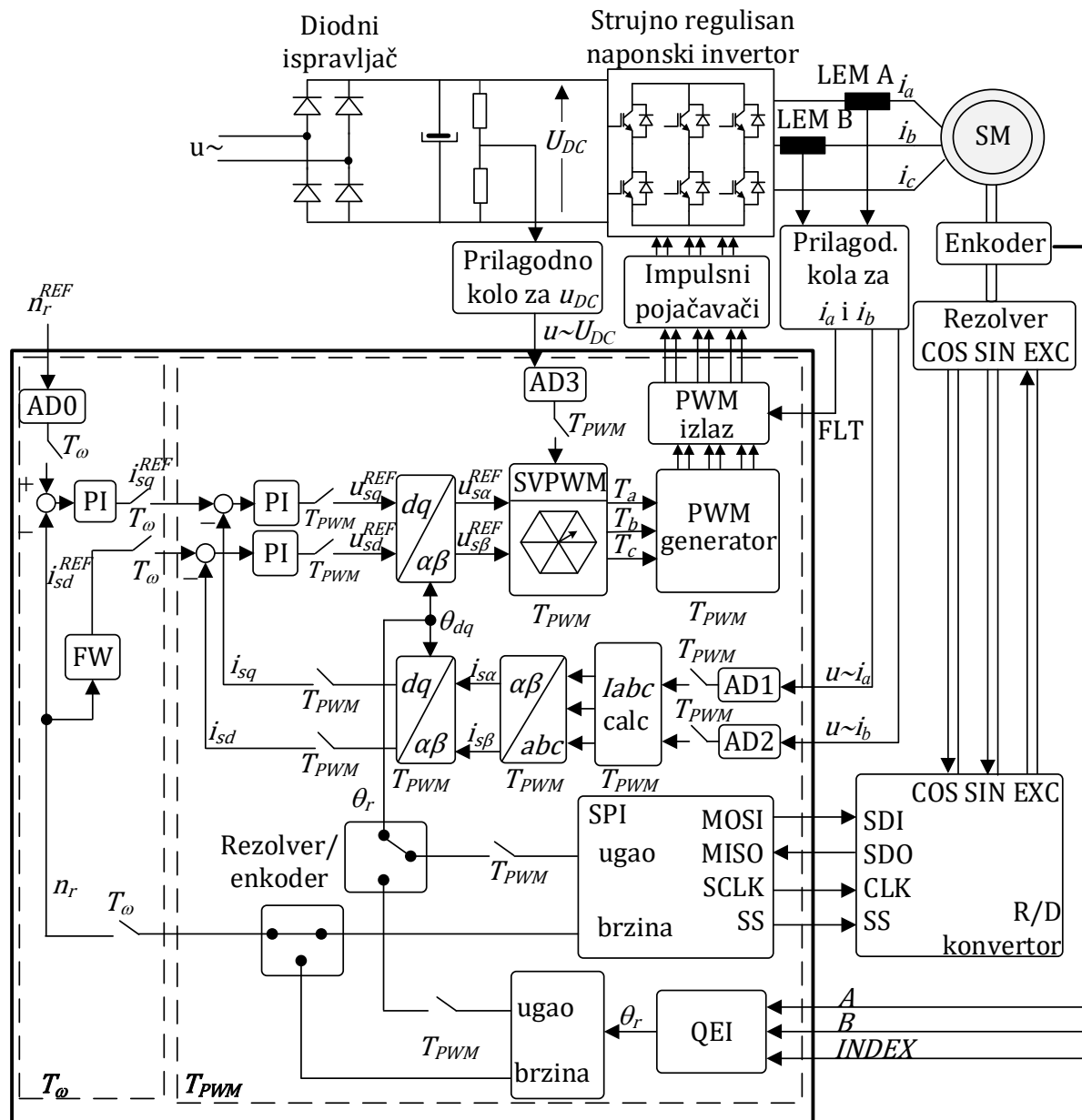


7.6. Mikroprocesorska realizacija pogona sa PMSM

U ovom poglavlju su dati detalji realizacije mikroprocesorski upravljanog pogona sa sinhronim motorom koji je vektorski upravljan i regulisan po brzini.

7.6.1. Blok dijagram pogona sa PMSM upravljanog po brzini

Na slici 7.26 je dat tipičan blok dijagram vektorski upravljanog pogona sa PMSM. Na dijagramu su ravnopravno tretirani potrebni hardverski blokovi, programski blokovi kojima se realizuju potrebne upravljačke funkcije, kao i specijalizovane periferije koje program koristi.



Slika 7.26. Blok dijagram mikroprocesorski upravljanog vektorskog pogona sa PMSM i regulacijom brzine rotora sa enkoderom i rezolverom kao dve opcije za davač položaja

Vektorski pogon sa PMSM ima slične hardverske blokove kao i pogon sa AM:

1. energetski pretvarač sa ulaznim ispravljačkim i izlaznim invertorskim stepenom,
2. impulsne pojačavače za poluprovodničke prekidače invertora,
3. prilagodna kola za obradu mernih signali linijskih struja i napona jednosmernog kola.

Razlika u odnosu na pogon sa AM su motor i davač položaja:

1. motor je PMSM, ali se povezuje na inverter na isti način kao i asinhroni motor,
2. pošto upravljanje PMSM zahteva poznavanja apsolutnog položaja rotora, najveća razlika u odnosu na pogon sa AM je u davaču na vratilu, za koji su na slici 7.26 prikazane dve opcije. Kada je davač inkrementalni enkoder, pogl. 6.7.2, neophodno je za PMSM koristiti INDEX signal, ali je i dalje pre pokretanja potrebno pronaći apsolutni položaj rotora, i presetovati sumu integrala položaja, ili dovesti rotor u poznati početni položaj, a zatim resetovati sumu integrala položaja, pogl. 7.4. Ukoliko je davač na vratilu rezolver, pogl. 7.6.2, apsolutni položaj rotora je uvek poznat i PMSM se može odmah startovati. Ali, pošto su generisanje signala potrebnih za rad rezolvera, kao i obrada njegovih izlaznih signala, veoma složeni, u pogonima sa PMSM koji poseduje rezolver se praktično uvek koristi dodatni R/D konvertor. R/D konvertor je složeno elektronsko kolo, na primer PGA411-Q1, koje generiše potrebni pobudni signal *EXC* ka pobudnom namotaju rezolvera, prima diferencijalne *COS/SIN* signale sa rezolvera, demoduliše te signale, pokreće interno PLL kolo koje se brzo zaključava u odnosu na položaj rotora i prati njegovu promenu uz minimalno kašnjenje. Rezultat PLL petlje su brzina i položaj rotora koje R/D konvertor čuva u internim registrima. Mikrokontroler uobičajeno pristupa ovim registrima digitalnim putem, čime se eliminiše mogućnost greške u prenosu informacije. Za ovu svrhu se često koristi serijski sinhroni SPI protokol (signali *MOSI*—*master out slave in*, *MISO*—*master in slave out*, *SS*—*slave select i clock*) koji omogućava brzo simultano čitanje registara za brzinu i položaj. Pored funkcije praćenja brzine i položaja, moderni R/D konvertori imaju i zaštitnu funkciju, tj. prate signale sa rezolvera i u slučaju nepravilnog rada signaliziraju grešku mikrokontroleru. U tom slučaju R/D konvertor ulazi u režim kvara (FAULT stanje), čime se obustavlja dalji rad pogona. Ovim se izbegava rad pogona sa pogrešnom informacijom o brzini, što je važan deo sigurnosnih standarda, na primer u automobilske industriji.

Što se tiče upravljačkih blokova koji su programski realizovani, oni su slični kao kod vektorskog pogona sa AM, i realizuju se u dužoj T_ω i kraćoj T_{PWM} periodi. Akcije koje kontroler izvršava u dužoj periodi odabiranja T_ω brzinske petlje su date na slici 7.26 levo:

1. čitanje referentne brzine, koja se u primeru zadaje kao napon i čita preko A/D ulaza,
2. merenje brzine, u slučaju enkodera račun inkrementa položaja pročitano iz QEI, dok u slučaju rezolvera direktno čitanje registra za brzinu unutar R/D konvertora,
3. proračun linearnog PI regulatora brzine koji na osnovu greške brzine upravlja referentnom amplitudom struje statora,
4. MTPA račun referentnih dq koordinata struje statora za datu referentnu amplitudu,
5. slabljenje polja koje po jednom rešenju tek u slučaju nedostatka napona dodatno upravlja referentnom d komponentom struje statora, i time slabi fluks statora.

Ovim postupkom vektorski kontroler jednom u periodi T_ω računa nove vrednosti i_{sd}^{REF} , i_{sq}^{REF} .

Veći deo mikroprocesorskog programa vrši vektorsko upravljanje PMSM koje se izvršava sa kraćom PWM periodom T_{PWM} , i čine ga akcije grupisane na desnoj strani slike 7.26:

1. Merenje i skaliranje veličina koje je potrebno pratiti svake PWM periode:

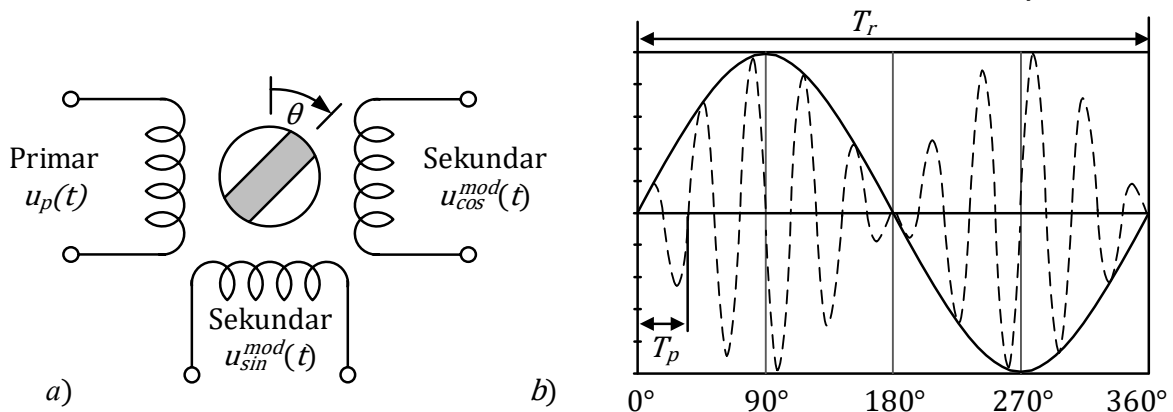
- a. merenje dve linijske struje (i_a, i_b), i račun treće fazne struje (i_c),
 - b. čitanje položaja iz R/D konvertora ili QEI periferije, rezultat ugao θ_r ,
 - c. čitanje napona jednosmernog međukola (u_{DC}).
2. Regulacija struje u dq koordinatnom sistemu:
- a. transformacija struja $abc/\alpha\beta$ i $\alpha\beta/dq$, rezultat i_{sd}, i_{sq} ,
 - b. račun dva diskretna PI regulatora struje u dq domenu, rezultat $u_{sd}^{REF}, u_{sq}^{REF}$,
 - c. transformacija referentnih napona $dq/\alpha\beta$, rezultat $u_{s\alpha}^{REF}, u_{s\beta}^{REF}$.
3. Postavljanje dq koordinatnog sistema na položaj vektora fluks magneta, tj. na najnovije pročitani apsolutni položaj rotora, $\theta_{dq} = \theta_r$.
4. SVPWM račun na osnovu $u_{s\alpha}^{REF}, u_{s\beta}^{REF}$ i upis u PDC registre PWM modula.

7.6.2. Merenje apsolutnog položaja putem rezolvera

Rezolver je senzor položaja prigraden na rotor koji meri njegov trenutni apsolutni položaj. Osnovna ideja rezolvera kao senzora je razlaganje mehaničkog ugla rotora na njegove ortogonalne komponente. U praksi, rezolver na svome izlazu daje dva signala, jedan proporcionalan sinusnom uglu, a drugi proporcionalan kosinusnom uglu rotora. Pomoću ova dva signala je potpuno definisan apsolutni položaj rotora, što je jedna od najvažnijih prednosti rezolvera u odnosu na inkrementalne enkodere čiji signal je proporcionalan promeni položaja.

Tradicionalni rezolver je rotacioni transformator sa magnetnom spregom između primarnog namotaja na rotoru, i dva sekundarna namotaja na statoru. Magnetna sprega varira sa položajem rotora, po sinusnoj funkciji ugla rotora za prvi, i kosinusnoj funkciji ugla rotora za drugi namotaj. Tradicionalni rezolver ima primarni namotaj na rotoru kome treba dovesti pobudni signal preko četkica i prstenova, ili preko rotacionog transformatora. Sve ovo komplikuje izradu i uvećava dimenzije rezolvera. Zato se danas češće koriste rezolveri kod kojih je i primarni pobudni namotaj na statoru. Bez obzira na mesto postavljanja, primarni namotaj zahteva naizmenični pobudni (eksitacioni) signal, koji će biti modulišući signal sin/cos signalima na sekundarnim namotajima.

Šematski prikaz rezolvera dat je na slici 7.27 (a), a na slici 7.27 (b) je kao primer dat signal koji se dobija na sinusnom namotaju kada se rotor okreće ugaonom brzinom $\omega_r = \frac{2\pi}{T_r}$, i kada se pobudni namotaj napaja modulišućim signalom ugaone učestanosti $\omega_p = \frac{2\pi}{T_p}$.



Slika 7.27. Šema rezolvera (a) i primer naponskog signala (b) na sinusnom sekundarnom namotaju za ugaonu brzinu rotora ω_r i modulišući signal ω [4_4]

Na pobudni namotaj se dovodi modulišući prostoperiodični signal,

$$u_p(t) = U_p \sin(\omega_p t), \quad (7.81)$$

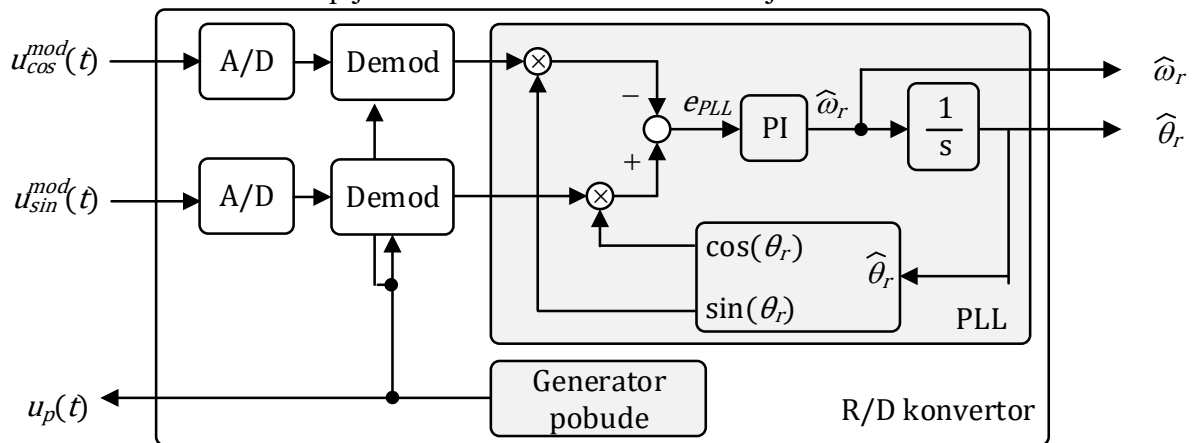
a na sekundarnim namotajima se dobijaju signali koji su proporcionalni sin/cos mehaničkog ugla rotora i koji su modulirani pobudnim signalom,

$$u_{sin}^{mod}(t) = U_p \cdot TR \cdot \sin(\omega_p t) \sin(\theta_r), \quad (7.82)$$

$$u_{cos}^{mod}(t) = U_p \cdot TR \cdot \sin(\omega_p t) \cos(\theta_r), \quad (7.83)$$

gde je TR odnos transformacije rezolvera, koji definiše tačku maksimalne sprege između primarnog i sekundarnih namotaja.

Signali sa rezolvera se obavezno dodatno obrađuju u specijalnim integrisanim R/D konvertorima, npr. kolo PGA411-Q1. Zadatak ovih kola je višestruk, jer kolo treba da: 1. generiše modulišući signal i njime pobudi rezolver, 2. prihvati kvadraturne sin/cos signale sa rezolvera, 3. demoduliše ih tako da iz njih eliminiše modulišući nosioc, 4. putem fazno zaključane petlje da odredi ugao i brzinu rotora. R/D konvertor zatim analognim putem ili serijskom vezom tipa SPI protokola šalje u realnom vremenu podatke o položaju i brzinu rotora ka mikrokontroleru. Principijalna šema R/D konvertora data je na slici 7.28.



Slika 7.28. Principijalna šema R/D konvertora [4_4]

Kvadrturni signali sa enkodera se preko A/D konvertora uvode u R/D konvertor. Tu se prvo demodulišu, a zatim se generiše signal greške,

$$e_{PLL} = u_{sin}(t) \cdot \cos(\hat{\theta}_{dq}) - u_{cos}(t) \cdot \sin(\hat{\theta}_{dq}). \quad (7.84)$$

Ako je stvarni ugao rotora θ_r , a procenjen ugao u konvertoru $\hat{\theta}_r$, za signal greške se dobija,

$$e_{PLL} = U_p \cdot TR \cdot \sin(\theta_{dq}) \cdot \cos(\hat{\theta}_{dq}) - U_p \cdot TR \cdot \cos(\theta_{dq}) \cdot \sin(\hat{\theta}_{dq}). \quad (7.85)$$

Za male greške ugla praktično se dobija linearna zavisnost greške PLL i greške ugla,

$$e_{PLL} = U_p \cdot TR \cdot \sin(\theta_{dq} - \hat{\theta}_{dq}) \approx U_p \cdot TR \cdot (\theta_{dq} - \hat{\theta}_{dq}). \quad (7.86)$$

Ova greška se dovodi na ulaz PI regulatora za procenu ugaone učestanosti rotora. Ako PLL ugao kasni dobijena greška ugla je pozitivna i PI regulator uvećava procenjenju ugaonu brzinu $\hat{\omega}_r$, čime i PLL integrator ugla ubrzava i dostiže ugao ulaznog signala. Za negativnu grešku važi obrnuto. U stacionarnom stanju PLL se zaključa, tj. integralno dejstvo PI regulatora “pamti” brzinu rotora, i ugao određen PLL-om bez greške dalje prati promenu ugla rotora.

7.6.3. Blok dijagram algoritma programa za vektorsko upravljanje PMSM sa rezolverom kao davačem apsolutnog položaja

Blok dijagram algoritma mikroprocesorskog programa za digitalno vektorsko upravljanje i regulaciju brzine PMSM je dat na slikama 7.29 – 7.33. Program podrazumeva da je davač na vratilu tipa rezolvera, sa fabrički podešenim ofsetom, tj. da je rezolverom izmereni položaj jednak apsolutnom uglu između magneta na rotoru i namotaja a faze statota.

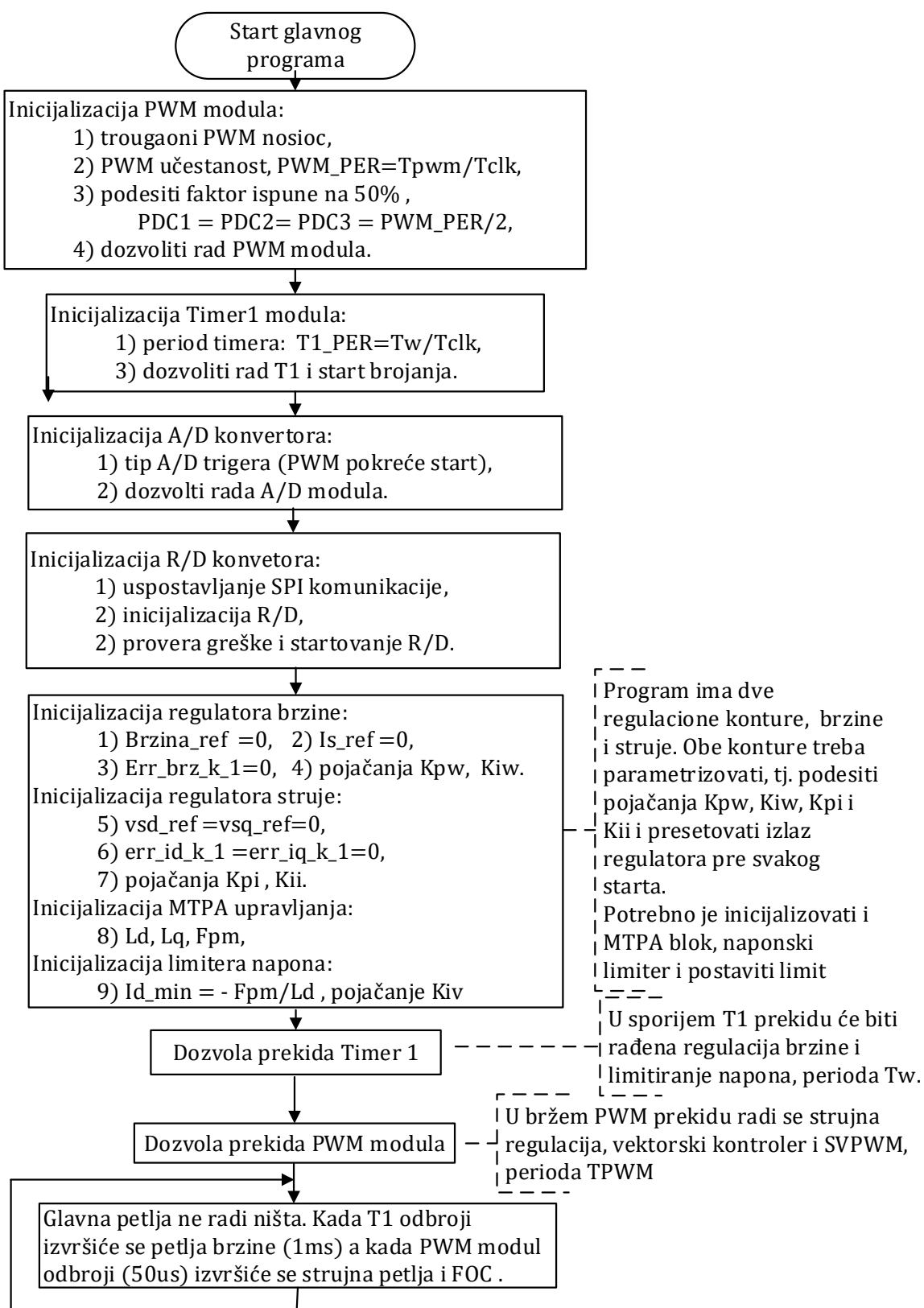
Na slici 7.29 je dat algoritam glavnog programa koji počinje da se izvršava nakon reseta i izvršava sledeće radnje: 1. inicijalizaciju potrebnih perifera (PWM modula, A/D konvertora i Timer modula), 2. inicijalizaciju spoljnog R/D konvertora putem SPI protokola, 3. inicijalizaciju programskih konstanti (regulatora struje i brzine, MTPA bloka i limitera napona), i 3. inicijalizaciju programskih promenljivih (akumulisana vrednost referentne struje statora na izlazu regulatora brzine se postavlja na nulu, itd). Glavni program zatim ulazi u glavnu petlju u kojoj se vrti u krug i čeka prekid brojačkog (Timer) modula (signal da je prošla perioda T_ω) ili PWM modula (signal da je prošla perioda T_{PWM}).

Na slici 7.30 je dat algoritam prekidne rutine Timer modula koji se izvršava svake periode T_ω (na primer 1ms) i u kome se poziva regulator brzine i po potrebi limitira napon statora PMSM. Regulator brzine na osnovu razlike između referentne i izmerene brzine upravlja referentnom vrednošću amplitude struje statora koju MTPA blok deli na optimalne vrednosti dq komponenti struje statora (u slučaju SPMSM d komponenta je nula). Zatim, blok za limitiranje napona proverava da li se amplituda napona na izlazu strujnih regulatora približila maksimalnom naponu koji inverter može da ostvari (za linearni režim $u_{DC}/\sqrt{3}$, za nelinearni $2u_{DC}/3$). Ukoliko je napon ispod limita, d komponenta struje ostaje nepromenjena, a ukoliko je napon blizu limita (npr. veći od 95% limita) limiter napona integralnim dejstvom dodatno uvećava negativnu d komponentu referentne struje i time smanjuje fluks statora, i samim tim vraća napon statora ispod limita. Ovim je obezbeđena naponska margina potrebna za rad strujnih regulatora. Nakon završetka ove rutine izračunate su nove referentne vrednosti dq struja (I_{sd_ref} , I_{sq_ref}), i one neće biti promenjene do sledećeg Timer prekida.

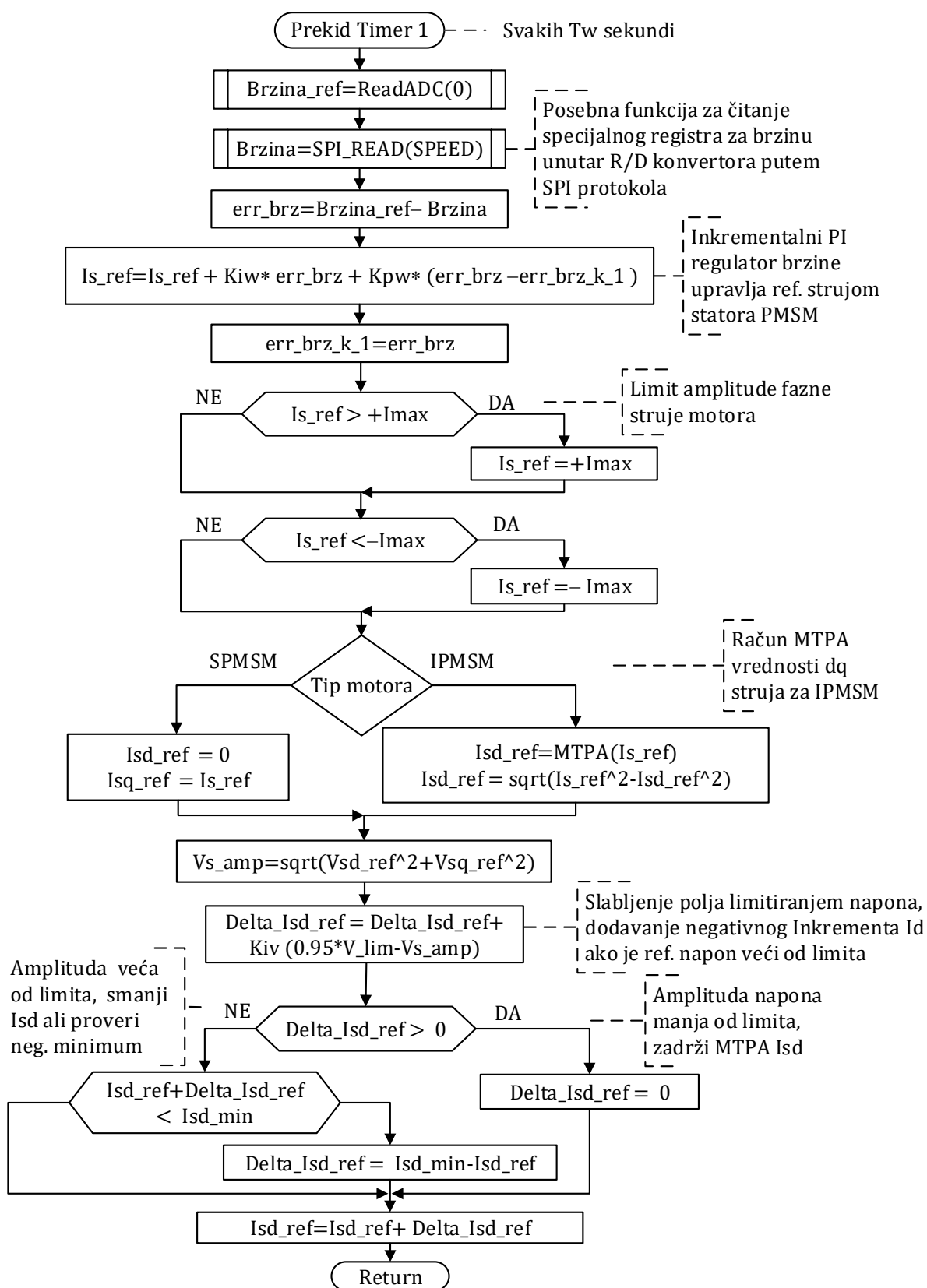
Na slikama 7.31 – 7.33 su dati delovi blok dijagrama algoritma PWM prekidne rutine. Prekidna rutina se izvršava na početku svake PWM periode i u njoj je realizovana strujna kontrola u dq domenu, račun vektorskog upravljanja i SVPWM račun.

Na slici 7.31 je dat početni deo PWM prekidne rutine (npr. 50 μs) u kojem program prvo prikuplja sva potrebna merenja potrebna za vektorsko upravljanje PMSM. Program preuzima vrednosti dve linijske struje (one su već automatski odabrane na sredini prethodne PWM periode i spremne za čitanje) i računa treću struju. U nastavku programa se $abc/\alpha\beta$ i $\alpha\beta/dq$ transformacija dobijaju merene vrednosti struja u dq koordinatnom sistemu (I_{sd} , I_{sq}). Program takođe čita izmereni ulazni napon (V_{dc}), i na osnovu njegove vrednosti računa i limiti napona (V_{sd_limit} , V_{sq_limit}), koji su time zajedno sa izmerenim strujama spremni za strujne regulatore. Konačno, program iz R/D konvertora čita i promenljivu (Θ_{rotor}), koja treba da nosi informaciju o apsolutnom položaju rotora u odnosu na fazu a statora.

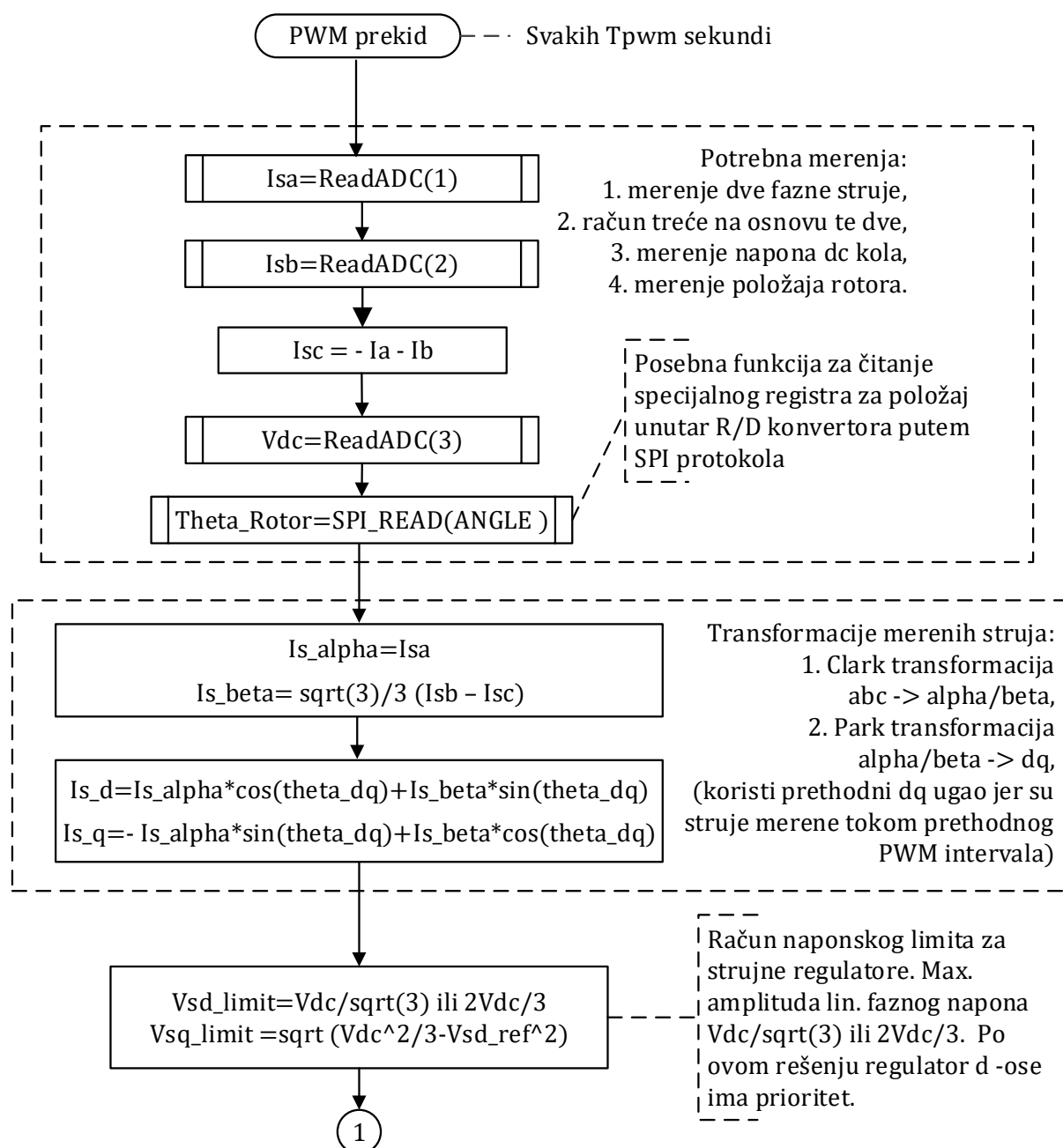
U delu algoritma na slici 7.32 su dalje računaju dva diskretna PI strujna regulatora u inkrementalnoj formi sa limitima na čijim izlazima se dobijaju referentni naponi (V_{sd_ref} , V_{sq_ref}). Zatim, u ovom delu programa se i dq koordinatni sistem (Θ_{dq}) postavlja na najnovije pročitano ugao magneta, tj. rotora (Θ_{rotor}) koji je na početku PWM periode već pročitano preko serijske SPI linije iz R/D konvertora.



Slika 7.29. Blok dijagram algoritma PMSM FOC program: glavni program

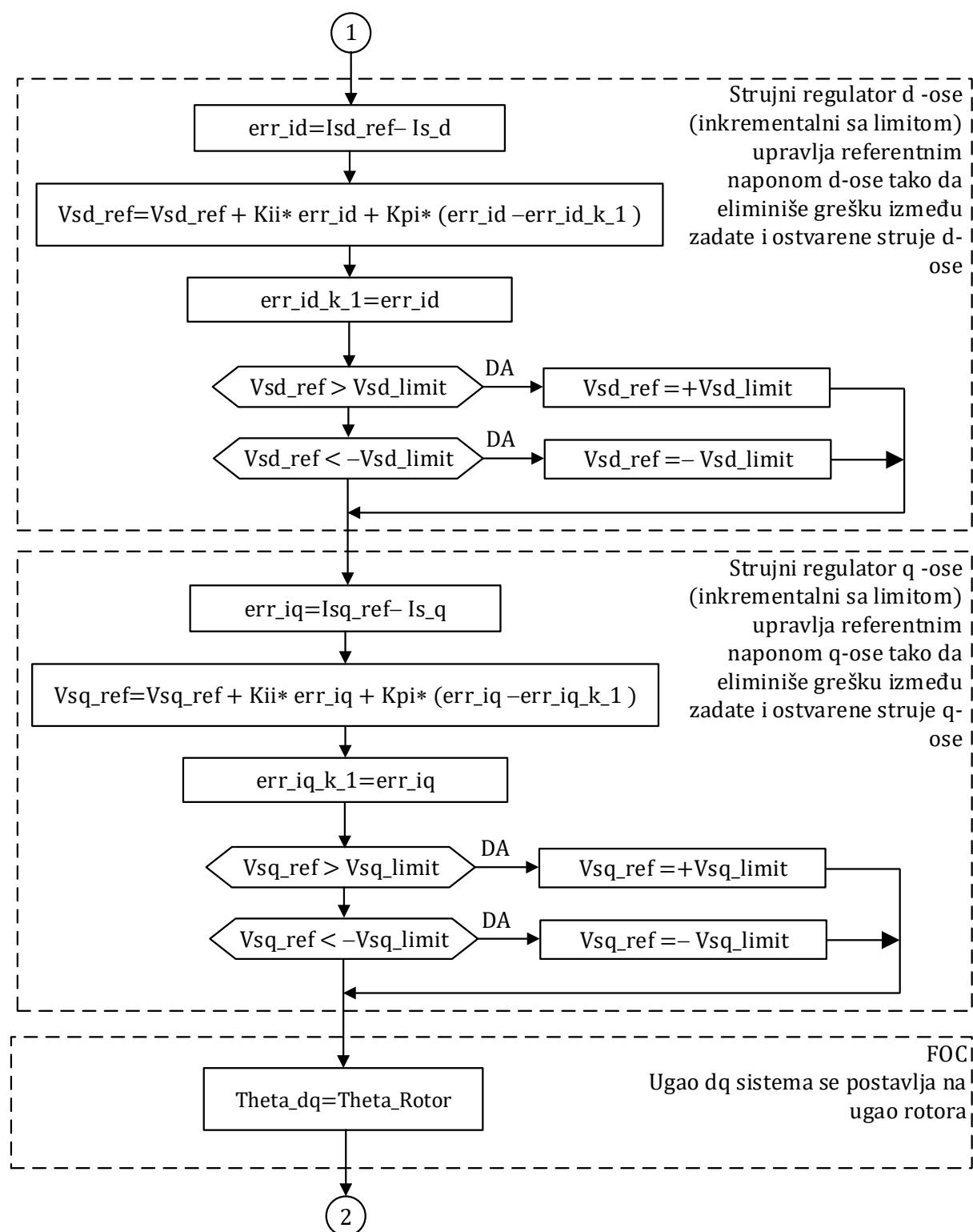


Slika 7.30. Blok dijagram algoritma PMSM FOC programa: brzinska petlja i blok za limitovanje napona (slabljenje polja)

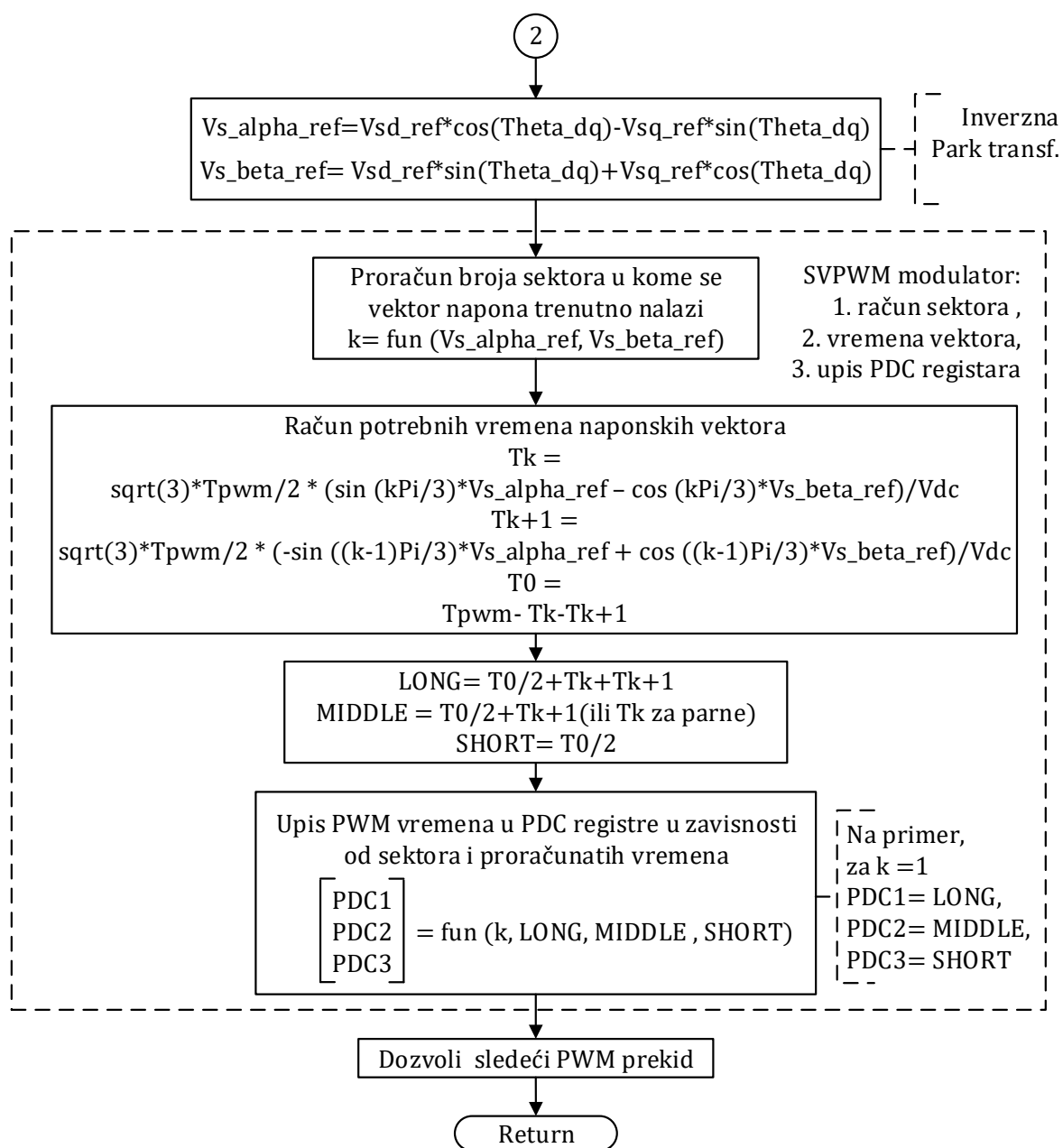


Slika 7.31. Blok dijagram algoritma PMSM FOC programa: PWM prekidna rutina– I deo

U delu na slici 7.33 algoritam dalje nastavlja sa $dq/\alpha\beta$ transformacijom referentnih napona (V_{sd_ref} i V_{sq_ref}) koji su upravo izračunati kao izlazi PI regulatora struja. U ovom računu se već koristi nova vrednost ugla dq sistema (Θ_{dq} kojoj je dodeljena najnoviji ugao rotora, izmeren u tekućoj PWM periodi). Ovo izmena ugla u odnosu na $\alpha\beta/dq$ transformaciju struja je i korektna, jer struje su izmerene u prethodnoj periodi, a ovaj napon će usled kašnjenja PWM modula biti aktivan tek u sledećoj PWM periodi. Nakon transformacije izračunati referentni naponi ($V_{s_alpha_ref}$ i $V_{s_beta_ref}$) se koriste kao ulazi SVPWM modula koji vrši proračun aktivnih vektora i upisuje potrebna vremena uključanja u PDC registre PWM modula za faktore ispunje.



Slika 7.32. Blok dijagram algoritma PMSM FOC programa: PWM prekidna rutina– II deo



Slika 7.33. Blok dijagram algoritma PMSM FOC programa: PWM prekidna rutina– III deo

Može se primetiti da je algoritam za vektorsko upravljanje PMSM praktično redukovana verzija vektorskog upravljanja AM, jer kod PMSM nema potrebe za računom klizanja. Ono što je složenije kod PMSM je potreba za poznavanjem apsolutnog položaja rotora, što se najlakše rešava primenom apsolutnog enkodera ili rezolvera, kao u ovom primeru. Ako PMSM poseduje samo inkrementalni enkoder, ili uopšte ne poseduje davač na vratilu, pre svakog polaska neophodno je izvršiti poravnanje rotora sa a osom statora, što se postiže utiskivanjem d komponente struje statora koja postavlja rotor u poznati početni položaj i pogon je tada spreman za rad.

7.6.4. Blok dijagram algoritma programa za vektorsko upravljanje PMSM sa enkoderom kao inkrementalnim davačem položaja

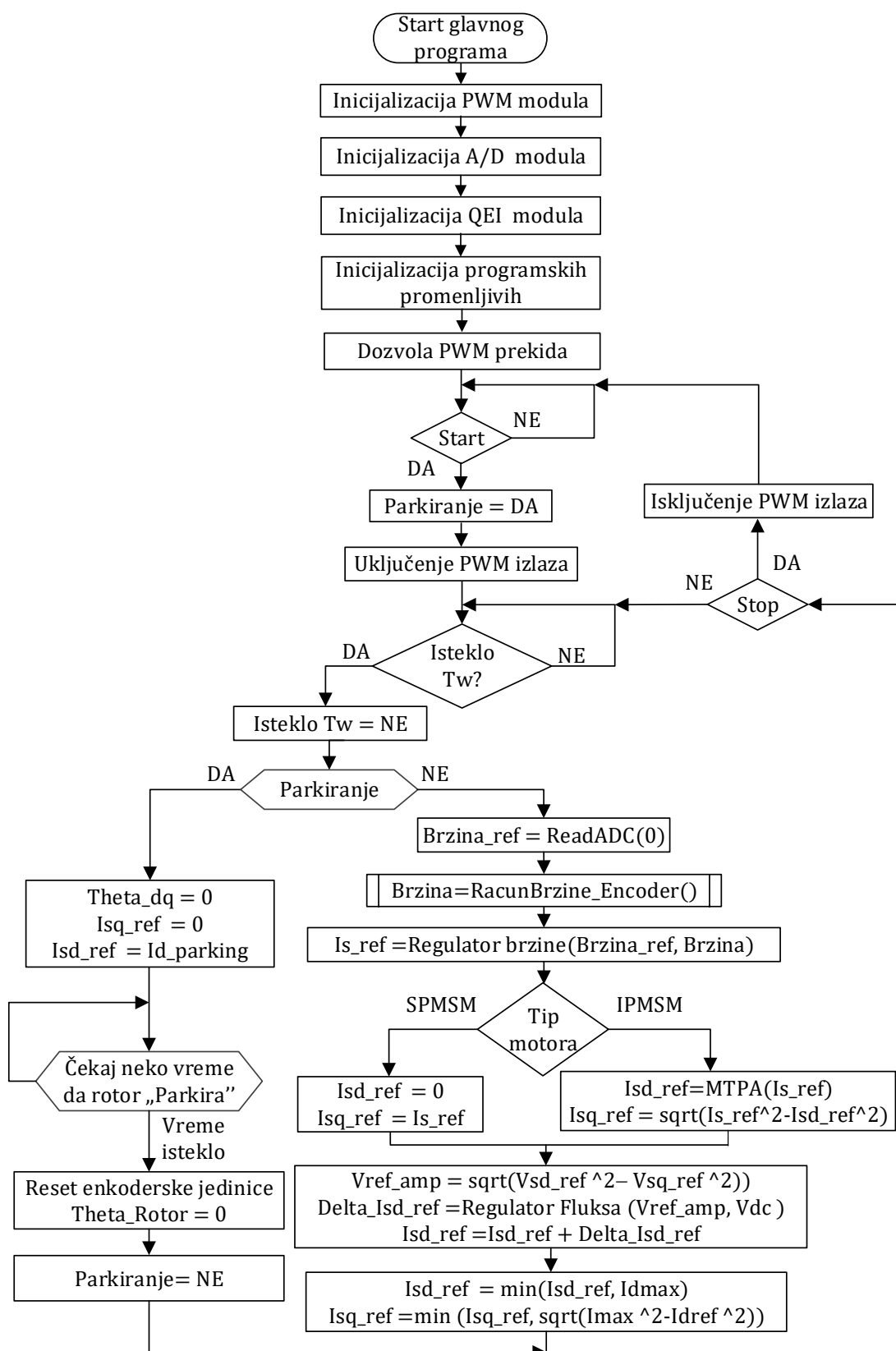
U ovom poglavlju je dat primer blok dijagrama algoritma mikroprocesorskog programa za vektorsko upravljanje PMSM ali u ovom slučaju se podrazumeva da je davač na vratilu tipa inkrementalnog enkodera. Sada pre pokretanja pogona nije moguće pretpostaviti da je apsolutni položaj rotora poznat, nego je neophodno taj početni položaj ili izmeriti, ili rotor postaviti u poznat početni položaj, pogl. 7.4. Naravno, nakon pokretanja motora i dolaska prvog INDEX signala sa enkodera, ako je fabrički podešen njegov ofset, može se u daljem radu smatrati da je položaj dobijen kao suma inkremenata ujedno jednak apsolutnom uglu između magneta na rotoru i namotaja a faze statota. U slučaju bilo kog odstupanja akumulisanog položaja u toku jednog obrtaja, npr. izgubljen impuls, novi dolazak INDEX signala resetuju sumu položaja na unapred poznati položaj, čime je izbegnuta akumulacija greške usled otvorene integracije, i pogon sa PMSM i inkrementalnim enkoderom kao davačem na vratilu može nesmetano da radi.

Blok dijagram algoritma programa prikazan je na slikama 7.34 – 7.35. Program važi za oba tipa PMSM i realizovan je tako da pre starta uvek vrši parkiranje rotora, pogl. 7.4, koje se realizuje pre svakog pokretanja motora.

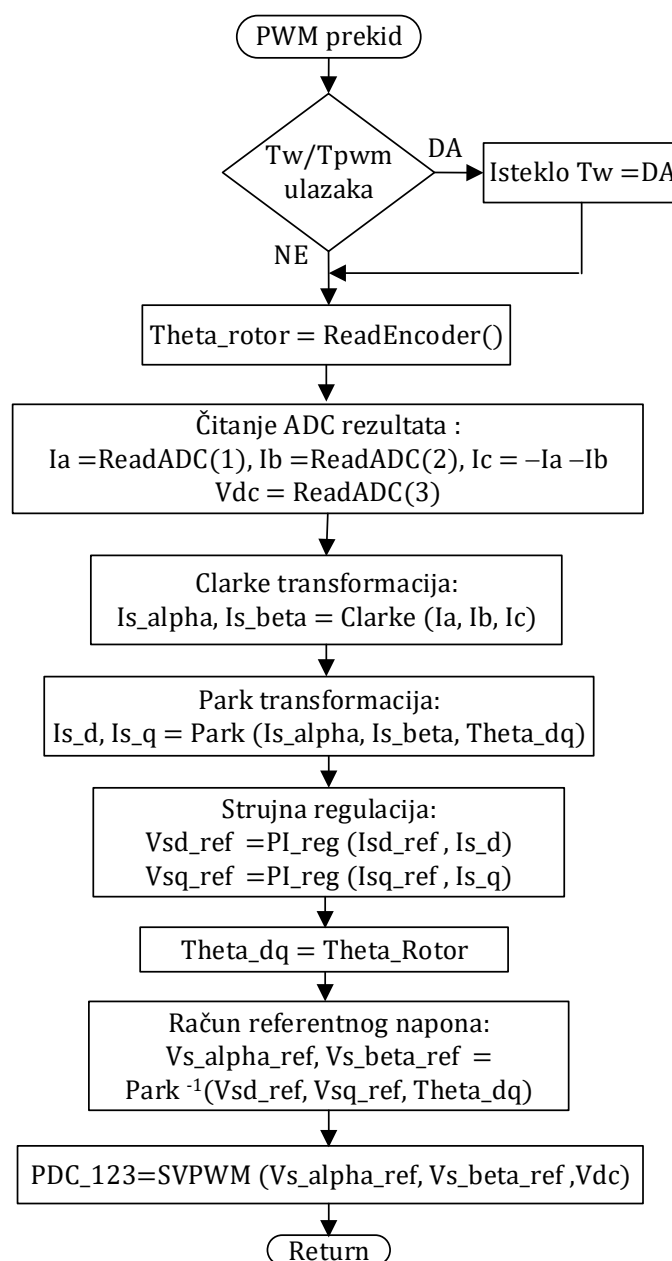
Na slici 7.34 je dat blok dijagram algoritma glavnog programa. Glavni program po resetu vrši inicijalizaciju potrebnih perifernih modula i programskih promenljivih. Zatim ulazi u glavnu petlju koja radi svake T_ω (koju signalizira PWM prekid nakon N ulazaka) i koja ima dva moda rada: „parkiranje”, i „režim regulacije brzine”. Program prvo čeka „Start” komandu, po čijem dolasku aktivira PWM izlaze i ulazi u glavnu petlju u „Parking” režimu. U „Parking” režimu neko vreme se pušta jednosmerna struja kroz namotaje statora i čeka da se rotor poravna sa a fazom, tj. da zauzme nulti apsolutni položaj. Nepomični strujni vektor u fazi a se tokom parkiranja dobija sa: $i_{sd}^{REF} = I_{sd}^{PARKING}$, $i_{sq}^{REF} = 0$ i $\theta_{dq} = 0$, gde je $I_{sd}^{PARKING}$ vrednost struje dovoljna da pokrene rotor i izvrši njegovo parkiranje pri datom opterećenju. Kada vreme za parkiranje istekne ukida se parking struja, ugao inkrementalnog enkodera se resetuje na nulti položaj, i program izlazi iz parking režima i prelazi u režim regulacije brzine i motor startuje.

U režimu regulacije brzine rotora („parkiranje”=NE) glavni program reguliše brzinu rotora PMSM sa periodom odabiranja od T_ω . Po isteku perioda T_ω se u glavnom delu programa čita zadata brzina, proračunava trenutna brzina i poziva regulator brzine. Regulator brzine upravlja amplitudom struje statora (I_{s_ref}) koja se prenosi na q struju (SPMSM) ili optimalno raspoređuje na d i q struju statora putem MTPA računa. U oba moda rada zadata struja d ose se po potrebi dodatno koriguje ako je amplituda zadatog napona veća od maksimalno moguće. Nakon provere limita ovaj deo programa završava i zadate I_{sd_ref} i I_{sq_ref} ostaju nepromenjene tokom T_ω periode. Po prijemu komande „Stop” glavni program isključuje PWM izlaze, motor se zaustavlja i program čeka novu „Start” komandu.

Na slici 7.35 je dat blok dijagram programa PWM prekidne rutine bez previše detalja jer su račun strujnih regulatora i orijentacija polja isti bez obzira na tip senzora. U prekidnoj rutini se čitaju ugao rotora iz QEI modula, funkcija `Read_Encoder()`, struje statora i napon jednosmernog međukola. Zatim se Park transformacijom dobijaju najnovije vrednosti dq struja koje se uvode u strujne regulatore. Nakon poziva strujnih regulatora se ugao dq sistema podešava na novi ugao rotora, $\theta_{dq} = \theta_{rotor}$. Referentne koordinate dq napona se inverznom Park transformacijom prenose nazad u stacionarni $\alpha\beta$ domen i postavljaju kao ulazi SVPWM modula koji računa potrebna vremena uključenja prekidača tri grane invertora i upisuje nove vrednosti u registre PWM modula.



Slika 7.34. Blok dijagram algoritma glavnog programa za vektorsko upravljanje PMSM



Slika 7.35. Blok dijagram algoritma PWM prekidne rutine za vektorsko upravljanje PMSM

Prikazani blok dijagram algoritma pokazuje da je uz dodatnu nadogradnju algoritma, posebno pri startu pogona, moguće realizovati mikroprocesorski upravljani pogon sa PMSM koji poseduje inkrementalni enkoder kao davač na vratilu. Ipak, uobičajeno je da se na PMSM motore postavljaju davači apsolutnog položaja tipa rezolvera, čime se izbegava kašnjenje pre starta pogona i u svakom trenutku rada pogona dobija pouzdanija informacija o trenutnom apsolutnom položaju magneta. Pošto rezolveri postaju sve popularniji, tehnologija njihove izrade napreduje, a posebno su interesantni reluktantni rezolveri sa sva tri namotaja (eksitacija i sin/cos) na statoru i time bez rotacionog transformatora, sa smanjenom cenom i gabaritom, a uvećanom pouzdanošću.