

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra elektroniky

**Laboratorní stanoviště se střídavými
regulovanými pohony**
AC Controlled Drive Laboratory Stand

2015

Bc. Lukáš Gelnar

Zadání diplomové práce

Student: **Bc. Lukáš Gelnar**
Studijní program: N2649 Elektrotechnika
Studijní obor: 2612T015 Elektronika
Téma: **Laboratorní stanoviště se střídavými regulovanými pohony
AC Controlled Drive Laboratory Stand**

Zásady pro vypracování:

1. Proveďte rozbor moderních metod řízení střídavých regulovaných pohonů s ohledem na požadavky pro laboratorní stanoviště
2. Prostřednictvím programu LabView vytvořte uživatelské prostředí pro komunikaci s řídicím systémem s možností modifikace vstupních a výstupních parametrů a veličin střídavého regulovaného pohonu
3. Na základě pokynů vedoucího diplomové práce vytvořte průvodní zprávy k vytipovaným laboratorním úlohám

Seznam doporučené odborné literatury:

Dle doporučení vedoucího závěrečné práce

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Martin Kuchař, Ph.D.**

Datum zadání: 01.09.2014

Datum odevzdání: 07.05.2015



doc. Ing. Petr Palacký, Ph.D.
vedoucí katedry

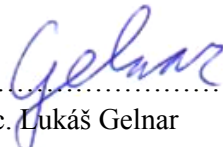


prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení:

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne: 7.5.2015


.....
Bc. Lukáš Gelnar

Poděkování:

Na tomto místě bych rád poděkoval svému vedoucímu diplomové práce, panu Ing. Martinovi Kuchařovi, Ph.D., panu doc. Ing. Petru Palackému, Ph.D. a Ing. Martinovi Sobkovi, Ph.D. za vedení a poskytnutí cenných rad týkajících se diplomové práce.

Abstrakt

Tato diplomová práce se zaměřuje na rozbor vybraných moderních metod řízení střídavých regulovaných pohonů. Jsou zde znázorněny struktury řízení a výpočet jednotlivých bloků, např. transformace ze statorového souřadného systému na rotorový souřadný systém a další. Práce obsahuje také řízení výstupního napětí střídače, kde jsou popsány různé typy PWM a jejich výhody, nevýhody a využití napětí meziobvodu. Zabývá se také vytvořením uživatelského prostředí pro komunikaci PC s řídicím systémem, kde jsou popsány jednotlivé bloky a celkový princip komunikace. Tato uživatelská prostředí jsou určena pro vektorové řízení synchronního a asynchronního motoru, kde se v PC nastavují parametry regulátorů a vyčítají se zde hodnoty. Následně byly vytvořeny laboratorní úlohy, pomocí nichž si budou moci studenti měřením ověřit funkčnost elektrického regulovaného pohonu.

Klíčová slova

LabVIEW, synchronní motor, asynchronní motor, vektorové řízení, střídavé regulované pohony

Abstract

This thesis focuses on analysis of modern methods of AC controlled drives. There are shown control structures and calculations for individual blocks e.g. transformation from stator coordinate system to rotor coordinate system and others. This thesis also covers control the output voltage inverter, which describes the different types of PWM and their advantages, disadvantages and the use of the DC bus voltage. It also deals with creating a user interface for communication between the PC and control system, which describes the individual blocks and the overall principle of communication. These user interfaces are designed for vector control of synchronous and the induction motor, the PC will set the parameters of regulators and read the values here. Subsequently, laboratory tasks were created through which students will be able to verify the functionality of the electrical controlled drive.

Key Words

LabVIEW, synchronous motor, asynchronous motor, vector control, AC controlled drive

Seznam použitých symbolů:

Veličina	Jednotka	Popis	Veličina	Jednotka	Popis
C	F	Kondenzátorová baterie	u_{idR}	V	Výstupní napětí z regulátoru R_{id}
f_1	Hz	Frekvence 1. harmonické	u_{1max}	V	Maximální statorové napětí
f_s	Hz	Spínací frekvence	u_{1q}	V	Momentotvorné napětí v RSS
i_1	A	Statorový proud	u_{1qe}	V	Napětí z bloku zrušení vazby
i_{1a}	A	Proud statoru fáze a	u_{1qR}	V	Výstupní napětí z regulátoru R_{1q}
i_{1b}	A	Proud statoru fáze b	u_{1x}	V	Tokotvorné napětí v OSS
i_{1d}	A	Tokotvorný proud v RSS	u_{1xe}	V	Napětí z bloku zrušení vazby
i_{1max}	A	Maximální statorový proud	u_{1xR}	V	Výstupní napětí z regulátoru R_{1x}
i_{1q}	A	Momentotvorný proud v RSS	u_{1y}	V	Momentotvorné napětí v OSS
i_{1qmax}	A	Maximální tokotvorný proud	u_{1ye}	V	Napětí z bloku zrušení vazby
i_{1x}	A	Tokotvorný proud v OSS	u_{1yR}	V	Výstupní napětí z regulátoru R_{1y}
i_{1y}	A	Momentotvorný proud v OSS	$u_{1\alpha}$	V	Statorové napětí α složka
$i_{1\alpha}$	A	Statorový proud α složky	$u_{1\beta}$	V	Statorové napětí β složka
$i_{1\beta}$	A	Statorový proud β složky	U_a	V	Fázové napětí fáze a
i_m	A	Magnetizační proud	U_{ab}	V	Sdružené napětí fází a, b
$i_{m\alpha}$	A	Magnetizační proud α složky	U_b	V	Fázové napětí fáze b
$i_{m\beta}$	A	Magnetizační proud β složky	\vec{u}	V	Žádaný vektor
K_i	-	Konstanta integrátoru	u_d	V	Napětí meziobvodu
K_r	-	Konstanta zesílení	u_p	V	Indukované protinapětí
L_s	H	Statorová indukčnost	u_{pmax}	V	Maximální indukované protinapětí
m	-	Modulační poměr	u_{ra}	V	Referenční napětí fáze a
M	-	Modulační index	u_{rb}	V	Referenční napětí fáze b
m_f	-	Frekvenční modulační poměr	u_{rc}	V	Referenční napětí fáze c
R_s	Ω	Statorový odpor	u_t	V	Okamžité napětí trojúhelníkového signálu
t_1	s	Doba sepnutí 1. vektoru	U_T	V	Napětí trojúhelníkového signálu
t_2	s	Doba sepnutí 2. vektoru	γ	rad	Úhel mezi OSS a SSS
T_r	s	Časová konstanta regulátoru	ε	rad	Úhel mezi RSS a SSS
T_s	s	Perioda spínání	ρ	rad	Úhel žádaného vektoru \vec{u}
$U_{(1)}$	V	Napětí 1. harmonické	σ	-	Činitel rozptylu
$U_{(3)}$	V	Napětí 3. harmonické	Ψ_{PM}	Wb	Magnetický tok rotoru
u_1	V	Statorové okamžité napětí	Ω	rad/s	Úhlová mechanická rychlost
U_1	V	Statorové napětí	ω_{el}	rad/s	Elektrická úhlová rychlost
u_{1d}	V	Tokotvorné napětí v RSS	ω_{im}	rad/s	Úhlová rychlost vektoru magnetizačního proudu
u_{1de}	V	Napětí z bloku zrušení vazby	Ω_N	rad/s	Jmenovitá úhlová rychlost

Seznam použitých zkratk:

Zkratka	Popis	Zkratka	Popis
2D	Dvourozměrné pole	Re	Reálná složka
AC	Střídavý proud	R _{ild}	Regulátor tokotvorné složky proudu
ADC	Analogově digitální převodník	R _{ilq}	Regulátor momentotvorné složky proudu
AM	Asynchronní motor	R _{ilx}	Regulátor tokotvorné složky proudu
CRC	Cyclic Redundancy Checks - cyklický redundantní součet	R _{ily}	Regulátor momentotvorné složky proudu
DA	Digitálně analogový	R _{im}	Regulátor magnetizačního proudu
DAC	Digitálně analogový převodník	R _{odbuz}	Regulátor odbuzení
DC	Stejnoseměrný proud	RS232	Sériové komunikační rozhraní
DSC	Digitální signálový kontrolér	RSS	Rotorový souřadný systém
GPIB	General Purpose Interface Bus	R _ω	Regulátor rychlostí
GPIO	General Purpose input/output	SCI	Serial Communication Interfaces
I	Integrační člen	SSS	Statorový souřadný systém
Im	Imaginární složka	subVI	Podprogram
OSS	Orientovaný souřadný systém	U/f	Poměr napětí/frekvence
PC	Osobní počítač	USB	Universal Serial Bus
PI	Proporcionálně integrační člen	VI	Virtual Instruments - virtuální přístroje
PMSM	Synchronní motor s permanentními magnety	VISA	Virtual Instrument Software Architecture
PWM	Pulsně-šířková modulace	VXI	VMEbus eXtension for Instrumentation
PXI	PCI eXtensions for Instrumentation		

Obsah

1. ÚVOD	9
2. LABVIEW.....	10
3. SIGNÁLOVÝ KONTROLÉR TMS320F28335.....	13
4. KOMUNIKACE PC S DSC.....	15
4.1. ZADÁVÁNÍ DAT DO DSC	15
4.2. VYČÍTÁNÍ DAT Z DSC.....	15
4.3. KONTROLNÍ SOUČET	15
5. VEKTOROVÉ ŘÍZENÍ SYNCHRONNÍHO MOTORU S PERMANENTNÍMI MAGNETY	17
5.1. POPIS STRUKTURY	17
5.2. UŽIVATELSKÉ PROSTŘEDÍ	21
6. ŘÍZENÍ VÝSTUPNÍHO NAPĚTÍ NEPŘÍMÉHO MĚNIČE FREKVENCE.....	35
6.1. KOMPARAČNÍ PWM	35
6.2. KOMPARAČNÍ PWM S PŘIDÁNÍM TŘETÍ HARMONICKÉ	37
6.3. VEKTOROVÁ PWM	38
6.4. UŽIVATELSKÉ PROSTŘEDÍ	40
7. VEKTOROVÉ ŘÍZENÍ ASYNCHRONNÍHO MOTORU	45
7.1. POPIS STRUKTURY	45
7.2. UŽIVATELSKÉ PROSTŘEDÍ	47
8. ZÁVĚR.....	50
9. SEZNAM POUŽITÉ LITERATURY	51
10. SEZNAM PŘÍLOH	52

1. Úvod

V dnešní době jsou regulované pohony pro dynamicky náročné aplikace řízeny většinou moderními způsoby řízení, které jsou realizované pomocí mikropočítačové techniky. Tato skutečnost je způsobena rychlým vývojem mikroprocesorů. Mezi nejpoužívanější moderní metody řízení patří vektorové řízení a přímé řízení momentu. Jedním z požadavků kladen na řídicí systém je vysoká rychlost výpočtu, pro dosažení takové rychlosti je nutné použít signálový procesor.

Možností zadávání nebo vyčítání parametrů je několik. Používá se klávesnice s displejem, častěji používaná je však možnost komunikace s uživatelem pomocí sériové linky, která má mnoho výhod. Důležitým aspektem je použití software pro komunikaci, analýzu a sběr dat. Jedním z nejznámějších software je LabVIEW od firmy National Instruments.

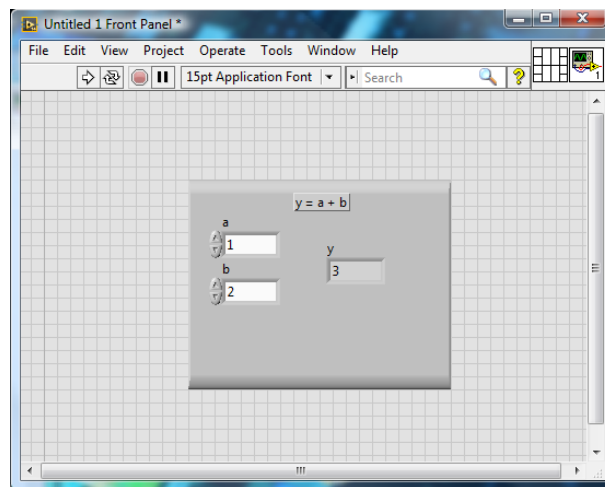
Stejnoseměrné motory s mechanickým komutátorem dnes ztrácejí podíl na trhu a jsou postupně nahrazovány střídavými pohony – synchronními a asynchronními motory. S rostoucí cenou elektrické energie bude kladen větší důraz na energeticky méně náročné pohony.

2. LabVIEW

Pro potřeby práce je nejprve nezbytné popsat software LabVIEW. Je to programové vývojové prostředí, podobně jako programovací jazyk C nebo BASIC. Tyto jazyky používají pro tvorbu řádkového kódu textové programování. Důležitým prvkem odlišnosti od těchto vývojových prostředí je tvorba programu v blokovém diagramu, tedy grafické programování. Stejně jako C nebo BASIC používá tento software rozsáhlé knihovní funkce pro tvorbu libovolného programového úkonu. LabVIEW je programovací systém pro všeobecné použití, obsahuje knihovny, např. pro GPIB a sériové ovládání přístrojů, sběr a analýzu dat, ukládání dat a mnohé další. Vytvořené programy se nazývají a mají příponu souboru VI (z anglického výrazu **V**irtual **I**nstruments – virtuální přístroje), jelikož vzhledem a činností lze napodobit skutečné přístroje.

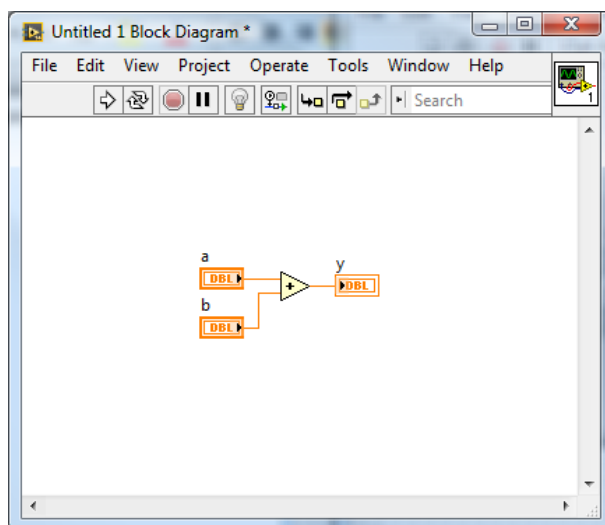
VI se dělí na následující struktury:

- 1) Uživatelské rozhraní se nazývá **přední panel**, protože napodobuje panel skutečného měřicího přístroje. Obsahuje tlačítka, přepínače, grafy, ostatní ovládací a indikační prvky.



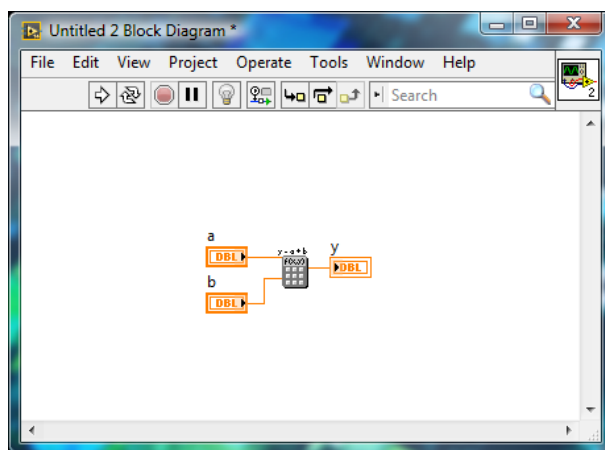
Obr. 2.1 Přední panel

- 2) VI dostane pokyny z naprogramovaného grafického prostředí, z tzv. **blokového diagramu**, pomocí něhož se tvoří grafický kód. Používají se zde matematické a logické funkce, nastavuje se komunikace, definuje se ukládání do souboru a je možno použít mnoho dalších užitečných funkcí.



Obr. 2.2 Blokový diagram

- 3) Jednotlivé VI mohou sloužit jako podprogramy v jiných VI. Tyto podprogramy nazýváme **subVI**, které musí mít nadefinovanou ikonu a konektor tak, aby se mohly data přenášet mezi VI a subVI. SubVI se dá představit jako blok, ve kterém se provedou určité operace. Pro správnou funkci se musí nastavit konektor, který se nastaví v předním panelu na hlavní liště nahoře vpravo – obr. 2.1. Označením jednotlivého pinu se k němu přiřadí ovládací prvek (control) nebo indikační prvek (indikátor).

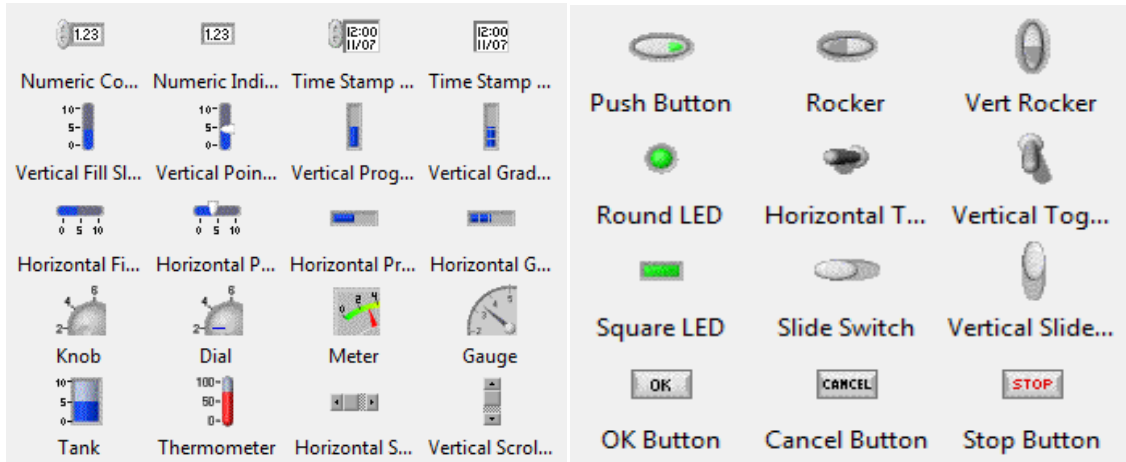


Obr. 2.3 VI s použitým podprogramem

Controly, konstanty a indikátory

Control je ovládací prvek umístěný na předním panelu určený pro zadávání dat do VI. Stiskem pravého tlačítka myši v předním panelu se otevře nabídka s prvky, které mohou být umístěny v uživatelském prostředí – obr. 2.4. Indikátor slouží jako zobrazovací prvek umístěný

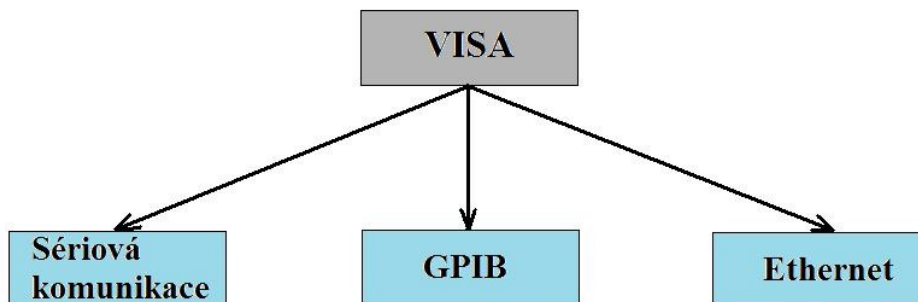
na předním panelu. Všechny indikátory a controly se dají zaměňovat, tzn. možnost přepnutí jednoho typu prvku za druhý typ a naopak. Konstanty jsou umístěné jen v blokovém diagramu a jsou určené k nastavené neměnné hodnotě. [1]



Obr. 2.4 Nabídka ovládacích a indikačních prvků

VISA

VISA je vrstva sjednocující rozdíly jednotlivých softwarových standardů, jako jsou např. GPIB, VXI, PXI, sériové rozhraní, USB nebo Ethernet. Výhoda sériové komunikace spočívá v jednoduchosti a nižší ceně. U osobních počítačů a notebooků už dnes není pravidlem mít zabudovaný sériový port a je potřeba si pořídit převodník, např. USB – RS232. Proto většina výrobců dnes přechází na USB komunikaci. VISA je nezávislá na sběrnici, platformě a prostředí, neumožňuje programovat přístroje, jen volá ovladače na nižší úrovni viz obr. 2.5. [2]



Obr. 2.5 Hierarchie VISA

3. Signálový kontrolér TMS320F28335

Digitální signálový kontrolér, označovaný také jako DSC, je v podstatě mikrokontrolér, na který se klade nárok na vysokou rychlost zpracování velkého množství dat. Nejčastěji se používá v real-time aplikacích. Typické signálové procesory jsou postaveny na harvardské architektuře, tudíž data a kód programu komunikují po vlastní sběrnici. Pro laboratorní stanoviště se střídavými regulovanými pohony byl použit DSC od firmy Texas Instruments typu TMS320F28335. DSC má možnost pracovat v plovoucí řadové čárce. TMS320F28335 je 32bitový systém pracující na frekvenci 150MHz. Při této frekvenci je doba zpracování jednoho cyklu 6,667 ns. Procesorová deska řídicího systému je doplněna o nezbytné a podpůrné obvody, které zajišťují jeho správnou funkci. Na obr. 3.1 je zobrazena procesorová deska řídicího systému bez analogové a digitální desky, které se zapojují do spodních a pravých horních konektorů. DSC slouží pro měření analogových signálů, jejich výpočtu a odeslání signálu na výstup. Jednou z důležitých funkcí je komunikace s uživatelem pomocí rozhraní USB.



Obr. 3.1 Procesorová deska řídicího systému

Základní vlastnosti TMS320F28335 jsou:

- Frekvence 150MHz
- Formát plovoucí řadové čárky IEEE-754 Single-Precision
- Harvardská architektura
- 6 kanálů DMA řadičů
- Flash paměť 256k x 16, SARAM 34k x 16
- GPIO0 až GPIO63 piny mohou být připojený k 1 z 8 externích přerušení
- 2 kvadrurní enkodéry
- osm 32-bitových časovačů
- tři 32-bitové CPU časovače
- 2 CAN moduly, 3 SCI moduly, 1 SPI modul, 1 I2C sběrnice
- 16 kanálů - 12-bitových AD převodníků, 2 sample & hold obvody
- 18 PWM výstupů

Komunikace s uživatelem probíhá pomocí převodníku USB na TTL-UART, což zajišťuje obvod FTDI232RL. Oproti převodníku USB-RS232 pracuje v 5V logice. Tento převodník je umístěn na procesorové desce řídicího systému. Na obr. 3.2 je zobrazen kompletní řídicí systém s analogovou i digitální deskou.



Obr. 3.2 Kompletní řídicí systém

4. Komunikace PC s DSC

4.1. Zadávání dat do DSC

Data z PC se odesílají pomocí USB do kontroléru přímo na adresu paměťového místa, na kterém je proměnná uložena. Podle počtu odesílaných dat se odesílá jen jedna hodnota, nebo více hodnot. Kontrolér podle datového typu přidělí adresu a velikost paměťového místa, v případě odeslání jedné hodnoty se adresuje na danou adresu, při odeslání více hodnot se data zapisují od vyčtené adresy a následně do adresy vyšší o délku slova. Při odesílání jakéhokoliv parametru se s ním společně odešle kontrolní součet.

4.2. Vyčítání dat z DSC

Vyčítání funguje stejným způsobem, jen opačným směrem – data se přenášejí z kontroléru do počítače.

Po provedení kontrolního součtu se data odešlou, jak z PC, tak i z kontroléru. V druhém zařízení se data zpracují, a pokud je jejich kontrolní součet v pořádku, data jsou přijata správně. Odesílané a přijímané datové typy jsou: Float32, Uint16 a Uint32.

4.3. Kontrolní součet

Kontrolní součet je doplňková informace odesílaná se spolu s daty do kontroléru. Slouží k ověření, zda je odesílána informace úplná a nedošlo během přenosu k chybě. V přijímacím zařízení – DSC se přepočte kontrolní součet, v případě nesouhlasu odesílaného kontrolního součtu se součtem spočteným v DSC dojde k chybě a kontrolér s těmito přijímanými daty dále nepracuje. Pokud jsou data přijata a vypočtené CRC jsou shodné, považuje data za přijatá.

Pojem CRC pochází z anglického spojení Cyclic Redundancy Checks, do češtiny se to překládá jako Cyklický redundantní součet. Je to speciální funkce, která se používá k detekci chyb během přenosu či ukládání dat. Díky své jednoduchosti a dobré matematické vlastnosti se jedná o nejrozšířenější způsob realizace kontrolního součtu. Princip cyklického součtu spočívá v cyklickém posuvu znaků kódového slova a následně vzniku opět kódového slova. Přenášené slovo lze popsat generujícím polynomem, který se určuje podle vztahu 4.1. V tomto případě se jedná o CRC-16 - to je řídicí polynom stupně 16 a může obsahovat nejvyšší mocninu x^{16} .

$$g(x) = x^{16} + x^{15} + x^2 + 1 \quad (4.1)$$

Příklad výpočtu CRC-3

- 1) Převod čísla např. 157_{10} se převede do binární formy, tj. 10011101_2 .
- 2) Binární číslo 10011101_2 se vyjádří v datovém polynomu: $x^7 + x^4 + x^3 + x^2 + 1$
- 3) Posunutí dat o stupeň generujícího polynomu (vynásobení x^3)

$$d(x) = x^{10} + x^7 + x^6 + x^5 + x^3 \quad (4.2)$$

- 4) Dělení datového polynomu polynomem generujícím: $g(x) = x^3 + x + 1$

$$\begin{array}{r}
 (x^{10} + x^7 + x^6 + x^5 + x^3) : (x^3 + x + 1) = x^7 + x^5 + 1 \\
 x^{10} + x^8 + x^7 \\
 \underline{x^8 + x^6 + x^5 + x^3} \\
 x^8 + x^6 + x^5 \\
 \underline{x^8 + x^6 + x^5} \\
 x^3 \\
 x^3 + x + 1 \\
 \underline{x + 1}
 \end{array}$$

- 5) Výsledné CRC je tedy 011_2 (délka CRC je rovna stupni generujícího polynomu)
- 6) Odeslaná posloupnost je $10011101011_2 \rightarrow x^{10} + x^7 + x^6 + x^5 + x^3 + x + 1$
- 7) Ověření správnosti příjmu:

$$\begin{array}{r}
 (x^{10} + x^7 + x^6 + x^5 + x^3 + x + 1) : (x^3 + x + 1) = x^7 + x^5 + 1 \\
 x^{10} + x^8 + x^7 \\
 \underline{x^8 + x^6 + x^5 + x^3 + x + 1} \\
 x^8 + x^6 + x^5 \\
 \underline{x^8 + x^6 + x^5} \\
 x^3 + x + 1 \\
 x^3 + x + 1 \\
 \underline{0}
 \end{array}$$

- 8) Výsledný zbytek po dělení je 0, data byla přijata správně.

Stejným způsobem funguje odesílání a příjem dat do kontroléru s jedním rozdílem - využívá CRC - 16. [4] [5]

5. Vektorové řízení synchronního motoru s permanentními magnety

5.1. Popis struktury

Pro dynamicky náročné pohony se synchronními motory se používá princip vektorově orientovaného řízení. Princip spočívá v tom, že se vektor statorového proudu \vec{i}_1 rozloží do dvou navzájem kolmých složek. Změna jedné složky neovlivňuje druhou složku a lze tedy nezávisle řídit moment a magnetický tok. Rozložení vektoru proudu statoru na dvě složky je zřejmé z obr. 5.1. [6]

Ze změřených statorových proudů i_{1a} a i_{1b} se provede pomocí Clarkové transformace přepočít z 3-fázového (a,b,c) do 2-fázového systému (α,β). V případě nevyvedeného středu stačí znát jen 2 statorové proudy i_{1a} a i_{1b} , výpočet pak bude vypadat následovně:

$$i_{1\alpha} = i_{1a} \quad (5.1)$$

$$i_{1\beta} = \frac{1}{\sqrt{3}}i_{1a} + \frac{2}{\sqrt{3}}i_{1b} \quad (5.2)$$

Dalším krokem je výpočet úhlu mezi statorovým souřadným systémem (SSS) a rotorovým souřadným systémem (RSS), z čehož se spočte sinus a cosinus tohoto úhlu a následně se výpočte pomocí Parkovy transformace i_{1d} a i_{1q} , výpočet je dán vztahy:

$$i_{1d} = i_{1\alpha} \cdot \cos \varepsilon + i_{1\beta} \cdot \sin \varepsilon \quad (5.3)$$

$$i_{1q} = -i_{1\alpha} \cdot \sin \varepsilon + i_{1\beta} \cdot \cos \varepsilon \quad (5.4)$$

Tokotvorný proud i_{1d} a momentotvorný proud i_{1q} jsou samostatně regulovány. Výstupy z těchto regulátorů se převedou zpět do SSS pomocí inverzní Parkovy transformace podle následujících vztahů:

$$u_{1\alpha} = u_{1d} \cdot \cos \varepsilon - u_{1q} \cdot \sin \varepsilon \quad (5.5)$$

$$u_{1\beta} = u_{1d} \cdot \sin \varepsilon + u_{1q} \cdot \cos \varepsilon \quad (5.6)$$

Výstupní veličiny z regulátorů R_{i1d} a R_{i1q} jsou napětí u_{1dR} a u_{1qR} . V bloku zrušení vazby se spočtou napětí u_{1de} a u_{1qe} podle vztahu 5.9 a 5.10.

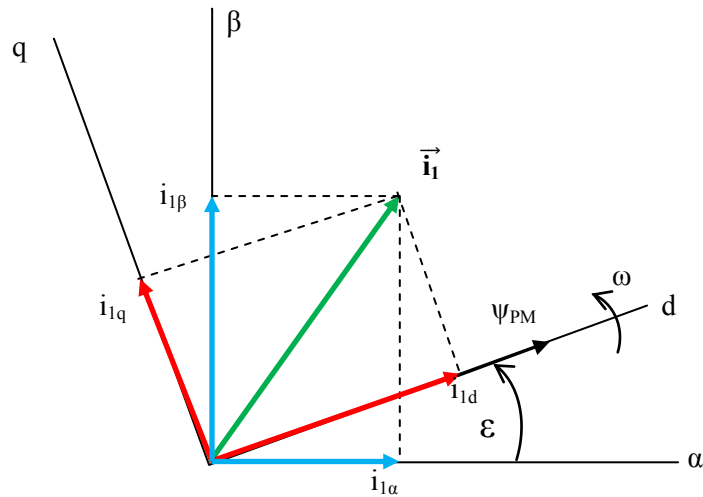
$$u_{1dR} = R_s i_{1d} + L_S \frac{di_{1d}}{dt} \quad u_{1qR} = R_s i_{1q} + L_S \frac{di_{1q}}{dt} \quad (5.7) (5.8)$$

$$u_{1de} = -\omega_{el} L_S i_{1q} \quad u_{1qe} = \omega_{el} (L_S i_{1d} + \psi_{PM}) \quad (5.9) (5.10)$$

Ke zkvalitnění regulací je vhodné přičíst k výstupním veličinám regulátorů proudů složky napětí.

$$u_{1d} = u_{1dR} + u_{1de} = R_s i_{1d} + L_S \frac{di_{1d}}{dt} - \omega_{el} L_S i_{1q} \quad (5.11)$$

$$u_{1q} = u_{1qR} + u_{1qe} = R_s i_{1q} + L_S \frac{di_{1q}}{dt} + \omega_{el} (L_S i_{1d} + \psi_{PM}) \quad (5.12)$$



Obr. 5.1 Vektorový diagram statorového proudu

Modul indukovaného protinapětí u_p vypočítává procesor, nesmí se překročit maximální mez, která je v daném rozsahu statorového napětí:

$$0,8 u_{1max} < u_{pmax} < 0,9 u_{1max} \quad (5.13)$$

Modul u_p je dán vztahem:

$$|u_p| = \omega_{el} \sqrt{(L_S i_{1d} + \psi_{PM})^2 + (L_S i_{1q})^2} \quad [7] \quad (5.14)$$

V normálním pracovním režimu je maximální účinnost ideálního PMSM dosažena při $i_{1d} = 0A$. Regulátor rychlosti řídí jen momentotvornou složku i_{1q} . Řízení je dáno vztahem:

$$u_{1q} = R_S i_{1q} + L_S \frac{di_{1q}}{dt} + \omega_{el} (L_S i_{1d} + \psi_{PM}) \quad (5.15)$$

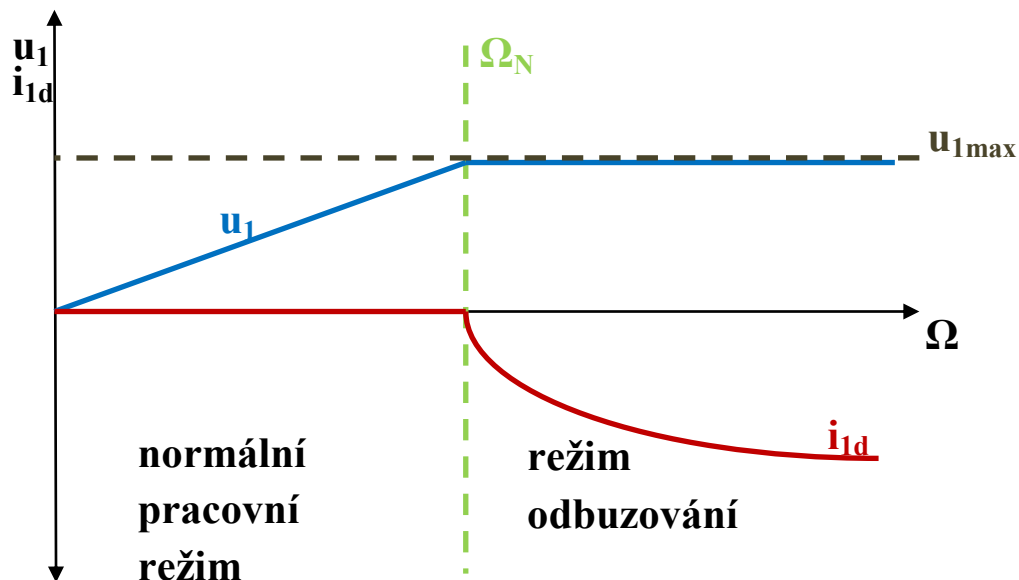
Ve skutečnosti má 3-fázový nepřímý měnič kmitočtu následující napěťové a proudové omezení:

- 1) Absolutní hodnota statorového napětí u_1 je limitována napětím v meziobvodu U_D a to určuje maximální napětí u_{1max} .
- 2) Absolutní hodnota statorového proudu i_1 je omezena maximální hodnotou proudu i_{1max} .

V režimu odbuzení jsou požadovány vyšší otáčky, protože měnič má maximální hodnotu proudu i_{1max} , musí se podle vztahu 5.16 snížit i_{1q} , čímž se sníží i moment stroje.

$$i_{1q} < \sqrt{i_{1max}^2 - i_{1d}^2} \quad (5.16)$$

Na obr. 5.2 jsou znázorněny pracovní režimy. V normálním pracovním režimu lze vidět, jak složka i_{1d} je nulová do jmenovité rychlosti motoru. S rostoucím statorovým napětím se zvyšuje rychlost, nad hodnotu jmenovitých otáček se musí omezit maximální statorové napětí na hodnotu u_{1max} . Oblast odbuzení se zajistí vhodným řízením i_{1d} do záporných hodnot. [8]



Obr. 5.2 Pracovní režimy

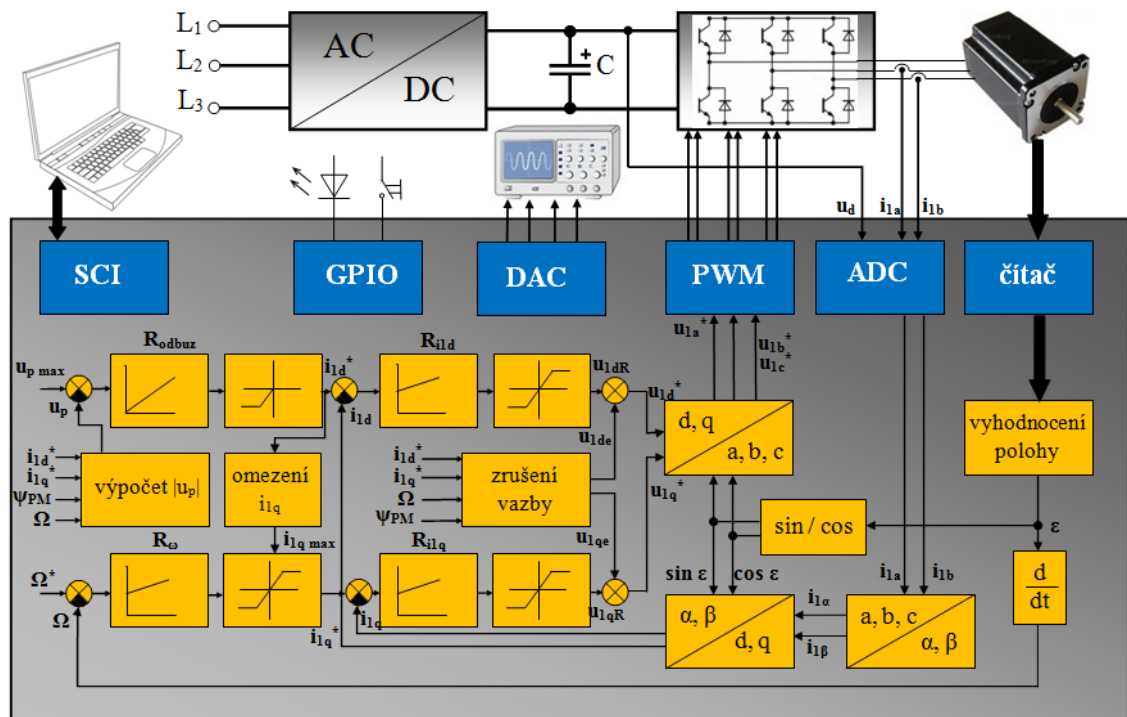
Popis regulátorů:

Do regulátoru odbuzení vstupuje rozdíl maximálního možného indukovaného protinapětí u_{pmax} a skutečné hodnoty u_p . Regulátor je pouze integrační, tudíž závisí jen na časové konstantě regulátoru T_r . Za regulátorem je umístěn blok omezovače, aby se mohla omezit kladná a záporná hodnota žádaného proudu i_{1d} .

Rychlostní regulátor je umístěn za rozdílovou složkou žádané a skutečné hodnoty rychlosti. V PI regulátoru je třeba nastavit konstantu zesílení K_r a časovou konstantu T_r . Omezovač za ním umístěný omezí maximální žádaný proud i_{1q} na požadované parametry, kde se bere v potaz i parametr i_{1qmax} .

Z výstupu regulátoru R_ω vystupuje žádaný parametr i_{1q} a od něj se odečte skutečný i_{1q} a výsledek vstupuje do PI regulátoru i_{1q} , jehož výstupní veličina je následně omezena. Stejným způsobem pracuje PI regulátor i_{1d} . Výstupní veličiny z těchto regulátorů (u_{1dR} a u_{1qR}) se sečtou s parametry (u_{1de} a u_{1qe}), které způsobí zrušení vazby k druhé složce.

Všechny tyto parametry regulátorů se zadávají pomocí LabVIEW, vždy se nastavuje kladné a záporné omezení, časová konstanta regulátoru, popř. zesílení regulátoru.



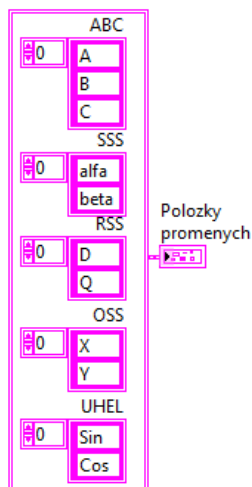
Obr. 5.3 Struktura regulace PMSM v RSS

5.2. Uživatelské prostředí

Aplikace vytvořena v LabVIEW musí být umístěna v adresáři, kde je umístěn program vytvořený v Code Composer Studio. Aplikace se spouští ikonou nazvanou PMSM.vi. Vytvořením kódu v programu C se vytvoří soubor main.c, který je umístěn ve složce source. Aplikace LabVIEW si z tohoto souboru vyčte všechny globální proměnné, jejich délky a typ dat a vypíše je do polí. Datové typy jsou typu float32, ABC, SSS, RSS, OSS, UHEL. Tyto typy jsou definovány pomocí souřadných systémů, úhlů a fází. V levé spodní části blokového diagramu jsou nadefinovány jednotlivé složky těchto souřadných systémů:

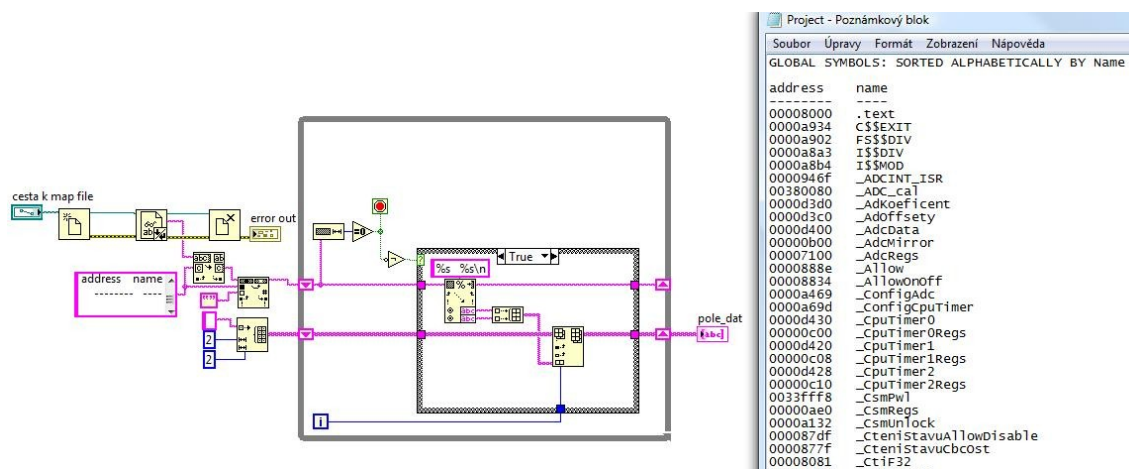
- 1) ABC – A,B,C
- 2) SSS – α, β
- 3) RSS – d,q
- 4) OSS – x,y
- 5) Uhel – sin, cos

Na obr. 5.4 jsou zobrazeny položky podle datového typu. Na obr. 5.6 jsou v levé spodní části definované datové typy, které se vyčítají z kontroléru.



Obr. 5.4 Položky proměnných

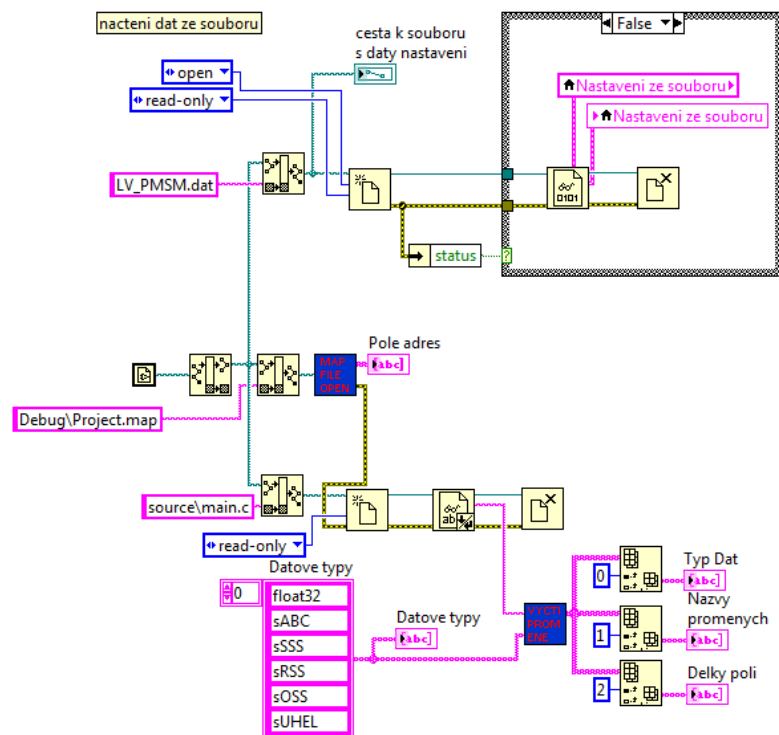
Code Composer Studio při přeložení a nahrání kódu do kontroléru vytvoří v adresáři Debug soubor s názvem Project.map, ve kterém se objeví všechny proměnné umístěné v kódu. Aplikace si z tohoto mapovacího souboru exportuje názvy všech globálních proměnných a jejich umístění v paměťových místech.



Obr. 5.5 Vyčítání pole adres

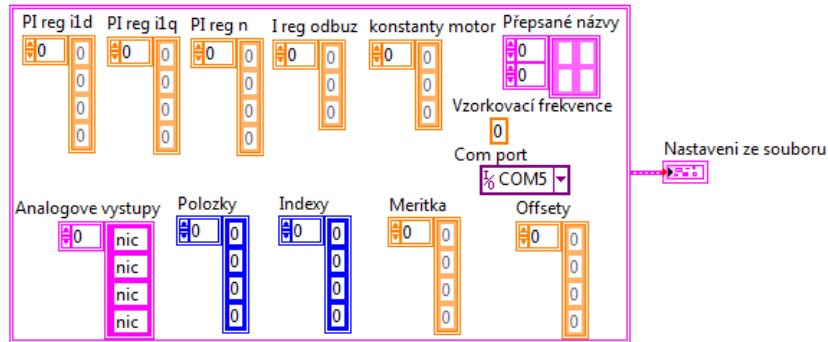
Na obr. 5.5 je zobrazeno subVI, pomocí něhož se provede výpis proměnných ze souboru. Vyhledá se v souboru název *GLOBAL SYMBOLS... adress name* a načte následující text. Z tohoto textu vytvoří 2D pole, v němž jsou v prvním sloupci nadefinované adresy paměťových míst v hexadecimálním tvaru a v druhém sloupci jsou přiřazeny proměnné odpovídající danému paměťovému místu.

Aplikace při otevření a spuštění automaticky načte poslední konfiguraci zadávaných parametrů. Tyto parametry se ukládají do souboru LV_PMSM.dat, který je umístěn ve stejné složce jako tato aplikace. V případě úspěšného načtení souboru uloží parametry do pole *Nastavení ze souboru*, ze kterého se dále vyčítají parametry.



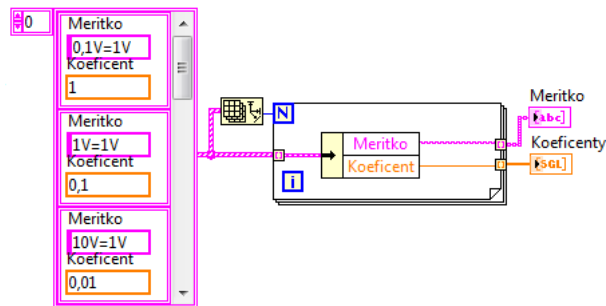
Obr. 5.6 Načtení dat ze souborů

V aplikaci, konkrétně v levé části blokového diagramu, se umísťují ovládací a indikační prvky, se kterými se poté pracuje. Jsou zde umístěny všechny kontroly a skryté pomocné indikátory. Patří zde i konstanty pro ukládání parametrů do souboru, které slouží k reprezentaci datového typu a velikosti pole. Tyto konstanty jsou umístěné v clusteru, pomocí něhož se poté přepisují hodnoty – obr. 5.7. Při otevření programu se načtou nulové hodnoty, což je způsobeno nulovými konstantami, a po spuštění se přepíší zadávanými parametry. Na obr. 5.6 nahoře vpravo lze vidět, že při načtení dat ze souboru se přepíše pole *Nastavení ze souboru*.



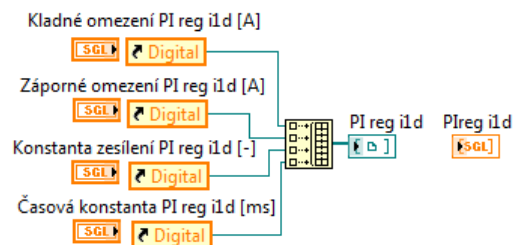
Obr. 5.7 Konstanty ukládané do clusteru

Zadávaní měřitek pro DA převodníky se provádí v bloku uvedeném na obr. 5.8, do kterého se zapíše měřítko, což slouží jako popis a koeficient (je podíl těchto měřitek podělený 10). Jsou zde uvedena měřítka pro napětí, proud, otáčky a úhly.



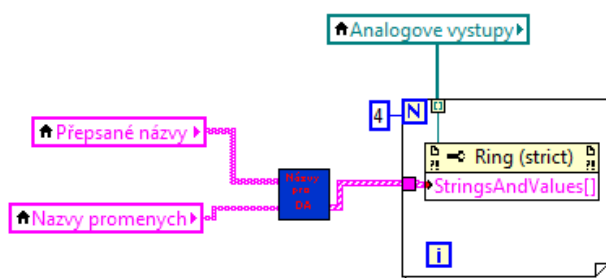
Obr. 5.8 Nastavení měřitek

Regulátory vytvářejí tzv. referenci, která se ukládá do paměti, a následně se k ní přistupuje. Indikační prvek **PIreg i1d** je skryté pole a slouží k zápisu parametrů pro odeslání do DSC – obr. 5.9. Při změně jakéhokoliv parametru se přepíše hodnota na daném místě v poli a poté se odešle, viz obr. 5.20.



Obr. 5.9 Zadávání parametrů regulátorů

Aby ve výběru zobrazovaných proměnných na DA výstupech byly celé názvy a ne jen proměnné, musí být použit blok, který je na obr. 5.12. Tento blok se skládá z pole názvů proměnných, subVI, zápisu do výběrového menu a 2D pole. V 2D poli jsou umístěny proměnné a přiřazeny názvy, které se mohou zobrazit ve výběru analogového výstupu. SubVI vyhledá v názvech proměnných veličiny, které se mají zobrazit, a přepíše jejich název se stejným pořadovým číslem uvedeným ve vlastnostech menu. Na předním panelu vpravo od zadávání/vyčítání parametrů je uvedena tabulka (obr. 5.13), do které se v levém sloupci zapisují přepsané názvy a v pravém sloupci jsou umístěny názvy proměnných, které se mají zobrazit při výběru.

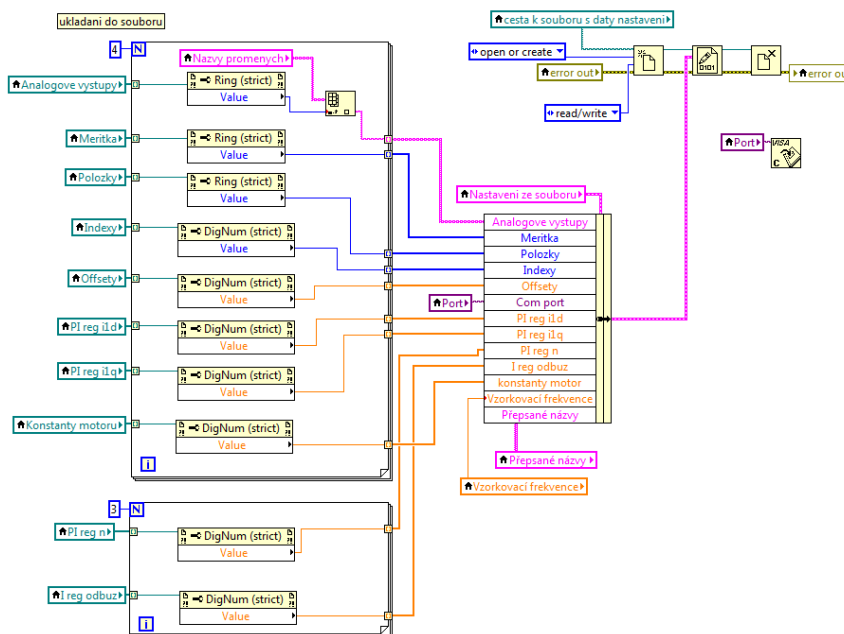


Obr. 5.12 Zápis názvu do výběru analogového výstupu

Přepsané názvy	Zobrazované proměnné pro DA
Proud iId	Id
Skutečná rychlost	rychlost
Žádaná rychlost	zadrych

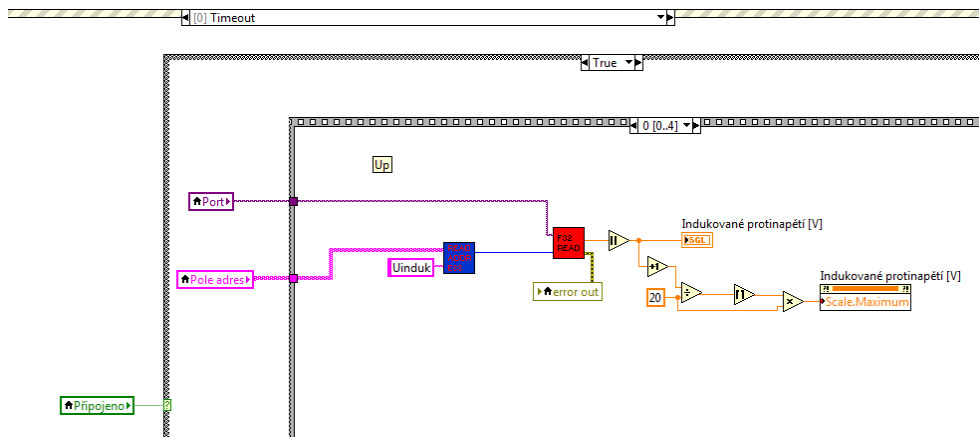
Obr. 5.13 Přední panel – názvy pro analogové výstupy

Ukládání hodnot do souboru řeší blok uvedený na obr. 5.14. Ukládají se hodnoty všech regulátorů, konstant motoru, port, vzorkovací frekvence, přepsané názvy a nastavení DA výstupů. Hodnoty se vyčítají pomocí 2 cyklů for, které jsou rozděleny podle počtu prvků v poli. Jeden provede 4 cykly a druhý 3 cykly. Soubor se uloží do stejného souboru, ze kterého se vyčítá, tedy LV_PMSM.dat a komunikace se ukončí.



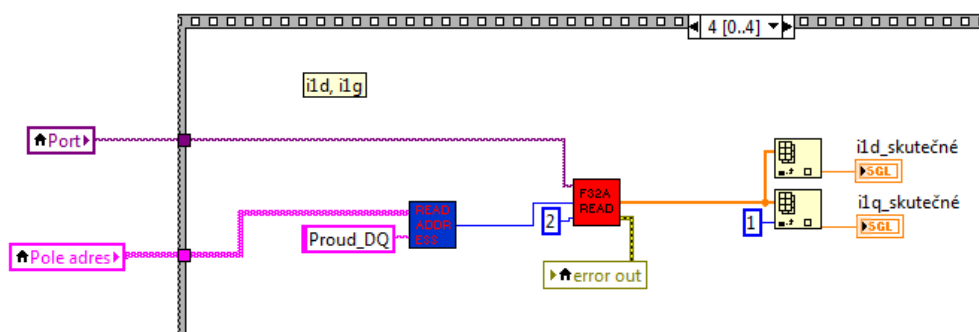
Obr. 5.14 Ukládání do souboru

Tato vytvořená aplikace je založena na principu Event structure, což je struktura, která se provede při změně daného parametru. Každý editační prvek má svoji strukturu, takže při změně parametru se odešle do kontroléru jen změněná hodnota. Je zde jedna záložka s názvem Timeout, která se provádí neustále, nezávisle na jakékoliv změně, proto se používá na vyčítání dat z kontroléru. Z DSC se vyčítá 5 parametrů – indukované napětí [V], vzorkovací frekvence [kHz], napětí meziobvodu [V], rychlost [ot⁻¹] a skutečný proud i_{id} , i_{iq} [A]. Všechny záložky jsou provedeny jen v případě úspěšného připojení ke kontroléru. Po splnění této podmínky se provede blokový diagram uvnitř struktury. Záložka Timeout se skládá ze 4 podzáložek, ze kterých se zvláště vyčítají parametry. Na obr. 5.15 je znázorněno vyčítání indukovaného napětí z kontroléru. Nejprve se přečte adresa z pole adres, a následně se z této adresy vyčte hodnota indukovaného napětí. Je zde i funkce, která mění napětíový rozsah indikátoru po 20V.



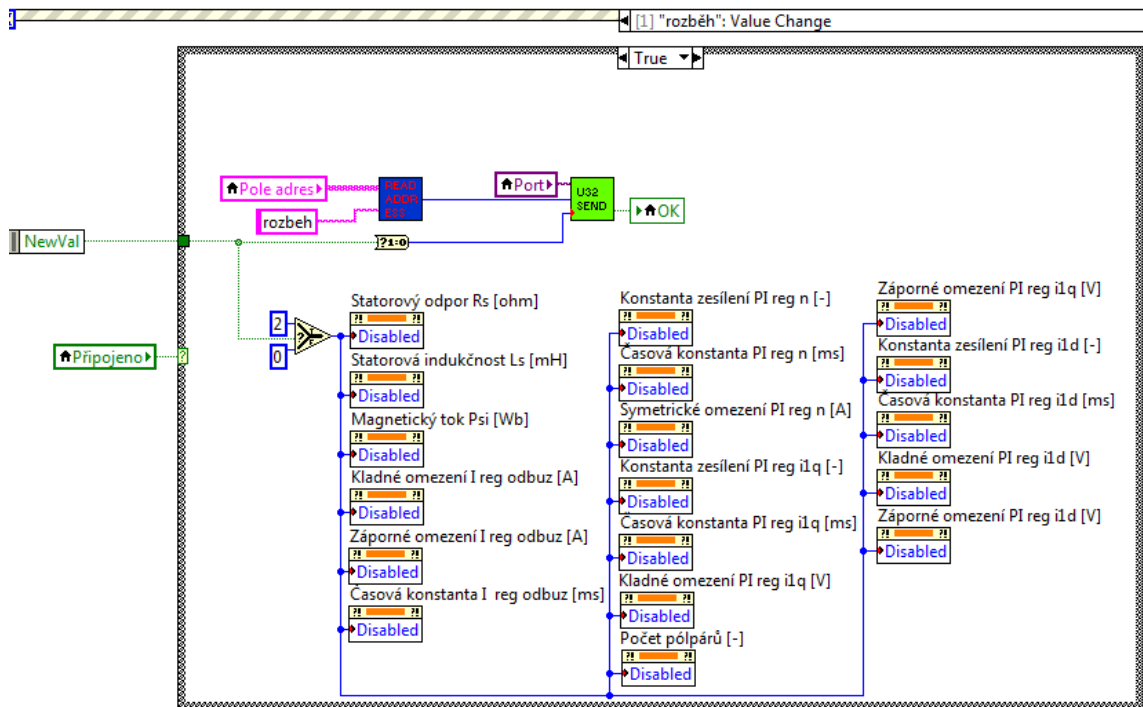
Obr. 5.15 Vyčítání indukovaného protinapětí

Stejným způsobem funguje i vyčítání vzorkovací frekvence a napětí meziobvodu. Rozsah napěťového měřítka u indikátoru napětí meziobvodu se přepíná po 100V. Vyčítání proudů i_{1d} a i_{1q} se provádí zadáním adresy a počtu vyčítaných prvků, v tomto případě to je 2. Na nulté pozici je umístěn proud i_{1d} a na první pozici i_{1q} .



Obr. 5.16 Vyčítání proudů i_{1d} a i_{1q}

Stisknutím tlačítka rozběh se odešle do DSC hodnota 1, při opětovném stisknutí tlačítka (vypnutí) se odešle hodnota 0. V případě spuštění tlačítka **rozběh** již není možné zadávat parametry regulátorů a konstant motoru a controly se skryjí. Na obr. 5.17 je ve spodní části vidět ovládací prvky, které se skryjí.

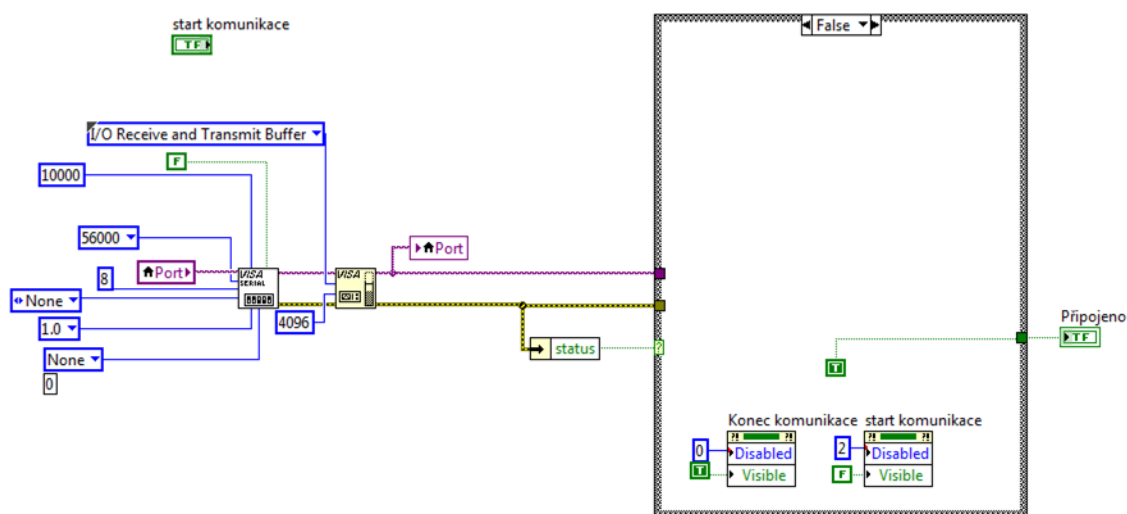


Obr. 5.17 Blokový diagram tlačítka rozběh

V odesílaném poli zobrazeném na obr. 5.18 je na nulté pozici uložena informace o přepnutí mezi žádaným proudem i_{1q} a žádanou rychlostí. Pokud je hodnota rovna 1, bere se v potaz parametr i_{1q} . Pokud hodnota je rovna 0, pak se bere v úvahu zadávání požadované rychlosti. V případě stisku tlačítka *reverzace* se hodnoty proudu i_{1q} a rychlosti znegují, přepíší se v controlech a odešlou se do DSC. Na první pozici pole je uložena hodnota i_{1q} a na druhé pozici žádaná rychlost. Přepínáním mezi těmito dvěma parametry se skrývají a odkrývají zadávací prvky.

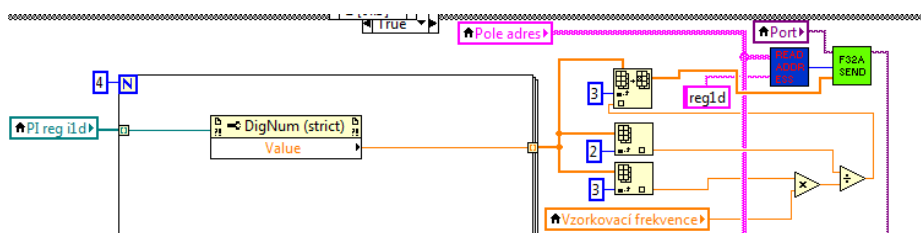
Stisknutím tlačítka **Připoj** se odešlou všechny hodnoty do kontroléru, např. parametry regulátorů, konstant motoru, rychlost, tokotvorného a momentotvorného proudu a nastavení analogových výstupů. Všechna tato prvotní odeslání jsou umístěna v záložce **start komunikace**, podokně **1**, **2** a **3**.

V LabVIEW je potřeba nastavit parametry přenosu, které se nastavují v záložce **start komunikace** - okně **0**. Konkrétně to je: port, přenosová rychlost - 56000Bd, počet bitů - 8, parita - není, stop bit - 1 a další parametry, které jsou znázorněny na obr. 5.22. Na předním panelu se nastavuje jen port, ostatní parametry jsou konstantně nastaveny.



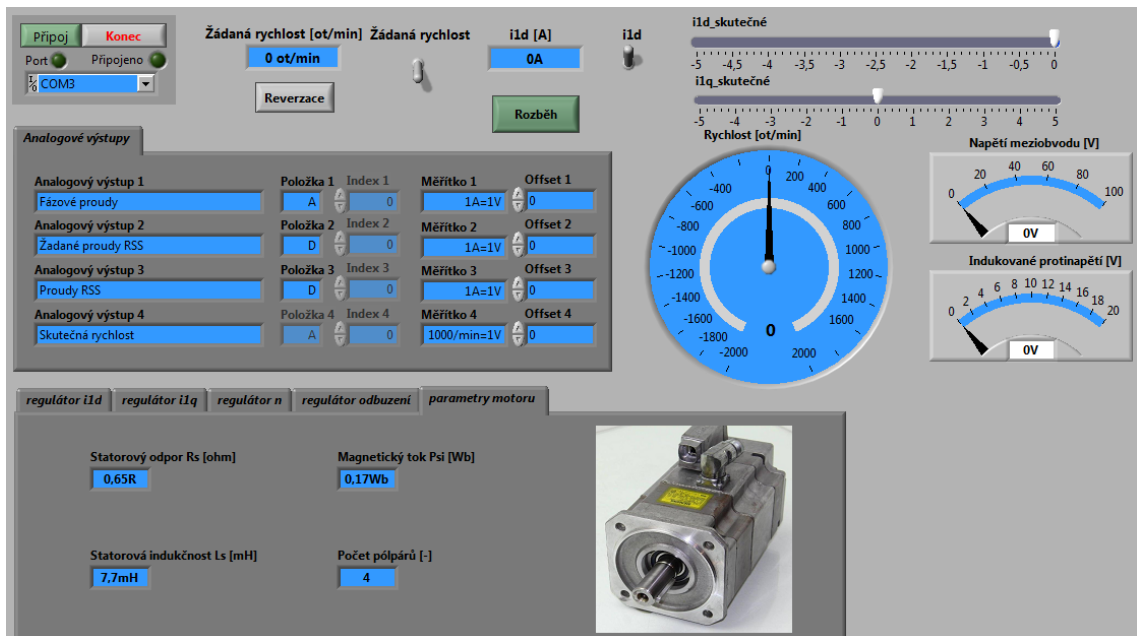
Obr. 5.22 Nastavení komunikace

V okně **2** je umístěno prvotní odeslání do DSC, které se provede při stisku tlačítka **Připoj**. Na obr. 5.23 lze vidět odeslání nastavení regulátoru, pole je o velikosti čtyř prvků – viz obr. 5.20. Ze zadávaných členů se odebere člen časová konstanta a nahradí se konstantou integrátoru podle rovnice 5.17. Stejným způsobem jsou řešeny všechny regulátory.



Obr. 5.23 Prvotní odeslání

Po spuštění aplikace a následně pro připojení k DSC se musí stisknout tlačítko **Připoj**. Při správně nastaveném portu se rozsvítí indikační LED dioda s názvem **Port**. Jakákoliv změna parametrů, např. nastavení regulátoru, se zapíše do paměti v DSC a při správném zápisu se rozsvítí LED dioda **Připojeno**. V případě ukončení komunikace a aplikace se stiskne tlačítko **Odpoj**, a poté se aplikace ukončí stisknutím **Konec**. Při dalším spuštění aplikace se nastaví port automaticky na port v předchozím spuštění aplikace. V dolní části uživatelského prostředí je možnost zadávat parametry do regulátorů a motoru. Na pravé straně je ukazatel napětí meziobvodu, u kterého se přepíná rozsah automaticky po 100V, a ukazatel indukovaného protinapětí, rozsah se přepíná po 20V. Rychlostní ukazatel uvádí rychlost v obou směrech. Ve vrchní části okna se nacházejí 2 přepínače, kterými je možno přepínat mezi **žádanými otáčkami/momentotvorným proudem** a **maximálním indukovaným protinapětí/tokotvorným proudem**.



Obr. 5.24 Přední panel PMSM

Aplikace LabVIEW pro zobrazení hodnot např. rychlosti, napětí meziobvodu a dalších, musí vyčítat z kontroléru proměnné uložené na dané adrese. Nastavení regulátoru apod. probíhá formou odeslání hodnoty na dané paměťové místo. Proto musí být v kontroléru nadefinovány následující proměnné:

```

Uint16 rozbeh = 0; // Rozběhnutí motoru
float32 fvz = 0; // Vzorkovací frekvence [kHz]
float32 Ud = 0; // Měřené napětí meziobvodu [V]
float32 Uinduk = 0; // Indukované protinapětí
float32 rychlost = 0; // Skutečná rychlost [ot/min]
sRSS Proud_DQ = {0, 0}; // Pole hodnot proudů i1d a i1q
float32 Id_bezRu[2] = {0, 0}; // Informace o přepnutí a hodnota i1d
float32 regn[3] = {0, 0, 0}; // Parametry regulátoru otáček
float32 regodb[3] = {0, 0, 0}; // Parametry regulátoru odbuzení
float32 konst[4] = {0, 0, 0, 0}; // Konstanty motoru
float32 Iq_bezRn[3] = {0, 0, 0}; // Pole hodnot i1q a žádané rychlosti
float32 reg1d[4] = {0, 0, 0, 0}; // Parametry regulátoru i1d
float32 reg1q[4] = {0, 0, 0, 0}; // Parametry regulátoru i1q

```

6. Řízení výstupního napětí nepřímého měniče frekvence

Oblast použití nepřímých měničů kmitočtů se čím dál víc rozšiřuje díky zlepšení polovodičových součástek, které poskytují jak vyšší napětí a proud, tak i dobré dynamické vlastnosti. Hlavní výhoda moderních výkonových měničů spočívá ve velké účinnosti, malých rozměrech a nízké hmotnosti. Měnič se skládá ze šesti spínacích prvků, u nichž jsou horní a dolní spínače řízeny komplementárně, tj. je-li horní spínač sepnutý, musí být spodní spínač rozepnutý a naopak. Protože doba vypnutí je delší než doba zapnutí, musí být nějaké časové zpoždění mezi vypnutím jednoho a zapnutím druhého tranzistoru v polo-můstku. Řízení probíhá pomocí různých typů pulsně-šířkové modulace (PWM), což je základní metoda řízení u výkonových polovodičových měničů. U moderních měničů bývá spínací frekvence PWM od několika kHz (řízení motoru) až do několika MHz (rezonanční měniče).

6.1. Komparační PWM

Tento typ PWM je založen na komparaci referenčního sinusového signálu s trojúhelníkovým, popř. pilovým nosným signálem. Při porovnání společného nosného signálu u_t s třemi referenčními sinusovými signály u_{ra} , u_{rb} , u_{rc} vzájemně posunutými o 120° vznikne signál definující spínací okamžiky tranzistorů. Frekvenční modulační poměr je dán vztahem:

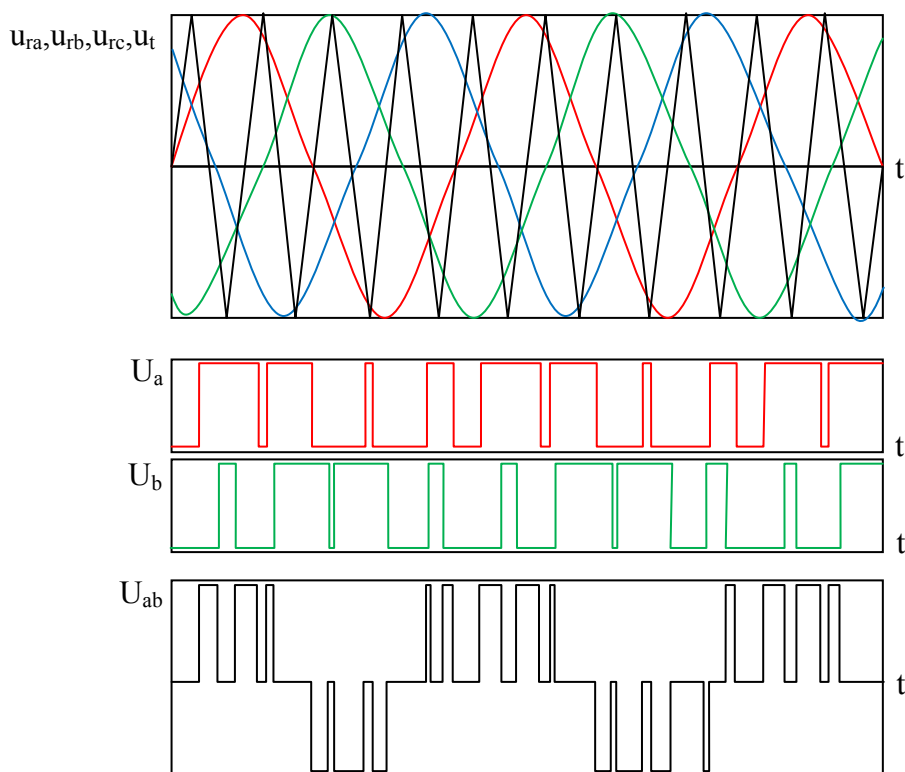
$$m_f = \frac{f_s}{f_1} \quad (6.1)$$

Kde f_s udává spínací frekvenci a f_1 frekvenci referenčního napětí.

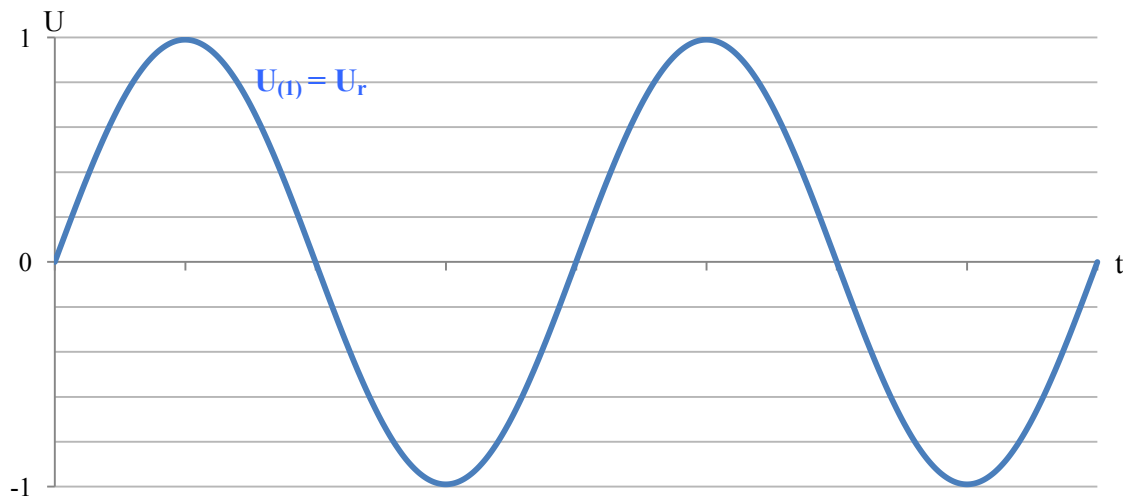
Modulační poměr se určí ze vztahu 6.2, který může nabývat hodnot 0 – 1.

$$m = \frac{U_{(1)}}{U_T} \quad (6.2)$$

$U_{(1)}$ – maximální referenční napětí, U_T – maximální trojúhelníkové napětí. [9]



Obr. 6.1 Průběhy komparační PWM



Obr. 6.2 Referenční napětí

Velkou nevýhodou této modulace je nedosažení maximálního napětí meziobvodu U_D . Maximální dosažitelná amplituda může dosahovat:

$$U_1 = 0,5 \cdot U_D \quad [7] \quad (6.3)$$

6.2. Komparační PWM s přidáním třetí harmonické

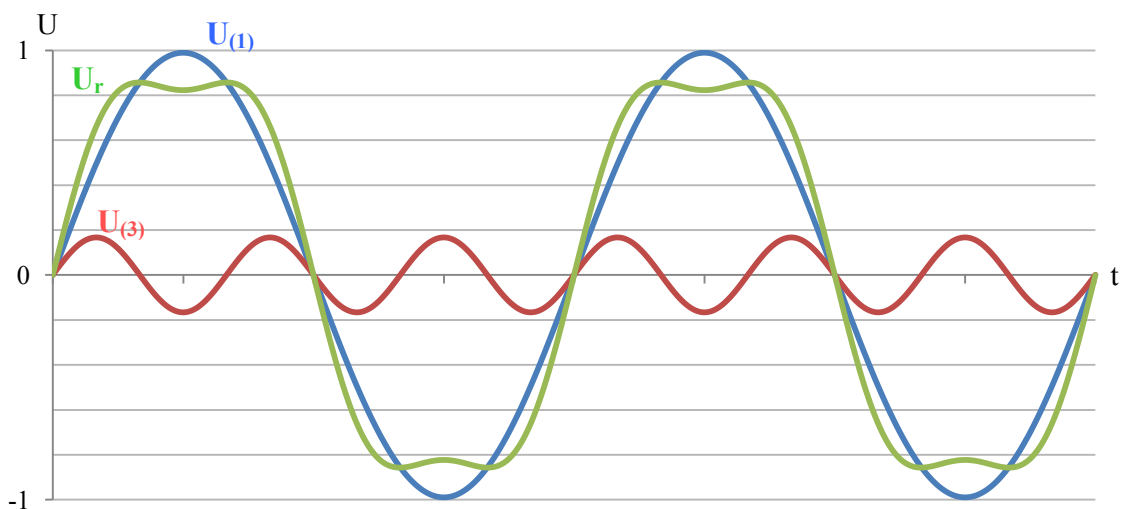
Pokud není do zátěže měniče připojený nulový vodič, může se ke komparační PWM přičíst signál 3. harmonické. Pro dosažení nejoptimálnějšího využití napěťového meziobvodu se používá napětí:

$$U_{(3)} = \frac{U_{(1)}}{6} \quad (6.4)$$

Pomocí přičtení napětí třetí harmonické o 16,6% do původního signálu se zvýší napěťové využití meziobvodu o 15,4%. Poté bude možné dosažené napětí:

$$U_1 = 0,577 \cdot U_D \quad (6.5)$$

Výhoda kromě vyššího napěťového využití spočívá i ve snížení vyšších harmonických. Průběh je zobrazen na obr. 6.3. Je zde znázorněna základní harmonická, třetí harmonická a jejich výsledný součet. [7]



Obr. 6.3 Referenční napětí s přidáním 3. harmonické

6.3. Vektorová PWM

Princip vektorové PWM spočívá v realizaci vypočtené velikosti a polohy prostorového vektoru statorového napětí. Vektorová PWM se stala díky svým vlastnostem velmi používanou. Třífázový dvouúrovňový střídač poskytuje 8 možných spínacích stavů, které jsou složeny z 6 aktivních vektorů ($\vec{u}_1 - \vec{u}_6$) a 2 nulových vektorů (\vec{u}_0 a \vec{u}_7). Aktivní vektory rozdělí prostor do 6 sektorů, kde se žádaný vektor \vec{u}^* získá sepnutím dvou sousedních vektorů. Na obr. 6.4 je vidět, že vektor \vec{u}^* je možné provést různými zapínacími a vypínacími sekvencemi \vec{u}_1 a \vec{u}_2 . Maximální velikost vektoru \vec{u}^* je:

$$\vec{u}^*_{\max} = \frac{U_D}{\sqrt{3}} \quad (6.6)$$

Velikost aktivních vektorů a nulových vektorů je dána vztahem 6.7 a 6.8:

$$\vec{u}_{1-6} = \frac{2}{3} \cdot U_D \quad (6.7)$$

$$\vec{u}_{0,7} = 0 \quad (6.8)$$

Oproti komparační PWM nejsou zde oddělené modulátory pro jednotlivé fáze. Žádaný vektor \vec{u}^* je vzorkován s pevnou frekvencí hodinového signálu:

$$2f_s = \frac{1}{T_s} \quad (6.9)$$

Realizace v mikroprocesoru se popisuje pomocí trigonometrického vztahu (6.10 a 6.11) pro první sektor, a poté se přepočtou do ostatních sektorů.

$$t_1 = \frac{2\sqrt{3}}{\pi} MT_s \cdot \sin\left(\frac{\pi}{3} - \rho\right) = \frac{2\sqrt{3}}{\pi} \cdot \frac{|\vec{u}^*|}{\frac{2}{3}U_D} \cdot T_s \cdot \sin\left(\frac{\pi}{3} - \rho\right) = \sqrt{3} \cdot \frac{|\vec{u}^*|}{U_D} \cdot T_s \cdot \sin\left(\frac{\pi}{3} - \rho\right) \quad (6.10)$$

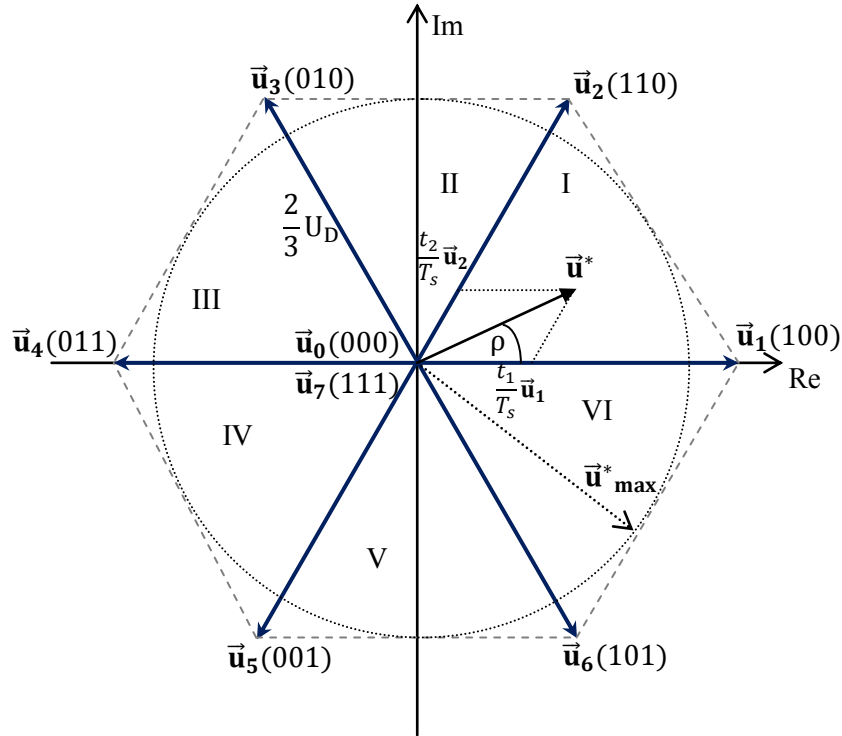
$$t_2 = \frac{2\sqrt{3}}{\pi} MT_s \cdot \sin \rho = \frac{2\sqrt{3}}{\pi} \cdot \frac{|\vec{u}^*|}{\frac{2}{3}U_D} \cdot T_s \cdot \sin \rho = \sqrt{3} \cdot \frac{|\vec{u}^*|}{U_D} \cdot T_s \cdot \sin \rho \quad (6.11)$$

Po výpočtu času t_1 a t_2 , zbývající čas patří nulovým vektorům \vec{u}_0 a \vec{u}_7 . Podmínkou je, že:

$$t_1 + t_2 \leq T_s \quad (6.12)$$

Celková doba trvání nulových vektorů pak musí být dána vztahem:

$$t_{0,7} = T_s - t_1 - t_2 = t_0 + t_7 \quad (6.13)$$



Obr. 6.4 Vektory napětí

Tab. 6.1 Tabulka spínání vektorů a výsledné fázové napětí

vektor	a	b	c	U_a	U_b	U_c	U_{AB}	U_{BC}	U_{CA}
\vec{u}_0	0	0	0	0	0	0	0	0	0
\vec{u}_1	1	0	0	$\frac{2}{3}U_D$	$-\frac{1}{3}U_D$	$-\frac{1}{3}U_D$	U_D	0	$-U_D$
\vec{u}_2	1	1	0	$\frac{1}{3}U_D$	$\frac{1}{3}U_D$	$-\frac{2}{3}U_D$	0	U_D	$-U_D$
\vec{u}_3	0	1	0	$-\frac{1}{3}U_D$	$\frac{2}{3}U_D$	$-\frac{1}{3}U_D$	$-U_D$	U_D	0
\vec{u}_4	0	1	1	$-\frac{2}{3}U_D$	$\frac{1}{3}U_D$	$\frac{1}{3}U_D$	$-U_D$	0	U_D
\vec{u}_5	0	0	1	$-\frac{1}{3}U_D$	$-\frac{1}{3}U_D$	$\frac{2}{3}U_D$	0	$-U_D$	U_D
\vec{u}_6	1	0	1	$\frac{1}{3}U_D$	$-\frac{2}{3}U_D$	$\frac{1}{3}U_D$	U_D	$-U_D$	0
\vec{u}_7	1	1	1	0	0	0	0	0	0

Kombinace zapnutí a vypnutí výkonových spínačů pro každý vektor napětí je naznačen na obr. 6.4 třímístným číslem. Každá z těchto číslic reprezentuje jednu fázi. Pokud je na určitém místě číslo 1, je sepnutý vrchní spínač a spodní je rozepnutý. V případě hodnoty 0 je sepnutý spodní spínač a horní je rozepnutý. Fázové napětí a velikost vektoru napětí jsou uvedeny v tab. 6.1. [9]

Maximální amplituda základní harmonické fázového napětí dosahuje:

$$U_1 = 0,577 \cdot U_D \quad (6.14)$$

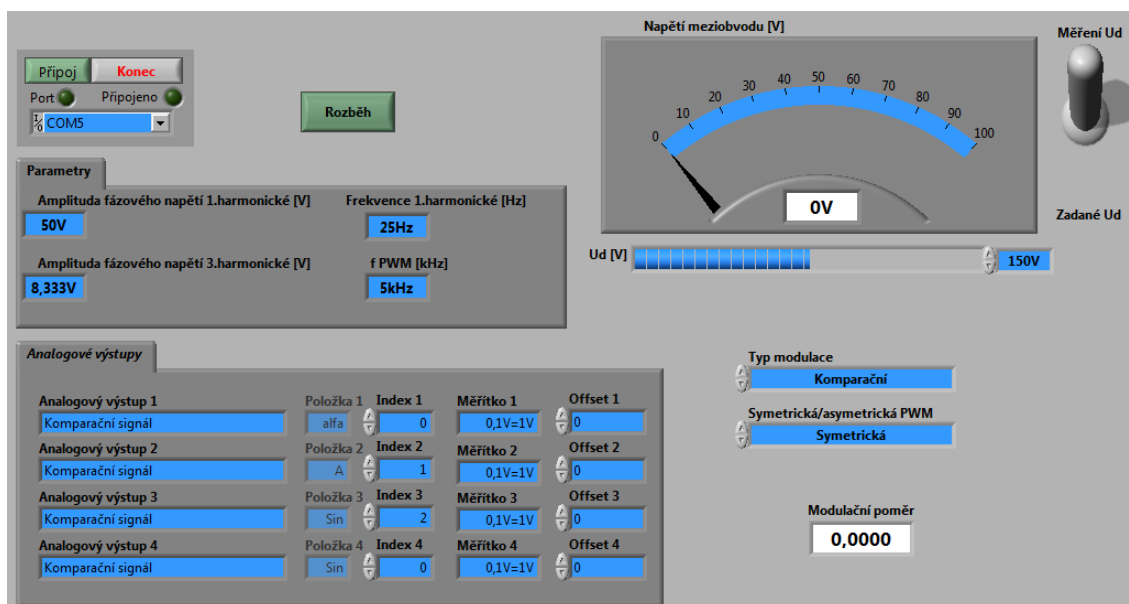
Amplituda je stejná jako u komparační PWM s přidáním 3. harmonické. Velkou výhodou je menší kolísání momentu, nižší ztráty ve vinutí a nižší obsah harmonických proudů. Tato metoda zaručí dynamičtější změny a dá se aplikovat pouze na rychlé mikroprocesory. [7]

6.4. Uživatelské prostředí

Řízení výstupního napětí střídače s napětovým meziobvodem se provádí pomocí této vytvořené aplikace, která je principiálně podobná aplikaci pro řízení synchronního motoru. Po spuštění aplikace se načte poslední konfigurace. Poté se vybere port komunikace, pokud není stejný port jako minule a klikne se na tlačítko **Připojit**. Úspěšné připojení zařízení indikuje LED dioda **Port**. Indikační prvek **Připojeno** se rozsvítí, pokud odeslání jakéhokoliv parametru proběhne v pořádku.

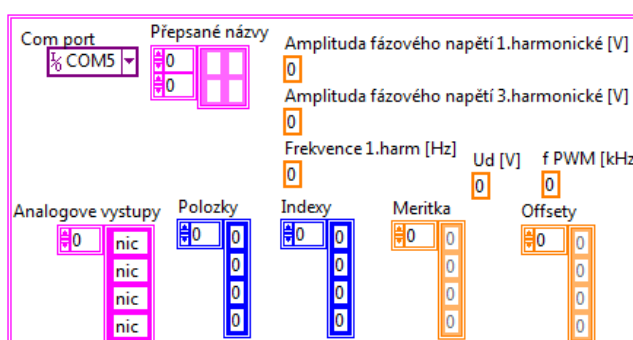
V případě změny jakéhokoliv parametru, odešle LabVIEW okamžitě daný parametr do kontroléru. Parametry, jež se mohou měnit:

- 1) **amplituda fázového napětí 1. harmonické**, udávána ve V,
- 2) **amplituda fázového napětí 3. harmonické**, udávána ve V, pro nejvyšší možné využití meziobvodu se udává šestinová amplituda oproti 1. Harmonické,
- 3) **frekvence 1. harmonické**, udávána v Hz,
- 4) **frekvence PWM**, udávána v kHz,
- 5) **U_d**, udává se ve V a slouží pro ruční zadání napětí meziobvodu – modulační poměr se nepočítá ze skutečného napětí meziobvodu,
- 6) **typ modulace** – vybere typ modulace,
- 7) **symetrická/asymetrická PWM** – výběr mezi symetrickou a asymetrickou PWM.



Obr. 6.5 Přední panel pro řízení výstupního napětí

Ve starším řídicím systému, kde byl použit signálový kontrolér TMS320F2812, probíhala komunikace PC z DSC pomocí jednoho pole parametrů, které se posílalo do kontroléru nebo PC. Do kontroléru se odesílalo pole o 12-ti prvcích a počítač vyčítal jen jeden prvek, a to napětí v meziobvodu U_d . Tento starší systém neměl kontrolu CRC, tudíž při přenosu mohlo docházet k chybám. Nový systém, popsáný v kap. 5.2 je řešen podobným způsobem jako aplikace pro řízení synchronního motoru. Jsou zde taky vyčítané parametry ze souboru, tentokrát pod názvem LV_pwm.dat. Vyčítají se jen zadávané parametry, které jsou uvedeny na obr. 6.6. Tyto parametry se po spuštění aplikace načtou a poté se může připojit k DSC.



Obr. 6.6 Nastavení ze souboru

Aby aplikace mohla správně fungovat, musí být přiloženy soubory main.c a Project.map. Z těchto souborů se importují proměnné používané v kontroléru. Tyto soubory se generují s nahráním programu do kontroléru. Aplikace je připravená pro tvorbu programu v C, program pro DSC je naplní jiné diplomové práce a zatím nebyl vytvořen.

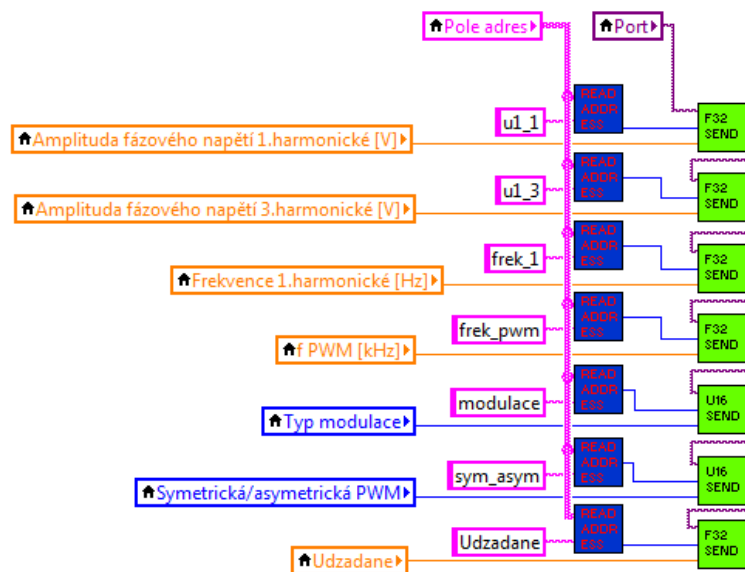
Při vytváření programu se musí nadefinovat v Code Composer studiu proměnné, na které bude LabVIEW odesílat nebo přijímat data. Proměnné se musí definovat v takovémto tvaru:

```
float32 u1_1 = 0; // Amplituda fázového napětí 1. harmonické [V]
float32 u1_3 = 0; // Amplituda fázového napětí 3. harmonické [V]
float32 frek_1 = 0; // Frekvence 1. harmonické [Hz]
float32 frek_pwm = 0; // f PWM [kHz]
float32 Udzadane = 0; // Zadané Ud, bez měření Ud [V]
Uint16 modulace = 0; // Výběr typu modulace
Uint16 sym_asym = 0; // Výběr Symetrické a Asymetrické PWM
Uint16 rozbeh = 0; // Rozběhnutí motoru
float32 Ud = 0; // Měřené napětí meziobvodu
sABC komp_signal[3] = {0, 0, 0}; // Komparační signál fáze a, b, c
```

Stisknutím tlačítka rozbeh se odešle do proměnné *rozbeh* hodnota 1, jinak je zde hodnota 0. Přepnutí přepínače do polohy *zadané Ud* se do kontroléru odešle do proměnné *Udzadane* hodnota zadaného napětí, pokud je přepínač v poloze *měření Ud*, odesílá se nulová hodnota.

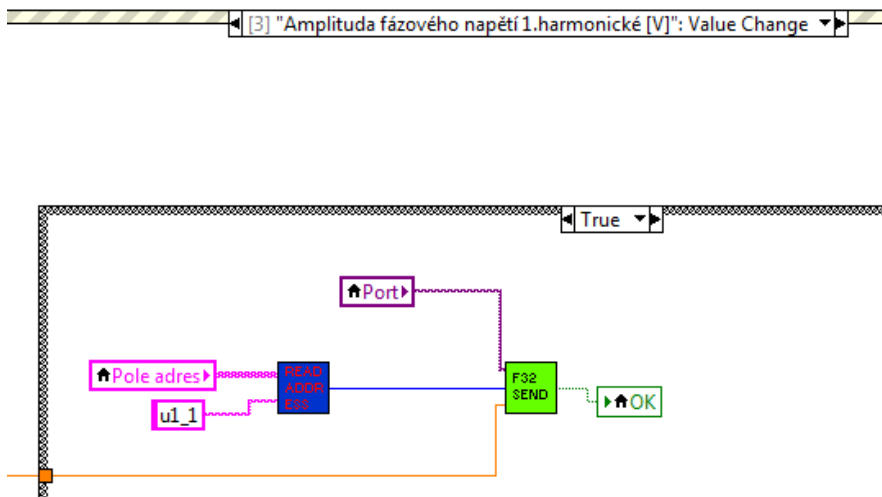
Výběr modulace se provádí mezi třemi typy modulace. Je to komparační PWM, komparační PWM s přidáním 3. harmonické a vektorová PWM. Výběrem konkrétní modulace se odešle do proměnné *modulace* hodnota v rozmezí 0 - 2. V případě výběru komparační PWM se odešle hodnota 0. Pokud se vybere komparační PWM s 3. harmonickou, odešle se hodnota 1. Poslední možností je výběr vektorové modulace, což je hodnota 2. Změna symetrické a asymetrické PWM se projeví v proměnné *sym_asym*, kde se zapíše hodnota 0 (symetrická) nebo 1 (asymetrická). Symetrická PWM komparuje referenční signál s trojúhelníkovým signálem a asymetrická PWM komparuje referenční signál s pilovým signálem.

Stisknutím tlačítka *Přípoj* se provede prvotní odeslání dat do kontroléru, kam se odešle nastavení DA převodníků a všechny zadávané parametry. Prvotní odeslání je znázorněno na obr. 6.7.



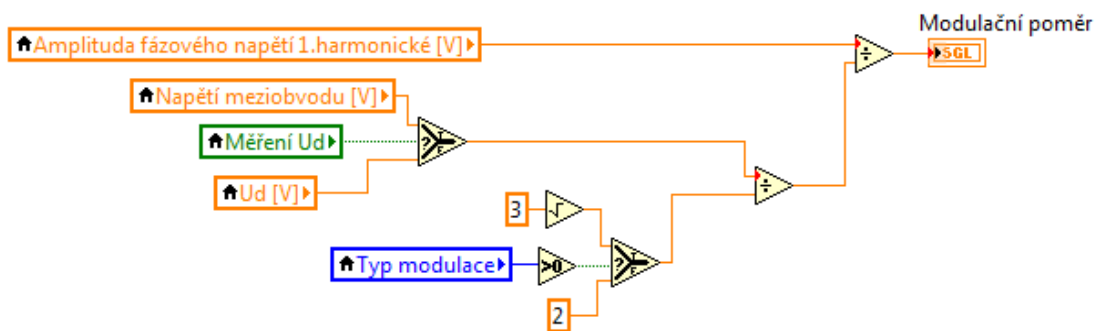
Obr. 6.7 Prvotní odeslání

Následné odesílání probíhá jen při změně jakéhokoliv parametru. Např. při změně amplitudy fázového napětí 1. harmonické se odešle jen tato hodnota, na obr. 6.8 je znázorněno blokové schéma odeslání jedné proměnné.



Obr. 6.8 Odesílání proměnné

Blokový diagram pro výpočet modulačního poměru je znázorněn na obr. 6.9. Je zde vidět výpočet modulačního poměru m s ohledem na výběr typu modulace. U komparační PWM se podělí napětí meziobvodu dvěma. V případě výběru komparační PWM s 3. harmonickou nebo vektorové PWM se dělí napětí $\sqrt{3}$. Další možností výběru je buď *měřené Ud* nebo *zadané Ud*. Modulační poměr se tedy počítá ze skutečného napětí nebo ze zadaného napětí meziobvodu.



Obr. 6.9 Výpočet modulačního poměru

Přepsané názvy	Zobrazované proměnné pro DA
Komparační signál	komp_signal

Obr. 6.10 Přepsané názvy pro DA převodník

7. Vektorové řízení asynchronního motoru

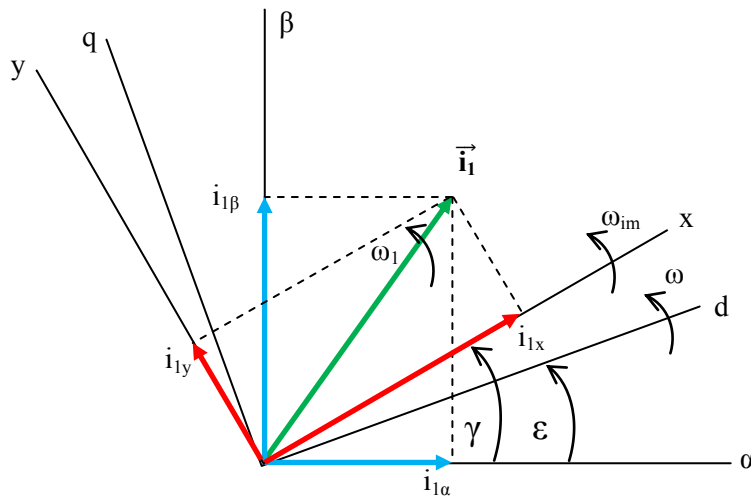
7.1. Popis struktury

Princip vektorového řízení spočívá v rozložení prostorového vektoru statorového proudu do dvou kolmých složek v rotujícím souřadnicovém systému (OSS), který může být orientován na prostorový vektor magnetického toku. Složky prostorového vektoru statorového proudu pak určují moment a magnetizaci stroje. Momentotvorná složka vektoru statorového proudu určuje společně s příslušným vektorem magnetického toku moment stroje. Magnetizační složka ležící ve směru vektoru magnetického toku ovlivňuje magnetizaci motoru. [10]

Transformace ze složek i_{1a} a i_{1b} do i_{1x} a i_{1y} se spočte podle rovnice 5.1 a 5.2. Poté se proudy i_{1x} a i_{1y} přepočtou pomocí úhlu γ na orientovaný souřadný systém na i_{1x} a i_{1y} podle vztahu 7.1 a 7.2. Z obr. 7.1 je patrné odvození výpočtu.

$$i_{1x} = i_{1\alpha} \cdot \cos \gamma + i_{1\beta} \cdot \sin \gamma \quad (7.1)$$

$$i_{1y} = -i_{1\alpha} \cdot \sin \gamma + i_{1\beta} \cdot \cos \gamma \quad (7.2)$$



Obr. 7.1 Vektorový diagram statorového proudu

Výstupní veličina z regulátorů R_{i1x} a R_{i1y} je napětí u_{1xR} a u_{1yR} . V bloku zrušení vazby se spočtou napětí u_{1xe} a u_{1ye} podle vztahů 7.5 a 7.6.

$$u_{1xR} = R_S i_{1x} + \sigma L_S \frac{di_{1x}}{dt} \quad u_{1yR} = R_S i_{1y} + \sigma L_S \frac{di_{1y}}{dt} \quad (7.3) \quad (7.4)$$

$$u_{1xe} = -\omega_{im}\sigma L_S i_{1y} \quad u_{1ye} = \omega_{im}[\sigma L_S i_{1x} + (1 - \sigma) \cdot L_S i_m] \quad (7.5) (7.6)$$

Ke zkvalitnění regulací je vhodné přičíst k výstupním veličinám regulátorů proudů složky napětí.

$$u_{1x} = u_{1xR} + u_{1xe} = R_s i_{1x} + \sigma L_S \frac{di_{1x}}{dt} - \omega_{im}\sigma L_S i_{1y} \quad (7.7)$$

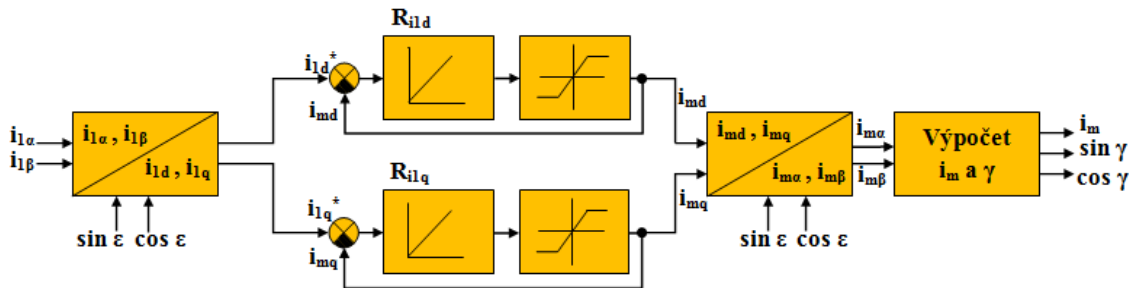
$$u_{1y} = u_{1yR} + u_{1ye} = R_s i_{1y} + \sigma L_S \frac{di_{1y}}{dt} + \omega_{im}[\sigma L_S i_{1x} + (1 - \sigma) \cdot L_S i_m] \quad (7.8)$$

Převod z orientovaného souřadného systému do statorového souřadného systému:

$$u_{1\alpha} = u_{1x} \cdot \cos \gamma - u_{1y} \cdot \sin \gamma \quad (7.9)$$

$$u_{1\beta} = u_{1x} \cdot \sin \gamma + u_{1y} \cdot \cos \gamma \quad (7.10)$$

Vyhodnocení magnetizačního proudu i_m a orientující veličiny $\sin \gamma$ a $\cos \gamma$ realizuje blok **vyhodnocení i_m a γ** zobrazený na obr. 7.2. Tyto veličiny se spočtou podle rovnic 7.11, 7.12 a 7.13. Struktura řízení AM je realizována pomocí proudového modelu, tudíž zde musí být použit snímač otáček a adaptace na teplotní změny rotorové časové konstanty. [7]

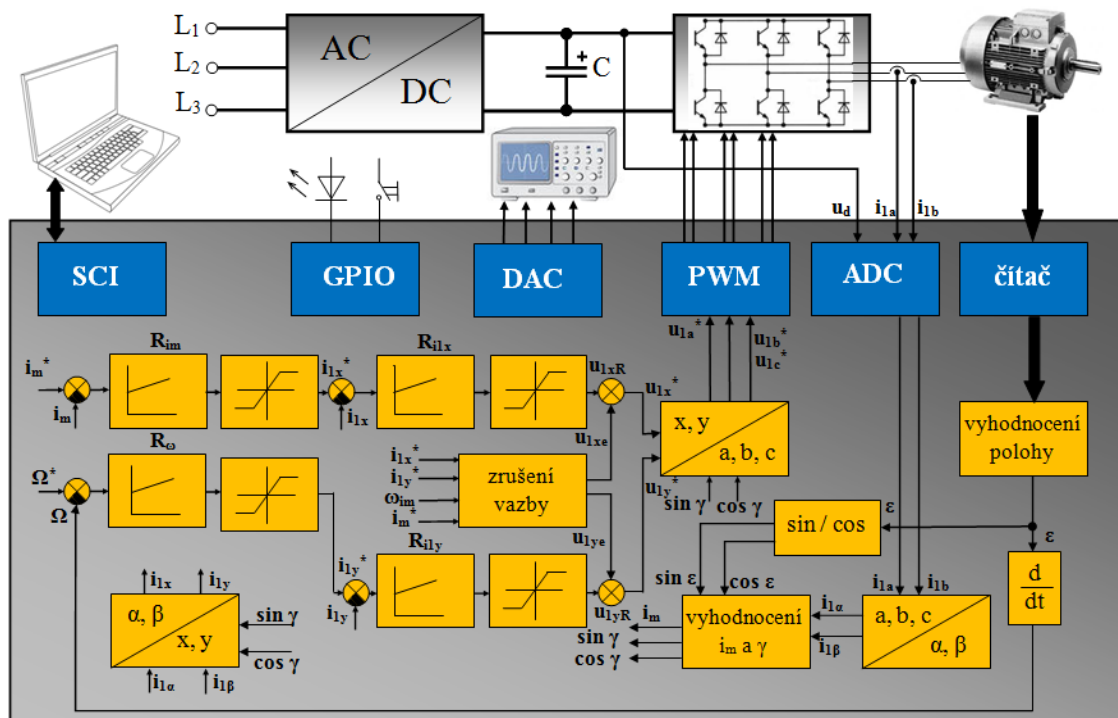


Obr. 7.2 Vyhodnocení i_m a γ

$$i_m = \sqrt{i_{m\alpha}^2 + i_{m\beta}^2} \quad (7.11)$$

$$\sin \gamma = \frac{i_{m\beta}}{i_m} \quad (7.12)$$

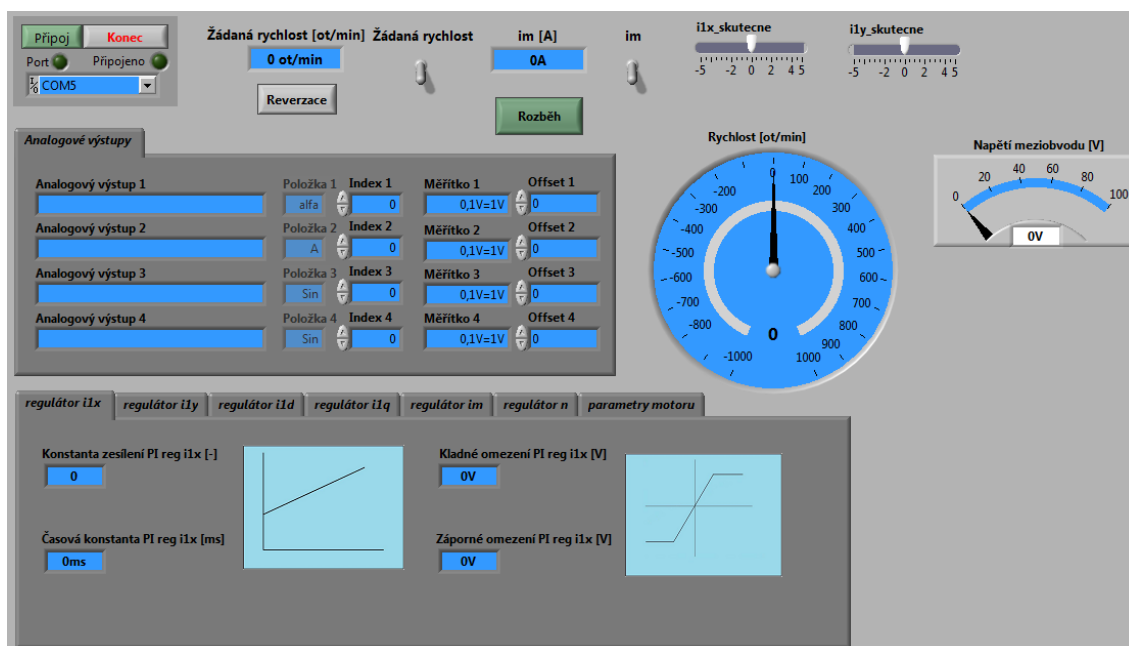
$$\cos \gamma = \frac{i_{m\alpha}}{i_m} \quad (7.13)$$



Obr. 7.3 Struktura regulace AM v OSS

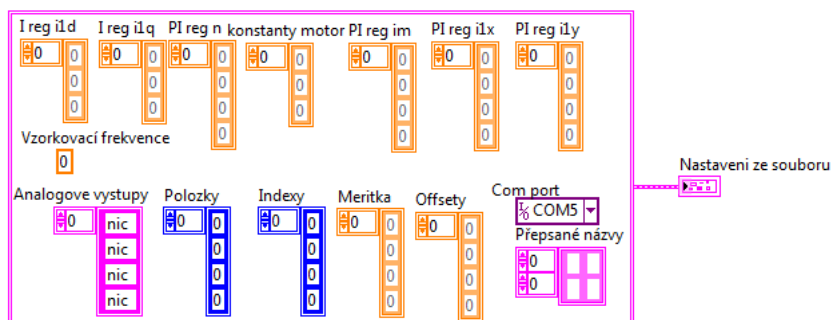
7.2. Uživatelské prostředí

Obdobným způsobem jako řízení synchronního motoru v kap. 5.2 pracuje i aplikace pro vektorové řízení asynchronního motoru. Program se spouští souborem AM.vi a nastavované parametry se ukládají do souboru s názvem LV_AM.dat. Řízení probíhá v orientovaném souřadném systému, tudíž důležité veličiny pro zobrazení jsou hlavně i_{lx} a i_{ly} , zobrazeny na obr. 7.4 vpravo nahoře.



Obr. 7.4 Přední panel AM

Vyčítané a následně ukládané parametry ze souboru jsou zobrazeny na obr. 7.5. Ukládají se všechny parametry regulátorů, konstanty motoru, port, vzorkovací frekvence, názvy DA převodníků a další parametry pro nastavení převodníků.



Obr. 7.5 Nastavení ze souboru

Při vytváření programu se musí nadefinovat v Code Composer studiu proměnné, na které bude LabVIEW odesílat nebo přijímat data. Proměnné se musí definovat v tomto tvaru:


```

Uint16 rozbeh = 0; // Rozběhnutí motoru
float32 Iy_bezRn[3] = {0, 0, 0}; // Pole hodnot  $i_{ly}$  a žádané rychlosti
float32 Ix_bezRim[3] = {0, 0, 0}; // Pole hodnot  $i_{lx}$  a  $i_m$ 
float32 reg1d[3] = {0, 0, 0}; // Parametry regulátoru  $i_{ld}$  – blok vyhodnocení  $i_m$  a  $\gamma$ 
float32 reg1q[3] = {0, 0, 0}; // Parametry regulátoru  $i_{lq}$  – blok vyhodnocení  $i_m$  a  $\gamma$ 
float32 regn[3] = {0, 0, 0}; // Parametry regulátoru otáček
float32 regim[4] = {0, 0, 0, 0}; // Parametry regulátoru magnetizačního proudu
float32 reglx[4] = {0, 0, 0, 0}; // Parametry regulátoru  $i_{lx}$ 
float32 regly[4] = {0, 0, 0, 0}; // Parametry regulátoru  $i_{ly}$ 
float32 konst[3] = {0, 0, 0}; // Konstanty motoru
float32 fvz = 0; // Vzorkovací frekvence [kHz]
float32 Ud = 0; // Měřené napětí meziobvodu [V]
float32 rychlost = 0; // Skutečná rychlost [ot/min]
sOSS Proud_XY[2] = {0, 0}; // Pole hodnot proudů  $i_{lx}$  a  $i_{ly}$ 

```

Pole s názvem ***Iy_bezRn*** se odesílá do kontroléru v poli o velikosti 3 prvků. Na nulté pozici se odesílá informace o stavu přepínače ***Žádaná rychlost/ i_{ly}*** , zda se zadává žádaná rychlost (hodnota 0) nebo momentotvorný proud (hodnota 1). Na první pozici je hodnota momentotvorného proudu a na druhé pozici je hodnota žádaných otáček. Obdobným způsobem je řešený i přepínač ***im/ i_{lx}*** . Pokud na nulté pozici bude hodnota 1, bere se v potaz hodnota tokotvorného proudu, jinak se bere hodnota magnetizačního proudu. Pole těchto hodnot se posílá do proměnné s názvem ***Ix_bezRim***.

Stejným způsobem pracuje vyčítání proměnných i_{lx} a i_{ly} , jako je u komunikace se systémem pro vektorové řízení synchronního motoru na obr. 5.16. Na nulté pozici je hodnota proudu i_{lx} a na první pozici v poli je i_{ly} .

8. Závěr

Důležitým aspektem pro řízení motoru je měnit parametry za běhu programu. K tomu slouží aplikace vytvořené v programu LabVIEW ve verzi 2014, pomocí níž se posílají do řídicího systému potřebné parametry. Z řídicího systému je potřeba i vyčítat hodnoty, např. otáčky nebo napětí meziobvodu. Proto je u řídicí procesorové desky použita USB komunikace. Na řídicí systém TMS320F28335 byly vytvořeny 3 aplikace pro různé řízení, konkrétně na vektorové řízení synchronního motoru s permanentními magnety, řízení výstupního napětí nepřímého měniče frekvence a vektorové řízení asynchronního motoru. Všechny tyto 3 aplikace pracují na stejném principu komunikace. Dále byly vytvořeny 2 aplikace na starší řídicí systém TMS320F2812, jedna aplikace je určena pro řízení výstupního napětí nepřímého měniče frekvence a druhá pro vektorové řízení asynchronního motoru. Všechny tyto aplikace určené pro oba systémy mají podobný vzhled.

Nejprve byla vytvořena aplikace pro řízení synchronního motoru, od které se další dvě aplikace odvíjely. Struktury vektorového řízení synchronního a asynchronního motoru jsou velmi podobné. Synchronní motor se řídí v systému rotorových souřadnic d , q a asynchronní motor v systému orientovaných souřadnic x , y . Tudíž jsou i odesílané parametry podobné. Do řídicí desky synchronního motoru se odesílají parametry regulátorů R_{i1d} , R_{i1y} , R_{odbuz} a R_{ω} , zato do řídicí desky asynchronního motoru se odesílají parametry regulátorů R_{i1x} , R_{i1y} , R_{im} , R_{ω} a do bloku vyhodnocení i_m a γ se odesílají R_{i1d} a R_{i1q} . Aplikace určená pro řízení výstupního napětí slouží k nastavování typu modulace a jejich parametrů. Je to jednodušší aplikace, ve které se nastavují amplitudy, frekvence a výstupy DA převodníků. Po zprovoznění komunikace PC s DSC se následně provedla měření těchto tří řízení.

Díky své totožnosti vzhledů aplikací a řízení těchto dvou typů motorů by se v budoucnu dala vytvořit univerzální aplikace pro komunikaci s řídicím systémem, pomocí níž se bude komunikovat se signálovým kontrolérem a bude zde možnost zadávat parametry pro vektorové řízení synchronního motoru s permanentními magnety, synchronního motoru s cizím buzením a asynchronního motoru.

9. Seznam použité literatury

- [1] **National Instruments.:** LabVIEW User Manual. *www.ni.com*. [Online] 1998. [Citace: 5. Únor 2015.] *www.ni.com*.
- [2] **Travis, Jeffrey, Kring.:** *LabVIEW for everyone: graphical programming made easy and fun, third edition*. Upper Saddle River : Prentice Hall, 2006. ISBN 978-0-13-185672-1.
- [3] **Clark, Cory L.:** *LabVIEW digital signal processing and digital communication*. New York : McGraw-Hill, 2005. ISBN 0-07-146966-4.
- [4] *Cyklický redundantní součet*. [online]. [cit. 2015-03-02]. Dostupné z: http://cs.wikipedia.org/wiki/Cyklický%BD_redundantní%AD_součet
- [5] *Cyclic redundancy check*. [online]. [cit. 2015-03-02]. Dostupné z: http://en.wikipedia.org/w/index.php?title=Cyclic_redundancy_check&redirect=no
- [6] **CAHA, Zdeněk.** *Elektrické pohony*. 1. vyd. Praha: SNTL, 1990, ISBN 80-030-0418-7.
- [7] **BRANDŠTETTER P.:** *Elektrické regulované pohony III*. VŠB - TU Ostrava, 2012.
- [8] **FREESCALE:** *3-Phase PM Synchronous Motor Vector Control Using a 56F80x, 56F8100, or 56F8300 Device: Design of Motor Control Application*. roč. 2005, č. 1, s. 64. Dostupné z: www.freescale.com
- [9] **KAŹMIERKOWSKI, Marian P, R KRISHNAN a Frede BLAABJERG.:** *Control in power electronics: selected problems*. New York: Academic Press, c2002, ISBN 01-240-2772-5.
- [10] **BRANDŠTETTER P.:** *Mikropočítačové řídicí systémy II*. VŠB - TU Ostrava, 2010.

10. Seznam příloh

Příloha č. I. – Vzorový protokol – *Vektorové řízení synchronního motoru s permanentními magnety* – 8 stran

Příloha č. II. – Vzorový protokol – *Řízení výstupního napětí frekvenčního měniče s napěťovým meziobvodem* - 12 stran

Příloha č. III. – Vzorový protokol – *Vektorové řízení asynchronního motoru* - 9 stran

Příloha č. IV. – Příloha na CD/DVD – *diplomová práce, vzorové protokoly, programy*