih vrijednosti. Kalibracijom se može smanjiti pogreška u mjerenjima. Iako je srednja vrijednost vremena, potrebnog da rotor prođe od početne pozicije do svakog senzora, dovoljno dobra moguće je povećati broj uzoraka. Također moguće je izmjeriti odstupanje za svaki od polova rotora motora, a ne pretpostaviti da su svi magneti razmaknuti jednako. Uz uklanjanje pogreške za svaki od magneta, potrebno je uzeti i u obzir negativan smjer rotacije jer se odstupanja mogu razlikovati za svaki od smjera kretanja. Uklanjanje pogrešaka i dobrim definiranjem parametara filtra može se povećati točnost estimacije kuta, a i time povećati efikasnost vektorskog upravljanja motora.

LITERATURA

- [1] M. Maheswaran A. Balashanmugham. Permanent-magnet synchronous machine drives, 2019.
- [2] A. Becker. Kalman filter in one dimension, 2018. URL <https://www.kalmanfilter.net/kalman1d.html>.
- [3] D. Collins. What is space vector pulse width modulation (svpwm)?, 2019. URL <https://www.motioncontroltips.com/what-is-space-vector-pulse-width-modulation-svpwm/>.
- [4] Texas Instruments Europe. Field orientated control of 3-phase ac-motors, 1998.
- [5] G. Moselli G. D'Angelo. Improve hall sensor rotor-position measurements: Part 1, 2007. URL <https://www.eetimes.com/improve-hall-sensor-rotor-position-measurements-part-1/>.
- [6] Honeywell. Hall effect sensing and application, 2020.
- [7] Milica Pužar I. Mandić, V. Tomljenović. *Sinkroni i asinkroni električni strojevi*. Tehničko veleučilište Zagreb - Elektrotehnički odjel, 2012.
- [8] Texas Instruments. Drv5013 digital-latch hall effect sensor, 2019. Datasheet.
- [9] Real Time Engineers Ltd. The freertos reference manual, 2016. API Functions and Configuration Options.
- [10] T. Lacey. Chapter 11 tutorial: The kalman filter, 1998. URL <http://web.mit.edu/kirtley/kirtley/binlustuff/literature/control/Kalman%20filter.pdf>.
- [11] STM32F407VG Reference Manual. Stmicroelectronics, 2014.
- [12] Description of STM32F2xx Standard Peripheral Library. Stmicroelectronics, 2011.

- [13] T. Petković. Kalmanov filter, 2020. URL https://www.fer.unizg.hr/_download/repository/SPUS_2019-2020_cjelina_15_kalmanov_filter_biljeske.pdf.
- [14] D. Seršić. Vektorsko upravljanje elektromotorima ii, 2019.
- [15] D. Seršić. Vektorsko upravljanje elektromotorima i, 2019.
- [16] STM32F407VG. Stmicroelectronics, 2013. Datasheet.
- [17] STMicroelectronics. Stpower sic mosfet, 2020. URL <https://www.st.com/en/power-transistors/stpower-sic-mosfets.html#overview>.
- [18] N. Stojanović. High accuracy speed and angular position detection by multiple sensing, 2017.
- [19] Wikipedia. Direct torque control, 2020. URL https://en.wikipedia.org/wiki/Direct_torque_control.
- [20] Wikipedia. Hall effect, 2020. URL https://en.wikipedia.org/wiki/Hall_effect#Hall_effect_in_semiconductors.
- [21] Wikipedia. Pid controller, 2020. URL https://en.wikipedia.org/wiki/PID_controller#PI_controller.
- [22] Wikipedia. Kalman filter, 2020. URL https://en.wikipedia.org/wiki/Kalman_filter.
- [23] Wikipedia. Synchronous motor, 2020. URL https://en.wikipedia.org/wiki/Synchronous_motor.

Estimacija kuta rotora motora korištenjem Hallovog senzora

Sažetak

Vektorsko upravljanje elektromotorima temelji se na podešavanju vektora struja kako bi se postigao puni moment. Kako bi se podesila ispravna vrijednost vektora struje, potrebna je informacija o trenutnom kutu rotora. Pozicija rotora kod sinkronih motora usklađena je sa smjerom magnetskog polja. Magnetsko polje, odnosno pozicija rotora, mjeri se pomoću Hallovih senzora koji omogućuju detekciju pozicije u 6 dijelova punog kruga. Za optimalnu estimaciju kuta rotora koristi se Kalman filtar. Algoritam Kalman filtra proveden je kroz simulacije u MATLAB-u te realiziran na ugradbenom računalu u programskom jeziku C. Kako bi se minimizirale sistemske pogreške provodi se kalibracija sustava.

Ključne riječi: Hallov senzor, vektorsko upravljanje, sinkroni motor, rotor, kut, Kalmanov filtar, kalibracija

Rotor Angle Estimation Using Hall Sensors

Abstract

Field-oriented control of electric motors is based on adjusting the current vector to achieve full torque. In order to adjust the correct value of the current vector, information about angle of the rotor is required. The position of the rotor in synchronous motors is aligned with the direction of the magnetic field. The magnetic field (position of the rotor), is measured by using Hall sensors that allow detection of positions in 6 parts of full circle. A Kalman filter is used to optimally estimate the rotor angle. The Kalman filter algorithm is proven through simulations in MATLAB and implemented on the embedded computer system in programming language C. In order to minimize system errors, system calibration is performed.

Keywords: Hall sensor, field-orientated control, synchronous motor, rotor, angle, Kalman filter, calibration