

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV VÝKONOVÉ ELEKTROTECHNIKY A ELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

DEPARTMENT OF POWER ELECTRICAL AND ELECTRONIC ENGINEERING

ALGORITMY GENEROVÁNÍ TROJFÁZOVÉ SINUSOVÉ PULSNÍ ŠÍŘKOVÉ MODULACE

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

Ladislav Rudolf

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**

**ÚSTAV VÝKONOVÉ ELEKTROTECHNIKY
A ELEKTRONIKY**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF POWER ELECTRICAL AND ELECTRONIC
ENGINEERING

ALGORITMY GENEROVÁNÍ TROJFÁZOVÉ SINUSOVÉ PULSNÍ ŠÍŘKOVÉ MODULACE

TREE-PHASE SINUSOIDAL PULSE WIDTH MODULATION ALGORITHMS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

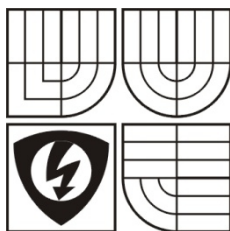
Ladislav Rudolf

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Bohumil Klíma, PhD.

BRNO, 2009



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav výkonové elektrotechniky a elektroniky

Bakalářská práce

bakalářský studijní obor
Silnoproudá elektrotechnika a výkonová elektronika

Student: Ladislav Rudolf

Ročník: 3

ID: 16516

Akademický rok: 2008/09

NÁZEV TÉMATU:

Algoritmy generování trojfázové sinusové pulzní šířkové modulace

POKYNY PRO VYPRACOVÁNÍ:

1. Popište algoritmy trojfázové sinusové PWM
2. Zvolený algoritmus naprogramujte do mikroprocesoru rady 56800E
3. Algoritmy otestujte a zobrazte průběhy výstupních napětí střídače a porovnejte vlastnosti uvedených algoritmu z hlediska nároku na mikroprocesor

DOPORUČENÁ LITERATURA:

Termín zadání: 1.10.2008

Termín odevzdání: 29.05.2009

Vedoucí projektu: Ing. Bohumil Klíma, PhD.

doc. Ing. Čestmír Ondrůšek, CSc.
předseda oborové rady

UPOZORNĚNÍ:

Autor semestrální práce nesmí při vytváření semestrální práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

Abstrakt

Cílem této bakalářské práce je uvést čtenáře do problematiky algoritmu generování trojfázové sinusové pulsní šířkové modulace. Hlavní pozornost je věnována metodám generování pulsně šířkové modulace, jako je komparační metoda PWM a modulační technika prostorové modulace vektoru napětí SVM. Jsou rozebrány jednotlivé funkční celky algoritmů a popsány jejich návrhy. V neposlední řadě tento projekt popisuje základní analogové řešení a i implementaci PWM modulátoru v signálovém procesoru DSP56F800.

Abstract

The main theme of this bachelors's thesis is to introduce readers into the algorithm of generating three-phase sinusoidal pulse width modulation. Main attention is concentrating to methods of generating Pulse width modulation method such as compare PWM modulation technique and space modulation of voltage vector SVM. The functional units of algorithms are analysed and described from the basics. In addition, this project describes the basic analog solutions and even the implementation of the PWM modulator into signal processors DSP56F800.

Klíčová slova

Algoritmus; dvoufázový; modulace; napětí; prostor; pulsní; referenční; šířková; vektor; vstřikovací; třífázový;

Keywords

Algorithm; injection; modulation; pulse; reference; space; tree-phase; two-phase; voltage; vector; width;

Bibliografická citace

RUDOLF, L. Algoritmy generování trojfázové sinusové pulsní šířkové modulace. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 35 s. Vedoucí bakalářské práce Ing. Bohumil Klíma, Ph.D.

Prohlášení

Prohlašuji, že svou bakalářskou práci na téma *ALGORITMY GENEROVÁNÍ TROJFÁZOVÉ SINUSOVÉ PULSNÍ ŠÍŘKOVÉ MODULACE (PWM)* jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne

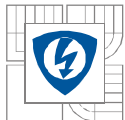
Podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Bohumilu Klímovi, PhD. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne

Podpis autora



Obsah

SEZNAM OBRÁZKŮ.....	8
SEZNAM TABULEK	9
SEZNAM SYMBOLŮ A ZKRATEK.....	10
1 ÚVOD.....	11
1.1 PROBLEMATIKA CELOČÍSELNÝCH DATOVÝCH TYPŮ.....	11
2 PRINCIPY GENEROVÁNÍ PWM.....	12
2.1 GENEROVÁNÍ POMOCÍ KOMPARACE S TROJÚHELNÍKOVÝM SIGNÁLEM.....	12
2.2 ČÍSLICOVÉ GENERÁTORY PWM.....	14
2.2.1 VLASTNOSTI ČÍSLICOVÉHO PWM MODULÁTORU V MIKROPROCESORU DSP56F800.....	14
2.3 DEFINICE VSTUPŮ A VÝSTUPŮ PWM MODULÁTORU.....	15
3 METODA STANDARDNÍ SVM	17
3.1 STAVY SEPnutí A JEJICH MATEMATICKÁ INTERPRETACE	17
3.2 URČENÍ SEKTORU.....	19
3.3 URČENÍ DOBY AKTIVNÍCH A NEAKTIVNÍCH STAVŮ	21
3.3.1 URČENÍ DOBY AKTIVNÍCH A NEAKTIVNÍCH STAVŮ PRO SEKTOR I.	21
3.4 VÝPOČET ZAPÍNACÍCH ČASŮ.....	25
3.5 REALIZACE STANDARD-SVM TECHNIKY PRO DSP56F800	26
3.5.1 GRAFICKÝ VÝSTUP PRO METODU STANDARDNÍ-SVM – SIMULÁTOR	27
3.5.2 GRAFICKÝ VÝSTUP PRO METODU STANDARD-SVM – DSP56F800	28
4 METODA SINE-CAP-SVM	29
4.1 POPIS	29
4.2 VÝPOČET ZÁKLADNÍCH STRÍD	29
4.3 URČENÍ A VÝPOČET SINE-CAP VOLTAGE.....	29
4.4 VÝPOČET VÝSLEDNÝCH POMĚRŮ STRÍD	30
4.5 REALIZACE SINE-CAP	30
4.5.1 GRAFICKÝ VÝSTUP PRO METODU SINE-CAP-SVM - SIMULÁTOR.....	31
4.5.2 GRAFICKÝ VÝSTUP PRO METODU SINE-CAP-SVM – DSP56F800	32
5 POROVNÁNÍ ALGORITMŮ	33
6 ZÁVĚR.....	34
LITERATURA.....	35
PŘÍLOHY	36
PŘÍLOHA Č.1 - PROGRAMOVÝ KÓD PRO METODU STANDARD-SVM.	36
PŘÍLOHA Č.2 - PROGRAMOVÝ KÓD PRO METODU SINE-CAP-SVM.	39



SEZNAM OBRÁZKŮ

Obrázek 2.1: Schéma zapojení principu pulsní šířkové modulace komparací nf. modulačního vf. Nosného signálu

Obrázek 2.2: Výsledný produkt pulsní šířkové modulace komparací nf. modulačního vf. nosného signálu

Obrázek 2.3: Diagram bloku pulsně šířkového modulátoru

Obrázek 2.4: Obecná realizace prostorového vektoru PWM modulátoru

Obrázek 2.5: Obecné vstupy a výstupy PWM modulátoru

Obrázek 2.6: Algoritmus komparační metody generování PWM

Obrázek 3.1: Schematické zapojení silové části metody SVM.

Obrázek 3.2: Referenční napětí U_{ref1} , U_{ref2} , U_{ref3}

Obrázek 3.3: Strom identifikace čísla sektoru

Obrázek 3.4: Možné přepínací stavy

Obrázek 3.5: Konstrukce vektorového statorového napětí U_S v sektoru I.

Obrázek 3.6: Detail konstrukce vektorového statorového napětí U_S v sektoru I.

Obrázek 3.7: Výsledek Standardní Vektorové modulace se zarovnáním na střed

Obrázek 3.8: Grafický výsledek Standardní Vektorové modulace v simulátoru

Obrázek 3.9: Grafický výsledek Standardní Vektorové modulace v DSP56F8320

Obrázek 4.1: Grafický výsledek sine-cap modulace v simulátoru

Obrázek 4.2: Grafický výsledek sine-cap modulace v DSP56F8320



SEZNAM TABULEK

Tabulka 2.1: Tabulka jednotlivých přepínacích stavů.

Tabulka 3.2: Clarkova transformace pro tří fázový systém.

Tabulka 3.3: Přiřazení poměrů pracovních cyklů k fázím motoru

Tabulka 5.1: Porovnání algoritmů z hlediska nároku na procesor



SEZNAM SYMBOLŮ A ZKRATEK

I	Elektrický proud	[A]
f	Frekvence	[Hz]
U	Elektrické napětí	[V]
Im	Imaginární část komplexního čísla	[-]
Re	Reálná část komplexního čísla	[-]
α	Obecný úhel	[° , rad.]
β	Obecný úhel	[° , rad.]
π	Ludolphovo číslo	[3,14159]
$u_{va}(t), u_{vb}(t), u_{vc}(t)$	Okamžité hodnoty větvoových napětí	[V]
$u_{ab}(t), u_{bc}(t), u_{ca}(t)$	Okamžité hodnoty sdružených napětí	[V]
$u_a(t), u_b(t), u_c(t)$	Okamžité hodnoty fázových napětí	[V]
$u_0(t)$	Okamžité hodnota vychylovacího napětí	[V]
U_{DC}	Napětí stejnosměrného meziobvodu	[V]



1 ÚVOD

Šířková modulace pro střídavý signál byla v realizována používáním analogových technik. Obvykle šířkové modulace (Pulse Width Modulation, PWM) signálů je dosaženo porovnáním trojúhelníkové nosné křivky s požadovanou modulační funkcí. Pro trojfázové zařízení jsou požadovány tři nezávislé šířkové modulační funkce, tzn. každá pro jednu fázi. Tyto tři modulační funkce využívané v PWM jsou úměrné požadovanému statorovému fázovému napětí. Těchto referenčních napětí je na výstupu dosaženo pomocí analogových proudových regulátorů.

Nicméně střídavé řízení se v současné době přiklání k digitálnímu provedení. Vygenerování signálu pomocí šířkové modulace se v digitální podobě stává základním problémem. Celý regulační algoritmus bývá realizován číslicově v mikroprocesoru a mikroprocesory obsahují periferní obvody pro generování pulsní šířkové modulace – generátory PWM. Tyto generátory jsou pak přímými digitální ekvivalenty komparace pilovitého nosného signálu a požadované modulační funkce. Tuto techniku známe jako pravidelné vzorkování, kdy jsou PWM signály odvozeny porovnáním pravidelného vzorkování modulační funkce a nosné vlny. Toto odvození je obdobné s analogovým způsobem generování PWM. Tyto schémata jsou aproximovány do tradičních sinus-trojúhelníkových strategií šířkových modulací.

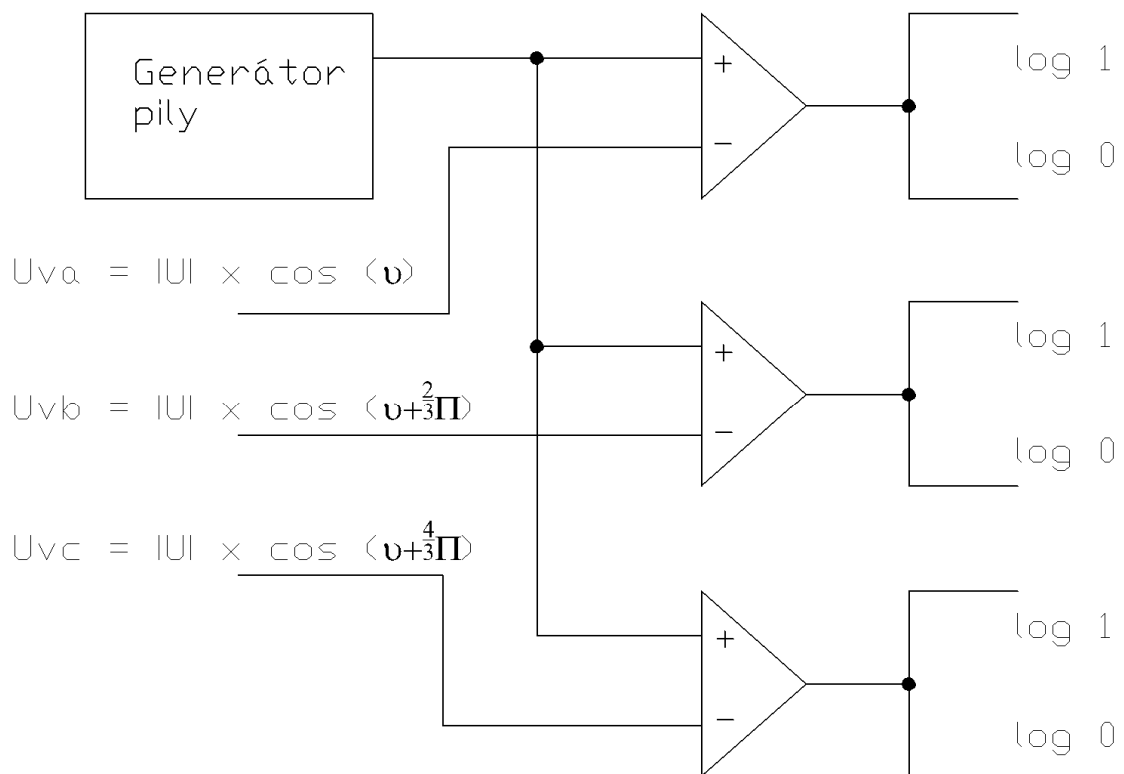
1.1 Problematika celočíselných datových typů

Veškeré výpočty, které jsem realizoval, v této bakalářské práci jsou prováděny v celočíselném datovém formátu `Frac16` (frakční část čísla). Rozsah datového typu `Frac16` je `<-32768;32767>` paměťová náročnost jsou 2 byte a je označován jako signed short integer. Pro náš případ používáme rozsah jednotkové kružnice `<-1,0;1,0>` výhodou je malá náročnost na výpočty tudíž na výkon procesorové jednotky a algoritmus používající tuto aritmetiku se tak stává velice rychlým, protože jak již bylo zmíněno dochází zde pouze k celočíselným operacím. Nevýhodou je určitá ztráta přesnosti jsme limitováni pouze určitým rozsahem, v našem případě počtem desetinných míst. Proto bylo zapotřebí použít speciálních funkcí jako je třeba násobení se zaokrouhlením do výše uvedeného rozsahu. Speciální funkce použitá v mém programu byla například „`mult_r`“ tedy násobení se zaokrouhlováním do celočíselného datového typu „`Frac16`“.

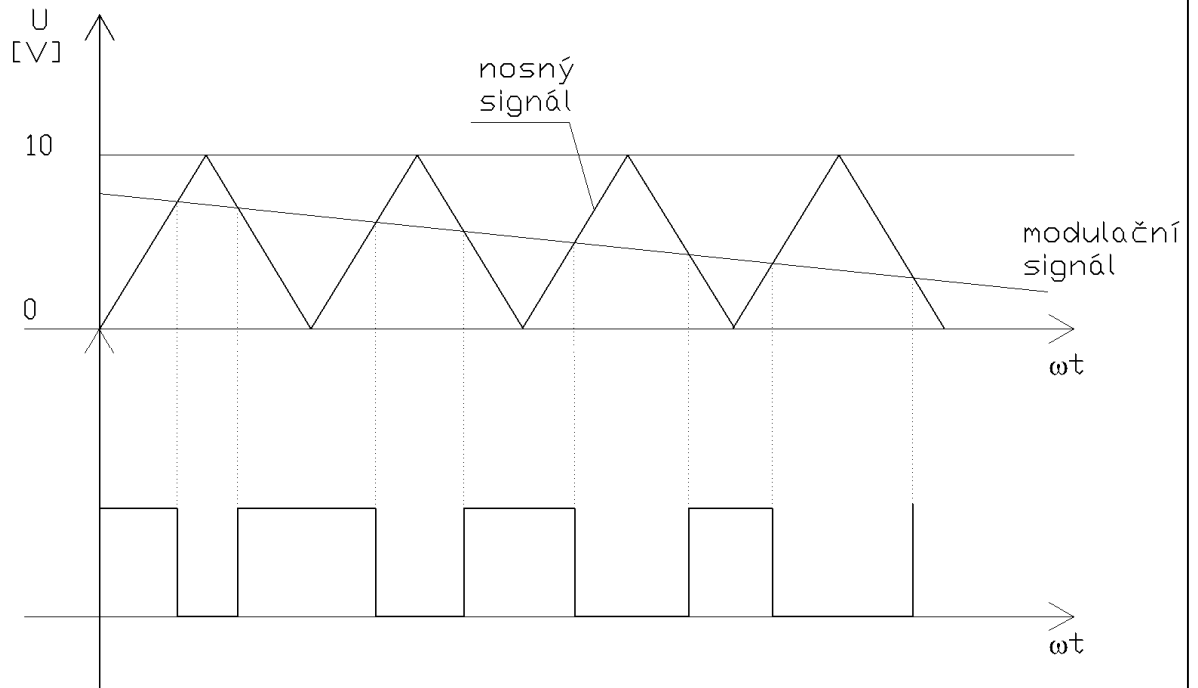
2 PRINCIPY GENEROVÁNÍ PWM

2.1 Generování pomocí komparace s trojúhelníkovým signálem

Pro generování spínacích signálů pro větve střídače existuje několik algoritmů, z nichž základní metoda je komparace vysokofrekvenčního nosného signálu tvaru pily s modulačním signálem - jak ukazuje obrázek 2.1 Na obrázku 2.2 je zakreslen takto vzniklý modulační produkt, ve kterém je obsažena rovněž nosná frekvence.



Obrázek 2.1 Schéma zapojení principu pulsní šířkové modulace komparací nf. modulačního vf. nosného signálu



Obrázek 2.2 Výsledný produkt pulsní šířkové modulace komparací nf. modulačního vf. nosného signálu

Budič výkonového tranzistoru by měl vždy zajišťovat tyto základní funkce:

- Úpravu napěťových úrovní
- Galvanické oddělení
- Nadproudová ochrana



2.2 Číslicové generátory PWM

Digitální řešení generátoru PWM signálu je shodné jako u analogového řešení. Pro aplikace v elektrických pohonech jsou vyráběny specializované mikroprocesory vybavené řadou periferních obvodů pro komunikaci a řízení. Mezi hlavní periferie potřebné k realizaci digitálních proudových smyček patří blok pulsních šířkových modulátorů PWM, bloky A/D převodníků, případně rozhraní pro snímače na hřídeli (IRC, resolver). Důležitým parametrem procesoru je výpočetní výkon udávaný v jednotkách MIPS (miliony instrukcí za sekundu). Moderní digitální signálové procesory pro aplikace v el. pohonech disponují výpočetní kapacitou 20 – 120 MIPS.

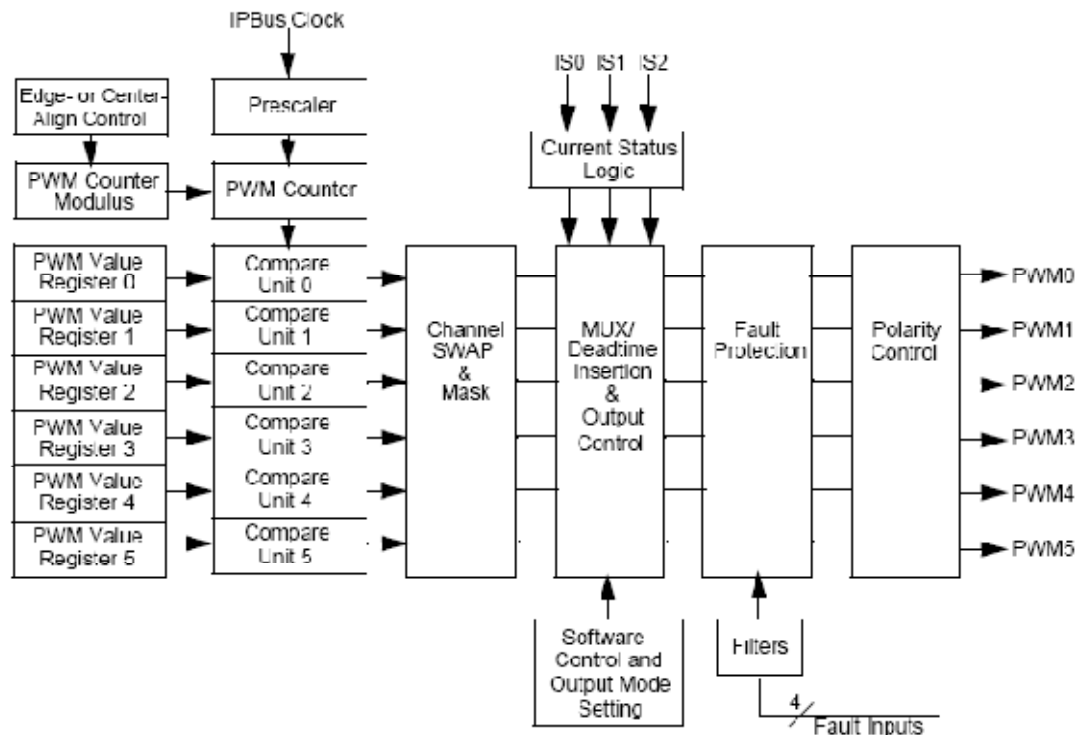
Číslicová (diskrétní) regulační smyčka je charakterizována dobou vzorkování. V případě elektrického pohonu s pulsně řízeným měničem se vzorkovací perioda vnitřní (proudové) regulační smyčky váže k periodě spínání měniče. Vzorkovací perioda je synchronizována s celými násobky periody nosného kmitočtu. Délka vzorkovací periody a tedy celý počet period nosné záleží na výpočetní kapacitě procesoru, za který je možné s rezervou provést výpočet proudových regulačních smyček a zajistit všechny ostatní procesorem vykonávané funkce pohonu. Pokud procesor disponuje dostatečným výpočetním výkonem, je optimální aby byla proudová smyčka vzorkována každou periodu nosného kmitočtu. Vysoký vzorkovací kmitočet pak zajišťuje dobré dynamické parametry proudové regulační smyčky.

V našem případě budeme signál PWM implementovat do signálového procesoru řady DSP56F800 FREESCALE Semiconductors.

2.2.1 Vlastnosti číslicového PWM modulátoru v mikroprocesoru DSP56F800

PWM modulátor může být konfigurován jako tři komplementární páry, šest nezávislých PWM signálů nebo jejich kombinací, třeba jako jeden komplementární nebo čtyři nezávislé.

Signálový procesor řady DSP56F800 podporuje oba typy pulsně šířkového řízení se zarovnáním na hranu a na střed (Edge-or Center-Align Control) pro modulaci od 0 do 100%. PWM čítač (PWM Counter) zajišťuje frekvenci nosného signálu pomocí frekvence taktu systémové sběrnice (IP Bus Clock) a předděličky (Prescaler). Požadovaná hodnota fázového napětí je uložena ve „value“ registru 0-5 (PWM Value Register 0-5), poté dochází ke komparaci nosné frekvence uložené v PWM čítači a modulačního signálu uloženého v „value“ registru 0-5. Při použití komplementárního PWM signálu, modulátor automaticky vkládá do každého pulsně šířkového výstupního páru ochranou dobu na fyzické přepnutí aktivních prvků (Deadtime Insertion). Blok modulátoru standardně zajišťuje ochranu programovatelnými vstupy (Fault Protection) a volbu polarit výstupních signálů (Polarity Control).

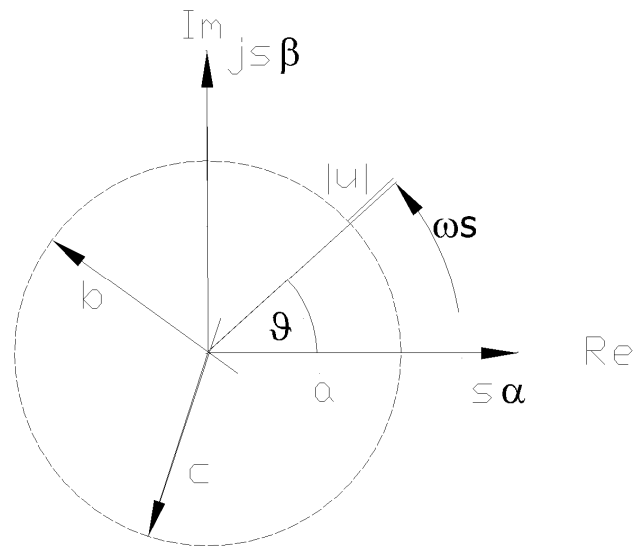


Obrázek 2.3. Diagram bloku pulsně šířkového modulátoru [2]

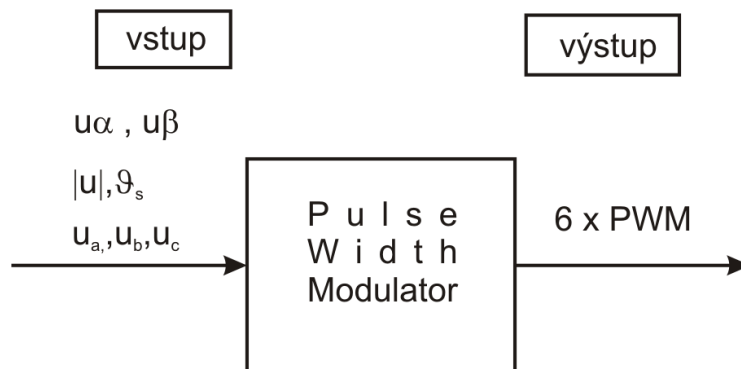
2.3 Definice vstupů a výstupů PWM modulátoru

Úkolem bloku PWM je, požadovaný okamžitý prostorový vektor napětí v podobě číselných údajů transformovat na tři okamžitá větvová napětí rovněž jako číselné signály, které vstupují do hardwarového generátoru PWM. Generátor PWM pak vytvoří odpovídající pulsní signály pro střídač. Střídač pak pulsní signály přemění na okamžité hodnoty napětí na motoru.

Pro naše účely definujeme jako vstup prostorový vektor napětí, který je dán buď ve složkovém tvaru \mathbf{u}_α , \mathbf{u}_β nebo jako modul a fáze $|u|, J$. Někdy je také vstupem požadavek fázových napětí na motoru u_a, u_b, u_c . Kvalitativně jsou uvedené možnosti vstupu rovnocenné.



Obrázek 2.4. Obecná realizace prostorového vektoru PWM modulátoru

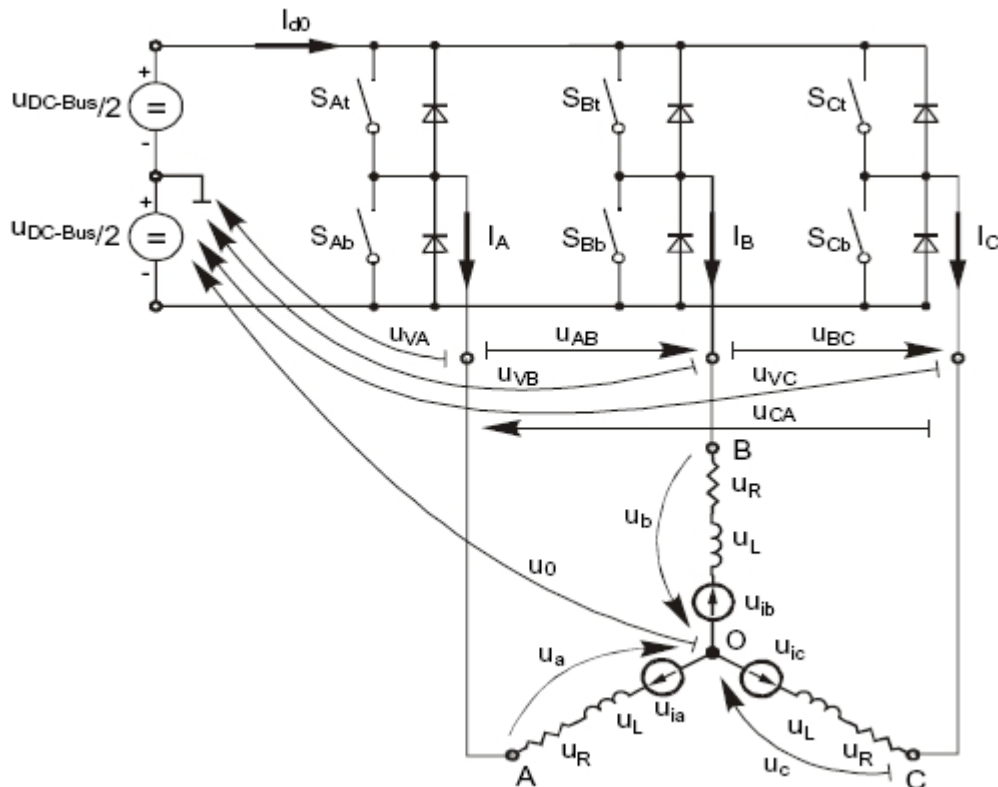


Obrázek 2.5. Obecné vstupy a výstupy PWM modulátoru

3 METODA STANDARDNÍ SVM

Pro regulační struktury, jejichž výstupem je prostorový vektor napětí ve složkovém tvaru se často používá modulační technika prostorové modulace vektoru napětí (Space Vector Modulation, SVM).

Základní princip standardní SVM techniky lze vysvětlit pomocí následujícího schematického obrázku napájení fází.



Obrázek 3.1. Definice napětí ve střídači.[3]

3.1 Stavby sepnutí a jejich matematická interpretace

Horní a dolní spínače pracují v doplňkovém režimu, tj. pokud horní spínač, "S_{At}", je ON, pak odpovídající dolní přepínač, "S_{Ab}", je OFF a naopak. Vzhledem k tomu, že hodnota log. „1“ je přiřazena k zapnutému stavu horního spínače a hodnota log. „0“ je přiřazena k zapnutému stavu spodního spínače, je pak možné definovat spínací vektor $[a, b, c]^T$. Vytvořením takového vektoru umožňuje numerické definice všech možných přepínání stavů. Sdružená napětí pak mohou být vyjádřeny v těchto stavech:

$$\begin{bmatrix} U_{AB} \\ U_{BC} \\ U_{CA} \end{bmatrix} = U_{DCBus} \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

Rovnice 3.1. Spínací vektor.



Kde U_{DC-BUS} je okamžité napětí naměřené na stejnosměrném meziobvodu. Předpokládáme, že motor je ideálně symetrický pak je možné psát maticové rovnice, která vyjadřují motorová fáze napětí, uvedené v rovnici 3.2.

$$\begin{bmatrix} U_a \\ U_b \\ U_c \end{bmatrix} = \frac{U_{DCBUS}}{3} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

Rovnice 3.2. Fázová napětí na motoru.

V trojfázovém střídači znázorněném na obrázku 3.1, je osm možných stavů sepnutí (diskrétních prostorových vektorů), které jsou uvedeny v obrázku 3.4. Tyto stavy spolu s výsledným okamžitým výstupem sdružených a fázových napětí, jsou uvedeny v tabulce 3.1.

a	b	c	U_a	U_b	U_c	U_{AB}	U_{BC}	U_{CA}	Vector
0	0	0	0	0	0	0	0	0	O_{000}
1	0	0	$2U_{DCBUS}/3$	$-U_{DCBUS}/3$	$-U_{DCBUS}/3$	U_{DCBUS}	0	U_{DCBUS}	U_0
1	1	0	$U_{DCBUS}/3$	$U_{DCBUS}/3$	$-2U_{DCBUS}/3$	0	U_{DCBUS}	U_{DCBUS}	U_{60}
0	1	0	$-U_{DCBUS}/3$	$2U_{DCBUS}/3$	$-U_{DCBUS}/3$	$-U_{DCBUS}$	U_{DCBUS}	0	U_{120}
0	1	1	$-2U_{DCBUS}/3$	$U_{DCBUS}/3$	$U_{DCBUS}/3$	$-U_{DCBUS}$	0	U_{DCBUS}	U_{240}
0	0	1	$-U_{DCBUS}/3$	$-U_{DCBUS}/3$	$2U_{DCBUS}/3$	0	$-U_{DCBUS}$	U_{DCBUS}	U_{300}
1	0	1	$U_{DCBUS}/3$	$-2U_{DCBUS}/3$	$U_{DCBUS}/3$	U_{DCBUS}	$-U_{DCBUS}$	0	U_{360}
1	1	1	0	0	0	0	0	0	O_{111}

Tabulka 3.1. Tabulka jednotlivých přepínacích stavů.

Složky prostorového vektoru α – podélná a β - příčná v dvoufázovém ortogonálním souřadnicovém systému popisují tři fázová statorová napětí, a jsou vyjádřeny v Clarkově transformaci, uspořádaných v rovnici matice 3.4.

$$\begin{bmatrix} U_\alpha \\ U_\beta \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} U_a \\ U_b \\ U_c \end{bmatrix}$$

Rovnice 3.4. Clarkova transformace pro tři fázový systém.

Tři fázová statorová napětí, U_A, U_B, U_C , jsou převedena pomocí Clarkovi transformace do α a β složek dvoufázového ortogonálního souřadnicového systému. Výsledek transformace je uveden v tabulce 3.2.



a	b	c	U_α	U_β	Vector
0	0	0	0	0	O_{000}
1	0	0	$2U_{DCBus}/3$	0	U_0
1	1	0	$U_{DCBus}/3$	$U_{DCBus}/\sqrt{3}$	U_{60}
0	1	0	$-U_{DCBus}/3$	$U_{DCBus}/\sqrt{3}$	U_{120}
0	1	1	$-2U_{DCBus}/3$	0	U_{240}
0	0	1	$-U_{DCBus}/3$	$-U_{DCBus}/\sqrt{3}$	U_{300}
1	0	1	$U_{DCBus}/3$	$-U_{DCBus}/\sqrt{3}$	U_{360}
1	1	1	0	0	O_{111}

Tabulka 3.2. Clarkova transformace pro tři fázový systém.

3.2 Určení sektoru

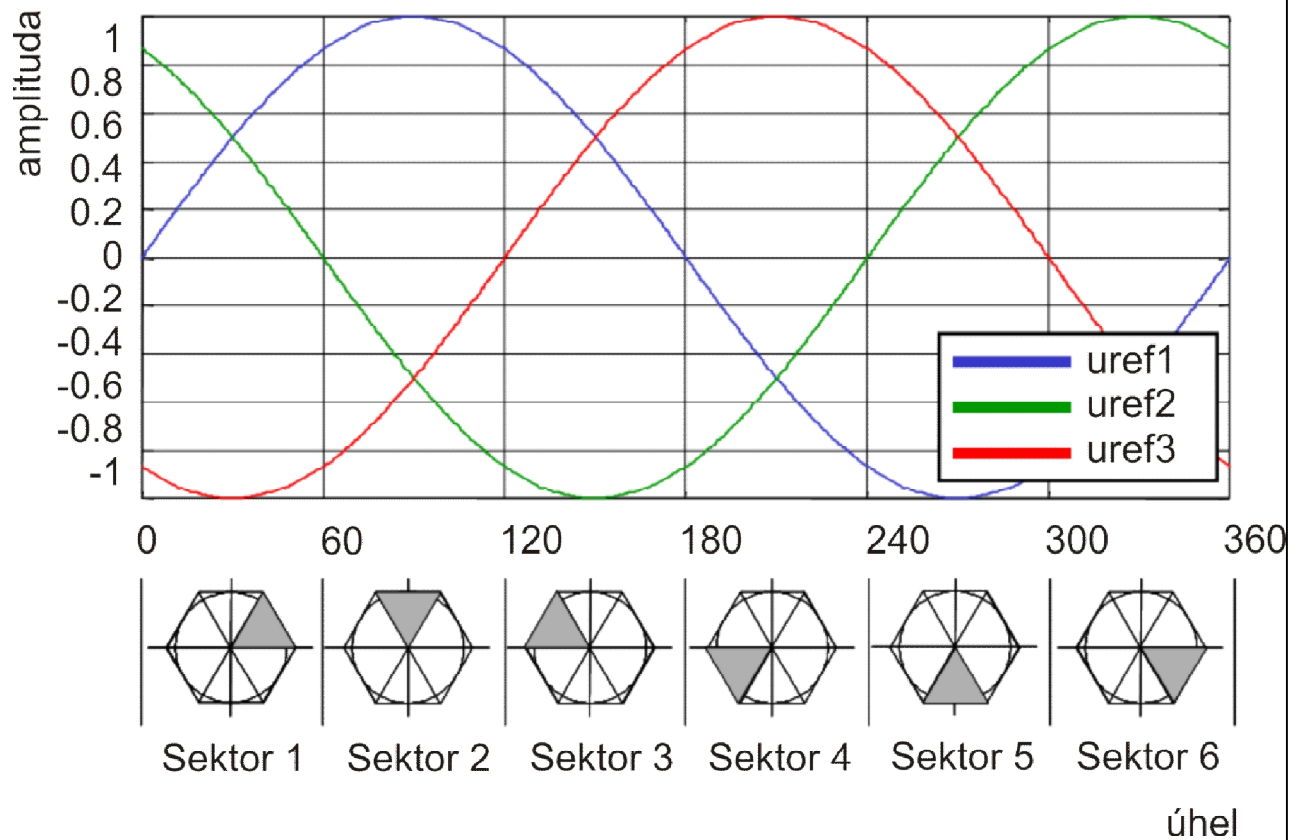
Prvním krokem v algoritmu prostorové modulace vektoru napětí je vymezení sektoru, ve kterém požadovaný napěťový prostorový vektor leží. K čemuž použijeme modifikovanou Inverzní Clarkovu Transformaci. α a β složky dvoufázového systému transformujeme do třífázového systému $u_{ref1}, u_{ref2}, u_{ref3}$, které se následně použijí pro určení sektoru.

$$u_{ref1} = u_\beta \quad (3.1)$$

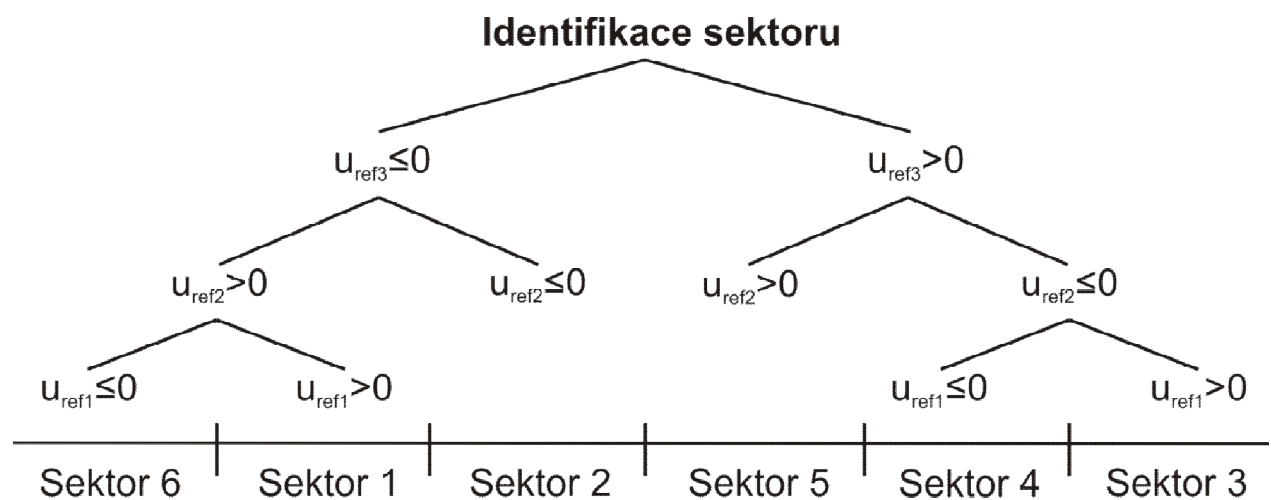
$$u_{ref2} = \frac{-u_\beta + \sqrt{3} \times u_\alpha}{2} \quad (3.2)$$

$$u_{ref3} = \frac{-u_\beta + \sqrt{3} \times u_\alpha}{2} \quad (3.3)$$

Třífázové referenční napětí



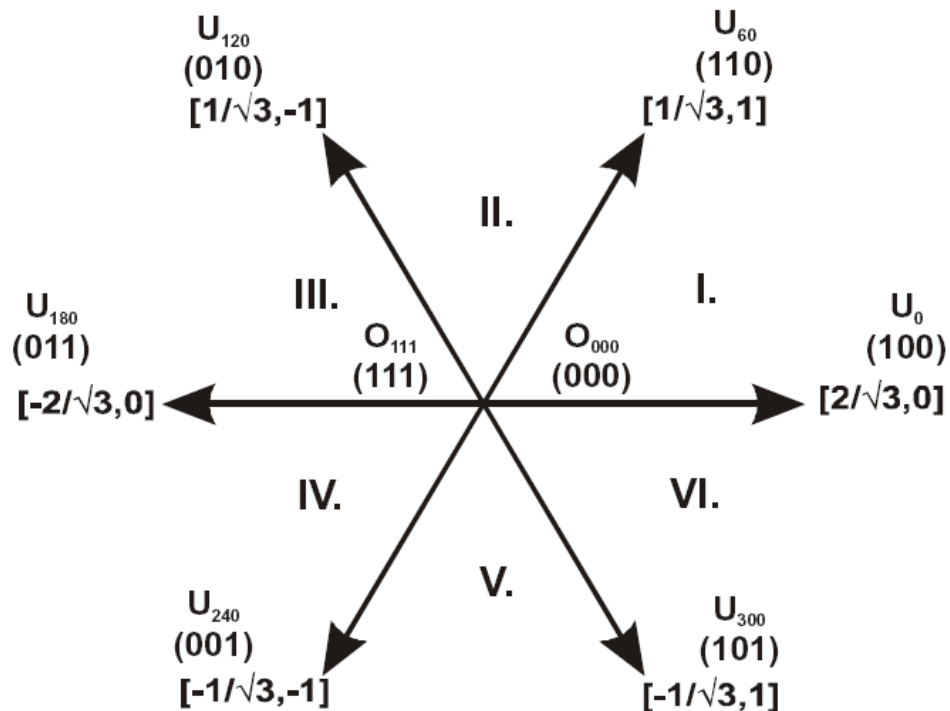
Obrázek 3.2. Referenční napětí



Obrázek 3.3. Strom identifikace čísla sektoru

3.3 Určení doby aktivních a neaktivních stavů

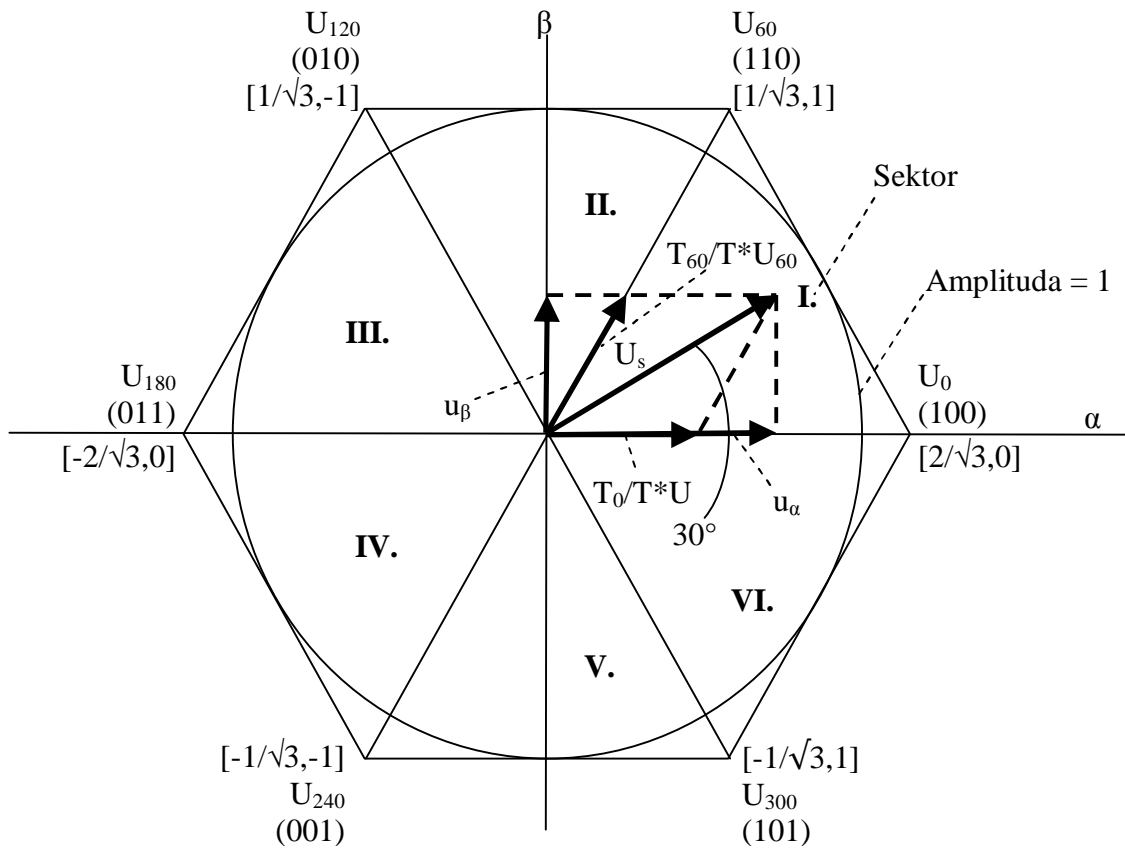
Pokud známe sektor, ve kterém leží výsledný vektor napětí prostorové modulace můžeme určit doby aktivních a neaktivních stavů. Obrázek 3.4 graficky znázorňuje některé možné základní přepínací stavy (vektory). Je zřejmé, že existuje šest nenulových vektorů, $U_0, U_{60}, U_{120}, U_{180}, U_{240}, U_{300}$ a dva vektory nulové, O_{111}, O_{000} , použitelné pro spínání. Proto standardní prostorová modulace vektoru napětí SVM spočívá v použití vhodného přepínání na určitou dobu, čímž vytváří výsledný napěťový vektor.



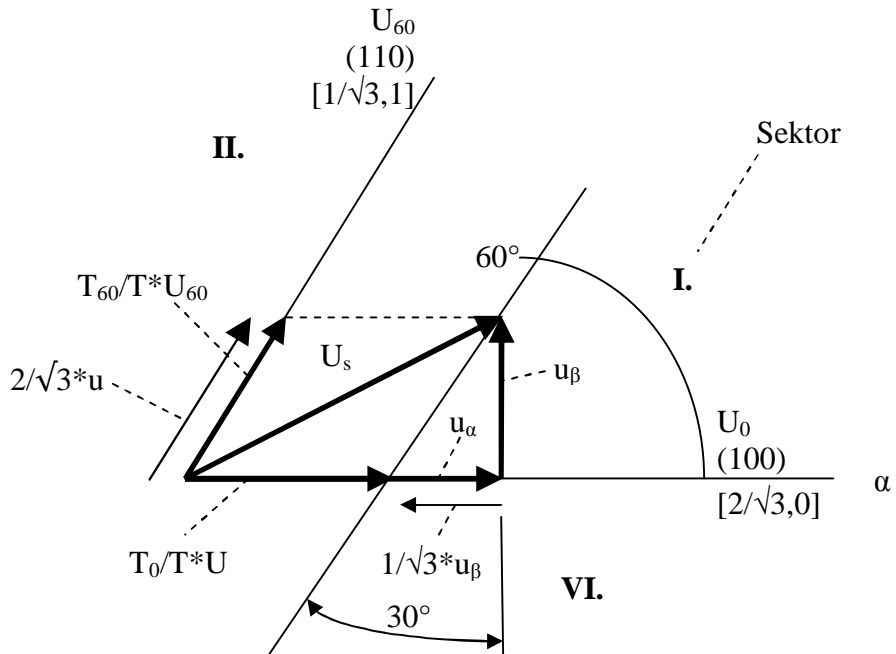
Obrázek 3.4. Možné přepínací stavy.

3.3.1 Určení doby aktivních a neaktivních stavů pro sektor I.

Cíl metody SVM je nahrazení požadovaného vektoru satorového napětí U_S s vhodným složením spínací sekvence ze základních prostorových vektorů. Vektor satorového referenčního napětí U_S je v případě sektoru I. fázově posunutý o 30° od reálné osy, a proto mohou být generovány vhodné kombinace z přilehlých základních přepínacích stavů U_0 a U_{60} . Tato projekce je ukázána na obrázku 3.5, detail projekce je vyobrazen na obrázku 3.6



Obrázek 3.5. Konstrukce vektorového statorového napětí U_s v sektoru I.



Obrázek 3.6. Detail konstrukce vektorového statorového napětí U_s v sektoru I.



Referenční vektor statorového napětí U_s se nachází v sektoru I a je možné tento vektor sestavit příslušnými poměry stříd základními přepínacími stavy U_0 a U_{60} . Hlavními rovnicemi tohoto vektoru polohy jsou:

$$T = T_{60} + T_0 + T_{null} \quad (3.4)$$

$$U_s = \frac{T_{60}}{T} \times U_{60} + \frac{T_0}{T} \times U_0 \quad (3.5)$$

kde T_{60} a T_0 , jsou doby, pro které jsou diskretní prostorové vektory U_0 a U_{60} v příslušné lhůtě T sepnuty.

T_{null} je doba, po kterou jsou nulové vektory O_{111}, O_{000} sepnuty. Tyto poměry lze vypočítat pomocí rovnice:

$$u_\beta = \frac{T_{60}}{T} \times |U_{60}| \times \sin 60^\circ \quad (3.6)$$

$$u_\alpha = \frac{T_0}{T} \times |U_0| + \frac{u_\beta}{\tan 60^\circ} \quad (3.7)$$

Vzhledem k tomu, že normované absolutní hodnoty diskretních prostorových vektorů vzhledem k napětí meziobvodu jsou:

$$|U_{60}| = |U_0| = \frac{2}{\sqrt{3}} \quad (3.8)$$

Nahrazením trigonometrických výrazů $\sin 60^\circ = \frac{2}{\sqrt{3}}$ a $\tan 60^\circ = \sqrt{3}$ a dosazením do rovnice 3.6 a 3.7 dostáváme výsledné poměry stříd

$$\frac{T_{60}}{T} = u_\beta \quad (3.9)$$

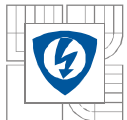
$$\frac{T_0}{T} = \frac{1}{2} \times (\sqrt{3} \times u_\alpha - u_\beta) \quad (3.10)$$

Stejným způsobem je možné dosáhnout příslušných spínacích poměrů stříd ve zbývajících sektorech.

V následující kapitole budeme tyto proměnné označovat jako T_0 a T_1 , přičemž v případě prvního sektoru je přiřazení následující:

$$T_0 = \frac{T_0}{T} \quad (3.11)$$

$$T_1 = \frac{T_{60}}{T} \quad (3.12)$$



3.4 Výpočet zapínacích časů

Proměnné t_1 , t_2 a t_3 , představující přepínací poměry stříd v odpovídajícím třífázovém systému a jsou uvedeny v následujících rovnicích:

$$t_1 = \frac{T - T_0 - T_1}{2} \quad (3.12)$$

$$t_2 = t_1 + T_0 \quad (3.13)$$

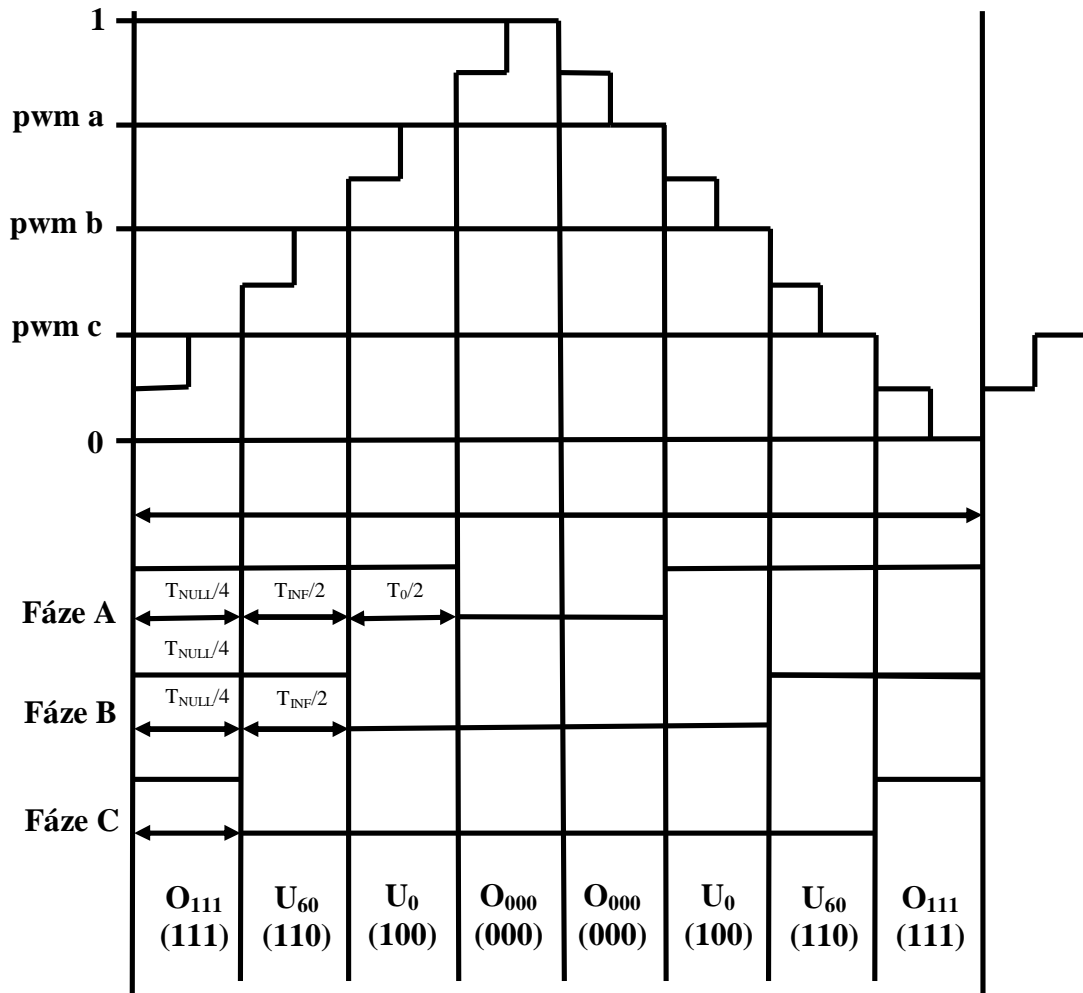
$$t_3 = t_2 + T_1 \quad (3.14)$$

kde T je doba přepínání (perioda), T_0 a T_1 jsou poměry stříd základních prostorů napěťových vektorů, které jsme sestavili v kapitole [3.3]. Dalším krokem je přiřazení správných poměrů pracovních cyklů, t_1 , t_2 a t_3 , příslušným fázím motoru. Kompletní přehled pozicí vektorů referenčních statorových napětí je v tabulce 3.2.

sectors	U_0, U_{60}	U_{60}, U_{120}	U_{120}, U_{180}	U_{180}, U_{240}	U_{240}, U_{300}	U_{300}, U_0
pwm_a	t_3	t_2	t_1	t_1	t_2	t_3
pwm_b	t_2	t_3	t_3	t_2	t_1	t_1
pwm_c	t_1	t_1	t_2	t_3	t_3	t_2

Tabulka 3.2. Přiřazení poměrů pracovních cyklů k fázím motoru

Generování pulsně šířkové modulace se zarovnáním na střed je dokončena porovnáním prahových úrovní, **pwm_a**, **pwm_b** a **pwm_c** s volně běžícím časovačem nahoru a dolů. Časovač počítá do 1 (0x7FFF) a poté dolů na 0 (0x0000). Předpokládá se, že když nějaký práh je větší než je hodnota časovače, příslušný PWM výstup je aktivní. V opačném případě je neaktivní.



Obrázek 3.7. Výsledek Standardní Vektorové modulace se zarovnáním na střed.

3.5 Realizace Standard-SVM techniky pro DSP56F800

Tato technika je již v dnešní době napsaná jako hotová knihovna pro náš cílový procesor resp. pro celou škálu různých obecně signálových procesorů. Celý program při použití této knihovny je pak velice jednoduchý a zabere pár řádků programového kódu.

```
sector = MCLIB_SvmStd (&twoPhSystem, &threePhSystem);
```

sector – číslo sektoru, ve kterém leží výsledný vektor

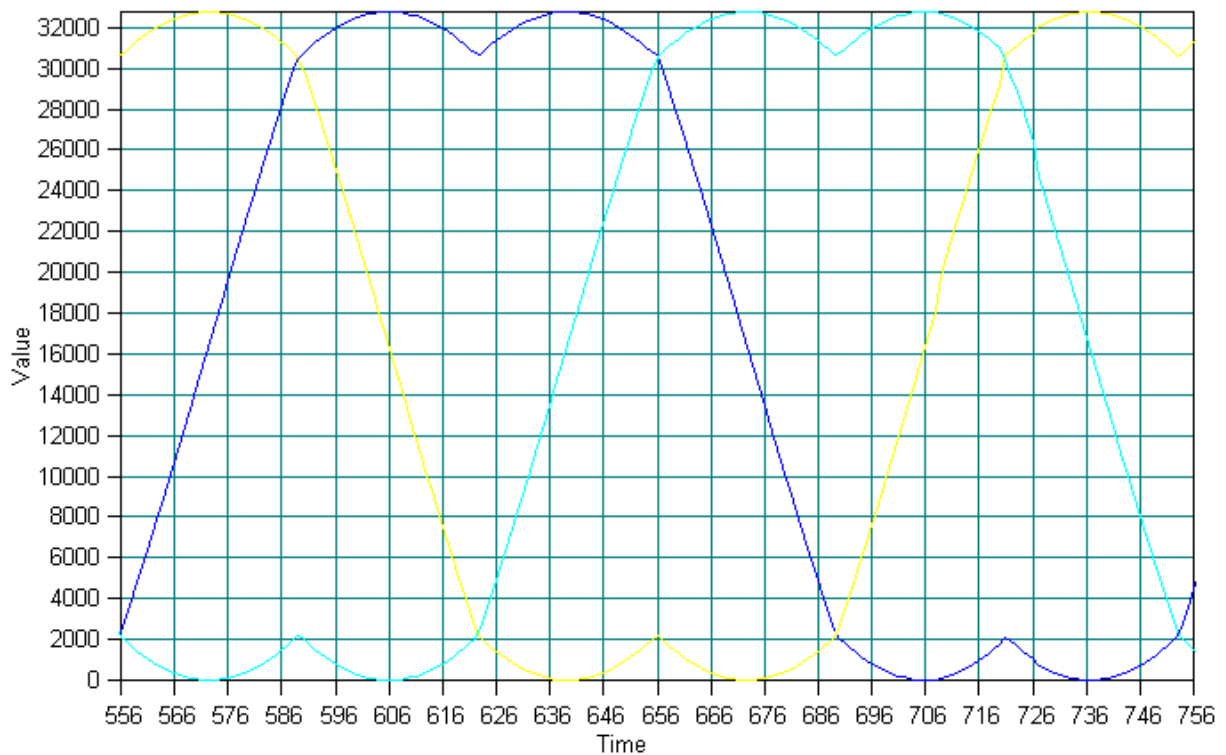
twoPhSystem – dvoufázový systém U_α, U_β

threePhSystem – výsledný třífázový systém pwm_a, pwm_b, pwm_c

Mým cílem bylo již hotovou knihovnu naprogramovat a zjistit jak je celý algoritmus náročný na mikroprocesor, abych byl schopen ji porovnat s jiným algoritmem nežli je Standardní SVM. Teoreticky popsanou techniku v kapitole [3], jsem realizoval dle zadání do signálového procesoru řady DSP56F800 FREESCALE Semiconductors. Výsledný kód je uveden v příloze této práce. Program probíhá dle teoretické části ve třech krocích

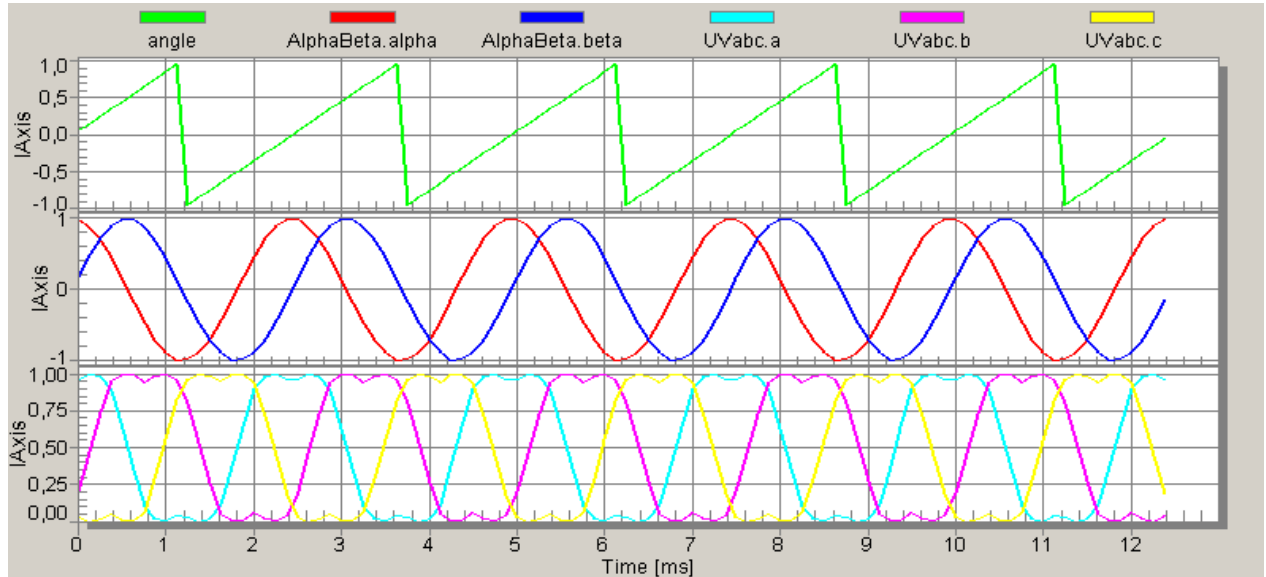
1. Zjištění sektoru, ve kterém leží výsledný vektor napětí.
2. Určení dob aktivních a neaktivních stavů.
3. Výpočet zapínacích časů.

3.5.1 Grafický výstup pro metodu Standardní-SVM – simulátor



Obrázek 3.8. Grafický výsledek Standardní Vektorové modulace v simulátoru

3.5.2 Grafický výstup pro metodu Standard-SVM – DSP56F800



Obrázek 3.9. Grafický výsledek Standardní Vektorové modulace v DSP56F800



4 METODA SINE-CAP-SVM

4.1 Popis

Tato funkce počítá příslušné střidy pro výrobu daného referenčního statorového napětí vektoru pomocí sinusové modulace tzv. Sine-Cap injection algorithm.

4.2 Výpočet základních stříd

Kalkulace základních poměrů stříd použitím inverzní Clarkovi Transformace jsou uvedeny v následujících rovnicích a jsou v rozsahu $[-1; 1]$

$$u_a = u_\alpha \quad (4.1)$$

$$u_b = \frac{-u_\alpha + \sqrt{3} \times u_\beta}{2} \quad (4.2)$$

$$u_c = \frac{-u_\alpha - \sqrt{3} \times u_\beta}{2} \quad (4.3)$$

Rovnice [4.1], [4.2] a [4.3] vynásobíme koeficientem

$$u'_a = \frac{2}{\sqrt{3}} \times u_a \quad (4.4)$$

$$u'_b = \frac{2}{\sqrt{3}} \times u_b \quad (4.5)$$

$$u'_c = \frac{2}{\sqrt{3}} \times u_c \quad (4.6)$$

4.3 Určení a výpočet Sine-Cap voltage

Tyto hodnoty poměných jsou v rozsahu $-2/\sqrt{3} < u'x < 2/\sqrt{3}$, proto je potřeba vhodně použít přepočítání na celočíselná frakční čísla, zároveň musí být zachována dostatečná přesnost. Pokud se hodnoty proměnných u'_a, u'_b, u'_c dostanou mimo rozměr uložený v pomocné proměnné U_0 , bude se tato proměnná se nazývat Sine-Cap voltage. Postup získání této proměnné je možné dle matematicky definovaných sérii tří vzorců:



$$u_0 = \begin{cases} 1.0 - u'_a & \text{if } u'_a > 1.0 \\ -1.0 - u'_a & \text{if } u'_a < -1.0 \\ 0 & \text{jinak} \end{cases} \quad (4.7)$$

$$u_0 = \begin{cases} 1.0 - u'_b & \text{if } u'_b > 1.0 \\ -1.0 - u'_b & \text{if } u'_b < -1.0 \\ 0 & \text{jinak} \end{cases} \quad (4.8)$$

$$u_0 = \begin{cases} 1.0 - u'_c & \text{if } u'_c > 1.0 \\ -1.0 - u'_c & \text{if } u'_c < -1.0 \\ 0 & \text{jinak} \end{cases} \quad (4.9)$$

4.4 Výpočet výsledných poměrů stříd

Vzhledem k 120° napět'ovému fázovému posunu, který je rozlišovací pro vyvážený třífázový systém bude pouze jedna fáze přispívat k vytvoření Sine-Cap voltage U_0 v každém bodě.

Výsledné poměry stříd jsou počítány pomocí následujících rovnic:

$$pwm_a = \frac{1}{2} \times (u_0 + u'_0 + 1) \quad (4.10)$$

$$pwm_b = \frac{1}{2} \times (u_0 + u'_b + 1) \quad (4.11)$$

$$pwm_c = \frac{1}{2} \times (u_0 + u'_c + 1) \quad (4.12)$$

4.5 Realizace Sine-Cap

Stejně jako předchozí metoda, je tato také vytvořena v podobě knihovny, která se volá stejným způsobem

```
sector = MCLIB_SvmSci (&twoPhSystem, &threePhSystem);
```

sector – číslo sektoru, ve kterém leží výsledný vektor

twoPhSystem – dvoufázový systém U_α, U_β

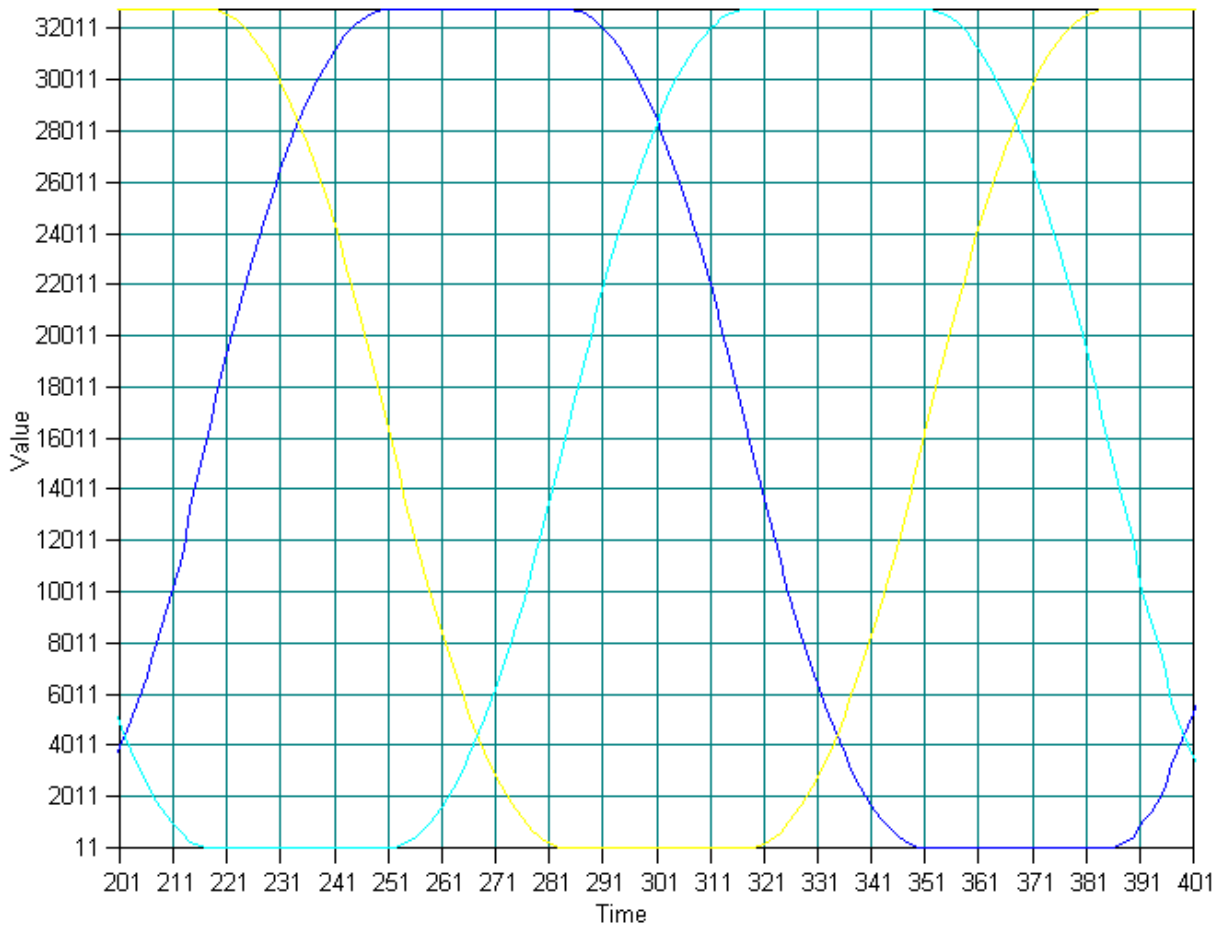
threePhSystem – výsledný třífázový systém pwm_a, pwm_b, pwm_c

Teoreticky popsanou techniku v kapitole [4], jsem opět realizoval dle zadání do signálového procesoru řady DSP56F8000 FREESCALE Semiconductors. Výsledný kód je uveden v příloze této práce.

Program probíhá dle teoretické části ve třech krocích

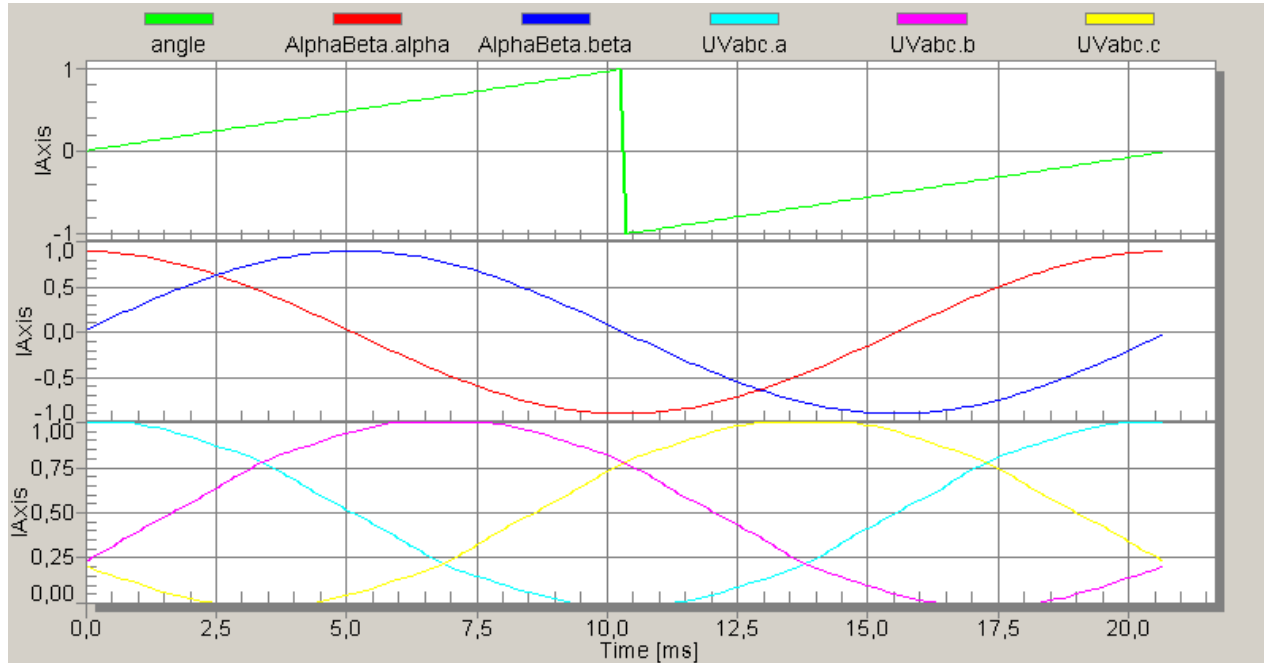
1. Výpočet základních stříd
2. Určení a výpočet sine-cap voltage
3. Výpočet výsledných poměrů stříd

4.5.1 Grafický výstup pro metodu Sine-Cap-SVM - simulátor



Obrázek 4.1. Grafický výsledek sine-cap modulace v simulátoru

4.5.2 Grafický výstup pro metodu Sine-Cap-SVM – DSP56F800



Obrázek 4.2. Grafický výsledek sine-cap modulace v DSP56F800



5 POROVNÁNÍ ALGORITMŮ

Porovnáním algoritmů z hlediska nároku na procesor jsem došel k těmto výsledkům, souhrnně jsou ukázány v tabulce 5.1. Pro porovnání jsem použil interní nástroj Metrowerks CodeWarrior, Display Cycle/Instruction count. Porovnání počtu cyklů a instrukcí probíhalo pro jednu iteraci cyklu algoritmu.

	SVM_STD	SVM_SCI
Cykl, MCLIB	334	572
Instrukce, MCLIB	177	315
Cykl, vlastní fce	446	524
Instrukce, vlastní fce	216	283

Tabulka 5.1. Porovnání algoritmů z hlediska nároku na procesor



6 ZÁVĚR

Hlavním cílem této bakalářské práce bylo popsat a vytvořit dva algoritmy generování pulsně šířkové modulace.

Bakalářská práce byla rozložena do čtyř logických celků. Nejprve jsem obecně popsal generování pulsně šířkové modulace pro analogové obvody. Druhá kapitola byla věnována algoritmu Standard-SVM. Předposlední kapitola se zabývá algoritmem Sine-Cap-SVM. Poslední část rozebírá naměřené hodnoty a grafické výsledky.

Porovnáním jednotlivých algoritmů za účelem zjištění nároku na procesor, je metoda Standard-SVM (svmStd) méně náročnější. Již hotový algoritmus MCLIB_svmStd od společnosti Freescale Semiconductors je méně náročný nežli můj vlastní algoritmus. U algoritmu Sine-Cap-SVM (svmSci) je tomu naopak, kde můj algoritmus zabírá o 48cyklů taktů procesoru méně. Zřejmě je to způsobeno tím, že algoritmus MCLIB_svmSci, volá funkci zjišťování sektoru, kde leží výsledný napěťový vektor, to ovšem u metody Sine-Cap není třeba.

Tato bakalářská práce splňuje všechny body zadání. Navíc jsem si osvojil problematiku generování pulsně šířkových algoritmů pro řízení třífázových asynchronních motorů a částečně jsem pronikl do problematiky celočíselného zpracování kódu v signálových mikroprocesorech, které jsou v dnešní době široce použité pro řízení v mnoha odvětvích průmyslu. Vyznačují se především rychlostí, jednoduchostí provedení a cenou. Zdrojový kód je uveden v příloze a zároveň je zaznamenán na přiloženém CD.



LITERATURA

- [01] Ing.Bohumil Klíma PhD. Disertační práce, Vektorově orientované řízení synchronního stroje s permanentními magnety. 2002 Brno.
- [02] Analog Devices, Applications Motor Control – Space Vector Modulation , datasheet.
- [03] Freescale Semiconductor, Motor Control Library for 56F83xx Hybrid Controllers
- [04] DSP56800E 16-bit DSP Core Reference Manual; Motorola (DSP56800ERM/D)
- [05] DSP56F83x Family 16-Bit Digital Signal Processor Peripheral Manual; Motorola (DSP56F83xUM/D)
- [06] Prata, Stephen: *Mistrovství v C++*, 3. aktualizované vydání. Computer Press, 10/2008, ISBN: 978-80-251-1749-1, EAN: 978-80-251-1749-1
- [07] The C++ Resources Network, 2008, Dostupné z <<http://www.cplusplus.com/>>



PŘÍLOHY

Příloha č.1 - Programový kód pro metodu Standard-SVM.

```
//Ziskani dvoufazoveho systemu
Ualpha = mult_r(MCLIB_Sin(angle),amplitude);
Ubeta = mult_r(MCLIB_Cos(angle),amplitude);

////Urceni cisla sektoru, vypocet referencnich napeti
Urf1=Ubeta;
Urf2=mult_r(-Ubeta,k05)+(mult_r(k0866,Ualpha));
Urf3=mult_r(-Ubeta,k05)-(mult_r(k0866,Ualpha));
if (Urf3>0)
{
    if (Urf2>0)
    {
        mSector=5;
    }
    else
    {
        if (Urf1>0)
        {
            mSector=3;
        }
        else
        {
            mSector=4;
        }
    }
}
else
{
    if (Urf2<=0)
    {
```



```
mSector=2;
}
else
{
    if (Urf1>0)
    {
        mSector=1;
    }
    else
    {
        mSector=6;
    }
}
}

//Vypocet dob aktivnich stavu
switch(mSector)
{
    case 1: T0=Ubeta; //T0
            T1=mult_r(k0866,Ualpha)-mult_r(k05,Ubeta); //T60
            break;
    case 2: T0=mult_r(k05,Ubeta)+mult_r(k0866,Ualpha); //T60
            T1=mult_r(k05,Ubeta)-mult_r(k0866,Ualpha); //T120
            break;
    case 3: T0=mult_r(k0866,-Ualpha)-mult_r(k05,Ubeta); //T120
            T1=Ubeta; //T180
            break;
    case 4: T0=mult_r(FRAC16(-0.866),Ualpha)+mult_r(k05,Ubeta); //T180
            T1=-Ubeta; //T240
            break;
    case 5: T0=mult_r(k05,-Ubeta)+mult_r(k0866,Ualpha); //T240
            T1=mult_r(k05,-Ubeta)-mult_r(k0866,Ualpha); //T300
            break;
```



```
case 6: T0=-Ubeta; //T300
        T1=mult_r(k0866,Ualpha)+mult_r(k05,Ubeta); //T0
        break;
    }
//Vypocet stavu jednotlivych vetvi
t1=mult_r(k05,(FRAC16(1)-T0-T1));
t2=t1+T0;
t3=t2+T1;
//Prirazeni spravnym vystupum
switch(mSector)
{
case 1: pwm_a=t3;
        pwm_b=t2;
        pwm_c=t1;
        break;
case 2: pwm_a=t2;
        pwm_b=t3;
        pwm_c=t1;
        break;
case 3: pwm_a=t1;
        pwm_b=t3;
        pwm_c=t2;
        break;
case 4: pwm_a=t1;
        pwm_b=t2;
        pwm_c=t3;
        break;
case 5: pwm_a=t2;
        pwm_b=t1;
        pwm_c=t3;
        break;
case 6: pwm_a=t3;
        pwm_b=t1;
```



```
pwm_c=t2;  
break;  
}
```

Příloha č.2 - Programový kód pro metodu Sine-Cap-SVM.

```
//Ziskani dvoufazoveho systemu
```

```
Ualpha = mult_r(MCLIB_Sin(angle),amplitude);
```

```
Ubeta = mult_r(MCLIB_Cos(angle),amplitude);
```

```
//Vypocet zakladnich strid
```

```
Ua=Ualpha;
```

```
Ub=mult_r(-Ualpha,FRAC16(0.5))+(mult_r(FRAC16(0.866),Ubeta));
```

```
Uc=mult_r(-Ualpha,FRAC16(0.5))-(mult_r(FRAC16(0.866),Ubeta));
```

```
//Urceni a vypocet sine-cap voltage
```

```
if (Ua>(FRAC16(0.866)))
```

```
{
```

```
U0=(FRAC16(0.866))-Ua;
```

```
}
```

```
else {
```

```
    if (Ua<(FRAC16(-0.866)))
```

```
    {
```

```
        U0=(FRAC16(-0.866))-Ua;
```

```
    }
```

```
    else {
```

```
        if (Ub>(FRAC16(0.866)))
```

```
        {
```

```
            U0=(FRAC16(0.866))-Ub;
```

```
        }
```

```
    else {
```

```
        if (Ub<(FRAC16(-0.866)))
```

```
        {
```

```
            U0=(FRAC16(-0.866))-Ub;
```

```
        }
```

```
    else{
```




```
if (Uc>(FRAC16(0.866)))
{
U0=(FRAC16(0.866))-Uc;
}
else{
    if (Uc<(FRAC16(-0.866)))
    {
        U0=(FRAC16(-0.866))-Uc;
    }
    else
    {
        U0=0;
    }
}
}
}
}
```

pwm_a=FRAC16(0.5)+mult_r(FRAC16(0.577),(U0+Ua));

pwm_b=FRAC16(0.5)+mult_r(FRAC16(0.577),(U0+Ub));

pwm_c=FRAC16(0.5)+mult_r(FRAC16(0.577),(U0+Uc));