

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2016

Bc. Jiří Bareš



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV VÝKONOVÉ ELEKTROTECHNIKY A ELEKTRONIKY

DEPARTMENT OF POWER ELECTRICAL AND ELECTRONIC ENGINEERING

NÁVRH A REALIZACE AKTIVNÍHO TROJFÁZOVÉHO USMĚRŇOVAČE

CONCEPT OF THREE-PHASE ACTIVE RECTIFIER

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Jiří Bareš

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Petr Huták, Ph.D.

BRNO 2016

Diplomová práce

magisterský navazující studijní obor **Silnoproudá elektrotechnika a výkonová elektronika**

Ústav výkonové elektrotechniky a elektroniky

Student: Bc. Jiří Bareš

ID: 146780

Ročník: 2

Akademický rok: 2015/16

NÁZEV TÉMATU:

Návrh a realizace aktivního trojfázového usměrňovače

POKYNY PRO VYPRACOVÁNÍ:

1. Nasimulujte regulaci aktivního usměrňovače.
2. Realizujte navržené řídicí algoritmy na usměrňovači firmy ELCOM.
3. Proveďte měření a vyhodnocení navrženého usměrňovače.

DOPORUČENÁ LITERATURA:

[1] Skalický, J.: Teorie řízení, skripta FEKT, 2002

[2] Shinnars, S.,M.:Advanced Modern Control System Theory and Design, Wiley

Termín zadání: 21.9.2015

Termín odevzdání: 24.5.2016

Vedoucí práce: Ing. Petr Huták, Ph.D.

Konzultant diplomové práce:

Ing. Ondřej Vítek, Ph.D., předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové 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 části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt

Tato práce pojednává o řízení trojfázového aktivního usměrňovače. V první, teoretické části, popisuje několik různých strategií řízení, se zaměřením na napět'ově orientovanou strategii. Pro ni je v práci vypracován návrh, popsán vytvořený model a jeho následná simulace.

Druhá, praktická část se zabývá realizací řízení na kontroleru TMS320F28335, který je řídicí jednotkou zařízení zapůjčeného firmou Elcom. V této části jsou tedy popsány možnosti a nastavení kontroleru a vytvořený algoritmus řízení. Následují dosažené výsledky měření, které jsou v závěru zhodnoceny.

Abstract

This work deals with three-phase active rectifier control. In first, theoretical part it describes several control strategies with focus on voltage oriented control. For this type of control a design is worked out, and created model and its simulation is described.

Second, practical part deals with realisation on microcontroller TMS320F28335, which is main processing unit of device lent by company Elcom. Therefore in this part abilities and settings of the controller along with developed control algorithm are described. Reached results of measurement are then presented and in the conclusion they are evaluated.

Klíčová slova

Trojfázový aktivní usměřovač, PWM, kaskádní regulace, prostorový vektor, napětově orientované řízení, VOC, Matlab Simulink, model, TMS320F28335, Code Composer Studio, fázový závěs, PLL

Keywords

Three phase active rectifier, PWM, cascade regulation, space vector, Voltage oriented Control, VOC, Matlab Simulink, model, TMS320F28335, Code Composer Studio, phase-locked loop, PLL

Bibliografická citace

BAREŠ, J. *Návrh a realizace aktivního trojfázového usměrňovače*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2016. 44 s. Vedoucí diplomové práce Ing. Petr Huták, Ph.D..

Prohlášení

Prohlašuji, že svou diplomovou práci na téma Návrh a realizace aktivního trojfázového usměrňovače jsem vypracoval samostatně pod vedením vedoucího diplomové 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 diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové 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 diplomové práce Ing. Petru Hutákovi, Ph.D za odbornou pomoc, konzultace a vedení při zpracování mé diplomové práce. Dále bych chtěl poděkovat Dr. Ing Tomáši Bůbelovi za téma diplomové práce, poskytnuté materiály a zajištění spolupráce s firmou Elcom, Ing. Tomáši Žůrkovi za pomoc při měření na zařízení, cenné rady a poskytnuté materiály a hlavně čas a ochotu se mi věnovat. Také bych chtěl poděkovat Bohu i dalším lidem, kteří mi s prací jakkoli pomohli.

V Brně dne

Podpis autora



OBSAH

1 ÚVOD.....	13
2 AKTIVNÍ USMĚRŇOVAČ.....	13
2.1 SCHÉMA.....	13
2.2 PULZNÍ ŠÍRKOVÁ MODULACE - PWM.....	14
3 TRANSFORMACE.....	15
4 ZPŮSOBY ŘÍZENÍ.....	16
4.1 NAPĚŤOVĚ ORIENTOVANÉ ŘÍZENÍ.....	17
4.2 METODA VIRTUÁLNÍHO TOKU.....	18
4.3 PŘÍMÉ ŘÍZENÍ VÝKONU.....	19
5 NÁVRH REGULÁTORŮ.....	20
5.1 REGULÁTOR PROUDU.....	20
5.2 REGULÁTOR NAPĚTÍ.....	21
6 SIMULACE.....	22
6.1 MODEL.....	22
6.2 VLASTNÍ SIMULACE.....	23
7 PROSTŘEDÍ PRO REALIZACI.....	25
7.1 PARAMETRY ZAŘÍZENÍ.....	25
7.2 TYP ŘÍDÍCÍ JEDNOTKY.....	25
7.3 VÝVOJOVÉ PROSTŘEDÍ.....	26
7.4 MODULY MIKROKONTROLERU.....	26
7.4.1 A/D PŘEVODNÍK.....	26
7.4.2 PWM MODUL.....	27
7.4.3 PIE MODUL.....	28
7.5 CODE COMPOSER STUDIO.....	29
8 KÓD.....	30
8.1 POUŽITÉ EXTERNÍ FUNKCE.....	30
8.2 VLASTNÍ FUNKCE.....	31
8.2.1 PI REGULÁTOR.....	31
8.2.2 DOLNÍ PROPUST.....	32
8.2.3 FÁZOVÝ ZÁVĚS.....	32
8.2.4 KRUHOVÉ OMEZENÍ.....	33
8.2.5 CLARKOVA TRANSFORMACE.....	34
8.2.6 PARKOVA TRANSFORMACE.....	34
8.2.7 INVERZNÍ PARKOVA TRANSFORMACE.....	34
8.2.8 ABSOLUTNÍ HODNOTA.....	35
8.2.9 MODULACE PROSTOROVÉHO VEKTORU (SVM).....	35
8.3 DALŠÍ VÝPOČTY.....	37
8.4 NASTAVENÍ MODULŮ.....	37
8.4.1 NASTAVENÍ ADC.....	37



8.4.2 NASTAVENÍ EPWM.....	38
8.4.3 NASTAVENÍ PIE.....	38
9 MEŘENÍ.....	39
9.1 NASTAVENÍ PROMĚNNÝCH.....	39
9.2 VÝSLEDKY MĚŘENÍ.....	40
10 ZÁVĚR.....	42
LITERATURA.....	43
PŘÍLOHY.....	44



SEZNAM OBRÁZKŮ

Obr. 1: Schéma trojfázového aktivního usměrňovače.....	14
Obr. 2: Schéma náhradního jednofázového modelu usměrňovače.....	14
Obr. 3: Princip pulzní šířkové modulace, převzato z [5].....	15
Obr. 4: Blokové schéma napětově orientovaného řízení.....	17
Obr. 5: Blokové schéma napětového řízení s virtuálním tokem.....	19
Obr. 6: Blokové schéma vnitřní proudové smyčky.....	20
Obr. 7: Blokové schéma vnější napětové smyčky.....	21
Obr. 8: Schéma PI regulátoru.....	23
Obr. 9: Výsledné průběhy simulace modelu. Odshora: vstupní síťové napětí, vstupní odebírané proudy, výstupní usměrněné napětí.....	23
Obr. 10: Detail průběhů odebíraných proudů a výstupního napětí.....	24
Obr. 11: Blokový diagram A/D převodníku, převzato z [7].....	26
Obr. 12: Blokový diagram ePWM modulu, převzato z [8].....	27
Obr. 13: Diagram přerušení s použitím modulu PIE, převzato z [9].....	27
Obr. 14: Prostředí programu Code Composer Studio.....	28
Obr. 15: Blokové schéma realizace kódu, převzato z [10].....	29
Obr. 16: Blokové schéma PI regulátoru.....	30
Obr. 17: Blokové schéma fázového závěsu.....	32
Obr. 18: Grafické znázornění výpočtu svm, převzato z [13].....	35
Obr. 19: Schéma nastavení odskoku ePWM modulu, převzato z [8].....	38
Obr. 20: Graf fázového napětí a generovaného sinusového průběhu.....	40
Obr. 21: Graf úhlu vypočteného fázovým závěsem.....	41
Obr. 22: Graf fázového proudu a průběhu signálu z modulátoru prostorového vektoru.....	41



SEZNAM TABULEK

Tabulka 1: Parametry usměřovače.....	22
Tabulka 2: Parametry měřeného zařízení.....	24



SEZNAM SYMBOLŮ A ZKRATEK

Symboly:

U	napětí
I	proud
U_s	větвовé napětí
R_L	parazitní odpor cívky
L	indukčnost
C	kapacita
R	odpor
P	výkon
Ψ	virtuální tok
x	obecná veličina
θ	úhel
ω	úhlová rychlost
f	frekvence
t	čas
τ	časová konstanta
F	Přenos systému
K	Konstanta zesílení
c	Clarkova konstanta
T_s	vzorkovací frekvence
$Y(p)$	výstup systému
$U(p)$	vstup systému

Indexy:

a,b,c	trojfázový systém
l	vstupní, síťová veličina
s	větвовá veličina
α, β	dvoufázový statický souřadný systém
d,q	synchronní dvoufázový systém
DC	stejnoseměrná, usměrněná veličina
w	žádaná hodnota
m	měníč



ci	čidlo otáček
cu	čidlo proudu
W	uzavřená smyčka
p	proporcionální
i	integrační
lp	dolní propust

Zkratky:

PWM	pulzní šířková modulace
PI	proporcionálně integrační
EC	elektronicky komutovaný
VOC	napětově orientované řízení
VF	virtuální tok
VFOC	napětově orientované řízení s virtuálním tokem
DPC	přímé řízení výkonu
MOS-FET	unipolární tranzistor
IGBT	tranzistor s izolovanou bází
RTS	s podporou v reálném čase
ADC	analogově digitální převodník
PIE	rozšíření přerušení pro periferie
CPU	hlavní řídicí jednotka
TI	Texas Instruments
CCS	Code Composer Studio



1 ÚVOD

Usměrňovač je měnič převádějící střídavé napětí na stejnosměrné a je častou součástí velkého množství různých zařízení. Pro větší výkony však není vhodné použít diodový usměrňovač, který odebírá z napájecí sítě proud s velkým podílem vyšších harmonických frekvencí. V těchto aplikacích se využívá usměrňovač aktivní. Aktivní usměrňovač obsahuje místo diod tranzistory, jejichž spínáním umožňuje usměrňovat s průběhem proudu odebíraného ze sítě bez vyšších harmonických složek a využívat dalších výhod oproti klasickému diodovému usměrňovači. Proto je často používán například pro EC motor nebo stejnosměrný meziobvod pro asynchronní motor s měničem.

Pro využití všech požadovaných funkcí měniče je potřeba zvolit vhodnou strategii řízení. Strategií existuje mnoho druhů, zde se však budeme zabývat pouze některými. Řízením je především potřeba zajistit správnou funkci usměrňovače a poté jej řídit tak, aby bylo dosaženo požadovaných výsledků v oblasti rušení a odebíraného proudu. Spolehlivost řízení a jeho správné nastavení je tedy pro fungování měniče klíčové.

V první části práce je popsáno zapojení usměrňovače a princip funkce. Následuje popis pulzní šířkové modulace potřebné pro fungování měniče. Dále je rozebrána transformace do $\alpha\beta$ a dq souřadného systému, která je potřebná pro řízení. Poté jsou rozebrány některé možné strategie řízení. Po strategiích řízení je proveden návrh regulátorů a popis modelu vytvořeného v prostředí Matlab Simulink včetně výsledků simulace.

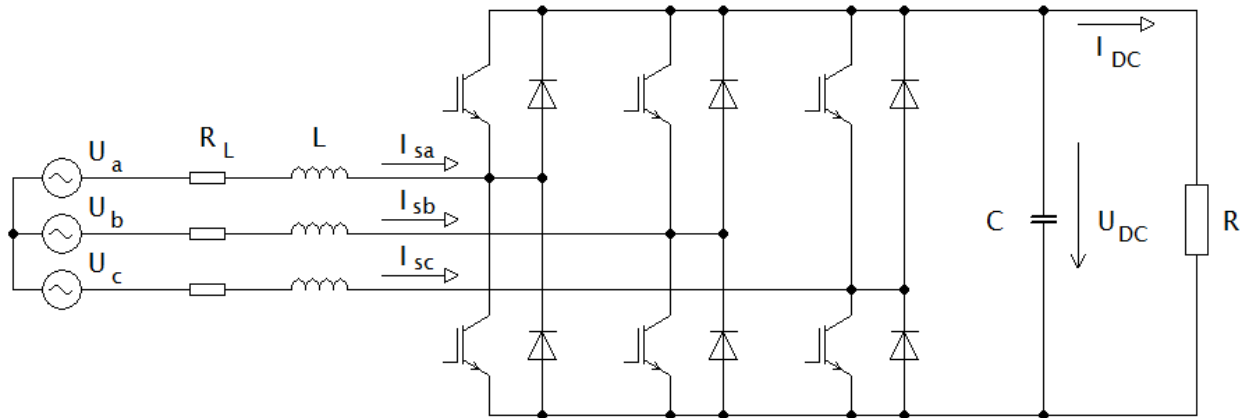
V druhé části je nejprve popsáno zařízení firmy Elcom, na kterém bylo prováděno měření. Následuje popis prostředí pro utváření kódu pro řídicí mikrokontroler typu TMS320F28335. Jeho popisem a popisem jeho periférií se zabývá další část práce. Poté je podrobně popsán vlastní kód. Na závěr je popsáno konkrétní nastavení a dosažené výsledky měření. Výstupem práce je kód pro řízení usměrňovače, který však není pro zařízení plně odladen.

2 AKTIVNÍ USMĚRŇOVAČ

Oproti běžnému diodovému usměrňovači umí aktivní usměrňovač i vracet energii zpět ze spotřebiče do zdroje. V tomto zapojení může fungovat jako usměrňovač s napětovým meziobvodem, nebo jako střídač. Velkou výhodou je možnost regulovat odebíraný proud na sinusový tvar, což je ideální stav pro síť, a při některých algoritmech řízení i vylepšovat celkový účinník sítě. Řízením je možné nastavit fázový posuv odebíraného proudu, takže usměrňovač se vůči síti může chovat jako induktivní, kapacitní i čistě odporová zátěž. [1]

2.1 Schéma

Usměrňovač se skládá ze tří větví tranzistorů a diod, vstupních tlumivek a výstupního kondenzátoru. Na vstup usměrňovače je vhodné přidat k tlumivkám ještě kondenzátory zapojené do hvězdy, pro lepší filtraci a potlačení rušení měniče do sítě. Pro spínání je možné použít tranzistory MOS-FET nebo u měničů větších výkonů tranzistory IGBT.



Obr. 1: Schéma trojfázového aktivního usměrňovače

Usměrnovač pracuje jako zvyšující měnič, takže na výstupu je stejnosměrné napětí vyšší než vstupní střídavé. Spínáním tranzistorů jsou vždy připojovány a odpojovány fáze tak, že na tlumivkách na vstupu do měniče vzniká napětí, které se přičítá ke vstupnímu pro dosažení vyššího usměrněného napětí. Pro napětí na měniči tedy platí:

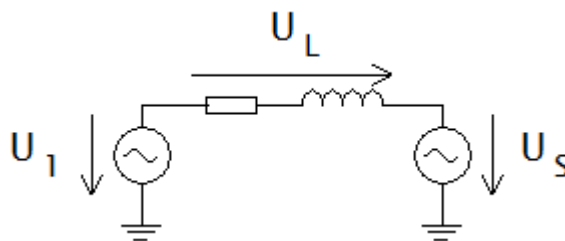
$$\bar{u}_1 = \bar{u}_L + \bar{u}_S \quad (1)$$

$$\bar{u}_1 = R * i + \frac{L * d\bar{i}}{dt} + \bar{u}_S \quad (2)$$

Pro proudy měničem pak

$$i_{sa} + i_{sb} + i_{sc} = C * \frac{du}{dt} + i_{DC} \quad (3)$$

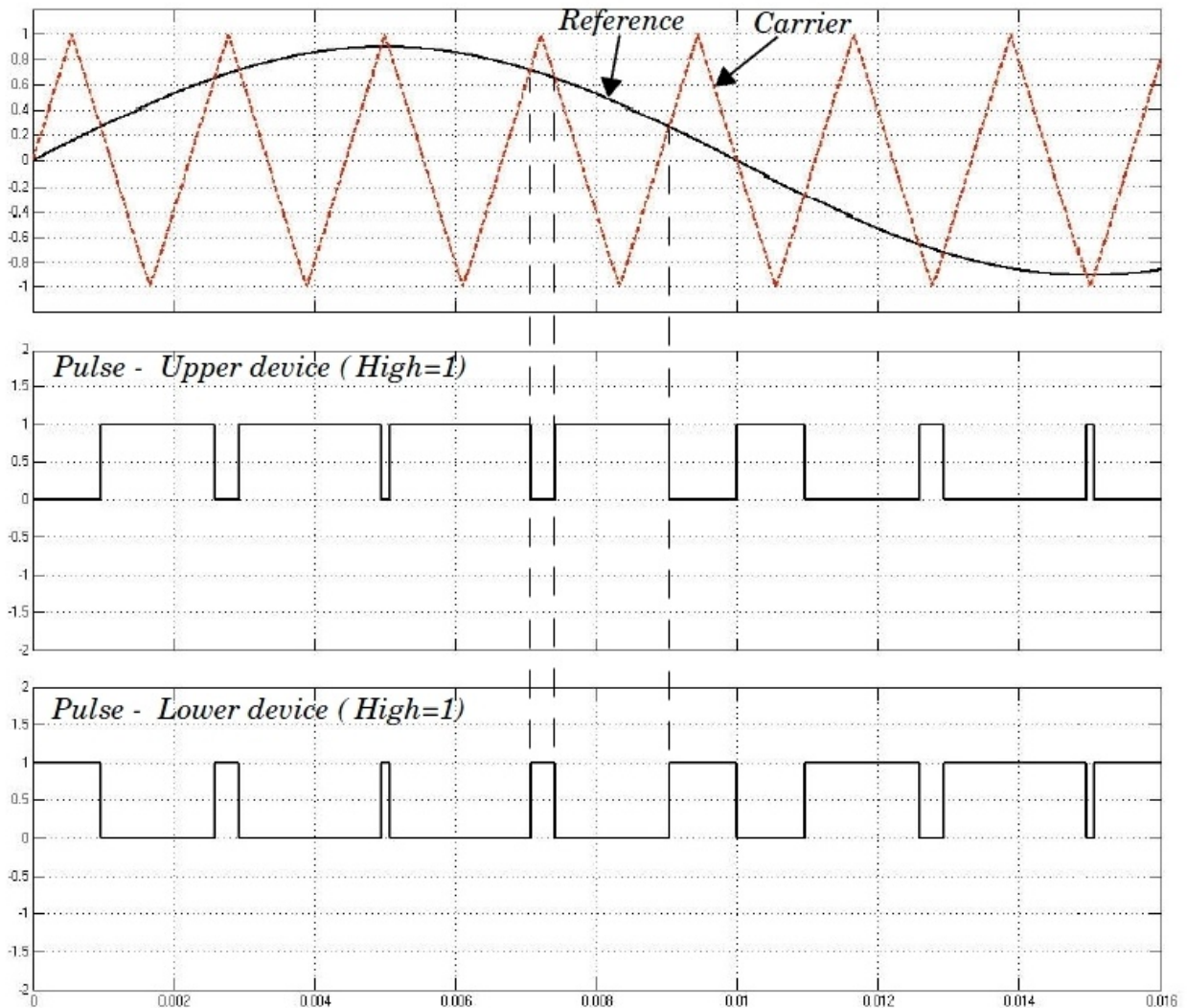
kde i_s jsou proudy vtékající do jednotlivých větví měniče a i_{DC} je proud tekoucí do zátěže.[2]



Obr. 2: Schéma náhradního jednofázového modelu usměrňovače

2.2 Pulzní šířková modulace - PWM

Řízení je prováděno pulzní šířkovou modulací signálu pro spínání tranzistorů. Ty pracují ve spínacím režimu a i řídicí signály pro ně tedy jsou buď logická 1, nebo 0. Stav řídicího signálu je dán referenčním signálem, který je porovnáván s trojúhelníkovým nosným signálem. Frekvence nosného signálu je pak i frekvencí spínání tranzistorů. Referenční signál pak můžeme chápat jako střídu spínání tranzistorů.



Obr. 3: Princip pulzní šířkové modulace, převzato z [5]

Pro každý tranzistor je řídicí signál vytvářen zvlášť podle větve kterou spíná a jeho pozici ve větvi. Tranzistory v jedné větvi nikdy nesmí být sepnuté zároveň, proto signál pro horní tranzistor dané větve a pro spodní tranzistor je stejný, pouze pro jeden z nich invertovaný. Protože tranzistory se nespínají a hlavně nevypínají nekonečně rychle, je třeba přidat do řízení odskok, aby se tranzistor ve větvi sepnul až když je druhý tranzistor stejné větve bezpečně vypnutý. [1][5]

3 TRANSFORMACE

Transformací veličin umožňujeme zjednodušení a větší stabilitu a přesnost řízení. Základní transformací je transformace třífázového systému na dvoufázový zavedením prostorového vektoru. Budeme předpokládat, že trojfázový systém je vyvážený a vektory mají stejnou velikost a jsou od sebe posunuty o $2\pi/3$ rad. Dvoufázový systém nám později zjednoduší práci, protože nám budou stačit k popisu jen dvě složky. Pro nás nejvýhodnější je transformace do $\alpha\beta$ souřadného systému, nazývaná také Clarkova. Získáme tím jediný vektor otáčející se synchronně



rychlostí s původními vektory. Clarkova transformace používá konstantu pro transformaci $C = 2/3$ a je možné ji zapsat jako

$$\begin{bmatrix} x_\alpha \\ x_\beta \end{bmatrix} = \begin{bmatrix} \frac{2}{3} & \frac{-1}{3} & \frac{-1}{3} \\ 0 & \frac{-1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \end{bmatrix} * \begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} \quad (4)$$

kde x značí obecně transformovanou veličinu a indexy a, b, c trojfázový systém.

Transformace je někdy nazývána také jako amplitudově invariantní, protože velikost prostorového vektoru je stejná jako velikost jednotlivých vektorů. Výkon je po transformaci ale menší, a aby odpovídal reálnému výkonu je potřeba jej vynásobit

$$P_{abc} = \frac{3}{2} P_{\alpha\beta} \quad (5)$$

Druhou transformací, kterou využijeme je transformace do stacionárních d - q souřadnic. Souřadnicový systém je podobný $\alpha\beta$ systému, ale celý se synchronní rychlostí otáčí. Tím se prostorový vektor nepohybuje, a má konstantní velikost. Střídavé veličiny se tak touto transformací zobrazí jako stejnosměrné, což práci s nimi usnadňuje. Pro transformaci je však nutné znát úhel prostorového vektoru, aby se souřadnicový systém otáčel stejnou rychlostí a se stejnou fází jako vektor. Transformaci pak bude možné zapsat maticí

$$\begin{bmatrix} x_d \\ x_q \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} * \begin{bmatrix} x_\alpha \\ x_\beta \end{bmatrix} \quad (6)$$

kde θ je úhel otáčejícího se prostorového vektoru v $\alpha\beta$. [4]

Pro aplikaci řízení je však potřeba převést veličiny do původních systémů. Zpětné transformace je proto možné vyjádřit jako

$$\begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{-1}{2} & \frac{-\sqrt{3}}{2} \\ \frac{-1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix} * \begin{bmatrix} x_\alpha \\ x_\beta \end{bmatrix} \quad (7)$$

$$\begin{bmatrix} x_\alpha \\ x_\beta \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} * \begin{bmatrix} x_d \\ x_q \end{bmatrix} \quad (8)$$

4 ZPŮSOBY ŘÍZENÍ

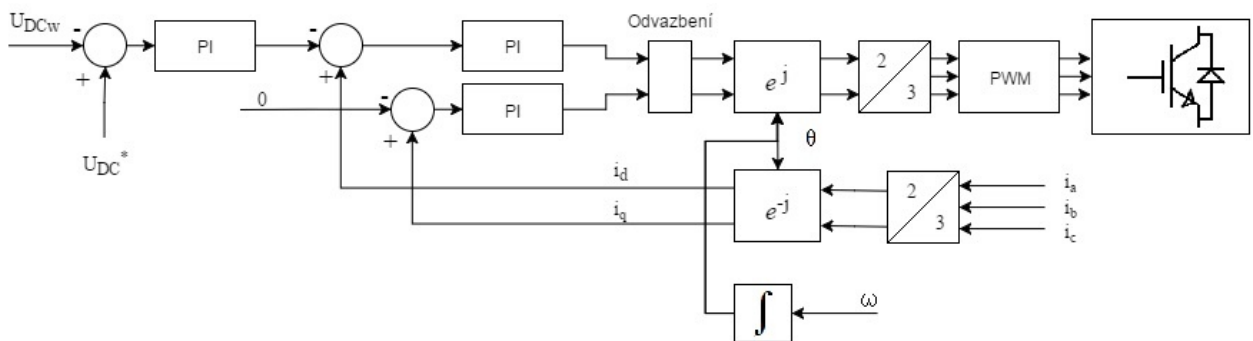
Usměrňovač je řízen v uzavřené smyčce se zpětnou vazbou. Řízení je tedy prováděno na základě požadované hodnoty a skutečných změřených hodnot. Ideální je použití PI regulátorů, které umí úplně vyrovnat regulační odchylku a jsou dostatečně rychlé pro odstranění rozdílů žádané a skutečné hodnoty. Podle měřených veličin existují různé strategie řízení.

Hlavním úkolem řízení je správné spínání tranzistorů. Kombinací jejich sepnutí se měnič dostane do jednoho stavu. Celkem je možných osm různých stavů, ve kterých se měnič může

nacházet, podle konkrétních sepnutých tranzistorů. Šest z nich jsou stavy aktivní, kdy je propojen vstup a výstup měniče, a dva stavy jsou nulové, při kterých jsou fáze střídavé části spojeny do hvězdy. Všechny strategie řízení tedy přepínají měnič mezi těmito stavy, volba stavu a čas sepnutí však závisí na konkrétní strategii.

4.1 Napětově orientované řízení

Asi nejběžnější je řízení napětí s podřízenou proudovou smyčkou a nadřazenou napětíovou. Toto řízení umožňuje udržovat výstupní napětí na požadované hodnotě a zároveň umožňuje regulovat odebíraný proud ze sítě na požadovanou maximální velikost a sinusový tvar. Díky vnější smyčce se tato strategie nazývá napětově orientovaná (VOC - voltage oriented control). Řízení je založeno na transformaci veličin do d-q souřadnic. Regulátory tedy pracují se stejnosměrnými hodnotami, což umožňuje větší přesnost regulace a rychlou reakci na změnu.



Obr. 4: Blokové schéma napětově orientovaného řízení

Vnitřní proudová smyčka je tedy tvořena snímači proudu, blokem transformace do d-q souřadnic, regulátory, blokem zpětné transformace a měničem s PWM modulací. Při transformaci získáváme dvě složky proudu na sebe kolmé. Synchronizací s reálnou a imaginární osou docílíme toho, že d osa bude reprezentovat činnou složku proudu a q osa jalovou složku. Protože přenos systému je pro obě složky stejný, budou i regulátory pro obě složky totožné. Rozdílná však bude žádaná hodnota. Výhodou je, že můžeme ovlivnit činný a jalový výkon nezávisle na sobě. Pro q složku je tedy žádaná hodnota 0, zatímco pro d složku je jako žádaná hodnota do regulátoru přiveden výstup z nadřazeného regulátoru napětí. Transformací do d-q souřadného systému dochází ke zkřížení vazeb podle následujících rovnic.

$$u_{1d} = R \cdot i_d + L \cdot \frac{di_d}{dt} + u_{sd} - \omega \cdot L \cdot i_q \quad (9)$$

$$u_{1q} = R \cdot i_q + L \cdot \frac{di_q}{dt} + u_{sq} + \omega \cdot L \cdot i_d \quad (10)$$

Proto po regulaci každé složky proudu musí dojít k odvazbení. K signálu z regulátoru je tedy přičtena resp. odečtena složka proudu vynásobená ωL . Požadovaný sinusový průběh proudu je zajištěn zpětnou transformací z dq do třífázového systému. Aby byl proud harmonický, je třeba pro transformace určit přesně a rovnoměrně úhel, jakým se soustava otáčí. Toho je dosaženo synchronizací na frekvenci sítě, nejčastěji fázovým závěsem. Nadřazená napětíová regulace pak pracuje pouze se stejnosměrnými hodnotami. Vyrovnává rozdíl měřené a požadované hodnoty stejnosměrného meziobvodu a není proto potřeba provádět žádné další transformace.



4.2 Metoda virtuálního toku

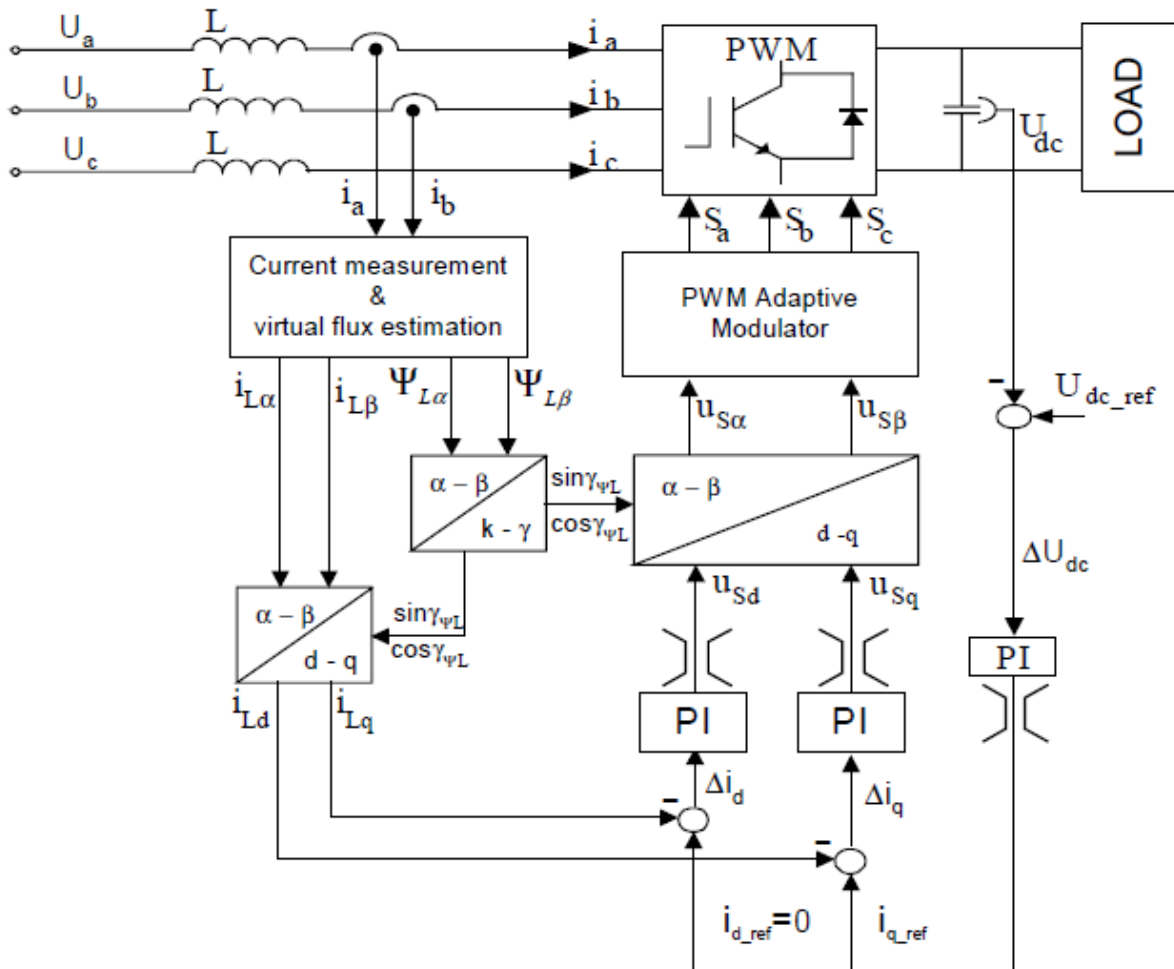
Jinou možností řízení je pomocí tzv. virtuálního toku (VF - virtual flux). Tento způsob má výhodu, že není potřeba snímat střídavé napětí, a řízení je tedy částečně bezsenzorové. Zatímco u napětěově orientovaného řízení jsme úhel pro transformace získávali z napětí, zde jej získáváme z estimovaného toku. Vstupní střídavé napětí si lze představit jako indukované a vstupní tlumivky jako rozptylové tlumivky motoru. Tento virtuální motor je pak připojen přímo na měnič. Tok tohoto motoru tedy můžeme vyjádřit jako

$$\Psi = \Psi_L + \Psi_S \quad (11)$$

$$\Psi_\alpha = \int (u_{s\alpha} + L \frac{di_{L\alpha}}{dt}) dt \quad (12)$$

$$\Psi_\beta = \int (u_{s\beta} + L \frac{di_{L\beta}}{dt}) dt \quad (13)$$

Je tedy možné jej spočítat i bez znalosti síťových napětí. Úhel toku pak je vždy o 90° posunut oproti úhlu napětí, protože vzniká jako jeho integrál. Aby řízení fungovalo, je tedy potřeba zavést osy d a q obráceně, čímž se posun úhlu vyrovná. Tím vznikne kombinace s předchozím napětěově orientovaným řízením nazývaná napětěově orientované řízení s virtuálním tokem – VF-VOC nebo VFOC



Obr. 5: Blokové schéma napěťového řízení s virtuálním tokem, převzato z [2]

Při bezsenzorovém řízení však není tak dobře zajištěna ochrana, protože nemáme v každý časový okamžik přesné informace.

Určování úhlu z virtuálního toku má ale lineárnější průběh a menší reakci na rušení v síti. To je dáno filtrem dolní propusti, který je přirozeně v každém integrátoru. [2]

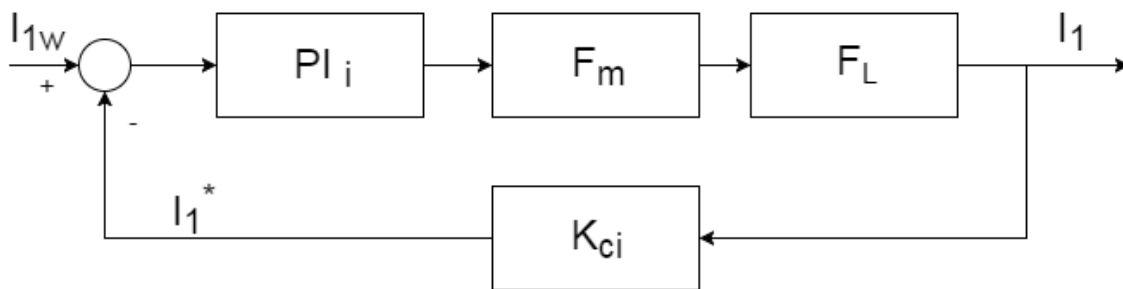
4.3 Přímé řízení výkonu

Dalším možným způsobem je přímé řízení výkonu (DPC - direct power control). Toto řízení neobsahuje smyčky proudu a je tedy ještě méně bezpečné z hlediska ochrany. Spínání tranzistorů je prováděno přímo na základě okamžité odchylky od požadovaného výkonu. Doby sepnutí jsou vyhodnoceny podle tabulky a v řízení tedy není potřeba žádný PWM modulátor. Pro tuto metodu je proto klíčové přesné určení výkonu. Tento způsob řízení však způsobuje větší rušení do sítě [3] a proto se jím zde nebudeme dále zabývat.

5 NÁVRH REGULÁTORŮ

5.1 Regulátor proudu

Regulátor navrhne podle celkového přenosu soustavy. Přenos soustavy pro vnitřní smyčku proudu je dán přenosem měniče, vlastního usměrňovače, a ve zpětné vazbě čidla proudu, podle obr.6. Regulátor navrhujeme pro přenos soustavy vyjádřený v Laplaceově transformaci.



Obr. 6: Blokové schéma vnitřní proudové smyčky

$$F_m(p) = \frac{K_m}{1+p\tau_m} = \frac{1}{1+p\frac{1}{2f}} \quad (14)$$

Přenos soustavy je dán vstupní tlumivkou, protože ta ovlivňuje odebíraný proud ze sítě. Laplaceovou transformací indukčnosti a parazitního odporu tedy získáme přenos

$$F_L = \frac{1/R_L}{p\frac{L}{R_L}+1} \quad (15)$$

Přenos čidla proudu K_{ci} je v modelu také roven jedné, bez časové konstanty, protože měření probíhá okamžitě a změřená hodnota je přivedena přímo k regulátoru.

Výsledný přenos soustavy je pak

$$F_{Si} = F_m * F_L * F_{ci} = \frac{K_m * 1/R_L * K_{ci}}{\left(1+p\frac{1}{2f}\right)\left(1+p\frac{L}{R_L}\right)} \quad (16)$$

Protože se jedná o soustavu statickou je výhodné pro návrh regulátoru použít metodu optimálního modulu. Díky tomu vznikne PI regulátor, který je rychlý a stabilní.

$$F_{Ri} = F_{OM} * \frac{1}{F_S} = \frac{1}{2\tau_\sigma p(1+p\tau_\sigma)} * \frac{\left(1+p\frac{1}{2f}\right)\left(1+p\frac{L}{R_L}\right)}{K_m * 1/R_L * K_{ci}} \quad (17)$$



$$F_{Ri} = \frac{1 + p \frac{L}{R_L}}{p K_m * \frac{1}{f * R_L} * K_{ci}} \quad (18)$$

Pro paralelní formu regulátoru pak složkový tvar bude vypadat

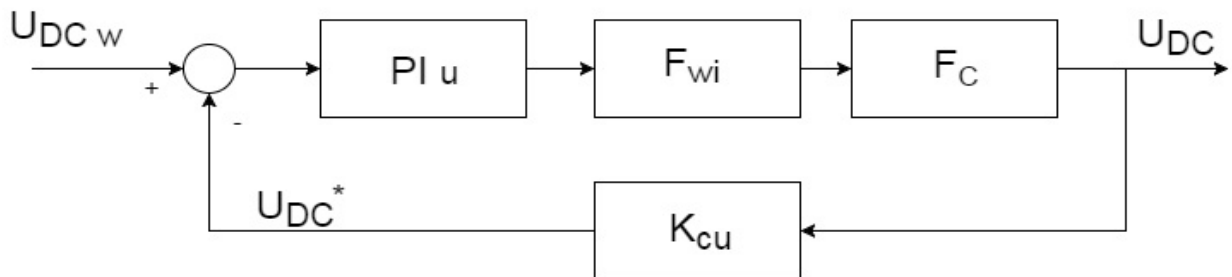
$$K_{pi} = \frac{L * f}{K_m * K_{ci}} \quad (19)$$

$$K_{ii} = \frac{R_L * f}{K_m * K_{ci}} \quad (20)$$

kde K_{pi} je proporcionální a K_{ii} integrální zesílení složek regulátoru.

5.2 Regulátor napětí

Při návrhu regulátoru nadřazené smyčky napětí postupujeme obdobně. Stanovíme přenos soustavy jako přenos smyčky proudu, čidla napětí a systému tvořeného výstupním kondenzátorem a zátěží



Obr. 7: Blokové schéma vnější napěťové smyčky

$$F_S = F_{wi} * F_{cu} * F_C \quad (21)$$

kde zjednodušený přenos smyčky proudu má tvar

$$F_{wi} = \frac{1}{1 + p \frac{1}{f}} \quad (22)$$

Přenos čidla napětí K_{cu} je roven jedné obdobně jako u proudu. A přenos kondenzátoru se zátěží je dán jako

$$F_C = \frac{R}{1 + pC} \quad (23)$$



Výsledný přenos soustavy je pak roven

$$F_{su} = \frac{K_{cu} * R}{\left(1 + p \frac{1}{f}\right) (1 + pL)} \quad (24)$$

Protože tvar přenosu soustavy je stejný, jako u vnitřní soustavy pro regulaci proudu, použijeme opět metodu optimálního modulu pro návrh regulátoru napětí.

$$F_{Ru} = \frac{(1 + pC) * f}{2p * K_{cu} * R} \quad (25)$$

Ve složkovém tvaru pak

$$K_{pu} = \frac{C * f}{2 * K_{cu}} \quad (26)$$

$$K_{iu} = \frac{f}{2 * K_{cu} * R} \quad (27)$$

6 SIMULACE

6.1 Model

Model byl vytvořen v MATLAB Simulinku. Jako napájení pro měnič byly použity čisté, nezkreslené sinusové zdroje, jako zátěž pak odpor, pro odběr pouze činného výkonu. Parametry jednotlivých prvků jsou pak v následující tabulce:

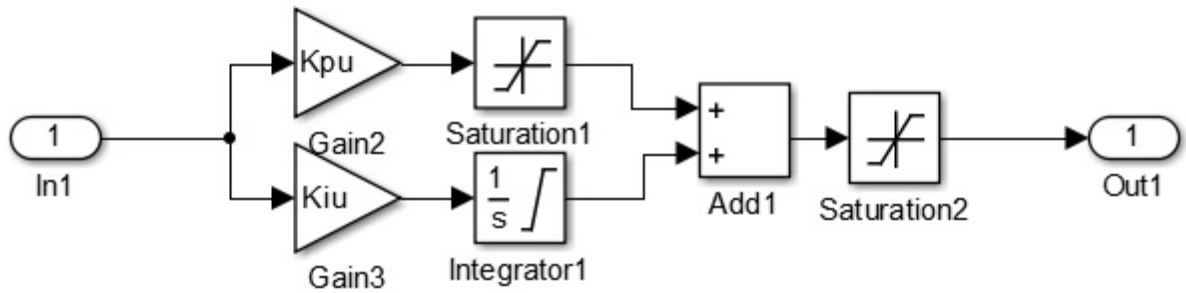
Prvek	Hodnota	Jednotka
Amplituda vstupního fázového napětí	325	V
Vstupní tlumivka	1	mH
Výstupní kondenzátor	1	mF
Druhý výstupní kondenzátor	0,5	uF
Zátěž	70	Ω
Parazitní odpor tlumivky	0,01	Ω
Požadovaná velikost usměrněného napětí	700	V
Frekvence	10	kHz

Tabulka 1: Parametry usměrňovače

Parametry jsou voleny podobné parametrům reálného přístroje a požadavkům na řízení. Model je částečně idealizován čistým napájením, činnou zátěží a zanedbáním většiny parazitních prvků. Taktéž čidla proudu a napětí byly zvoleny s ideálním přenosem.

Omezení regulátoru proudu je nastaveno na +/-1, protože PWM generátor má jako referenční signál trojúhelník také v rozmezí +/-1. [5] Omezení regulátoru napětí vychází z možností reálného zařízení a je rovno 20A.

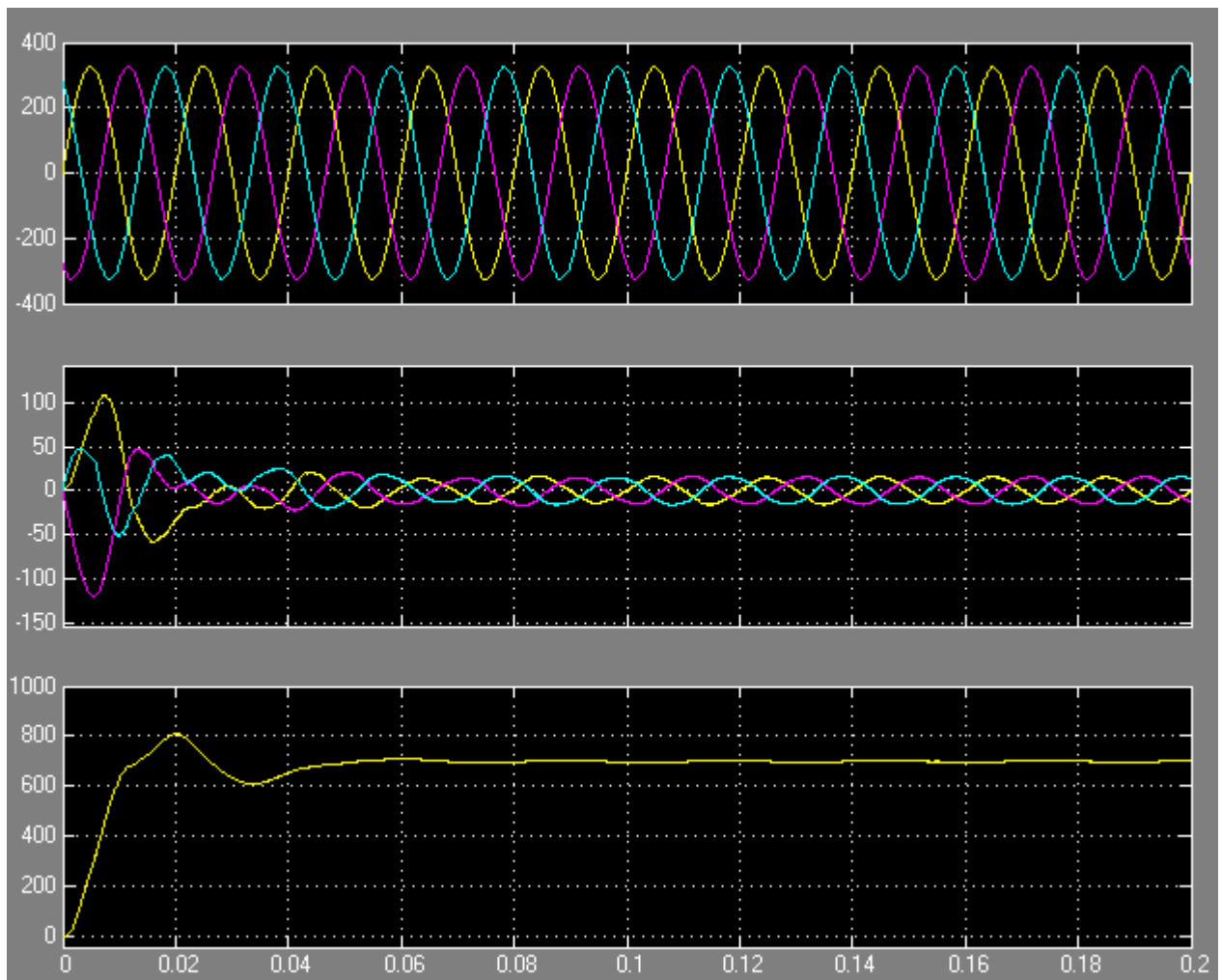
Blok PI regulátoru v napěťové smyčce byl vytvořen, protože standardní blok Matlabu špatně zvládal anti-windup. PI regulátor má klasické schéma, jak je vidět z obr.8



Obr. 8: Schéma PI regulátoru

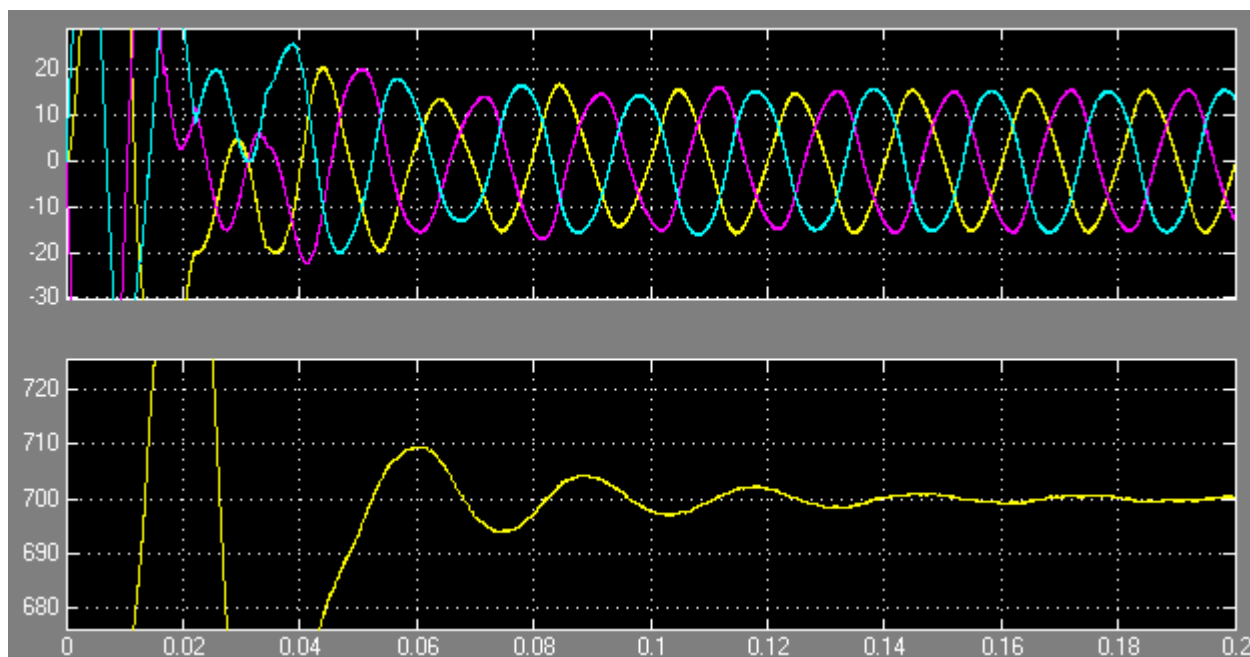
6.2 Vlastní simulace

Simulace byla provedena taktéž v MATLAB Simulinku. Čas simulace byl zvolen 0,2s, což odpovídá deseti periodám napájecího napětí. Na obrázku 9 můžeme vidět výsledné průběhy napětí a proudů.



Obr. 9: Výsledné průběhy simulace modelu. Odshora: vstupní síťové napětí, vstupní odebírané proudy, výstupní usměrněné napětí

Po počátečním nabití kondenzátoru už měnič pracuje stabilně, jak je vidět z průběhu proudů. Proudů jsou sinusové jen s malým zkreslením a napětí se postupně ustaluje na požadované hodnotě. Proud zátěží má stejný průběh jako výstupní napětí s ustálenou hodnotou 10A.



Obr. 10: Detail průběhů odebíraných proudů a výstupního napětí

7 PROSTŘEDÍ PRO REALIZACI

Tato kapitola popisuje použité zařízení a jeho vlastnosti. Zařízení bylo zapůjčeno firmou Elcom, a jedná se o testovací zařízení pro vývoj. Zařízení obsahuje více systémů, než je v této práci popsáno. Práce se zabývá pouze těmi částmi, které přímo souvisí s tématem a do kterých bylo zasahováno. Dále je popsáno programové prostředí, ve kterém bylo řízení realizováno.

7.1 Parametry zařízení

Realizace byla provedena na testovacím přístroji firmy Elcom. Toto zařízení mělo parametry rozdílné od zařízení, pro které byla prováděna simulace. Měřené zařízení je stavěno na menší výkon a jeho parametry jsou následující:

vstupní tlumivka	400 μ H	v každé fázi
vstupní kondenzátory	2x 4,7 μ F	zapojené do trojúhelníka
kondenzátory meziobvodu	6x 1 mF	sériově
vybíjecí odpory meziobvodu	5x 72 k Ω	paralelně
tranzistorový modul	PM150CL1A060	

Tabulka 2: Parametry měřeného zařízení



Použitá jednotka je TMS320F28335 Delfino microcontroller od firmy Texas Instruments(TI). Jedná se o 32 bitový signálový mikrokontroler s funkcemi zaměřenými především na řízení a regulaci pohonů spadající do rodiny kontrolerů C2000. Velká výhoda mikrokontroleru je 32 bitová harwarová jednotka pro výpočty s čísly s plovoucí řádovou čárkou. S knihovnou Fast Run-Time Support (RTS) je vhodným nástrojem pro jednoduchou, stále přesnou a rychlou implementaci řídicích algoritmů. Dále obsahuje například periférii pro A/D převodník se synchronizací na PWM modul a systém přerušení. [6]

7.3 Vývojové prostředí

Realizace byla provedena v programu Code Composer Studio (CCS) od Texas Instruments. Výrobce procesoru poskytuje pro snazší pochopení funkcí mikrokontroleru výukové CD. Na něm jsou vysvětleny postupně základní vlastnosti jednotlivých periférií a odpovídajících registrů. Ke každé kapitole pro danou periférii je uveden i příklad na procvičení a praktické ukázání funkcí a možností nastavení. Toto CD je určeno především pro univerzity pro výuku na mikrokontrolerech TI a je tedy užitečné pro uvedení do problematiky mikrokontrolerů TI.

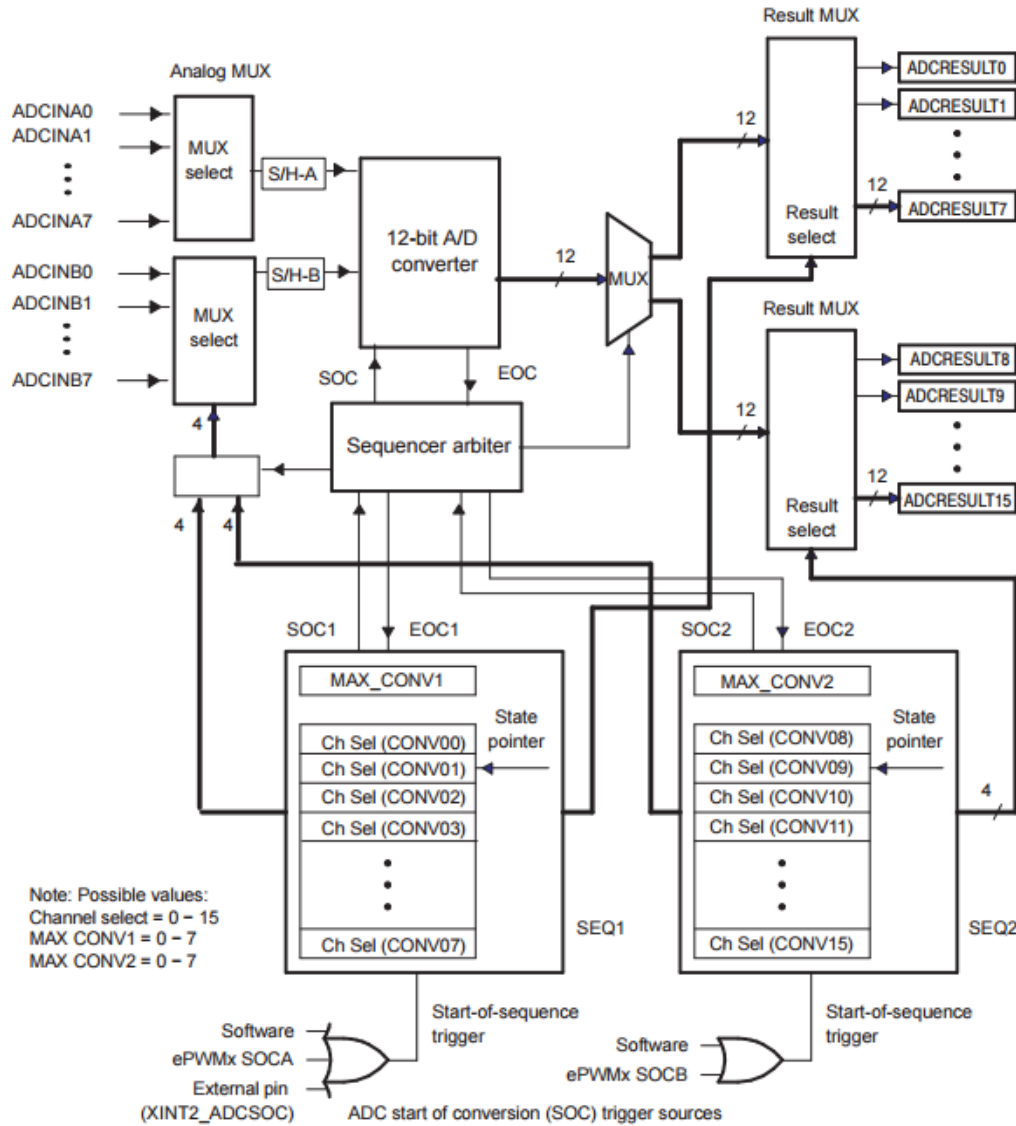
K seznámení s mikrokontrolerem firma Texas Instruments poskytuje vývojové kity. Také výukové CD má příklady koncipované na využití mikrokontroleru s daným kitem. Pro vývoj této práce byl Firmou Elcom zapůjčen také vývojový Experimenter's kit. Kit je univerzální bez konkrétního mikroprocesoru, který je možné připojit v rámci ControlCARD. Control CARD se shodným mikrokontrolerem byl taktéž firmou zapůjčen. Kit je poté připojen k počítači přes USB JTAG emulátor, které jej zároveň napájí.

7.4 Moduly mikrokontroleru

Nejdůležitější moduly mikrokontroleru jsou A/D převodník, PWM, PIE. A/D převodník pro vstup naměřených hodnot a převodu do formátu vhodného pro výpočet, PWM pro výstupní signály pro spínání tranzistorů a PIE modul pro práci s přerušením, ve kterém dochází k výpočtu.

7.4.1 A/D převodník

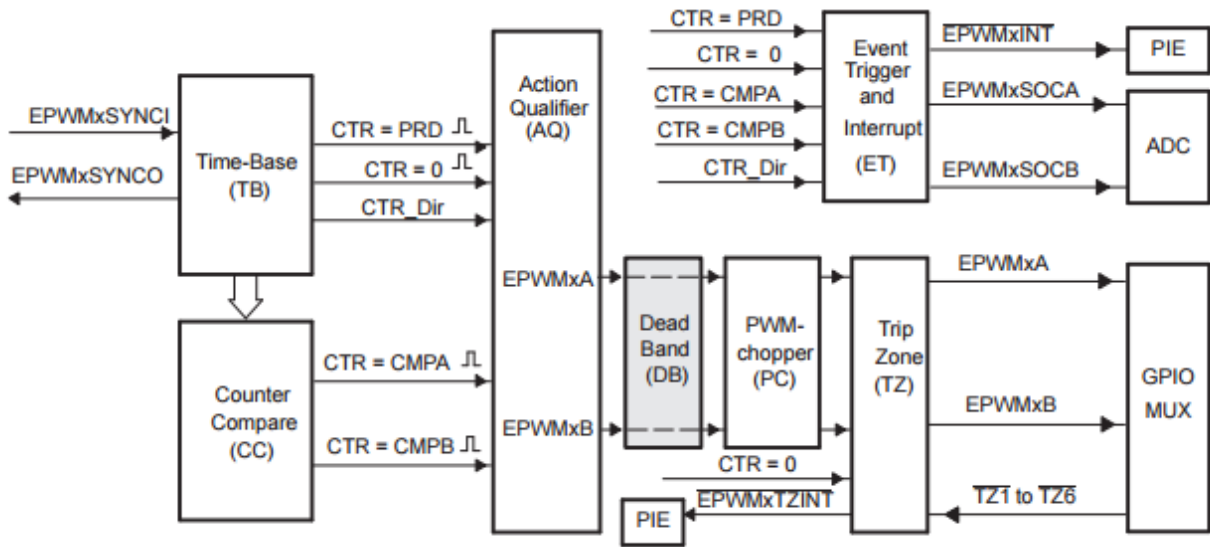
Analogově digitální převodník (ADC) přiřazuje úrovni napětí určitou hodnotu. Převodník má rozsah 0 – 3 V, které převádí na dvanáctibitové číslo. Rozsah možných naměřených hodnot je tedy 0 až 4096. Převodník má několik módů ve kterých může pracovat, podle potřeby. ADC modul má celkem 16 kanálů rozdělených do dvou multiplexerů. Každý multiplexer je pak připojen k jednomu vzorkovači a sekvenceru. Modul však má pouze jeden převodník a převody tedy probíhají postupně podle priority dané sekvencerem (nejdříve hodnota ze sekvenceru A, poté ze sekvenceru B). Modul může pracovat v sekvenčním módu, kdy jeden sekvencer postupně převádí všechny vzorky, nebo v simultánním módu, kdy oba sekvencery pracují paralelně. V paralelním módu pak jsou vždy převedeny oba vzorky ze sekvenceru A i B (např. ADCINA2 a ADCINB2). Převodník je možné spouštět signálem z PWM modulu, nastavením příslušného bitu nebo přes pin. Po spuštění dojde k načtení vzorku do vzorkovače, aby byla zajištěna neměnná hodnota v průběhu převodu. Poté je v sekvenceru nastaven celkový počet převodů a pořadí jednotlivých kanálů. Kanály mohou být nastaveny v libovolném pořadí, dokonce je možné zvolit některý kanál vícekrát. Tím dojde k větší rychlosti vzorkování – tzv. oversampling. Sekvencer řídí celý převod a výsledná čísla jsou uložena do výstupního registru v nastaveném pořadí. Na konci převodu je možné generovat přerušení a to každé, nebo každé druhé dokončení převodu.[7]



Obr. 11: Blokový diagram A/D převodníku, převzato z [7]

7.4.2 PWM modul

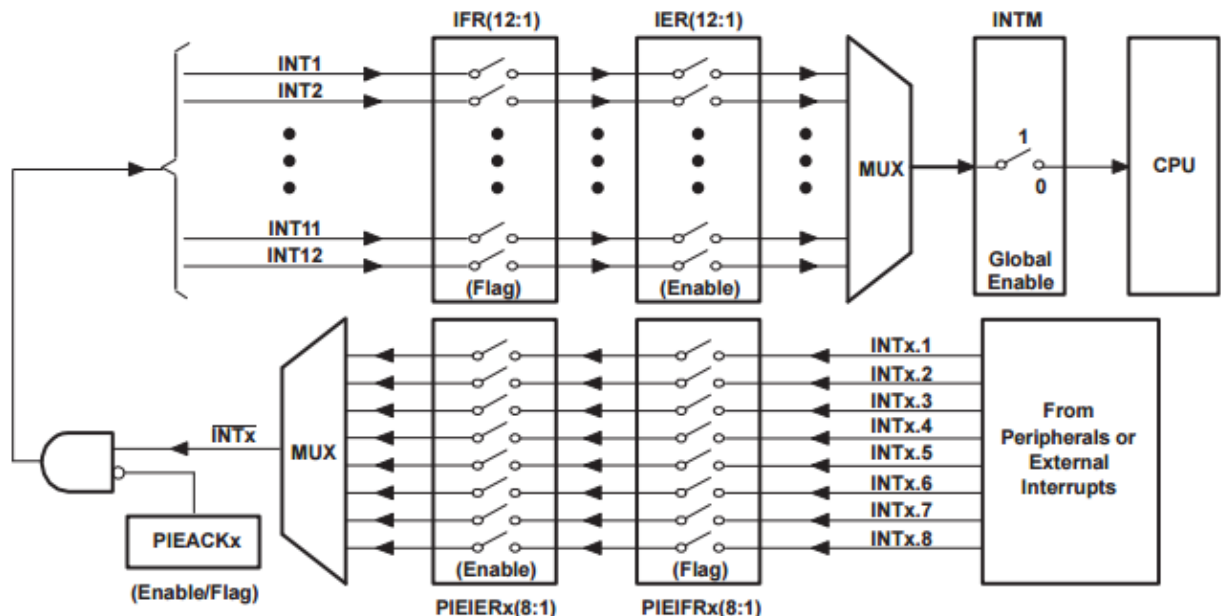
EPWM modul slouží k vytvoření pulzních signálů s co nejmenším využitím CPU a v co nejkratším čase se zachováním jednoduchosti a srozumitelnosti nastavování. Mikrokontroler má celkem 6 modulů, každý se dvěma výstupy ePWMxA, ePWMxB, kde x značí číslo modulu. Každý modul má šestnáctibitový čítač k nastavení frekvence, možnost odskoku (dead-band) s nezávislým nastavením na vzestupnou a sestupnou hranu, synchronizace na další PWM moduly s případným fázovým posuvem, možnost reakce na hlášení chyby (trip zone) a možnost vyvolání přerušeni nebo spuštění A/D převodníku. Modul má ještě další funkce, které však nejsou v této práci využity.[8]



Obr. 12: Blokový diagram ePWM modulu, převzato z [8]

7.4.3 PIE modul

Rozšíření přerušení periferií (Peripheral interrupt expansion, PIE) je modul rozlišující jednotlivá přerušení mikrokontroleru. Mikrokontroler má možnost 96 přerušení, které je nutné rozdělit do skupin pro menší počet vstupů do jádra. Modul tedy dělí přerušení po osmi a přiřazuje je jádru na vstupy INT1-INT12. Zbylá přerušení do jiných vstupů CPU modul nijak neovlivňuje. Při příchodu požadavku na přerušení z periferie je nastaven flag bit příslušného přerušení. Pokud je přerušení povoleno, je zkontrolováno, že CPU je připraveno pro přerušení dané skupiny. V případě, že CPU je přerušeno, je požadavek na přerušení předán CPU, kde probíhá obdobný proces. Místo kontroly CPU je na konci procesu kontrolováno globální povolení přerušení. Přehledně je celý proces zobrazen na obrázku 13.



Obr. 13: Diagram přerušení s použitím modulu PIE, převzato z [9]

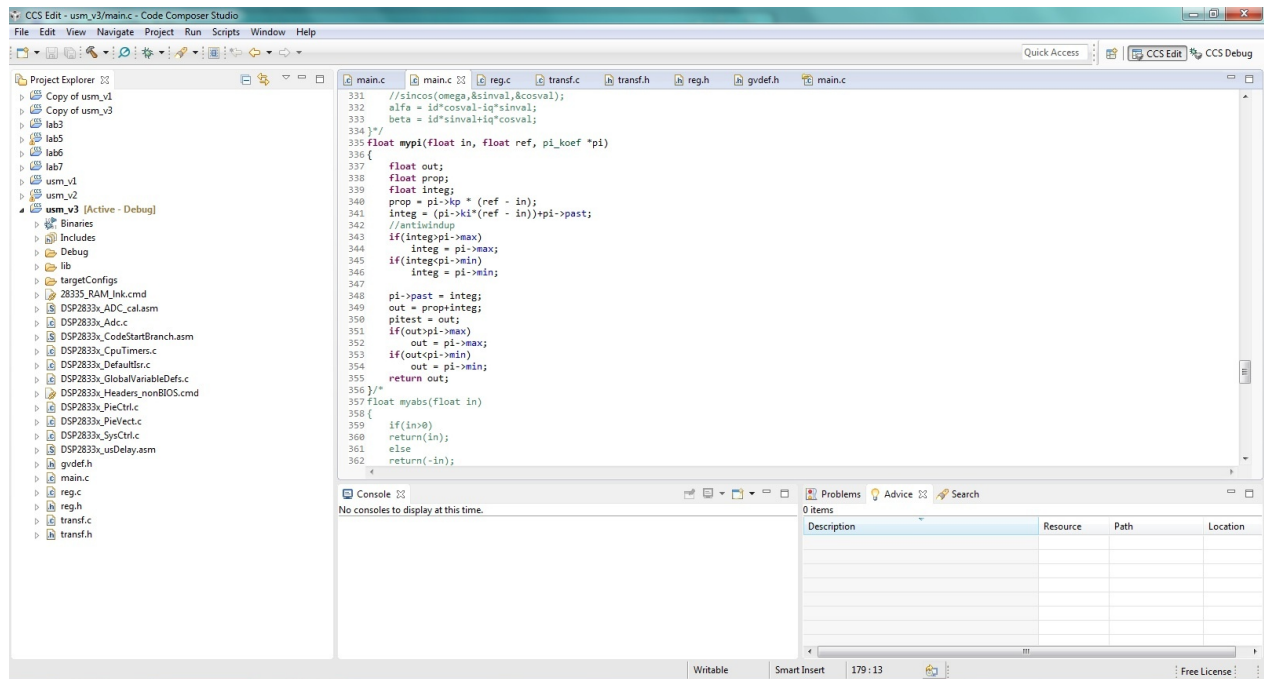


Každé z 96 přerušení je určeno svým vektorem směřujícím do tabulky uložené v paměti RAM. Při přijetí dotazu na přerušení se CPU podle vektoru podívá do tabulky, a zajistí příslušné registry. Tato tabulka musí být inicializována v kódu, a do ní je možné nastavit adresu na požadovanou část kódu, která se má v přerušení provést. [9]

7.5 Code Composer Studio

Kód pro mikrokontroler byl vytvářen v Code Composer Studiu v.6.1.2. Tento program je poskytován přímo od Texas Instruments a je založen na Eclipse 4.x. Program pracuje s jazykem C/C++ a výsledný kód kompiluje pro příslušný mikrokontroler. Ten je v každém projektu specifikován spolu s typem připojení k počítači.

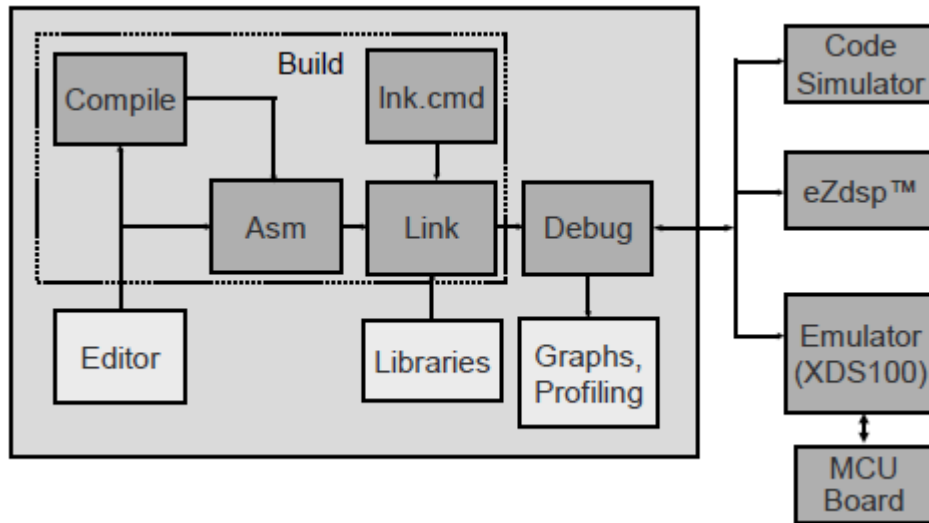
Program má hlavní okno (Workbench) rozděleno na více částí. Rozložení částí je možné měnit podle preference uživatele. V základním nastavení je zobrazen především Project Explorer a zdrojový kód.



Obr. 14: Prostředí programu Code Composer Studio

V Project Exploreru je možné vidět otevřený projekt i další projekty ve stejné pracovní ploše(Workspace). Je zde přehledně vidět, které soubory jsou součástí projektu.

V části se zdrojovým kódem je zpravidla otevřen soubor, do kterého je kód zapisován. Je možné otevřít více souborů, které se zobrazí jako karty, mezi kterými je možné přepínat. Kód je systematicky zvýrazňován podobně jako v jiných programovacích programech. Poté, co je kód hotov, je kompilován a linkován pomocí tlačítka Build. Tlačítkem Debug je pak nahrán do cílového mikrokontroleru, nebo simulátoru. Po tomto kroku se přepnou části okna do Debug perspektivy, kdy jsou zobrazeny především funkce pro testování běhu programu.



Obr. 15: Blokové schéma realizace kódu, převzato z [10]

V Debug perspektivě je možné v tabulce sledovat hodnoty proměnných a registrů. Pokud se při běhu nějaká hodnota změní, je příslušný řádek tabulky zvýrazněn. Hodnoty jsou aktualizovány v určitém časovém intervalu, a nemusí stíhat zobrazovat velmi rychlé změny hodnot. Druhou možností sledování je zobrazení hodnot do grafu. Graf ukazuje průběh hodnot v čase, ale vychází z načtených hodnot. Prudkou změnu tedy také nemusí zachytit správně.[10]

Důležitou částí programu pro tuto práci bylo zahrnutí knihovny Fast RTS, která obsahuje optimalizované funkce pro výpočty s čísly s pohyblivou řádovou čárkou. Tato knihovna je velmi užitečná pro aplikace vyžadující co nejkratší čas výpočtu. Například funkce sinus s touto knihovnou trvá pouze přibližně 40 cyklů, zatímco bez knihovny okolo 500.[11]

8 KÓD

Řízení bylo vytvořeno jako napěťově orientované. Tento typ řízení byl odsimulován a představuje stabilní a bezpečnou variantu. Vlastní regulace běží v přerušení vyvolaném modulem PWM pro převodník. Přerušení je vyvoláno vždy když je hodnota PWM modulu nosného signálu („trojúhelníku“) rovna nule. Na začátku přerušení dojde k převedení měřených veličin na čísla. Měřené veličiny jsou dvě sdružená napětí, dva fázové proudy a stejnosměrné napětí meziobvodu. Po převedení je dopočítán zbylý proud a jsou vypočtena fázová napětí. Z fázového napětí je pomocí fázového závěsu získána frekvence a úhel potřebný pro transformace.

Protože měření probíhalo na testovacím zařízení, byl pro ně vytvořen druhý model, který odpovídá jak zařízení, tak i vytvořenému kódu. Model má obdobné průběhy veličin, jako model s původními parametry.

8.1 Použité externí funkce

V kódu byly použity funkce od TI pro inicializaci periférií a funkce firmy Elcom pro nastavení mikrokontroleru pro práci se zařízením. Použité funkce jsou:

InitSysCtrl() - funkce k inicializaci hlavních prvků systému, jako jsou hodiny a paměť flash. Funkce zakazuje watch dog, který je v kódu znovu povolen.

InitPieCtrl() - funkce nuluje nastavení v registrech PIE. Požadovaná přerušení včetně tabulky vektorů je poté nutné nastavit ručně.

InitPieVectTable() - funkce přemapovává vektory pro použití modulem PIE

InitAdc() - funkce slouží ke kalibraci převodníku

BIOS_PinInit() - funkce nastavující piny mikrokontroleru

CPU_InstallRamFuncs() - funkce, která kopíruje některé funkce do paměti RAM

CPU_InitFlash() - slouží pro inicializaci flash paměti

CPU_StdInitXintClk() - slouží pro nastavení hodin externí sběrnice

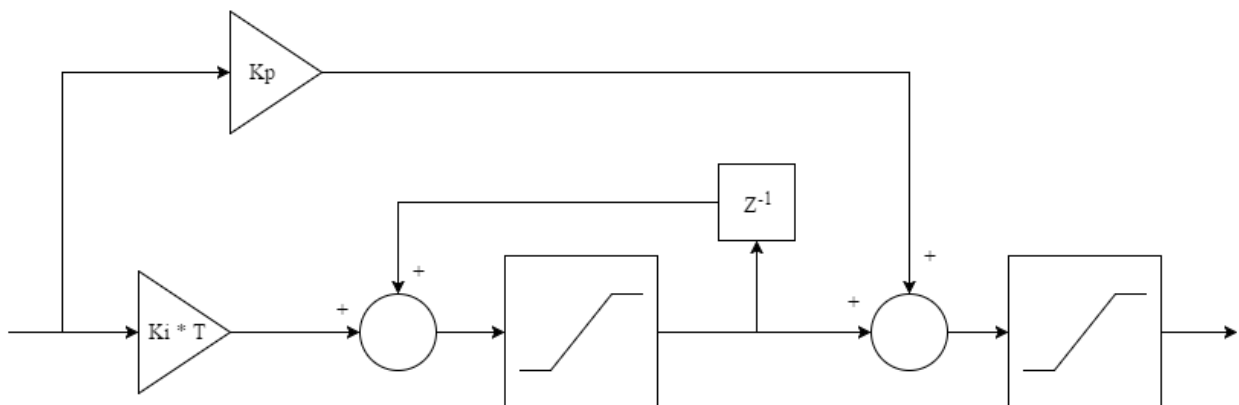
CPU_StdInitXintf0() - slouží k nastavení wait stavů externí sběrnice

8.2 Vlastní funkce

Při realizaci regulace byly vytvořeny pro jednoduchost kódu vlastní funkce. Tyto funkce vychází z teorie výpočtu a ze simulovaného modelu. Funkce byly rozděleny do dvou souborů pro větší přehlednost. Soubor transf.c obsahuje funkce týkající se transformací mezi souřadnými systémy, soubor reg.c pak obsahuje zbylé funkce přímo související s regulací. Pro některé funkce pak byl vytvořen soubor gvdef.h s definicemi struktur a s globálními proměnnými, které tyto funkce využívají.

8.2.1 PI Regulátor

Funkce realizuje paralelní PI regulátor jednoduchou formou podle schématu:



Obr. 16: Blokové schéma PI regulátoru



Rozdíl vstupu a výstupu je zesílen proporcionálním koeficientem a přičten k integrační složce. V té je rozdíl také zesílen integračním koeficientem, a integrován. Výsledný součet je omezen, aby regulátor nenastavoval příliš vysokou hodnotu. V integrační složce je navíc přidáno další omezení, které realizuje tzv. antiwindup, tedy to, že i když je výstup regulátoru v saturaci, integrační složka se dále nezvětšuje a při snížení hodnoty je schopná hned reagovat.

```
float mypi2(float in, float ref, pi_koef *pi)
{
    float out;
    float prop;
    float integ;
    prop = pi->kp * (ref - in);

    integ = (0.0001 * (pi->ki * (ref - in))) + pi->past;
    //antiwindup
    if(integ > pi->max)
        integ = pi->max;
    if(integ < pi->min)
        integ = pi->min;

    pi->past = integ;

    out = prop + integ;
    if(out > pi->max)
        out = pi->max;
    if(out < pi->min)
        out = pi->min;
    return out;
}
```

Vstupní struktura pi_koef je struktura definující konkrétní regulátor. Obsahuje koeficienty zesílení, maximum a minimum pro omezení a proměnnou pro zpožděnou hodnotu pro výpočet integrační složky.

8.2.2 Dolní propust

Dolní propust je realizována vztahem určeným diskretizací přenosové funkce. Přenosová funkce má podobu

$$F(p) = \frac{Y(p)}{U(p)} = \frac{1}{\tau p + 1} \quad (28)$$

kde τ je časová konstanta. Dosazením dopředné Eulerovy diskretizace a následným vyjádřením výstupu dostáváme výsledný vztah pro výpočet. [12]

$$p = \frac{1 - z^{-1}}{z^{-1} T_s} \quad (29)$$

$$y(k) = y(k+1) + C_{lp} * (u(k+1) - y(k+1)) \quad (30)$$

Časová konstanta byla nastavena pro mezní frekvenci 0,5 Hz. Tato frekvence bude dobře filtrovat součtovou složku dolního závěsu. Konstanta C_{lp} pro výpočet dolní propusti je dána poměrem vzorkovací frekvence a časové konstanty. [13]

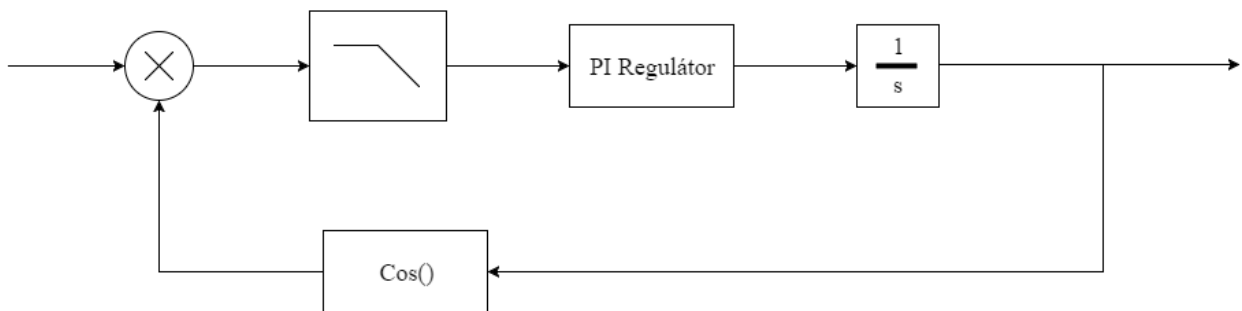


```
float mylp(float in)
{
float out;
out = pastlp+0.0003141097102*(pastin-pastlp);
pastlp = out;
pastin = in;
return out;
}
```

Proměnné „pastlp“ a „pastin“ jsou globální a uchovávají předchozí prvky.

8.2.3 Fázový závěs

Fázový závěs je realizován jako součinnový, který je vhodný pro interpretaci v kódu.



Obr. 17: Blokové schéma fázového závěsu

Princip regulátoru vychází ze vztahu

$$\sin \alpha * \cos \beta = \frac{\sin(\alpha + \beta)}{2} + \frac{\sin(\alpha - \beta)}{2} \quad (31)$$

kdy vynásobením získáváme signál o rozdílové frekvenci a na něm nasuperponovaný signál o součtové frekvenci. Signál o vysoké frekvenci je filtrován dolní propustí, a zbylý signál je přiveden na vstup PI regulátoru. Signál z regulátoru tak odpovídá frekvenci, jejíž integrací získáváme výraz ωt . Požadovaný úhel pak získáme jako modulo 2π . Kosinus tohoto úhlu je přiveden zpětnovazebně na vstup fázového závěsu. Pro rychlejší zavěšení na vstupní signál, kterým je fázové napětí, je regulátor omezen na frekvence přibližně od 47,8 Hz do 52,5 Hz. [14]

```
float mypll(float ua)
{
pll.in = mylp(ua*1*pll.ref);
freq = mypi2(-pll.in, 0, &piw);
pll.out = freq + pll.past;
if(pll.out >= (2 * 3.1415926))
    pll.out = 0;
pll.ref = cos(pll.out);
pll.past = pll.out;
return pll.out;
}
```



Funkce pracuje se speciálně vytvořenou strukturou. Struktura obsahuje proměnné in, out, past a ref, které jsou použity při výpočtu. Dále funkce používá funkci dolní propust a PI regulátor.

8.2.4 Kruhové omezení

Kruhové omezení slouží k regulaci alfa a beta složky výstupu regulátoru. Pokud je jejich vektorový součet větší než 1, je potřeba jej zmenšit, aby do PWM modulace vstupovaly hodnoty v rozmezí 0 až 1. Omezení je provedeno vydělením koeficientem, představujícím přesah přes 1. Výpočet vychází ze vzorce

$$\sin^2 + \cos^2 = 1 \quad (32)$$

Koeficientem je potřeba vydělit obě složky, aby byl zachován jejich poměr.[13]

```
void mykruhlim()
{
    float koef;
    koef = sqrt( alfa * alfa + beta * beta);
    if( koef > 1 )
    {
        alfa = alfa / koef;
        beta = beta / koef;
    }
}
```

8.2.5 Clarkova transformace

Oproti transformaci uvedené výše má funkce zjednodušený výpočet. Výstupem je opět alfa a beta, ale oproti výše uvedenému výpočtu je beta inverzní, jakoby vynásobená -1. Tím se vektor točí opačným směrem než by se točil po původní transformaci. Vzhledem k tomu, že transformace zjednodušeným výpočtem je počítána stejně, jako v knihovně TI, je použita v modelu i v kódu.[15]

```
void myclarke()
{
    alfa = ia;
    beta = (2*ib+ia)*0.5773502692;
}
```

Funkce upravuje globální proměnné se kterými je dále v kódu pracováno.

8.2.6 Parkova transformace

Výpočet parkovy transformace odpovídá teorii uvedené výše. V kombinaci s použitou Clarkovou transformací vychází id téměř nulové a iq záporné. Toto je třeba zohlednit při výpočtu regulace i při zpětné transformaci.

```
void mypark()
{
    float sinval, cosval;
    sincos(omega,&sinval,&cosval);
    id = alfa*cosval+beta*sinval;
    iq = -alfa*sinval+beta*cosval;
}
```



Ve funkci je použita funkce sincos z knihovny FastRTS pro rychlý výpočet sinu i kosinu zároveň. Funkce je rychlejší než samostatné volání funkcí pro jednotlivé výpočty. Proměnné iq, id, alfa, beta jsou globální, tato funkce je přepisuje.

8.2.7 Inverzní Parkova transformace

Aby regulace fungovala, je použita transformace podle teorie výše. Tím je zajištěno, že výstupy transformace budou mít správnou hodnotu. Pokud použijeme Parkovu a inverzní Parkovu transformaci ihned za sebou, dostáváme na výstupu stejný signál, jaký do transformací vstupuje.

```
void myipark()
{
    float sinval, cosval;
    sincos(omega,&sinval,&cosval);
    alfa = id*cosval-iq*sinval;
    beta = id*sinval+iq*cosval;
}
```

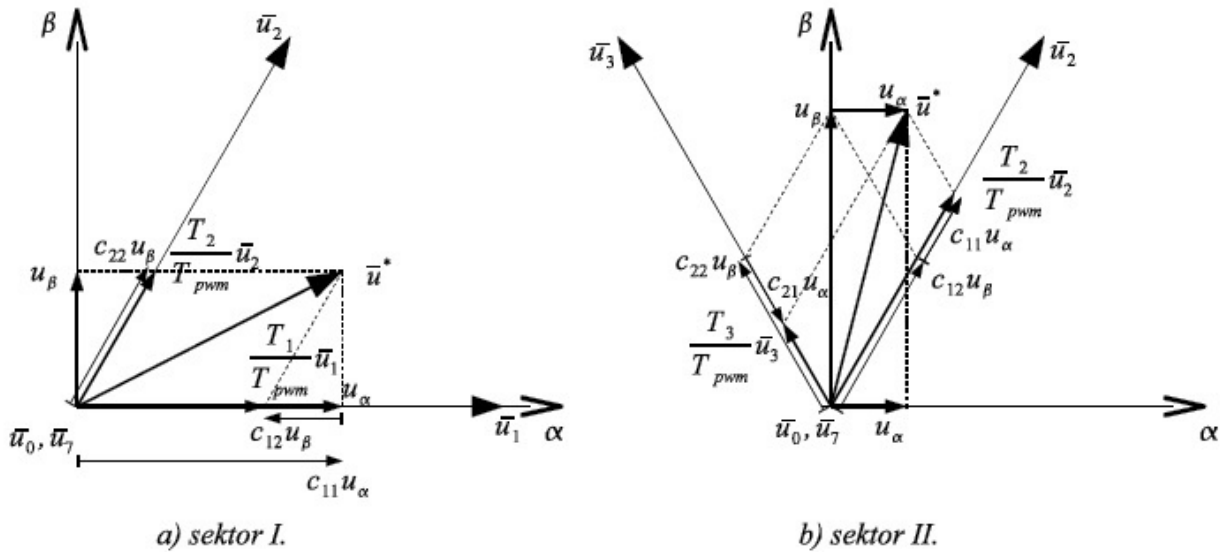
8.2.8 Absolutní hodnota

Funkce vrací absolutní hodnotu vstupu. Je možné, že tato funkce v některé z knihoven už existuje, ale vzhledem k jednoduchosti jsem si ji také vytvořil.

```
float myabs(float in)
{
    if(in>0)
        return(in);
    else
        return(-in);
}
```

8.2.9 Modulace prostorového vektoru (SVM)

Způsob modulace prostorového vektoru umožňuje měnič pracovat s činitelem modulace 1. Umožňuje tedy lepší využití napětí. Princip modulace je založen na výpočtu času po který jsou jednotlivé tranzistory sepnuty. Toho je dosaženo určením sektoru, ve kterém se prostorový vektor nachází. Sektory jsou určeny jednou z šesti kombinací sepnutí. Zbylé dvě možné kombinace jsou nulové.



Obr. 18: Grafické znázornění výpočtu svm, převzato z [13]

Pokud se vektor nachází v určitém sektoru, jsou spínány tranzistory odpovídající stavům krajních vektorů a nulových vektorů. Při přepínání jednotlivých vektorů je nevhodnější přepínat vždy jen jeden tranzistor. Tím dochází k nejmenším přepínacím ztrátám. Nulové vektory jsou navíc společné pro všechny sektory a je tedy snadné přes ně do jiného sektoru přejít. Čas sepnutí daného vektoru je určen poměrem vzdálenosti požadovaného vektoru napětí mezi stavy ohraničujícími daný sektor. Velikost požadovaného vektoru pak určuje poměr nulových vektorů. [13]

```

void mysvm(float ualpha, float ubeta)
{
    unsigned int m;
    float Tm, Tm1, T0;
    if (ubeta > 0) //volba sektoru
    {
        if (myabs(ualpha) > (myabs(ubeta)*0.5773502692))
        {
            if (ualpha > 0)
                m = 1;
            else
                m = 3;
        }
        else
            m = 2;
    }
    else
    {
        if (myabs(ualpha) > (myabs(ubeta)*0.5773502692))
        {
            if (ualpha > 0)
                m = 6;
            else
                m = 4;
        }
    }
}
    
```



```
else

    m = 5;
}
Tm = ualpha*c11[m-1] + ubeta*c12[m-1]; //výpočet časů stavů
Tm1 = ualpha*c21[m-1] + ubeta*c22[m-1];
T0 = 0.5*(1-Tm-Tm1);
//výpočet střídý
switch (m)
{
    case 1:
        uva = T0 + Tm + Tm1;
        uvb = T0 + Tm1;
        uvc = (T0);
        break;
    case 2:
        uva = T0 + Tm;
        uvb = (T0 + Tm + Tm1);
        uvc = (T0);
        break;
    case 3:
        uva = T0;
        uvb = (T0 + Tm + Tm1);
        uvc = (T0 + Tm1);
        break;
    case 4:
        uva = T0;
        uvb = (T0+Tm);
        uvc = (T0+Tm+Tm1);
        break;
    case 5:
        uva = (T0 + Tm1);
        uvb = (T0);
        uvc = (T0+Tm+Tm1);
        break;
    case 6:
        uva = (T0+Tm+Tm1);
        uvb = (T0);
        uvc = (T0+Tm);
        break;
}
}
```

Určení sektoru je prováděno z velikosti vstupní alfa a beta složky. A konstanty c11-c22 jsou definovány jako

```
float c11[6] = {0.8660254038,0.8660254038,0,-0.8660254038,-0.8660254038,0 };
float c12[6] = {-0.5,0.5,1,0.5,-0.5,-1};
float c21[6] = {0,-0.8660254038,-0.8660254038,0,0.8660254038,0.8660254038};
float c22[6] = {1,0.5,-0.5,-1,-0.5,0.5};
```



8.3 Další výpočty

Na zařízení jsou měřena dvě vstupní sdružená napětí. Ve výpočtu je však počítáno s fázovými napětími, která je nutné počítat. Výpočet probíhá v přerušení ihned po převedení hodnot. Vztahy pro výpočet vychází z rovnic pro určení sdružených napětí z fázových.

$$\begin{aligned}fUu &= (2 * fUuv + fUvw) * 0.333333333 ; \\fUv &= (fUvw - fUuv) * 0.333333333 ; \\fUw &= - fUu - fUv ;\end{aligned}$$

Další provedený výpočet byl přepočítání hodnot z regulátorů na hodnoty pro vstup do PWM registrů. Přiřazení přímo výstupů z registrů bez vektorové modulace je pouze pro zkušební běh a v odladěném přístroji nebude použito. Protože výstupy z regulátorů mohou nabývat hodnot od -1 do 1, je potřeba hodnoty přepočítat na rozmezí 0 až 1. Dále je třeba hodnoty vynásobit maximem, kterého může vstupní registr nabývat, aby střída mohla dosáhnout plného rozsahu hodnot.

$$\begin{aligned}uva &= uva * 0.5 + 0.5 ; \\uvb &= uvb * 0.5 + 0.5 ; \\uvc &= uvc * 0.5 + 0.5 ;\end{aligned}$$

```
//prepocet a zapis do pwm CMPA
EPwm1Regs.CMPA.half.CMPA = (unsigned int) floor(uva * 3750);
EPwm2Regs.CMPA.half.CMPA = (unsigned int) floor(uvb * 3750);
EPwm3Regs.CMPA.half.CMPA = (unsigned int) floor(uvc * 3750);
```

8.4 Nastavení modulů

8.4.1 Nastavení ADC

Nastavení bylo začato vynulováním kontrolních registrů ADCTRL1 a ADCTRL2. Po vynulování kontrolních registrů jsou znovu nastaveny jen požadované bity. V prvním kontrolním registru je nastaven bit CONT_RUN, který nechá převodník proběhnout jednou a pak čeká na další signál pro spuštění, bit ACQ_PS, který nastavuje délku okna, bit CPS, který označuje předdělení hodin a bit SEQ_CASC, který označuje kaskádní mód převodníku. V kaskádním módu jsou všechny požadované vstupy převedeny postupně. V druhém kontrolním registru je nastaveno povolení startu převodníku podle signálu z PWM modulu a nastaveno povolení přerušení po každém převodu. Ve třetím kontrolním registru je nastaveno pouze další dělení hodin.

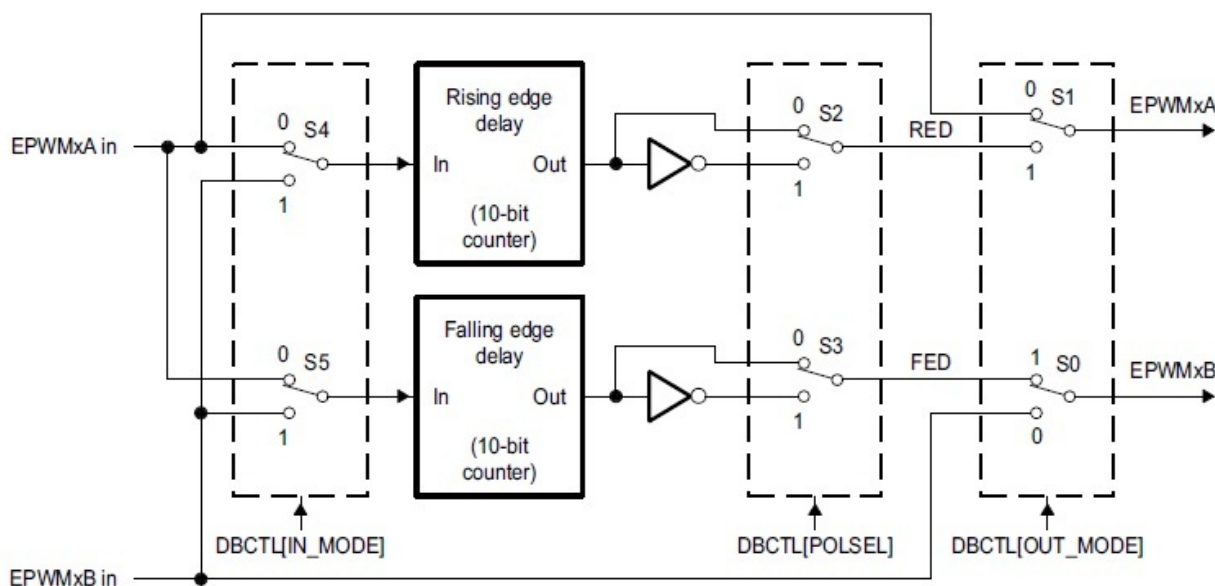
Celkový počet převodů je nastaven na 5, tedy registr ADCMAXCONV.all = 4. Poté jsou přiřazeny požadované vstupy podle zapojení zařízení.

Na konci přerušení bylo potřeba resetovat sekvencer a smazat mu příznak, aby mohl v dalším přerušení znovu naběhnout. [7]

8.4.2 Nastavení ePWM

Nastavení ePWM bylo provedeno pro tři moduly. Každý obsluhuje oba tranzistory jedné větve měniče a jejich nastavení je stejné. Nejdříve nastavený registr byl TBCTL určující hodiny pro modul a tvar nosného signálu. Základním registrem spolu s předchozím je registr TBPRD, jehož hodnota spolu s nastavením hodin určuje frekvenci nosného signálu a tím i frekvenci spínání. Pro požadovaných 10 kHz byl tedy registr nastaven na hodnotu 3750. V registru AQCTLA bylo nastaveno symetrické překlápění.

Dále byla nastavena část určující odskok. Ten je počítán z jednoho signálu pro oba tranzistory větve. Odskok byl nastaven na 3 μ s.



Obr. 19: Schéma nastavení odskoku ePWM modulu, převzato z [8]

Odskok je nastavením registru DBCTL nastaven pro oba tranzistory, pro jeden při náběžné hraně vstupního signálu, pro druhý při sestupné hraně.

Dalším důležitým submodule je Trip zone, který slouží k vypnutí tranzistorů při indikaci chyby. Pokud na vstupu dostane signál, tranzistory vypne a nechá je vypnuté do dalšího restartu.

Nakonec je nastaveno posílání signálu pro start A/D převodníku vždy, když je nosný signál roven nule.[8]

8.4.3 Nastavení PIE

Pro modul PIE byla přidána pouze funkce přerušení do tabulky, a příslušné přerušení bylo povoleno. Na konci přerušení bylo opět potřeba smazat příznak přerušení, aby mohlo být znovu spuštěno.[9]



9 MEŘENÍ

Měření probíhalo na testovacím zařízení firmy Elcom. Zařízení bylo připojeno přes autotransfornátor na trojfázovou síť a napájení řídicích obvodů bylo provedeno ze stolního zdroje 24 V.

9.1 Nastavení proměnných

Pro fázový závěs byla vytvořena struktura „pll_koef“ a proměnná typu této struktury nazvaná „pll“ s následujícími hodnotami

```
pll.in = 0;  
pll.out = 0;  
pll.past = 0;  
pll.ref = 1;
```

Protože funkce obsahuje také PI regulátor, byla i pro něj vytvořen vlastní typ struktury „pi_koef“ a proměnná tohoto typu „piw“ s parametry:

```
piw.ki = 0.0053;  
piw.kp = 0.0379;  
piw.max = 0.033;  
piw.min = 0.03;  
piw.past = 0;
```

Kde hodnoty minima a maxima odpovídají požadovanému omezení frekvence. Hodnoty proporcionálního a integračního zesílení byly vypočteny Matlabem, a vyděleny frekvencí, aby odpovídaly postupu výpočtu.

Do funkce fázového závěsu pak bylo jako parametr použito vypočtené fázové napětí.

Pro regulátory proudu byly vytvořeny proměnné typu struktury pi_koef se shodnými parametry v ose q i d. Do nastavení koeficientů zesílení byly zvoleny malé hodnoty, pro postupné ladění na zařízení.

```
pii_d.ki = 0.001;  
pii_d.kp = 1;  
pii_d.max = 1;  
pii_d.min = -1;  
pii_d.past = 0;
```

```
pii_q.ki = 0.001;  
pii_q.kp = 1;  
pii_q.max = 1;  
pii_q.min = -1;  
pii_q.past = 0;
```

Obdobně pak byla vytvořena i proměnná pro napěťový regulátor

```
piu.ki = 71.4286;  
piu.kp = 0.1429;  
piu.max = 15;  
piu.min = -15;  
piu.past = 0;
```

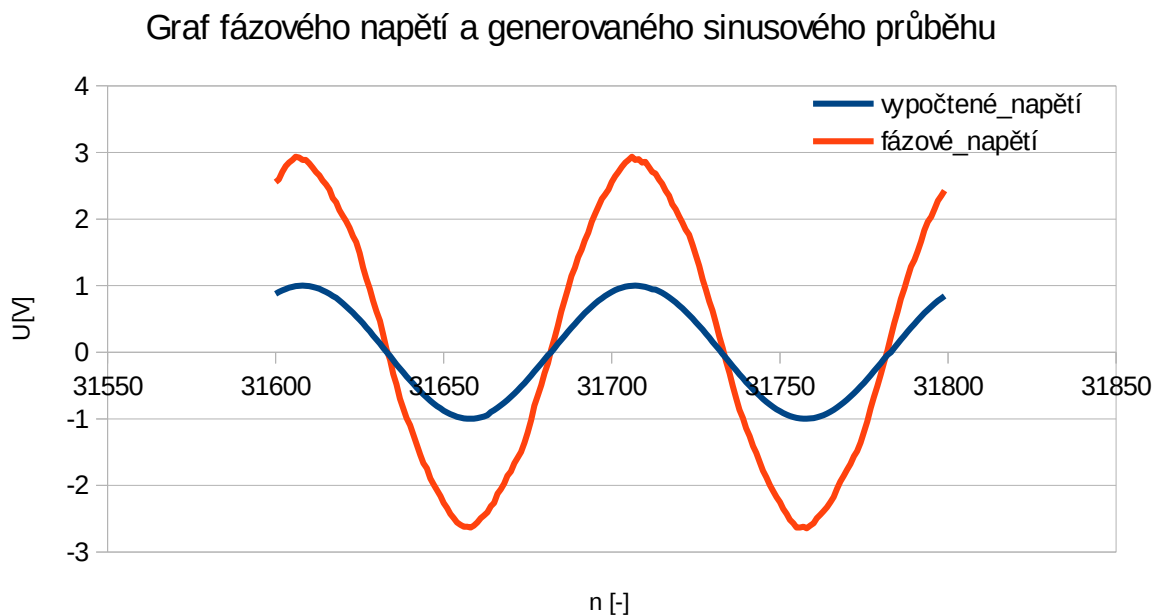



9.2 Výsledky měření

Při měření byly sledovány proměnné, které byly ukládány do polí, a zobrazovány v grafu Code Composer Studia přímo za běhu programu. Pro lepší přehlednost byly hodnoty z grafů exportovány a znovu zobrazeny v grafu pomocí LibreOffice. K ukládání jsem použil funkce ing. Tomáše Žúrka, který je měl zpracované pro snadné použití.

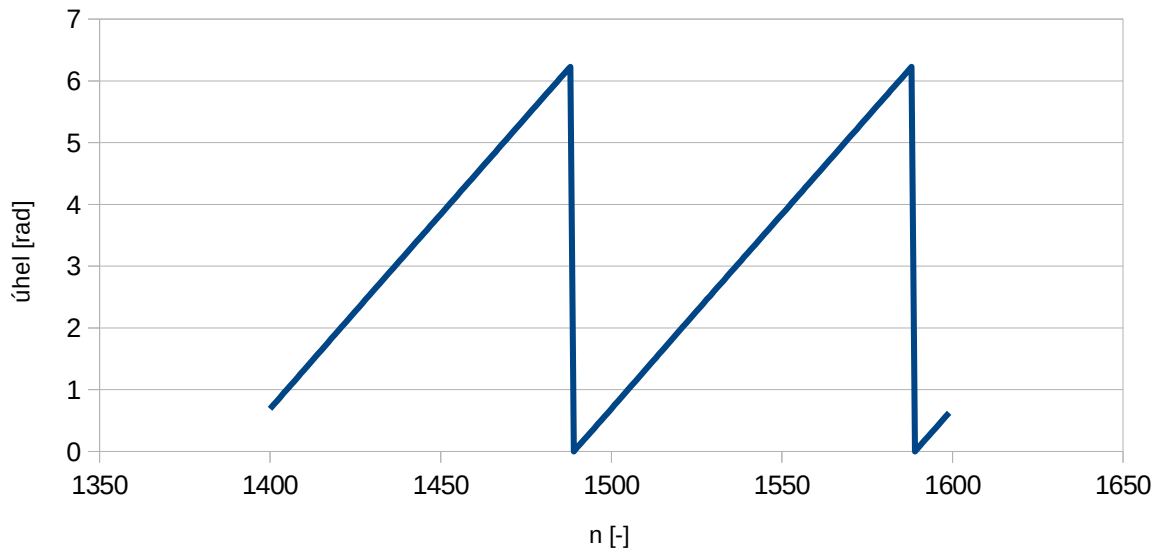
Měření probíhalo vždy nejdříve bez spínání tranzistorů pouze se sledováním výpočtů a regulace. Na následujícím grafu je zobrazen průběh fázového napětí vypočteného ze změřených sdružených a průběh spočítaný funkcí sinus z úhlu vystupujícího z fázového závěsu.

Osa x v grafech odpovídá zachyceným prvkům. Se vzorkováním 10 kHz tak zobrazovaných 200 hodnot, které byly uloženy do pole, odpovídá 20ms.



Obr. 20: Graf fázového napětí a generovaného sinusového průběhu

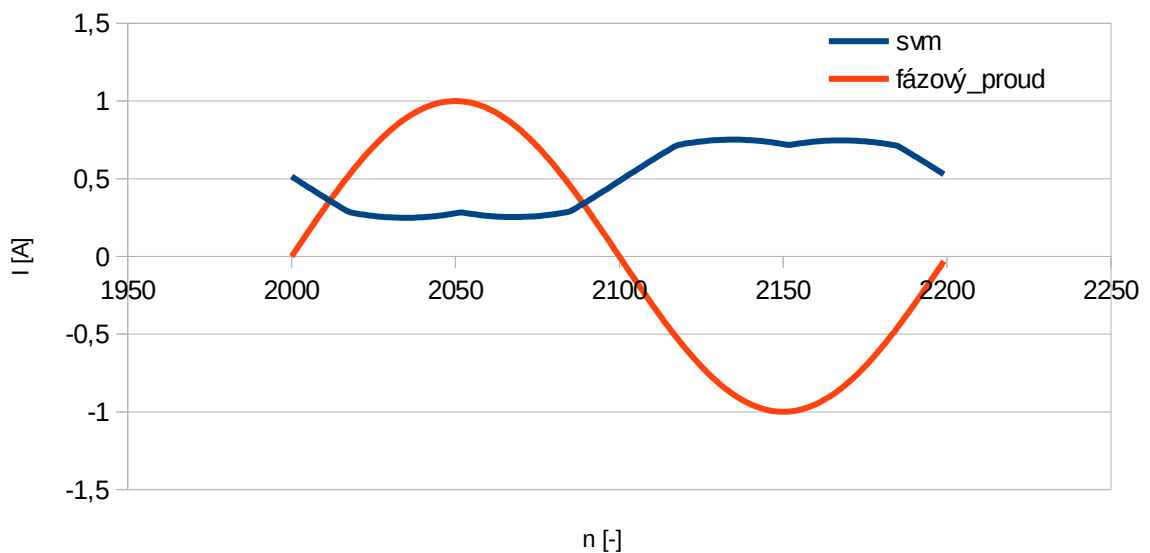
Graf úhlu vypočteného fázovým závěsem



Obr. 21: Graf úhlu vypočteného fázovým závěsem

Pro ověření proudové regulace byl proveden výpočet zatím pouze v mikrokontroleru. Jako vstup byly generovány sinusové průběhy proudů o velikosti 1 A, a byl pozorován výstup prostorové modulace. Požadovaný proud byl měněn v proměnné ručně za běhu. Následující graf ukazuje průběh regulace pro jednu fázi při požadovaném proudu 0,5 A.

Graf fázového proudu a průběhu signálu z modulátoru prostorového vektoru



Obr. 22: Graf fázového proudu a průběhu signálu z modulátoru prostorového vektoru



10 ZÁVĚR

Cílem této práce bylo vytvoření modelu aktivního usměřovače, následná simulace modelu, implementace řídicích algoritmů v reálném zařízení a měření realizovaného řízení. V první části práce byly rozebrány možné typy řízení s vysvětlením jejich principů. Kapitoly 2 až 5 se tedy zabývají popisem aktivního usměřovače, transformací veličin, jednotlivými typy řízení a návrhem regulátorů. Jako nejvhodnější typ bylo zvoleno napětově orientované řízení, protože se jedná o stabilní a bezpečný způsob regulace. Pro ověření tohoto typu řízení byl vytvořen model v programu Matlab Simulink. Parametry modelu odpovídají zařízení pouze přibližně, protože v této části nejsou klíčové. Model byl simulován s očekávanými výsledky, kdy je ze sítě odebírán sinusový proud a v inverzním režimu je opět sinusový vrácen zpět do sítě. Simulací tedy bylo potvrzeno, že zvolený typ řízení je vhodný pro následnou implementaci.

V druhé části se práce zabývá reálným zařízením. Nejdříve je rozebráno prostředí pro vytváření kódu a parametry skutečného zařízení firmy Elcom, na kterém byla realizace prováděna. Zařízení je řízeno mikrokontrolerem TMS320F28335 firmy TI. Stejný kontroler byl jako součást vývojového kitu zapůjčen firmou Elcom pro vytváření kódu. Popisu funkcí mikrokontroleru je věnována jedna z podkapitol. Další podkapitola se zabývá prostředím pro tvorbu programu. Tímto prostředím je Code Composer Studio od firmy TI. Kód je vytvářen v jazyce C/C++ a pomocí programu Code Composer Studio zkompilován a nahrán do mikrokontroleru. Následující kapitola se věnuje popisu kódu. Pro jednodušší ladění a lepší porovnání požadovaných průběhů byl vytvořen v Matlabu druhý model přesněji odpovídající reálnému zařízení. Kód byl poté realizován podle tohoto modelu. V kódu byly využity funkce od výrobce a od firmy Elcom pro nastavení mikrokontroleru a komunikaci s dalšími prvky zařízení. Vlastní funkce se pak týkají přímo řízení zařízení. Funkce používají datový typ float s pohyblivou řádovou čárkou. Mikrokontroler má pro tento typ hardwarové uzpůsobení a výpočty jsou proto přesné a stále dostatečně rychlé. V poslední kapitole je popsáno konkrétní nastavení a výsledky měření. Výsledky byly načítány z mikrokontroleru a zpracovány na počítači.

Nejdříve byl sledován výpočet fázových napětí a výpočet úhlu pomocí fázového závěsu. Fázový závěs se podařilo odladit, takže funguje velmi přesně, jak je patrné z uvedených grafů. Přestože napětí nemá čistě sinusový tvar, závěs je schopen se rychle synchronizovat na fázové napětí. Další řízení bylo prováděno pouze v mikrokontroleru, kde bylo nutné zavést idealizaci vstupních proudů. Z výsledného grafu je vidět, že regulace odpovídá očekávanému průběhu a i přes idealizaci by měla být schopna reálný přístroj regulovat. Z časových důvodů však další řízení nebylo na reálném zařízení zkoušeno, bude však dále postupně testováno a nastavováno, aby práce na usměřovači pro firmu byla dokončena a zařízení mohlo fungovat s požadovanými výsledky.



LITERATURA

- [1] RASHID Muhammad H. *Power electronics handbook*, ISBN 978-0-12-382036-5, 2011
- [2] MALINOWSKI Mariusz. *Sensorless Control Strategies for Three - Phase PWM Rectifiers*. Warsaw, 2001. PhD Thesis
- [3] GIGLIA G. Serpota C., Pucci M. Vitale G. *Comparison of Control Techniques for Three-Phase Distributed Generation Based on VOC and DPC*, International Conference on Renewable Energy and Power Quality (ICREPQ'08), Santander, Spain, 2008
- [4] KLÍMA, Bohumil. *Střídavé pohony*. Brno, 2014. skripta
- [5] MATHWORKS INC. *Napověda programu MATLAB*, 2013.
- [6] TEXAS INSTRUMENTS. *TMS320F28335/F28334/F28332/F28235/F28234/F28232 Digital Signal Controllers (Rev. M)* [online]. 2012 [cit. 2016-05-17]. Dostupné z: <http://www.ti.com/lit/ds/symlink/tms320f28335.pdf>
- [7] TEXAS INSTRUMENTS. *TMS320x2833x, 2823x Analog-to-Digital Converter (ADC) Module Reference Guide (Rev. A)* [online]. 2007 [cit. 2016-05-16]. Dostupné z: <http://www.ti.com/lit/ug/spru812a/spru812a.pdf>
- [8] TEXAS INSTRUMENTS. *TMS320x2833x, 2823x Enhanced Pulse Width Modulator (ePWM) Reference Guide (Rev. A)* [online]. 2009 [cit. 2016-05-17]. Dostupné z: <http://www.ti.com/lit/ug/sprug04a/sprug04a.pdf>
- [9] TEXAS INSTRUMENTS. *TMS320x2833x, 2823x System Control and Interrupts Reference Guide (Rev. D)* [online]. 2010 [cit. 2016-05-17]. Dostupné z: <http://www.ti.com/lit/ug/sprufb0d/sprufb0d.pdf>
- [10] BORMANN, Frank. TEXAS INSTRUMENTS. *Texas Instruments C2000 Teaching Materials*.
- [11] TEXAS INSTRUMENTS. *C28x Floating Point Unit fastRTS Library Module User's Guide* [online]. 2010 [cit. 2016-05-17]. Dostupné z: http://www.deyisupport.com/cfs-file.ashx/___key/communityserver-discussions-components-files/56/6281.C28x_5F00_FPU_5F00_FastRTS.pdf
- [12] WEISSTEIN, Eric W. *Euler Forward Method*. *MathWorld--A Wolfram Web Resource* [online]. [cit. 2016-05-20]. Dostupné z: <http://mathworld.wolfram.com/EulerForwardMethod.html>
- [13] KLÍMA, Bohumil *Mikroprocesorové řízení elektrických pohonů*. Brno, 2014, skriptum, VUT
- [14] BAREŠ, J. *Návrh a simulace řízení jednofázového aktivního usměrňovače*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2014. 28 s. Vedoucí bakalářské práce doc. Ing. Bohumil Klíma, Ph.D.
- [15] TEXAS INSTRUMENTS. *C2000 Systems and Applications: Digital Motor Control. Software Library: Target Independent Math Blocks v13.1* [online]. 2013 [cit. 2016-05-20]. Dostupné z: https://e2e.ti.com/cfs-file/___key/communityserver-discussions-components-files/312/DMC-MATH_5F00_v13.1.pdf



PŘÍLOHY

1. Model usměrňovače z teoretické části práce
2. Model usměrňovače z praktické části práce
3. Soubory kódu pro CCS bez souborů firmy Elcom

Přílohy jsou na přiloženém CD.