Modulacija prostornog vektora za troraziniski pretvarač s neutralnom točkom

Zenzerović, Ive

SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Diplomski sveučilišni studij elektrotehnike

Diplomski rad

Modulacija prostornog vektora za trorazinski pretvarač s neutralnom točkom

Zenzerović Ive

0069083832

Rijeka, rujan 2024.

SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Diplomski sveučilišni studij elektrotehnike

Diplomski rad

Modulacija prostornog vektora za trorazinski pretvarač s neutralnom točkom

Mentor: prof. dr. sc. Neven Bulić

Komentor: v. asist. dr. sc. Nikola Turk

Zenzerović Ive

0069083832

Rijeka, rujan 2024.

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
POVJERENSTVO ZA DIPLOMSKE ISPITE

Rijeka, 15.03.2024.

Zavod: Zavod za automatiku i elektroniku

Predmet: Upravljanje elektromotornim pogonima

ZADATAK ZA DIPLOMSKI RAD

Pristupnik: **Ive Zenzerović (0069083832)**

Studij: Sveučilišni diplomski studij elektrotehnike (1300)

Modul: Automatika (1331)

Zadatak: **Modulacija prostornog vektora za trorazinski pretvarač s neutralnom točkom / Space vector modulation for three Level Neutral Point Clamped Converter**

Opis zadatka:

Prvi dio rada obuhvaća pregled literature postojećih metoda impulsno širinske modulacije za slučaj trorazinskog pretvarača s neutralnom točkom (3L - NPC). Drugi dio rada obuhvaća implementaciju modulatora prostornog vektora za slučaj trorazinskog pretvarača s neutralnom točkom u programskom paketu Plecs. Modulator mora biti implementiran na način da uključuje mrtvo vrijeme te mora generirati okidni signal za upravljački algoritam. U radu je potrebno detaljno opisati princip rada modulatora te ispravnost rada provjeriti na primjeru upravljanja strujama kroz RL trošilo.

Rad mora biti napisan prema Uputama za pisanja diplomskih / završnih radova koje su objavljene na mrežnim stranicama studija.

Zadatak uručen pristupniku: 20.03.2024.

Predsjednik povjerenstva za diplomski ispit:
prof. dr. sc. Dubravko Franković

Mentor:
prof. dr. sc. Neven Bulić

Komentor:
dr. sc. Nikola Turk

Izjava

Izjavljujem da sam diplomski rad izradio samostalno uz pomoć znanja stećenog tijekom studija, navedene literature i pod vodstvom mentora prof. dr. sc. Nevena Bulića i komentora v. asist. dr. sc. Nikole Turka.

Rijeka, rujan 2024.

Ive Zenzerović

Ive Zenzerović

SADRŽAJ

1 UVOD

Energetski pretvarači predstavljaju ključan element u modernim elektroenergetskim sustavima iz razloga što omogućuju precizno upravljanje elektromotornih pogona, obnovljivih izvora energije i drugih električnih sustava.

Jedni od naprednijih pretvarača su pretvarači sa pritegnutnom neutralnom točkom, češće zvani trorazinski pretvarači ili NPC (*engl. „Neutral Point Clamped"*). Ova topologija pretvarača nudi nisko harmoničko izobličenje (*THD*) izlazne struje (kod RL trošila), što ga čini korisnim kod upravljanja elektromotornih pogona visokih snaga. Glavna karakteristika ovih pretvarača je prisutnost neutralne točke čime su omogućene tri naponske razine, ali uz to omogućuje se i balansiranje napona, bolje karakteristike u području premodulacije, itd.

Postoji mnogo metoda upravljanja ovakvim pretvaračima, a ona koja će se obraditi u ovom radu je upravljanje pomoću trorazinskog prostornog vektora, *3L-SPWM* (*engl. „3 Level Space Vector Pulse Width Modulation*). Dok i upravljanje pomoću dvorazinskog prostornog vektora nudi dobre karakteristike, implementacijom trorazinskog prostornog vektora omogućava se znatno preciznija aproksimacija sinusnih valnih oblika.

Cilj rada je napraviti sustav koji će generirati upravljačke modulacijske signale za 3L-NPC pretvarač na temelju *3L-SPWM* modulacije.

2 PLECS

Za implementaciju algoritma upravljanja i simulaciju električnog kruga koristi se programski paket *PLECS* (*„**P**iecewise **L**inear **E**lectrical **C**ircuit **S**imulation“*) [1] razvijen od tvrtke *Plexim GmbH*. *PLECS* je programski alat namijenjen uglavnom za simulaciju krugova energetske elektronike, ali omogućuje i projektiranje sustava upravljanja i fizičkih sustava (termički, magnetni i mehanički sustavi).

U opsegu ovog rada, *PLECS* se koristi za simulaciju svih energetskih pretvarača i za projektiranje upravljačkih sustava istih. U ovom radu, energetski pretvarači su sastavljeni od uobičajenih energetskih komponenti (diode, IGBT-ovi, otpori i induktiviteti), a upravljački sustavi se uglavnom sastoje samo od bloka *C-Script.*

2.1 Podjela koda unutar bloka

Blok *C-Script* [2] je komponenta ugrađena u *PLECS* pomoću koje je moguće implementirati vlastiti sustav sa ulazima i izlazima čije je ponašanje opisano C kodom. *C-Script* blok je opisan kao matematički sustav sa ulazima u, izlazima y, te varijablama unutarnjih stanja x_c i x_d. Navedene varijable povezane su sljedećim jednadžbama [2]:

$$\begin{aligned} y &= f_{output}(t, u, x_c, x_d) \\ x_d^{n+1} &= f_{update}(t, u, x_c, x_d) \\ \dot{x_c} &= f_{derivative}(t, u, x_c, x_d) \end{aligned} \tag{2.1}$$

Svaka od ove 3 jednadžbe se definira u posebnim pod-programima unutar bloka, te postoje dodatna dva pod-programa za kodove koji se izvršavaju na početku i kraju simulacije. Moguće je koristiti 8 pod-programa (*Start, Output, Update, Derivative, Terminate, Store Custom State, Restore Custom State* i *Code Declarations*), ali za potrebe ovog rada korištene su samo *Code Declarations,* te *Start, Output* i *Update Function* pod-programi.

Code Declarations je dio koda koji se izvršava jednom na početku simulacije, a u njemu se definiraju korišteni ulazi/izlazi, varijable, konstante, knjižnice, te funkcije koje će se koristiti.

Start Function se također izvršava samo jednom na početku simulacije, a služi za inicijalizaciju vrijednosti varijabli i varijabli stanja ukoliko se koriste.

Update Function kod se poziva periodički u definiranim vremenskim intervalima i u njemu je pohranjen najbitniji dio koda, odnosno kod u kojem se vrše svi potrebni izračuni algoritma.

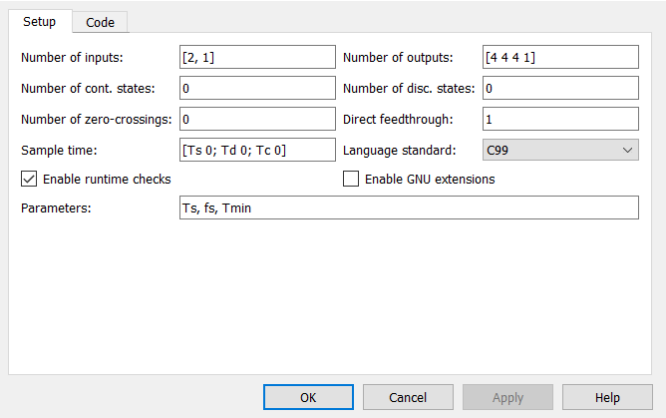
Output Function kod se također poziva u definiranim vremenskim intervalima, a glavna funkcija mu je „slanje“ vrijednosti na izlazne priključke bloka.

2.2 Vrijeme uzorkovanja

C-Script blok se može definirati kao kontinuirani, diskretni ili hibridni sustav. Za potrebe ovog rada, blok je definiran kao diskretni sustav sa više vremena uzorkovanja što znači da se određeni dijelovi koda pozivaju u različitim fiksnim vremenskim intervalima.

C-Script blok ima već ugrađene funkcije kojima je olakšano korištenje svih mogućnosti bloka. Bitna ugrađena funkcija u ovoj situaciji je *IsSampleHit(**n**)*. Ova funkcija vraća *boolean* vrijednost u obliku okidačkog signala, a parametar *n* je redni broj vremenskog intervala definiranog unutar bloka.

Na slici ispod prikazan je prozor sa postavkama *C-Script* bloka korištenog u ovom radu, a pojedine vrijednosti opisane su u tablici ispod slike.



Slika 1. Postavke C-Script bloka

Number of inputs	Broj ulaza u blok	[2 1] definira 2 ulaza, a prvi ulaz se sastoji od 2 signala
Number of outputs	Broj izlaza iz bloka	[4 4 4 1] definira 4 izlaza, a prva 3 izlaza se sastoje od 4 signala
Sample time	Vremenski intervali pozivanja bloka	$[Ts\ 0; Td\ 0; Tc\ 0]$ definira 3 diskretna vremenska intervala pozivanja bloka

Parameters	Parametri definirani izvan bloka koji se koriste unutar bloka	T_s, f_s, T_{min} su parametri definirani u incijalizacijskoj skripti u *PLECS*-u

Tablica 1. Korištene postavke C-Script bloka

3 ENERGETSKI PRETVARAČI

Energetski pretvarači su uređaji koji povezuju dva ili više energetska sustava. Povezivanje ovih energetskih sustava vrši se promjenom električne veličine prvog sustava, te dovođenjem promijenjene vrijednosti u drugi sustav. Električne veličine koja se mijenjaju najčešće su struja, napon i frekvencija. Osnovna podjela energetskih pretvarača je prema valnom obliku ulazne i izlazne veličine:

- Izmjenično-izmjenični *(AC/AC)* pretvarači
- Izmjenično-istosmjeni *(AC/DC)* pretvarači,
- Istosmjerni-istosmjerni *(DC/DC)* pretvarači,
- Istosmjerno-izmjenični *(DC/AC)* pretvarači.

AC/AC i *DC/DC* mogu biti izravni ili neizravni. Kod izravnih pretvarača se ulazna energija direktno pretvara u točan oblik izlazne energije, dok se kod neizravnih pretvarača vrši dvostruka pretvorba ulazne energije. Tako se kod neizravnog *AC/AC* pretvarača ulazna energija pretvara iz izmjenične u istosmjernu, pa na izlazu opet u izmjeničnu. Na istom principu funkcioniraju i neizravni *DC/DC* pretvarači, odnosno ulaznu istosmjernu energiju pretvaraju u izmjeničnu, a potom opet u istosmjernu.

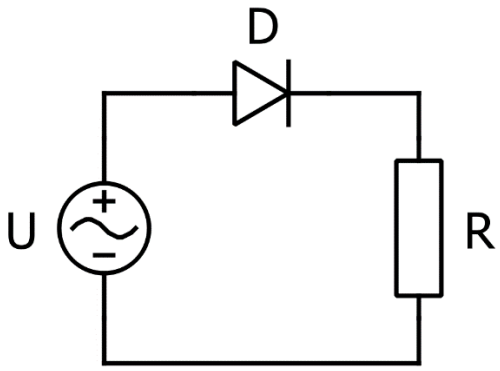
AC/DC pretvarači se češće nazivaju ispravljači zbog toga što se koriste za „ispravljanje“ izmjeničnog valnog oblika na ulazu kako bi se dobio istosmjerni valni oblik na izlazu. Ovi sklopovi se često koriste za smanjivanje i ispravljanje napona kod npr. punjača za mobitele.

DC/AC pretvarači, nazivaju se izmjenjivači (*engl. „Inverter“*) i to su uređaji koji istosmjerni ulazni valni oblik pretvaraju u izmjenični izlazni valni oblik. Izmjenjivaći su danas najčešće pretvaraći koji rade u sklopnom načinu rada.

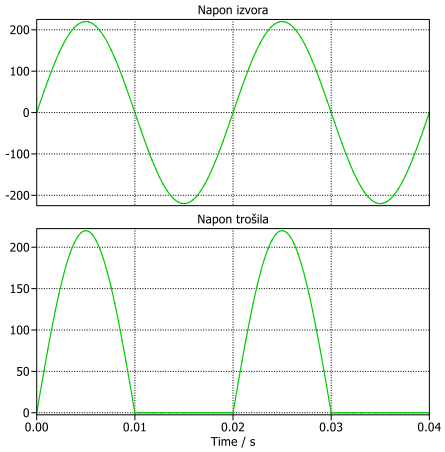
Važno je napomenuti da se neke topologije energetskih pretvarača mogu koristiti i kao ispravljači i kao izmjenjivači, odnosno omogućavaju dvosmjerni tok energije. Postoji više uvjeta da bi ovo bilo moguće, ali najosnovniji je da je pretvarač aktivnog tipa, odnosno da se ne koriste diode nego antiparalelni spoj tranzistora i diode.

3.1 Ispravljači

AC/DC pretvarači, ili ispravljači, su sklopovi kojima se istosmjerna energija na ulazu pretvara u izmjeničnu na izlazu. Ova funkcionalnost je moguća zbog svojstva dioda da vode samo kad su pravilno polarizirane. Na slici ispod prikazana je najjednostavnija topologija ispravljača sa jednom diodom.



Slika 2. Shema ispravljača sa jednom diodom

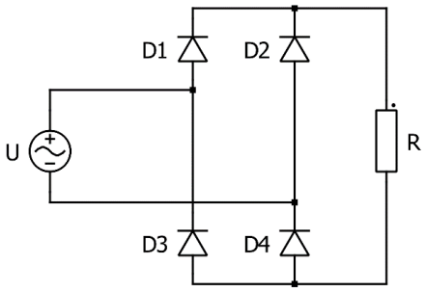


Slika 3. Valni oblici napona na ulazu i izlazu iz ispravljača

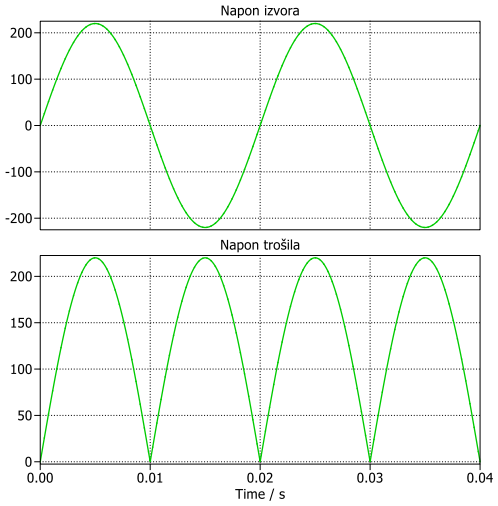
Ispravljači koji na izlazu imaju ovakav (poluvalni) valni oblik nazivaju se poluvalni ispravljači. Uz zanemarenje gubitaka, učinkovitost ovakvih ispravljača je $\eta \approx 40{,}6\%$ [3]. Zbog niske korisnosti se ovaj sklop ne primjenjuje često, ali postoje niskonaponski sustavi kojima niska iskoristivost nije problem, npr. jefitni modeli pumpa za akvarije.

Kako bi se podigla iskoristivost ispravljača potrebno je na izlazu imati punovalni valni oblik, odnosno potrebno je na izlaz prenositi napon tijekom cijelog perioda. Ovakvi ispravljači nazivaju se punovalni ispravljači i najjednostavnija topologija prikazana na slici 4. Ovakav ispravljač naziva se jedonfazni diodni mosni spoj, H-most ili Graetzov spoj. Iskoristivost punovalnih ispravljača iznosi $\eta \approx 81{,}2\%$ [3], što je dovoljno visoko da se implementiraju u jednostavnije sustave.

Kako bi se ispravljaču dodala funkcionalnost upravljanja iznosom napona, potrebno je umjesto dioda dodati aktivne poluvodičke elemente, odnosno tiristore ili tranzistore. Pravovremenim uključivanjem sklopki na izlazu se pojavljuje napon samo u određenim trenutcima čime je moguće regulirati iznos napona od $0V$ do napona izvora. Kako bi se sklopke pravovremeno uključivale potrebno je implementirati sustav upravljanja sklopkama čime se značajno povećava kompleksnost ispravljača.



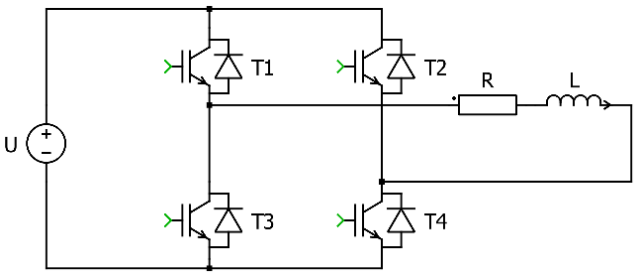
Slika 4. Shema punovalnog ispravljača u mosnom spoju



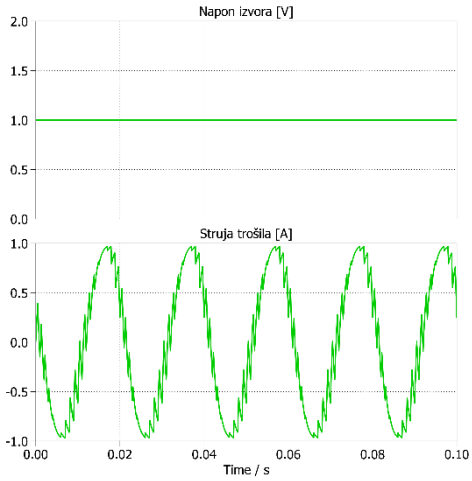
Slika 5. Valni oblici ulaznog i izlaznog napona

3.2 Izmjenjivači

Kako je navedeno u prethodnom poglavlju, izmjenjivači su sklopovi koji služe za „izmjenjivanje“ istosmjernog valnog oblika. Izmjenjivanje je dakle proces pretvorbe istosmjernog i izmjenični valni oblik. To se postiže pravovremenim sklapanjem tranzistora kako bi se na izlazu dobio željeni valni oblik. Najjednostavnija topologija izmjenjivača je mosni spoj prikazan na slici ispod. Sama topologija je ista kao i kod mosnog spoj za ispravljanje napona, jedino se kao sklopke ne koriste diode već tranzistori s antiparalelno spojednim diodama. Također, za razliku od ispravljača, tok energije kod izmjenjivača je od istosmjerne prema izmjeničnoj strani pretvarača.



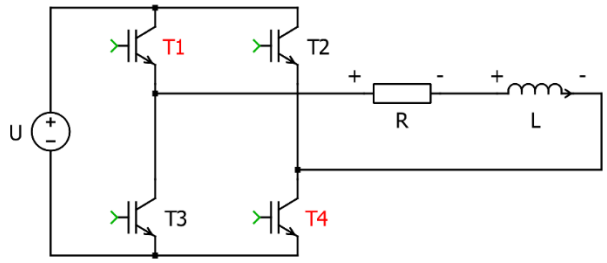
Slika 6. Shema mosnog spoja u izmjenjivačkom režimu rada



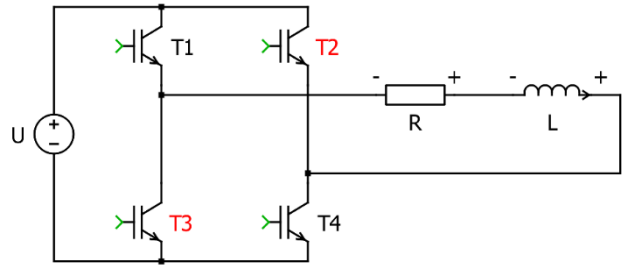
Slika 7. Valni oblik napona izvora i napona na otporu

Sve topologije izmjenjivača se baziraju na izmjeničnom polariziranju trošila uslijed uključivanja/isključivanja određenih sklopki. Princip rada je najjednostavnije objasniti na osnovnoj topologiji izmjenjivača prikazanoj na slici 6.

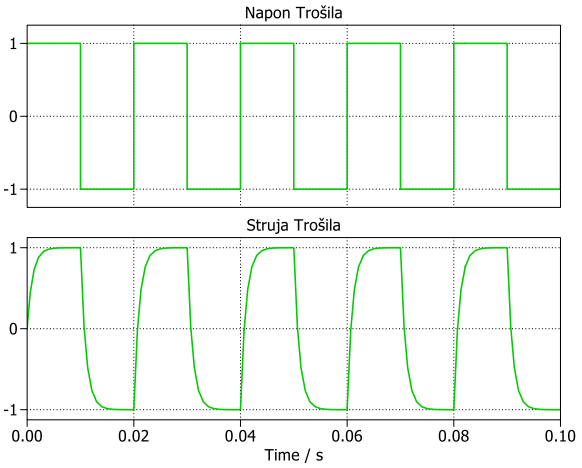
Na slikama 8. i 9. prikazana su stanja sklopki u dvije faze izmjenjivanja. Sklopke označene crvenim slovima su uključene, dok su sklopke označene crnim slovima isključene. Vidi se da se naizmjeničnim uključivanjem *T1/T4* i *T2/T3* može mijenjati polaritet napona na trošilu. Samim time dobije se izmjenični napon na trošilu od istosmjernog napona izvora.



Slika 8. Stanje u prvoj poluperiodi



Slika 10. Stanje u drugoj poluperiodi



Slika 9. Napon i struja trošila

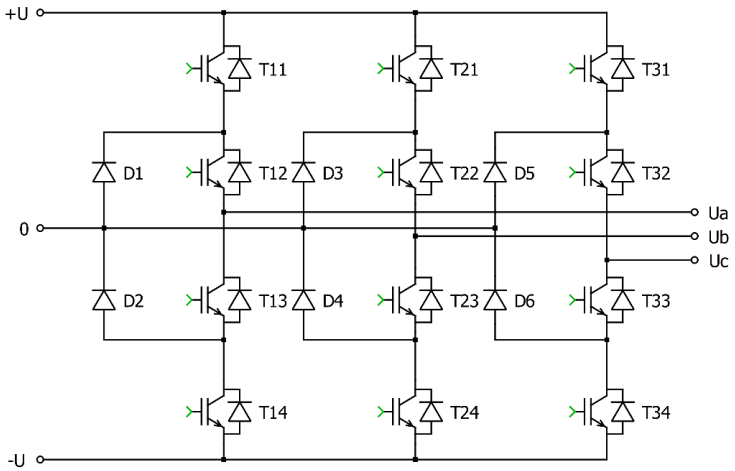
Valni oblici napona i struje trošila dobiveni ovakvim upravljanjem izmjenjivačem prikazani su na slici 9. Dobiveni izmjenični napon na trošilu ima najjednostavniji oblik, ali sadržava harmonike koji štete trošilu i smanjuju efikasnost sklopa. Kako bi se poboljšale karakteristike *PWM* signala potrebno je što točnije aproksimirati sinusni valni oblik, odnosno smanjiti prisutnost viših harmonika, što će biti detaljnije opisano u kasnijim poglavljima.

3.3 *3L-NPC* izmjenjivač

NPC (*engl. „Neutral Point Clamped“*) izmjenjivači su energetski pretvarači koji imaju mogućnpost postizanja većih naponskih razina sa manjim harmoničkim izobličenjem od uobičajenih dvorazinskih pretvarača. Taj efekt postižu korištenjem većeg broja naponskih razina za aproksimaciju sinusoidalnog signala, odnosno umjesto sklapanja između $+\frac{U_{DC}}{2}$ i $-\frac{U_{DC}}{2}$, *NPC* izmjenjivači sklapaju između npr. $+\frac{U_{DC}}{2}, 0$ i $-\frac{U_{DC}}{2}$ [4]. Navedene naponske razine predstavljaju fazne napone pojedinih grana pretvarača. Postoje i kompleksnije izvedbe poput peterorazinskog *NPC* izmjenjivača [5] kod kojeg izlazni napon može poprimiti sljedeće vrijednosti: $-\frac{U_{DC}}{2}, -\frac{1}{4}U_{DC}, 0, +\frac{1}{4}U_{DC}, +\frac{U_{DC}}{2}$.Naravno, sa povećanjem broja naponskih razina znatno se povećava kompleksnost samog pretvarača, a istovremeno i njegovog upravljačkog algoritma.

Korištenjem *3L-NPC* umjesto jednostavnijih dvorazinskih pretvarača (npr. H-most) omogućuje nam korištenje sklopki manjeg nazivnog napona od izlaznog napona pretvarača, smanjenje gubitaka sklapanja, te glavnu prednost ovog sklopa – trorazniski izlazni napon [6].

Topologija izmjenjivača korištena u ovom radu je *3L-NPC.* Puni naziv ove topologije je trorazinski pretvarač sa pritegnutom neutralnom točkom (*engl. „Three Level Neutral Point Clampled“*). Shema *3L-NPC* pretvarača prikazana je na slici ispod.



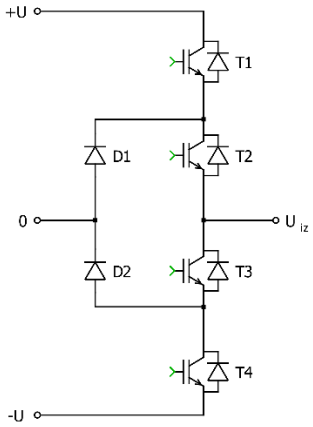
Slika 11. Shema 3L-NPC izmjenjivača

Zamjenjujući diode u svakoj grani sa dva IGBT-a dobije se *3L-ANPC* (*engl. „Three Level Active Neutral Point Clamped“*) topologija. Algoritmi za balansiranje DC napona ostaju nepromijenjeni, ali se korištenjem *3L-ANPC* pretvarača mogu znatno smanjiti gubitci snage, povećati izlazna

frekvencija, te upravljati opterećenosti pojedinih IGBT-a [7]. *3L-ANPC* topologija se neće razmatrati u ovom radu.

3.3.1 Princip rada

Zbog jednakosti, princip rada će se objasniti na samo jednoj grani (faze) sklopa, a ostale dvije grane funkcioniraju po istom principu. Kako bi se na izlaznom priključku dobile 3 naponske razine, koriste se 3 kombinacije sklopnih stanja. Shema grane pretvarača i tablica sklopnih stanja prikazana su ispod.

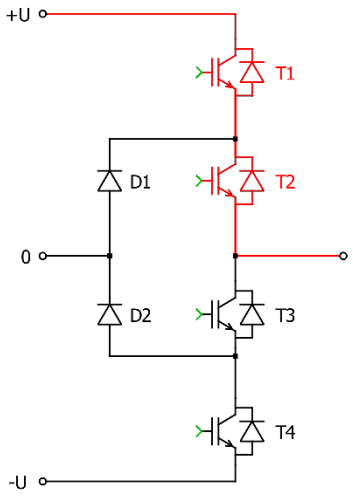


Slika 12. Shema jedne grane 3L-NPC pretvarača

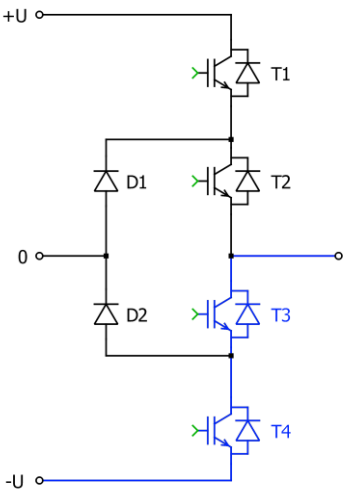
$T1$	$T2$	$T3$	$T4$	U_{iz}
1	1	0	0	$+\frac{U}{2}$
0	1	1	0	0
0	0	1	1	$-\frac{U}{2}$

Tablica 2. Sklopna stanja jedne grane pretvarača

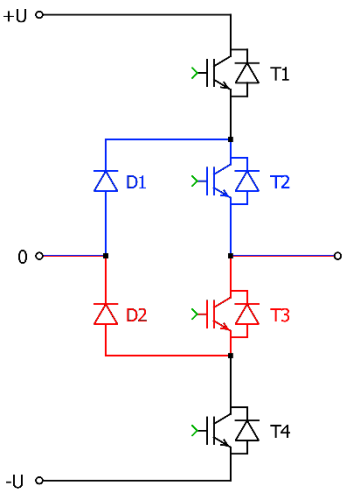
Sklopna stanja prikazana u tablici 2. vizualizirana su na slikama 13., 14. i 15. na način da su uključene sklopke označene crveno (ili plavo), a isključene crno. U prvoj situaciji su uključene sklopke $T1$ i $T2$, pa se preko njih na izlaz dovodi pozitivan napon. Slično tome se u drugoj situaciji preko sklopki $T3$ i $T4$ na izlaz dovodi negativan napon. Slučaj kada se izlaz spaja na $0V$ dijeli se na dva podslučaja ovisno o prethodnom smjeru struje trošila. Ako je na izlazu prethodno bila struja u pozitivnom smjeru, vode $D2$ i $T3$, a ako je bila u negativnom smjeru $D1$ i $T2$.



Slika 13. Stanje pretvarača za generiranje +U na izlazu



Slika 14. Stanje pretvarača za generiranje -U na izlazu



Slika 15. Stanje pretvarača za generiranje 0 na izlazu

U [8] su opisana sva dozvoljena i nedozvoljena sklopna stanja, te podijeljena ovisno o opasnosti za sklop.

Dozvoljena stanja:

- Sve sklopke su isključene → pretvarač je isključen,
- $T2$ ili $T3$ su pojedinačno uključeni,
- Dvije sklopke u nizu su uključene ($T1/T2$, $T2/T3$, $T3/T4$)

Potencijalno štetna stanja:

- $T1$ **ili** $T4$ su pojedinačno uključeni,
- Dvije sklopke koje nisu u nizu su uključene ($T1/T3$, $T2/T4$)

Posljedice potencijalno štetnih stanja ovise o algoritmu upravljanja, te o stanjima ostalih grana pretvarača.

Destruktivna stanja:

- Tri uzastopne sklopke su uključene $T1/T2/T3$ → kratki spoj gornje polovice istosmjernog međukruga, $T2/T3/T4$ → kratki spoj donje polovice istosmjernog međukruga),
- Tri neuzastopne sklopke su uključene ($T1/T2/T4$ → puni DC napon je spojen na $T3$, $T1/T3/T4$ → puni DC napon je spojen na $T2$),
- Sve četiri sklopke su uključene → $+U$, $-U$ i 0 su kratko spojeni

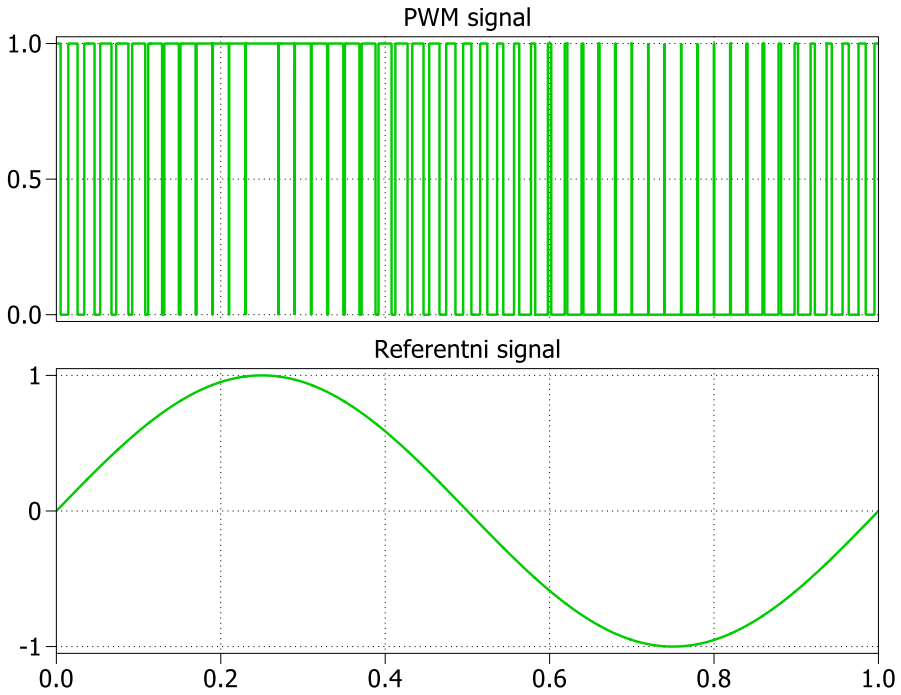
Sva opisana sklopna stanja prikazana su u tablici ispod.

***T*1**	0	0	0	1	0	0	1	0	1	1	0	1	1	1	0	1
***T*2**	0	1	0	1	1	0	0	0	0	0	1	1	1	0	1	1
***T*3**	0	0	1	0	1	1	0	0	0	1	0	1	0	1	1	1
***T*4**	0	0	0	0	0	1	0	1	1	0	1	0	1	1	1	1
Opasnost	Dozvoljeno						Potencijalno štetno					Destruktivno				

Tablica 3. Sklopna stanja podijeljena po opasnosti

4 Pulsno širinska modulacija (*PWM*)

Pulsno širinska modulacija, češće zvana *PWM (engl. „Pulse Width Modulation“)*, je modulacijska tehnika kojom generiramo digitalne impulse promjenjive širine kako bi se aproksimirala amplituda analognog signala. Najveća područja upotrebe *PWM* su upravljanje srednje snage predane trošilu, regulacija napona, te modulacija signala u komunikacijskim sustavima [9]. Nove *PWM* tehnike sve teže jednom cilju – smanjenju ukupnog harmoničkog izobličenja, *THD (engl. „Total Harmonic Distortion“)* izlaznog valnog oblika [10]. Ovisno o situaciji vrijednost *THD-a* ne mora biti od prevelikog značaja, ali je uvijek poželjno aproksimirati sinusoidni valni signal što točnije. Primjer *PWM* signala za aproksimaciju sinusoidalnog signala prikazan je na slici ispod.



Slika 16. Primjer sinusoidalnog PWM signala

Sve modulacijske tehnike baziraju se na istom principu generiranja niza impulsa takvih karakteristika da srednja vrijednost (integral signala) bude što bliži referentnoj vrijednosti u određenom trenutku. Jedan od većih problema je što generiranje ovih impulsa sadrži neželjene više harmonike.

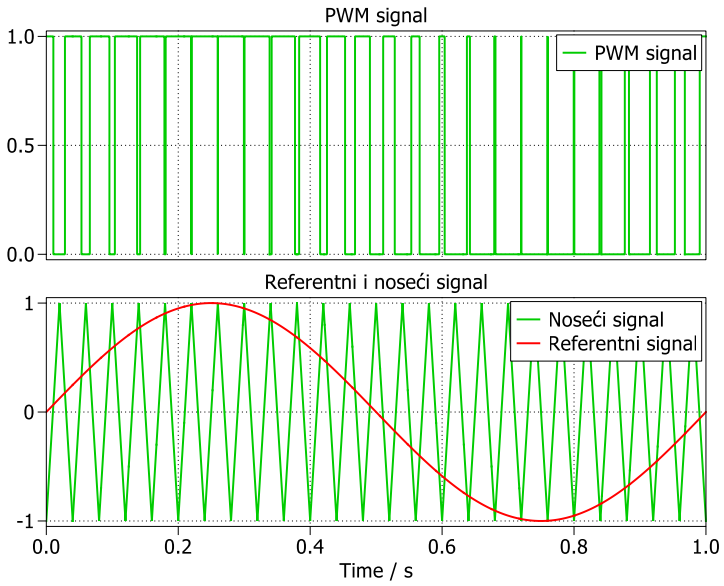
Svaka *PWM* modulacijska tehnika ima glavni zadatak izračunavati vremena trajanja odgovarajućih sklopnih stanja u pretvaraču kako bi se postigla referentna vrijednost napona (ili struje). Kada su vremena trajanja sklopnih stanja izračunata, potrebno je pronaći najefektivniji niz

generiranih upravljačkih impulsa kako bi se npr. minimizirale amplitude viših harmonika, smanjili gubitci sklapanja, ili zadovoljili neki drugi od zadanih kriterija [11].

4.1 Princip rada

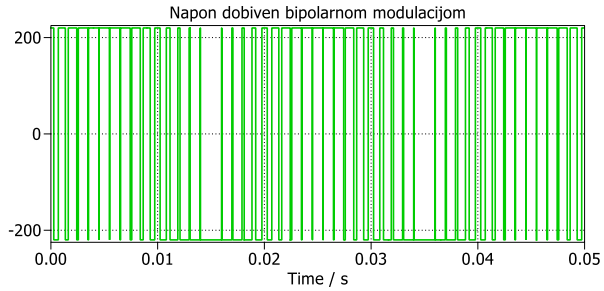
Za generiranje *PWM* signala bilo kojom tehnikom potrebno je koristiti signal nosioc. Signal nosioc je najčešće trokutastog ili pilastog valnog oblika, a njegova frekvencija naziva se sklopna frekvencija pošto ona određuje frekvenciju sklapanja pretvarača. Niz pravokutnih signala moduliranih po širini dobivaju se usporedbom signala nosioca i referentnog signala.

Na slici ispod prikazana je usporedba referentnog i signala nosioca kako bi se generirali *PWM* signali. Sa slike se vidi da je *PWM* signal u visokom stanju kada je referentni signal veći od signala nosioca, odnosno u niskom stanju kada je referentni signal manji od signala nosioca.

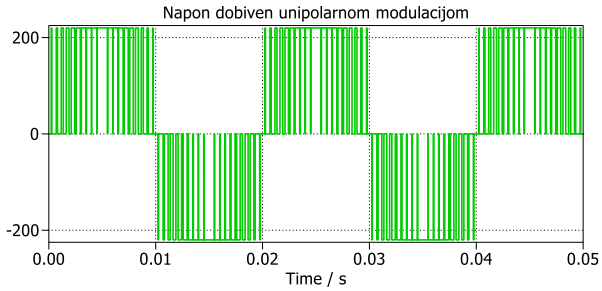


Slika 17. Generiranje sinusoidalnog PWM signala

Postoje dvije glavne podjele *PWM* tehnika: unipolarna i bipolarna modulacija. Kod bipolarne modulacije se napon na izlazu iz pretvarača mijenja između pozitivnog i negativnog napona, dok se kod unipolarne modulacije mijenja između pozitivnog napona i nule u pozitivnoj poluperiodi, te negativnog napona i nule u negativnoj poluperiodi. Primjeri bipolarne i unipolarne modulacije sinusoidalnog signala prikazane su na slikama ispod.

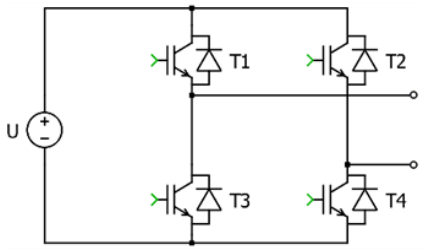


Slika 18. Aproksimacija sinusoidalnog signala bipolarnom modulacijom



Slika 19. Aproksimacija sinusoidalnog signala unipolarnom modulacijom

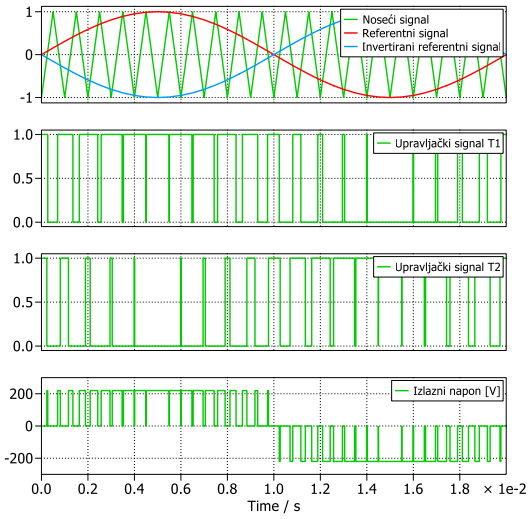
Ove dvije *PWM* tehnike najlakše je objasniti na primjeru H-most pretvarača prikazanog na slici 20.



Slika 20. H-most pretvarač

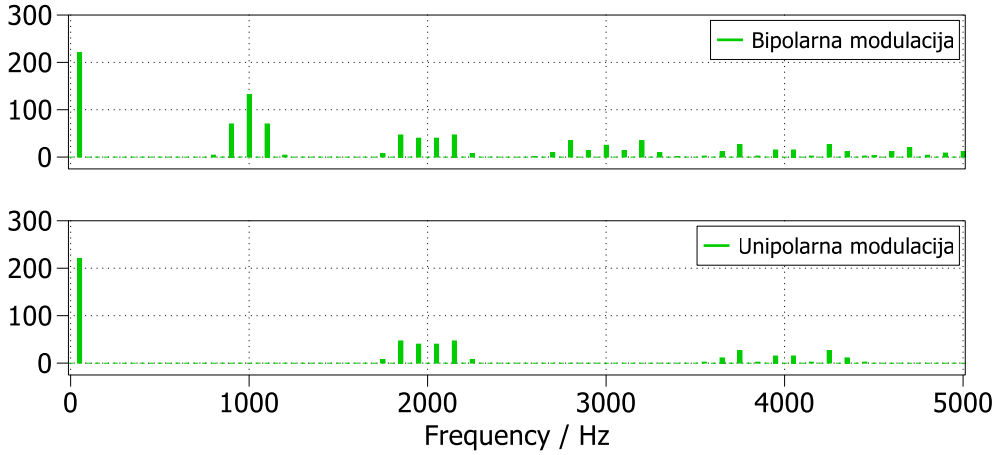
Kod bipolarne modulacije se signal moduliran po principu sa slike 17. dovodi na sklopke *T1* i *T4*, dok se na sklopke *T2* i *T3* dovodi invertiran signal. Iz tog razloga će izlaz pretvarača uvijek biti pod naponom, a sklapati će se između pozitivnog napona (vode *T1* i *T4*) i negativnog napona (vode *T2* i *T3*).

Generiranje upravljačkih signala kod unipolarne modulacije je malo kompleksnije, a grafički je prikazano na slici 21. Upravljački signal sklopke *T1* dobije se usporedbom referentnog signala sa signalom nosiocem (isti princip kao kod bipolarne modulacije). Za generiranje upravljačkog signala sklopke *T2* potrebno je invertirati referentni signal ili signal nosioc, te ih usporediti (isti princip kao kod bipolarne modulacije). Upravljački signal *T3* dobije se invertiranjem upravljačkog signala *T1*, a upravljački signal *T4* invertiranjem upravljačkog signala *T2*.



Slika 21. Princip generiranja upravljačkih signala unipolarnom modulacijom

Na slici ispod prikazane su prisutnosti viših harmonika kod bipolarne i unipolarne modulacije, te je vidljivo da se kod unipolarne modulacije javlja manje viših harmoničkih komponenti. Samim time pokazano je da je unipolarna modulacija bolja kod aproksimacije sinusoidalnih signala, te će negativne posljedice viših harmonika poput zagrijavanja uređaja imati manji utjecaj nego kod bipolarne modulacije [12]. S druge strane, bipolarna modulacija zahtijeva samo jedan signal nosioc, čime je automatski jednostavnija za implementirati i upravljački sustav je manje kompleksan.



Slika 22. Analiza harmoničkih komponenti kod unipolarne i bipolarne modulacije

5 Prostorni vektor

Modulacija prostornog vektora (*engl. „Space Vector Modulation“*), *SVPWM,* je modulacijska tehnika dovođenja određenog *naponskog vektora* na trofazni električni stroj. Ovakav način upravljanja trofaznim strojevima razvijen je u radu [13] za primjenu na izmjeničnom servo pogonu bez četkica (*engl. „Brushless AC Servo Drive“*). Od tada do danas je napisano mnogo radova na temu modulacije prostornog vektora, te su razvijene metode koje su dodatno unaprijedile ovaj koncept.

5.1 Ideja prostornog vektora

Zbog međusobno povezane prirode tereta spojenih na invertere (sinkroni i asinkroni strojevi), javila se potreba sa uvođenjem transformacije koordinata koje omogućuju lakše upravljanje u odnosu na stvarne fazne varijable [11].

Radi praktičnosti će se ideja prostornog vektora objasniti na strujama u trofaznom stroju, ali isti proračun vrijedi i za napone u trofaznom stroju. Trenutne vrijednosti struja u trofaznom simetričnom sustavu se mogu opisati kao:

$$\begin{cases} i_a = I_m \cos(\omega_0 t) \\ i_b = I_m \cos\left(\omega_0 t + \dfrac{2\pi}{3}\right) \\ i_c = I_m \cos\left(\omega_0 t - \dfrac{2\pi}{3}\right) \end{cases} \tag{5.1}$$

Gdje je: I_m – amplituda struje
ω – kutna brzina struje

Ove tri struje, iz izraza (5.1), pozicionirane su u prostoru sa pomacima od $120°$. Na taj način je svaka struja predstavljena vektorom sa promjenjivom amplitudom. Iznosi amplituda ovih vektora mijenjaju se prema jednadžbama iz izraza (5.1).

Rezultal vektorskog zbroja sva tri vektora dobije se rotirajući vektor, često zvan fazor, koji rotira kružnom frekvencijom ω_0. Ovaj rotirajući vektor naziva se prostorni vektor.

Prostorni vektor struje može se opisati jednadžbama:

$$\begin{aligned} \vec{I} &= i_a + a \cdot i_b + a^2 \cdot i_c \\ a &= 1\angle 120° \end{aligned} \tag{5.2}$$

Spomenute supstitucijske varijable su imaginarna i realna komponenta vektora struje: i_α i i_β. Uzimajući u obzir ovakav rastav vektora struje, moguće ga je zapisati kao:

$$\vec{I} = i_\alpha + i_\beta \cdot j \tag{5.3}$$

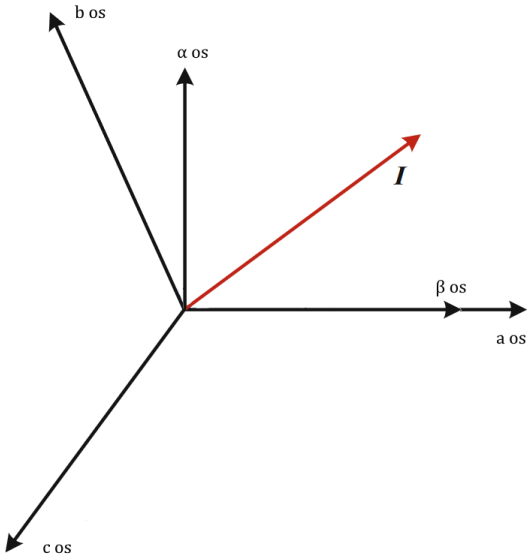
i_α i i_β su zamišljene dvofazne komponente struje u kvadraturnom $\alpha - \beta$ sustavu koje su sa stvarnim trofaznim strujama povezane preko sljedećih jednadžba:

$$\begin{aligned} i_\alpha &= k\left(i_a - \frac{1}{2}i_b - \frac{1}{2}i_c\right) \\ i_\beta &= k \cdot \frac{\sqrt{3}}{2}(i_b - i_c) \end{aligned} \tag{5.4}$$

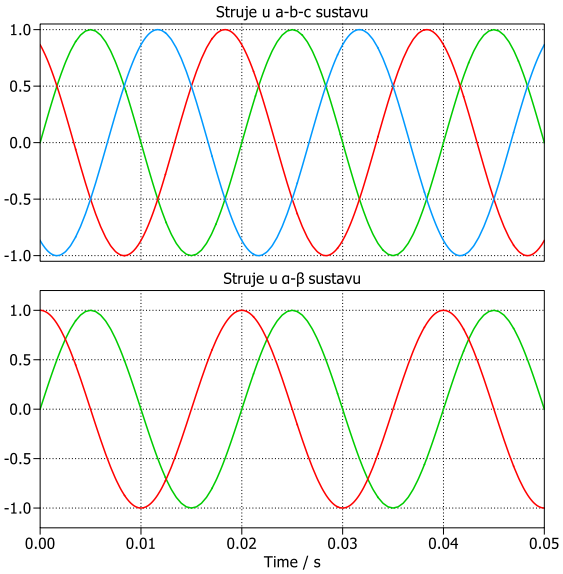
Zapisujući jednadžbe (5.*4*) u matričnom obliku, dobije se transformacijska matrica zvana *Clarke*-ina matrica. Transformacija iz trofaznom $a - b - c$ sustava u stacionarni kvadratni $\alpha - \beta$ sustav naziva se *Clarke*-ina transformacija. Koeficijent k naziva se transformacijska konstanta čija je preporučena vrijednost $k = \frac{3}{2}$ [14].

$$\begin{aligned} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} &= [Clarke\ matrica] \times \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} \\ [Clarke\ matrica] &= \frac{2}{3} \cdot \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} \end{bmatrix} \end{aligned} \tag{5.5}$$

Odnos trofaznog $a - b - c$ i kvadratnog $\alpha - \beta$ sustava, te valni oblici u oba sustava u slučaju sinusoidalnih trofaznih struja prikazani su na slikama ispod.



Slika 23. Odnos a-b-c i α-β koordinatnih sustava



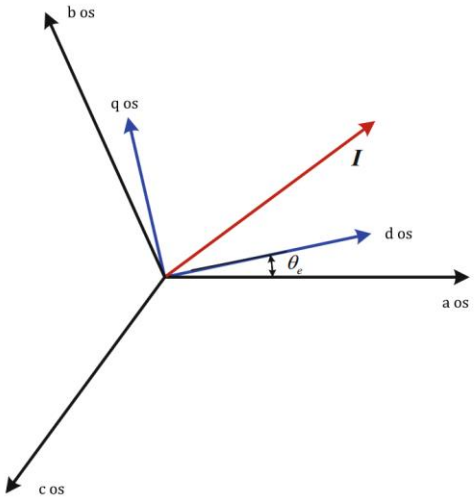
Slika 24. Valni oblici struja u a-b-c i α-β sustavu

Sa slike 24. vidljivo je da su struje u sustavu $\alpha - \beta$ sinusoidalnog oblika, odnosno izmjenične su prirode. Iako je upravljanje pojednostavljeno jer se upravlja sa dvije varijable umjesto tri, upravljanje bi bilo značajno lakše kada bi se struje mogle predstaviti konstantim vrijednostima. Ovo se postiže uvođenjem tzv. rotirajućeg koordinatnog sustava $d - q$. Rotirajući koordinatni sustav $d - q$ rotira istom brzinom kao i fazor struje ω_0, pa su tako i komponente struja i_d i i_q konstantog iznosa.

Transformacija iz dvoosnog stacionarnog $\alpha - \beta$ u rotirajući $d - q$ sustav naziva se *Park*-ova transformacija, a opisana je jednadžbama:

$$\begin{bmatrix} i_d \\ i_q \end{bmatrix} = [Park\ matrica] \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix}$$
$$[Park\ matrica] = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \quad (5.6)$$

U jednadžbi (5.6) θ predstavlja kut između $\alpha - \beta$ i $d - q$ koordinatnih sustava, a kako $d - q$ sustav rotira kutnom brzinom vektora struje ω_0, kut θ je jedna trenutnom kutu vektora struje. Na slici ispod je prikazan odnos $a - b - c$ i $d - q$ koordinatnog sustava.



Slika 25. Odnos a-b-c i d-q koordinatnog sustava

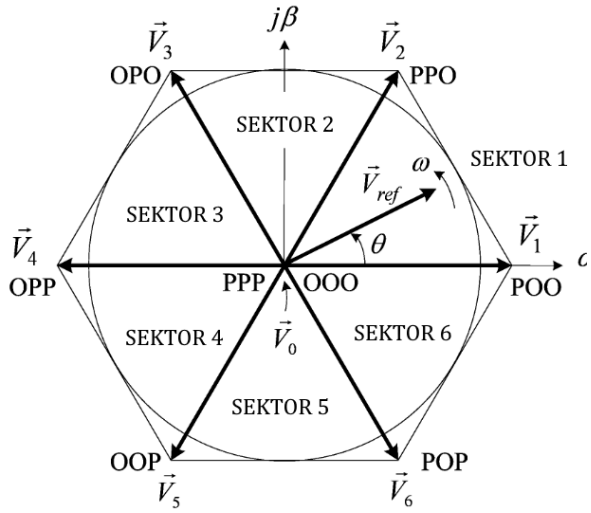
U simetričnim trofaznim sustavima *Park*-ova i *Clark*-ova transformacija su reverzibilni, pa je moguće iznose struja iz $d-q$ vratiti nazad u $\alpha-\beta$ sustav, te iz $\alpha-\beta$ u $a-b-c$ sustav. Ove operacije se vrše tzv. inverznim *Park*-ovim i *Clarke*-ovim transformacijama.

6 *PWM* prostornog vektora

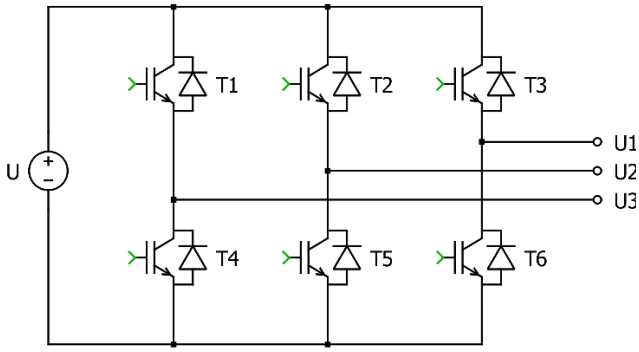
6.1 Dvorazinski *SVPWM*

Iako se u radu implementira modulacija trorazinskog prostornog vektora, radi jednostavnosti će se princip modulacije objasniti na modulaciji dvorazinskog, a zatim proširiti na modulaciju trorazinskog prostornog vektora.

Osnovna ideja modulacije prostornog vektora je pravovremeno generiranje sklopnih signala pretvarača kako bi napon na izlazu iz pretvarača predstavljao određeni prostorni vektor (napon određenog kuta i amplitude). Ovo se postiže korištenjem tzv. sklopnih stanja, odnosno kombinacija stanja sklopki pretvarača koje na izlazu daju određeni vektor napona. Kod dvorazinskih pretvarača postoji 8 sklopnih stanja prikazanih na slici ispod. Svakom sklopnom stanju pridružen je prostorni vektor, te postoji šest aktivnih i dva pasivna prostorna vektora. Aktivni prostorni vektori predstavljaju sklopna stanja kod kojih se na izlaznim granama pretvarača stvara razlika potencijala. Pasivni prostorni vektori prestavljaju sklopna stanja koja na sve 3 grane generiraju isti potencijal, pozitivan napon (*P*) ili beznaponsko stanje (*O*).



Slika 26. Sklopna stanja dvorazinskog pretvarača u α-β ravnini



Slika 27. Shema trofaznog dvorazinskog pretvarača

Sklopna stanja su definirana stanjima pojedinih grana pretvarača, npr. vektor $\vec{V}_2$ dobije se kombinacijom *PPO*, odnosno prva (U_1) i druga (U_2) grana su pod naponom $+U$, a treća (U_3) grana pod negativnim naponom ($-U$). Pravovremenim primjenom odgovarajućih aktivnih i pasivnih vektora na izlazu iz pretvarača može se dobiti prostorni vektor koji rotira željenom frekvencijom, te ima željenu amplituda. U tablici 4. je prikazano svih 8 sklopnih stanja trofaznog dvorazinskog pretvarača, te sklopke koje su uključene za dobivanje određenog vektora. Upisana su samo stanja

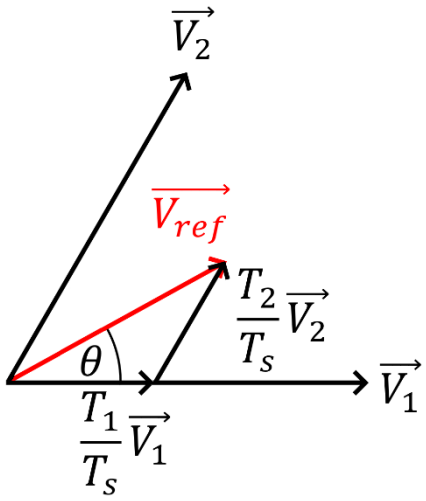
gornjih sklopki (T_1, T_2, T_3) iz razloga što su stanja donjih sklopki (T_4, T_5, T_6) obrnuta od stanja pripadajućih gornjih sklopki.

Prostorni vektor	Sklopno stanje	Uključene sklopke
$\overrightarrow{V_0}$	$[OOO]$	/
$\overrightarrow{V_1}$	$[POO]$	T_1
$\overrightarrow{V_2}$	$[PPO]$	T_1, T_2
$\overrightarrow{V_3}$	$[OPO]$	T_2
$\overrightarrow{V_4}$	$[OPP]$	T_2, T_3
$\overrightarrow{V_5}$	$[OP]$	T_3
$\overrightarrow{V_6}$	$[POP]$	T_1, T_3
$\overrightarrow{V_7}$	$[PPP]$	T_1, T_2, T_3

Tablica 4. Tablica sklopnih stanja trofaznog dvorazinskih pretvarača

Kako bi se postigao što pravilniji sinusoidalni valni oblik, nije dovoljno na izlazu samo postaviti određeni prostorni vektor. Potrebno je unutar sklopne periode na izlaz slati kombinaciju dva susjedna vektora kako bi se aproksimirao referentni vektor kada se on nalazi između aktivnih prostornih vektora. Amplitude pojedinih aktivnih prostornih vektora jednake su amplitudi itosmjernog napona na ulazu u pretvarač, a kutevi su takvi da aktivni vektori dijele krug na 6 jednakih sektora ($\overrightarrow{V_1}$ je pod 0°, $\overrightarrow{V_2}$ je pod 60°, itd.).

Na primjer, referentni vektor $\overrightarrow{V_{ref}}$ na slici 26. se aproksimira određenom kombinacijom prostornih vektora $\overrightarrow{V_1}$ i $\overrightarrow{V_2}$. Kombinacija vektora je ubiti vektorski zbroj dvaju aktivna vektora (prikazano na slici 28.) skaliranih sa određenim koeficijentima. Koeficijenti kojima se skaliraju prostorni vektori $\frac{T_1}{T_s}$ i $\frac{T_2}{T_s}$ predstavljaju vrijeme koliko je pojedini prostorni vektor aktivan unutar jedne sklopne periode.



Slika 28. Zbrajanje vektora

Ovaj postupak opisan je jednadžbama ispod:

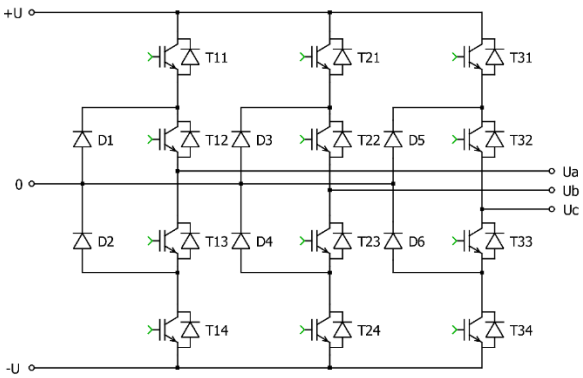
$$\overrightarrow{V_{ref}} = \frac{T_1}{T_s}\overrightarrow{V_1} + \frac{T_2}{T_s}\overrightarrow{V_2}$$

$$\begin{cases} \left|\overrightarrow{V_{ref}}\right| \cdot \sin(\theta) = \frac{T_1}{T_s}\left|\overrightarrow{V_1}\right| \cdot \sin(0°) + \frac{T_2}{T_s}\left|\overrightarrow{V_2}\right| \cdot \sin(60°) \\ \left|\overrightarrow{V_{ref}}\right| \cdot \cos(\theta) = \frac{T_1}{T_s}\left|\overrightarrow{V_1}\right| \cdot \cos(0°) + \frac{T_2}{T_s}\left|\overrightarrow{V_2}\right| \cdot \cos(60°) \end{cases} \quad (6.1)$$

Jedine nepoznaznice u jednadžbama (*6.1*) su T_1 i T_2, pa prema tome imamo jednostavan sustav dvije jednadžbe s dvije nepoznaznice. Vektor u sektoru n se aproksimira prostornim vektorima $\overrightarrow{V_n}$ i $\overrightarrow{V_{n+1}}$ po istom principu.

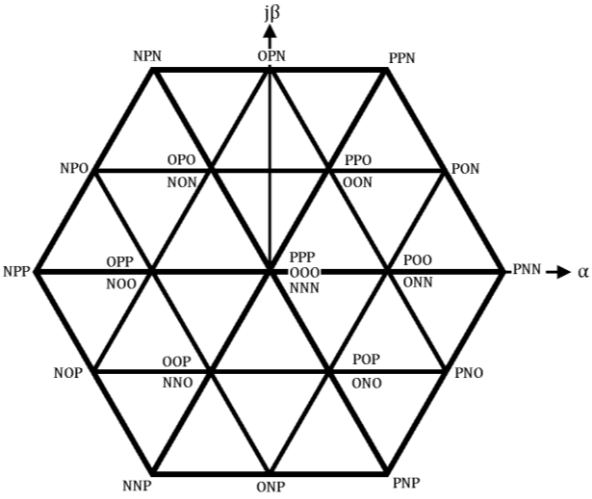
6.2 Trorazinski *SVPWM*

Trorazinski prostorni vektori se baziraju na istom principu kao i dvorazinski, jedina razlika je što se za aproksimaciju vektora umjesto dva koriste tri najbliža aktivna vektora. Kako bi se to postiglo, potrebno je imati pretvarač koji na izlazu može dati tri naponske razine. Iz tog razloga će se koristiti *3L-NPC* pretvarač opisan u poglavlju 3.3. *3L-NPC* pretvarač prikazan je na slici ispod.

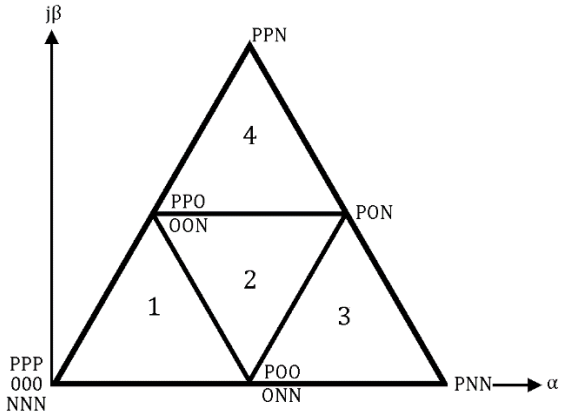


Slika 29. Shema 3L-NPC pretvarača

Tri fazne naponske razine u svakoj grani pretvarača omogućuju tri amplitude prostornih vektora (četiri uz nulvektor): male $\left(\frac{1}{3}U_{DC}\right)$, srednje $\left(\frac{\sqrt{3}}{3}U_{DC}\right)$ i velike $\left(\frac{2}{3}U_{DC}\right)$. Raspodjela trorazinskih prostornih vektora u $\alpha - \beta$ ravnini i raspodjela tzv. *regija* unutar prvog sektora prikazane su na slikama ispod. Nulvektori i mali vektori imaju redudantna stanja, odnosno mogu se ostvariti sa više sklopnih stanja. Nulvektor može se ostvariti sa 3 različita sklopna stanja, a svaki mali vektor može se ostvariti sa 2 različita sklopna stanja.



Slika 30. Trorazinski prostorni vektori u α-β ravnini



Slika 31. Raspodjela regija unutar prvog sektora

Kod modulacije trorazinskog prostornog vektora, referentni vektor se aproksimira kombinacijom tri najbliža prostorna vektora, odnosno sa tri prostorna vektora čiji vrhovi opisuju regiju (trokut) unutar kojeg se referentni vektor nalazi.

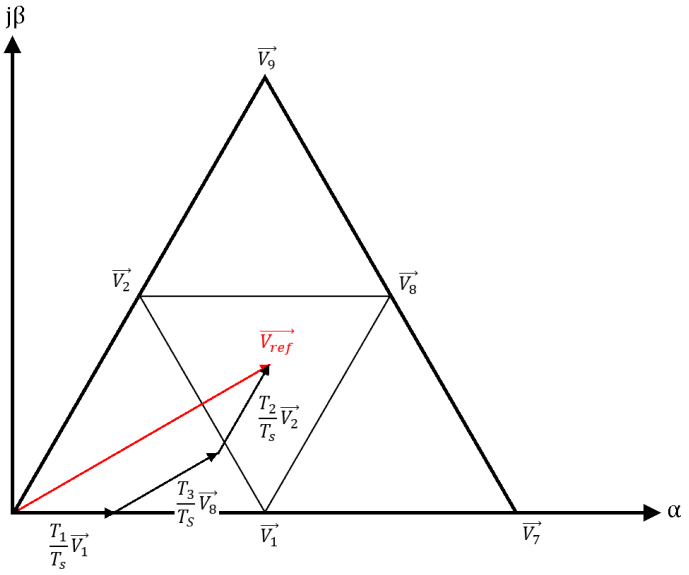
Prostorni vektor	**Sklopno stanje**	**Kategorija prostornog vektora**
$\overrightarrow{V_0}$	$[OOO]$ $[PPP]$ $[NNN]$	Mali vektor
$\overrightarrow{V_1}$	$[POO]$ $[ONN]$	
$\overrightarrow{V_2}$	$[PPO]$ $[OON]$	
$\overrightarrow{V_3}$	$[OPO]$ $[NPN]$	
$\overrightarrow{V_4}$	$[OPP]$ $[NOO]$	
$\overrightarrow{V_5}$	$[OOP]$ $[NNO]$	
$\overrightarrow{V_6}$	$[POP]$ $[ONO]$	
$\overrightarrow{V_8}$	$[PON]$	Srednji vektor
$\overrightarrow{V_{10}}$	$[OPN]$	
$\overrightarrow{V_{12}}$	$[NPO]$	
$\overrightarrow{V_{14}}$	$[NOP]$	
$\overrightarrow{V_{16}}$	$[ONP]$	
$\overrightarrow{V_{18}}$	$[PNO]$	
$\overrightarrow{V_7}$	$[PNN]$	Veliki vektor
$\overrightarrow{V_9}$	$[PPN]$	
$\overrightarrow{V_{11}}$	$[NPN]$	
$\overrightarrow{V_{13}}$	$[NPP]$	

$\overrightarrow{V_{15}}$	$[NNP]$	
$\overrightarrow{V_{17}}$	$[PNP]$	

Tablica 5. Popis trorazinskih prostornih vektora

Oznake sklopnih stanja kod trorazinskih prostornih vektora su slične kao i kod dvorazinskih uz dodavanje oznake O, odnosno P označava da je grana pod pozitivnim naponom, N označava da je grana pod negativnim naponom, a O označava da grana nije pod naponom. Svaki od spomenutih napona predstavlja fazni napon te grane. Princip dobivanja sve tri razine napona u grani pretvarača opisan je u poglavlju 3.3.1.

Primjer izračuna referentnog vektora u drugoj regiji prvog sektora kombiniranjem tri prostorna vektora koja označuju tu regiju prikazan je na slici ispod. Kao i kod dvorazinskih prostornih vektora, koeficijenti $\frac{T_n}{T_s}$ definiraju koliko će pojedini prostorni vektor biti uključen unutar sklopne periode. Više o tome će se objasniti u sljedećem poglavlju.



Slika 32. Primjer izračuna referentnog vektora unutar 2 regije

6.3 Generiranje upravljačkih signala za dvorazinski pretvarač

Radi jednostavnosti, princip generiranja upravljačkih signala će se prvo objasniti na modulaciji dvorazinskih prostornih vektora, te zatim proširiti na trorazinske. Ideja iza generiranja upravljačkih signala je sljedeća – untuar sklopne periode potrebno je u određenom nizu generirati prostorne vektore točnog trajanja kako bi srednja vrijednost unutar te iste sklopne periode bila jednaka vektoru kojeg želimo aproksimirati. Za vrijeme koje je preostalo unutar sklopne periode potrebno je uključiti nul vektor. Niz prema kojem se ukjučuju sklopke naziva se sklopna sekvenca.

Svaka sklopna sekvenca treba imati zadovoljen uvjet $T_s = T_a + T_b + T_c$, gdje T_a, T_b i T_c predstavljaju vrijeme koliko je pojedini prostorni vektor uključen.

Postoji mnogo algoritama sklopnih sekvenci, a glavne razlike su koriste li se za smanjenje sklopnih gubitaka, što zbog smanjenjih zahtjeva hlađenja dovodi do moguće jeftinije proizvodnje. Neke od korištenih algoritama sklopnih sekvenci su simetrična sekvenca i diskontinuirane tehnike pulsno širinske modulacije (*DPWM*) [15].

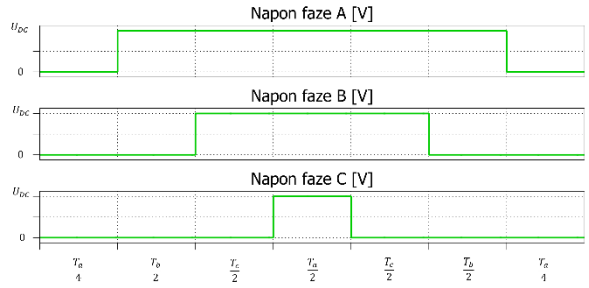
Također, potrebno je obratiti pažnju pri projektiranju sklopne sekvence da se u svakom trenutku mijenja stanje samo jedne grane. Npr. iz stanja $[POO]$ u stanje $[OPO]$ je potrebno prelaziti na sljedeći način: $[POO] \rightarrow [OOO] \rightarrow [OPO]$. Ako se u istom trenutku promijeni stanje dvije grane, moguće je oštetiti sklop kao posljedicu kratkog spoja.

U ovom radu modulacija prostornog vektora biti će implementirana pomoću simetrične sklopne sekvence. U ovom algoritmu je sklopna sekvenca simetrična oko $\frac{T_s}{2}$. Unutar raspona $\frac{T_s}{2}$ se svaki postorni vektor pojavi jednom osim nulvektora koji se pojavi dvaput zbog dva moguća načina generiranja ($[PPP]$ i $[NNN]$). Simetrična sklopna sekvenca se sastoji od sedam segmenata od kojih prostorni vektori zauzimaju 4 segmenata, a preostala 3 segmenta zauzima nulvektor.

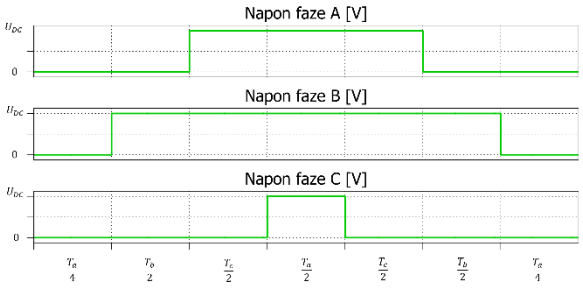
Korištena sklopna sekvenca za dvorazinsku *SVPWM* ima sljedeći oblik:

$$\frac{T_a}{4} \rightarrow \frac{T_b}{2} \rightarrow \frac{T_c}{2} \rightarrow \frac{T_a}{2} \rightarrow \frac{T_c}{2} \rightarrow \frac{T_b}{2} \rightarrow \frac{T_a}{4}$$

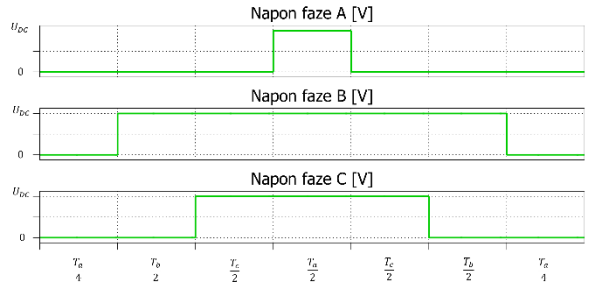
Na slikama ispod prikazani su naponi grana pretvarača za svaki pojedini sektor. Upravljanje amplitudom i frekvencijom napona na izlazu izvodi se promjenom parametara T_a, T_b i T_c.



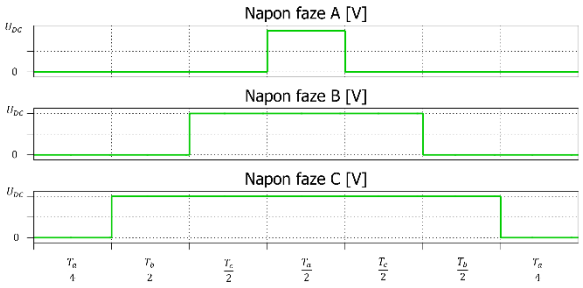
Slika 33. Sklopna sekvenca prvog sektora



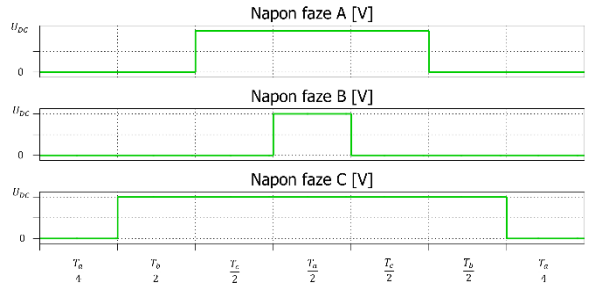
Slika 34. Sklopna sekvenca drugog sektora



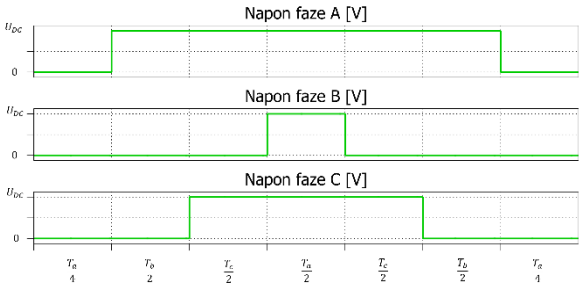
Slika 35. Sklopna sekvenca trećeg sektora



Slika 36. Sklopna sekvenca četvrtog sektora



Slika 37. Sklopna sekvenca petog sektora



Slika 38. Sklopna sekvenca šestog sektora

Vremena T_a, T_b i T_c u sklopnoj sekvenci su prije spomenuti koeficijenti skaliranja prostornih vektora. Računaju se preko sljedećeg sustava jednadžbi:

$$\begin{cases} \left|\overrightarrow{V_{ref}}\right| \cdot \sin(\theta) = \dfrac{T_b}{T_s}\left|\overrightarrow{V_1}\right| \cdot \sin(\alpha_1) + \dfrac{T_c}{T_s}\left|\overrightarrow{V_2}\right| \cdot \sin(\alpha_2) \\ \left|\overrightarrow{V_{ref}}\right| \cdot \cos(\theta) = \dfrac{T_b}{T_s}\left|\overrightarrow{V_1}\right| \cdot \cos(\alpha_1) + \dfrac{T_c}{T_s}\left|\overrightarrow{V_2}\right| \cdot \cos(\alpha_2) \\ T_s = T_a + T_b + T_c \end{cases} \tag{6.2}$$

Gdje je:

$|\overrightarrow{V_{ref}}|$ - magnituda referentnog vektora

θ – kut referentnog vektora

$|\overrightarrow{V_1}|, |\overrightarrow{V_2}|$ - magnitude prvog i drugog prostornog vektora

α_1, α_2 – kutevi prvog i drugog prostornog vektora

T_a – trajanje nulvektora

T_b, T_c – trajanje prvog i drugog prostornog vektora

T_s – trajanje sklopne periode

Odnosno ako prikažemo sustav jednadžbi u matričnom obliku:

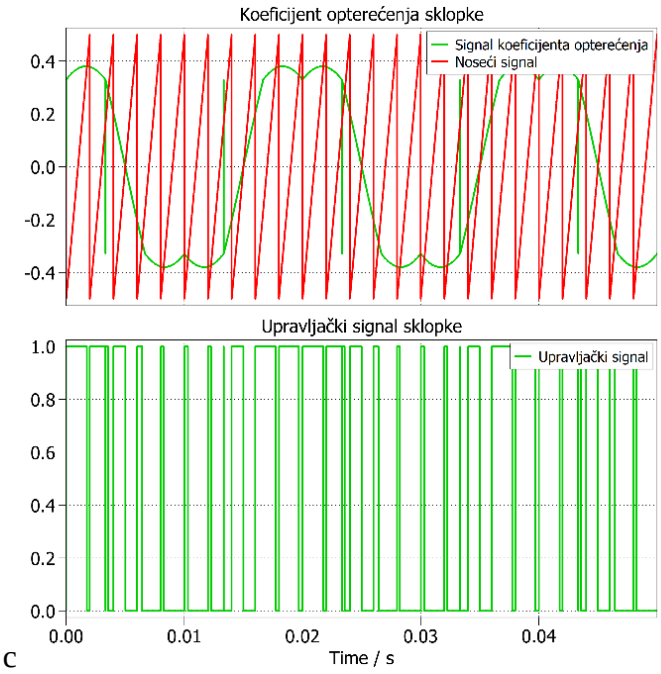
$$\begin{bmatrix} T_s \cdot \left|\overrightarrow{V_{ref}}\right| \cdot \sin(\theta) \\ T_s \cdot \left|\overrightarrow{V_{ref}}\right| \cdot \cos(\theta) \\ T_s \end{bmatrix} = \begin{bmatrix} 0 & \left|\overrightarrow{V_1}\right| \cdot \sin(\alpha_1) & \left|\overrightarrow{V_2}\right| \cdot \sin(\alpha_2) \\ 0 & \left|\overrightarrow{V_1}\right| \cdot \cos(\alpha_1) & \left|\overrightarrow{V_2}\right| \cdot \cos(\alpha_2) \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} T_a \\ T_b \\ T_c \end{bmatrix} \tag{6.3}$$

Promatrajući napone grana pretvarača unutar jednog sektora, može se primjetiti da vrijeme koliko je grana pretvarača pod naponom unutar jedne sklopne periode predstavlja koeficijent opterećenja sklopke u toj grani. Koeficijenti opterećenja sklopki zapravo predstavljaju referentni signal koji se pulsno širinskom modulacijom preslikava na izlaz iz pretvarača. Na primjer, u prvom sektoru prikazanom na slici 33. koeficijenti opterećenja sklopki (gornjih sklopki u sve 3 grane) iznose:

$$\begin{aligned} D_a &= \frac{1}{T_s}\left(\frac{T_a}{2} + T_b + T_c\right) \\ D_b &= \frac{1}{T_s}\left(\frac{T_a}{2} + T_c\right) \\ D_c &= \frac{1}{T_s}\left(\frac{T_a}{2}\right) \end{aligned} \tag{6.4}$$

Gdje je:
D_a – koeficijent opterećenja sklopke T_1 (1. grana)
D_b – koeficijent opterećenja sklopke T_3 (2. grana)
D_c – koeficijent opterećenja sklopke T_5 (3. grana)

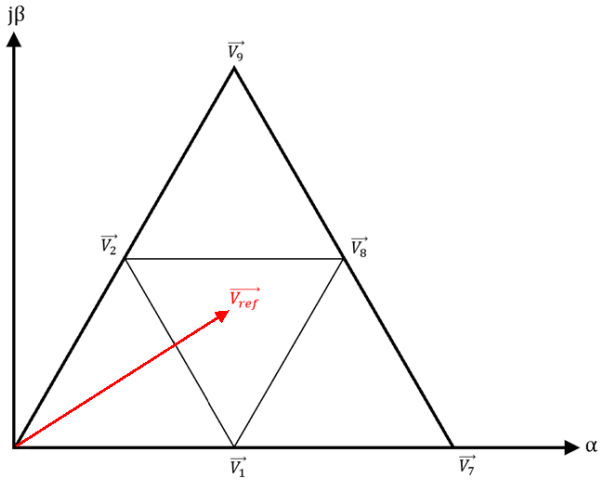
Uspoređujući koeficijente opterećenja sa nosećim *PWM* signalom (opisano u poglavlju 4.1), generiramo upravljačke signale za gornje sklopke svih 3 grana. Primjer signala koeficijenta opterećenja, te generiranog upravljačkog signala jedne grane prikazan je na slici 39.



Slika 39. Primjer generiranja upravljačkog signala sklopke

6.4 Generiranje upravljačkih signala za trorazinski pretvarač

Glavna razlika između generiranja upravljačkih signala za dvorazinski i trorazinski pretvarač je broj prostornih vektora kojima se aproksimira referentni vektor. Kod trorazisnkog *SVPWM* su to 3 prostorna vektora koji opisuju regiju unutar koje se nalazi referentni vektor. Npr. referntni vektor $\overrightarrow{V_{ref}}$ na slici ispod nalazi se u drugoj regiji prvog sektora, pa se zbog toga aproksimira prostornim vektorima $\overrightarrow{V_1}$, $\overrightarrow{V_2}$ i $\overrightarrow{V_8}$.



Slika 40. Primjer aproksimacije vektora u trorazinskom SVPWM

Ako se trorazisnki *SVPWM* uspoređuje sa dvorazinskim, aproksimacija referentnog vektora izvodi se na isti način. U dvorazinskom se aproksimira sa tri prostorna vektora (dva prostorna

vektora i nulvektor), isto kao i kod trorazinskog *SVPWM*. Zbog toga je u kod trorazinske *SVPWM* modulacije moguće primijeniti istu sedam segmentnu sklopnu sekvencu koja je objašnjena u prethodnom poglavlju. Naravno, umjesto nulvektora, koristi se treći vektor koji opisuje regiju referentnog vektora.

Iako je moguće koristiti sedam segmentnu sklopnu sekvencu, problem se javlja zbog toga što sedam segmenata unutar sklopne sekvence nije dovoljno za sve načine generiranja prostornih vektora, što uzrokuje nestabilan napon u DC istosmjernom međukrugu. Iz tog razloga je potrebno koristiti različite sklopne sekvence za različite regije sektora.

U prvoj regiji svakog sektora postoji sveukupno 7 sklopnih stanja, 3 za nulvektor i 4 za preostala 2 prostorna vektora (2 svaki). Zbog toga se za prvu regiju koristi sljedeća 13 segmentna sklopna sekvenca:

$$\frac{T_a}{8} \rightarrow \frac{T_b}{4} \rightarrow \frac{T_c}{4} \rightarrow \frac{T_a}{4} \rightarrow \frac{T_b}{4} \rightarrow \frac{T_c}{4} \rightarrow \frac{T_a}{4} \rightarrow \frac{T_c}{4} \rightarrow \frac{T_b}{4} \rightarrow \frac{T_a}{4} \rightarrow \frac{T_c}{4} \rightarrow \frac{T_b}{4} \rightarrow \frac{T_a}{8}$$

Ovakva sklopna sekvenca osigurava uvjet mijenjaja stanja samo jedne grane u svakom trenutku, te osigurava balansiranje napona u DC međukrugu. U gornjem izrazu sklopna sekvenca zapisana je pomoću vremena uključenosti pojedinih prostornih vektora, zapisujući taj izraz za simbolima vektora dobijemo sekvencu:

$$\overrightarrow{v_0} \rightarrow \overrightarrow{v_1} \rightarrow \overrightarrow{v_2} \rightarrow \overrightarrow{v_0} \rightarrow \overrightarrow{v_1} \rightarrow \overrightarrow{v_2} \rightarrow \overrightarrow{v_0} \rightarrow \overrightarrow{v_2} \rightarrow \overrightarrow{v_1} \rightarrow \overrightarrow{v_0} \rightarrow \overrightarrow{v_2} \rightarrow \overrightarrow{v_1} \rightarrow \overrightarrow{v_0}$$

U drugoj regiji svakog sektora postoje sveukupno 5 sklopnih stanja, pa je sklopnu sekvencu potrebno podijeliti u 9 segmenata kako bi se unutar sklopne periode aktiviralo svako sklopno stanje. Druga regija se sastoji od 2 mala vektora i 1 srednjeg vektora, pa sklopna sekvenca ima sljedeću formu:

$$\frac{T_a}{4} \rightarrow \frac{T_b}{3} \rightarrow \frac{T_c}{2} \rightarrow \frac{T_a}{4} \rightarrow \frac{T_b}{3} \rightarrow \frac{T_a}{4} \rightarrow \frac{T_c}{2} \rightarrow \frac{T_b}{3} \rightarrow \frac{T_a}{4}$$

Ili, zapisano u obliku sa nazivima vektora kao i prije, sklopna sekvenca ima formu:

$$\overrightarrow{v_1} \rightarrow \overrightarrow{v_2} \rightarrow \overrightarrow{v_8} \rightarrow \overrightarrow{v_1} \rightarrow \overrightarrow{v_2} \rightarrow \overrightarrow{v_1} \rightarrow \overrightarrow{v_8} \rightarrow \overrightarrow{v_2} \rightarrow \overrightarrow{v_1}$$

Usporedbom treće i četvrte regije trorazinskih prostornih vektora i sektora dvorazinskih prostornih vektora, može se vidjeti sličnost u smislu postojanja 4 sklopnih stanja. Kod dvorazinskih prostornih vektora, nul vektor ima dva sklopna stanja, te svaki drugi vektor ima jedno sklopno stanje. Kod treće i četvrte regije trorazinskih prostornih vektora, mali prostorni vektor ima

2 sklopna stanja, a veliki i srednji prostorni vektori imaju po jedno sklopno stanje. Prema tome, sklopna sekvenca treće i četvrte regije kod trorazinskih prostornih vektora jednaka je sklopnoj sekvenci kod dvorazinskih prostornih vektora:

$$\frac{T_a}{4} \rightarrow \frac{T_b}{2} \rightarrow \frac{T_c}{2} \rightarrow \frac{T_a}{2} \rightarrow \frac{T_c}{2} \rightarrow \frac{T_b}{2} \rightarrow \frac{T_a}{4}$$

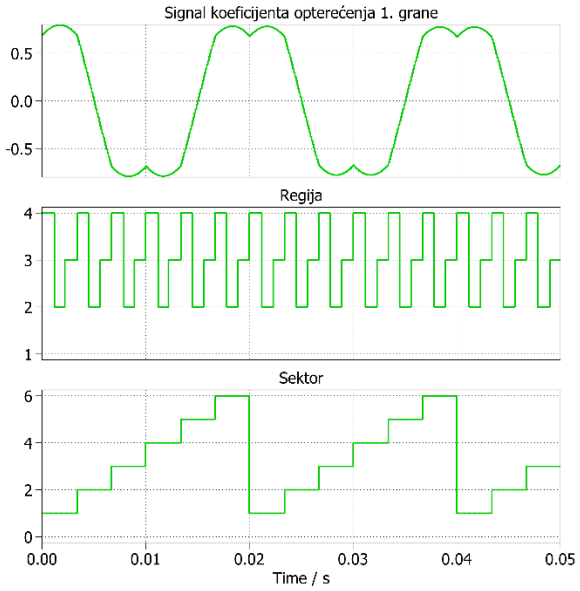
Vrijednosti koeficijanata skaliranja prostornih vektora T_a, T_b i T_c računaju se po istom principu kao i kod dvorazinskih prostornih vektora. Odnosno računaju se skalirani prostorni vektori, koji kad se vektorski zbroje daju referentni prostorni vektor. Rješavanjem sustava jednadžbi (*6.5*) dobiju se skalirajući koeficijenti koeficijenti kojima se generiraju upravljački signali tranzistora.

$$\begin{cases} \left|\overrightarrow{V_{ref}}\right| \cdot \sin(\theta) = \frac{T_a}{T_s}\left|\overrightarrow{V_1}\right| \cdot \sin(\alpha_1) + \frac{T_b}{T_s}\left|\overrightarrow{V_2}\right| \cdot \sin(\alpha_2) + \frac{T_3}{T_s}\left|\overrightarrow{V_3}\right| \cdot \sin(\alpha_3) \\ \left|\overrightarrow{V_{ref}}\right| \cdot \cos(\theta) = \frac{T_a}{T_s}\left|\overrightarrow{V_1}\right| \cdot \cos(\alpha_1) + \frac{T_b}{T_s}\left|\overrightarrow{V_2}\right| \cdot \cos(\alpha_2) + \frac{T_c}{T_s}\left|\overrightarrow{V_3}\right| \cdot \sin(\alpha_3) \\ T_s = T_a + T_b + T_c \end{cases} \quad (6.5)$$

Odnosno, zapisano u matričnom obliku:

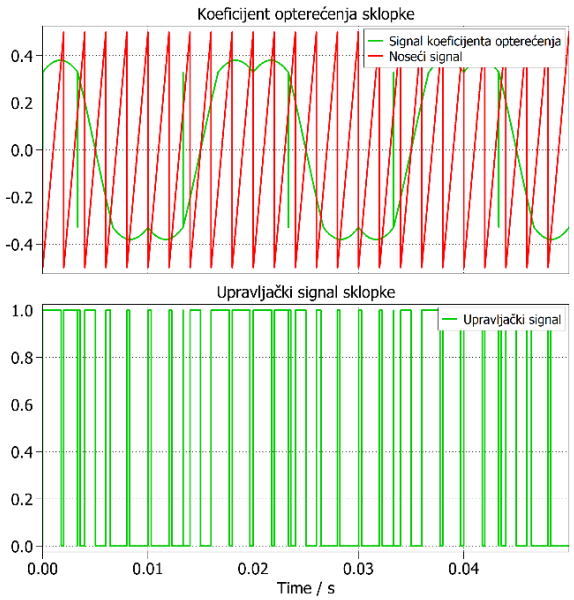
$$\begin{bmatrix} \frac{\left|\overrightarrow{V_1}\right| \cdot \sin(\alpha_1)}{T_s} & \frac{\left|\overrightarrow{V_2}\right| \cdot \sin(\alpha_2)}{T_s} & \frac{\left|\overrightarrow{V_3}\right| \cdot \sin(\alpha_3)}{T_s} \\ \frac{\left|\overrightarrow{V_1}\right| \cdot \cos(\alpha_1)}{T_s} & \frac{\left|\overrightarrow{V_2}\right| \cdot \cos(\alpha_2)}{T_s} & \frac{\left|\overrightarrow{V_3}\right| \cdot \cos(\alpha_3)}{T_s} \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} T_a \\ T_b \\ T_c \end{bmatrix} = \begin{bmatrix} \left|\overrightarrow{V_{ref}}\right| \cdot \sin(\theta) \\ \left|\overrightarrow{V_{ref}}\right| \cdot \cos(\theta) \\ Ts \end{bmatrix} \quad (6.6)$$

Na slici ispod prikazan je primjer generiranog signala koeficijenta opterećenja 1. grane skupa sa regijom i sektorom untuar koje se nalazi referentni vektor. Promatrajući slike 30. i 31., da se zaključiti da će sektor kod bilo koje amplitude referentnog vektora poprimit sve vrijednosti od 1 do 6, a vrijednost regije ovisi o amplitudi referenntog vektora.



Slika 41. Signal koeficijenta opterećenja, te regija i sektor referentnog vektora

Upravljački signali se generiraju na istom principu kao i kod *PWM* dvorazinskog prostornog vektora, odnosno signal koeficijenta opterećenja se uspoređuje sa nosećim signalom. Taj postupak prikazan je na slici ispod.



Slika 42. Generiranje upravljačkih signala

6.5 Utjecaj sklopnih stanja na napon srednje točke *DC* međukruga

Jedan od velikih nedostataka trorazinskog *NPC* invertera je neravnomjerna raspodjela napona u kondenzatorima što uzrokuje gubitak ravnoteže u kondenzatorima *DC* međukruga [16]. Za rješavanje ovog problema se često koristi metoda koja se bazira na izbjegavanju uskih upravljačkih impulsa [16]. U ovom radu se ravnoteža napina kondenzatora u *DC* međukrugu postiže korištenjem redundatnih sklopnih stanja.

Za balansiranje napona u DC međukrugu je potrebno unutar sklopne periode „proći“ sve načine dobivanja prostornih vektora , što nije potrebno kod algoritama za minimalne gubitke sklapanja [16]. Kod takvih algoritama je dovoljno samo generirati samo jedan način dobivanja prostornog vektora.

Implementacija ovog algoritma zahtijeva korištenje sklopnih sekvenci koje untar periode uzorkovanja svaki prostorni vektor aktiviraju minimalno puta koliko prostorni vektor ima redundantnih stanja. Na primjer, nulvektor je u sklopnoj sekvenci potrebno aktivirati tri puta jer postoje 3 načina za generiranje nulvektora - $[PPP]$, $[NNN]$ ili $[OOO]$. Za svaki prostorni vektore srednje veličine postoji tri sklopna stanja kako bi se generirali, pa je unutar sklopne sekvence srednje prostorne vektore potrebno aktivirati minimalno dva puta.

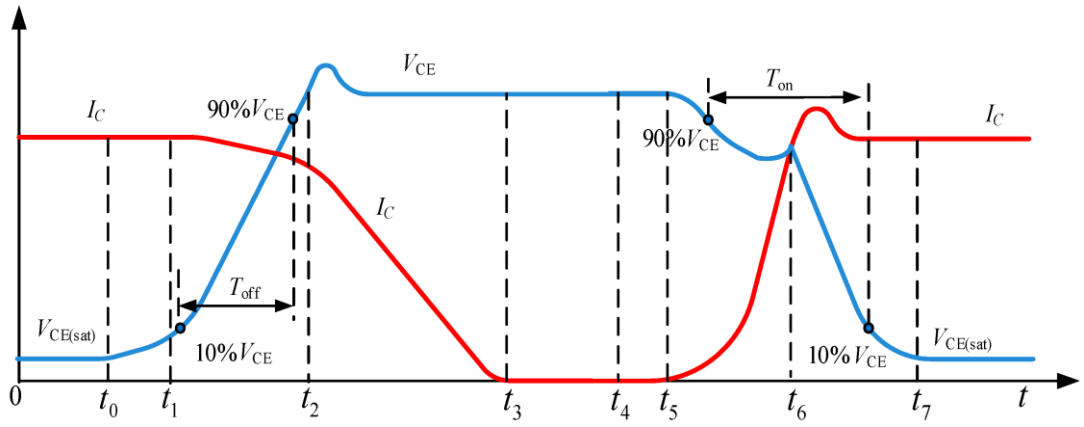
Ako se u upravljački algoritam nebi implementirale metode za ravnotežu napona u *DC* međukrugu može doći do preopterećenja i/ili oštećenja kondenzatora i poluvodičkih elementa. Osim opasnosti od oštećenja sklopa, neravnoteža napona u *DC* međukrugu direktno utječe i na izlazni napon i struju [17].

7 Problematika u realnim sustavima

Princip generiranja upravljačkih signala opisan u prethodnom poglavlju razvijen je za korištenje u idealnim sustavima. U stvarnim sustavima se koriste poluvodičke komponente koje imaju neidalne karakteristike, zbog čega treba prilagoditi algoritam kako bi se izbjegle situacije koje mogu uzrokovati oštećenje sustava. Dvije modifikacije koje su implementirane u ovaj algoritam su mrtvo vrijeme i minimalna širina upravljačkog impulsa tranzistora.

7.1 Mrtvo vrijeme

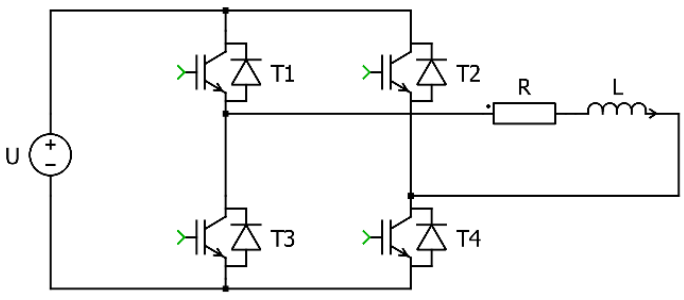
Na slici ispod prikazane su prijelazne pojave IGBT-a pri uključenju ($t_4 - t_7$) i isključenju ($t_0 - t_3$) sklopke (spajanje i odspajanje baze IGBT-a sa napona).



Slika 43. Prijelazna pojava IGBT-a [18]

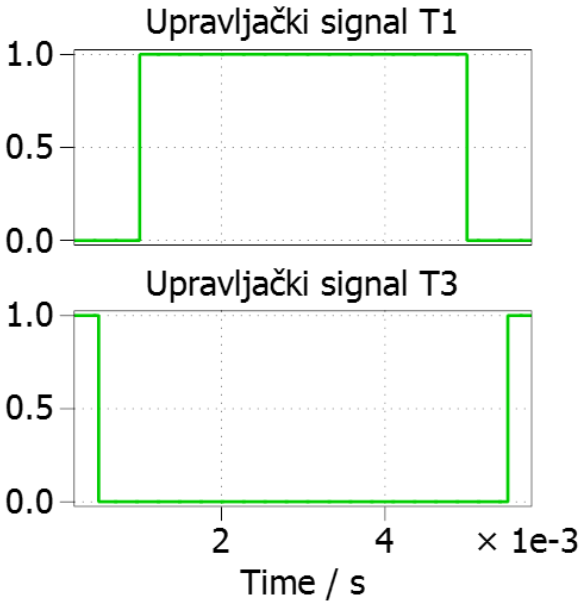
U trenutku t_0 na bazi IGBT-a signal prijeđe iz visokog u nisko stanje, ali se ne pojavi odmah promjena napona U_{CE}. U trenutki t_1 napon U_{CE} počinje rasti, a struja I_C padati. Slična pojava se događa kada signal na bazi IGBT-a prijeđe iz niskog u visoko stanje (od trenutka t_4).

Posljedica ovog efekta može biti kratki spoj istosmjernog izvora kod izmjenjivača. Npr. ako u krugu sa slike ispod istovremeno uključimo T_1 i isključimo T_3, postoji određeni period kada obadva tranzistora provode što znači da je izvor U kratko spojen.

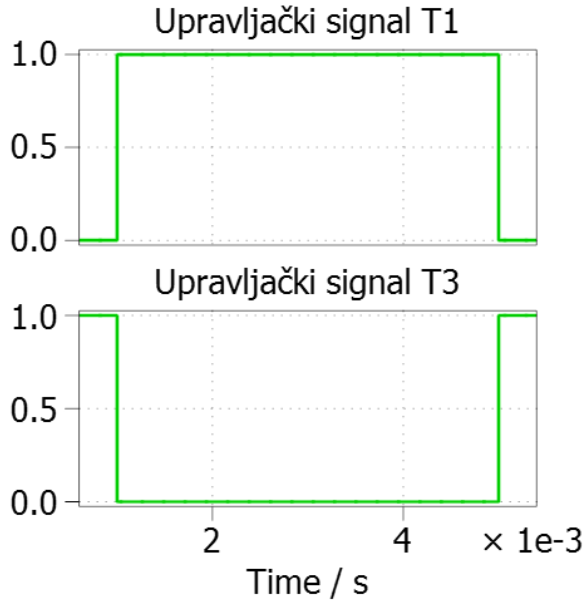


Slika 44. Izmjenjivač

Ovaj problem se rješava dodavanjem tzv. mrtvog vremena u algoritam. Odnosno, uvodi se „pauza“ određenog trajanja između uključenja i isključenja komplementarnih sklopki. Primjeri upravljačkih signala sa i bez mrtvog vremena prikazani su na slikama ispod. Na slici 45. se vidi period od $0{,}5ms$ prije i nakon upravljačkog signala tranzistora T_1 u kojem ne vode ni T_1 ni T_3, čime je osiguran uvjet da jedan tranzistor prestane vodit prije nego drugi provede.



Slika 45. Upravljački signali tranzistora sa mrtvim vremenom



Slika 46. Upravljački signali tranzistora bez mrtvog vremena

Ovisno o implementiranom algoritmu *SVPWM* razlikuju se načini implementiranja mrtvog vremena. Kako se u ovom radu za simulaciju koristi program *PLECS*, te blok *C-Script*, za implementaciju mrtvog vremena će se iskoristiti mogućnost više vremena uzorkovanja skripte. Kako je opisano u poglavlju 3.3.1, sklopke T_1 i T_3, odnosno T_2 i T_4 nesmiju nikada biti istovremeno uključene, a upravljački signali tih sklopki su međusobno komplementarni što znači da je na njihove upravljačke signale potrebno uvesti mrtvo vrijeme. Mrtvo vrijeme je implementirano na način da upravljački signal koji prelazi u visoko stanje kasni za onim koji prelazi u nisko stanje.

U trenutku kada se zadovolji uvjet unutar programa da komplementarne sklopke istovremeno mijenjaju stanje, sklopki koja prelazi u visoko stanje se prelazak odgađa za jedan ciklus programa. Odnosno, ta sklopka će visoko stanje primijeniti tek u sljedećem ciklusu. Ovaj dio programa se vrti vremenskim intervalom jednakim definiram mrtvim vremenom što znači da će visoko stanje sklopke biti primjenjeno tek nakon prolaska definiranog mrtvog vremena. Promjena mrtvog vremena se izvodi u inicijalizacijskoj skripti čime se automatski mijenja i vrijeme uzorkovanja skripte vezano za mrtvo vrijeme.

7.2 Minimalna širina impulsa upravljačkog signala

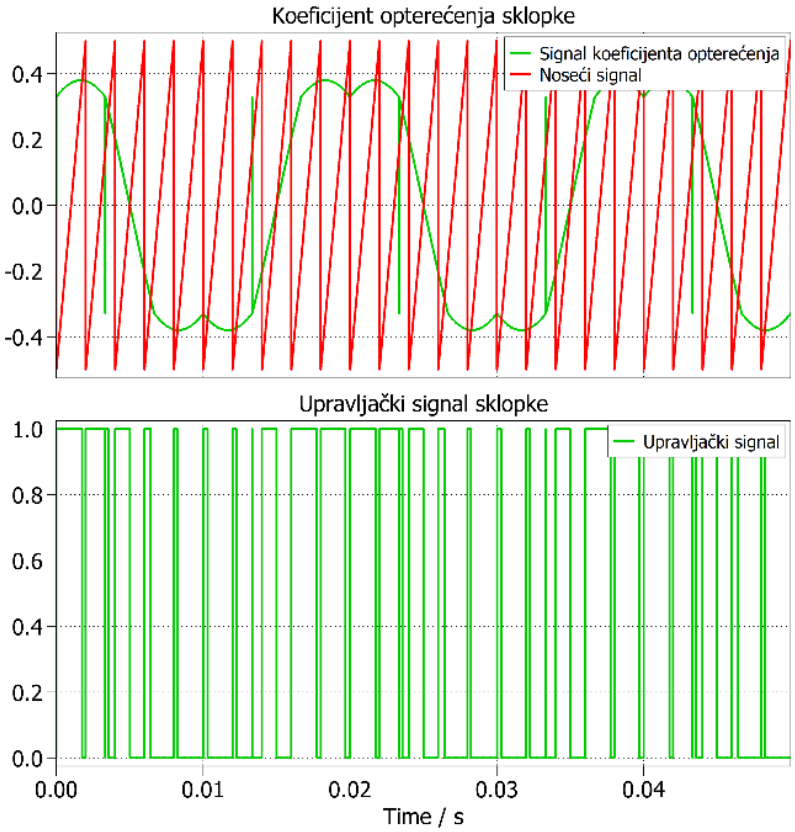
Drugi uvjet koji treba biti zadovoljen u realnim sustavima je minimalna širina impulsa upravljačkog signala. Minimalna širina impulsa kod IGBT-a predstavlja najkraće vrijeme tokom kojeg tranzistor može biti uključen ili isključen kako bi djelovao bez negativnih posljedica ili riskiranja oštećenja sklopa. Ovaj problem se povremeno javlja kod visokofrekventnih PWM pretvarača izvedenih pomoću IGBT sklopki.

Parametar minimalne širine impulsa je potrebno precizno odrediti ovisno o IGBT sklopkama koje se koriste u pretvaraču, a točnu vrijednost potrebno je izračunati ovisno o više faktora. Minimalna širina impulsa, među ostalim, ovisi o brzini uključivanja/isključivanja, parazitskim kapacitetima unutar poluvodiča, te o vršnoj struji za koju je sklopka projektirana [19]. U opsegu ovog rada neće biti izrađen proračun optimalne minimalne širine impulsa, već je samo razvijen algoritam pomoću kojeg se može implementirati željena vrijednost u upravljački sustav.

Problem koji se javlja kod implementacije uvjeta opisanih u ovom poglavlju (mrtvo vrijeme i minimalna širina impulsa) je taj što se ta dva uvjeta „sjeku“. Odnosno, kod krivo unesenih vrijednosti ova dva uvjeta, moguće je da se dogodi situacija u kojoj jedan uvjet ostane ne zadovoljen kako bi drugi bio zadovoljen. U sustavu razvijenom u ovom radu prednost je dana uvjetu mrtvog vremena kako bi se u potpunosti izbjegla mogućnost kratkog spoja izvora.

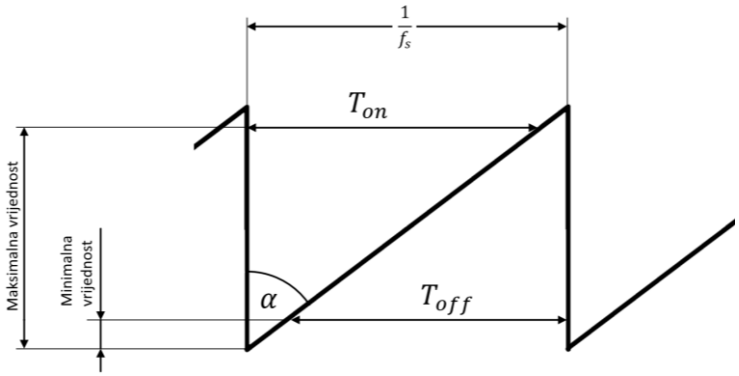
7.2.1 Princip rada

Kako je opisano u poglavlju 6.3 i 6.4, upravljački signali se generiraju usporedbom izračunatog koeficijenta opterećenja sklopke (referentnog signala) sa signalom nosiocem. Radi praktičnosti su ta dva signala, uz generirani upravljački signal sklopke, još jednom prikazana na slici 47. Sa slike je moguće vidjeti, da je širina impulsa upravljačkog signala definirana odnosom između referentnog i nosećeg signala. Da se zaključiti, da se minimalna širina impulsa može ograničiti ograničavanjem referentnog signala unutar određenog raspona.



Slika 47. Primjer generiranja upravljačkog signala sklopke

Spomenuti raspon unutar kojeg se ograničava vrijednost signala koeficijenta opterećenja ovisi o traženoj vrijednosti minimalne širine impulsa [s], frekvenciji nosećeg signala [Hz] i amplitudi nosećeg signala. Vrijeme T_{on} i T_{off} predstavljaju vremena koliko će *PWM* signal biti uključen, odnosno isključen, a izjednačavaju se sa vrijednosti minimalne širine impulsa. Pomoću slike ispod moguće je trigonometrijskim funkcijama izvesti izraze za minimalnu i maksimalnu vrijednost referentnog signala kako bi se zadovoljio uvjet minimalne širine impulsa. Ti izrazi su dani jednadžbama

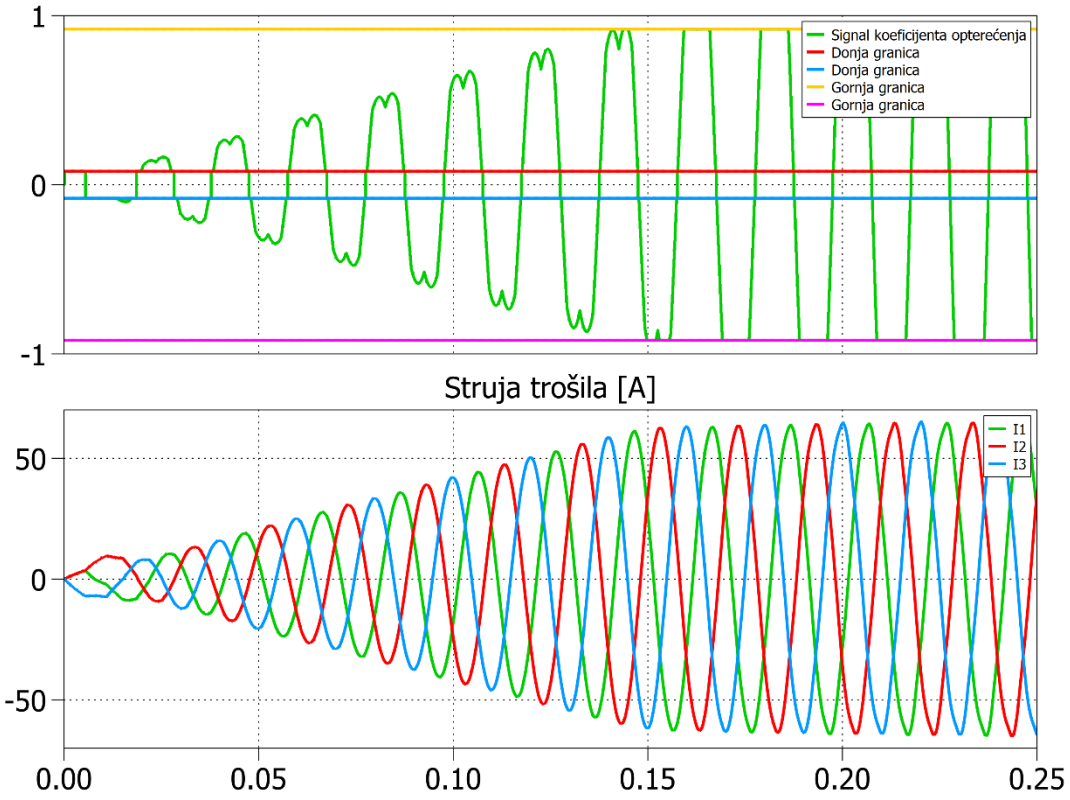


Slika 48. Noseći signal sa označenim vrijednostima za izračun graničnih vrijednosti

$$maksimalna\ vrijednost = \frac{T_{min}}{\tan\left(\frac{\pi}{2} - atan(fs)\right)} \qquad (7.1)$$

$$minimalna\ vrijednost = 1 - T_{min} \cdot f_s$$

Ograničenjem vrijednosti signala koeficijenta opterećenja (referentnog signala) unutar raspona $[minimalna\ vrijednost, maksimalna\ vrijednost]$, modificirani referentni signal izgleda kao što je prikazan na slici ispod.

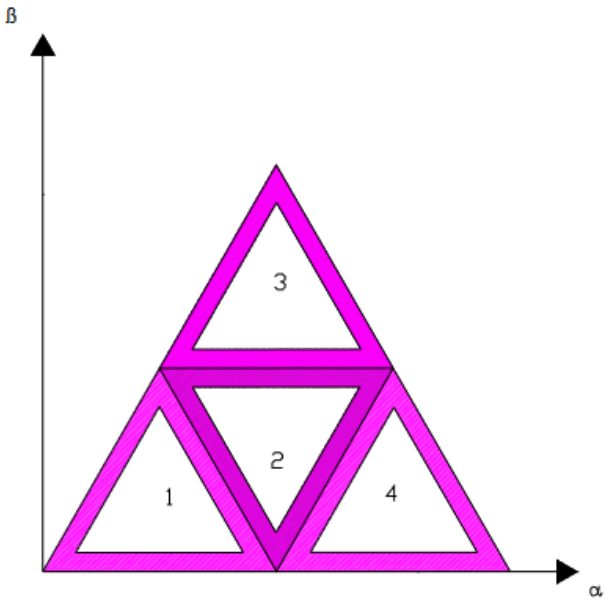


Slika 49. Ovisnost signala koeficijenta opterećenja o graničnim vrijednostima

U trenutku kada signal koeficijenta opterećenja počne ulaziti u zasićenje (u trenutku $t = 0.15$), valni oblik struje na izlazu iz pretvarača počinje biti sve deformiraniji.

Iz slike 49. vidljiv je utjecaj uvjeta mrtvog vremena i minimalne širine impulsa na performanse pretvarača. Uvođenjem tih uvjeta u algoritam, deformira se idealni upravljački signal, što direktno utječe na kvalitetu izlaznog valnog oblika. Iz tog razloga je potrebno precizno izračunati vrijednosti mrtvog vremena i minimalne širine impulsa kako bi se zadržala kvaliteta valnog oblika izlaza pretvarača i osigurao sklop od oštećenja.

Princip generiranja upravljačkih signala sa minimalnom širinom impulsa objašnjen u ovom poglavlju je samo jedan od mnogih načina implementacije istog. U radu [20] implementacija ovog uvjeta izvedena je na način da se ograniči prostor u šesterokutu (slika 30.) unutar kojeg se referentni rotirajući vektor može nalaziti. Na slici ispod je prikazan prvi sektor u $\alpha - j\beta$ ravnini sa ograničenom površinom regija.



Slika 50. Prvi sektor šesterokuta sa ograničenim površinama regija [20]

Prostor unutar regija označen rozom bojom predstavlja zabranjeno područje unutar kojeg se referentni rotirajući prostorni vektor nesmije nalaziti. U trenutku kada referentni vektor dođe u poziciju gdje bi jedan od prostornih vektora, kojima se definira referentni vektor, imao trajanje kraće od definirane minimalne širine impulsa, vremena trajanja prostornih vektora definiraju se na sljedeći način.

$$T_1' = T_{min};\ \ T_2' = \delta \cdot T_2;\ \ T_3' = \delta \cdot T_3$$
$$\delta = \frac{T_s - T_{min}}{T_2 + T_3} \tag{7.2}$$

Gdje je: T_n – idealno vrijeme trajanja prostornog vektora $\overrightarrow{U_n}$

$T_n{}'$ - modificirano vrijeme trajanja prostornog vektora $\overrightarrow{U_n}$

T_s – vrijeme uzorkovanja

T_{min} – minimalna širina impulsa

δ – koeficijent skaliranja vremena T_n

Na ovaj način se prostornom vektoru koji ima trajanje kraće od minimalne širine impulsa, trajanje izjednačava sa minimalnom širinom impulsa, a preostala dva vektora se skaliraju pomoću koeficijenta δ kako bi vrijednila jednakost $T_s = T_1 + T_2 + T_3$.

U radu [20] je dalje navedeno da je utjecaj ovakvog generiranja impulsa minimalne širine na izlazni napon potrebno promatrati od slučaja do slučaja. S obzirom na to da referenti vektor „preskače“ preko zabranjene zone, generirani upravljački signali isti su onima generiranim pomoću „rezanja“ signala koeficijenta opterećenja. Odnosno, na oba načina se „preskače“ zabranjeno područje čime se gubi kontinuiranost rotirajućeg referentnog vektora, što unosi nepravilnosti u valni oblik izlaznog napona.

7.3 Premodulacija

U kontekstu *PWM* invertera, modulacijski koeficijent definiran je kao:

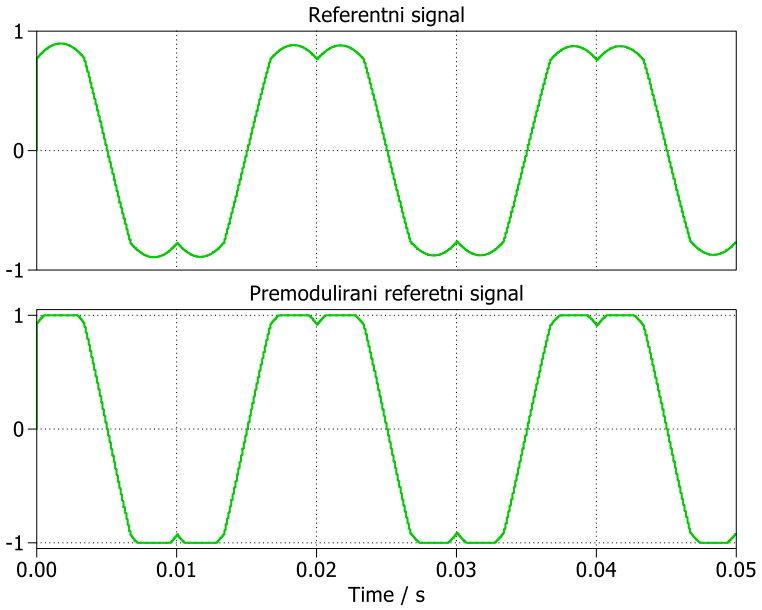
$$m = \frac{A_{ref}}{A_{nosioca}} \tag{7.3}$$

Gdje je: A_{ref} – amplituda referentnog signala

$A_{nosioca}$ – amplituda signala nosioca

U rasponu gdje je modulacija linearna, koeficijent m se kreće od 0 do 1 kako bi se osigurao pravilno skalirani referentni signal. U slučaju kada modulacijski koeficijent postane veći od 1, inverter ulazi u područje premodulacije (*engl. „Overmodulation“*).

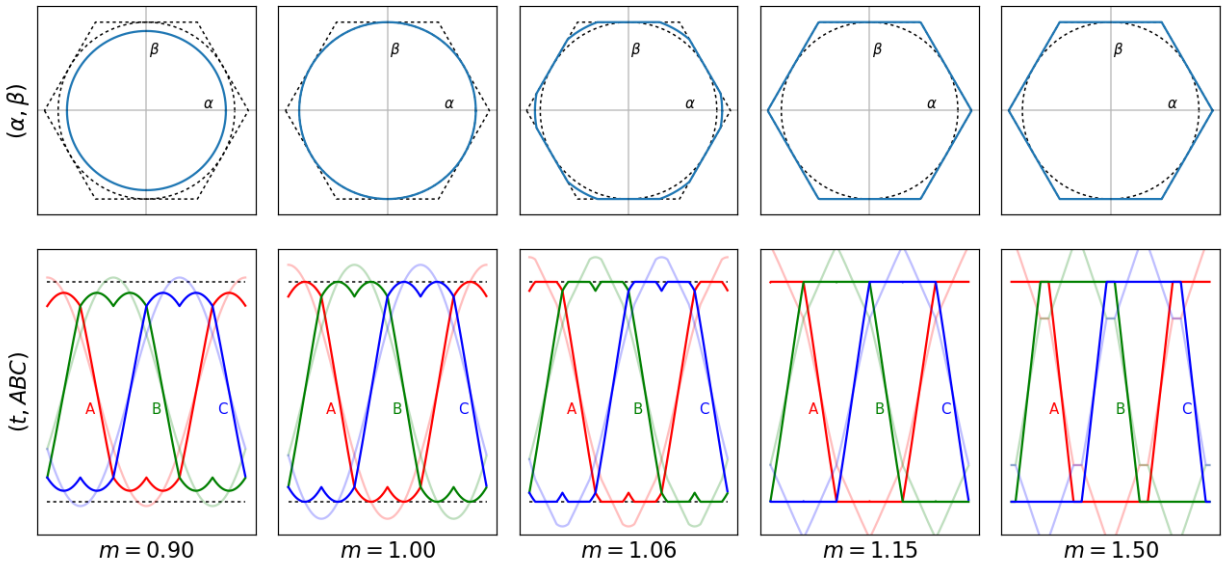
Premodulacija je definirana referentnim valnim oblikom čija amplituda prelazi amplitudu signaa nosioca, što uzrokuje „rezanje“ referentnog signala na vrhovima. Kao posljedica tog „rezanja“ izlazni valni oblik invertera sadrži više harmonike. Primjer običnog i „izrezanog“ referentnog signala prikazan je na slici ispod.



Slika 51. Usporedba običnog i premoduliranog referentnog signala

Unatoč deformaciji izlaznog valnog oblika, predomulacija omogućava povećanje izlaznog napona invertera. Uz povećanje izlaznog napona, povećava se i mogući raspon brzina priključenog motora, te relativno brža kontrola struje motora uslijed povećanog napona [21].

Slika 52. prikazuje valne oblike za različitie koeficijente modulacije. Gornji red prikazuje radijus (amplitudu) referentnog vektora u $\alpha - \beta$ sustavu. Iscrtkani šesterokut definira maksimalni mogući napon na izlazu iz invertera, a iscrtkana kružnica definira amplitudu referentnog napona sa modulacijskim koeficijentom 1. Donji red prikazuje referentne signale u ovisnosti o vremenu. Sa slike se može primjetiti da referentni signali počnu ulaziti u zasićenje u trenutku kada radijus (amplituda) referentnog vektora u bilo kojoj točki premaši iscrtkani šesterokut.



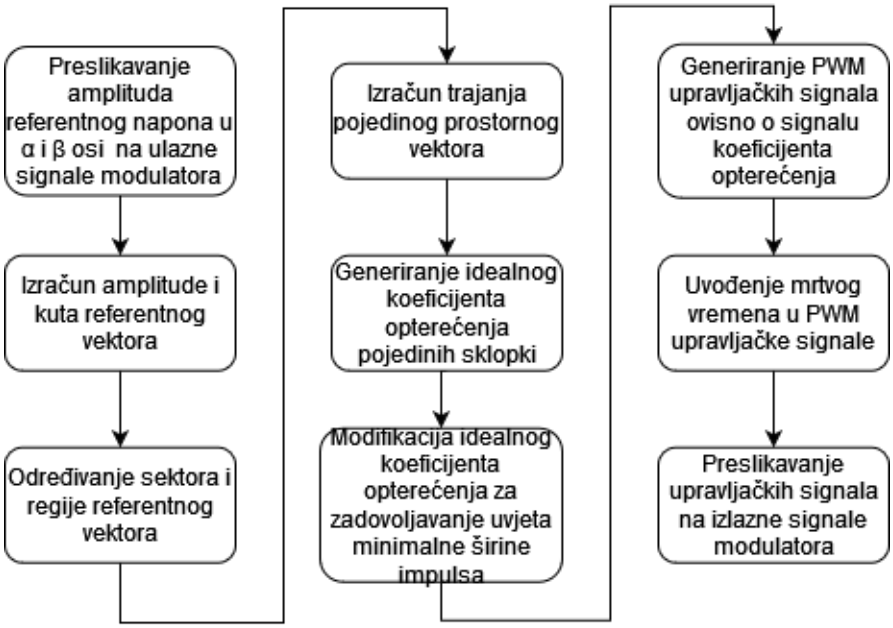
Slika 52. Primjer predmodulacije [21]

Premodulacija, unatoč povećanom izlaznom naponu, ima nekoliko neželjenih posljedica. Zbog distorzije izlaznog valnog oblika povećava se harmonička distorzija, odnosno *THD*. *THD* (engl. „*Total Harmonic Distortion*“) je mjera odstupanja valnog oblika od čistog sinusoidalnog valnog oblika uslijed postojanja viših harmonika. Prisutstvo viših harmonika također utječe na valovitost struje, tj. struja više oscilira oko srednje vrijednosti. Konačno, rad u području premodulacije može uzrokovati veće sklopne gubitke i podlagati komponente invertera većim termičkim opterećenjima [8].

8 Simulacija *PWM* prostornog vektora *3L-NPC* izmjenjivača

Kako je spomenuto u poglavlju 2., simulacije pretvarača i upravljačkih sustava odrađene su u programskom paketu *PLECS*. U ovom poglavlju opisati će se ponašanje pretvarača pri različitim pogonskim uvjetima (amplituda i frekvencija napona, te parametri trošila), te sa različitim parametrima upravljačkog sustava (vrijeme uzorkovanja, frekvencija sklapanja, itd.). Također, opisati će se kako funkcionira kod kojim se izvodi modulacija prostornog vektora.

Na slici ispod prikazan je dijagram toka programa korištenog u ovom radu.



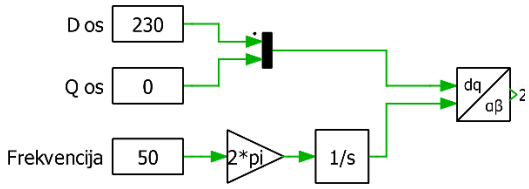
Slika 53. Dijagram toka programa implementiranog u ovom radu

8.1 Transformacija stacionarnog u rotirajući ortogonalni sustav

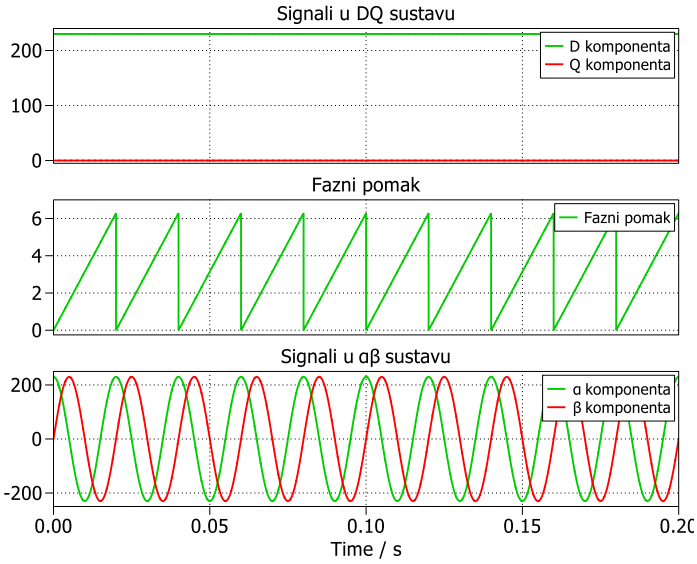
Sustav je dizajniran tako da se referentne vrijednosti zadaju u obliku d i q komponente napona u stacionarnom ortogonalnom sustavu. Za izračun rotirajućeg referentnog vektora napona potrebno je prebaciti referentni vektor iz stacionarnog u rotirajući ortogonalni sustav.

Ovaj postupak se izvodi uz pomoć amplituda d i q osi, te faznog pomaka ϕ. Fazni pomak dobije se integriranjem frekvencije u rasponu od 0 do 2π (frekvenciju u Hz je potrebno pomožiti konstatom 2π kako bi se dobila kružna frekvencija u rad/s). Transformacija između dva sustava odvija se pomoću bloka ugrađenog u PLECS, *RRF->SRF*, odnosno *„Rotating reference frame -> Static reference frame“*. Blok na ulazu prima 3 ulaza (d komponentu, q komponentu i fazni pomak ϕ), a na izlazu daje 2 vrijednosti (α i β komponentu).

Dio sustava koji izvodi ovu transformaciju prikazan je na slici ispod. Također su prikazani valni oblici ulaznih i izlaznih signala.



Slika 54. Dio sustava za transformaciju



Slika 55. Ulazni i izlazni signali trasnformacije

8.2 Generiranje referentnog rotirajućg vektora

Nakon što su refrentne vrijednosti napona prebačene iz stacionarnog u rotirajući ortogonalni sustav, potrebno je generirati referentni rotirajući vektor. Taj vektor se dobije vektorskim zbrojem komponenata napona u $\alpha - \beta$ sustavu. Referntni vektor je definiran magnitudom i faznim pomakom, a kod kojim se dobiju ove vrijednosti prikazan je na slici ispod.

```
46    uref = sqrt(ualpha*ualpha + ubeta*ubeta);
47    theta = fmod((atan2(ubeta, ualpha) + 2.0*PI), 2.0*PI);
```

Slika 56. Dio koda za izračun parametara referentnog rotirajućeg vektora

Ovi paramteri vektora omogućavaju nam izračune svih drugih vrijednosti potrebnih za određivanje koeficijenta opterećenja sklopki, poput regije i sektora referentnog vektora.

8.3 Određivanje regije i sektora referentnog vektora

Sa poznatim paramterima referentnog rotirajućeg vektora, moguće je izračunati ostale parametre koji su potrebni da se u konačnici izračunaju koeficijenti opterećenja sklopki.

Prvi parametar koji je potrebno izračunati je trenutni sektor u kojem se referentni vektor nalazi. Kako je kružnica podijeljena na 6 sektora, odnosno 6 dijelova od 60 stupnjeva, trenutni sektor nalazi se jednostavno preko kuta referntnog vektora. Kod kojim se definira trenutni sektor prikazan je na slici ispod.

```
35 void selectSector(int *sector, float theta_){
36
37     *sector = floor(theta/(PI/3) + 1) > 6? 1: floor(theta/(PI/3) + 1);
38
39 }
```

Slika 57. Kod za određivanje trenutnog sektora

Sljedeći parametar koji je potrebno izračunati je kut α, odnosno kut referentnog vektora u odnosu na trenutni sektor. Kut α računa se pomoću funkcije na slici ispod. Kut θ predstavlja kut referntnog vektora ograničen na raspon $[0, 2\pi]$.

```
57 void calcAlpha(float *alpha, float theta, float sector){
58
59     *alpha = theta - (sector-1) * PI/3;
60
61 }
```

Slika 58. Kod za izračun kuta α

Uz poznati kut α i magnitudu referntnog vektora moguće je izračunati trenutnu regiju u kojoj se referntni vektor nalazi. Regije su područja unutar svakog sektora koje dijele sektor na 4 jednakostranična trokuta. Definiranje regije u kojoj se referntni vektor nalazi je potrebno zbog određivanja prostornih vektora koji se koriste za aproksimaciju referentnog vektora. Na slici ispod prikazan je kod kojim se određuje trenutna regija u kojoj se referntni vektor nalazi.

```
41 void selectRegion(int *region, float uref_, float udc_, float alpha){
42
43     float K1 = uref_ * (sin(PI/3 - alpha)/sin(2*PI/3));
44     float K2 = uref_ * (sin(alpha)/sin(2*PI/3));
45
46     if(K1 < udc_/3 && K2 < udc_/3 && (K1+K2) < udc_/3){
47         *region = 1;
48     }else if(K1 < udc_/3 && K2 < udc_/3 && (K1+K2) > udc_/3){
49         *region = 2;
50     }else if(K1 > udc_/3){
51         *region = 4;
52     }else if(K2 > udc_/3){
53         *region = 3;
54     }
55 }
```

Slika 59. Kod za određivanje trenutne regije

8.4 Izračun idealnog signala koeficijenata opterećenja sklopki

Zadnja vrijednost koju je potrebno izračunati prije dobivanja vrijednosti koeficijenata opterećenja pojedinih sklopki je sveukupno vrijeme koliko koji prostorni vektor mora biti uključen za vrijeme jedne periode uzorkovanja.

Funkcija koja izvršava ovaj zadatak istovremeno određuje prostorne vektore koji se koriste za aproksimaciju referntnog vektora (pomoću informacije o trenutnoj regiji), i vraća vrijednosti vremena koliko određeni prostorni vektor mora biti uključen unutar jedne periode uzorkovanja. Kod koji se koristi za izračun ovih vrijednosti prikazan je na slici ispod. Ovi izrazi dobiveni su rješavanjem sustava jednadžbi iz izraza (*6.6*).

```
63 void calcTime(float Uref, float alpha, float Udc, int region, float Ts_, float *T1, float *T2, float *T3, int sector){
64    float A = Ts_*(Uref/Udc)*cos(alpha);
65    float B = Ts_*(Uref/Udc)*sin(alpha);
66    float C = Ts_;
67    if(region == 1){
68       *T1 = C - 3*A - sqrt(3)*B;
69       *T2 = 3*A - sqrt(3)*B;
70       *T3 = 2*sqrt(3)*B;
71    }else if(region == 2){
72       *T1 = -2*sqrt(3)*B + C;
73       *T2 = sqrt(3)*B + 3*A - C;
74       *T3 = C - 3*A + sqrt(3)*B;
75    }else if(region == 3){
76       *T1 = -3*A - sqrt(3)*B + 2*C;
77       *T2 = -sqrt(3) * (B - sqrt(3)*A);
78       *T3 = 2*sqrt(3)*B - C;
79    }else if(region == 4){
80       *T1 = -3*A + 2*C - sqrt(3)*B;
81       *T2 = -C + 3*A - sqrt(3)*B;
82       *T3 = 2*sqrt(3)*B;
83    }
84    *T1 = *T1/Ts_;
85    *T2 = *T2/Ts_;
86    *T3 = *T3/Ts_;
87 }
```

Slika 60. Kod za izračun trajanja prostornih vektora

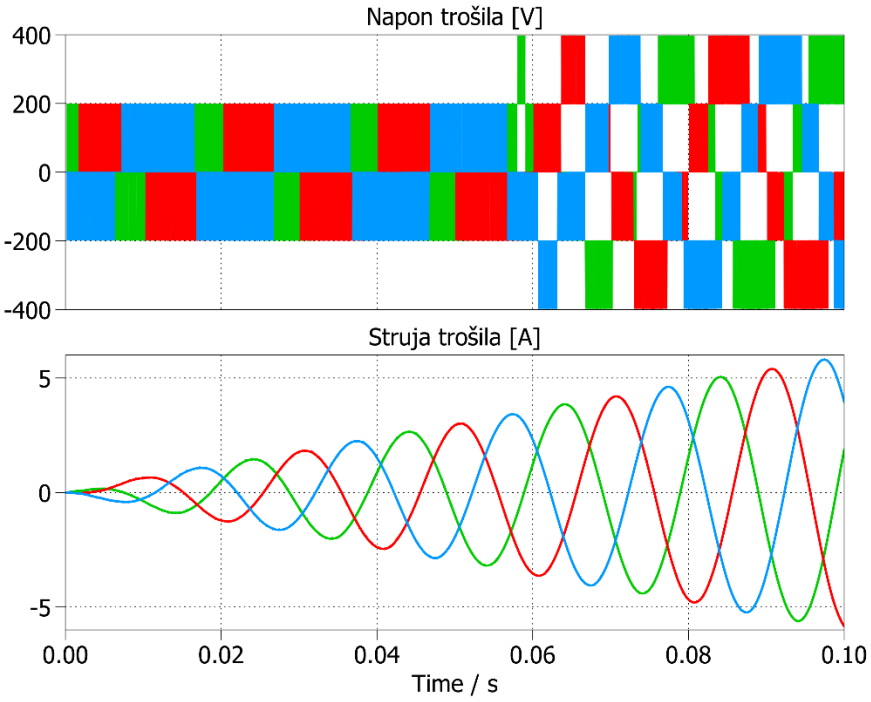
Prethodno opisana funkcija vraća vremensko trajanje pojedinih prostornih vektora, a zatim se te vrijednosti kombiniraju na način opisan u poglavlju 6.4. kako bi se dobili koeficijenti opterećenja pojedinih sklopki. Na slici ispod prikazan je kod za izračun koeficijenta opterećenja za sve regije unutar prvog sektora, a kod za ostale sektore funkcionira na istom principu.

```
54    //SEKTOR 1
55    if(sector == 1){
56       if(region == 1){
57          Sp[0] = Ta/4 + Tb/2 + Tc/2;
58          Sn[0] = Ta/4;
59          Sp[1] = Ta/4 + Tc/2;
60          Sn[1] = Ta/4 + Tb/2;
61          Sp[2] = Ta/4;
62          Sn[2] = Ta/4 + Tb/2 + Tc/2;
63       }else if(region == 2){
64          Sp[0] = Ta/2 + Tb + Tc/2;
65          Sn[0] = 0;
66          Sp[1] = Tc/2;
67          Sn[1] = Ta/2;
68          Sp[2] = 0;
69          Sn[2] = Ta/2 + Tb + Tc/2;
70       }else if(region == 3){
71          Sp[0] = Ta/2 + Tb + Tc;
72          Sn[0] = 0;
73          Sp[1] = Ta/2 + Tc;
74          Sn[1] = 0;
75          Sp[2] = 0;
76          Sn[2] = Ta/2 + Tb + Tc;
77       }else if(region == 4){
78          Sp[0] = Ta/2 + Tb + Tc;
79          Sn[0] = 0;
80          Sp[1] = 0;
81          Sn[1] = Ta/2 + Tb;
82          Sp[2] = 0;
83          Sn[2] = Ta/2 + Tb + Tc;
84       }
85    }
```

Slika 61. Kod za izračun koeficijenta opterećenja pojedinih sklopki

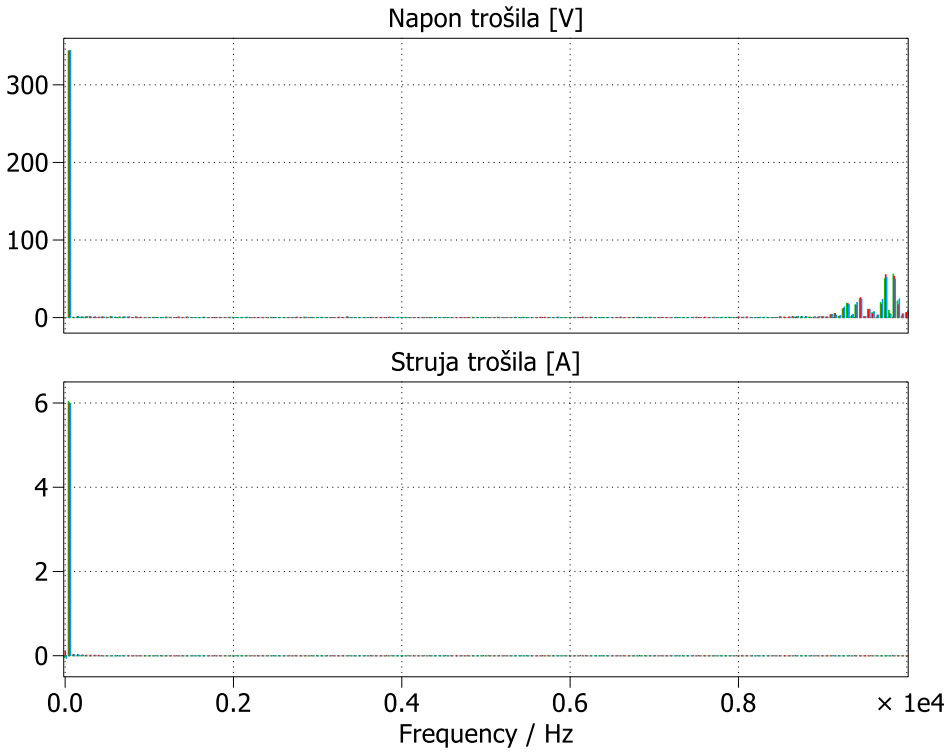
Trenutno je za svaku sklopku signal koeficijenta opterećenja pohranjen u dvije varijable $s_n[n]$ i $s_p[n]$, koje sadrže negativnu i pozitivnu poluperiodu idealnog signala koeficijenta opterećenja. U sljedećem poglavlju će se objasniti kako se te dvije varijable kombiniraju da bi se dobio konačan signal koeficijenta opterećenja.

Upravljački signali generirani pomoću ovakvih (idealnih) koeficijenata opterećenja ne mogu se primjenjivati u stvarnim sustavima jer bi u izmjenjivaču, među ostalim kvarovima, došlo do kratkog spoja. Ali, kako se u *PLECS*-u simulacija izvodi sa idealnim *IGBT* sklopkama moguće je simulirati ponašanje kruga bez pojave kratkog spoja. Takvi idealni valni oblici napona i struje, uz referentni napon koji raste po rampi do maksimalnog mogućeg, prikazani su na slikama ispod.



Slika 62. Valni oblici struje i napona trošila sa idealnim upravljačkim signalima

Rezultati frekvencijske analize valnih oblika napona i struje trošila prikazani su na slici ispod. Sa slike se mogu vidjeti očekivani rezultati, odnosno izraženi viši harmonici na frekvenciji f kod valnog oblika struje, a kod valnog oblika napona viši harmonici na frekvencijama f i $n \cdot f_s$.



Slika 63. frekvencijska analiza izlaznih valnih oblika uz $U_{ref} = 200V, f = 50Hz, f_s = 10kHz$

8.5 Modificiranje signala koeficijenta opterećenja

Kako je napomenuto, signal koeficijenta opterećenja generiran u prošlom poglavlju je idealan, odnosno ne zadovoljava uvjete mrtvog vremena i minimalnog vremena uključenja/isključenja.

Kako bi se zadovoljio uvjet minimalnog vremena uključenja, u kod je implementirana funkcionalnost opisana u poglavlju 7.2.1., a kod kojim je to izvedeno prikazan je na slici ispod.

```
2   counter1 = (counter1 + fs * SampleTimePeriod(1)) > 1 ?  0 : (counter1 + fs * SampleTimePeriod(1));
3   counter2 = (counter2 + fs * SampleTimePeriod(1)) > 0 ? -1 : (counter2 + fs * SampleTimePeriod(1));
4
5   for(int i = 0; i < 3; i++){
6      if(fabs(Sp[i] - Sn[i]) < minValue){
7         if((Sp[i] - Sn[i]) > 0){
8            temp[i] = minValue;
9         }else if((Sp[i] - Sn[i]) < 0){
10           temp[i] = -minValue;
11        }
12     }else if(fabs(Sp[i] - Sn[i]) > maxValue){
13        if((Sp[i] - Sn[i]) > 0){
14           temp[i] = maxValue;
15        }else if((Sp[i] - Sn[i]) < 0){
16           temp[i] = -maxValue;
17        }
18     }
19     else{
20        temp[i] = Sp[i] - Sn[i];
21     }
22
23     S[i][0] = temp[i] >= counter1;
24     S[i][2] = 1 - S[i][0];
25     S[i][1] = temp[i] >= counter2;
26     S[i][3] = 1 - S[i][1];
```

Slika 64. Kod za implementaciju uvjeta u minimalnog vremenu uključenja/isključenja

U suštini, ovaj kod idealan signal koeficijenta prebacuje unutar raspona $[minValue, maxValue]$ za pozitivnu poluperiodu, a $[-minValue, -maxValue]$ za negativnu poluperiodu. Vrijednosti *minValue* i *maxValue* izračunaju se u odnosu na signal nosioc prema izrazima na slici ispod.

```
1 minValue = Tmin/(tan(PI/2 - atan(fs)));
2 maxValue = 1 - Tmin/(1/fs);
```

Slika 65. Izrazi za izračun minValue i maxValue vrijednosti

U varijablama $S[i][j]$ su sada pohranjeni upravljački signali *j*-te sklopke u *i*-toj grani sa implementiranim uvjetom minimalne širine impulsa.

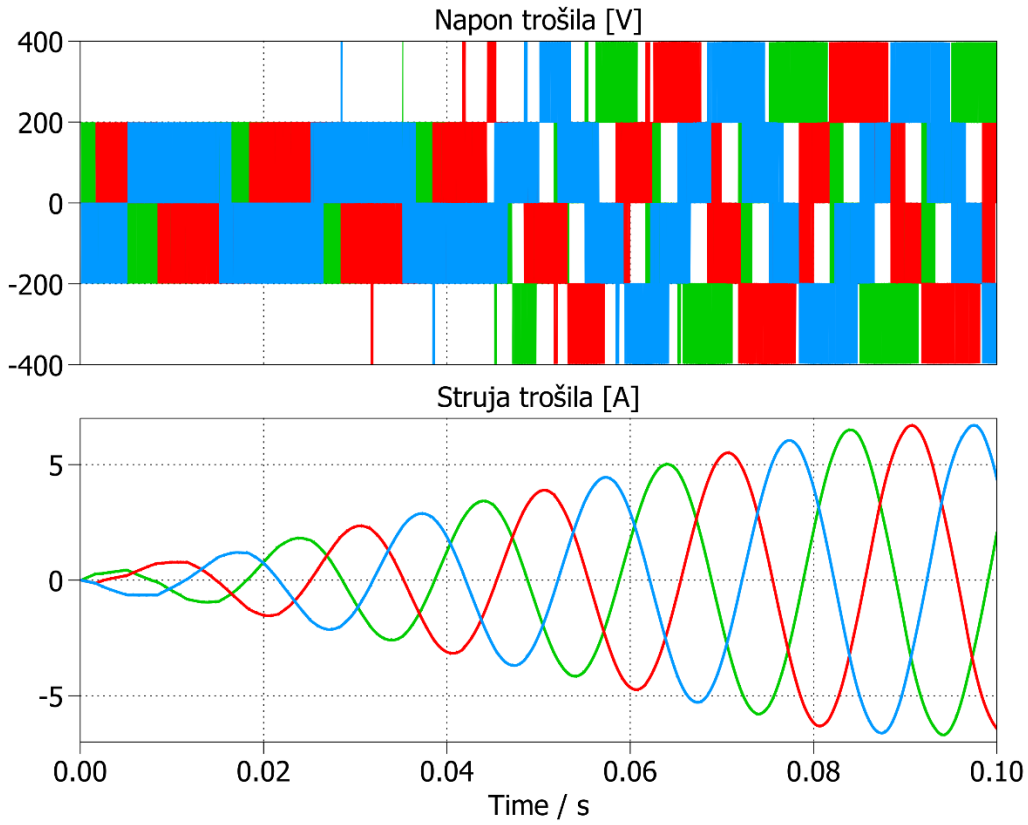
Zadnji korak je zadovoljavanje uvjeta mrtvog vremena, a navedeno se implementira na način opisan u poglavlju 7.1. Kod sa slike ispod modificira signal koeficijenta opterećenja na način da se uvede mrtvo vrijeme, odnosno kašnjenje od jednog programskog ciklusa određenog trajanja ako se primjeti situacija koja dovodi do oštećenja sklopa.

```
28        if(((S[i][0] - S_old[i][0]) > 0) && ((S[i][2] - S_old[i][2]) < 0)){
29           S[i][0] = S_old[i][0];
30        }else if(((S[i][0] - S_old[i][0]) < 0) && ((S[i][2] - S_old[i][2]) > 0)){
31           S[i][2] = S_old[i][2];
32        }
33
34        if(((S[i][3] - S_old[i][3]) > 0) && ((S[i][1] - S_old[i][1]) < 0)){
35           S[i][3] = S_old[i][3];
36        }else if(((S[i][3] - S_old[i][3]) < 0) && ((S[i][1] - S_old[i][1]) > 0)){
37           S[i][1] = S_old[i][1];
38        }
39        for(int j = 0; j < 4; j++){
40           S_old[i][j] = S[i][j];
41        }
42     }
```

Slika 66. Kod za implementaciju mrtvog vremena u upravljačke signale

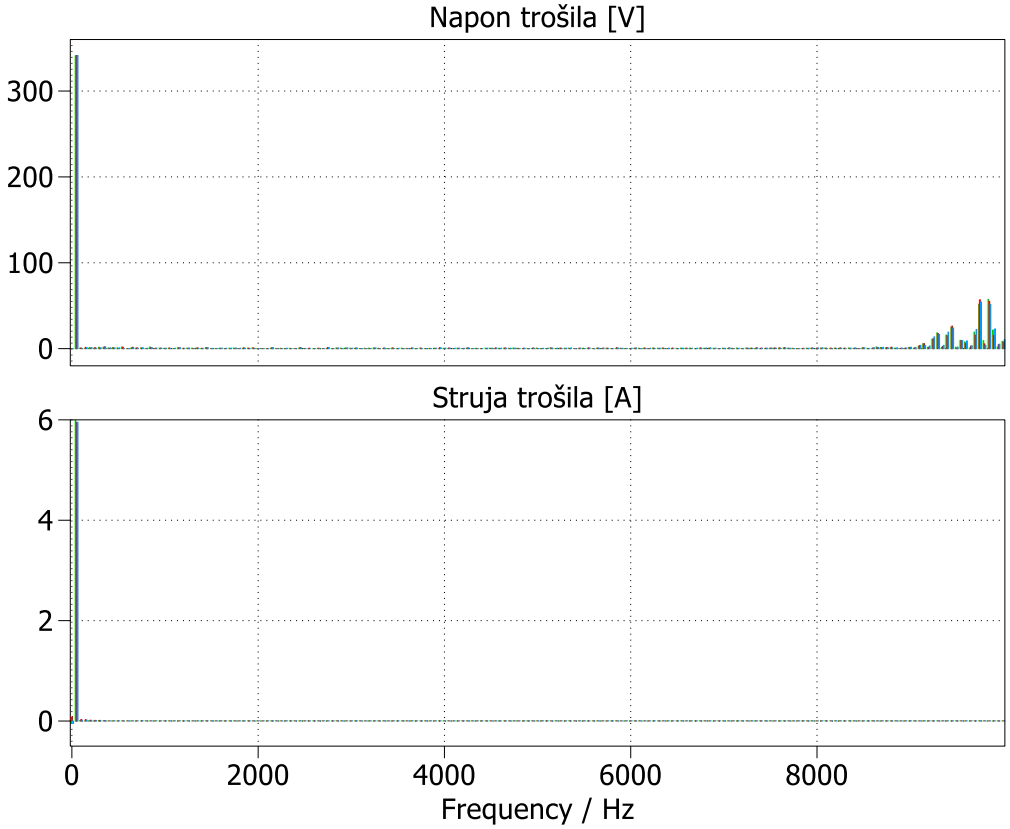
Upravljački signali sa uvedenim mrtvim vremenom šalju se na izlaz iz bloka *C-Script* te se koriste za upravljanje sklopkama *3L-NPC* izmjenjivača.

Na slici ispod prikazani su valni oblici napona i struje sve tri faze na izlazu iz izmjenjivača za referentni napon koji raste od $0V$ do maksimalnog mogućeg napona (amplituda velikog prostornog vektora $\approx U_{DC} \cdot \frac{2}{3} = 270V$), uz frekvenciju od $50\ Hz$.



Slika 67. Napon i struja trošila uz rastući referentni napon na ulazu

Frekvencijskom analizom izlaznih valnih oblika pri referentnom naponu od $U_{ref} = 200V$ vidi se da se viši harmonici značajnih amplituda javljaju tek oko višekratnika sklopne frekvencije ($n \cdot f_s = n \cdot 10kHz$).



Slika 68. frekvencijska analiza izlaznih valnih oblika uz $U_{ref} = 200V, f = 50Hz, f_s = 10kHz$

8.6 Numerička složenost algoritma

Numerička složenost je parametar određenog algoritma koji je definiran brojem artimetičkih i trigonometrijskih operacija, te brojem i tipom korištenih varijabli. Ovaj parametar bitan je pri implementaciji algoritma na realnom mikroprocesoru kako bi se osigurala pravilno funkcioniranje programa.

U ovom algoritmu korišteno je 18 varijabli od kojih je 11 tipa *float,* 2 tipa *int,* 3 tipa *float array* veličine 3, te 2 tipa *float array* veličine 3x4. Korišteno je 10 trigonometrijskih operacija, od kojih su 5 sinus, 1 kosinus, te 4 tangens funkcija.

Aritmetičkih operacija korišteno je 447, odnosno 140 operacija zbrajanja, 41 operacija oduzimanja, 73 operacija množenja i 193 operacije dijeljenja.

9 Zaključak

U ovom radu implementiran je algoritam za vektorsko upravljanje trorazinskim pretvaračem s pritegnutom nultom točkom (*3L-NPC*). Kod algoritma napisan je u *C* jeziku, a implementiran pomoću *C-Script* bloka u simulacijskom programu *PLECS.*

U poglavlju 8. prikazani su valni oblici koji se dobiju primjenom generiranih upravljačkih signala na izmjenjivač. Korištene su dvije vrste upravljačkih signala, idealni i modificrani. Idealni upravljački signali generirani su sa pretpostavkom da tranzistorske sklopke nemaju kašnjenje pri mijenjanju stanja, i da nemaju ograničenja o vremenu uključenosti/isključenosti Ovakve karakteristike se u stvarnim sustavima ne smiju pretpostaviti jer bi dovele do uništenja izmjenjivača ili uređaja koji se pogoni. Iz tog razloga je potrebno modificirati idealne upravljačke signale kako bi se dobili upravljački signali koji u realnom sustavom neće dovesti do kvara.

Iz frekvencijske analize valnih oblika vidljivo je da su više harmoničke komponente kod struje i napona zanemarivih veličina, odnosno jedina značajna harmonička komponenta je na referentnoj frekvenciji f. Kod valnog oblika izlaznog napona se, zbog *PWM* oblika signala, javljaju i više harmoničke komponente na frekvencijama $n \cdot f_s$, gdje je f_s sklopna frekvencija.

Simulacije izvedene u ovom radu pokazuju da razvijeni algoritam modulacije prostornog vektora za upravljanje *3L-NPC* izmjenjivača funkcionira bez greške, te uz to uspješno zadovoljava uvjete kritične za primjenu u realnim sustavima. Sljedeći korak prema primjeni ovog upravljačkog sustava u stvarnom sustavu bio bi implementacija algoritma mikroprocesoru koji podržava *C* kod, te prilagodba koda memoriji i brzini korištenog mikroprocesora.

Literatura

[1] Plexim GmBH, [Mrežno]. Available: https://www.plexim.com/products/plecs.

[2] Plexim GmBH, *PLECS User Manual.*

[3] R. Teja, »Electronics Hub,« [Mrežno]. Available: https://www.electronicshub.org/characteristics-and-working-of-half-wave-rectifier/.

[4] M. Ikonen, O. Laakkonen i M. Kettunen, *Two-Level and Three-Level Converter Comparison in Wind Power Application,* 2005.

[5] Z. &. H. M. &. H. M. &. K. Mahmoud, »Space vector modulation of multilevel inverters: a simple and fast method of two-level hexagon's selection,« *International Journal of Power Electronics,* p. 107, 2017.

[6] T. E. Michael Frisch, »Advantages of NPC Inverter Topologies with Power,« Vincotech , 2009.

[7] J. Dodge, »3L-ANPC vs. 3L-NPC Inverters,« UnitedSic, 2021.

[8] I. Staudt, »3L NPC & TNPC Topology,« SEMIKRON, 2015.

[9] M. Bolić, »Electronics,« u *Pervasive Cardiovascular and Respiratory Monitoring Devices,* Academic Press, 2023, pp. 73-123.

[10] K. Z. I. A. M. K. Eram Bushra, »A comprehensive review on recent trends and future prospects of PWM techniques for harmonic suppression in renewable energies based power converters,« *Results in Engineering,* svez. 22, 2024.

[11] G. D. Holmes, Pulse Width Modulation For Power Converters, Wiley Press, 2003.

[12] A. Namboodiri i S. H. Wani, »Unipolar and Bipolar PWM Inverter,« *IJIRST,* svez. 1, br. 7, 2014.

[13] G. Pfaff, A. Weschta i F. Wick, »Design and Experimental Results of a Brushless AC Servo,« *IEEE Transactions On Industry Applications,* Sves. %1 od %21A-20, pp. 814-821, 1984.

[14] M. Madhuchhanda, S. Samarjit i S. Chattopadhyay, Electric Power Quality, Springer Press, 2011.

[15] J. Kujala, *Discontinuous PWM techniques in three-phase power converters,* 2020.

[16] S. Mekhilef, I. Hudhaifa i H. Belkamel, *DC Link Capacitor Voltage Balancing in Three Level Neutral Point Clamped Inverter,* 2012.

[17] G. Fernandez i F. Ledent, »imperix,« [Mrežno]. Available: https://imperix.com/doc/implementation/balancing-of-npc-converters.

[18] L. Hao, Z. Meng, Y. Hao i Y. Xingwu, *Nanoseconds Switching Time Monitoring of Insulated Gate Bipolar Transistor Module by Under-Sampling Reconstruction of High-Speed Switching Transitions Signal,* Shanghai, China: College of Electrical Power Engineering, Shanghai University of Electric Power, 2019.

[19] »IGBT MOS,« [Mrežno]. Available: https://www.igbtmos.com/blog/interpretation-of-igbt-narrow-pulse-phenomenon.

[20] »Analisi, Simulazione Ed Implementazione Di Un Modulatore Per Invertitore NPC A Tensione Impressa A Tre Livelli,« Universita Degli Studi Di Trieste Facolta A Ingeneria, Trieste, 2007..

[21] Microchip, »MICROCHIP,« [Mrežno]. Available: https://developerhelp.microchip.com/xwiki/bin/view/applications/motors/control-algorithms/3-phase-foc/.

[22] G. Holmes i T. Lipo, Pulse Width Modulation For Power Converters - Principles and Practice, IEEE Press, 2003.

[23] G. Wang, G. Zhang i D. Xu, Position Sensorless Control Techniques for Permanent Magnet Synchronous Machine Drives, Springer, 2020.

Sažetak

U ovom radu obrađen je razvoj algoritma za upravljanje trorazinskim pretvaračem sa pritegnutom nultom točkom (*3L-NPC*) pomoću pulsno širinske modulacije prostornog vektora. Rad započinje introdukcijom *PLECS*-a, simulacijskog programa u kojem je modeliran pretvarač. Sljedeće poglavlje bavi se energetskim pretvaračima, te se uvodi pojam izmjenjivač, odnosno *DC/AC* pretvarač. Objašnjene su neke jednostavnije vrste ispravljača i izmjenjivača, a zatim se rad fokusira na trorazinski izmjenjivač sa pritegnutom nultom točkom. Definira se funkcionalnost izmjenjivača, te načini njegovog upravljanja.

Do sada se u radu objašnjavo energetski dio sustava (izmjenjivač), a u sljedećim poglavljima se pažnja prebacuje na upravljački dio sustava. Opisuje se što je puslno širinska modulacija, različite metode pulsno širinske modulacije, te ideja prostornog vektora kao upravljane veličine. U radu se prvo opisuje modulacija prostornog vektora na primjeru dvorazinskog izmjenjivača, te se zatim ta ideja proširuje na modulaciju prostornog vektora za upravljanje trorazinskog pretvarača. Ovakvo upravljanje izmjenjivačima funkcionira samo u idealnim uvjetima (u simulacijskom programu), te nije primjenjivo u stvarnom svijetu. Iz tog razloga se u sljedećem poglavlju objašnjavaju situacije koje u realnim sustavima mogu dovesti do oštećenja sklopa, te se isto tako objašnjavaju načini za rješavanje istih problema. Također se opisuje kako su rješenja tih problema implementirana u algoritmu.

Zadnje poglavlje rada fokusira se na sažeto opisivanje cijelog programa, odnosno algoritma, te se objašnjava postepeno što se događa u kojem dijelu programa. Na kraju rada navedena je numerička složenost algoritma koja je bitan parametar pri implementaciji istog u realnom mikrokontroleru.

Ključne riječi: izmjenjivač, prostorni vektor, *PWM*

Summary

This paper deals with the development of an algorithm for controlling a three-level converter with neutral point clamped (3L-NPC) using pulse-width modulation of a spatial vector. The paper begins with the introduction of PLECS, a simulation program in which the converter is modeled. The next chapter deals with energy converters, and the term inverter, or DC/AC converter, is introduced. Some simpler types of rectifiers and inverters are explained, and then the paper focuses on the three-level neutral point clamped inverter. The functionality of the inverter and its control methods are defined.

So far, the paper has explained the high voltage part of the system (inverter), and in the following chapters the attention is shifted to the control part of the system. It describes what pulse-width modulation is, different methods of pulse-width modulation, and the idea of a space vector as a controlled quantity. The paper first describes the modulation of the spatial vector on the example of a two-level inverter, and then this idea is extended to the modulation of the space vector for the control of a three-level inverter. This kind of control of inverter works only in ideal conditions (e.g. in a simulation software), and is not applicable in the real world. For this reason, the following chapter explains problemss that can lead to circuit damage in real systems, and also explains ways to solve the same problems. It also describes how the solutions to these problems are implemented in the algorithm.

The last chapter of the paper focuses on a concise description of the entire program, i.e. the algorithm, and gradually explains what happens in which part of the program. At the end of the paper, the numerical complexity of the algorithm is stated, which is an important parameter when implementing it in a real microcontroller.

Key words: inverter, space vector, PWM

10 Popis slika

11 Dodaci

11.1 Kodovi za dvorazinski *SVPWM*

Kod 1. Code Declarations

```
// Definiranje korištenih knjižnica
#include <math.h>

//Definiranje ulaznih i izlaznih signala
#define u_a InputSignal(0, 0)
#define u_b InputSignal(0, 1)
#define u_c InputSignal(0, 2)
#define u_dc InputSignal(1, 0)

// Definiranje konstanta (unutarnjih i vanjskih)
#define Ts ParamRealData(0,0)
#define PI  3.14159265359

// Definiranje korištenih varijabli
float u_ref, theta, alpha, u_d, u_q;
float Ta, Tb, T0, m;
float S1, S3, S5;
float T1, T2, T3, T4;
float out1, out2, out3;
float trig, trigA, trigF = 0;
int n;
```

Kod 2. Start Function Code

```
trigF = 10e3;
trigA = 1;
```

Kod 3. Output Function Code

```
OutputSignal(0, 0) = S1>trig;
OutputSignal(0, 1) = S3>trig;
OutputSignal(0, 2) = S5>trig;
```

Kod 4. Update Function Code

```
u_d = 1.0/3.0 * (2*u_a - u_b - u_c);
u_q = 1/sqrt(3) * (u_b - u_c);

u_ref = sqrt(u_d*u_d + u_q*u_q);
theta = fmod((atan2(u_q, u_d) + 2*PI), 2*PI);

alpha = theta - (n-1) * PI/3;
alpha = (alpha >= PI/3) ? PI/3 : alpha;
alpha = (alpha <= 0) ? 0 : alpha;

n = floor(theta/(PI/3) + 1);
if(n > 6){
```

```
      n = 1;
}

Ta = u_ref/u_dc * sin(PI/3 - alpha)/sin(PI/3);
Tb = u_ref/u_dc * sin(alpha)/sin(PI/3);
T0 = Ts - Ta - Tb;

T1 = Ta + Tb + T0/2;
T2 = Ta + T0/2;
T3 = Tb + T0/2;
T4 = T0/2;

if(n >= 0 && n <= 1){
             S1 = T1;
             S3 = T3;
             S5 = T4;
}if(n > 1 && n <= 2){
             S1 = T2;
             S3 = T1;
             S5 = T4;
}if(n > 2 && n <= 3){
             S1 = T4;
             S3 = T1;
             S5 = T3;
}if(n > 3 && n <= 4){
             S1 = T4;
             S3 = T2;
             S5 = T1;
}if(n > 4 && n <= 5){
             S1 = T3;
             S3 = T4;
             S5 = T1;
}if(n > 5 && n <= 6){
             S1 = T1;
             S3 = T4;
             S5 = T2;
}

trig = trigA * trigF * fabs(fmod(CurrentTime, 1/trigF)) - trigA/2;
```

11.2 Kodovi za trorazinski *SVPWM*

Kod 5. Code Declarations

```
// Definiranje korištenih knjižnica
#include <math.h>

// Definiranje ulaza
#define ualpha      InputSignal(0, 0)
#define ubeta       InputSignal(0, 1)
#define udc  InputSignal(1, 0)

// Definiranje konstanti (ulaznih i izlaznih)
#define Ts          ParamRealData(0, 0)
#define fs          ParamRealData(1, 0)
#define Tmin ParamRealData(2, 0)

#define PI 3.14159265359

// Definiranje korištenih varijabli
float uref, theta, alpha = 0;
int sector, region = 1;
float Ta, Tb, Tc = 0;
float phase = 0;
float counter1 = 0;
float counter2 = -1;
float Sp[3] = {0, 0, 0};
float Sn[3] = {0, 0, 0};
float S[3][4] = {{0, 0, 0, 0},{0, 0, 0, 0},{0, 0, 0, 0}};;
float S_old[3][4] = {{0, 0, 0, 0},{0, 0, 0, 0},{0, 0, 0, 0}};

float minValue = 0;
float maxValue = 0;
float temp[3] = {0, 0, 0};


// Definiranje korištenih funkcija

void selectSector(int *sector, float theta_){
      // funkcija za izračun sektora u kojem se nalazi referentni vektor
      *sector = floor(theta/(PI/3) + 1) > 6? 1: floor(theta/(PI/3) + 1);

}

void selectRegion(int *region, float uref_, float udc_, float alpha, int sector,
float theta){
      //funkcija za izračun u kojoj se regiji unutar sektora nalazi referentni
      vektor
       //regija se računa prema dvije komponente referentnog vektora (pod 0 i 60°)

      float K1 = uref_ * (sin(PI/3 - alpha)/sin(2*PI/3));
      float K2 = uref_ * (sin(alpha)/sin(2*PI/3));

      if(K1 < udc_/3 && K2 < udc_/3 && (K1+K2) < udc_/3){
            *region = 1;
      }else if(K1 < udc_/3 && K2 < udc_/3 && (K1+K2) > udc_/3){
            *region = 2;
```

```
	}else if(K1 > udc_/3){
		*region = 4;
	}else if(K2 > udc_/3){
		*region = 3;
	}
}

void calcAlpha(float *alpha, float theta, float sector){
	//funkcija za izračun kuta referentnog vektora unutar pojedinog sektora
	//kut alpha ograničen je na vrijednost od 0 do 60°
	*alpha = theta - (sector-1) * PI/3;

}

void calcTime(float Uref, float alpha, float Udc, int region, float Ts_, float *T1,
float *T2, float *T3, int sector){
	//funkcija za izračun vremena pojedinog prostornog vektora unutar periode
	//uzorkovanja

	//za svaku regiju se vrijeme trajanja vektora izračunava na drugačiji način
	float A = Ts_*(Uref/Udc)*cos(alpha);
	float B = Ts_*(Uref/Udc)*sin(alpha);
	float C = Ts_;
	if(region == 1){
		*T1 = C - 3*A - sqrt(3)*B;
		*T2 = 3*A - sqrt(3)*B;
		*T3 = 2*sqrt(3)*B;
	}else if(region == 2){
		*T1 = -2*sqrt(3)*B + C;
		*T2 = sqrt(3)*B + 3*A - C;
		*T3 = C - 3*A + sqrt(3)*B;
	}else if(region == 3){
		*T1 = -3*A - sqrt(3)*B + 2*C;
		*T2 = -sqrt(3) * (B - sqrt(3)*A);
		*T3 = 2*sqrt(3)*B - C;
	}else if(region == 4){
		*T1 = -3*A + 2*C - sqrt(3)*B;
		*T2 = -C + 3*A - sqrt(3)*B;
		*T3 = 2*sqrt(3)*B;
	}
	*T1 = *T1/Ts_;
	*T2 = *T2/Ts_;
	*T3 = *T3/Ts_;
}
```

Kod 6. Start Funcion Code

```
//Izračun minimalne i maksimalne vrijednosti signala koeficijenta opterećenja kako bi
//bio zadovoljen uvjet minimalne širine impulsa
minValue = Tmin/(tan(PI/2 - atan(fs)));
maxValue = 1 - Tmin/(1/fs);
```

Kod 7. Output Function Code

```
OutputSignal(0, 0) = S[0][0];
OutputSignal(0, 1) = S[0][1];
OutputSignal(0, 2) = S[0][2];
OutputSignal(0, 3) = S[0][3];

OutputSignal(1, 0) = S[1][0];
OutputSignal(1, 1) = S[1][1];
OutputSignal(1, 2) = S[1][2];
OutputSignal(1, 3) = S[1][3];

OutputSignal(2, 0) = S[2][0];
OutputSignal(2, 1) = S[2][1];
OutputSignal(2, 2) = S[2][2];
OutputSignal(2, 3) = S[2][3];

OutputSignal(3, 0) = IsSampleHit(2);
```

Kod 8. Update Function Code

```
if(IsSampleHit(1)){
	//generiranje dva pilasta signala nosioca od 0->1 i od -1->0
	counter1 = (counter1 + fs * SampleTimePeriod(1)) > 1 ?  0 : (counter1 + fs *
SampleTimePeriod(1));
	counter2 = (counter2 + fs * SampleTimePeriod(1)) > 0 ? -1 : (counter2 + fs *
SampleTimePeriod(1));

	//U sljedećoj for petlji se ograničava vrijednost signala koeficijenta
	//opterećenja unutar raspona [minValue, maxValue] kako bi se zadovoljio uvjet
	//minimalne širine impulsa
	for(int i = 0; i < 3; i++){
		if(fabs(Sp[i] - Sn[i]) < minValue){
			if((Sp[i] - Sn[i]) > 0){
				temp[i] = minValue;
			}else if((Sp[i] - Sn[i]) < 0){
				temp[i] = -minValue;
			}
		}else if(fabs(Sp[i] - Sn[i]) > maxValue){
			if((Sp[i] - Sn[i]) > 0){
				temp[i] = maxValue;
			}else if((Sp[i] - Sn[i]) < 0){
				temp[i] = -maxValue;
			}
		}
		else{
			temp[i] = Sp[i] - Sn[i];
		}

		//Usporedba signala koeficijenta opterećenja sa signalom nosiocem kako
		//bi se generirali PWM upravljački signali
		S[i][0] = temp[i] >= counter1;
		S[i][2] = 1 - S[i][0];
		S[i][1] = temp[i] >= counter2;
		S[i][3] = 1 - S[i][1];
```

```
			//Ostatak ovog dijela koda služi za uvođenje mrtvog vremena između 1. i
			//3., te 2. i 4. sklopke u svakoj grani
			if(((S[i][0] - S_old[i][0]) > 0) && ((S[i][2] - S_old[i][2]) < 0)){
				S[i][0] = S_old[i][0];
			}else if(((S[i][0] - S_old[i][0]) < 0) && ((S[i][2] - S_old[i][2]) >
0)){
				S[i][2] = S_old[i][2];
			}

			if(((S[i][3] - S_old[i][3]) > 0) && ((S[i][1] - S_old[i][1]) < 0)){
				S[i][3] = S_old[i][3];
			}else if(((S[i][3] - S_old[i][3]) < 0) && ((S[i][1] - S_old[i][1]) >
0)){
				S[i][1] = S_old[i][1];
			}
			for(int j = 0; j < 4; j++){
				S_old[i][j] = S[i][j];
			}
		}
}

if(IsSampleHit(0)){
		//Izračun amplitude i kuta referentnog vektora
		uref = sqrt(ualpha*ualpha + ubeta*ubeta);
		theta = fmod((atan2(ubeta, ualpha) + 2.0*PI), 2.0*PI);

		//U sljedeće 4 linije se računaju potrebni parametri za izračun vremena
		trajanja pojedinog vektora uz pomoć funkcija definiranih u Code Declarations
		selectSector(&sector, theta);
		calcAlpha(&alpha, theta, sector);
		selectRegion(&region, uref, udc, alpha, sector, theta);
		calcTime(uref, alpha, udc, region, Ts, &Ta, &Tb, &Tc, sector);

		//U sljedećem dijelu koda se ovisno o trenutnom sektoru i regiji referentnog
		//vektora računa signal koeficijenta opterećenja svake grane invertera
		//Sp[n] i Sn[n] predstavljaju pozitivnu i negativnu poluperiodu signala
		//koeficijenta opterećenja n-te grane invertera

		//SEKTOR 1
		if(sector == 1){
			if(region == 1){
				Sp[0] = Ta/4 + Tb/2 + Tc/2;
				Sn[0] = Ta/4;
				Sp[1] = Ta/4 + Tc/2;
				Sn[1] = Ta/4 + Tb/2;
				Sp[2] = Ta/4;
				Sn[2] = Ta/4 + Tb/2 + Tc/2;
			}else if(region == 2){
				Sp[0] = Ta/2 + Tb + Tc/2;
				Sn[0] = 0;
				Sp[1] = Tc/2;
				Sn[1] = Ta/2;
				Sp[2] = 0;
				Sn[2] = Ta/2 + Tb + Tc/2;
			}else if(region == 3){
				Sp[0] = Ta/2 + Tb + Tc;
				Sn[0] = 0;
```

```
                Sp[1] = Ta/2 + Tc;
                Sn[1] = 0;
                Sp[2] = 0;
                Sn[2] = Ta/2 + Tb + Tc;
        }else if(region == 4){
                Sp[0] = Ta/2 + Tb + Tc;
                Sn[0] = 0;
                Sp[1] = 0;
                Sn[1] = Ta/2 + Tb;
                Sp[2] = 0;
                Sn[2] = Ta/2 + Tb + Tc;
        }
}
//SEKTOR 2
else if(sector == 2){
        if(region == 1){
                Sp[0] = Ta/4 + Tb/2;
                Sn[0] = Ta/4 + Tc/2;
                Sp[1] = Ta/4 + Tb/2 + Tc/2;
                Sn[1] = Ta/4;
                Sp[2] = Ta/4;
                Sn[2] = Ta/4 + Tb/2 + Tc/2;
        }else if(region == 2){
                Sp[0] = Ta/2;
                Sn[0] = Tc/2;
                Sp[1] = Ta/2 + Tb + Tc/2;
                Sn[1] = 0;
                Sp[2] = 0;
                Sn[2] = Ta/2 + Tb + Tc/2;
        }else if(region == 3){
                Sp[0] = 0;
                Sn[0] = Ta/2 + Tc;
                Sp[1] = Ta/2 + Tb + Tc;
                Sn[1] = 0;
                Sp[2] = 0;
                Sn[2] = Ta/2 + Tb + Tc;
        }else if(region == 4){
                Sp[0] = Ta/2 + Tb;
                Sn[0] = 0;
                Sp[1] = Ta/2 + Tb + Tc;
                Sn[1] = 0;
                Sp[2] = 0;
                Sn[2] = Ta/2 + Tb + Tc;
        }
}
//SEKTOR 3
else if(sector == 3){
        if(region == 1){
                Sp[0] = Ta/4;
                Sn[0] = Ta/4 + Tb/2 + Tc/2;
                Sp[1] = Ta/4 + Tb/2 + Tc/2;
                Sn[1] = Ta/4;
                Sp[2] = Ta/4 + Tc/2;
                Sn[2] = Ta/4 + Tb/2;
        }else if(region == 2){
                Sp[0] = 0;
                Sn[0] = Ta/2 + Tb + Tc/2;
                Sp[1] = Ta/2 + Tb + Tc/2;
```

```
                Sn[1] = 0;
                Sp[2] = Tc/2;
                Sn[2] = Ta/2;
        }else if(region == 3){
                Sp[0] = 0;
                Sn[0] = Ta/2 + Tb + Tc;
                Sp[1] = Ta/2 + Tb + Tc;
                Sn[1] = 0;
                Sp[2] = Ta/2 + Tc;
                Sn[2] = 0;
        }else if(region == 4){
                Sp[0] = 0;
                Sn[0] = Ta/2 + Tb + Tc;
                Sp[1] = Ta/2 + Tb + Tc;
                Sn[1] = 0;
                Sp[2] = 0;
                Sn[2] = Ta/2 + Tb;
        }
}
//SEKTOR 4
else if(sector == 4){
        if(region == 1){
                Sp[0] = Ta/4;
                Sn[0] = Ta/4 + Tb/2 + Tc/2;
                Sp[1] = Ta/4 + Tb/2;
                Sn[1] = Ta/4 + Tc/2;
                Sp[2] = Ta/4 + Tb/2 + Tc/2;
                Sn[2] = Ta/4;
        }else if(region == 2){
                Sp[0] = 0;
                Sn[0] = Ta/2 + Tb + Tc/2;
                Sp[1] = Ta/2;
                Sn[1] = Tc/2;
                Sp[2] = Ta/2 + Tb + Tc/2;
                Sn[2] = 0;
        }else if(region == 3){
                Sp[0] = 0;
                Sn[0] = Ta/2 + Tb + Tc;
                Sp[1] = 0;
                Sn[1] = Ta/2 + Tc;
                Sp[2] = Ta/2 + Tb + Tc;
                Sn[2] = 0;
        }else if(region == 4){
                Sp[0] = 0;
                Sn[0] = Ta/2 + Tb + Tc;
                Sp[1] = Ta/2 + Tb;
                Sn[1] = 0;
                Sp[2] = Ta/2 + Tb + Tc;
                Sn[2] = 0;
        }
}
//SEKTOR 5
else if(sector == 5){
        if(region == 1){
                Sp[0] = Ta/4 + Tc/2;
                Sn[0] = Ta/4 + Tb/2;
                Sp[1] = Ta/4;
                Sn[1] = Ta/4 + Tb/2 + Tc/2;
```

```
                Sp[2] = Ta/4 + Tb/2 + Tc/2;
                Sn[2] = Ta/4;
            }else if(region == 2){
                Sp[0] = Tc/2;
                Sn[0] = Ta/2;
                Sp[1] = 0;
                Sn[1] = Ta/2 + Tb + Tc/2;
                Sp[2] = Ta/2 + Tb + Tc/2;
                Sn[2] = 0;
            }else if(region == 3){
                Sp[0] = Ta/2 + Tc;
                Sn[0] = 0;
                Sp[1] = 0;
                Sn[1] = Ta/2 + Tb + Tc;
                Sp[2] = Ta/2 + Tb + Tc;
                Sn[2] = 0;
            }else if(region == 4){
                Sp[0] = 0;
                Sn[0] = Ta/2 + Tb;
                Sp[1] = 0;
                Sn[1] = Ta/2 + Tb + Tc;
                Sp[2] = Ta/2 + Tb + Tc;
                Sn[2] = 0;
            }
        }
        //SEKTOR 6
        else if(sector == 6){
            if(region == 1){
                Sp[0] = Ta/4 + Tb/2 + Tc/2;
                Sn[0] = Ta/4;
                Sp[1] = Ta/4;
                Sn[1] = Ta/4 + Tb/2 + Tc/2;
                Sp[2] = Ta/4 + Tb/2;
                Sn[2] = Ta/4 + Tc/2;
            }else if(region == 2){
                Sp[0] = Ta/2 + Tb + Tc/2;
                Sn[0] = 0;
                Sp[1] = 0;
                Sn[1] = Ta/2 + Tb + Tc/2;
                Sp[2] = Ta/2;
                Sn[2] = Tc/2;
            }else if(region == 3){
                Sp[0] = Ta/2 + Tb + Tc;
                Sn[0] = 0;
                Sp[1] = 0;
                Sn[1] = Ta/2 + Tb + Tc;
                Sp[2] = 0;
                Sn[2] = Ta/2 + Tc;
            }else if(region == 4){
                Sp[0] = Ta/2 + Tb + Tc;
                Sn[0] = 0;
                Sp[1] = 0;
                Sn[1] = Ta/2 + Tb + Tc;
                Sp[2] = Ta/2 + Tb;
                Sn[2] = 0;
            }
        }
}
```