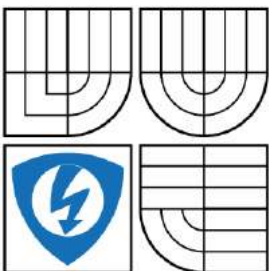




VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

ŘÍZENÍ STŘÍDAVÝCH ELEKTRICKÝCH POHONŮ V PROSTŘEDÍ SIMULINK REAL-TIME

CONTROL OF ELECTRIC DRIVES USING SIMULINK REAL-TIME

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Ján Schrötter

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Lukáš Pohl, PhD.

BRNO 2016



Bakalářská práce

bakalářský studijní obor **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

Student: Ján Schrötter

ID: 161914

Ročník: 3

Akademický rok: 2015/16

NÁZEV TÉMATU:

Řízení střídavých elektrických pohonů v prostředí Simulink Real-Time

POKYNY PRO VYPRACOVÁNÍ:

1. Zpracujte rešerši na téma vektorové řízení elektrických pohonů.
2. Realizujte vektorové řízení střídavých elektrických pohonů v prostředí Matlab Simulink.
3. Zpracujte rešerši na téma HIL (Hardware-in-the-loop) simulace, seznamte se s možnostmi použití Simulink Real-Time pro HIL.
4. Navrhněte experiment pro řízení vícefázového střídavého motoru pomocí Simulink Real-Time na cílové platformě s kartou MF624.
5. Vyhodnoťte možnosti prostředí Simulink Real-Time a karty MF624 pro řízení proudové, otáčkové případně polohové smyčky.

DOPORUČENÁ LITERATURA:

Neborak I.: Modelování a simulace elektrických regulovaných pohonů, Ostrava 2002.

Termín zadání: 8.2.2016

Termín odevzdání: 23.5.2016

Vedoucí práce: Ing. Lukáš Pohl, Ph.D.

Konzultant bakalářské práce:

doc. Ing. Václav Jirsík, CSc., předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské 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

Cieľom tejto bakalárskej práce je zrealizovať riadenie striedavého motoru s permanentými magnetmi pomocou Simulink Real-Time na platforme s kartou MF624. Riadenie bude realizované simulačnou metódou hardware-in-the-loop. V práci je popísaná štruktúra motoru a transformácie potrebné pre zrealizovanie vektorového riadenia. Ďalej je detailne popísaná cieľová platforma a riadiaci systém celej úlohy.

Klíčová slova

Simulink Real-Time, PMSM, HIL, vektorové riadenie, MF624

Abstract

The goal of this bachelor's thesis is implementation of control of permanent magnet synchronous motor using Simulink Real-Time on a platform with MF624 I/O card. The control will be realized by hardware-in-the-loop simulation method. The thesis deals with structure of the motor and transformations needed for vector control. The platform and control system of the whole task is described in the next part of the thesis.

Keywords

Simulink Real-Time, PMSM, HIL, vector control, MF624

Bibliografická citácia:

SCHRÖTTER, J. *Řízení střídavých elektrických pohonů v prostředí Simulink Real-Time*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2016. 44s. Vedúci bakalárskej práce bol Ing. Lukáš Pohl, PhD.

Prehlásenie

„Prehlasujem, že svoju bakalársku prácu na tému *Řízení střídavých elektrických pohonů v prostředí Simulink Real-Time* som vypracoval samostatne pod vedením vedúceho bakalárskej práce a s použitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej bakalárskej práce ďalej prehlasujem, že v súvislosti s vytvorením tejto bakalárskej práce som neporušil autorské práva tretích osôb, hlavne som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a som si plne vedomí následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona č. 121/2000 Sb., vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovení časti druhej, hlavy VI. diel 4 Trestného zákoníku č. 40/2009 Sb.

V Brne dňa: **20. mája 2016**

.....
podpis autora

Pod'akovanie

Ďakujem vedúcemu bakalárskej práce Ing. Lukášovi Pohlovi, PhD., za odborné rady, metodickú a pedagogickú pomoc a trpezlivosť pri vypracovávaní mojej bakalárskej práce.

V Brne dňa: **20. mája 2016**

.....
podpis autora

Obsah

1	Úvod.....	8
2	Synchronný motor s permanentnými magnetmi.....	9
2.1	Modelovanie PMSM.....	9
2.1.1	Transformácie súradníc.....	9
2.1.2	Matematický model.....	12
3	Simulink Real-Time.....	14
3.1	Reálny čas.....	14
3.2	Vývojový počítač.....	15
3.3	Cieľový počítač.....	15
3.4	Hardware-in-the-loop simulácia.....	16
3.5	Naša konfigurácia.....	17
4	Meracia karta MF624.....	18
4.1	Analógové vstupy.....	18
4.2	Vstupy pre kvadráturme enkóдеры.....	19
4.2.1	Enkóder.....	19
4.3	Čítače a časovače.....	20
4.3.1	Pulzne šírková modulácia.....	21
5	Pracovisko experimentu.....	22
5.1	Výkonová doska.....	23
5.1.1	Rozhranie dosky.....	24
5.1.2	Výkonová časť.....	25
5.2	Trojfázový FET pre-driver MC33937A.....	25
5.2.1	Operačné módy.....	25
5.2.2	SPI príkazy.....	26
5.2.3	Stavové registre.....	26
5.2.4	Inicializácia čipu.....	26
6	Vektorové riadenie.....	28
6.1	Simulácia.....	29
6.1.1	Návrh regulátorov.....	29
6.1.2	Anti wind-up.....	31
6.1.3	Simulácia vektorového riadenia PMSM.....	32
6.2	Experiment.....	34
6.2.1	Popis algoritmu.....	34

6.2.2	Návrh regulátorov	37
6.2.3	Výsledky HIL simulácie	39
7	Záver.....	42

1 ÚVOD

Táto bakalárska práca sa venuje problematike riadenia striedavých elektrických pohonov v reálnom čase, konkrétne synchronných motorov s permanentnými magnetmi (PMSM). V práci sa budeme zaoberať vývojom platformy pre realizáciu Hardware-in-the-loop (HIL) simulácie.

Význam HIL simulácie spočíva v zapojení riadaceho hardware a riadeného hardware, prípadne riadenej reálnej sústavy do simulačnej sľučky. Simulácia HIL sa často vykonáva na konci vývojového reťazca riadiacej jednotky, kedy je nutné overiť funkčnosť celého systému.

Na začiatku práce bude rozobraný princíp PMSM – jeho matematický model v dvojfázových súradniciach spojených so súradnicami rotoru a transformácie z trojfázových súradníc práve do spomínaných dvojfázových. Transformácie sa robia kvôli značnému zjednodušeniu riadenia.

Nasledovať bude popis platformy, na ktorej budeme realizovať HIL simuláciu. Naša platforma bude pozostávať z vývojového počítača, cieľového počítača vybaveného meracou I/O kartou MF624 a z výkonovej dosky pre riadenie trojfázových motorov. Práve karta MF624 bude popísaná ako prvá. Priblížime si jej základné vlastnosti a objasníme si, ako ju budeme používať, a na čo nám bude slúžiť. Pri popise nami využívaných vstupov a výstupov karty si vysvetlíme princíp pulzne šírkovej modulácie (PWM), snímača natočenia rotoru (inkrementálny enkóder) a ich aplikáciu v riadení motorov.

Ďalšia kapitola sa bude venovať pracovisku experimentu. Rozoberieme si z čoho pracovisko pozostáva a popíšeme výkonovú dosku od firmy Freescale a taktiež funkciu budiča MC33937A a jeho inicializáciu.

Pre vytváranie real time aplikácií budeme používať programovú nadstavbu Matlabu Simulink Real-Time. Jedná sa o relatívne nový produkt, ktorý nahrádza staršie prostredie xPC Target. V kapitole popisujúcej Simulink Real-Time si rozoberieme čo to znamená riadenie v reálnom čase a popíšeme si konfiguráciu vývojového a cieľového počítača.

Pre riadenie motoru budeme používať metódu zvanú vektorové riadenie, ktorá je popísaná v nasledujúcej časti práce. Táto kapitola bude rozdelená na 2 hlavné časti: simuláciu a experiment. V oboch častiach bude naznačený návrh regulátorov a odprezentované výsledky simulácie, respektíve HIL simulácie. V kapitole experimentu taktiež detailne vysvetlíme algoritmus našej HIL simulácie.

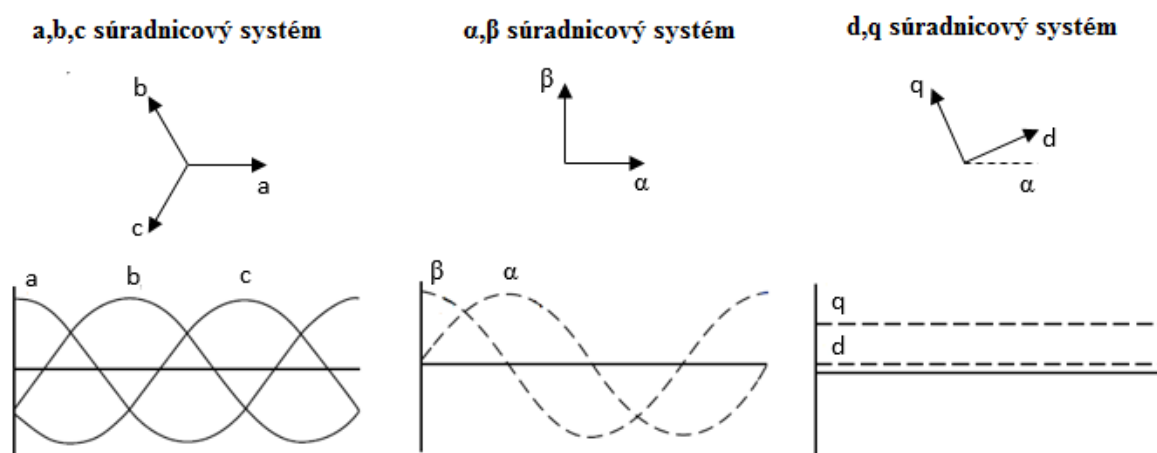
2 SYNCHRÓNNY MOTOR S PERMANENTNÝMI MAGNETMI

Synchrónny motor s premanentnými magnetmi (ďalej len PMSM) je točivý stroj, ktorého princíp je založený na vlastnostiach striedavého asynchrónneho motoru a bezkefového motoru. Na rotore PMSM sú umiestnené permanentné magnety, ktoré svojou magnetickou indukciou prispievajú k vytvoreniu magnetického poľa vo vzduchovej medzere. Stator tvorí trojfázové vinutie s fázami posunutými o 120° . Po pripojení trojfázového harmonického napätia na statorové vinutia, vzniká pri priechode prúdu statorovým vinutím točivé magnetické pole, ktoré rozotáča rotor s permanentnými magnetmi na rýchlosť, ktorá je rovnaká ako rýchlosť statorového poľa – otáčky rotoru a statorového poľa sú synchronné.

2.1 Modelovanie PMSM

2.1.1 Transformácie súradníc

Transformácie súradníc sa vykonávajú z dôvodu zjednodušenia trojfázového modelu motoru. Základné transformácie ktoré používame sa nazývajú Clarkovej transformácia (súradnicový systém $\alpha\beta$) a Parkova transformácia (súradnicový systém dq). Je výhodné riadiť motor v súradnicovom systéme dq spojenom s rotorom motoru, pretože sa v tomto systéme odstráni závislosť uhlu natočenia medzi statorom a rotorom. Nasledujúci obrázok ukazuje všetky tri súradnicové systémy.



Obrázok 1: Súradnicové systémy

Clarkovej transformácia

Trojfázové veličiny pretransformujeme z trojfázového súradného systému abc do dvojfázového stacionárneho súradného systému $\alpha\beta$. Transformácia je graficky zobrazaná na obrázku 2. Na základe znalosti uhlov medzi osami jednotlivých súradnicových systémov, vieme určiť uhly $\varphi_1, \varphi_2, \varphi_3, \varphi_4$. Clarkovej transformáciu popisujú nasledovné rovnice:

$$i_\alpha = \frac{2}{3}i_a - \frac{1}{3}i_b + \frac{1}{3}i_c \quad (1)$$

$$i_\beta = \frac{2}{\sqrt{3}}i_b - \frac{2}{\sqrt{3}}i_c \quad (2)$$

Kde i_a, i_b, i_c sú trojfázové veličiny, i_α, i_β sú dvojfázové stacionárne veličiny. Konštanty pred jednotlivými veličinami sme získali zo známych uhlov $\varphi_1-\varphi_4$ pomocou goniometrických funkcií sinus a kosinus. Ak trojfázový motor spĺňa nasledovnú podmienku symetrie:

$$i_a + i_b + i_c = 0 \quad (3)$$

Môžeme Clarkovej transformáciu zapísať v jednoduchšom tvare:

$$i_\alpha = i_a \quad (4)$$

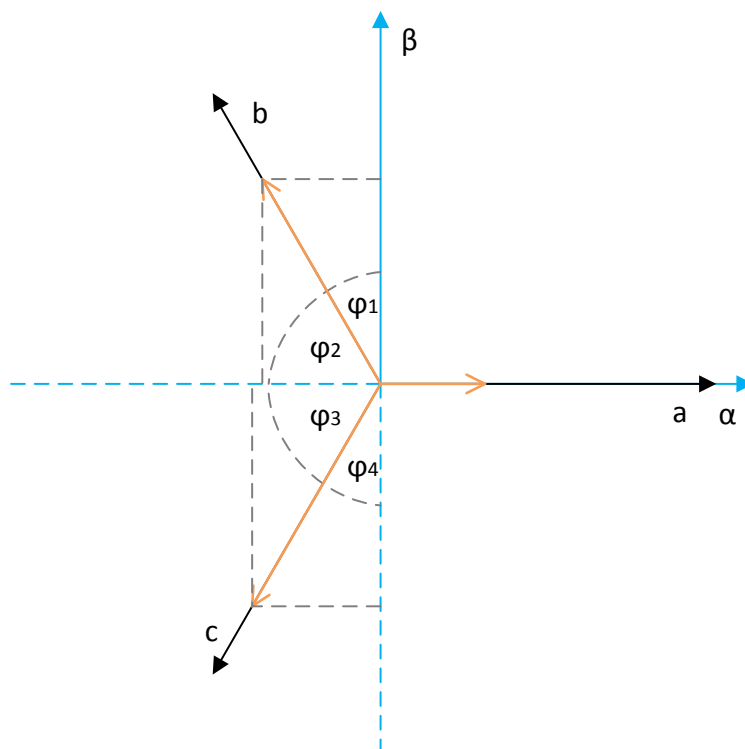
$$i_\beta = \frac{1}{\sqrt{3}}i_a + \frac{2}{\sqrt{3}}i_b \quad (5)$$

Táto transformácia platí pre prúdy, napätia a aj magnetické sprážené toky. Podobným spôsobom môžeme vyjadriť aj inverznú Clarkovej transformáciu:

$$i_a = i_\alpha \quad (6)$$

$$i_b = -\frac{1}{2}i_\alpha + \frac{\sqrt{3}}{2}i_\beta \quad (7)$$

$$i_c = -\frac{1}{2}i_\alpha - \frac{\sqrt{3}}{2}i_\beta \quad (8)$$



Obrázok 2: Clarkovej transformácia

Parkova transformácia

Dvojfázovú súradnú sústavu $\alpha\beta$ transformujeme do rotujúcej sústavy dq ktorá má zhodnú súradnicovú sústavu so súradnou sústavou rotoru. Vytvoríme tým pevnú magnetickú väzbu medzi rotorom a statorom. Transformáciou sa značne sprehládni a uľahčí riešenie riadenia, pretože regulátor pri riadení v dq sústave nemusí sledovať uhol natočenia medzi statorom a rotorom. Parkovu transformáciu popisujú nasledovné rovnice:

$$i_d = i_\alpha \cos(\varphi) + i_\beta \sin(\varphi) \quad (9)$$

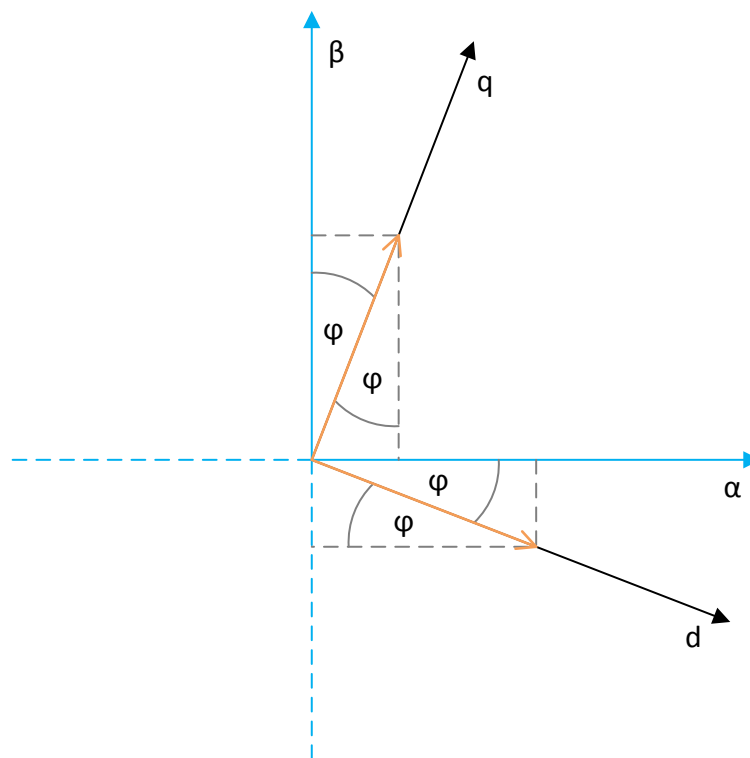
$$i_q = i_\beta \cos(\varphi) - i_\alpha \sin(\varphi) \quad (10)$$

Kde i_d, i_q sú rotujúce dvojfázové veličiny, i_α, i_β sú stacionárne dvojfázové veličiny, φ je uhol natočenia rotoru.

Inverznú Parkovu transformáciu popisujú rovnice:

$$i_\alpha = i_d \cos(\varphi) - i_q \sin(\varphi) \quad (11)$$

$$i_\beta = i_q \cos(\varphi) + i_d \sin(\varphi) \quad (12)$$



Obrázok 3: Parkova transformácia

2.1.2 Matematický model

Matematický model PMSM je tvorený sústavou diferenciálnych rovníc, ktoré za predpokladu zjednodušujúcich podmienok popisujú elektromagnetické a mechanické vlastnosti motoru. Zjednodušujúce predpoklady [1]:

- Priebeh magnetickej indukcie vo vzduchovej medzere a teda aj indukované napätie je sinusové, pričom je obecné uvažovaný rotor s vyniknutými pólmi, tj. S rôznou magneticou vodivosťou v pozdĺžnom a priečnom smere
- Parametre (R, L) stroja sú konštantné a rovnaké vo všetkých troch fázach
- Straty v železe sú zanedbané
- Tlmiace vinutie na rotore nie je urobené a rovnako sa zanedbávajú tlmiace účinky materiálu rotoru
- Nulový vodič nie je pripojený

Model budeme tvoriť v súradnicovej sústave dq. Voľba premenných:

- vstupné premenné: u_d, u_q, M_L
- stavové premenné: i_d, i_q, Ω_m
- výstupné premenné: i_d, i_q, Ω_m, M_e , event. ψ_d, ψ_q

Pre magnetické spráženie platí:

$$\psi_d = L_d i_d + \psi_f \quad (13)$$

$$\psi_q = L_q i_q \quad (14)$$

kde ψ_d je magnetický tok v priamom smere, ψ_q je magnetický tok v kolmom smere, ψ_f je magnetický tok budenia, i_d je prúd v priamom smere, i_q je prúd v kolmom smere, L_d je indukčnosť v priamom smere, L_q je indukčnosť v kolmom smere.

Napäťové rovnice budú vyzerat' nasledovne:

$$u_d = R_s i_d + \frac{d\psi_d}{dt} - \omega \psi_q = R_s i_d + \frac{d(L_d i_d + \psi_f)}{dt} - \omega L_q i_q = R_s i_d + L_d \frac{di_d}{dt} - \omega L_q i_q \quad (15)$$

$$u_q = R_s i_q + \frac{d\psi_q}{dt} + \omega \psi_d = R_s i_q + L_q \frac{di_q}{dt} + \omega (L_d i_d + \psi_f) \quad (16)$$

kde R_s je odpor statoru, ω je elektrická uhlová rýchlosť.

Z pohybovej rovnice získame deriváciou tretej stavovej premennej – mechanickej rýchlosti

$$p_p \Omega_m = \frac{1}{J_c} (M_e - M_L) \quad (17)$$

kde p_p je počet pólových párov, Ω_m je mechanická rýchlosť, J_c je celkový moment zotrvačnosti, Ω_m je mechanická rýchlosť, M_e je elektrický moment, M_L je záťažný moment.

$$\text{Elektrická rýchlosť } \omega = p_p \Omega_m \quad (18)$$

Elektromagnetický moment PMSM

$$M_e = \frac{3}{2} p_p (\psi_d i_q - \psi_q i_d) = \frac{3}{2} p_p [\psi_f + (L_d - L_q) i_d] i_q \quad (19)$$

Ak $L_d = L_q$, môžeme rovnicu zjednodušiť na:

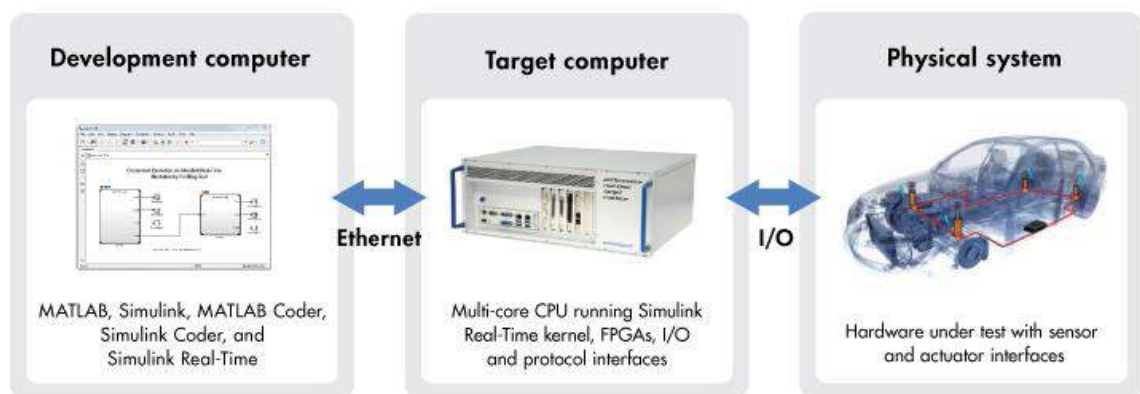
$$M_e = \frac{3}{2} p_p \psi_f i_q \quad (20)$$

Tento vzťah nám udáva, že moment stroja je daný súčinom momentovej zložky statorového prúdu a konštantného budiaceho magnetického toku, ktorý je daný len permanentnými magnetmi a nie výsledným tokom ovplyvneným statorovým prúdom. Zjednodušený vzťah pre elektromagnetický moment (20) platí aj v prípade, keď prúd $i_d = 0$.

3 SIMULINK REAL-TIME

Simulink Real-Time (ďalej len SLRT) slúži k návrhu a testovaniu modelov, aplikácií a algoritmov bežiacich v reálnom čase. Aplikácie vytvorené v prostredí Simulink je možné spúšťať na cieľových počítačoch, ktoré sú spojené s reálnym systémom. Produkt SLRT podporuje simulácie a testovanie v reálnom čase, vrátane prototypovania riadiacich systémov, DSP prototypovania, prototypovania vision systémov a hardware-in-the-loop simulácií [2]. Kapitoly 3.2 a 3.3 boli z časti prevzaté a preložené z [2].

Princíp SLRT je zobrazený na nasledujúcom obrázku:



Obrázok 4 [3]: Princíp SLRT

3.1 Reálny čas

Riadenie v reálnom čase (z anglického Real Time) znamená, že riadiaci obvod má presne stanovené časové okamžiky, v ktorých musí zbierať a spracovávať dáta a reagovať na zmeny v obvode. Systémy, ktoré takéto riadenie zabezpečujú, sa nazývajú operačné systémy reálneho času (z anglického Real Time Operation System - RTOS). Tieto operačné systémy si môžeme rozdeliť do dvoch kategórií. Prvá kategória, tzv. mäkký RTOS (z anglického soft RTOS), nám zabezpečuje reakciu na podnety v nejakom priemernom časovom intervale alebo do určitého priemerného času. V druhej kategórii, tvrdom RTOS (z anglického hard RTOS), musia byť požiadavky na chovanie systému bezpodmienečne splnené pri každom vyskytnutí sa udalosti v systéme. Riadenie v reálnom čase musí byť spoľahlivé a deterministické.

3.2 Vývojový počítač

Vývojový počítač je počítač, na ktorom vytvárame modely a aplikácie, ktoré budú bežať na cieľovom počítači. Väčšinou sa teda jedná o osobný desktop alebo notebook. Inštalácia produktu SLRT prebieha spolu s inštaláciou softvéru MATLAB a Simulink.

Periférie

Na nainštalovanie a spustenie produktu SLRT vyžaduje vývojový počítač 60 MB voľného priestoru na pevnom disku. Pre vytvorenie bootovacieho média sa požaduje jedno z:

- CD-RW alebo DVD-RW mechanika
- USB mechanika
- SD flash mechanika
- Externý pevný disk
- 3,5-palcová disketová mechanika

Komunikácia

Pre komunikáciu medzi vývojovým a cieľovým počítačom vyžaduje vývojový počítač jednu z nasledujúcich metód:

- Jedno voľné Ethernetové rozhranie (PCI, ISA, alebo USB)
- Jeden voľný sériový port (COM1 alebo COM2)

Komunikácia nám zabezpečuje stiahnutie a nahranie modelu a aplikácie z vývojového počítača do cieľového počítača.

3.3 Cieľový počítač

Pri výbere cieľového počítača máme 2 možnosti. Prvá je cieľový počítač od firmy Speedgoat, ktorá sa špecializuje na vývoj a výrobu priemyselných počítačov určených ako cieľové platformy priamo pre SLRT. Druhá možnosť je zvoliť akýkoľvek iný cieľový počítač. Túto možnosť sme zvolili aj my. Rozoberieme si podrobnejšie nastavenie a konfiguráciu cieľového počítača.

Na nakonfigurovanie cieľového počítača sme použili prostredie Simulink Real-Time Explorer, do ktorého sa dostaneme zadaním príkazu *slrtexplr* v MATLABe. V bloku Targets si otvoríme nastavenia cieľového počítača (štandardne označený ako TargetPC1) a následne klikneme na Target settings, kde robíme nasledujúce nastavenia pre cieľový počítač:

- Graphics mode – zobrazenie grafických informácií na cieľovom počítači (napríklad Scope na cieľovom počítači)
- USB Support – používanie USB portu na cieľovom počítači (napríklad použitie USB myši)
- Secondary IDE – používanie disku pripojeného k sekundárnemu IDE rozhraniu

- Multicore CPU – využitie viacerých jadier procesora ak nimi cieľový počítač disponuje
- Target is a 386/486 – cieľový počítač má procesor z rodiny 386 alebo 486
- RAM size – celková veľkosť nainštalovanej pamäte RAM v cieľovom počítači
- Model size – rezervovanie daného množstva pamäte pre real-time aplikáciu

Pre naboťovanie SLRT do cieľového počítača máme na výber niekoľko metód: Bootovanie cez sieť, z CD/DVD, z vymeniteľného disku, DOS Loader metóda, Standalone metóda. Vytvorenie bootovacieho média prebieha opäť v Simulink Real-Time Exploreri. Vyberieme si požadovanú metódu v Boot configuration v nastavení cieľového počítača. Predtým je ešte potrebné nastaviť spôsob komunikácie medzi vývojovým a cieľovým počítačom v Host-to-Target communication. Na výber máme medzi TCP/IP komunikáciou alebo sériovou RS-232 komunikáciou. V prípade výberu TCP/IP komunikácie je nutné nastaviť sieťové parametre cieľového počítača a parametre Ethernetovej linky.

Po vykonaní všetkých vyššie uvedených nastavení a vytvorení bootovacieho média musíme v BIOSe cieľového počítača nastaviť bootovanie z daného typu média a presvedčíme sa o prepojení cieľového a vývojového počítača zvoleným typom komunikačnej linky.

3.4 Hardware-in-the-loop simulácia

V tejto bakalárskej práci sa venujeme simulácii s hardware v sľučke, známej z anglickej literatúry ako Hardware In the Loop (HIL). Simulácia spôsobom HIL sa vyznačuje zapojením hardware do regulačnej sľučky vyvíjaného a zatiaľ nie úplne kompletného systému. Týmto typom simulácie je možné priblížiť sa reálnemu svetu a väčšinou sa využíva ako posledný článok pred spustením reálneho technologického zariadenia. Tejto simulácii často predchádzajú simulácie Model In the Loop (MIL) a Processor In the Loop (PIL) [4].

Model in the loop

Pri simulácii MIL je nutné identifikovať riadenú sústavu (vytvoriť jej model) a navrhnuť k nej riadiaci systém. Po zhotovení modelov riadenej sústavy a riadiaceho systému, je takýto celkový model vytvorený vo vhodnom simulačnom nástroji a je možné simulovať jeho chovanie. Miera idealizácie pri simulácii MIL je však veľmi vysoká, preto návrh ani výsledky simulácie nemožno považovať za presné.

Processor in the loop

Simulácia typu PIL sa oproti MIL oveľa bližšie približuje realite. Ako simulátor riadiaceho systému sa použije riadiaci hardware s procesorom, ktorý bude v budúcnosti riadiť reálny technologický proces. Riadená sústava nie je fyzický systém, ale jeho model, ktorý je nasimulovaný napríklad v osobnom počítači alebo v inom hardware. Riadiaci

hardware a riadený hardware sú spojené komunikačným rozhraním, napríklad RS232 alebo Ethernet.

Celá simulácia je riadená z počítača (host), ktorý obsahuje model celého systému. Počítač naviaže spojenie s riadiacim systémom a takto dokáže simuláciu spustiť, zastaviť a riadiaci systém za behu meniť.

Hardware in the loop

Princíp zo simulácie PIL je zachovaný, ale je použité reálne komunikačné rozhranie a taktiež je inakšie modelovaná riadená sústava. Riadiaci systém sa môže odlaďovať s reálnou riadenou sústavou, alebo sa miesto nej používa hardware simulujúci reálne fyzikálne veličiny. Tieto veličiny sú potom cez vhodné rozhranie pripojené k riadiacemu systému (A/D, D/A prevodníky a podobne). Preto sme schopní sa pomocou HIL simulácie najviac priblížiť reálnemu svetu aj pri použití riadeného hardware (model sústavy), alebo v konečnom dôsledku pri použití reálnej sústavy.

3.5 Naša konfigurácia

V našom prípade disponujeme dvomi osobnými počítačmi. Jeden počítač slúži ako vývojový, vybavený programom MATLAB a jeho nástrojom Simulink. Z vývojového počítača sa riadi celá simulácia. Cieľový počítač vybavený meracou kartou MF624 slúži spolu s výkonovou doskou Freescale ako riadiaci hardware. Riadiaci hardware je prepojený s reálnou sústavou – PMSM.

Medzi cieľovým a vývojovým počítačom sme zvolili komunikáciu typu TCP/IP cez Ethernetovú linku. Bootovanie na cieľovom počítači prebieha z bootovacieho CD. Pokúšali sme sa aj o sieťové bootovanie, ale narazili sme na problém kedy by sme museli využívať DHCP server pre pridelovanie IP adresy a masky siete cieľovému počítaču. Taktiež sme skúsili bootovanie z USB, ale to cieľový počítač nepodporoval. Na cieľovom počítači využívame grafický mód, pripojený monitor a klávesnicu.

4 MERACIA KARTA MF624

Meracia karta (iné označenia: karta pre zber dát, I/O karta) MF624, nám bude slúžiť na meranie alebo generovanie signálu medzi cieľovým počítačom a reálnym modelom.

Karta využíva PCI zbernicu stolného PC. Obsahuje analógové vstupy a výstupy, vstupy pre kvadrátne enkodéry, čítače/časovače a digitálne vstupy a výstupy. Karta taktiež obsahuje ovládač pre SLRT. Podrobný popis vlastností je voľne prístupný na internete [5]. V našej práci budeme využívať analógové vstupy, vstup pre kvadrátne enkóder a čítač pre generovanie PWM. Všetky periférie budeme ovládať pomocou Simulinkových bločkov od firmy Humusoft určených pre túto kartu.



Obrázok 5 [6]: Meracia karta MF624

4.1 Analógové vstupy

Karta obsahuje 8 single-ended analógových vstupov so 14-bitovými prevodníkmi, doba prevodu závisí na počte použitých kanálov. Vstupný rozsah napätia je ± 10 V.

Analógové vstupy nám budú slúžiť pre meranie fázových prúdov motora cez LEM prevodníky. Budeme ich ovládať pomocou Simulinkového bločku od firmy Humusoft, ktorý nájdeme v *Simulink Library Browser* → *Simulink Real-Time* → *A/D* → *Humusoft* → *MF624*. V bločku máme možnosť nastavenia čísla používaného kanálu (1-8, na karte sú však kanály číslované od nuly), vzorkovaciu periódu (čas kedy bloček produkuje výstup alebo obnovuje svoju hodnotu), a PCI slot ktorý ponechávame v hodnote -1 lebo máme len jednu kartu a Simulink si port nájde sám. Hardvérový vstup pre bloček sú, ako už bolo spomenuté, volty, výstup je číslo vo formáte double. Škálovanie je 1:1.

Pri pokusoch o sprevádzkovanie analógového vstupu dochádzalo ku zamrznutiu cieľového počítača a nebolo s ním možné ďalej komunikovať. To sme vyriešili kontaktovaním zákazníckej podpory firmy Humusoft. V odpovedi nám zaslali

revizovanú S-funkciu pre ovládanie A/D prevodníka, ktorou sme nahradili pôvodnú funkciu a testovacia aplikácia už bežala správne.

4.2 Vstupy pre kvadráturne enkóдеры

Karta obsahuje 4 vstupné kanály s 32-bitovým rozlíšením pre single-ended alebo diferenciálne enkóдеры. Vstupy na karte sú signály z enkóderu A, B a index.

Vstup pre inkrementálny snímač nám bude slúžiť pre určovanie pozície rotoru motora. Simulinkový bloček nájdeme v *Simulink Library Browser* → *Simulink Real-Time* → *Incremental Encoder* → *Humusoft* → *MF624*. V bločku robíme rovnaké nastavenia ako v bločku pre analógové vstupy, okrem toho ešte nastavujeme filtrovanie meraného signálu a funkciu pre vstupný signál (resetovanie/spustenie čítača na zostupnú/vzostupnú hranu, ...). Vektor pre filtrovanie a funkciu musí byť rovnako dlhý ako vektor použitých kanálov (každému kanálu určujeme filtrovanie a funkciu). Hardvérový výstup z enkóderu je TTL logika, do bločku potom vstupuje číslo double a škálovanie je: $< 0,5 = \text{TTL}_{\text{low}}$; $\geq 0,5 = \text{TTL}_{\text{high}}$.

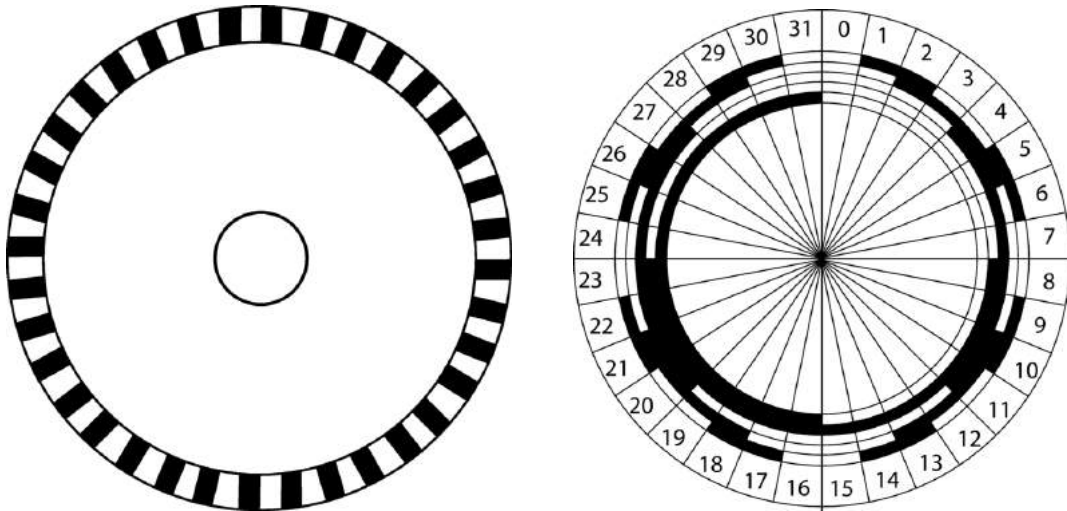
4.2.1 Enkóder

Enkóder je rotačný snímač polohy, v našej práci ho budeme používať na určovanie polohy rotoru. Najčastejšie používané sú optické enkóдеры, ktorých hlavnou časťou je kotúč, po obvode ktorého sa striedajú pre svetlo priepustné a nepriepustné plochy. Základné delenie enkóderov je na absolútne a inkrementálne.

Inkrementálne enkóдеры

Inkrementálne enkóдеры generujú pri rotácii za sebou idúce obdĺžnikové pulzy, kedy každý obdĺžnik odpovedá otočeniu vždy o rovnaký počet stupňov. Inkrementálne enkóдеры nám dávajú relatívnu informáciu o polohe.

Poznáme ich niekoľko druhov: jednokanálový, dvojkanálový a dvojkanálový s nulovým indexom. Pri jednokanálovom sme schopný určiť len informáciu o natočení, pri dvojkanálovom sme schopný určiť aj smer otáčania, pretože druhý kanál má rovnaký počet priepustných plôch posunutých o 90° . Podľa toho ktorý kanál predbieha ten druhý, vieme určiť smer otáčania. Kanál označený ako index generuje jeden pulz za minútu a slúži ako referencia začiatkovej pozície.



Obrázok 6 [7]: Jednokanálový inkrementálny enkóder a absolútny enkóder

Absolútne enkóдеры

Narozdiel od inkrementálnych enkóderov, ktoré nám dávajú zmenu polohy danú počtom pulzov, nám absolútne enkóдеры hovoria presnú polohu disku v rámci 360° . Priepustné a nepriepustné plochy po obvode sú rozmiestnené najčastejšie podľa binárneho alebo greyovho kódovania. Pri absolútnom snímači tak vieme zistiť polohu disku aj po odpojení od napájania.

Výpočet rýchlosti otáčania

Pre zmeranie rýchlosti rotoru potrebujeme poznať zmenu pozície za zmenu času. Poznáme 2 metódy ako rýchlosť zmerať: počet pulzov za časový interval, alebo časový interval medzi nástupnou a zostupnou hranou pulzu (dĺžka pulzu). V našej aplikácii používame prvú metódu. Je potrebné zohľadniť rozlíšenie enkódera (počet pulzov na jednu otáčku), aby sme vedeli prepočítať daný počet pulzov na uhol, z čoho už následne nie je problém určiť uhlovú rýchlosť v akýchkoľvek jednotkách (radiány za sekundu, otáčky za minútu,...)

4.3 Čítače a časovače

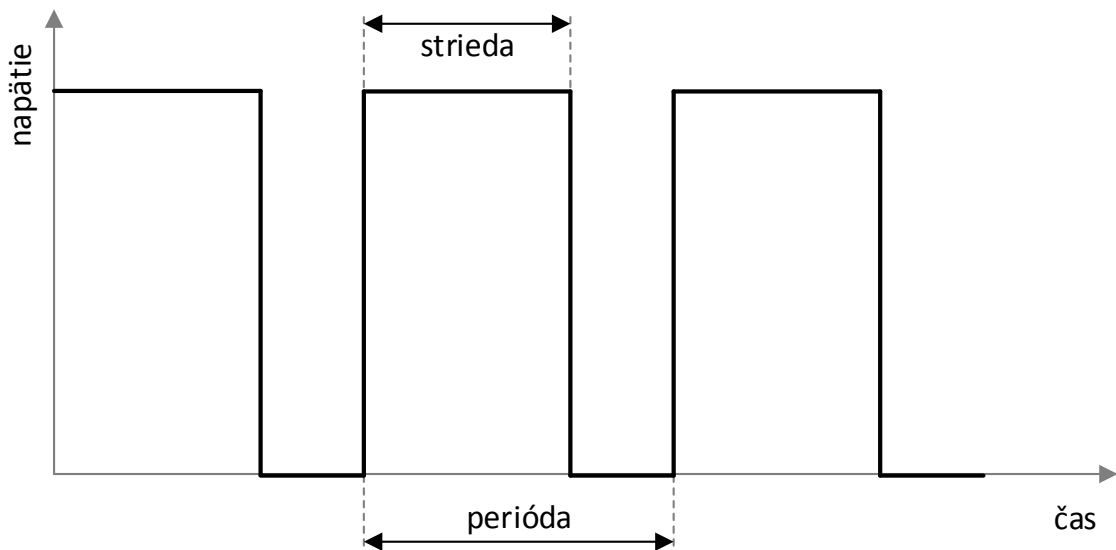
Karta obsahuje 4 kanály pre čítače a časovače s rozlíšením 32 bitov. Čítače nám poskytujú široké množstvo operačných módov, a to napríklad: pričítanie/odčítanie, generovanie pulzov vrátane PWM, softvérové/externé trigrovanie,...

V našej práci budeme využívať práve pulzne šírkovú moduláciu (z anglického Pulse-Width Modulation - PWM), pre riadenie veľkosti generovaného napätia privádzaného na fázy motoru. Simulinkový bloček nájdeme v *Simulink Library Browser* → *Simulink Real-Time* → *Counter* → *Humusoft* → *MF624PWM*. V bločku je opäť nutné nastaviť číslo použitého kanálu, vzorkovaciu periódu a PCI slot. Hardvérový výstup je logika TTL, vstup do bločku sú čísla double, škálovanie je rovnaké ako pri bločku

enkóderu. Na prvý vstup do bločku sa privádza informácia o striede v rozsahu 0-1, na druhý vstup informácia o frekvencii v Hz.

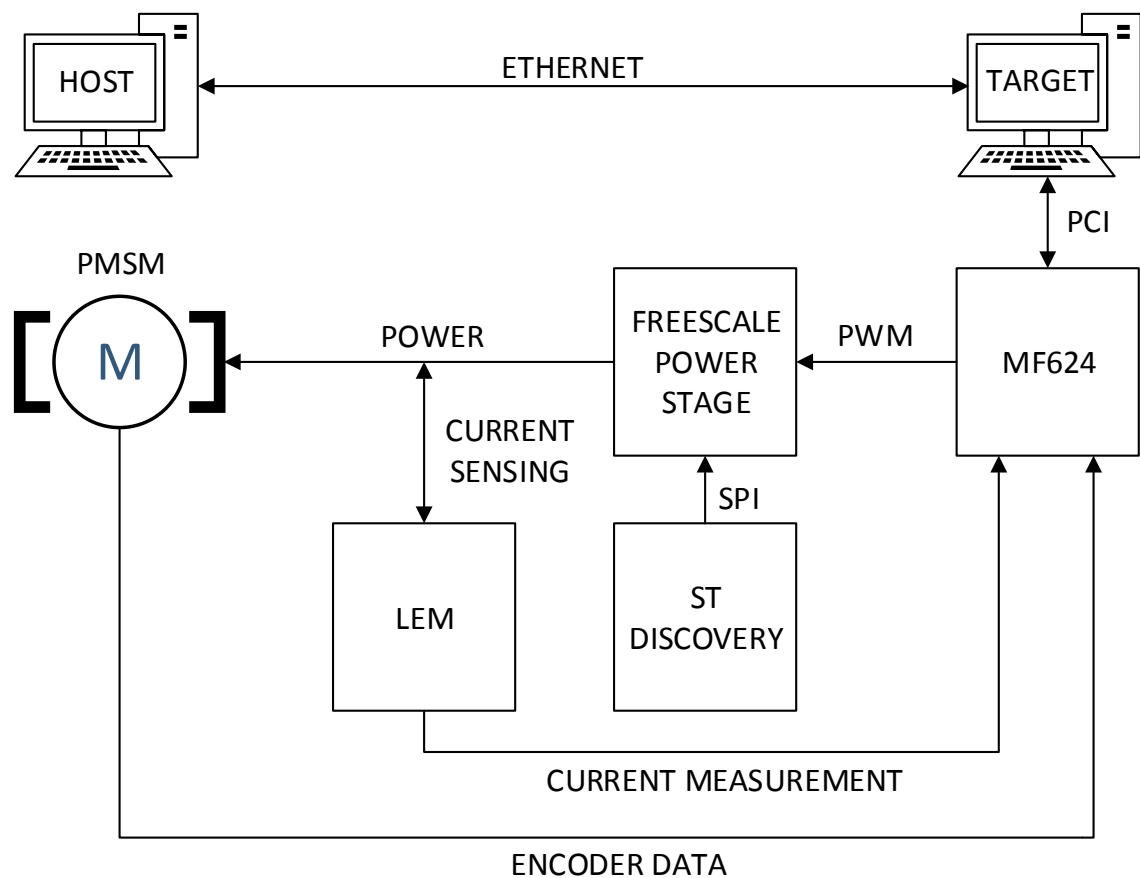
4.3.1 Pulzne šírková modulácia

Princípom PWM je, že prenášaný signál je zakódovaný medzi stavmi 1/0 (zapnuté/vypnuté) a jeho hodnota je určená ako pomer medzi týmito dobami – pomer sa nazýva strieda. Čas, za ktorý prebehne jedna strieda, sa nazýva perióda. Pojmy sú vysvetlené na obrázku číslo 7. Parameter strieda sa teda môže meniť v intervale 0-1 respektíve 0-100%. Kvôli relatívnemu vyjadreniu striedy musíme vždy poznať aj skutočnú hodnotu signálu ktorý modulujeme (v našom prípade karta MF624 používa logiku TTL, a teda pracujeme s napätím 0-5V). Poznáme niekoľko druhov PWM: centrovaná, zarovnaná doľava/doprava. Pri zarovnaní doľava/doprava zostáva vzostupná/zostupná hrana vždy na jednom mieste, podľa striedy sa posúva druhá hrana. Pri centrovanej PWM sa posúvajú obidve hrany. Karta MF624 generuje PWM zarovnanú doľava. PWM pri motoroch riadi dobu zopnutia napájacieho napätia.



Obrázok 7: PWM

5 PRACOVISKO EXPERIMENTU

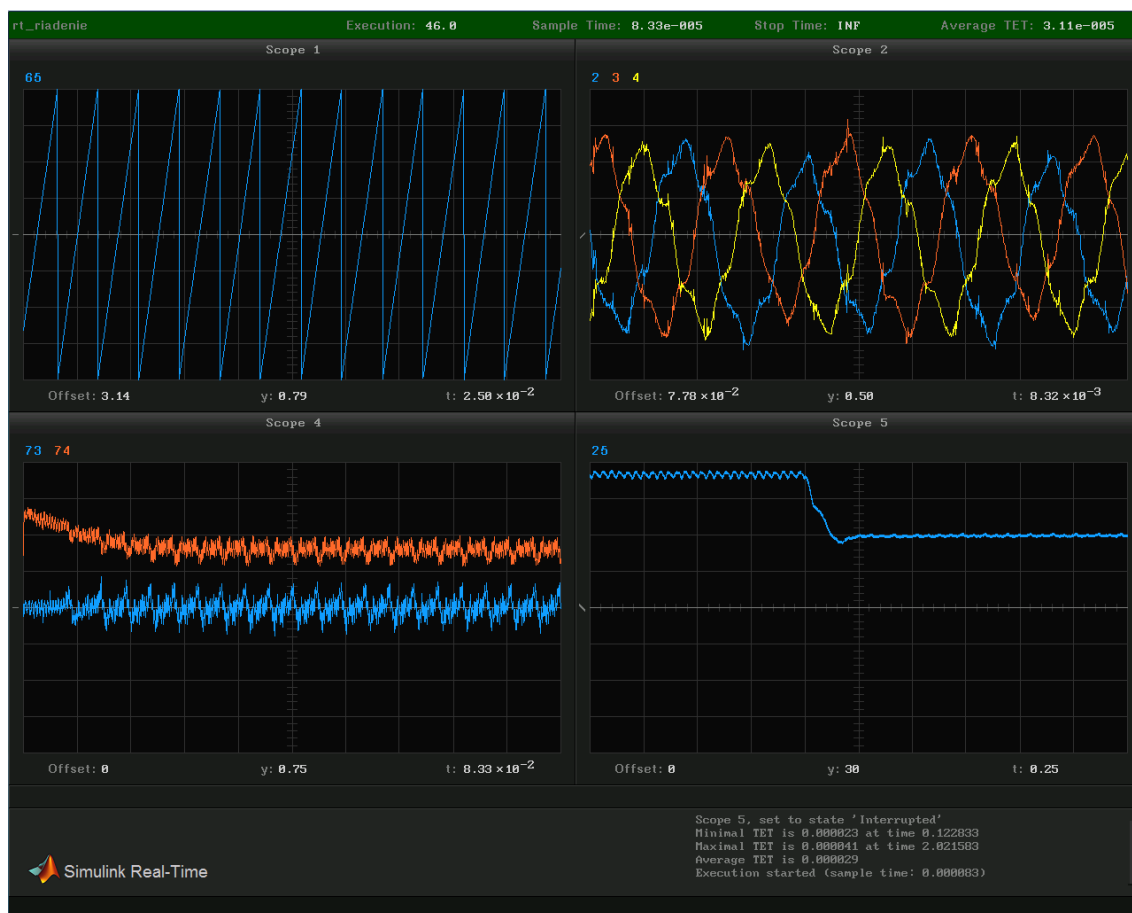


Obrázok 8: Bloková schéma experimentu

Pracovisku experimentu pozostáva z:

- Vývojového počítača, ktorý obsahuje model celého systému v programe Simulink, a z ktorého je celá simulácia riadená
- Cieľového počítača, na ktorom beží SLRT a spolu s kartou MF624 a doskou Freescale predstavuje riadiaci hardware simulácie. K počítaču sme mali pripojený monitor. V prípade, že sa v Simulink modeli úlohy použili bločky Target Scope, na monitore bolo možné sledovať priebehy meraných veličín v reálnom čase. Screenshot z monitoru je na obrázku číslo 9
- Karty MF624, ktorá slúži na meranie alebo generovanie signálu medzi riadenou sústavou a riadiacim systémom
- Výkonovej dosky Freescale, ktorá napája motor na základe PWM signálov z MF624
- Dosky ST Discovery, ktorá inicializuje pre-driver MC33937A na doske Freescale
- LEM prevodníkov, ktoré slúžia na meranie prúdov na fázach motora

- Synchronného motoru s permanentnými magnetmi TG Drives, ktorý predstavuje riadenú sústavu. Motor je vybavený inkrementálnym enkóderom



Obrázok 9: Príklad grafického rozhrania SLRT

5.1 Výkonová doska

V práci používame trojfázovú nízkonapäťovú výkonovú dosku od firmy Freescale pre navrhovanie a testovanie riadiacich algoritmov BLDC a PMSM motorov. Doska podporuje rýchlostné a pozičné snímanie pomocou Hallových senzorov, resolverov, enkóderov a BackEMF signálov pre bezsnímačové riadenie. Doska je konštruovaná pre vstupné napätie 8-50 V. Obvod pre meranie prúdu je nastavený pre ± 10 A. MC33937A MOSFET pre-driver sa stará o riadenie tranzistorových mostíkov. Doska obsahuje 2 externé rozhrania: UNI-3 a MC33937A. Informácie obsiahnuté v tejto kapitole sú z veľkej časti prevzaté z datasheetu tejto dosky [8].

5.1.1 Rozhranie dosky

Vstupy a výstupy sú umiestnené na ôsmich konektoroch prístupných na doske.



Obrázok 10: 3-PHASE LOW VOLTAGE POWER STAGE Freescale výkonová doska

Napájanie J1 a J2

Vstupný konektor J1 a vstupný terminál J2 slúžia na napájanie. Doska je konštruovaná na napájacie napätie 8-50V.

UNI3 rozhranie J3

UNI3 konektor definuje rozhranie medzi výkonovou doskou a priradenou riadiacou jednotkou (v našom prípade cieľový počítač s kartou MF624). Medzi UNI3 signály, ktoré využívame, patria napríklad PWM signály pre riadenie fázových napätí.

MC33937A rozhranie J4

Pre-driver MC33937A má na starosti riadenie tranzistorového mosta. Čip je nastavovaný cez SPI zbernicu. Konektor J4 obsahuje vstupy a výstupy pre základné SPI signály (SCK, SIN, SOUT, CS) a signály EN (device enable), OC (overcurrent comparator), RST (reset), INT (interrupt).

Konektory motoru J6, J7, J8

Výkonové výstupy pre motor sú umiestnené na konektoroch J6, J7, J8. Výstupy sú označené A, B, C, podľa fáz motora.

Konektor pre prerušovač J5

Prerušovací rezistor môže byť pridaný do konektoru J5.

5.1.2 Výkonová časť

Výkonová časť dosky je koncipovaná ako trojfázový most s MOSFET tranzistormi. Most pozostáva zo šiestich MOSFET tranzistorov (dva pre každú fázu), ktoré slúžia ako PWM striedače pre riadenie motoru. Čip MC33937A zabezpečuje napájacie napätie pre hradlo (gate) vrchných (high-side, HS) respektíve spodných (low-side, LS) častí tranzistorového mostu. Otváranie a zatváranie horných a spodných tranzistorov musí byť ošetrené takzvanou mŕtvou dobou (dead time). Ak by bol otvorený naraz horný aj spodný tranzistor na jednej fáze, došlo by ku skratu. Zabezpečenie mŕtvej doby má na starosti taktiež čip MC33937A. Viac o jeho funkcii a nastavovaní v nasledujúcej kapitole.

5.2 Trojfázový FET pre-driver MC33937A

Budič MC33937A je navrhnutý pre riadenie trojfázových motorov a podobné aplikácie. Integrovaný obvod obsahuje tri HS FET pre-drivere a tri LS FET pre-drivere. Inicializácia prebieha cez SPI rozhranie. V našom prípade nám na inicializáciu slúži doska Discovery od firmy ST. Informácie v tejto kapitole sú prevzaté z datasheetu tohto zariadenia [9].

5.2.1 Operačné módy

MC33937A má 5 operačných módov, definovaných pomocou troch riadiacich vstupov ($\overline{\text{RST}}$, EN1, EN2). Módy sú popísané nasledovne:

- **Sleep mode** - $\overline{\text{RST}}$ je low (vypnuté), obvod je móde spánku
- **Standby mode** - $\overline{\text{RST}}$ je high (zapnuté), EN1 alebo EN2 je low. Všetky externé FET tranzistory sú vypnuté
- **Initialization mode** – EN1, EN2, $\overline{\text{RST}}$ sú high. Zopnutím LS a potom HS tranzistorov sa inicializuje driver a začína normálny beh v Enable móde. Viac o inicializácii v kapitole 5.2.4
- **Enable mode** – po inicializácii zariadenia pracuje v normálnom režime, pokým sú obidva enable piny a $\overline{\text{RST}}$ pin high
- **Fault protection mode** – drivre sú vypnuté ak nastane chyba. Po odstránení chýb je oživenie možné až po opätovnej inicializácii

5.2.2 SPI príkazy

Integrovaný obvod obsahuje interné registre, ktoré riadia a kontrolujú operačné parametre, módy a prerušenia. K registrom sa pristupuje cez 8 bitové SPI príkazy. SPI logika generuje chybu a ignoruje príkazy ak počet prijatých bitov nie je 8 alebo násobok 8. Po $\overline{\text{RST}}$ je prvý SPI prijatý výsledok Status Register 0. Zoznam príkazov:

- NULL – tieto príkazy čítajú stav zariadenia, nemenia ho. Vracia Status Register 0-3
- MASK0, MASK1 – maska pre prerušenia. Nastavený bit aktivuje korešpondujúce prerušenie. Vracia Status Register 0
- Mode – aktivuje desaturačný chybový mód, FULLON mód, vypína ďalšie zmeny módu. Vracia Status Register 0
- CLINT0, CLINT1 – mažú prerušenie nastavené príkazom MASK0, resp. MASK1
- DEADTIME – Deadtime (mŕtva doba) je čas, ktorý zabraňuje zopnutiu oboch tranzistorov (HS aj LS) jednej fázy naraz. Po vypnutí jedného tranzistora z dvojice dôjde k zopnutiu druhého až po uplynutí tohto času. Deadtime je nastavený poslaním príkazu DEADTIME a následným kalibračným pulzom $\overline{\text{CS}}$, ktorý je 16 krát dlhší ako požadovaný deadtime

5.2.3 Stavové registre

Po každom SPI príkaze je stav zariadenia oznámený správou z SPI portu. V MC33973A sa nachádzajú štyri stavové registre. Register 0 je väčšinou používaný pre všeobecný stav. Registre 1-3 sú používané pre čítanie alebo potvrdzovanie interných nastavení zariadenia.

5.2.4 Inicializácia čipu

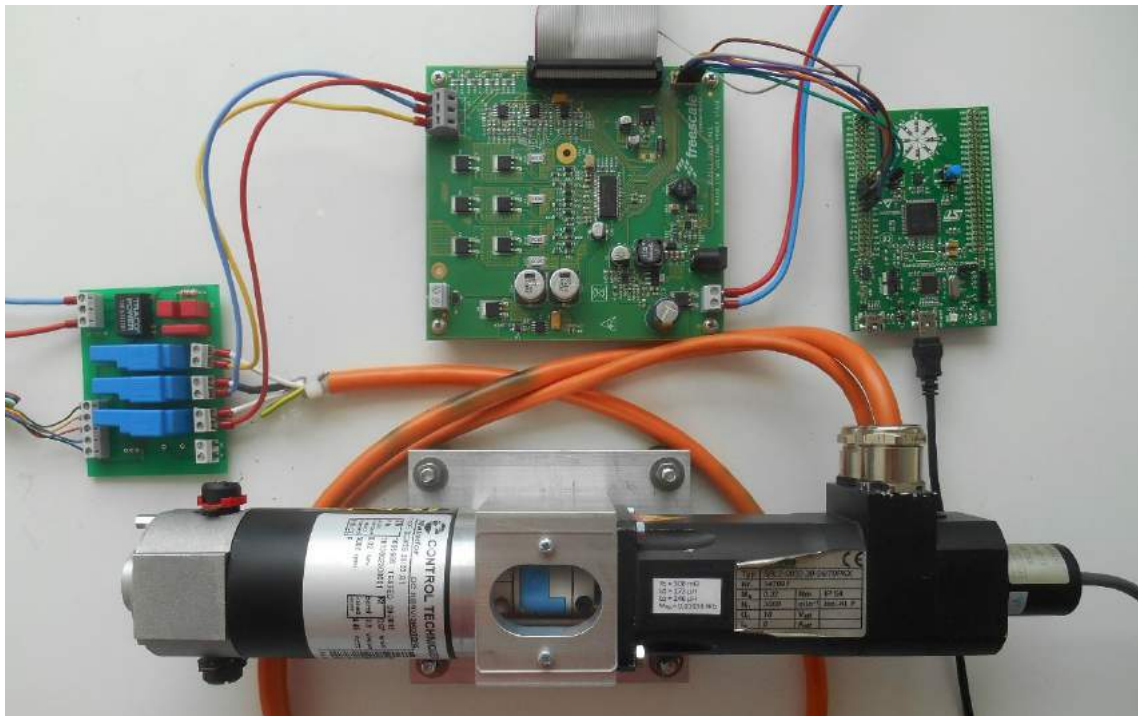
MC33973A poskytuje bezpečné a spoľahlivé riadenie pre riadiace jednotky trojfázových motorov v prípade, že bol čip správne inicializovaný.

Postup inicializácie je nasledovný:

- 1) Napojenie zariadenia
- 2) $\overline{\text{RST}}$ nastavené na high, EN1 a EN2 na low
- 3) Inicializácia registrov
 - a) Odstránenie všetkých signalizácií prerušenia (príkazy CINT0 a CINT1)
 - b) Inicializácia MASK registra na „vymaskovanie“ nežiadúcich prerušení
 - c) Nastavenie mŕtvej doby
 - d) Nastavenie módu príkazom MODE, zablokovanie zmeny módu
- 4) Nastavenie EN1 a EN2 na high
- 5) Inicializácia výstupov

- a) Vypnutie všetkých HS
 - b) Zapnutie všetkých LS. Prepnutie z low na high musí prebehnúť po tom, ako sa EN1 a EN2 nastavili na high
 - c) Čakanie na nabitie bootstrap kondenzátorov
 - d) Vypnutie všetkých LS
 - e) Zapnutie všetkých HS
 - f) Vypnutie všetkých HS
- 6) Zariadenie je pripravené a čaká na ďalšie operácie

Nasledujúci obrázok zobrazuje zapojenie nášho experimentu. Doska Freescale a LEM prevodníky boli napájané z laboratórneho zdroja Diametral, inicializačná doska ST bola napájaná cez USB z počítača.



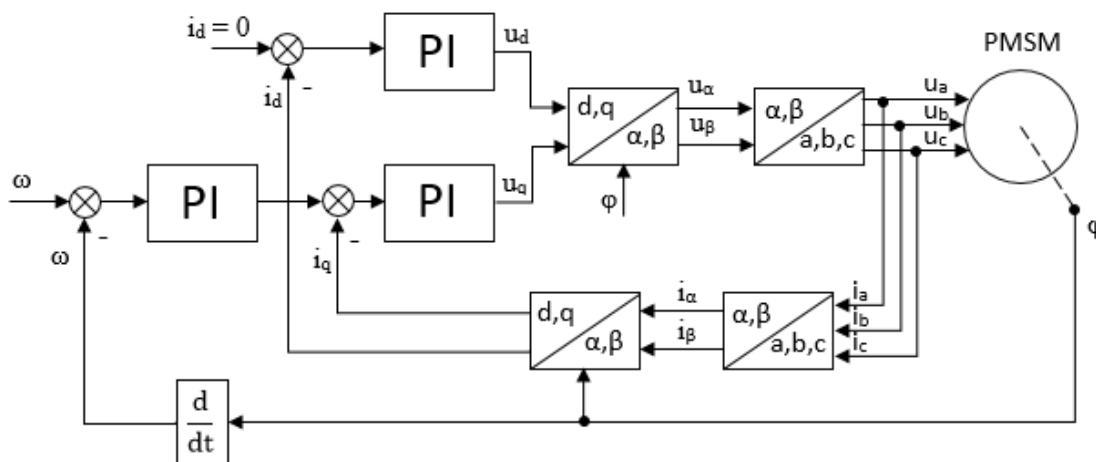
Obrázok 11: Zapojenie experimentu

6 VEKTOROVÉ RIADENIE

Jednou z najčastejšie používaných metód pre riadenie PMSM je vektorové riadenie. Hlavnou myšlienkou vektorového riadenia PMSM je riadenie magnetického poľa statoru a rotoru tak, aby boli navzájom kolmé. Kvôli zjednodušeniu riadenia motoru je vhodné prejsť z riadenia trojfázového do riadenia dvojfázového v súradnom systéme dq. Tieto prevody sú realizované za pomoci Parkovej a Clarkovej transformácie, ktoré sú popísané v kapitole 2.1.1. Osa d statorového prúdu sa nazýva tokotvorná (ovplyvňuje magnetický sprážený tok motoru) a osa q sa nazýva momentotvorná (ovplyvňuje moment motoru). Vektorové riadenie požaduje presnú informáciu o pozícii rotoru – získame ju napríklad z enkóderu. Základný princíp vektorového riadenia:

1. Zmerajú sa hodnoty prúdov na fázach motoru
2. Prepočet $abc \rightarrow \alpha\beta$
3. Prepočet $\alpha\beta \rightarrow dq$
4. Regulácia zložiek dq
5. Prepočet $dq \rightarrow \alpha\beta$
6. Prepočet $\alpha\beta \rightarrow abc$

Obrázok 12 potom zobrazuje blokovú schému vektorového riadenia. Na motore sú merané fázové prúdy i_a, i_b, i_c , ktoré sú za pomoci informácie z enkóderu o natočení rotoru transformované na i_d, i_q . Rozdiel medzi i_d, i_q a referenčnými hodnotami kompenzujú PI regulátory, ktoré odchýlku transformujú na napätia u_d, u_q . Napätia u_d, u_q sa opäť transformujú na napätia u_a, u_b, u_c . V PMSM permanentné magnety na rotore vytvárajú konštantný magnetický tok, preto motor netreba budiť a zložku i_d tak regulujeme na nulu. Požadovanú hodnotu prúdu i_q udáva výstup PI regulátora otáčok.



Obrázok 12: Bloková schéma vektorového riadenia

6.1 Simulácia

6.1.1 Návrh regulátorov

V regulačnej schéme pre vektorové riadenie PMSM sa nachádzajú tri regulátory. Dva z nich sú regulátory prúdov i_d a i_q , tretí je regulátor otáčok. Jedná sa o rozvetvený regulačný obvod, takzvanú kaskádnú reguláciu, kde je regulátor prúdu i_q podradený regulátoru otáčok. Slučka prúdu i_d už do tohto rozvetveného obvodu nepatrí a je regulovaná na nulu, viď úvod kapitoly 6. Vnútoraná, prúdová slučka, sa realizuje ako rádovo rýchlejšia oproti vonkajšej, otáčkovej slučke.

Sústavu pre návrh vnútornej slučky prúdu i_q v Laplaceovej transformácii odvodíme z (16):

$$F_{Si_q}(p) = \frac{1}{R_S + L_q p} = \frac{K_{Si_q}}{1 + T p} \quad (21)$$

Kde $K_{Si_q} = \frac{1}{R_S}$, $T = \frac{L_q}{R_S}$, p je Laplaceov operátor. Budeme realizovať PI regulátor, vďaka ktorému budeme schopný vykompenzovať nulu prenosu, výsledný prenos riadenia potom bude jednoduchý zotrvačný článok. Prenos PI regulátoru:

$$F_{Ri_q}(p) = \frac{K_{Ri_q}(T p + 1)}{p} \quad (22)$$

Prenos otvorenej slučky po kompenzovaní nuly prenosu:

$$F_{0i_q}(p) = F_{Si_q}(p) F_{Ri_q}(p) = \frac{K}{p} \quad (23)$$

Kde $K = K_{Si_q} * K_{Ri_q}$. Výsledný prenos riadenia:

$$F_{Wi_q}(p) = \frac{F_{0i_q}(p)}{1 + F_{0i_q}(p)} = \frac{1}{1 + \frac{1}{K} p} \quad (24)$$

Regulátor druhej zložky prúdu i_d bude odvodený rovnakým spôsobom.

Sústavu pre rýchlostnú slučku odvodíme z (19), (20) a (24). Za predpokladu nulového záťažného momentu, je Laplaceov prenos sústavy:

$$F_{S\omega}(p) = F_{Wi_q}(p) \frac{3p\psi_f}{2J_c} \frac{1}{p} = \frac{3p\psi_f}{2J_c} \frac{1}{p(1 + \frac{1}{K} p)} = \frac{K_{S\omega}}{p(1 + \frac{1}{K} p)} \quad (25)$$

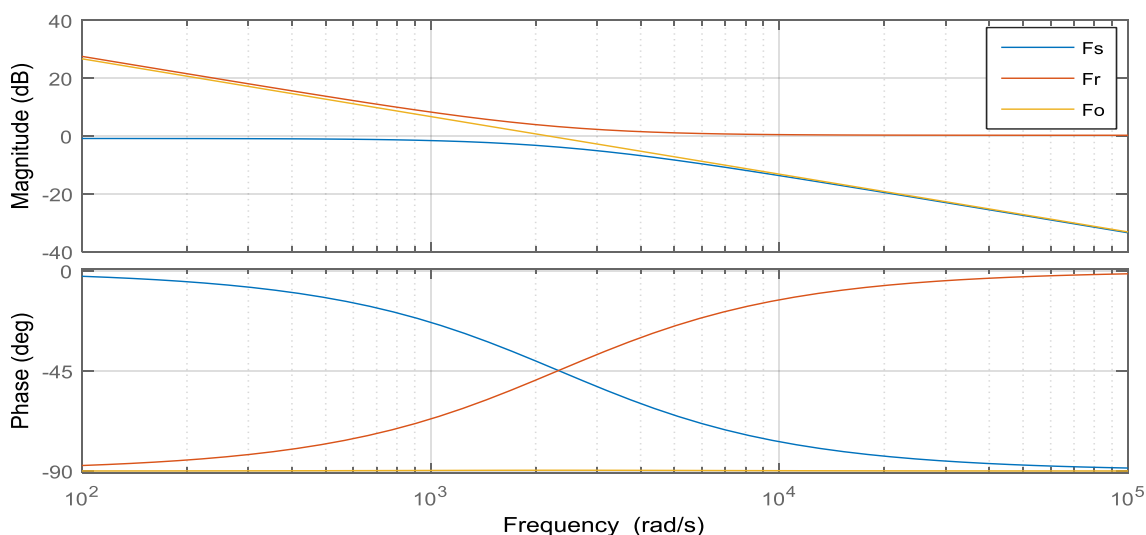
kde $K_{S\omega} = \frac{3p\psi_f}{2J_c}$. Ako regulátor otáčok sa používa regulátor typu PI alebo PID.

Regulátory prúdov i_q a i_d sme navrhli tvarovaním frekvenčných charakteristík pomocou programu Matlab a jeho nástroja Sisotool. Parametre motora pre ktorý sme regulátory navrhovali nájdeme v kapitole 6.1.3 v tabuľke číslo 1. Výsledné prenosy regulátorov sú nasledujúce:

$$F_{Ri_q}(p) = 1,03 \left(\frac{p + 2312}{p} \right) \quad (26)$$

$$F_{Ri_d}(p) = 1,05 \left(\frac{p+2868}{p} \right) \quad (27)$$

Frekvenčné charakteristiky sústavy, regulátoru a otvorenej smyčky sú na nasledujúcich grafoch získaných pomocou funkcie *bode()* v programe Matlab. Vzhľadom k tomu, že prenosy pre q zložku a d zložku sa takmer zhodujú, uvádzame grafy len pre jednu zložku:

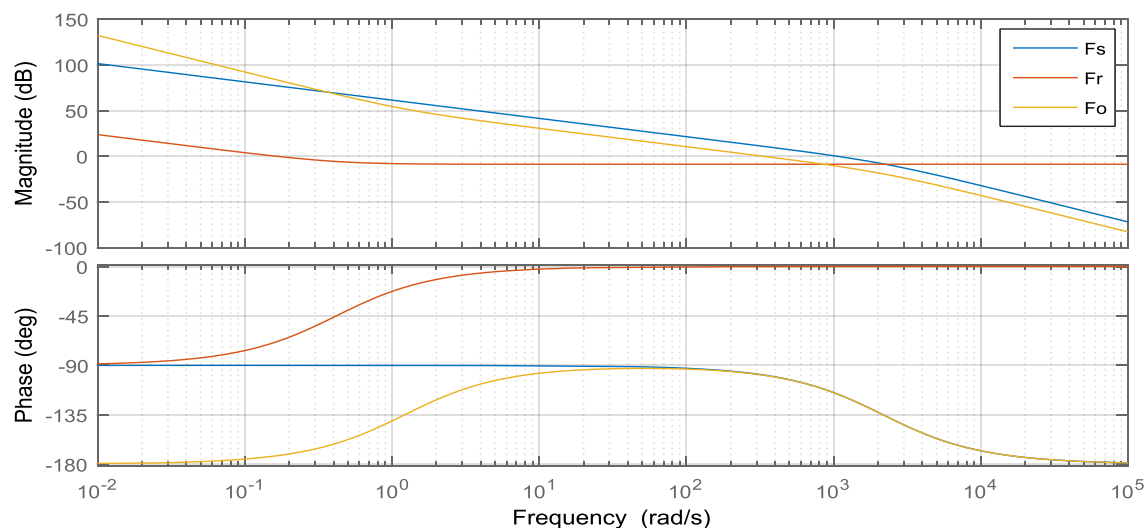


Obrázok 13: Frekvenčné charakteristiky prúdových slučiek

PI regulátor otáčok bol navrhnutý rovnakým spôsobom. Výsledný prenos regulátoru otáčok:

$$F_{R\omega}(p) = 0,364 \left(\frac{p+0,4171}{p} \right) \quad (28)$$

Frekvenčné charakteristiky:



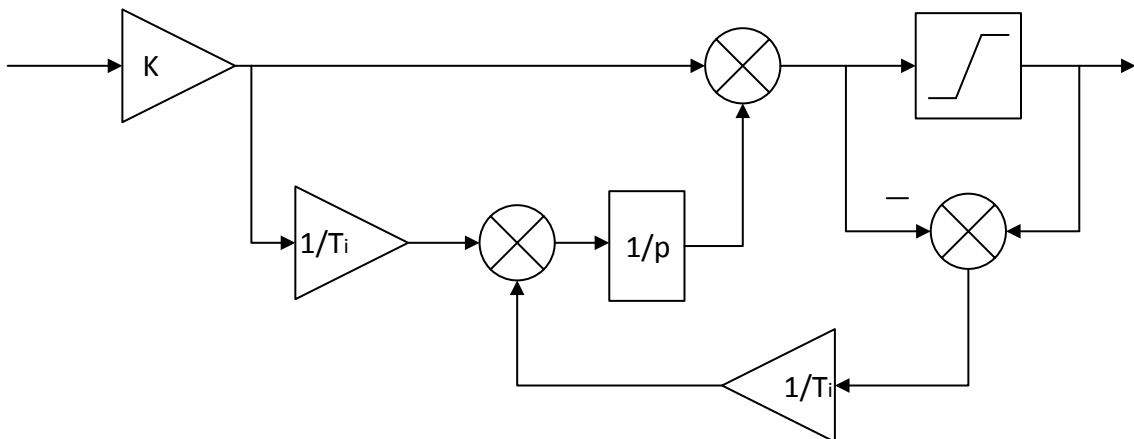
Obrázok 14: Frekvenčné charakteristiky otáčkovej slučky

6.1.2 Anti wind-up

V každom reálnom fyzikálnom systéme existuje určitá medz, nad ktorú regulátory nemôžu pôsobiť. Pri motoroch sa jedná o obmedzenie elektrického napätia a elektrických prúdov.

Ak vložíme za výstup regulátoru nelinearitu typu nasýtenie, obmedzíme tým síce výstup regulátoru, no ak regulátor obsahuje integračnú zložku, nastáva takzvaný wind-up efekt. Regulátor nevie o tom, že sa za ním nachádza obmedzenie, a integračná zložka tak bude aj naďalej narastať. Ak sa zmení znamienko regulačnej odchýlky, bude nejakú dobu trvať, kým sa hodnota v integračnej zložke „odintegruje“ [10]. Toto má zlý vplyv na regulačný dej, pretože je pomalý a môžu v ňom nastávať pomerne veľké výchylky. Riešenie hľadáme v tzv. anti wind-up algoritme.

V simulácii sme použili riešenie, kedy sa rozdiel medzi výstupom regulátoru a hodnotou signálu za nasýtením privádza cez určité zosilnenie naspäť do integračnej zložky a zabraňuje tak wind-up javu. Riešenie pre PI regulátor zobrazuje obrázok. Konštantu T sme v simulácii volili rovnakú ako konštantu T_i .



Obrázok 15: Realizácia anti wind-up na PI regulátore

6.1.3 Simulácia vektorového riadenia PMSM

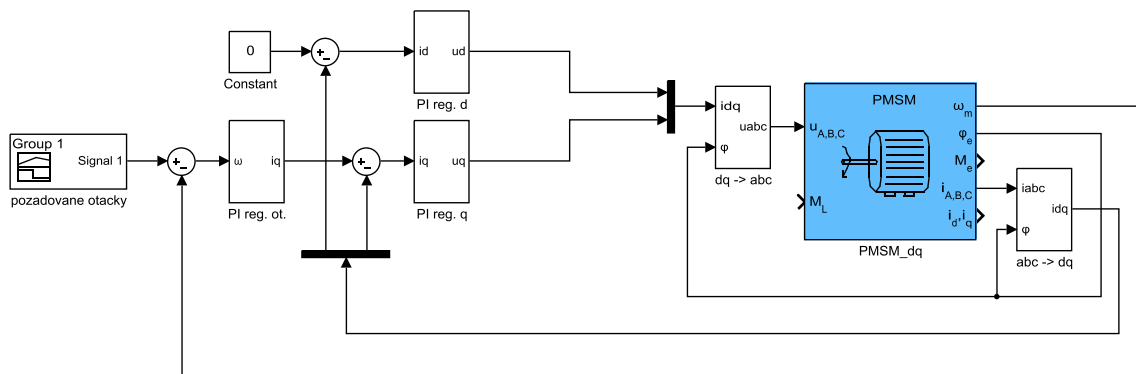
Parametre modelu PMSM

Parametre motoru ktorý bol použitý pre demonštráciu vektorového riadenia sú uvedené v nasledujúcej tabuľke:

Činný odpor statorovej fázy	$R_s = 1,1 \Omega$
Počet pólových dvojíc	$p_p = 3$
Celkový moment zotrvačnosti	$J_m = 8 \cdot 10^{-5} \text{ kg} \cdot \text{m}^2$
Indukčnosť v priamom smere	$L_d = 3,9 \cdot 10^{-4} \text{ H}$
Indukčnosť v kolmom smere	$L_q = 4,7 \cdot 10^{-4} \text{ H}$
Magnetický tok rotora	$\Psi_f = 0,0208 \text{ Wb}$
Obmedzenie statorového prúdu	10 A
Obmedzenie statorového napätia	35 V

Tabuľka 1: Parametre PMSM

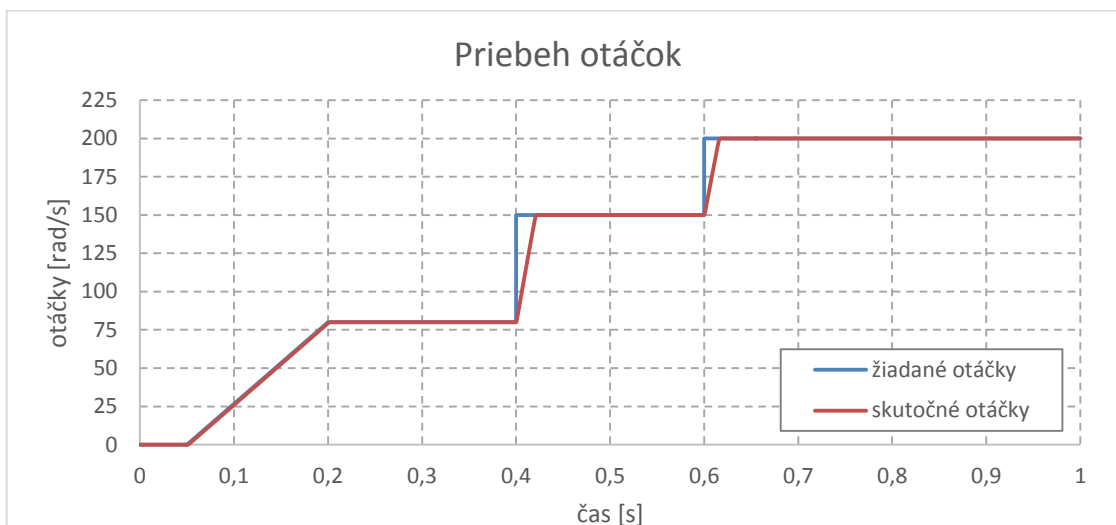
Simulačná schéma



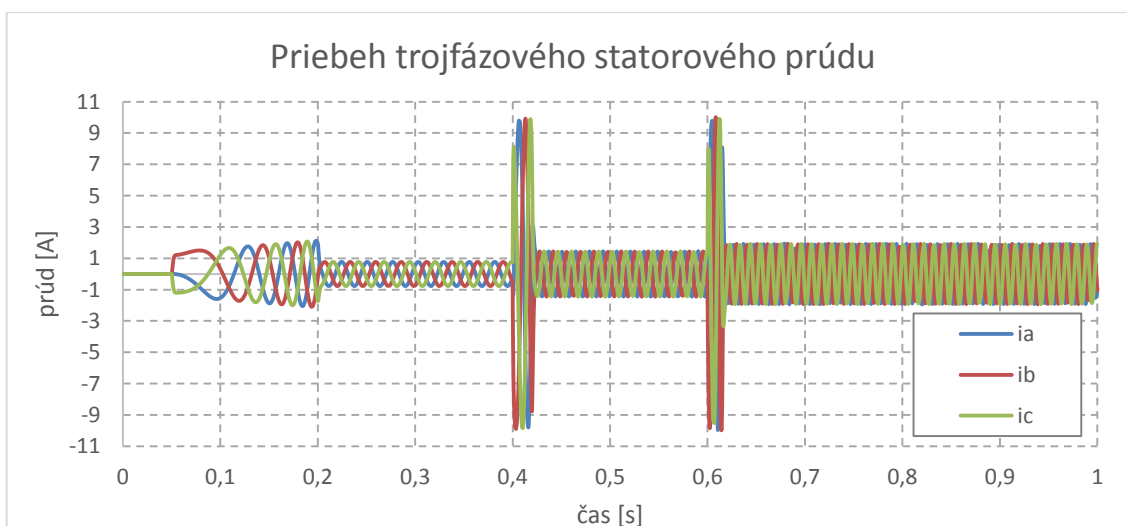
Obrázok 16: Simulačná schéma vektorového riadenia v programe Simulink

Všetky regulátory v simulačnej schéme na obrázku 16 v sebe obsahujú anti wind-up v zapojení z obrázku 15.

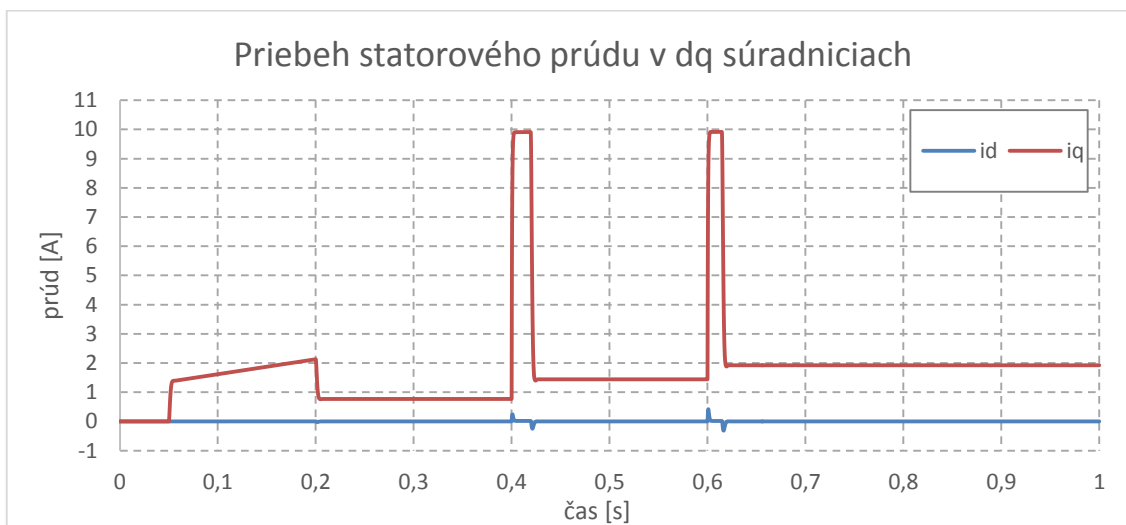
Simulované priebehy



Obrázok 17: Priebeh otáčok



Obrázok 18: Priebeh trojfázového statorového prúdu



Obrázok 19: Priebeh statorového prúdu v dq súradniciach

6.2 Experiment

Táto kapitola sa zaoberá realizáciou vektorového riadenia na reálnom motore. Experiment prebiehal na pracovisku v zapojení, ktoré je uvedené v kapitole číslo 4. Pracovisko pozostávalo z vývojového počítača, v ktorom sa vytváral riadiaci algoritmus v prostredí Matlab - Simulink. Algoritmus bol nahraný a následne spustený na cieľovom počítači, na ktorom bežal RTOS SLRT. Cieľový počítač bol vybavený meracou kartou MF624. Táto I/O karta slúžila na:

- Sledovanie signálu z inkrementálneho enkóderu
- Meranie prúdu na jednotlivých fázach motoru
- Generovanie PWM signálu, na základe ktorého sa pomocou výkonovej dosky aplikoval akčný zásah na jednotlivé fázy motoru

6.2.1 Popis algoritmu

Riadiaci algoritmus funguje na rovnakom princípe ako pri simulácii, ktorá je popísaná v predchádzajúcej kapitole. V tejto kapitole si uvedieme postup ktorým sme sprevádzkovali riadenie na reálnom systéme.

- Celý program je riadený prepínačom medzi tromi stavmi. Stavom sú priradené nasledovné funkcie:
- Stav 0 – na fázy motora nie je privádzané žiadne napätie
- Stav 1 – na fázy motora sú privádzané napätia, ktoré zabezpečia zarovnanie motora na prvú fázu (fáza A)
- Stav 2 – na fázy motora sú privádzané napätia, ktoré zodpovedajú akčným zásahom prúdových regulátorov, ktoré boli vypočítané zo žiadanej hodnoty prúdov, prípadne zo žiadanej hodnoty otáčok. Zároveň sa vyresetujú integrátory v PI regulátoroch

Algoritmus bol namodelovaný v Simulinku a jeho model je zobrazený v prílohe A. Jeho popis je uvedený na nasledovných riadkoch.

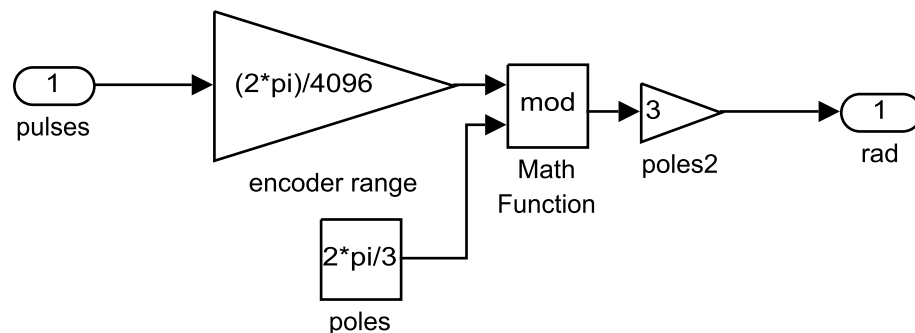
Zarovnanie rotora a výpočet uhlu natočenia

Po prepnutí do stavu 1 sa rotor zarovná na elektrickú fázu A. To je zabezpečené tak, že na fázu A sa privedie isté malé kladné napätie, na fázy B a C sa privedú záporné napätia, ktoré majú polovičnú veľkosť oproti napätiu privedenému na fázu A.

Blok zastupujúci vstup inkrementálneho enkóderu na karte MF624 napočíta počet pulzov, o ktoré sa rotor posunul pri zarovnaní. Táto hodnota zostáva po prepnutí do ďalšieho stavu (beh motoru) zapamätaná. To je zabezpečené ďalším prepínačom.

Následne sa pomocou rozdielového bloku vypočíta skutočná hodnota natočenia rotora v pulzoch. Skutočnú hodnotu natočenia v pulzoch je potrebné prepočítať na elektrický uhol. Tento prepočet je realizovaný funkciou modulo, výsledný uhol je potom v intervale $\langle 0; 2\pi \rangle$.

Elektrický uhol používame v algoritme na prepočet medzi súradnicovými sústavami dq - $\alpha\beta$. Nasledujúci obrázok zobrazuje subsystém výpočtu elektrického uhlu.



Obrázok 20: Výpočet elektrického uhlu

Prúdová sľučka

Rozdiel medzi žiadanou a skutočnou hodnotou prúdov i_d a i_q kompenzujú PI regulátory, ktoré odchýlku transformujú na napätia u_d a u_q . Napätia v dq súradniciach sa prepočítajú do abc súradnicového systému pomocou Parkovej a Clarkovej transformácie. Tieto hodnoty napätí sa následne musia prepočítať do intervalu $\langle 0; 1 \rangle$, kde:

- 0 zodpovedá -100% striede PWM
- 0,5 zodpovedá 0% striede PWM
- 1 zodpovedá 100% striede PWM

Takto prepočítané hodnoty napätia sú privedené na vstupy bločku MF624 PWM.

Skutočné hodnoty prúdu na fázach motora sú pomocou LEM prevodníkov transformované na napätia, ktoré sú privedené na analógové vstupy karty MF624. V algoritme je potom výstup bločku, reprezentujúci analógové vstupy karty, cez prevodnú konštantu spätne transformovaný na fázové prúdy. Prúdy po prepočte do dq súradnicového systému a vypočítaní odchýlky od žiadaných hodnôt, vstupujú do PI regulátorov.

Otáčková sľučka

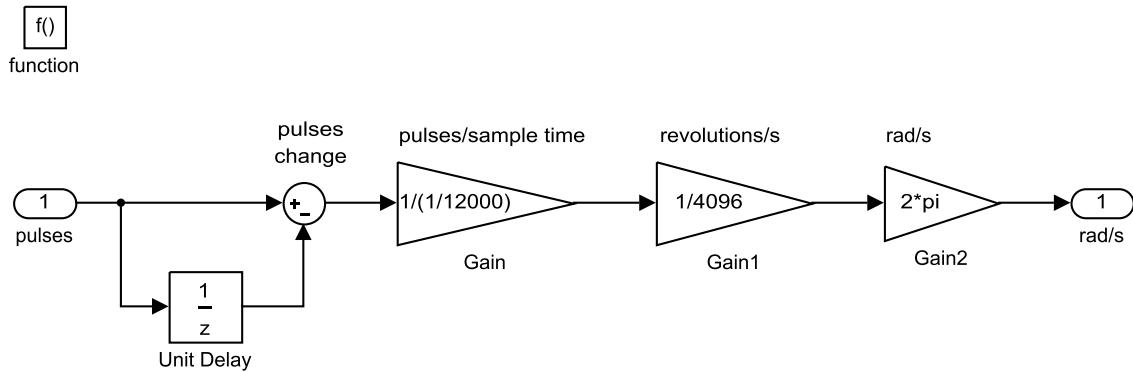
Rozdiel medzi žiadanou a skutočnou hodnotou otáčok kompenzuje PI regulátor, ktorý odchýlku transformuje na žiadanú hodnotu prúdu i_q .

Skutočnú hodnotu otáčok získavame výpočtom z počtu pulzov inkrementálneho enkóderu za časový interval. Výpočet prebieha nasledovne:

- Skutočná hodnota natočenia rotora v pulzoch vstupuje do funkciou volaného subsystému pre výpočet rýchlosti
- V subsystéme sa od aktuálnej hodnoty pulzov odpočíta hodnota z predchádzajúcej iterácie
- Rozdiel sa podelí časovým intervalom – vzorkovacou periódou simulácie. Dostávame tak zmenu počtu pulzov za časový interval

- Táto hodnota sa podelí rozlíšením inkrementálneho enkóderu – získame rýchlosť v pulzoch za časový interval
- Následným prenasobením vhodnou konštantou dostaneme rýchlosť v požadovaných jednotkách (rad/s, otáčky/min,...)

Subsystém výpočtu rýchlosti je zobrazený na nasledujúcom obrázku.

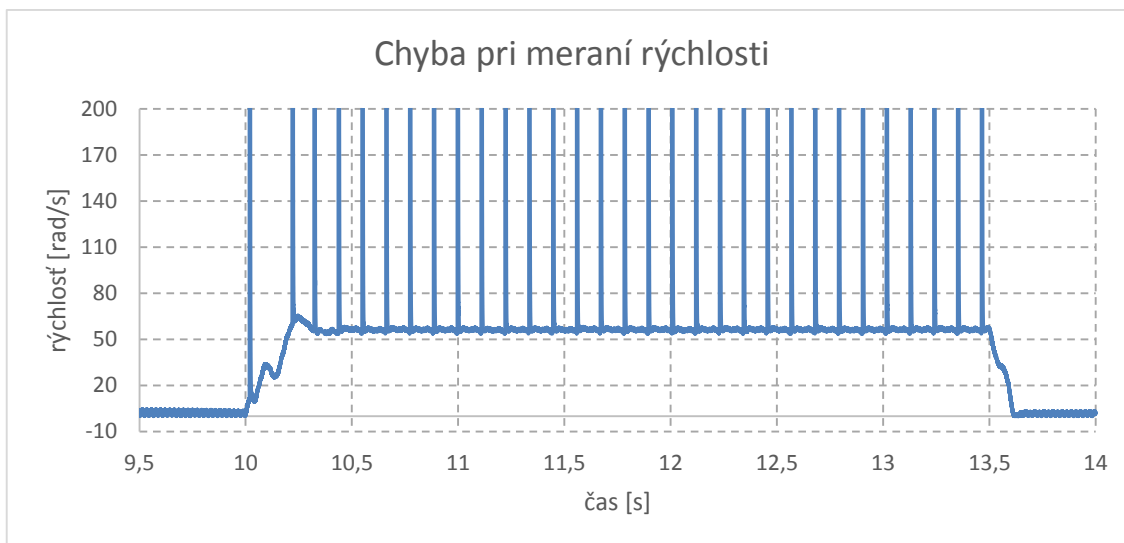


Obrázok 21: Výpočet rýchlosti

Výstupná hodnota bola značne zašumená, čo sme vyriešili pridaním dolnopriepustného filtra prvého rádu. Časovú konštantu filtra sme experimentálne určili na 0,001. Následným zdiskretizovaním pomocou funkcie *c2d* v programe Matlab, kde sme periódu vzorkovania zvolili rovnakú pri akej beží model na cieľovom počítači, sme získali filter v diskretnom tvare. Tento filter sme vložili za výstup bloku na výpočet rýchlosti. Výsledná hodnota zodpovedá skutočnej rýchlosti, čo bolo overené pomocou kontaktného snímača otáčok.

Z takto určenej rýchlosti pomocou rozdielového bloku vypočítame odchýlku od žiadanej hodnoty a tú kompenzuje PI regulátor.

Pri sprevádzkovaní merania otáčok dochádzalo ku chybe, ktorú sa nám podarilo odstrániť len nie veľmi elegantným spôsobom. V prípade, že bol v bločku pre enkóder povolený index pulz, dochádzalo v nameraných hodnotách k pravidelným veľkým odchýlkám – a to konkrétne 3 krát za jednu mechanickú otáčku – pri každej elektrickej otáčke. Chybu sme sa snažili vyriešiť rôznymi spôsobmi, napríklad: veľmi presné nastavenie index pulzu na fázu, na ktorú sa zarovnávalo, zarovnávanie na inú fázu, filtrovanie signálu alebo pokusmi o iný spôsob výpočtu rýchlosti. Pri žiadnom z pokusov sme neboli úspešní, index pulz sme preto vypli. Očakávali sme výrazné zvýšenie chyby pri počítaní uhlu natočenia a následné prebudzovanie/odbudzovanie motora, no ani pri dlhom behu motoru k tomuto nedochádzalo. Na nasledujúcom obrázku je zobrazený nameraný priebeh rýchlosti s vyššie popísanými chybami.



Obrázok 22: Chyba pri meraní rýchlosti

PI regulátor otáčok, ani PI regulátory prúdu neboli ošetrené anti wind-up zapojením, ako tomu bolo v simulácii. Dôvodom je, že akčné zásahy regulátorov nedosahovali hodnoty, ktoré by prekračovali medze laboratórneho zdroja alebo maximálne prípustné hodnoty prúdu motorom.

6.2.2 Návrh regulátorov

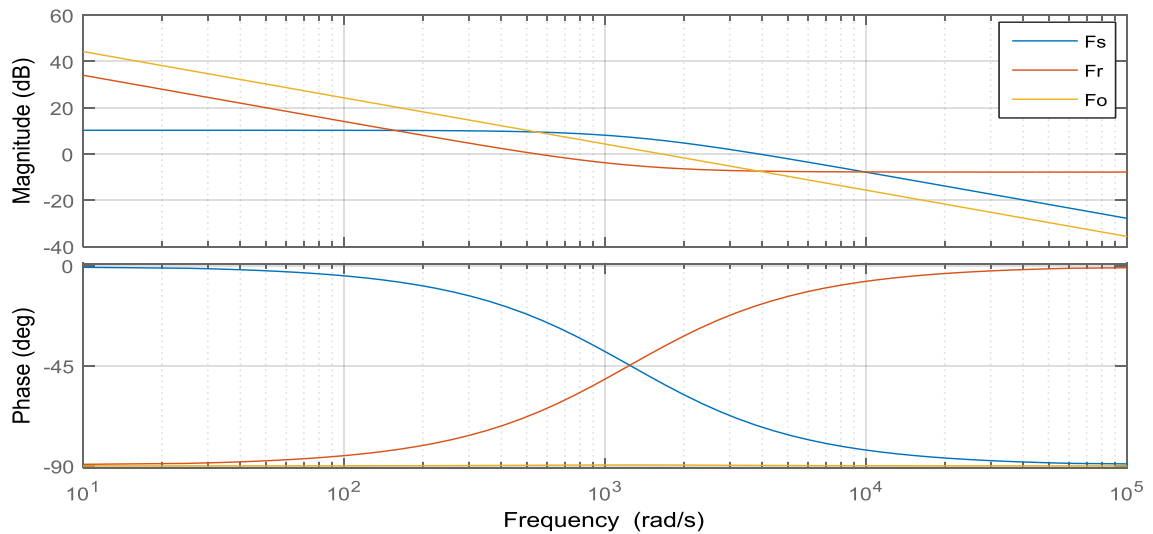
Regulátory pre experiment s reálnym motorom sme navrhovali rovnakým spôsobom ako v prípade simulácie s modelom motoru. Parametre motoru na ktorý boli regulátory navrhnuté sa nachádzajú v kapitole 6.2.3.

PI regulátory prúdových zložiek d a q fungovali uspokojivo s rovnakými konštantami v prenosoch. Výsledné prenosy boli nasledujúce:

$$F_{Ri_d}(p) = 1 \left(\frac{p+500}{p} \right) \quad (29)$$

$$F_{Ri_d}(p) = 1 \left(\frac{p+500}{p} \right) \quad (30)$$

Frekvenčné charakteristiky:

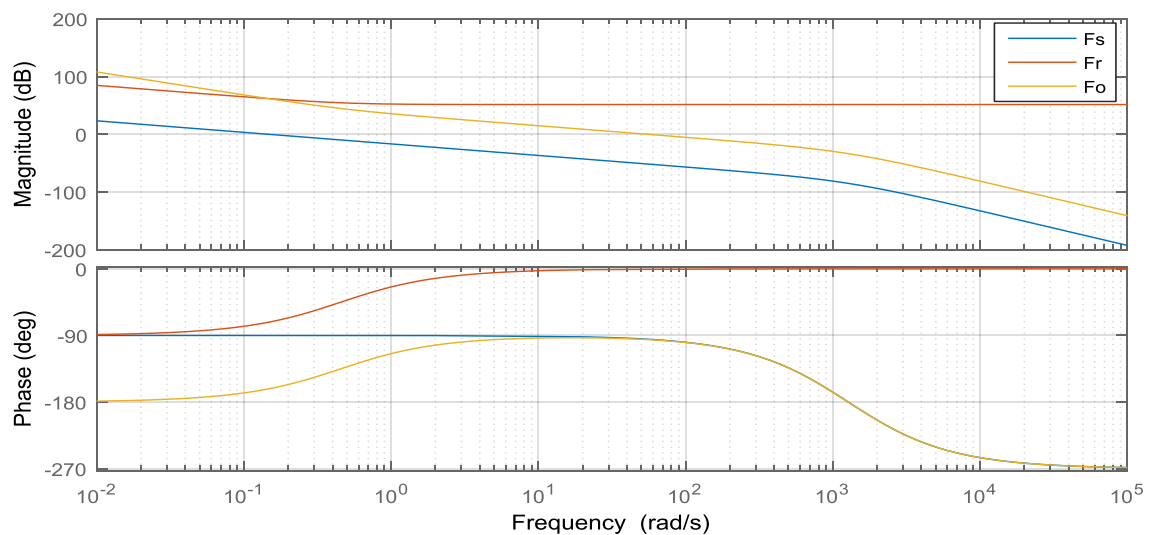


Obrázok 23: Frekvenčné charakteristiky prúdových slučiek

Otáčkový PI regulátor sme museli navrhnuť s ohľadom na použitý dolnopriepustný filter. Tento filter nám ovplyvňuje sústavu a regulátor sa tak musí navrhovať na sústavu pozostávajúcu z motora a filtra. Prenos otáčkového regulátoru:

$$F_{R\omega}(p) = 374,98 \left(\frac{p+0,4552}{p} \right) \quad (31)$$

Frekvenčné charakteristiky:



Obrázok 24: Frekvenčné charakteristiky otáčkovej slučky

6.2.3 Výsledky HIL simulácie

Parametre PMSM

Pri HIL simulácii sme mali k dispozícii motor značky TG drives, typ SBL2-0032-3024/T0PKX. Jeho parametre sú zhrnuté v nasledovnej tabuľke:

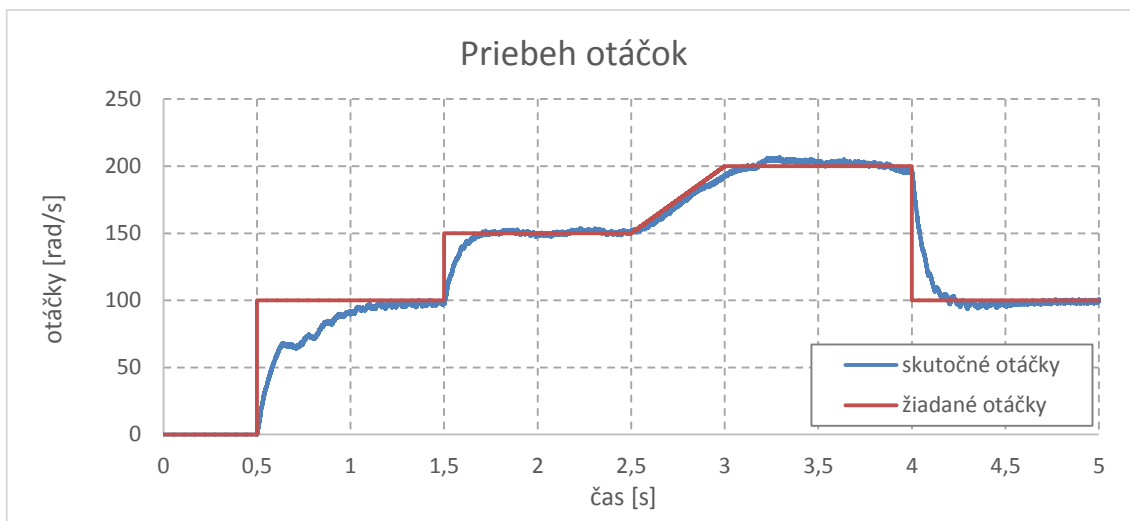
Činný odpor statorovej fázy	$R_s = 308 \text{ m}\Omega$
Počet pólových dvojíc	$p_p = 3$
Indukčnosť v priamom smere	$L_d = 173 \text{ }\mu\text{H}$
Indukčnosť v kolmom smere	$L_q = 246 \text{ }\mu\text{H}$
Magnetický tok rotoru	$\Psi_{PM} = 0,01036 \text{ Wb}$

Tabuľka 2: Parametre PMSM

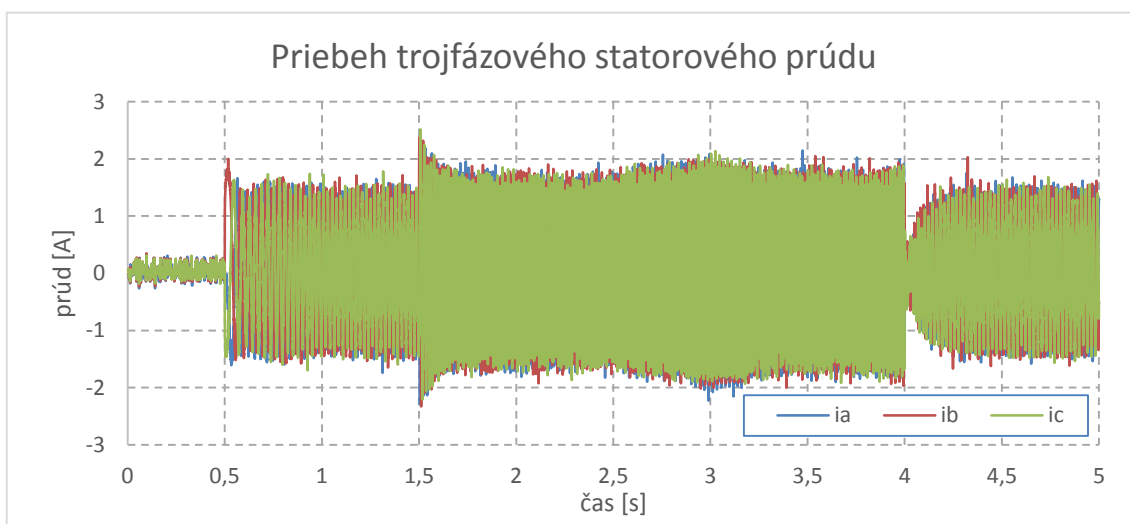
Namerané priebehy

V tejto časti sú zobrazené priebehy demonštračných meraní. Namerané údaje sa nezískavajú tak jednoducho, ako pri bežnej simulácii, nakoľko HIL simulácia nebeží na vývojovom počítači. Podarilo sa nám efektívne využívať dva spôsoby získavania meraných veličín:

- Vo vytvorenom modeli celého systému sa použijú bločky Real-Time Scope - typ File Scope. Pri použití tohto scope sa namerané dáta ukladajú na pevný disk cieľového počítača. V našom prípade cieľový počítač pevným diskom nedisponoval, no po vložení USB flash disku sformátovaného na systém súborov FAT sa dáta automaticky ukladali naňho. Jedná sa o štandardný dátový súbor programu Matlab s príponou *.dat, ktorý treba ďalej spracovať v Matlabe. Pri použití niekoľkých File Scope sa na flash disku vytvorí viac súborov
- Vo vytvorenom modeli celého systému sa použijú bločky Outport. V konfigurácii simulácie sa v karte Data Import/Export zaškrtnie checkbox Signal logging a ako Signal logging format sa vyberie Dataset. Po spustení simulácie a jej následnom ukončení môžeme k nameraným dátam prísť z programu Matlab a to cez automaticky vygenerovaný objekt tg, ktorý nájdeme vo Workspace (generuje sa pri kompilácii modelu, pokiaľ je ako System Target File zvolený slrt.tcl). Nevýhodou je, že takýmto spôsobom môžeme sledovať len jeden signál



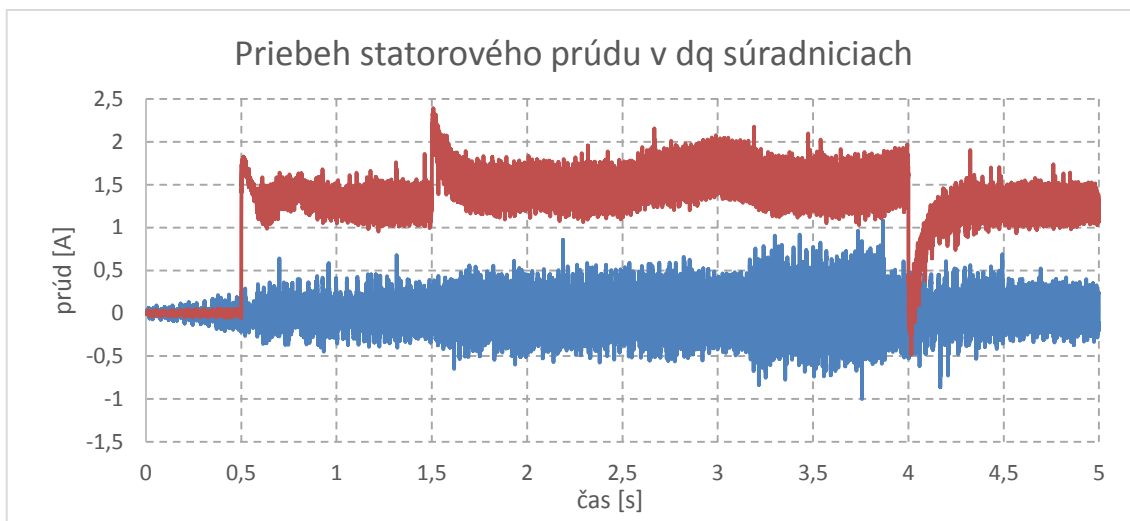
Obrázok 25: Priebeh otáčok



Obrázok 26: Priebeh trojfázového satorového prúdu



Obrázok 27: Detail priebehu trojfázového satorového prúdu



Obrázok 28: Pribeh statorového prúdu v dq súradniciach

7 ZÁVER

Hlavným cieľom tejto bakalárskej práce bolo zrealizovanie HIL simulácie riadenia PMSM na platforme pozostávajúcej z osobného počítača s operačným systémom reálneho času Simulink Real-Time a s meracou kartou MF624.

Začali sme postupným zoznamovaním sa s RTOS Simulink Real-Time a vytváraním algoritmov pre tento systém s využitím karty MF624. Najprv sme sprevádzkovali všetky potrebné vstupy a výstupy karty. Pri tomto sme narazili na prvý problém, kedy dochádzalo k zastaveniu simulácie na cieľovom počítači s chybovou hláškou „CPU overload“ už pri veľmi nízkych vzorkovacích frekvenciách. Tento problém sme vyriešili výmenou cieľového počítača za iný, s výkonnejším štvorjadrovým procesorom Intel Core i7. Avšak ani na tomto počítači sme nedosiahli ideálne hodnoty vzorkovacej frekvencie pre riadenie motoru, ale dostali sme sa aspoň na 12 kHz, čo pre riadenie prúdovej sľučky postačuje.

Na riadenie PMSM sme využívali trojfázovú nízkonapäťovú dosku Freescale osadenú budičom gate-ov MOSFET tranzistorového mostíka. Do dosky vstupoval PWM signál generovaný kartou MF624, na základe ktorého budič riadil tranzistorový most. Funkcia dosky a a inicializácia budiča boli dostatočne popísané v kapitole číslo 5.

Meranie prúdu na fázach motoru bolo realizované pomocou LEM prevodníkov fungujúcich na princípe Hall efektu. Signál z týchto prevodníkov vstupoval do karty MF624 a následne bol softwarovo spracovaný.

Úlohu sledovania natočenia rotoru mal na starosti inkrementálny enkóder, ktorého výstup taktiež spracovávala karta MF624. Úspešne sme zrealizovali prepočet mechanického natočenia na elektrický uhol a taktiež výpočet rýchlosti otáčania. Pri tomto sme narazili na ďalší problém, ktorý je popísaný v kapitole 6.2.1.

Súčasne s vyššie popísaným sme sa zoznamovali s princípmi vektorového riadenia, ktoré sú popísané v poslednej kapitole. Algoritmus vektorového riadenia bol najprv demonštrovaný na simulácii. Regulátory prúdov a otáčok boli navrhnuté metódou tvarovania frekvenčných charakteristík za pomoci programu Matlab a jeho nástroja Sisotool.

Na záver sme kompletne zrealizovali simuláciu vektorového riadenia v reálnom čase metódou Hardware-in-the-loop na reálnom PMSM. Celý algoritmus riadenia bol vytvorený v programe Simulink a bol nahraný na cieľovú platformu na ktorej bežal. Algoritmus je detailne popísaný v kapitole 6.2.1. Regulátory boli navrhnuté rovnakou metódou ako pri simulácii a ich konštanty sa následne doladili ručne za behu HIL simulácie.

Literatúra

- [1] Neborák, I.: Modelování a simulace elektrických regulovaných pohonů, VŠB - Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2002, 172 s.
- [2] Help programu MATLAB
- [3] Simulink Real-Time, Build, run and test real-time applications, MathWorks, Dostupné z: <http://www.mathworks.com/products/simulink-real-time/>
- [4] Simulace Processor In the Loop a Hardware In the Loop, Pavel Jelínek, časopis AUTOMA, 05/2007, Dostupné z: http://automa.cz/index.php?id_document=34311
- [5] MF624 Multifunction I/O Card user's manual, HUMUSOFT, 2014, Dostupné z: <http://www2.humusoft.cz/www/datacq/manuals/mf624um.pdf>
- [6] MF624 – PCI multifunction I/O card, HUMUSOFT, Dostupné z: <http://www.humusoft.cz/datacq/mf624/>
- [7] Encoders, ROCKWELL AUTOMATION, Dostupné z: <http://www.ab.com/en/epub/catalogs/12772/6543185/12041221/12041235/Incremental-Versus-Absolute-Encoders.html>
- [8] 3-Phase BLDC/PMSM Low Voltage Power Stage User Guide, Freescale Semiconductor, 2012, Dostupné z: http://cache.nxp.com/files/microcontrollers/doc/user_guide/3PHLVPSUM.pdf
- [9] Three Phase Field Effect Transistor Pre-driver, Freescale Semiconductor, 2015, Dostupné z: http://cache.nxp.com/files/analog/doc/data_sheet/MC33937.pdf
- [10] Šolc F., Václavek P., Vavřín P.: Řízení a regulace II, analýza a řízení nelineárních systémů, Vysoké učení technické, Fakulta elektrotechniky a komunikačních technologií, Brno 2006, 228 s.

Zoznam príloh

Príloha A Schéma riadiaceho algoritmu z programu Simulink

Príloha A

