

SINHRONIZACIJA MIKROPROCESORSKI UPRAVLJANIH ISPRAVLJAČA I INVERTORA SA MREŽOM / SYNCHRONIZATION OF MICROPROCESSOR CONTROLLED RECTIFIERS AND INVERTERS WITH THE GRID

Bekir Fulan, BA, email: bekir.fulan@gmail.com

Internacionalni univerzitet Travnik u Travniku

Amira Trako, BA, email: amira.grabus1@hotmail.com

Internacionalni univerzitet Travnik u Travniku

Stručni članak

Sažetak: U radu se analizira jedna konkretna realizacija fazno spregnute petlje koja je u potpunosti realizovana digitalno. Upravljačke elektronike ispravljača i invertora zasnovane su na mikrokontrolerima. Cilj konstrukcije upravljačke elektronike je u oba slučaja bio, da se, pošto se već koriste mikrokontroleri, minimizira dodatna elektronika. Kod ispravljača, tiristorima u mostu upravlja se direktno sa porta mikrokontrolera, dok se kod invertora izlazni signal brojača neposredno koristi za upravljanje tranzistorima mosta. Za oba uređaja potrebno je na izvjestan način obezbjediti sinhroni rad sa mrežom, jer je u ovu svrhu rezervisan jedan broj u memoriji mikrokontrolera čija se vrijednost mijenja sinhrono sa fazom mrežnog napona. U slučaju upravljanja ispravljačem, koristi se vrijednost tabele "mustri" za paljenje. Slično je i kod invertora, kod koga se na osnovu vrijednosti u memoriji mikrokontrolera očitava u tablici sinusne funkcije. Ukoliko se u program ugradi mogućnost da se utiče na vrijednost broja koji predstavlja frekvenciju (inkrement faze), ali tako, da se kao konačni rezultat postigne, da su faze referentnog i izlaznog talasa identične, može se smatrati da će i ispravljač i inverter raditi sinhronizovano sa mrežom.

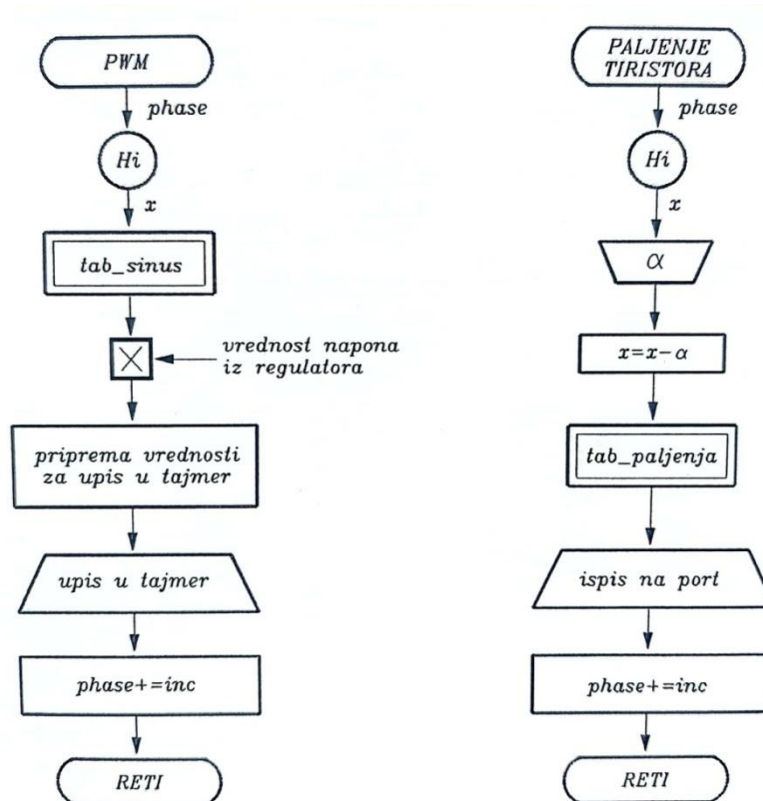
Ključne riječi: mikrokontroler, inverter, ispravljač

Abstract: The paper analyzes one concrete implementation of a phase-coupled loop, which was fully implemented digitally. Control electronics of rectifiers and inverters are based on microcontrollers. The goal of the design of control electronics in both cases was to minimize additional electronics, since microcontrollers are already used. In the rectifier, the thyristors in the bridge are controlled practically directly from the microcontroller port, while in the inverter, the output signal of the counter is directly used to control the bridge transistors. For both devices, it is necessary to provide synchronous operation with the network in a certain way, because for this purpose a number is reserved in the memory of the microcontroller whose value changes synchronously with the phase of the network voltage. In the case of rectifier control, the value of the "sample" table is used for ignition. It is similar with the inverter, where based on the value in the memory of the microcontroller, the sine function is read in the table. If the program incorporates the possibility to influence the value of the number representing the frequency (phase increment), but in such a way that the final result is that the phases of the reference and output waves are identical, it can be considered that both the rectifier and the inverter will work in sync with network.

Keywords: microcontroller, inverter, rectifier

1. Uvod

Kada govorimo o upravljačkoj elektronici ispravljača i invertora, mikrokontroler je baza. Cilj konstrukcije upravljačke elektronike je, u oba slučaja bio, da se, pošto se već koriste mikrokontroleri, minimizira dodatna elektronika. Ovo je u dobroj mjeri i ostvareno, odnosno, osim činjenice da su regulatori izlaznih veličina riješeni digitalno, izvršena su i prilagođenja izvršnih organa tako da su im ulazi baš digitalni signali (nivo 0/1). Kod ispravljača, tiristorima u mostu upravlja se praktično direktno sa porta mikrokontrolera, dok se kod invertora izlazni signal brojača B1 neposredno koristi za upravljanje tranzistorima mosta. Za oba uređaja potrebno je na izvjestan način obezbjediti sinhroni rad sa mrežom. Već je opisano da je u ovu svrhu rezervisan jedan broj u memoriji mikrokontrolera (u daljem tekstu phase) čija se vrijednost mjenja sinhrono sa fazom mrežnog napona. U slučaju upravljanja ispravljačem, ovaj broj se koristi za iščitavanje vrijednosti tabele "mustri" za paljenje. Slično je i kod invertora, kod koga se na osnovu vrijednosti phase očitava tablica sinusne funkcije, te se tako dobijena vrijednost množi sa brojem koji predstavlja željenu vrijednost napona. Ovako dobijeni rezultat (postupak izračunavanja PWM-a detaljno je opisan u tabeli) koristi se za programiranje tajmera, čiji izlaz određuje izlazni napon na mostu. Zbog pravilnog rada statičkog prekidača, neophodno je da invertorski napon bude sinhron sa mrežnim.



Slika 1. Dijagram toka prekidnih rutina ispravljača i invertora

2. Postavka problema

U konkretnim realizacijama ispravljača i invertora, prekidne rutine u kojima se vrši inkrementiranje vrijednosti phase izvršavaju se svakih $T=400\mu s$. Pošto je pri tom još usvojeno i da se vrijednost phase čuva u dva bajta (vrijednost 65536 odgovara fazi 27), vrijednost inkrementa (inc) u slučaju ulazne sinusoide periode T_f , jednoznačno je određena sa:

$$inc = \frac{65536}{\frac{T_f}{T}}$$

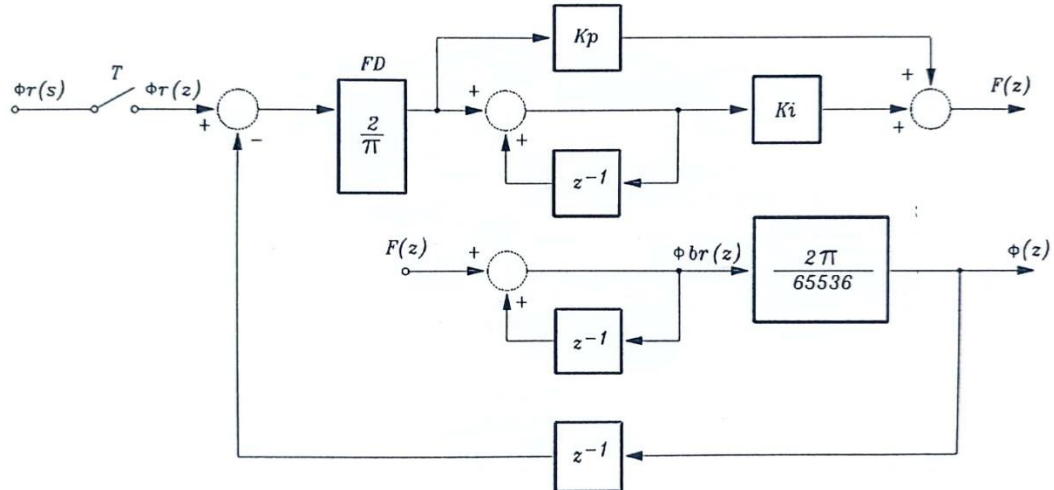
Inkrement za slučaj 50Hz-nog ulaznog talasa iznosi:

$$inc = \frac{65536}{\frac{20ms}{400\mu s}} = 1310.7$$

Potrebno je dakle u $400\mu s$ -oj prekidnoj rutini realizovati digitalnu fazno spregnutu petlju, uz minimalnu dodatnu elektroniku. Pokazaće se da je za realizaciju ovog zadatka dovoljno, na ulazni pin mikrokontrolera dovesti samo signal znaka referentne sinusoide. Softverski su realizovani: fazni diskriminator, koji na osnovu vrijednosti promjenljive phase i znaka talasa referentnog napona izračunava grešku, kao i regulator, koji na osnovu izlaza iz faznog diskriminatora generiše vrijednost frekvencije. U slučaju invertora se ovakvom realizacijom postiže, da je napon invertora (prije izlaznog filtra) u fazi sa mrežnim naponom. Izlazni napon, zbog postojanja filtra, malo kasni za mrežnim naponom. U konkretnoj realizaciji ovo kašnjenje može se zanemariti pošto je prijesčna učestanost filtra postavljena na 1,5kHz, pa se može smatrati da je fazno kašnjenje, koje unosi filter, praktično zanemarljivo za pun opseg promjene opterećenja. U slučajevima gdje kašnjenje na filteru nije zanemarljivo, na fazni diskriminator može se dovesti i informacija o znaku izlaznog napona.

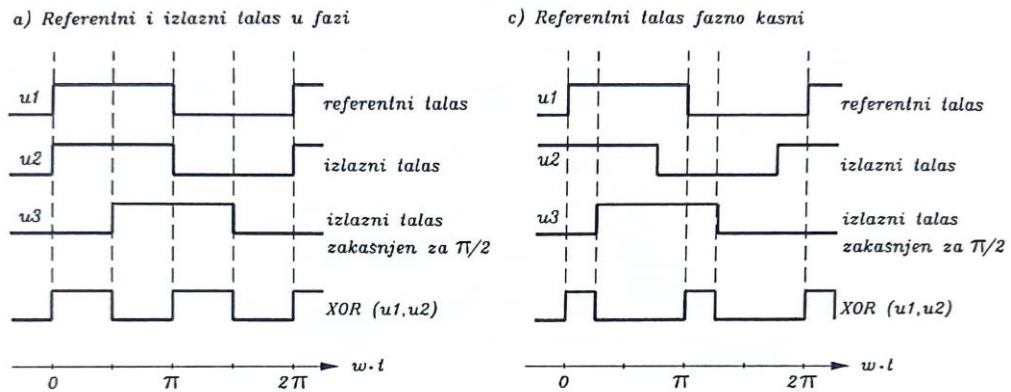
2.1 Sinteza regulatora

Strukturna šema fazno spregnute petlje prikazana je na slici 2. Sa slike se vidi da je usvojen PI regulator. Ukupan sistem, uz prisustvo ovakvog kompenzatora, u otvorenoj povratnoj sprezi posjeduje astatizam drugog reda, što obezbjeđuje nultu faznu grešku. Potrebno je dakle, za usvojenu strukturu regulatora i s obzirom na željenu dinamiku, odrediti konstante integralnog i proporcionalnog dejstva.



Slika 2. Struktura šema fazno spregnute petlje

Član pojačanja $2\pi/65536$ prisutan je usljed izabranog opsega brojeva za vrijednost faze i frekvencije kako je opisano u odeljku 4.2. Član $2/7$ je praktično karakteristika faznog diskriminatora. Kao fazni diskriminator koristi se XOR logička funkcija, na čije ulaze se dovode signal referentne faze i signal izlazne faze pomjeren za četvrtinu periode. Logička vrijednost 0 sa izlaza XOR funkcije u daljem računu tretira se kao vrijednost -1. Na slici 3. prikazani su ulazi u fazni diskriminator i odgovarajući izlaz.



Slika 3. Izlaz iz faznog diskriminatora za razne fazne stavove referentnog i izlaznog talas

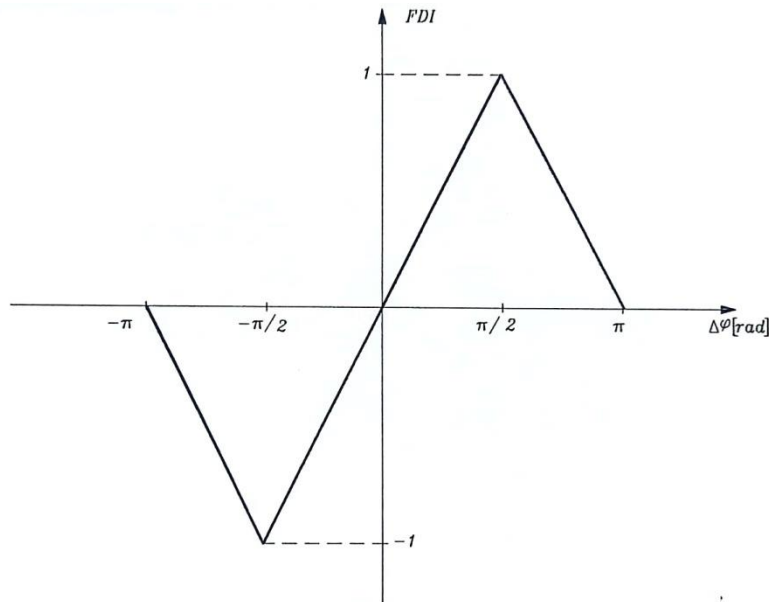
Srednja vrijednost izlaza iz opisanog faznog diskriminatora (FDI) može se predstaviti izrazom:

$$FDI = \begin{cases} \frac{\left(\frac{\pi}{2} + \Delta\alpha\right) * 1 + \left(\pi - \left(\frac{\pi}{2} + \Delta\alpha\right)\right) * (-1)}{\pi} ; \Delta\alpha \in \left(0, \frac{\pi}{2}\right) \\ \frac{\left(\Delta\alpha - \frac{\pi}{2}\right) * (-1) + \left(\pi - \left(\Delta\alpha - \frac{\pi}{2}\right)\right) * 1}{\pi} ; \Delta\alpha \in \left(\frac{\pi}{2}, 0\right) \end{cases}$$

Kada se ovaj izraz uprosti, i znajući da je prenosna funkcija pretpostavljenog faznog diskriminatora neparna funkcija, može se napisati:

$$FDI = \begin{cases} \frac{\left(\frac{\pi}{2} + \Delta\alpha\right) * 1 + \left(\pi - \left(\frac{\pi}{2} + \Delta\alpha\right)\right) * (-1)}{\pi} ; \Delta\alpha \in \left(0, \frac{\pi}{2}\right) \\ \frac{\left(\Delta\alpha - \frac{\pi}{2}\right) * (-1) + \left(\pi - \left(\Delta\alpha - \frac{\pi}{2}\right)\right) * 1}{\pi} ; \Delta\alpha \in \left(\frac{\pi}{2}, \pi\right) \end{cases}$$

Prenosna funkcija faznog diskriminatora je kao što se vidi izrazito nelinearna, slika 4. Ipak fazni diskriminator može se predstaviti konstantom pojačanja $2/\pi$ za $\Delta\alpha \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$



Slika 4. Prenosna funkcija faznog diskriminatora na bazi XOR funkcije

Funkcija povratnog prenosa Sistema prikazana je izrazom:

$$G_{ol}(z) = \frac{2\pi}{65536} * \frac{2}{\pi} * Ki * \frac{z * \left(\frac{Kp}{Ki} + 1\right) - \frac{Kp}{Ki}}{(z - 1)^2}$$

Karakteristični polinom funkcije spregnutog prenosa iznosi:

$$z^2 + z \left[\frac{2\pi}{65536} * \frac{2}{\pi} * (Kp + Ki) - 2 \right] - \left[\frac{2\pi}{65536} * \frac{2}{\pi} * Kp - 1 \right]$$

Korjeni karakterističnog polinoma su:

$$z_{\frac{1}{2}} = -\frac{1}{2} \left[\frac{2\pi}{65536} * \frac{2}{\pi} * (Kp + Ki) - 2 \right] \pm \frac{1}{2} \sqrt{\left[\frac{2\pi}{65536} * \frac{2}{\pi} * (Kp + Ki) - 2 \right]^2 - 4 \left[1 - \frac{2\pi}{65536} * \frac{2}{\pi} * Kp \right]}$$

Parametre Kp i Ki, treba izabrati tako da sistem u zatvorenoj povratnoj sprezi ima dominantnu vremensku konstantu Td=1s i faktor prigušenja $\varepsilon = 0.707$. Dominantna vremenska konstanta od 1s je uslov, zato što se u konkretnoj aplikaciji radi o SBN uređaju, a opšte je prihvaćeno da se ne dozvoljava veća dinamika promjene učestanosti od 1Hz/sec. Faktor prigušenja usvaja se proizvoljno (ovde je usvojen optimalan odziv). Potrebno je dakle odrediti konstante K₂ i K, tako da bude zadovoljena jednakost:

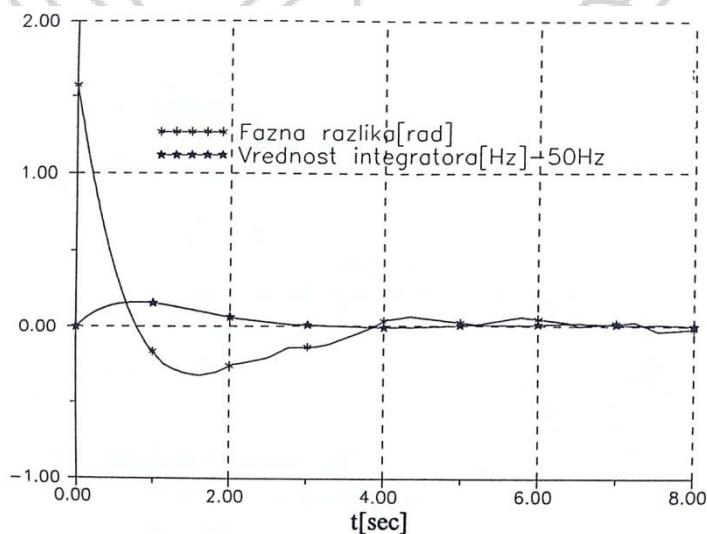
$$z_{\frac{1}{2}} = e^{\frac{s_1}{2T}} \quad S_{1/2} = 1 \pm j, \quad T = 400\mu s$$

Iz postavljenog uslova moguće je izračunati konstante Kp i Ki koje za ovaj slučaj uzimaju vrijednosti:

$$Kp = 13.101958518$$

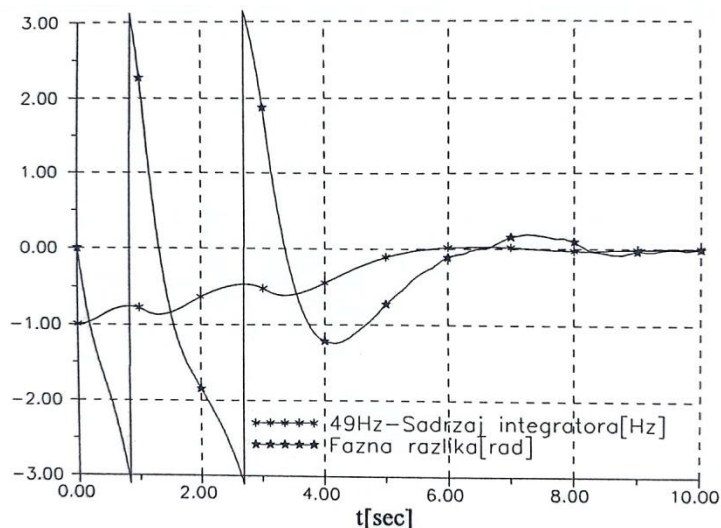
$$Ki = 5.2407832669 \cdot 10^{-3}$$

Izloženi postupak sinteze namjenjen je sinhronizaciji mikroprocesorski upravljanih ispravljača i invertora sa mrežnom učestanošću. Dakle nominalna izlazna učestanost je 50Hz, a dozvoljeni opseg promjene učestanosti ± 1 Hz. Simulacijom se provjeravaju dva režima rada sinhronizatora. Prvi režim je sinhronizacija na talas iste učestanosti, sa početnom razlikom faze $\pi/2$. Karakteristične veličine prikazane su na slici 5. Drugi režim je sinhronizacija na talas različite učestanosti - slika 6.



Slika 5. Sinhronizacija na talas iste učestanosti sa početnom razlikom faze $\pi/2$

Sa slike se vidi da su dobijeni rezultati u potpunosti u skladu sa očekivanjem, tj. Da fazna razlika iščezava sa dominantnom vremenskom konstantom 1s, te da odziv ima konstantu prigušenja = 0.707. Slika 6. prikazuje sinhronizaciju na talas čija se početna učestanost razlikuje od početne učestanosti invertora. Početna razlika faze nije bitna, pa je usvojeno da bude nula.



Slika 6. Sinhronizacija na talas različite učestanosti

Sa slike se vidi uticaj nelinearnosti prenosne funkcije faznog diskriminatora (talasanje krive razlike učestanosti). Na obe slike je umjesto stvarne učestanosti crtan sadržaj integratora. Ovo je usvojeno zbog toga, što bi na razmjeri u kojoj su slike urađene proporcionalno dejstvo imalo priličan uticaj na izgled krive, pa zbog relativno velikog "ripla" stvarne vrijednosti učestanosti, ne bi bilo moguće sagledati dinamiku promjene.

2.2 Programsko rješenje

Kao što je u ranijem tekstu već naznačeno, fazno spregnuta petlja čija je sinteza i realizacija predmet ovog poglavlja, namjenjena je da obavlja sinhronizaciju mikroprocesorski upravljanih ispravljača i invertora sa mrežnim naponom. S obzirom na tu činjenicu potrebno je ograničiti opseg učestanosti na koje se dozvoljava sinhronizacija. Za konkretnu realizaciju nominalna učestanost je mrežna 50Hz, a dozvoljava se sinhronizacija u opsegu učestanosti od 49 do 51Hz. Veći opseg učestanosti se u slučaju invertora ne smije dozvoliti, moraju se imati u vidu zahtjevi priključenog potrošača. Zato je neophodno ograničiti vrijednost integratora. Postupak izračunavanja granica za integrator je sljedeći:

$$\text{Inc}_{\min} = 65536 * f_{\min} * T = 65536 * 49 \text{ Hz} * 400 \mu\text{s} = 1284.51$$

$$\text{Inc}_{\text{nom}} = 65536 * f_{\text{nom}} * T = 65536 * 50 \text{ Hz} * 400 \mu\text{s} = 1310.72$$

$$\text{Inc}_{\max} = 65536 * f_{\max} * T = 65536 * 51 \text{ Hz} * 400 \mu\text{s} = 1336.93$$

$$\text{Int}_{\min} = \text{inc}_{\min} \text{ I } K_i = 3BD6A \text{ hex}$$

$$\text{Int}_{\text{nom}} = \text{inc}_{\text{nom}} \text{ I } K_i = 3D0F4 \text{ hex}$$

$$\text{Int}_{\text{msx}} = \text{inc}_{\max} \text{ I } K_i = 3E47E \text{ hex}$$

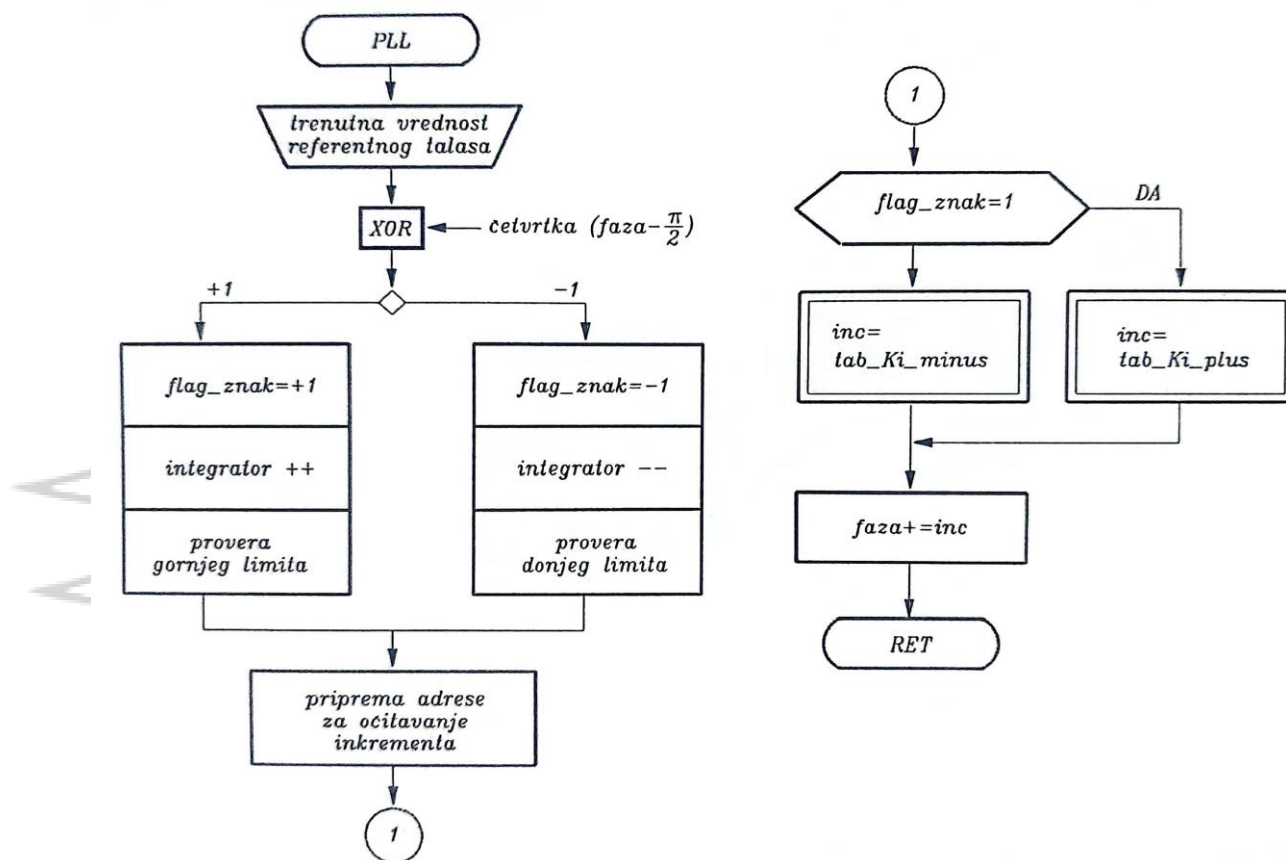
S obzirom na to da u periodu odabiranja nije moguće izvršiti množenje sadržaja integratora konstantom K_i , pa čak ni sabiranje rezultata sa konstantom K_p , bar ne dok se u istom vremenu priprema i adresa za tabelu paljenja tiristora (ispravljač), ili izračunava i PWM (inveror), usvojen je pristup da se na osnovu vrijednosti integratora i znaka na izlazu faznog diskriminatora (takođe realizovan programski) očitavaju unaprijed pripremljene tablice smještene u EPROM-u. Praktično, jedino što se stvarno izračunava je vrijednost integratora (naravno prilikom tog računa uzimaju se u obzir limiti). Tablice se adresiraju na osnovu 10 bita najveće težine, razlike stvarne i minimalne vrijednosti integratora. Kapacitet tablica izračunava se na sljedeći način:

$$\text{razlika} = \text{int}_{\max} - \text{int}_{\min} = 2714 \text{ hex}$$

Dovoljno je koristiti samo 10 najznačajnijih bita razlike, pa kapacitet iznosi:

$$\text{kapacitet} = 271 \text{ hex} * 2 \text{ bajta rezultata} = 1250 \text{ bajta}$$

Već je rečeno da se tablice očitavaju na osnovu vrijednosti integratora i znaka na izlazu iz faznog diskriminatora. Radi se naime o sljedećem. Izlaz iz faznog diskriminatora može uzeti vrijednosti 1 ili -1. Ova činjenica se koristi tako što se za množenje vrijednosti integratora sa konstantom K_i , pripremaju dvije tablice. Jedna sadrži vrijednost izlaza integralnog člana uvećanu za K_p , dok je u drugoj vrijednost iz integralnog člana umanjena za K_p . Ovim se postiže ne samo veća brzina izračunavanja nego i veća preciznost u računu. Dijagram toka programa kojim se realizuje fazno spregnuta petlja prikazan je na slici 7.



Slika 7. Dijagram toka programa kojim se realizuje fazno spregnuta petlja

Postupak je sljedeći:

1. Fazni detektor realizuje XOR funkciju nad znakom faze referentnog i izlaznog talasa zakašnjenog za četvrtinu periode. Na osnovu te operacije formira se signalna promjenljiva "flag_znak" koja se kasnije koristi.
2. Vrijednost integratora se inkrementira ili dekrementira u zavisnosti od vrijednosti "flag_znak". Pri tim operacijama provjeravaju se limiti za vrijednost integratora.
3. Vrš se oduzimanje minimalne vrijednosti integratora (donji limit) od aktuelne, te se na osnovu tog broja određuje ofset za adresiranje odgovarajuće tablice.
4. Na osnovu vrijednosti "flag_znak" određuje se koju tablicu treba koristiti (tab_ki_plus ili tab_ki_minus), te se na osnovu ofseta izračunatog u koraku 3 i početne adrese očitava vrijednost inkrementa.
5. Vrijednost inkrementa dodaje se na trenutnu vrijednost faze, čime je postupak određivanja faze završen. Ova vrijednost faze koristi se u sljedećoj periodi odabiranja na faznom detektoru.

Prikazana realizacija troši oko 60% računskog vremena na mikrokontroleru 80535. Dalje ubrzanje, uz izvesnu degradaciju osobina sinhronizatora, moglo bi se postići tako što bi se vrijednosti K_p i K_i (kao u slučaju sinteze regulatora za ispravljač) izabrale tako da odgovaraju prirodi binarnog računanja. To bi pomjerilo polove sistema, te promijenilo njegove dinamičke osobine, ali bi račun mogao biti drastično ubrzan. Primjenom 16-to bitnog. mikrokontrolera račun bi se nesrazmjerno ubrzao, s obzirom na to da se većina operacija izvodi nad brojevima koji se zapisuju u dva, ili čak tri bajta. Na osmobitnom mikrokontroleru se dobar dio ukupnog vremena troši na memorijske instrukcije za privremeno čuvanje među rezultata.

3. Zaključak

Postoji nekoliko pravaca u kojima bi se moglo raditi u cilju unapređenja prikazane konstrukcije. Kao prvo, upravljačku elektroniku ispravljača i invertora, trebalo bi realizovati na jednom šesnaesto-bitnom mikrokontroleru. Sa ugradnjom performantnijeg mikrokontrolera otvorila bi se mogućnost kvalitetnijeg upravljanja invertorom, u smislu da bi umjesto spore regulacije srednje vrijednosti usmjerenog napona, postalo moguće realizovati i regulaciju trenutne vrijednosti napona. Drugi pravac za poboljšanje konstrukcije, bio bi u smislu reorganizacije postojeće upravljačke elektronike. U osnovnoj konfiguraciji, trebalo bi na baznom mikrokontroleru (onaj na kome se vrši upravljanje ispravljačem i invertorom) realizovati i grubo komandovanje uređajem. Ovakva realizacija bila bi prilagođena ugradnji u jednostavne uređaje male snage. Kao opciju, trebalo bi realizovati mjerno upravljačku jedinicu, čija bi se ugradnja vršila samo na poseban zahtjev. Mjerno upravljačka jedinica, mogla bi biti izmjenjena, tako da osim komunikacije sa nadređenim računarom u lokalnu, podržava i dogradnju modema, čime bi se moglo postići daljinsko očitavanje stanja uređaja. Na taj način smanjili bi se eventualni troškovi održavanja, te bih se ubrzalo kreiranje baze podataka o uslovima eksploatacije, eventualnim manjkavostima konstrukcije i slično.

4. Literatura

- [1] M.R.Stojić, "Kontinualni sistemi upravljanja", Naučna knjiga, Beograd 2015. god.
- [2] M.R.Stojić, "Digitalni sistemi upravljanja", Naučna Knjiga, Beograd 2016. god.
- [3] S. B. Dewan, W. G. Dunford, "A Microprocessor-Based Controller for a Three-Phase Controlled Rectifier Bridge", IEEE Trans. on IA, Vol. IA-19, No. 1, Jan/Feb 1983, pp. 113-119
- [4] A. Mirbod, A. El-Amawy, "A General Purpose Microprocessor-Based Control Circuit for a Three Phase Controlled Rectifier Bridge", IEEE Trans. on IE, Vol. IE-33, No. 3, August 1986, pp. 310-317.
- [5] J. G. Kassakian, M. F. Schlecht, G. C. Verghese "Principles of Power Electronics", Addison-Wesley Publishing Company, Inc. 2020. god.

