



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

## MODELOVÁNÍ ELEKTRICKÝCH POHONŮ NA PLATFORMĚ COMPACTRIO S VYUŽITÍM FPGA

MODELING OF ELECTRICAL DRIVES ON THE COMPACTRIO PLATFORM USING FPGA

### DIPLOMOVÁ PRÁCE

MASTER'S THESIS

### AUTOR PRÁCE

AUTHOR

**Bc. Karel Kozumplík**

### VEDOUCÍ PRÁCE

SUPERVISOR

**Ing. Libor Veselý, Ph.D.**

**BRNO 2016**



# Diplomová práce

magisterský navazující studijní obor **Kybernetika, automatizace a měření**  
Ústav automatizace a měřicí techniky

**Student:** Bc. Karel Kozumplík

**ID:** 146867

**Ročník:** 2

**Akademický rok:** 2015/16

## NÁZEV TÉMATU:

### **Modelování elektrických pohonů na platformě CompactRIO s využitím FPGA**

## POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je vytvoření automaticky generovaného HDL kódu, který bude implementován do FPGA na platformě CompactRIO. Toto zařízení, bude emulovat chování elektrického pohonu v reálném čase.

1. Provedte literární rešerši modelování různých typů elektrických pohonů.
2. Vytvořte model zvoleného elektrického pohonu v prostředí Matlab Simulink.
3. Vytvořte diskrétní model vhodný pro automatické generování HDL kódu.
4. Real-time testování na platformě CompactRIO a ověření funkčnosti modelu.

## DOPORUČENÁ LITERATURA:

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

**Termín zadání:** 8.2.2016

**Termín odevzdání:** 16.5.2016

**Vedoucí práce:** Ing. Libor Veselý, Ph.D.

**Konzultant diplomové práce:**

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

## UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## **Abstrakt**

Tato diplomová práce se zabývá modelováním elektrických pohonů na platformě CompactRIO s využitím FPGA. Konkrétně je zde popsán, namodelován a otestován synchronní motor s permanentními magnety (PMSM).

V práci je uveden kompletní postup vytvoření modelu simulujícího chování elektrického pohonu v reálném čase, jsou zde probrány různé druhy diskretizace a model je diskutován z hlediska přesnosti simulace i z hlediska vytížení hradlového pole.

Na závěr je v práci popsáno testování vytvořeného modelu s reálným motorem i s výsledky tohoto testování.

## **Klíčová slova**

Elektrické pohony, PMSM, CompactRIO 9082, modelování elektromotorů, FPGA

## **Abstract**

This Master's thesis engage in modeling of electric motions on CompactRIO platform with FPGA. Permanent magnet synchronous motor (PMSM) is namely described, modeled and tested here.

This thesis demonstrates the complete work procedure for modeling of electric motion and testing behavior of motor in real time. Various types of discretization are described here and model si discussed from accuracy same as from FPGA utilization point of view.

The last part of this project describes testing comparison between created model and real PMSM motor and results from this testing.

## **Keywords**

Electric motion, PMSM, CompactRIO 9082, Modeling of electric motors, FPGA

## **Bibliografická citace:**

KOZUMPLÍK, K. *Modelování elektrických pohonů na platformě CompactRIO s využitím FPGA*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2016. 92s.

Vedoucí diplomové práce byl Ing. Libor Veselý, Ph.D.

## Prohlášení

Prohlašuji, že svoji diplomovou práci na téma Modelování elektrických pohonů na platformě CompactRIO s využitím FPGA jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením tohoto projektu jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009Sb.

V Brně dne .....

.....

Kozumplík Karel

# Obsah

Použité značení.....	8
1 Úvod.....	10
2 Způsob řešení.....	11
2.1 Logické schéma úlohy.....	11
2.2 Použitý Hardware.....	12
2.3 Použitý Software.....	14
3 Teoretický úvod.....	15
3.1 Elektrické motory a jejich třídění.....	15
3.1.1 Asynchronní motor.....	17
3.1.2 Synchronní motor.....	17
3.2 Souřadné systémy, transformace.....	18
3.2.1 Clarkové transformace.....	19
3.2.2 Parkova transformace.....	21
3.3 Diskretizace systému.....	22
4 Přípravné úlohy.....	24
4.1 Sčítačka dvou desetinných čísel.....	24
4.1.1 Cíl úlohy.....	24
4.1.2 Realizace – Simulink.....	25
4.1.3. Realizace – Labview.....	26
4.2 Implementace funkce sinus.....	28
4.2.1 Cíl úlohy.....	28
4.2.2 Realizace - Simulink.....	28
4.2.3 Realizace – Labview.....	29
5 Vektorové řízení.....	31
6 Matematický model PMSM motoru.....	34
6.1 Spojité modely PMSM.....	35
6.1.1 Model PMSM motoru v abc souřadnicích.....	35
6.1.2 Model PMSM motoru v abc souřadnicích – s obecným rozložením indukčností.....	41
6.1.3 Model PMSM motoru v dq souřadnicích.....	44

6.2 Diskrétní modely PMSM.....	46
6.2.1 Diskrétní model PMSM v abc souřadnicích 1.řádu.....	47
6.2.2 Diskrétní model PMSM v dq souřadnicích 1.řádu.....	48
6.2.3 Diskrétní model PMSM v dq souřadnicích 2.řádu.....	49
7 Realizace Matlab.....	50
7.1 Finální modely v Simulinku.....	50
7.1.1 Diskrétní model Eulerovou diskretizací v dq systému.....	51
7.1.2 Diskrétní model kvadratickou diskretizací v dq systému.....	52
7.1.3 Diskrétní model Eulerovou diskretizací v abc systému.....	53
7.2 Testovací schéma v Simulinku.....	54
7.3 Volba periody vzorkování.....	58
7.4 Nastavení Fixed Pointů.....	60
8 Realizace LabVIEW.....	63
8.1 FPGA část s modelem a řízením.....	63
8.2 Transformace souřadného systému v LabVIEW.....	66
8.3 RT testování modelu.....	68
8.4 Srovnání zatížení CompactRIA.....	73
9 Porovnání modelu s reálným motorem.....	75
9.1 Logické schéma porovnání modelu s reálným motorem.....	75
9.2 Nastavení parametrů modelu a regulátorů.....	77
9.3 Výsledky testování shody modelu a reálného motoru.....	80
Závěr.....	86
Seznam obrázků.....	87
Seznam tabulek.....	88
Seznam grafů.....	89
Literatura.....	90
Seznam příloh.....	92

# POUŽITÉ ZNAČENÍ

## Zkratky

FPGA	Field Programmable Gate Array	Programovatelné hradlové pole
HDL	Hardware Description Language	Jazyk pro popis hardwaru
cRIO	CompactRIO	CompactRIO
NI	National Instruments	National Instruments
DC	Direct Current	Stejnoseměrný proud
AC	Alternate Current	Střídavý proud
EM	Electric Motor	Elektrický motor
VC	Vector Control	Vektorové řízení
ENC	Encoder	Enkodér
TD	Tachodynamo	Tachodynamo
PMSM	Permanent Magnet Synchronous Motor	Synchronní motor s permanentními magnety
PWM	Pulse Width Modulation	Pulsně šířková modulace

## Indexy

a	statorová fáze a
b	statorová fáze b
c	statorová fáze c
f	buzení
m	mechanický
r	rotorový
s	statorový
$\alpha$	reálná osa statorového souřadného systému
$\beta$	imaginární osa statorového souřadného systému
d	osa d souřadného systému spojeného s rotorem
q	osa q souřadného systému spojeného s rotorem
tl	tlumení



## Značky a symboly

f	[Hz]	kmitočet
i, I	[A]	proud
J	[kg m <sup>2</sup> ]	moment setrvačnosti
M	[Nm]	moment
n	[ot/min]	otáčky
p	[-]	operátor v Laplaceově transformaci
p <sub>p</sub>	[-]	počet pólových dvojic motoru
R	[Ω]	činný odpor
t	[s]	čas
u, U	[V]	napětí
Ω	[rad/s]	mechanická úhlová rychlost
Ψ	[Wb]	spřažený magnetický tok
ω	[rad/s]	elektrická úhlová rychlost
θ	[rad]	úhlové natočení motoru
L	[H]	indukčnost
L <sub>s</sub>	[H]	matice indukčností v abc souřadnicích
- L <sub>ls</sub>	[H]	průchozí indukčnost
- L	[H]	průměrná hodnota magnetické indukčnosti
- L <sub>m</sub>	[H]	polovina amplitudy sinusově proměnné magnetické indukčnosti

# 1 ÚVOD

Modelování elektrických pohonů je velmi efektivní a často používanou metodou především pro návrh regulační části pohonu. Bez rizika zničení motoru nám takový model umožní navrhnout schéma regulační struktury a návrh optimálních parametrů regulátoru bez nutnosti složitých analytických výpočtů.

V této práci popisuji různé druhy elektrických pohonů a uvádím obecné postupy pro vytváření a real-time testování jejich modelů na platformě CompactRIO, včetně přípravných úloh popisujících možnosti tohoto procesu. Řídící aplikace pro testování je vytvořena v programu Labview, do kterého je implementován HDL kód modelu vygenerovaný z programu Matlab Simulink.

Vzhledem k relativně snadnému popisu a vzhledem k dostupnosti reálného motoru tohoto typu jsem pro svou práci zvolil synchronní motor s permanentními magnety (PMSM).

Zabývám se různými druhy použitých souřadných systémů pro popis PMSM motoru i různými druhy diskretizace. V programu Matlab Simulink testuji definování různých bitových rozsahů proměnných pomocí nastavení Fixed pointů a možnosti převodu modelu do HDL kódu.

Veškerá testování směřuji k nalezení ideálního kompromisu mezi přesností vytvořeného modelu a zatížením hradlového pole CompactRIA, aby výsledná aplikace věrně sledovala real-time chování skutečného motoru.

V závěrečné části práce otestuji real-time chování vytvořeného modelu na platformě CompactRIO přímo v paralelním zapojení s reálným motorem PMSM typu, uvedu výsledky tohoto měření a tím zhodnotím výsledky celé práce.

## 2 ZPŮSOB ŘEŠENÍ

V této kapitole je popsán postup vytvoření modelu elektrického pohonu pro platformu CompactRIO a veškeré HW a SW vybavení, které v této práci používám. Jednotlivé kroky jsou podrobněji popsány v dalších kapitolách.

Aby vytvořená aplikace s modelem motoru zvládla správně reagovat na změny vstupů a věrně tak simulovat chování skutečného motoru, musí pracovat na frekvenci v řádech Mhz. Program provádějící výpočty v RT části na frekvenci například 20kHz je pro tuto práci nevyhovující, proto využívám jednotku cRIO s integrovaným programovatelným hradlovým polem.

Programovatelná hradlová pole (FPGA) mají obrovský přínos ve velmi rychlých úlohách s uživatelsky nastaveným časováním, synchronizací a rozhodovací logikou při rychlostech dosahujících až 40 Mhz, proto je budu využívat i v této práci. [8]

Úlohy v FPGA dosahují obecně lepšího deterministického chování než programy v RT, nedochází například k přerušování programu úlohami s vyšší prioritou, úlohy v FPGA mohou pracovat paralelně. [8]

Program je implementován přímo v hardwarové struktuře, je tedy vynechán nadřazený řídicí systém, což ještě zvyšuje jeho spolehlivost a rychlost výpočtů.

V architektuře CompactRIA nejsou I/O moduly připojeny přes sběrnici, ale přímo k FPGA. Systém tedy vykazuje velmi malou latenci a umožňuje programům na RT kontroléru přístup k I/O s chybou nižší než 500 ns. [8]

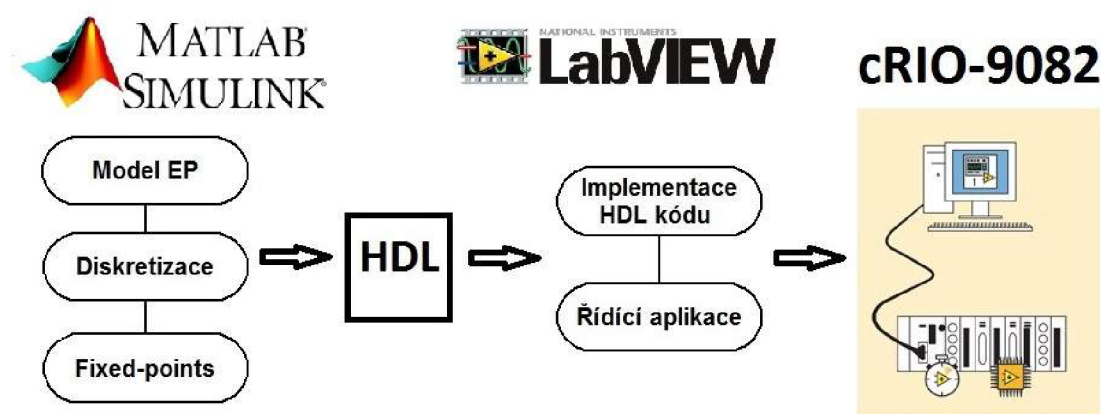
### 2.1 Logické schéma úlohy

Tvorbu modelu elektrického pohonu provádím ve dvou krocích v různých programech, v programu Matlab Simulink a v LabVIEW.

V prvním kroku v programu Matlab Simulink vytvořím matematický model konkrétního motoru včetně správných převodů mezi souřadnými systémy a včetně řízení pro tento motor, zároveň je zde potřeba tento model zdiskretizovat a také vhodně nadefinovat přesnosti a rozsahy jednotlivých proměnných pomocí Fixed pointů, aby následně v aplikaci nezabíraly příliš mnoho místa a celou simulaci nezpomalovaly. Celý takto vytvořený a zdiskretizovaný model elektrického motoru vygeneruji ze Simulinku v HDL formátu.

V druhém kroku do programu LabVIEW HDL kód naimportuji a vytvořím vhodnou aplikaci pro Real-time testování elektrického motoru na platformě CompactRIO. Tento projekt nahraji do programovatelného hradlového pole (FPGA) na jednotce cRIO. Pomocí vytvořené aplikace pak otestuji funkčnost modelu a vyzkouším si chování daného elektropohonu.

Výše popsané schéma této úlohy je znázorněno na následujícím obrázku:



Obr. 1 : Logické schéma úlohy

## 2.2 Použitý Hardware

### - Platforma CompactRIO - 9082

Jedná se o vysoce výkonné vícejádrové zařízení od firmy National Instruments, které umožňuje široké spektrum složitých výpočetních úkonů. Šasi obsahuje zabudovaný Real-time procesor, kontrolér, programovatelné hradlové pole (FPGA) a 8 slotů pro přídatné moduly. [6]

Platforma cRIO-9082 umožňuje používat operační systém Microsoft Windows Embedded Standart 7 (WES7), nebo jako v mém případě Phar Lap ETS. [6]

Součástí zařízení je FPGA typu Spartan-6. Do tohoto hradlového pole lze nahrávat programy v jazyce HDL. V mé práci využívám možnosti grafického programování pomocí LabVIEW. [6]



Obr. 2 : Fotografie platformy CompactRIO-9082 [6]

**- Využití slotů cRIA při testování s reálným motorem**

Slot 1	NI 9401	Generování PWM
Slot 2	NI 9401	Zpracování enkodéru
Slot 3	NI 9223	Kanál 0-2 pro měření statorových proudů
Slot 4	NI 9223	Kanál 0 pro měření momentu

Tab. 1: Využití slotů cRIA 9082 při testování s reálným motorem [6]

**- PMSM motor + enkodér**

Pro tuto práci používám AC synchronní bezkartáčový servomotor TGT od firmy TG drives s točivým výkonem 1 Nm a s výkonem až 850 W. [5]  
(TGT3-130-15-36/T4KX)

Snímání otáček a polohy motoru je prováděno malým hřídelovým inkrementálním optickým enkodérem typu ES 28.  
(ES 28-6-1024-05-D-R)

### **- Další HW použitý pro testování shody reálného motoru s jeho modelem**

Dále při testování shody modelu s reálným motorem využívám **Asynchronní dynamometr (ASD 2000-2-40)** od výrobce VUES, kterým na PMSM motoru simuluji zátěžný moment.

Většina částí testovacího zapojení jako enkodér, budiče tranzistorů měniče, měření proudů a tenzometrická hřídel jsou napájeny laboratorním zdrojem **DIAMETRAL P230R51D**.

Měnič PMSM je pak řízen napájecím napětím ze zdroje **KEYSIGHT E3634A** v režimu 25V/7A.

## **2.3 Použitý Software**

### **- Matlab R2015a - Simulink**

První část práce provádím ve výpočetním prostředí Matlab R2015a, jehož součástí je grafický editor pro blokové modelování Simulink. V tomto programu následně využívám generátor HDL kódu.

### **- LabVIEW 2014 – verze 14.0 (32-bit)**

Pro vytvoření řídicí aplikace a nahrání programu s modelem elektrického pohonu do FPGA na platformě CompactRIO používám program LabVIEW.

Nutnou součástí tohoto programu pro potřeby této práce jsou i některé integrovatelné moduly. Jedná se především o FPGA modul, Real-time modul nebo SoftMotion modul.

## 3 TEORETICKÝ ÚVOD

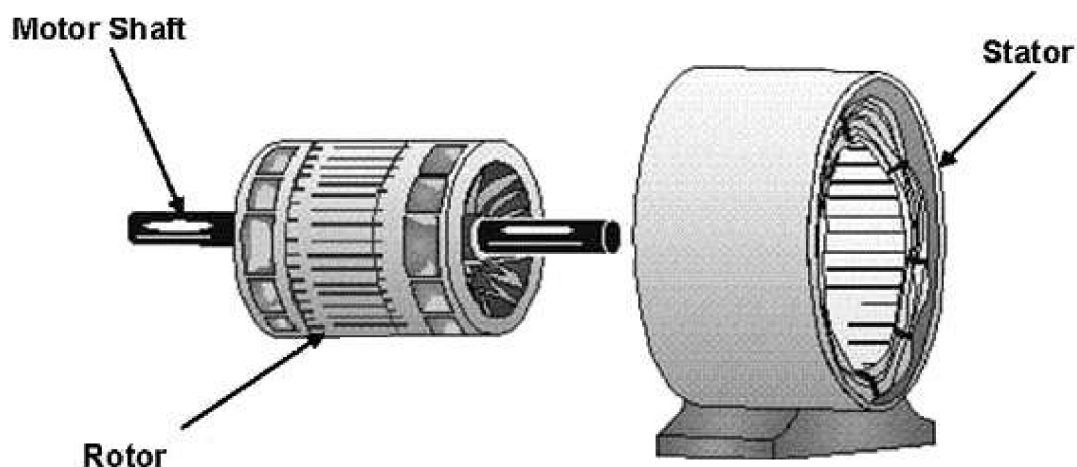
V této kapitole jsou teoreticky rozebrány elektrické pohony, jejich dělení a některé jejich konkrétní typy. Dále jsou zde vysvětleny souřadné systémy, kterými lze elektrické pohony popisovat a transformace, pomocí kterých lze souřadné systémy přepočítávat.

Teoreticky jsou zde rozebrány i typy diskretizace, které jsem si vybral pro zdiskretizování modelu elektrického motoru před jeho převodem do HDL kódu.

### 3.1 Elektrické motory a jejich třídění

Elektromotor je stroj, který převádí elektrickou energii na energii mechanickou, většinou vytváří rotační pohyb. Všechny rotační motory se skládají ze statoru a rotoru.

Na statoru jsou umístěny póly, na kterých je navinuto vinutí, jehož konce jsou vyvedeny na svorky motoru. Společným působením magnetického pole vzniklého průchodem proudu vinutím statoru a magnetického pole v rotoru (vzniklého pomocí rotorového vinutí nebo permanentním magnetem) je dosaženo rotačního pohybu motoru.

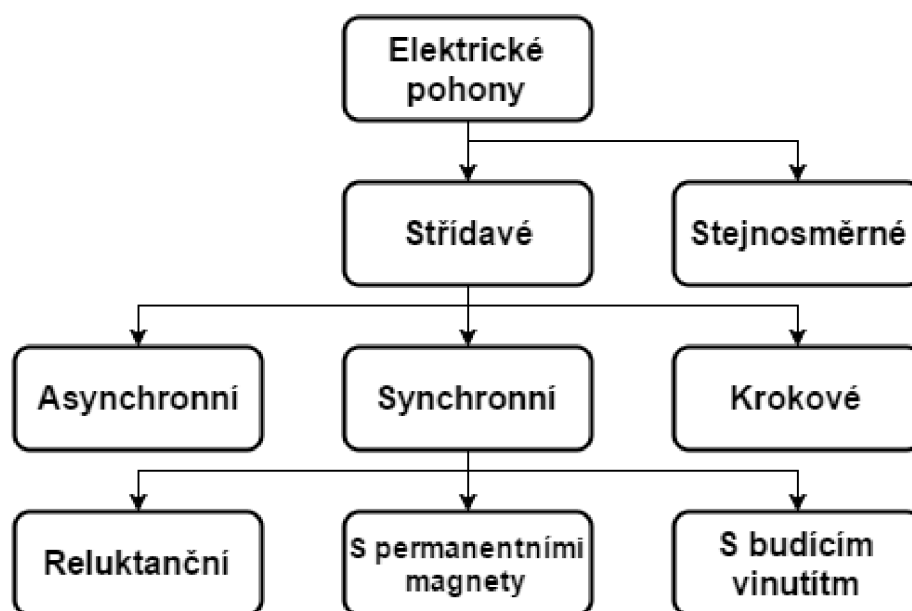


Obr. 3 : Elektromotor – stator, rotor [9]

Dělit elektrické pohony lze podle mnoha kritérií, v této práci potřebuji především rozdělení podle typu pohonu. Takto elektropohony třídíme na pohony se stejnosměrnými (DC) a střídavými (AC) motory podle typu napájecího proudu.

Zaměřím se především na modelování střídavých pohonů. Ty můžu dále rozdělit na asynchronní, synchronní a krokové. Rotor synchronního motoru se točí synchronními otáčkami, to znamená stejnými, jako jsou otáčky magnetického pole buzeného statorem.

U synchronních motorů zmíním ještě důležité rozdělení na motory reluktanční, motory s permanentními magnety a synchronní motory s budícím vinutím.



Obr. 4 : Dělení elektrických pohonů [2]



### 3.1.1 Asynchronní motor

Jedná se o nejrozšířenější typ elektrického motoru, je robustní, velmi spolehlivý a díky své jednoduché konstrukci také cenově přijatelný. Velkou výhodou je i nepřítomnost magnetického pole v odpojeném stavu. Díky všem těmto vlastnostem nachází široké uplatnění v nejrůznějších průmyslových aplikacích.

Mezi nevýhody tohoto motoru patří jeho skluz (způsobený rozdílnou rychlostí rotoru a magnetického pole), nepříliš velká účinnost a také poměrně složité řízení, proto se uplatňuje především u menších úloh.

Asynchronní motor má vinutí na statoru i rotoru. Statorové vinutí je připojeno na zdroj střídavého proudu, rotorové je nejčastěji spojeno nakrátko, proud v něm vzniká elektromagnetickou indukcí.

Rozložením tří fázových vinutí napájených třífázovým harmonickým napětím vzniká ve vzduchové mezeře stroje točivé magnetické pole. Díky rozdílným rychlostem točení rotoru a magnetického pole statoru se indukují ve vinutí na rotoru napětí uměrné rozdílu těchto rychlostí.

### 3.1.2 Synchronní motor

Synchronní motory jsou na rozdíl od asynchronních menší a lehčí, při stejném výkonu dosahují vyšší účinnosti, mají velký startovací moment a dají se snadněji chladit. Naopak nevýhodou je nutná znalost počáteční polohy motoru, pro řízení pohonu pak nutná neustálá znalost pozice rotoru. Magnetické pole působí u synchronního motoru s permanentními magnety i po jeho vypnutí.

Stator synchronního motoru je obvykle stejný jako u motoru asynchronního, to znamená, že je na něm střídavé trojfázové vinutí. Nejpodstatnějším rozdílem vůči asynchronnímu motoru je, že se rotor synchronního motoru točí současně s točivým magnetickým polem statoru.

Na rotoru je použita soustava pólů buzených stejnosměrným proudem nebo permanentní magnety. Další možnou konstrukcí rotoru je použití vhodně tvarovaných plechů, takto uspořádané motory se nazývají reluktanční.

## 3.2 Souřadné systémy, transformace

Elektrické pohony můžeme popisovat a modelovat pomocí různých souřadných systémů. Klasickým souřadným systémem je ten se souřadnicemi abc. Osy abc v tomto uspořádání odpovídají směrům třech statorových vinutí motoru.

Rovnice v souřadném systému abc jsou však výpočetně velmi náročné, proto se k popisu elektromotorů využívá Clarkové transformace, která tento trojosý systém převede na systém dvojosý. Tím snížíme počet rovnic popisujících motor a zároveň zkrátíme čas výpočtu, což nám pomůže lépe testovat model motoru v reálném čase.

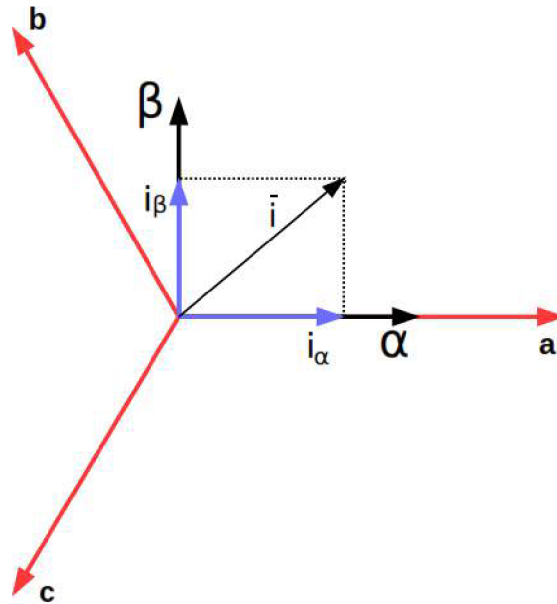
Pro převod mezi různými dvojosými transformacemi pootočenými navzájem o úhel  $\theta$  používáme Parkovu transformaci. Prostorové vektory lze obecně vyjádřit v různých komplexních rovinách rotujících vůči statoru úhlovou rychlostí  $\omega$ . Podle volby  $\omega$  rozlišujeme následující souřadné systémy:

Název souřadné soustavy	Označení os	Úhlová rychlost
Spojená se státorem	$\alpha, \beta$	0
Spojená s magnet. pólem	x, y	$\omega_s$
Spojená s rotorem	d, q	$\omega$
Rotující všeobecnou úhl. rychl.	u, v	$\omega_k$

Tab. 2: Souřadné systémy v různých komplexních rovinách [1]

### 3.2.1 Clarkové transformace

Clarkové transformace převede proud jednotlivými fázemi na jeden vektor v komplexní rovině, díky tomu se mi zjednoduší trojosý souřadnicový systém abc na systém, který lze popsat pomocí dvou os  $\alpha\beta$ . Tento převod je znázorněn na obr. 5.



Obr. 5 : Clarkové transformace [2]

Pro aplikování Clarkové transformace musí mít motor vinutí zapojená do hvězdy, pak platí pro proudy v jednotlivých vinutích tento vztah: [2]

$$i_a(t) + i_b(t) + i_c(t) = 0 \quad (3.1)$$

Po převedení této rovnice do podoby vektoru v komplexní rovině, kde jsou jednotlivé fáze navzájem posunuty o úhel  $\delta = 120^\circ$ , dostanu rovnici ve tvaru:

$$\bar{i} = i_a + i_b e^{j\delta} + i_c e^{2j\delta} \quad (3.2)$$

Ekvivalentně lze rovnici zapsat pomocí goniometrických funkcí, k převodu na tento zápis použijí Eulerův vzorec (3.3). Upravená rovnice pro vektor v komplexní rovině bude vypadat následovně (3.4):

$$e^{j\phi} = \cos \phi + j \sin \phi \quad (3.3)$$

$$\bar{i} = i_a + i_b(\cos \delta + j \sin \delta) + i_c(\cos 2\delta + j \sin 2\delta) \quad (3.4)$$

Tento vztah nyní můžeme jednoduše rozdělit na reálnou a imaginární část, čímž získáme hodnoty proudů v soustavě  $\alpha\beta$ . Hodnoty těchto proudů zapíšeme v maticové formě, která odpovídá obecnému maticovému vyjádření Clarkovy transformace. [2]

$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \begin{bmatrix} 1 & \cos \delta & \cos 2\delta \\ 0 & \sin \delta & \sin 2\delta \end{bmatrix} * \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} \quad (3.5)$$

Za úhel  $\delta$  ve vzorci dosadíme  $\delta = 120^\circ$ , tím získáme jednoduché převodní vztahy mezi souřadným systémem  $abc$  a systémem  $\alpha\beta$ . [2]

$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} * \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} \quad (3.6)$$

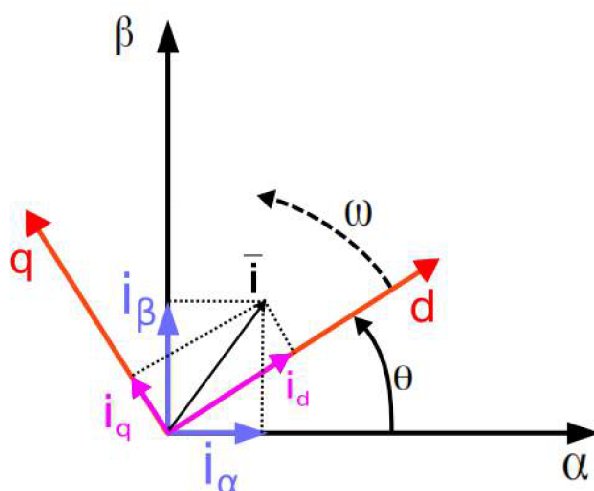
Obdobně lze pomocí Clarkovy zpětné transformace přepočítat systém se souřadnicemi  $\alpha\beta$  na tříosý systém souřadnic  $abc$  podle následujícího vztahu. [2]

$$\begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} * \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} \quad (3.7)$$

### 3.2.2 Parkova transformace

Místo dvouosého systému  $\alpha\beta$  spojeného se statorem, je často výhodnější použít souřadný systém  $dq$  spojený s rotorem, který je vůči systému  $\alpha\beta$  posunutý o úhel  $\theta$  ve směru, kterým se otáčí rotor rychlostí  $\omega$ .

K převodu mezi těmito dvěma soustavami se používá Parkova transformace. Slouží ke zjednodušení popisu dějů v elektrických pohonech například při vektorovém řízení.



Obr. 6 : Parkova transformace [2]

Komplexní vektor proudu můžeme v osách  $dq$  vyjádřit takto: [2]

$$\bar{i} = i_d + j i_q = i e^{-j\theta} \quad (3.8)$$

Úhel  $\theta$  nám ve vztahu udává aktuální natočení systému  $dq$  proti satorovým osám  $\alpha\beta$ . Dosazením Eulerova vzorce pak tuto rovnici převedeme do následujícího tvaru:

$$\bar{i} = i_d + j i_q = (i_\alpha + j i_\beta) (\cos \theta - j \sin \theta) \quad (3.9)$$

Parkovu transformaci pak můžeme vyjádřit v maticovém tvaru jako: [2]

$$\begin{bmatrix} i_d \\ i_q \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} \quad (3.10)$$

Obdobně se lze pomocí inverzní Parkovi transformace přenést ze systému dq do systému s osami  $\alpha\beta$ : [2]

$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} \quad (3.11)$$

### 3.3 Diskretizace systému

Pomocí matematických modelů daných elektrických pohonů dosáhneme jejich spojité simulace. Abych však mohl elektromotor simulovat pomocí hradlového pole na platformě CompactRIO, musím tento spojitý model nejdříve převést na model diskrétní.

Před diskretizací je nejprve nutné si všechny průběhy správně navzorkovat a zvolit vyhovující vzorkovací periodu, která nebude příliš velká, aby nám spojitý průběhy příliš nezkrusila, a zároveň nebude příliš malá, aby zbytečně nezvyšovala nároky na bitové rozsahy mezivýpočtů, které by mohly vést až k zaplnění celého hradlového pole.

Vzorec pro diskretizaci spojitého modelu můžeme najít pomocí Taylorova rozvoje funkce  $e^{AT}$ . [10]

$$A_k = e^{AT} = I + AT + \frac{(AT)^2}{2} + \frac{(AT)^3}{6} + \dots \quad (3.12)$$

$$B_k = A^{-1}(e^{AT} - I)B \quad (3.13)$$

Nejjednodušším způsobem jak model zdiskretizovat je metoda Eulerovy aproximace, nazývaná též metoda přímé diference. Jedná se o metodu lineární, při které z Taylorovy řady vezmu pouze polynom 1.řádu. Vztahy pro matice diskrétního modelu  $A_k$ ,  $B_k$ ,  $C_k$  tedy pomocí matic spojitého modelu  $A$ ,  $B$ ,  $C$  a periody vzorkování  $T$  mohou vyjádřit takto: [10]

$$A_k = I + AT \quad (3.14)$$

$$B_k = A^{-1}(I + AT - I)B = TB \quad (3.15)$$

$$C_k = C \quad (3.16)$$

Další možností je použití kvadratické metody, to znamená použít z Taylorovy řady polynom druhého řádu. Vztahy pro diskrétní matice  $A_k$ ,  $B_k$ ,  $C_k$ , pak vypadají následovně: [10]

$$A_k = I + AT + \frac{(AT)^2}{2} \quad (3.17)$$

$$B_k = A^{-1}\left(I + AT + \frac{(AT)^2}{2} - I\right)B = TB + \frac{AT^2}{2}B \quad (3.18)$$

$$C_k = C \quad (3.19)$$

Metody ještě vyšších řádů už vytváří příliš složité a rozsáhlé matice. Velký počet matematických operací v modelu by potom příliš zatížil jednotku CompactRIO, velikost hradlového pole by už pak nemusela aplikaci dostačovat.

## 4 PŘÍPRAVNÉ ÚLOHY

Popsané ukázkové úlohy slouží k postupnému seznámení s hardwarovým a softwarovým vybavením a také s postupy potřebnými k dosažení cílů mé diplomové práce, tedy k vytvoření funkčních modelů elektrického pohonu v FPGA na platformě CompactRIO.

### 4.1 Sčítačka dvou desetinných čísel

#### 4.1.1 Cíl úlohy

V této úloze vytvářím v programu Matlab Simulink jednoduché sčítání dvou vstupů a zobrazení výsledku tohoto sčítání. Výsledné schéma z tohoto programu vygeneruji do HDL kódu a v programu Labview vytvořím aplikaci této úlohy, kterou nahraji a otestuji pomocí CompactRIA.

Úloha slouží především k odzkoušení celého pracovního postupu, který budu používat i při modelování elektrického pohonu. Tedy také k otestování správné funkčnosti celého softwarového a hardwarového vybavení.

Při rozsáhlejších aplikacích, jakými testování modelů elektrických pohonů může být, si musíme dát pozor na velikost programu, který chceme do hradlového pole nahrávat. FPGA má omezený počet bloků pro násobení a další operace, který použitou aplikací nesmíme překročit.

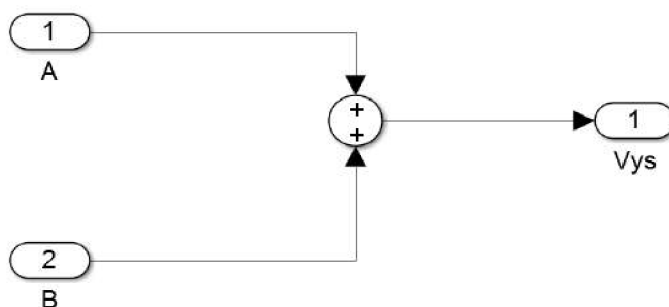
Počty potřebných bloků se výrazně zvyšují se zvyšujícími se nároky na přesnost modelu a jeho mezivýpočtů. Z Matlabu generovaný HDL kód musí mít přesnosti, pomocí bitových rozsahů, celé a desetinné části čísla pro všechny tyto výpočty pevně definovány pomocí Fixed pointů.

Touto ukázkovou úlohou si vyzkouším nastavování těchto Fixed pointů v Simulinku i jejich správné nadefinování v LabVIEW aplikaci.



## 4.1.2 Realizace – Simulink

Aby se model správně převedl do HDL kódu a dal se následně použít v Labview, musím pro vstupy i výstup v Simulinku použít bloky pro inport a outport proměnné. Dále použiji klasický blok pro sčítání a celé schéma propojím tak, jak je ukázáno na obrázku.

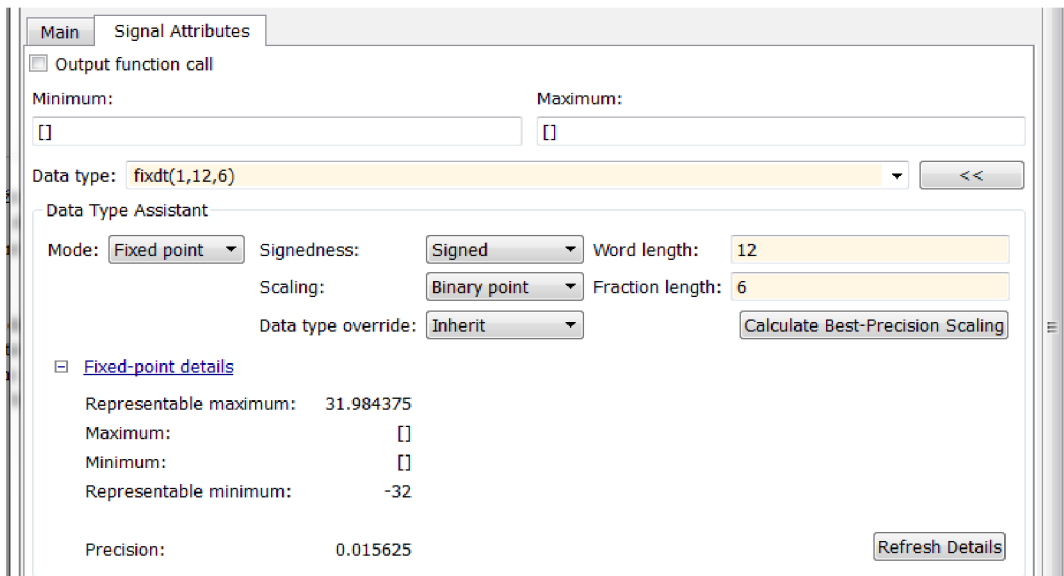


Obr. 7 : Schéma sčítačky - Simulink

Nastavení Fixed pointů najdu po rozkliknutí jednotlivých bloků v záložce „Signal Attributes“. Po nastavení datového typu jako „fixdt“ mohu zadat binární délku slova, přidat nebo odebrat znaménkový bit a poté rozdělit desetinnou část a část pro celé číslo.

Rozdělení slova na desetinnou a celou část mohu provést pevně zadaným počtem bitů v jednotlivých částech nebo pomocí zadání hodnot „Slope“ pro rozlišení slova a „Bias“ pro posunutí nulové hodnoty tohoto slova. Tento druhý způsob zápisu ale nepodporuje HDL kódér programu Simulink, tedy ho nemůžu použít. Použité proměnné tedy mohou být buď bez znaménkového bitu nebo musí mít rozsah nastavený symetricky kolem nulové hodnoty.

Celé nastavení může vypadat například tak, jak je vidět z následujícího obrázku, kde jsem zadal 12 bitové slovo s 6 bitovou desetinnou částí a se znaménkovým bitem.

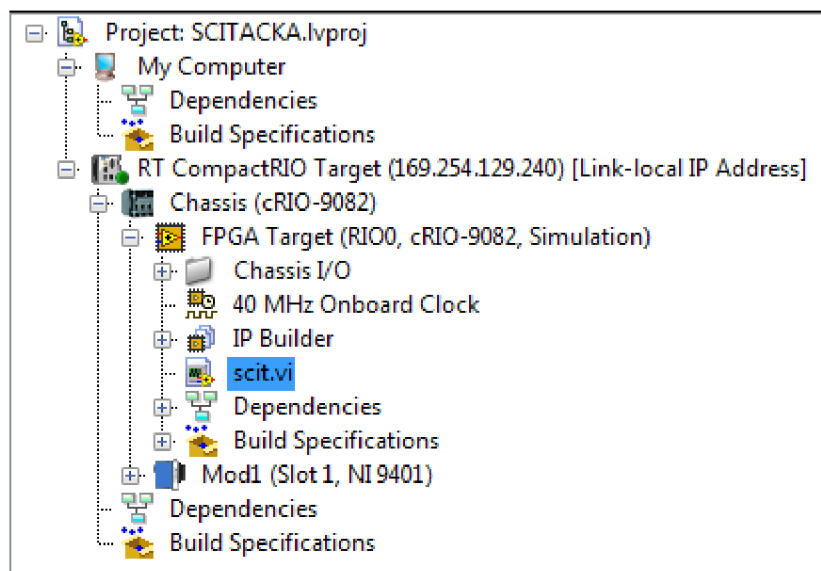


Obr. 8 : Definování délky slov pomocí Fixed pointů

Z takto upraveného schématu vygeneruji HDL soubor, který budu následně implementovat do Labview projektu pro CompactRIO.

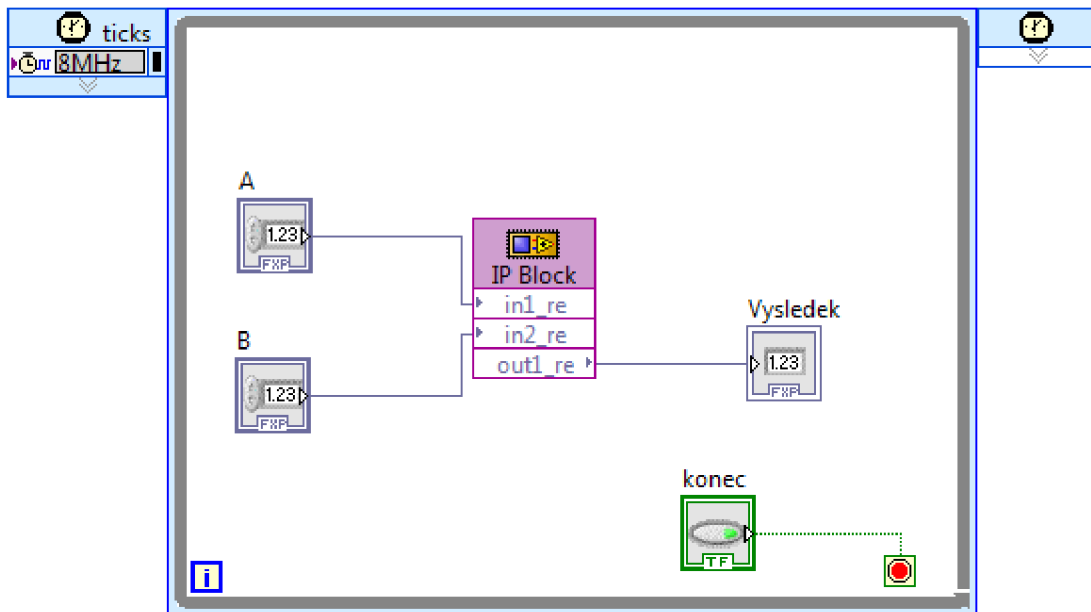
### 4.1.3. Realizace – Labview

V programu Labview jsem založil projekt obsahující konkrétní CompactRIO, do FPGA na této jednotce jsem přidal soubor typu VI.



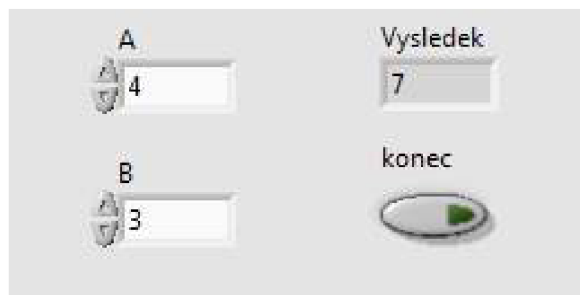
Obr. 9 : Sčítačka – strom projektu v Labview

V Block diagramu tohoto VI souboru přidám „IP\_Block“, který mi do schématu umožní implementovat vytvořený HDL kód ze Simulinku a ve kterém správně nastavím rozsahy všech vstupů a výstupů tohoto bloku. Celý blokový diagram musí být umístěn uvnitř časové smyčky, pro jejíž ukončení používám tlačítko „konec“. Blokový diagram sčítačky je vidět na obrázku č. 10.



Obr. 10 : Sčítačka – Block diagram

Součástí VI souboru je pak také definování Front panelu, pomocí kterého je aplikace ovládána. Sčítačku jsem pomocí tohoto rozhraní úspěšně otestoval v simulačním módu i přímo nahranou v hradlovém poli CompactRIA.



Obr. 11 : Sčítačka – spuštěná aplikace

## 4.2 Implementace funkce sinus

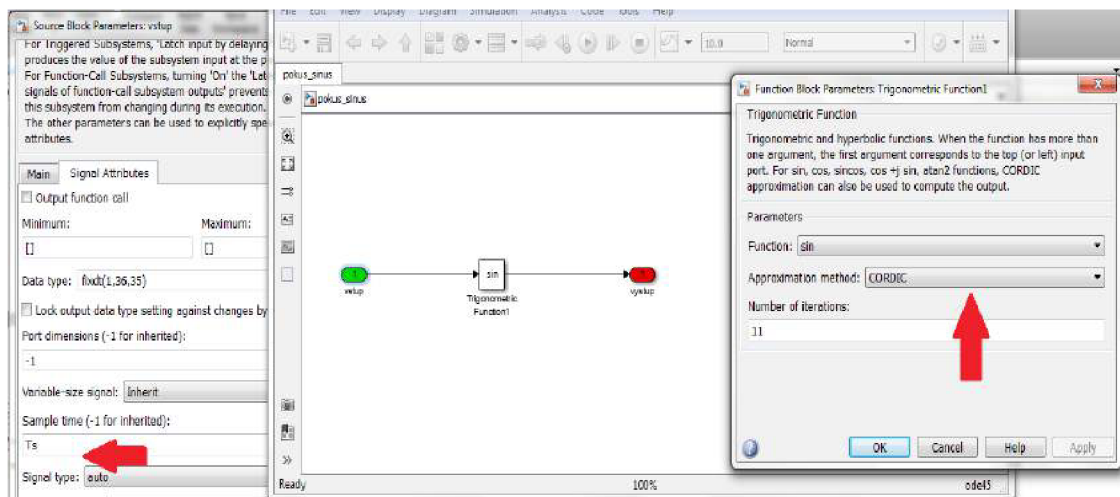
### 4.2.1 Cíl úlohy

Cílem druhé úlohy je vytvořit model s trigonometrickou funkcí, jeho převod do HDL kódu a následná implementace do hradlového pole na platformě CompactRIO.

Úlohou si chci také porovnat možnosti převodu trigonometrických funkcí přes HDL kód ze Simulinku s vytvářením těchto funkcí přímo v Labview. Výsledky mi pak poslouží při práci s modelem motoru a při transformacích mezi souřadnými systémy pohonu.

### 4.2.2 Realizace - Simulink

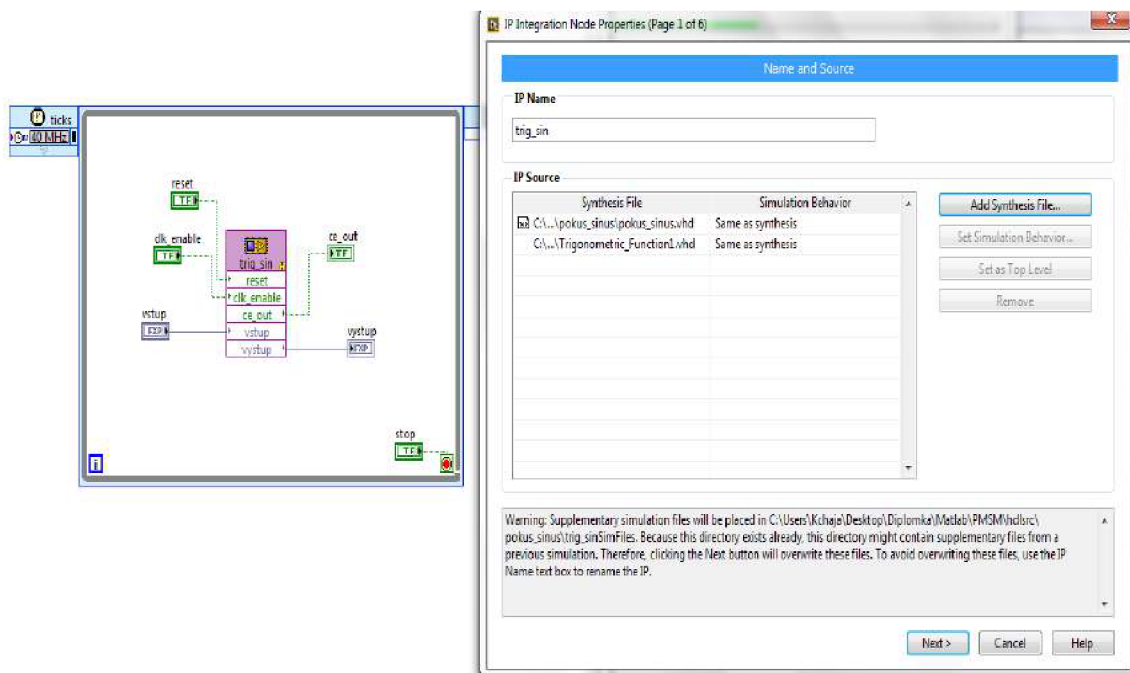
Pro převod sinu nebo cosinu do HDL se v Matlabu používá aproximační metoda CORDIC, kterou můžeme nastavit uvnitř bločku trigonometrické funkce. Vstup do tohoto bločku musí být navzorkovaný, v tomto případě vzorkovací periodou  $T_s$ .



Obr. 12 : Trigonometrická funkce - Matlab

## 4.2.3 Realizace – Labview

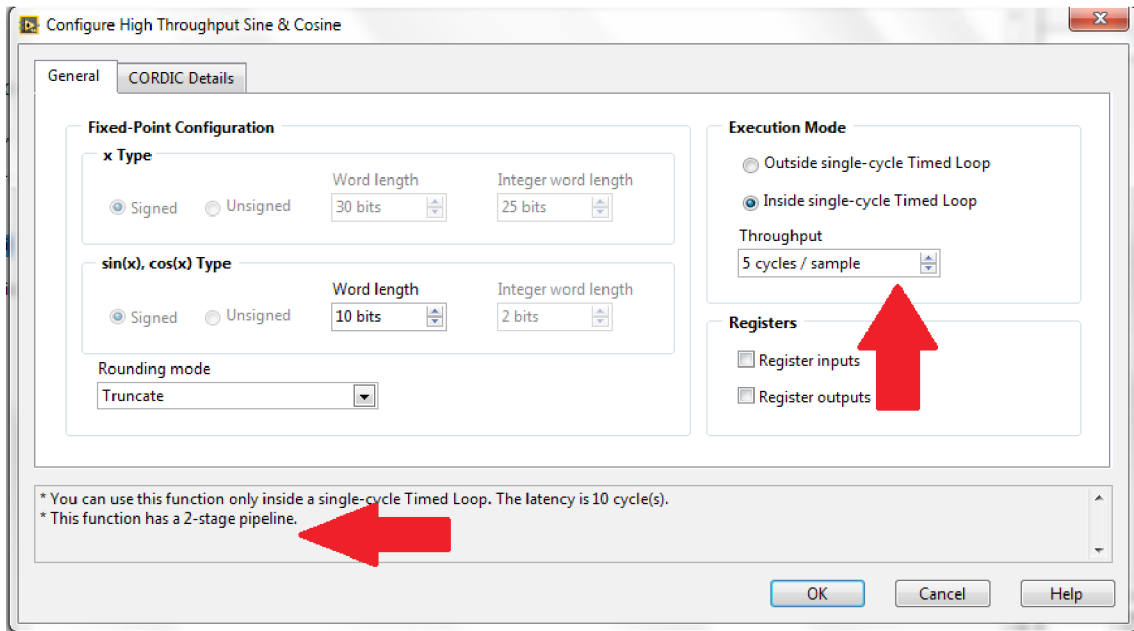
V programu Labview vytvořím soubor VI v hradlovém poli FPGA, do časové smyčky v něm umístím IP Node pro implementaci HDL kódu funkce sinus. Mimo hlavní soubor musím přidat i všechny HDL soubory pro jednotlivé trigonometrické funkce, pro každou z nich kodér v Matlabu automaticky vytvoří vlastní soubor. Musím také pohlídat, aby byl v nastavení IP Nodu správně nastaven “Top level“ k hlavnímu souboru.



Obr. 13 : Implementace funkce sinus - Labview

V pomocném HDL souboru pro trigonometrickou funkci se nachází výpočet aproximační metody CORDIC. Jedná se o funkci iterační, v tomto případě nastavenou na 11 iterací, program tedy musí vytvořit 11 pipeline, pokud má stíhat vypočítávat tuto funkci v každém cyklu. Tento způsob velmi zvyšuje náročnost výpočtu a není proto úplně vhodný.

Další možností je použít funkci sinus přímo z knihovny funkcí programu Labview. Výhodou tohoto způsobu je možnost nastavení si počtu cyklů, v kterých se provede výpočet trigonometrické funkce. Lze tedy nastavit rozumný kompromis mezi počtem využitých pipeline a počtem cyklů výpočtu.



Obr. 14 : Nastavení trigonometrické funkce - Labview

Porovnáním těchto způsobů zjistíme, že při použití 11 pipeline může být zatížení hradlového pole až dvojnásobné proti funkci sinus bez jejich použití. Při práci s více trigonometrickými funkcemi při vyšších přesnostech už tedy může být použití těchto automaticky vygenerovaných funkcí do HDL kódu problematické.

<div data-bbox="357 1476 799 1720"> <p><b>Device Utilization</b></p> <p>-----</p> <p>Total Slices: 2,5% (587 out of 23038)            Slice Registers: 0,9% (1568 out of 184304)            Slice LUTs: 1,8% (1649 out of 92152)            Block RAMs: 0,0% (0 out of 268)            DSP48s: 0,0% (0 out of 180)</p> </div> <p style="text-align: center;"><b>Sinus - Matlab</b> 11x pipeline, 1x cycle</p>	<div data-bbox="884 1476 1315 1720"> <p><b>Device Utilization</b></p> <p>-----</p> <p>Total Slices: 1,4% (317 out of 23038)            Slice Registers: 0,4% (680 out of 184304)            Slice LUTs: 0,7% (644 out of 92152)            Block RAMs: 0,0% (0 out of 268)            DSP48s: 0,0% (0 out of 180)</p> </div> <p style="text-align: center;"><b>Sinus - Labview</b> 1x pipeline, 11x cycle</p>
---	---

Obr. 15 : Porovnání výpočetní náročnosti trig. funkcí s a bez využití pipeline

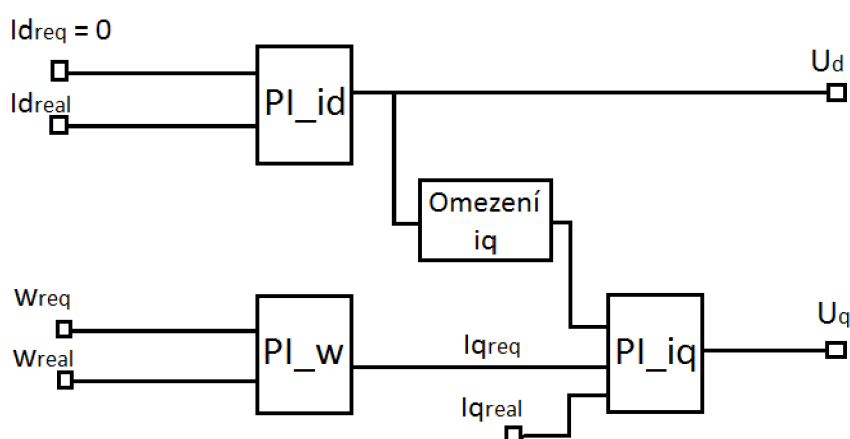
## 5 VEKTOROVÉ ŘÍZENÍ

K ověření správné funkčnosti vytvořených modelů a jejich porovnání s reálným motorem potřebuji vhodné řízení. V této práci používám vektorové řízení s orientací na rotorový tok, které je vhodné pro dynamicky náročnější aplikace, jakými jsou i servopohony.

Vektorové řízení je založeno na rozdělení statorového proudu do dvou složek. U synchronního motoru s permanentními magnety je výhodnější provádět toto řízení v transformovaných souřadnicích dq, v těchto osách pak řídíme proudy na konstantní hodnoty

Statorový proud v dq souřadnicích spojených s natočením rotoru je tedy rozdělen do dvou navzájem kolmých složek. Složka proudu v d-ose ovlivňuje magnetizaci pohonu, tedy ovlivňuje velikost výsledného magnetického toku. Na ni kolmá složka v q-ose vytváří moment a činný výkon motoru. Tyto složky se navzájem neovlivňují a tak je možné řídit zvlášť moment a magnetický tok. Pro každý pracovní bod je nutné vypočítávat hodnoty těchto složek proudu a určit pomocí nich řídicí veličiny pro nastavení jednotlivých napětí statoru.

Strukturu tohoto vektorového řízení tvoří dvě základní proudové smyčky a nadřazená rychlostní smyčka. V první proudové smyčce budu udržovat hodnotu proudu  $i_d$  na nulové hodnotě. Fázor statorového proudu tedy bude ovlivňován především proudem  $i_q$ . Podle regulátoru  $PI_{i_d}$  omezují hodnoty druhého regulátoru  $PI_{i_q}$ , regulátor proudu v d-ose má tedy vyšší prioritu.

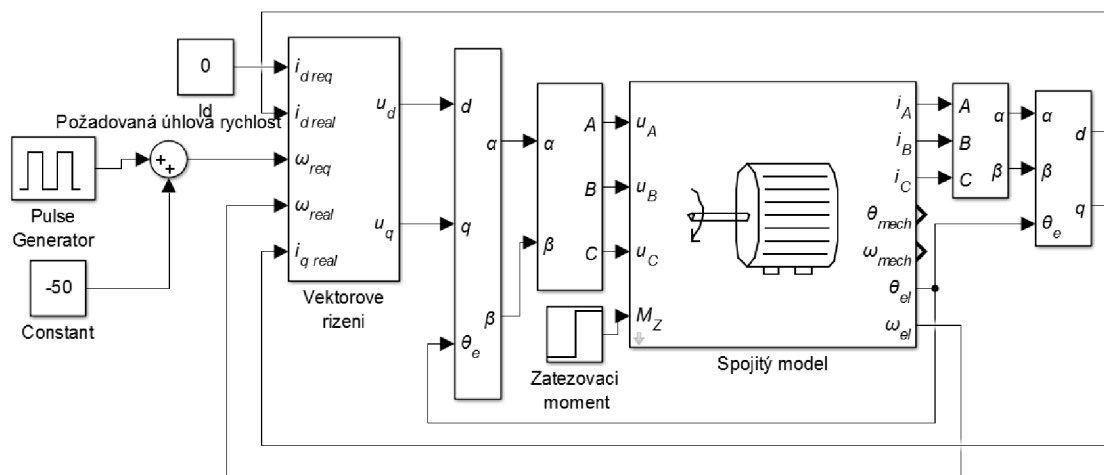


Obr. 16 : Schéma vektorového řízení

S ohledem na maximální povolený proud tekoucí synchronním motorem musíme dbát ještě na proudové omezení. V závislosti na velikosti proudu  $i_d$  je potřeba hlídat velikost proudu  $i_q$ , aby celkový proud nepřesáhl povolené maximum. Obdobně můžeme ošetřit i napěťové omezení.

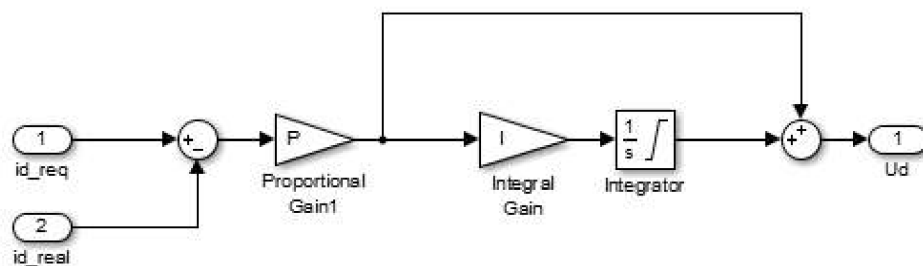
$$i_{qMAX} = \sqrt{i_{MAX}^2 - i_d^2} \quad (5.1)$$

Výsledný regulační obvod spojitého modelu motoru s popsáním vektorovým řízením pak v Simulinku včetně transformací souřadnic vypadá následovně:



Obr. 17 : Regulační schéma spojitého modelu s VC a transformacemi

Všechny tři regulátory umístěné ve vektorovém řízení jsou klasické PI regulátory s proporcionální a integrační složkou.



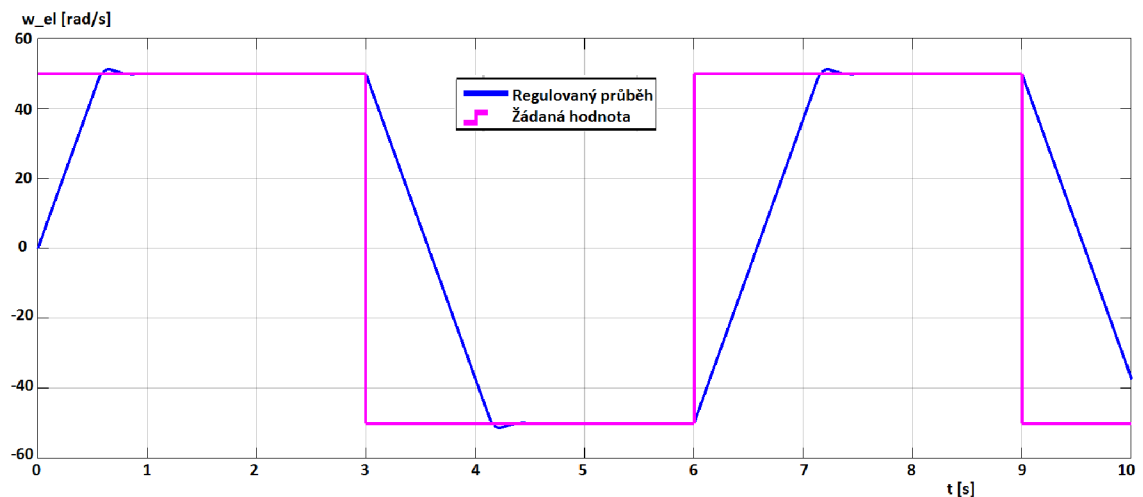
Obr. 18 : Vektorové řízení - PI regulátor



U všech testování modelů v programu Matlab Simulink používám pro žádanou hodnotu elektrických otáček obdélkový signál o amplitudě 50 rad/s a periodě 6 sekund, to znamená, že každé tři sekundy dochází ke skokové změně žádané hodnoty elektrických otáček.

Regulátory proudu ponechávám s nastavením  $P = 3$  a  $I = 1375$ , regulátor otáček pak má zesílení v proporcionalní složce  $P = 3$  a v integrační složce  $I = 15$ . Vektorový součet obou proudů je saturován na hodnotu 6A, kterou by neměly překročit proudy v reálném motoru.

Výsledný regulovaný průběh otáček PMSM motoru pak může vypadat jako na grafu 1, rychlejší odezva na změnu žádané hodnoty je znemožněna volbou maximální hodnoty výstupních napětí regulátorů.



Graf 1: Regulovaný průběh otáček sledující skokové změny žádané hodnoty

## 6 MATEMATICKÝ MODEL PMSM MOTORU

Vzhledem k možnosti porovnání pohonu s reálným motorem a vzhledem k možnému jednoduchému matematickému popisu tohoto stroje jsem pro svou práci vybral synchronní motor s permanentními magnety (PMSM).

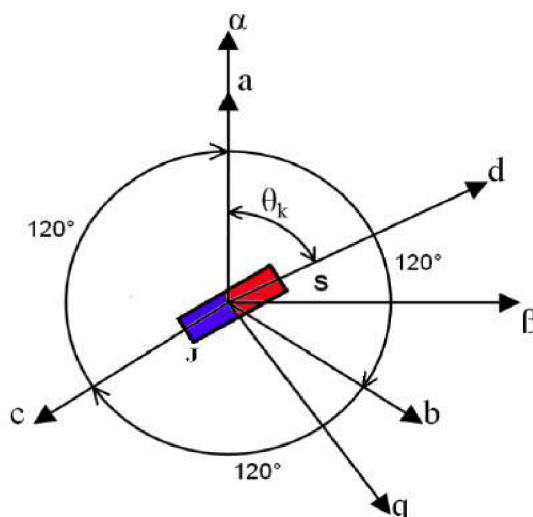
Synchronní motor s permanentními magnety nazývaný také jako bezkartáčový servopohon je jedním z nejvýznamnějších střídavých regulačních pohonů s výkonem řádově až několika stovek kilowattů, i když nejčastěji se využívají výkony do několika kW.

Pohon má běžný třífázový stator s vinutím zapojeným do hvězdy, který je napájen harmonickými proudy. Rotor je tvořen permanentními magnety především ze vzácných zemin. Magnetická indukce ve vzduchové mezeře a tedy i indukované napětí mají harmonický, sinusový tvar.

### Zjednodušující předpoklady pro matematický model: [1]

- Magnetická indukce ve vzduchové mezeře a tedy i indukované napětí mají sinusový průběh.
- Parametry stroje (R,L) jsou konstantní a stejné ve všech třech fázích stroje. (mimo obecný model v abc souřadném systému)
- Zanedbávají se ztráty v železe, tlumící vinutí na rotoru není provedeno.
- Nulový vodič není připojen.

Nejvýhodnějším pro matematický popis je dq systém spojený s rotorem. V práci se pak zabývám i popisem přímo v abc systému.



Obr. 19 : Zobrazení souřadnicových systémů [2]

## 6.1 Spojité modely PMSM

Ze základních napěťových rovnic PMSM motoru v této kapitole sestavím stavové rovnice pro matematické modely motoru v abc i v dq souřadnicích.

Pro lepší napodobení skutečného motoru zakomponuji do rovnic i konstantu tlumení motoru  $K_{TL}$  a zátěžný moment  $M_z$ . Kde je to z hlediska složitosti výpočtu možné, snažím se ponechávat indukčnosti modelu obecné nesymetrické pro různá testování nesymetričností motoru.

### 6.1.1 Model PMSM motoru v abc souřadnicích

Model motoru v abc souřadném systému je podstatně složitější než model v dq systému, navíc se v jeho napěťových rovnicích vyskytují trigonometrické funkce. Na rozdíl od modelu v dq systému však může sloužit například k testování zkratu na některém z vinutí nebo k jiným nesymetričnostem motoru. Navíc při jeho výpočtech není zapotřebí používat žádné transformace systému.

Na základě definice PMSM motoru je dána v souřadnicích abc tato napěťová rovnice: [11]

$$u_{abc} = R_s i_{abc} + \frac{d}{dt} \psi_{abc} \quad (6.1)$$

Kde magnetické spřažené toky  $\psi_{abc}$  jsou navzájem posunuté o  $120^\circ$  a můžeme je vyjádřit tímto způsobem: [11]

$$\psi_{abc} = L_s i_{abc} + \psi_f \begin{bmatrix} \sin \theta \\ \sin(\theta - \frac{2}{3}\pi) \\ \sin(\theta + \frac{2}{3}\pi) \end{bmatrix} \quad (6.2)$$

Celý model v abc souřadnicích nám zkomplikuje matice indukčnosti  $L_s$ . Jedná se o symetrickou matici, kde na hlavní diagonále leží vlastní indukčnosti v jednotlivých osách a mimo diagonálu vzájemné indukčnosti mezi jednotlivými osami. [11]

$$L_s = \begin{bmatrix} L_{aa} & L_{ab} & L_{ac} \\ L_{ab} & L_{bb} & L_{bc} \\ L_{ac} & L_{bc} & L_{cc} \end{bmatrix} \quad (6.3)$$

Vlastní indukčnosti můžeme vyjádřit jako součet průchozí indukčnosti  $L_{ls}$ , která je konstantní, a magnetizační indukčnosti  $L_m$ , kterou naopak jako konstantní brát nemůžeme, protože výrazně závisí na úhlu natočení rotoru. [11]

$$L_{aa} = L_{ls} + L_m = L_{ls} + L - L_{\Delta} \cos 2\theta \quad (6.4)$$

Kde  $L$  je průměrná hodnota magnetizační indukčnosti a  $L_{\Delta}$  je polovina amplitudy sinusově proměnné magnetizační indukčnosti.

Obdobně budou zapsány i vlastní indukčnosti  $L_{bb}$  a  $L_{cc}$ , které budou opět pouze posunuté o úhel  $120^\circ$ : [11]

$$L_{bb} = L_{ls} + L_m = L_{ls} + L - L_{\Delta} \cos\left(2\left(\theta - \frac{2\pi}{3}\right)\right) = L_{ls} + L - L_{\Delta} \cos\left(2\theta + \frac{2\pi}{3}\right) \quad (6.5)$$

$$L_{cc} = L_{ls} + L_m = L_{ls} + L - L_{\Delta} \cos\left(2\left(\theta + \frac{2\pi}{3}\right)\right) = L_{ls} + L - L_{\Delta} \cos\left(2\theta - \frac{2\pi}{3}\right) \quad (6.6)$$

Podobně jsou vyjádřeny i vzájemné indukčnosti: [11]

$$L_{ab} = -\frac{L}{2} - L_{\Delta} \cos\left(2\theta - \frac{2\pi}{3}\right) \quad (6.7)$$

$$L_{ac} = -\frac{L}{2} - L_{\Delta} \cos\left(2\theta + \frac{2\pi}{3}\right) \quad (6.8)$$

$$L_{bc} = -\frac{L}{2} - L_{\Delta} \cos(2\theta) \quad (6.9)$$

Celou matici indukčností  $L_s$  závislou na úhlu natočení  $\theta$ , která se v napěťové rovnici motoru bude roznásobovat s maticí proudů, zapíšeme následovně: [11]

$$L_s = \begin{bmatrix} L_{ls} + L - L_{\Delta} \cos 2\theta & -\frac{L}{2} - L_{\Delta} \cos(2\theta - \frac{2\pi}{3}) & -\frac{L}{2} - L_{\Delta} \cos(2\theta + \frac{2\pi}{3}) \\ -\frac{L}{2} - L_{\Delta} \cos(2\theta - \frac{2\pi}{3}) & L_{ls} + L - L_{\Delta} \cos(2\theta + \frac{2\pi}{3}) & -\frac{L}{2} - L_{\Delta} \cos(2\theta) \\ -\frac{L}{2} - L_{\Delta} \cos(2\theta + \frac{2\pi}{3}) & -\frac{L}{2} - L_{\Delta} \cos(2\theta) & L_{ls} + L - L_{\Delta} \cos(2\theta - \frac{2\pi}{3}) \end{bmatrix} \quad (6.10)$$

Pro zjednodušení situace budeme nyní uvažovat symetrické rozložení indukčností ve všech třech osách, tedy  $L_a = L_b = L_c$ . Z této rovnosti vyplývá, že  $L_{\Delta}$  se rovná nule, tedy matice indukčností už nebude závislá na úhlu natočení.

$$L_s = \begin{bmatrix} L_{ls} + L & -\frac{L}{2} & -\frac{L}{2} \\ -\frac{L}{2} & L_{ls} + L & -\frac{L}{2} \\ -\frac{L}{2} & -\frac{L}{2} & L_{ls} + L \end{bmatrix} \quad (6.11)$$

Dosazením do výše uvedených vztahů tedy získám zjednodušené napěťové rovnice pro symetrické rozložení indukčností:

$$u_a = R_s i_a + \frac{d}{dt} [(L_{ls} + L) i_a - \frac{L}{2} (i_b + i_c) + \psi_f \sin \theta] \quad (6.12)$$

$$u_b = R_s i_b + \frac{d}{dt} [(L_{ls} + L) i_b - \frac{L}{2} (i_a + i_c) + \psi_f \sin(\theta - \frac{2}{3}\pi)] \quad (6.13)$$

$$u_c = R_s i_c + \frac{d}{dt} [(L_{ls} + L) i_c - \frac{L}{2} (i_a + i_b) + \psi_f \sin(\theta + \frac{2}{3}\pi)] \quad (6.14)$$

Rovnici pro čtvrtou stavovou proměnnou, úhlovou rychlost  $\omega$ , včetně započítaného zátěžného momentu  $M_Z$  a tlumení motoru  $K_{TL}$ , odvodím z pohybové rovnice jako: [1]

$$\frac{d}{dt} \omega = \frac{P_p}{J} (M_E - M_Z - K_{TL} \frac{\omega}{P_p}) \quad (6.15)$$

Do této rovnice dosadím vztah pro elektromagnetický moment stroje  $M_E$ , který ještě musím upravit pro souřadný systém s osami abc. [11]

$$M_E = \frac{3}{2} P_p \psi_f I_s \quad (6.16)$$

$$M_E = \frac{3}{2} P_p \psi_f \left[ \frac{\sqrt{3}}{2} (i_b - i_c) \cos \theta - \left( i_a - \frac{1}{2} i_b - \frac{1}{2} i_c \right) \sin \theta \right] \quad (6.17)$$

Získávám tedy stavové rovnice pro úhlovou rychlost  $\omega$  a také pro aktuální polohu motoru  $\theta$ :

$$\frac{d\omega}{dt} = \frac{P_p}{J} \left[ \frac{3}{2} P_p \psi_f \left( \frac{\sqrt{3}}{2} (i_b - i_c) \cos \theta - \left( i_a - \frac{1}{2} i_b - \frac{1}{2} i_c \right) \sin \theta \right) - M_z - K_{TL} \frac{\omega}{P_p} \right] \quad (6.18)$$

$$\frac{d\theta}{dt} = \omega \quad (6.19)$$

Z napěťových rovnic 6.12-14 bych teď chtěl získat stavové rovnice pro proudy ve všech třech osách. Úpravou těchto rovnic získám následující vztahy:

$$u_a = R_s i_a + \frac{d i_a}{dt} (L_{ls} + L) - \frac{L}{2} \left( \frac{d i_b}{dt} + \frac{d i_c}{dt} \right) + \psi_f \omega \cos \theta \quad (6.20)$$

$$u_b = R_s i_b + \frac{d i_b}{dt} (L_{ls} + L) - \frac{L}{2} \left( \frac{d i_a}{dt} + \frac{d i_c}{dt} \right) + \psi_f \omega \cos \left( \theta - \frac{2}{3} \pi \right) \quad (6.21)$$

$$u_c = R_s i_c + \frac{d i_c}{dt} (L_{ls} + L) - \frac{L}{2} \left( \frac{d i_a}{dt} + \frac{d i_b}{dt} \right) + \psi_f \omega \cos \left( \theta + \frac{2}{3} \pi \right) \quad (6.22)$$

Je však patrné, že po vyjádření proudových stavových rovnic zůstane ve vztazích závislost na derivacích zbylých proudů, což při vytváření modelu způsobí neřešitelné algebraické smyčky.

Budu tedy muset při výpočtu stavových rovnic opět vyjít z původní napěťové rovnice a celý výpočet provést v maticové formě:

$$u_{abc} = R_s i_{abc} + \frac{d}{dt} \left( L_s i_{abc} + \psi_f \begin{bmatrix} \sin \theta \\ \sin(\theta - \frac{2}{3} \pi) \\ \sin(\theta + \frac{2}{3} \pi) \end{bmatrix} \right) \quad (6.23)$$

Z této rovnice si nyní vyjádřím derivace stavových proměnných, tedy proudů  $i_{abc}$ :

$$u_{abc} = R_s i_{abc} + \frac{d}{dt} L_s i_{abc} + \frac{d}{dt} \left( \psi_f \begin{bmatrix} \sin \theta \\ \sin(\theta - \frac{2}{3} \pi) \\ \sin(\theta + \frac{2}{3} \pi) \end{bmatrix} \right) \quad (6.24)$$

$$\frac{d}{dt} i_{abc} = L_s^{-1} \left( u_{abc} - R_s i_{abc} - \frac{d}{dt} \left( \psi_f \begin{bmatrix} \sin \theta \\ \sin(\theta - \frac{2}{3} \pi) \\ \sin(\theta + \frac{2}{3} \pi) \end{bmatrix} \right) \right) \quad (6.25)$$

Pro výpočet těchto napěťových rovnic tedy nyní potřebuji určit inverzi matice indukčností  $L_s$ . Z důvodu výpočetní náročnosti už budu uvažovat indukčnosti ve všech větvích shodné (6.11). Příklad s obecným rozložením indukčností je popsán v další kapitole.

Jednotlivé prvky inverzní matice spočítám jako podíl determinantu submatice počítaného prvku a determinantu celé matice. U jednotlivých prvků inverzní matice se musím řídit následujícími znaménky:

$$L_s^{-1} = \begin{bmatrix} +1 & -1 & +1 \\ -1 & +1 & -1 \\ +1 & -1 & +1 \end{bmatrix} \quad (6.26)$$

Všechny prvky na hlavní diagonále budou mít následující tvar:

$$L_{aa}^{-1} = L_{bb}^{-1} = L_{cc}^{-1} = \frac{\det L_{aa}}{\det L_s} = \frac{L_{ls}^2 + 2 L_{ls} L + \frac{3}{4} L^2}{L_{ls}^3 + 3 L_{ls}^2 L + \frac{9}{4} L_{ls} L^2} \quad (6.27)$$

Stejně tak i všechny prvky mimo hlavní diagonálu budou mít shodný tvar a budou vypadat po zkrácení následovně:

$$L_{ab}^{-1} = L_{bc}^{-1} = L_{ac}^{-1} = \frac{\det L_{ac}}{\det L_s} = \frac{\frac{L_{ls} L}{2} + \frac{3}{4} L^2}{L_{ls}^3 + 3 L_{ls}^2 L + \frac{9}{4} L_{ls} L^2} = \frac{L}{L_{ls} (2 L_{ls} + 3 L)} \quad (6.28)$$

Vypočítanou inverzní matici indukčností nyní dosadím do stavových rovnic proudů, které se pokusím maximálně upravit.

$$\frac{d}{dt} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} = \begin{bmatrix} \frac{L_{ls}^2 + 2 L_{ls} L + 0,75 L^2}{L_{ls}^3 + 3 L_{ls}^2 L + 2,25 L_{ls} L^2} & \frac{L}{L_{ls} (2 L_{ls} + 3 L)} \\ \frac{L}{L_{ls} (2 L_{ls} + 3 L)} & \frac{L_{ls}^2 + 2 L_{ls} L + 0,75 L^2}{L_{ls}^3 + 3 L_{ls}^2 L + 2,25 L_{ls} L^2} \\ \frac{L}{L_{ls} (2 L_{ls} + 3 L)} & \frac{L}{L_{ls} (2 L_{ls} + 3 L)} \end{bmatrix} \begin{bmatrix} L_{ac} \\ L_{bc} \\ L_{cc} \end{bmatrix} \begin{bmatrix} u_a - R_s i_a - \psi_f \omega \cos \theta \\ u_b - R_s i_b - \psi_f \omega \cos(\theta - \frac{2}{3} \pi) \\ u_c - R_s i_c - \psi_f \omega \cos(\theta + \frac{2}{3} \pi) \end{bmatrix} \quad (6.29)$$

Celková matice stavových proměnných pak bude mít následující tvar. Algebraické smyčky už se v ní nevyskytují a lze tedy pro tento systém sestavit model.

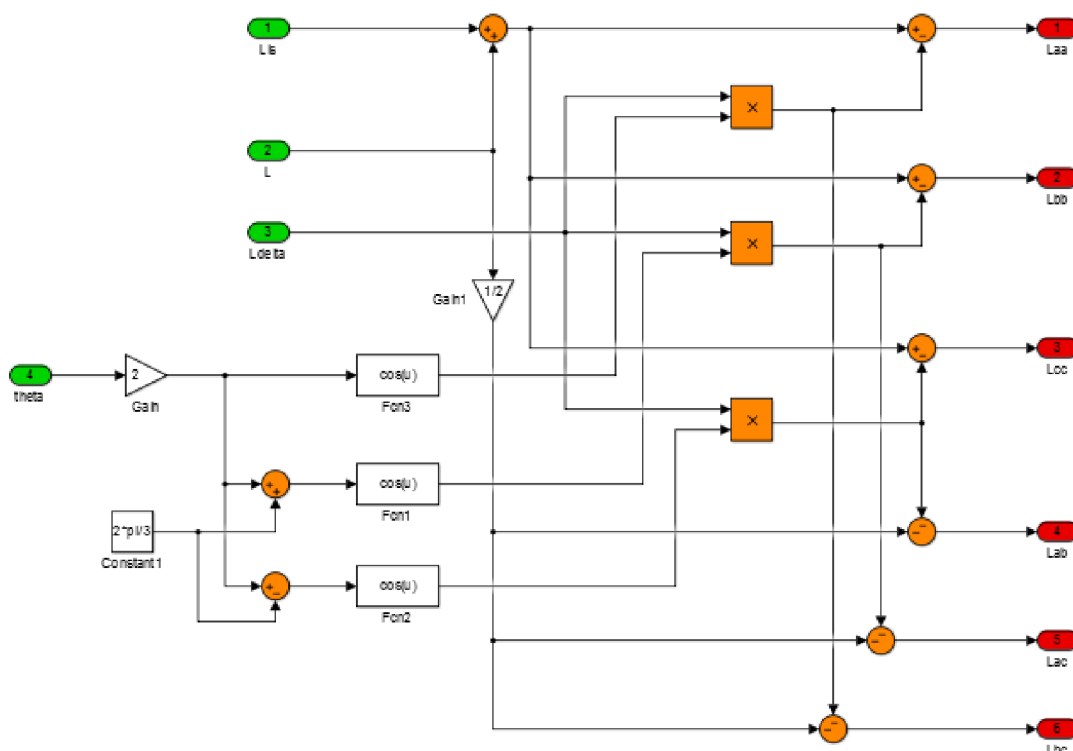
$$\frac{d}{dt} \begin{bmatrix} i_a \\ i_b \\ i_c \\ \omega \\ \theta \end{bmatrix} = \begin{bmatrix} L_{aa}^{-1} (u_a - R_s i_a - \psi_f \omega \cos \theta) + L_{bc}^{-1} [u_b + u_c - R_s (i_b + i_c) - \psi_f \omega (\cos(\theta - \frac{2}{3} \pi) + \cos(\theta + \frac{2}{3} \pi))] \\ L_{bb}^{-1} (u_b - R_s i_b - \psi_f \omega \cos(\theta - \frac{2}{3} \pi)) + L_{ac}^{-1} [u_a + u_c - R_s (i_a + i_c) - \psi_f \omega (\cos \theta + \cos(\theta + \frac{2}{3} \pi))] \\ L_{cc}^{-1} (u_c - R_s i_c - \psi_f \omega \cos(\theta + \frac{2}{3} \pi)) + L_{ab}^{-1} [u_a + u_b - R_s (i_a + i_b) - \psi_f \omega (\cos \theta + \cos(\theta - \frac{2}{3} \pi))] \\ \frac{1}{J} [\frac{3}{2} P_p \psi_f (\frac{\sqrt{3}}{2} (i_b - i_c) \cos \theta - (i_a - \frac{1}{2} i_b - \frac{1}{2} i_c) \sin \theta) - M_z - K_{TL} \frac{\omega}{P_p}] \\ \omega \\ \theta \end{bmatrix} \quad (6.30)$$



## 6.1.2 Model PMSM motoru v abc souřadnicích – s obecným rozložením indukčností

Zjednodušený model PMSM motoru v abc souřadnicích se symetrickým rozložením indukčností popsáný v předchozí kapitole, již neumožňuje testování různých nesymetričností motoru, proto jsem vytvořil model tohoto motoru i bez zmíněného zjednodušení.

Vzhledem k vysokému počtu členů tohoto modelu s obecným rozložením indukčností nebudu jednotlivé rovnice modelu vypisovat a ponechám členy matice  $L_s$  (vzorec 6.10) v nevyjádřené podobě. Podrobně tedy matici  $L_s$  namodeluji přímo v Simulinku.

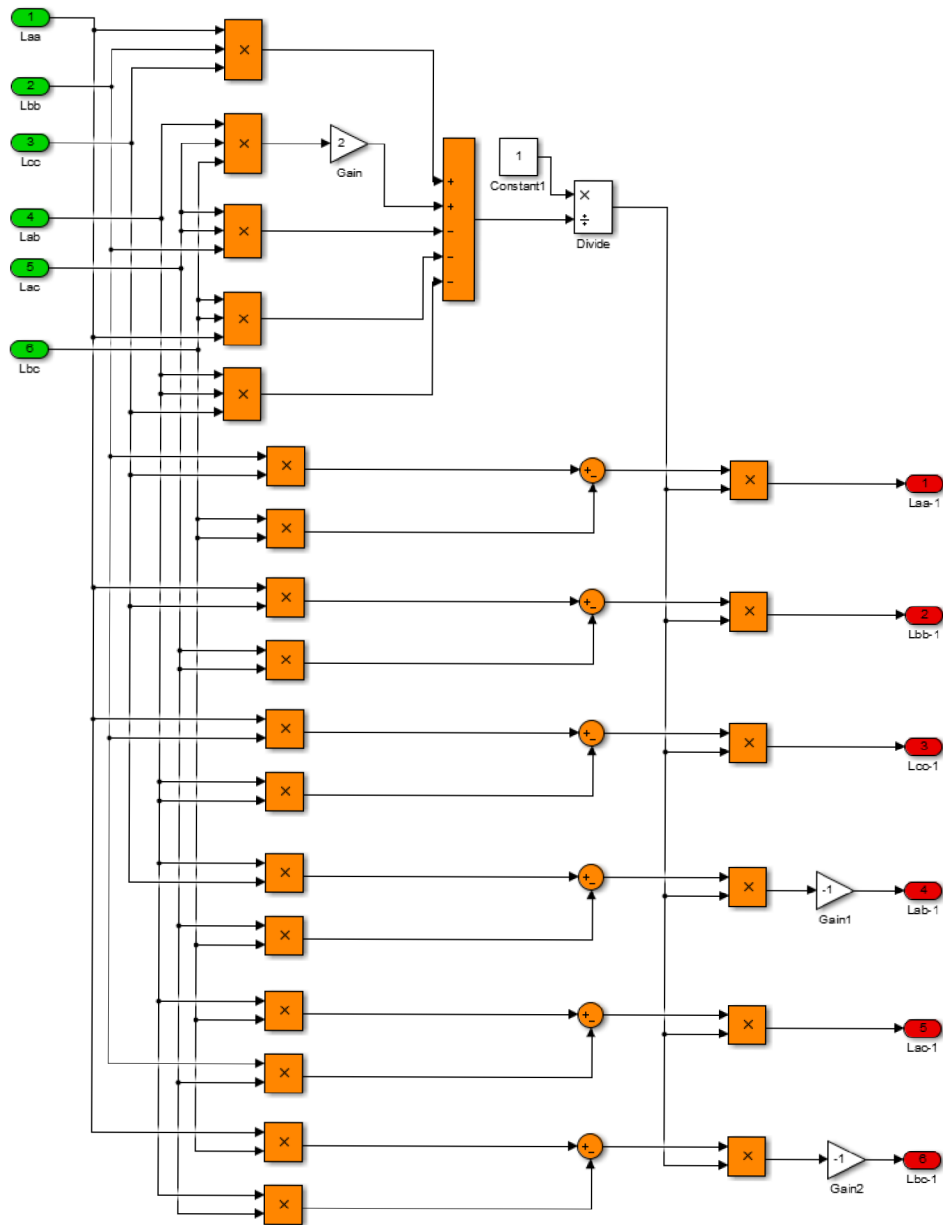


Obr. 20 : Namodelovaná matice indukčností pro abc model v Simulinku

Stejně jako v předchozí kapitole musím určit inverzi matice indukčností  $L_s$ , tentokrát přímo v nezjednodušené podobě (vzorec 6.10). Obecně pomocí determinantu a subdeterminantů matice  $L_s$  můžu vyjádřit inverzní matici takto:

$$L_s^{-1} = \begin{bmatrix} L_{aa}^{-1} & L_{ab}^{-1} & L_{ac}^{-1} \\ L_{ab}^{-1} & L_{bb}^{-1} & L_{bc}^{-1} \\ L_{ac}^{-1} & L_{bc}^{-1} & L_{cc}^{-1} \end{bmatrix} = \begin{bmatrix} \frac{\det L_{aa}}{\det L_s} & -\frac{\det L_{ab}}{\det L_s} & \frac{\det L_{ac}}{\det L_s} \\ -\frac{\det L_{ab}}{\det L_s} & \frac{\det L_{bb}}{\det L_s} & -\frac{\det L_{bc}}{\det L_s} \\ \frac{\det L_{ac}}{\det L_s} & -\frac{\det L_{bc}}{\det L_s} & \frac{\det L_{cc}}{\det L_s} \end{bmatrix} \quad (6.31)$$

Podrobně inverzi této matice vyjádřím opět přímo modelem v Simulinku. Výstupy této inverze pak použiji pro model v abc souřadnicích.



Obr. 21 : Namodelovaná inverze matice indukčností pro abc model

Stavové rovnice pro proudy  $i_{abc}$  pak můžu vyjádřit z rovnice 6.25 následovně:

$$\frac{d}{dt} i_{abc} = \begin{bmatrix} L_{aa}^{-1} & L_{ab}^{-1} & L_{ac}^{-1} \\ L_{ab}^{-1} & L_{bb}^{-1} & L_{bc}^{-1} \\ L_{ac}^{-1} & L_{bc}^{-1} & L_{cc}^{-1} \end{bmatrix} \begin{bmatrix} u_a - R_s i_a - \psi_f \cos \theta \\ u_b - R_s i_b - \psi_f \cos(\theta - \frac{2}{3} \pi) \\ u_c - R_s i_c - \psi_f \cos(\theta + \frac{2}{3} \pi) \end{bmatrix} = \dots$$

$$\dots = \begin{bmatrix} L_{aa}^{-1}(u_a - R_s i_a - \psi_f \cos \theta) - L_{ab}^{-1}(u_b - R_s i_b - \psi_f \cos(\theta - \frac{2}{3} \pi)) + L_{ac}^{-1}(u_c - R_s i_c - \psi_f \cos(\theta + \frac{2}{3} \pi)) \\ -L_{ab}^{-1}(u_a - R_s i_a - \psi_f \cos \theta) + L_{bb}^{-1}(u_b - R_s i_b - \psi_f \cos(\theta - \frac{2}{3} \pi)) - L_{bc}^{-1}(u_c - R_s i_c - \psi_f \cos(\theta + \frac{2}{3} \pi)) \\ L_{ac}^{-1}(u_a - R_s i_a - \psi_f \cos \theta) - L_{bc}^{-1}(u_b - R_s i_b - \psi_f \cos(\theta - \frac{2}{3} \pi)) + L_{cc}^{-1}(u_c - R_s i_c - \psi_f \cos(\theta + \frac{2}{3} \pi)) \end{bmatrix} \quad (6.32)$$

Stavové rovnice pro úhlovou rychlost  $\omega$  a pro polohu motoru  $\theta$  zůstávají stejné jako u abc modelu se symetrickým rozložením indukčností. (6.30)

Namodelované jsou tyto stavové rovnice ukázány v Simulinku v další kapitole Realizace Matlab, stejně tak jsou tam k vidění i průběhy otáček a proudů tohoto abc modelu a jeho srovnání s dq modelem.

Model v abc souřadnicích s obecným rozložením indukčností se chová obdobně jako model se symetrickým rozložením. Ani jeden z těchto modelů vytvořených v mé práci však neodpovídá chování vytvořeného modelu PMSM motoru v dq souřadnicích.

Tato nepřesnost může být dána závislostí celého modelu na aktuální hodnotě polohy motoru, kterou nelze měřit ani vypočítávat zcela přesně. Podrobněji jsou výsledky tohoto testování popsány ve výše zmíněné kapitole společně s grafy popisujícími tento model.

### 6.1.3 Model PMSM motoru v dq souřadnicích

Pro řešení rovnic matematického modelu elektrického pohonu je vhodné použít souřadný systém spojený s rotorem stroje (dq). V rovnicích tohoto modelu se nevyskytují trigonometrické funkce a velmi se zjednoduší i vyjádření indukčností motoru, které lze nyní vyjádřit přímo v osách dq systému.

Pro odvození matematického modelu tohoto synchronního motoru v dq souřadnicích vyjdeme opět z napěťových rovnic obecného synchronního stroje: [1]

$$u_d = R_s i_d + \frac{d\psi_d}{dt} - \omega \psi_q \quad (6.33)$$

$$u_q = R_s i_q + \frac{d\psi_q}{dt} + \omega \psi_d \quad (6.34)$$

Na základě zjednodušení pro magnetická spřažení motoru platí vztahy: [1]

$$\psi_d = L_d i_d + \psi_f \quad (6.35)$$

$$\psi_q = L_q i_q \quad (6.36)$$

Dosazením vztahů pro magnetická spřažení (6.35) a (6.36) do rovnic obecného synchronního stroje a jejich úpravou získám následující vztahy:

$$u_d = R_s i_d + \frac{d\psi_d}{dt} - \omega \psi_q = R_s i_d + L_d \frac{di_d}{dt} - \omega L_q i_q \quad (6.37)$$

$$u_q = R_s i_q + \frac{d\psi_q}{dt} + \omega \psi_d = R_s i_q + L_q \frac{di_q}{dt} + \omega (L_d i_d + \psi_f) \quad (6.38)$$

Z těchto rovnic vyjádřím derivace statorových proudů, které budou pro můj model stavovými veličinami. Po Laplaceově transformaci získám první dvě stavové rovnice v tomto tvaru: [1]

$$pI_d = \frac{1}{L_d}(U_d - R_s I_d + \omega L_q I_q) \quad (6.39)$$

$$pI_q = \frac{1}{L_q}(U_q - R_s I_q - \omega L_d I_d - \omega \psi_f) \quad (6.40)$$

Maticově zapsané tyto dvě stavové rovnice vypadají následovně:

$$p \begin{bmatrix} I_d \\ I_q \end{bmatrix} = \begin{bmatrix} \frac{-R_s}{L_d} & \frac{\omega L_q}{L_d} \\ \frac{-\omega L_d}{L_q} & \frac{-R_s}{L_q} \end{bmatrix} \begin{bmatrix} I_d \\ I_q \end{bmatrix} + \begin{bmatrix} \frac{1}{L_d} & 0 \\ 0 & \frac{1}{L_q} \end{bmatrix} \begin{bmatrix} U_d \\ U_q - \omega \psi_f \end{bmatrix} \quad (6.41)$$

Pro mechanickou rychlost, třetí stavovou proměnnou, odvodím stavovou rovnici z pohybové rovnice. [1]

$$p_p \Omega_m = \frac{1}{J}(M_e - M_z) \quad (6.42)$$

Kde elektromagnetický moment stroje  $M_e$  mohu vyjádřit takto: [1]

$$M_e = \frac{3}{2} p_p (\psi_d i_q - \psi_q i_d) = \frac{3}{2} p [\psi_f + (L_d - L_q) i_d] i_q \quad (6.43)$$

Pro stroj s hladkým rotorem ( $L_d = L_q$ ) se rovnice dá zjednodušit na tvar: [1]

$$M_e = \frac{3}{2} p_p \psi_f i_q \quad (6.44)$$

Moment stroje je tedy vyjádřen jako součin kolmé složky prostorového vektoru statorového proudu a konstantního budícího magnetického spřažení, které je dáno pouze vlastnostmi použitých permanentních magnetů.

Pro přesnější simulaci reálného motoru rovnici doplním ještě o člen zahrnující tlumení stroje  $K_{TL}$ . Stavová rovnice pro úhlovou rychlost pak vypadá následovně:

$$\frac{d}{dt}\omega = \frac{1}{J} \left( \frac{3}{2} P_p [\psi_f + (L_d - L_q) i_d] i_q - M_z - K_{TL} \frac{\omega}{P_p} \right) \quad (6.45)$$

Maticově zapsané stavové rovnice modelu v dq souřadnicích PMSM motoru pak mají tento tvar:

$$\frac{d}{dt} \begin{bmatrix} i_d \\ i_q \\ \omega \\ \theta \end{bmatrix} = \begin{bmatrix} \frac{-R}{L_d} & \frac{L_q P_p \omega}{L_d} & 0 & 0 \\ \frac{-L_d P_p \omega}{L_q} & \frac{-R}{L_q} & \frac{-\psi_{PM} P_p}{L_q} & 0 \\ 0 & \frac{1}{J} 1,5 P_p (\psi_f + L_d i_d - L_q i_q) & \frac{-K_{TL}}{P_p} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} i_d \\ i_q \\ \omega \\ \theta \end{bmatrix} + \begin{bmatrix} \frac{1}{L_d} & 0 & 0 \\ 0 & \frac{1}{L_q} & 0 \\ 0 & 0 & \frac{-1}{J} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_d \\ u_q \\ M_z \end{bmatrix} \quad (6.46)$$

## 6.2 Diskrétní modely PMSM

Pro diskretizaci vypočtených matematických modelů v dq systému použijí Eulerovu diskretizaci i diskretizaci 2. řádu, abych mohl porovnat, zda-li se výrazně liší přesnost těchto způsobů diskretizace. Pro diskretizace použijí vztahy odvozené v teoretickém úvodu této práce.

Před výpočtem diskrétních matic  $A_k$  a  $B_k$  musím také vhodně zvolit vzorkovací periodu  $T$ . Její volbě se budu věnovat v kapitole zabývající se realizací modelu v Matlabu.

Při tvorbě modelu pak nesmím zapomenout nahradit bločky integrátorů zpoždovacími členy a nastavit u nich správnou vzorkovací periodu.

## 6.2.1 Diskrétní model PMSM v abc souřadnicích 1.řádu – symetrické rozložení indukčností

Již dříve odvozené vztahy pro Eulerovu diskretizaci systému s periodou vzorkování T: [10]

$$A_K = I + AT \quad (6.47)$$

$$B_K = A^{-1}(I + AT - I)B = TB \quad (6.48)$$

Po dosazení matic A a B spojitého modelu v souřadných osách abc do těchto rovnic získám následující diskrétní matice  $A_K$  a  $B_K$ . Matici  $A_K$  jsem musel graficky rozdělit na dvě části:

$$A_K = \begin{bmatrix} 1 - T L_{aa}^{-1} R_s & -T L_{bc}^{-1} R_s & -T L_{bc}^{-1} R_s & \dots \\ -T L_{ac}^{-1} R_s & 1 - T R_s L_{bb}^{-1} & -T L_{ac}^{-1} R_s & \dots \\ -T L_{ab}^{-1} R_s & -T R_s L_{ab}^{-1} & 1 - T L_{cc}^{-1} R_s & \dots \\ \frac{-T}{J} \frac{3}{2} p_p \psi_f \sin \theta & \frac{T}{J} \frac{3}{2} p_p \psi_f \left( \frac{\sqrt{3}}{2} \cos \theta + \frac{1}{2} \sin \theta \right) & \frac{T}{J} \frac{3}{2} p_p \psi_f \left( \frac{-\sqrt{3}}{2} \cos \theta + \frac{1}{2} \sin \theta \right) & \dots \\ 0 & 0 & 0 & \dots \\ \dots & -T \psi_f (L_{aa}^{-1} \cos \theta + L_{bc}^{-1} (\cos(\theta - \frac{2}{3} \pi) + \cos(\theta + \frac{2}{3} \pi))) & 0 & \dots \\ \dots & -T \psi_f (L_{bb}^{-1} \cos(\theta - \frac{2}{3} \pi) + L_{ac}^{-1} (\cos \theta + \cos(\theta + \frac{2}{3} \pi))) & 0 & \dots \\ \dots & -T \psi_f (L_{cc}^{-1} \cos(\theta + \frac{2}{3} \pi) + L_{ab}^{-1} (\cos \theta + \cos(\theta - \frac{2}{3} \pi))) & 0 & \dots \\ \dots & 1 - T \frac{K_{TL}}{P_p} & 0 & \dots \\ \dots & T & 1 & \dots \end{bmatrix} \quad (6.49)$$

$$B_K = \begin{bmatrix} T L_{aa}^{-1} & T L_{bc}^{-1} & T L_{bc}^{-1} & 0 \\ T L_{ac}^{-1} & T L_{bb}^{-1} & T L_{ac}^{-1} & 0 \\ T L_{ab}^{-1} & T L_{ab}^{-1} & T L_{cc}^{-1} & 0 \\ 0 & 0 & 0 & \frac{-T}{J} \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (6.50)$$

## 6.2.2 Diskrétní model PMSM v dq souřadnicích 1.řádu

I zde použijí pro diskretizaci systému stejné vztahy jako v předchozí kapitole. V dq systému celé diskrétní matice  $A_k$  i  $B_k$  vyjdou výrazně jednodušeji. [10]

$$A_k = I + AT \quad (6.51)$$

$$B_k = TB \quad (6.52)$$

Použitím vzorce se nám díky přičtení jednotkové matice objeví jednoduchá zpětná vazba u všech zpoždovacích členů. Všechny členy matic  $A_k$  i  $B_k$  jsou navíc pronásobeny periodou vzorkování  $T$ :

$$A_k = \begin{bmatrix} 1 - T \frac{R}{L_d} & T \frac{L_q \omega_{el}}{L_d} & 0 & 0 \\ -T \frac{L_d \omega_{el}}{L_q} & 1 - T \frac{R}{L_q} & -T \frac{\psi_f P_p}{L_q} & 0 \\ 0 & \frac{T}{J} 1,5 P_p (\psi_f + L_d i_d - L_q i_q) & 1 - T \frac{K_{TL}}{P_p} & 0 \\ 0 & 0 & T & 1 \end{bmatrix} \quad (6.53)$$

$$B_k = \begin{bmatrix} \frac{T}{L_d} & 0 & 0 \\ 0 & \frac{T}{L_q} & 0 \\ 0 & 0 & \frac{-T}{J} \\ 0 & 0 & 0 \end{bmatrix} \quad (6.54)$$



## 6.2.3 Diskrétní model PMSM v dq souřadnicích 2.řádu

Na rozdíl od Eulerovy diskretizace se u diskretizace druhého řádu rozšíří vzorce pro matici  $A_k$  a  $B_k$  ještě vždy o kvadratický člen. Diskrétní matice systému se tím poměrně výrazně zesložití. Odvození těchto vzorců provádím v kapitole Teoretického úvodu. [10]

$$A_K = I + AT + \frac{(AT)^2}{2} \quad (6.55)$$

$$B_K = TB + \frac{AT^2}{2} B \quad (6.56)$$

Použitím těchto vzorců získám kvadraticky diskretizované matice systému, kde jsou opět patrné jednoduché zpětné vazby všech stavových proměnných:

$$A_K = \begin{bmatrix} 1 - T \frac{R}{L_d} + \frac{T^2}{2} \left[ \left( \frac{R}{L_d} \right)^2 - \omega_{el}^2 \right] & T \frac{L_q P_p \omega_{el}}{L_d} - \frac{T^2}{2} \omega_{el} R \left( \frac{L_q}{L_d^2} + \frac{1}{L_d} \right) & -\frac{T^2}{2} \omega_{el} \frac{P_p \psi_{PM}}{L_d} & 0 \\ -T \omega_{el} \frac{L_d}{L_q} + \frac{T^2}{2} \omega_{el} R \left( \frac{L_d}{L_q^2} + \frac{1}{L_q} \right) & 1 - T \frac{R}{L_q} + \frac{T^2}{2} \left[ \left( \frac{R}{L_q} \right)^2 - \omega_{el}^2 \right] & -T \frac{\psi_{PM} P_p}{L_q} + \frac{T^2}{2} \frac{R P_p \psi_{PM}}{L_q^2} & 0 \\ -\frac{T^2}{2} \omega_{el} \frac{1}{J} 1,5 P_p \psi_{PM} \frac{L_d}{L_q} & \left( \frac{T}{J} - \frac{T^2}{2} \frac{1}{J} \frac{R}{L_q} \right) 1,5 P_p \psi_{PM} & 1 - \frac{T^2}{2} \frac{1}{J} 1,5 P_p \psi_{PM} \frac{P_p \psi_{PM}}{L_q} & 0 \\ 0 & \frac{T^2}{2} \frac{1}{J} 1,5 P_p \psi_{PM} & T & 1 \end{bmatrix} \quad (6.57)$$

$$B_K = \begin{bmatrix} \frac{T}{L_d} - \frac{T^2}{2} \frac{1}{L_d} \frac{R}{L_d} & \frac{T^2}{2} \omega_{el} \frac{1}{L_d} & 0 \\ -\frac{T^2}{2} \frac{1}{L_q} & \frac{T}{L_q} - \frac{T^2}{2} \frac{1}{L_q} \frac{R}{L_q} & \frac{T^2}{2} \frac{1}{J} \frac{P_p \psi_{PM}}{L_q} \\ 0 & \frac{T^2}{2} \frac{1}{L_q} \frac{1}{J} 1,5 P_p \psi_{PM} & -\frac{T}{J} \\ 0 & 0 & -\frac{T^2}{2} \frac{1}{J} \end{bmatrix} \quad (6.58)$$

## 7 REALIZACE MATLAB

Po vytvoření diskretizovaných matematických modelů PMSM motoru můžu přejít k dalšímu kroku této práce, vytvořit tyto modely v programu Matlab Simulink a to podle pravidel, s kterými je bude možné převést do HDL kódu. Zároveň se v tomto kroku pokusím o nalezení ideálního diskrétního řešení, které bude co nejpřesněji odpovídat spojitému modelu.

Aby mohl HDL kodér programu Matlab Simulink diskretizovaný model převést do HDL kódu, musí splňovat několik pravidel. Některé z těchto pravidel, jako je nutnost nadefinování přesností proměnných nebo pravidla pro použití trigonometrických funkcí, už jsem popsal v kapitole s přípravnými úlohami. Mezi další požadavky patří například používat blok násobičky s maximálně dvěma vstupy nebo nepoužívat v modelu žádné dělení.

Pro co nejpřesnější výsledek, který zároveň nebude příliš výpočetně náročný, se budu zabývat především možnostmi volby periody vzorkování a nastavováním nejvhodnějších přesností jednotlivých modelů. Podle přesností jednotlivých řešení porovnám, která z možností diskretizace je pro tuto úlohu vhodnější, zda Eulerova, či kvadratická diskretizace.

### 7.1 Finální modely v Simulinku

Podle stavových diferenciálních rovnic jednotlivých diskrétních matematických modelů popsaných v předchozí kapitole jsem v programu Matlab Simulink namodeloval následující schémata.

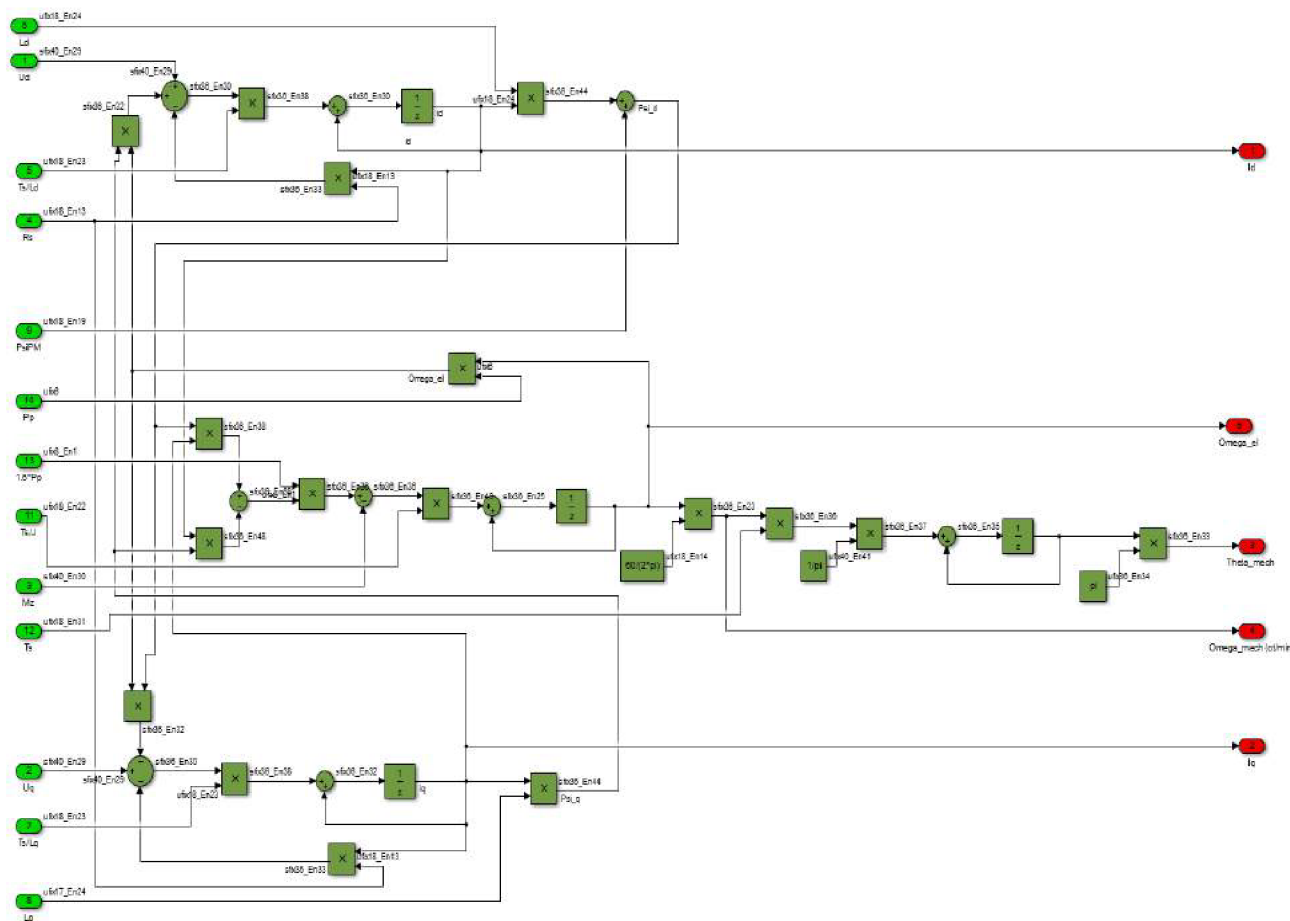
Modely jsou příliš velké na to, aby se daly vložit přehledně a dostatečně podrobně přímo do textu, vloží sem tedy pouze náhledy pro jistou představu. Modely k podrobnému prohlédnutí budou k dispozici v příloze této práce.

Při vytváření modelů jsem se snažil zachovat přehlednost pomyslným grafickým oddělením jednotlivých stavových rovnic a přehledným uspořádáním vstupů na levé a výstupů na pravé straně. Samozřejmě jsem při tvorbě dbal na všechna pravidla popsaná výše, sloužící k správnému převodu modelů do HDL kódu.

## 7.1.1 Diskrétní model Eulerovou diskretizací v dq systému

K podrobnému náhledu je toto schéma diskrétního modelu Eulerovou diskretizací v dq souřadnicích k dispozici na přiloženém CD.

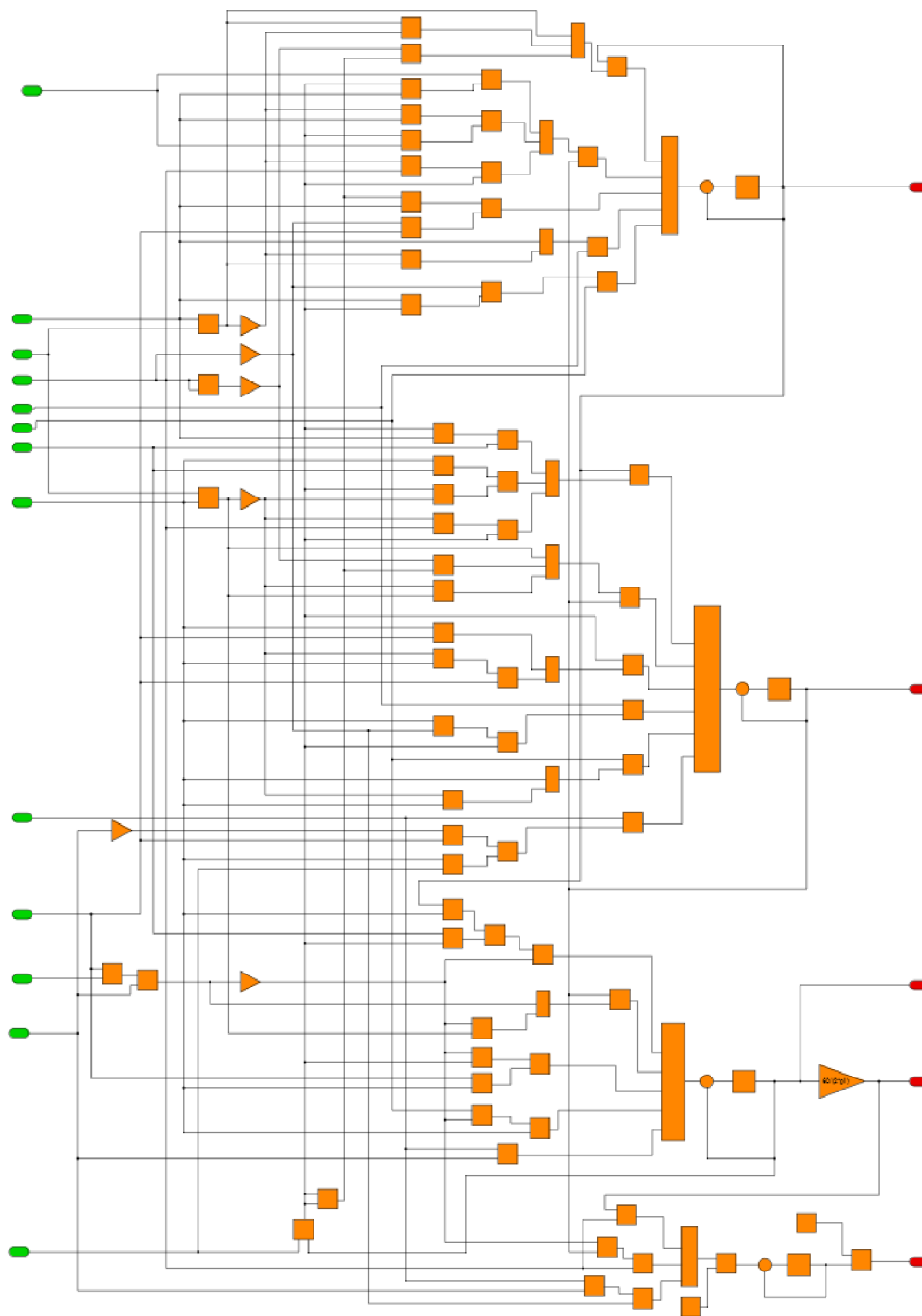
Proti modelu získanému kvadratickou diskretizací vyniká tento na první pohled svou jednoduchostí. Před modelem v abc souřadnicích má pak obrovskou výhodu v nepřítomnosti trigonometrických funkcí. Jeho nevýhodou je naopak nutná symetričnost modelu, takže na něm není možné testovat různé zkraty a další nesymetričnosti v jednotlivých větvích vinutí.



Obr. 22 : Náhled diskrétního modelu Eulerovou diskretizací v dq systému

## 7.1.2 Diskrétní model kvadratickou diskretizací v dq systému

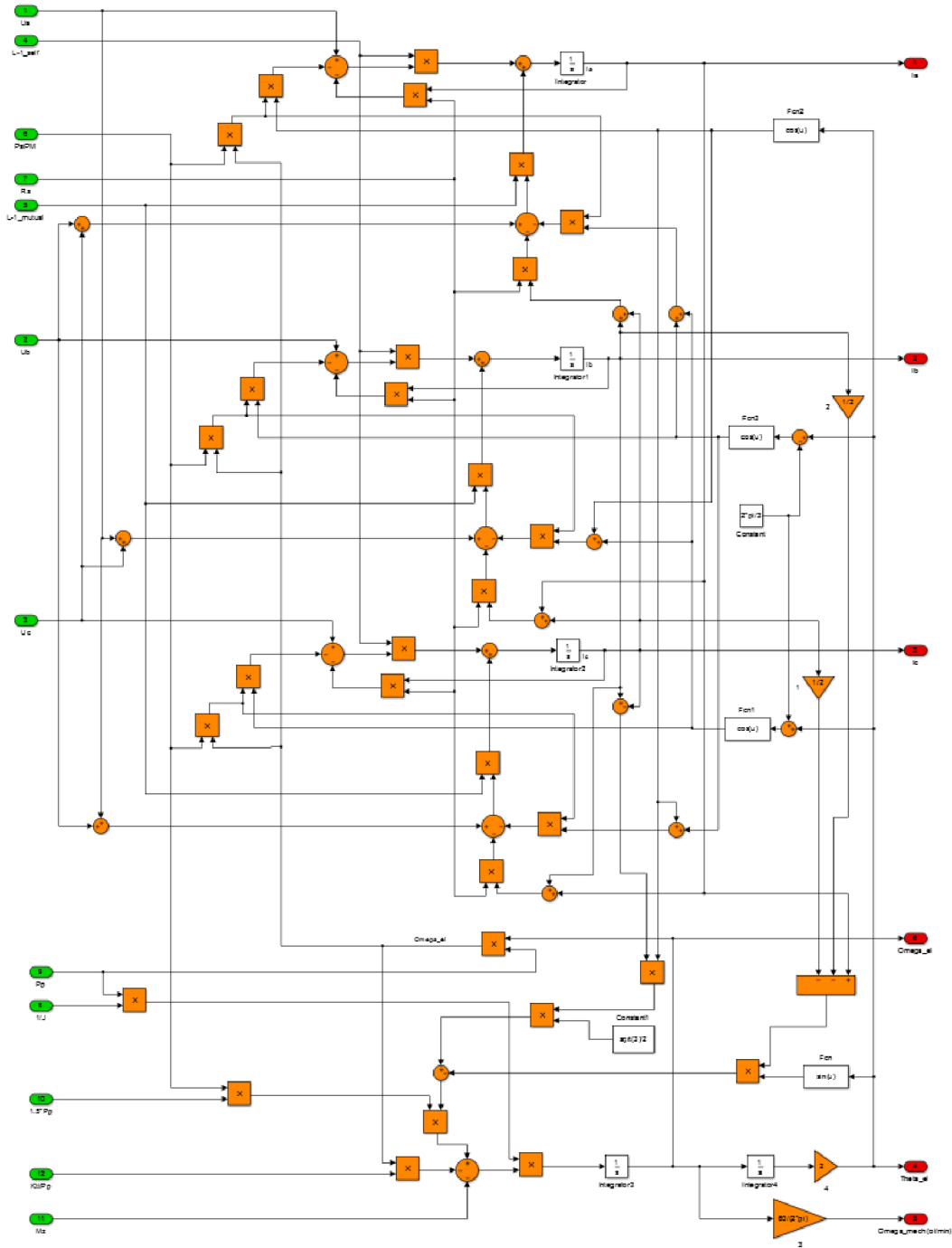
K podrobnému náhledu je toto schéma diskrétního modelu kvadratickou diskretizací v dq souřadnicích k dispozici na příloženém CD.



Obr. 23 : Náhled diskrétního modelu kvadratickou diskretizací v dq systému

### 7.1.3 Diskrétní model Eulerovou diskretizací v abc systému

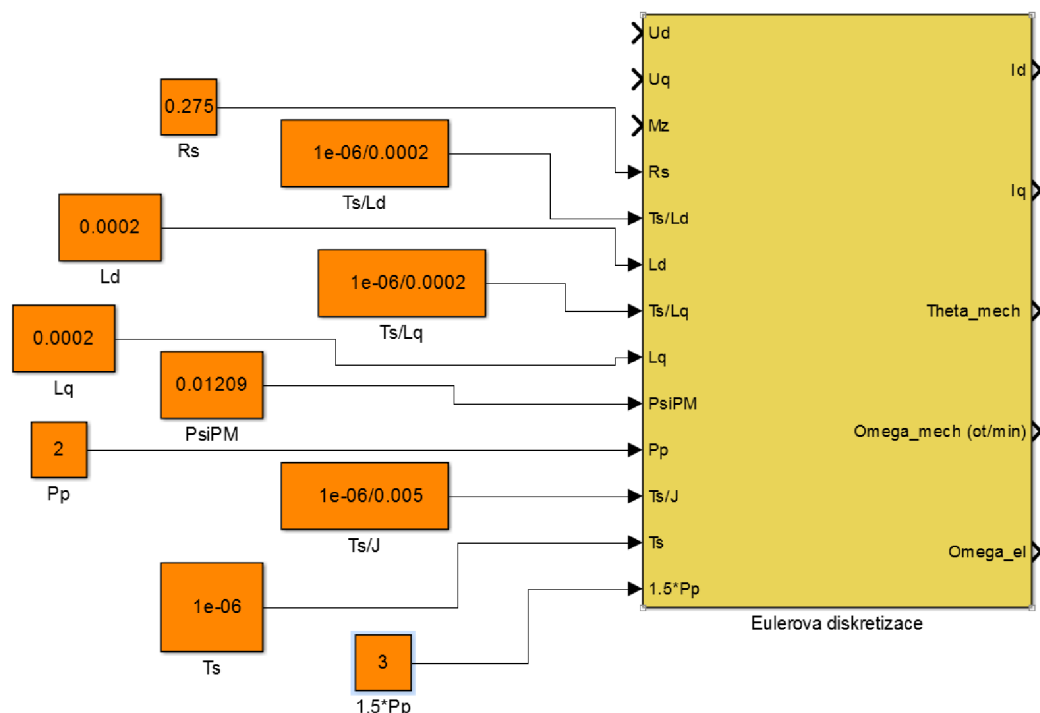
K podrobnému náhledu je toto schéma diskrétního modelu Eulerovou diskretizací v abc souřadnicích k dispozici na příloženém CD.



Obr. 24 : Náhled diskrétního modelu Eulerovou diskretizací v abc systému

## 7.2 Testovací schéma v Simulinku

Pro větší přehlednost testovacího programu zapouzdřuji jednotlivé modely do subsystémů, jak je vidět na obrázku níže. Na obrázku jsou také vidět parametry PMSM motoru, s kterými pracuji při testování v Simulinku. (Pozn. Při testování s reálným motorem jsem tyto hodnoty měnil)



Obr. 25 : Subsystém modelu PMSM motoru se vstupními hodnotami

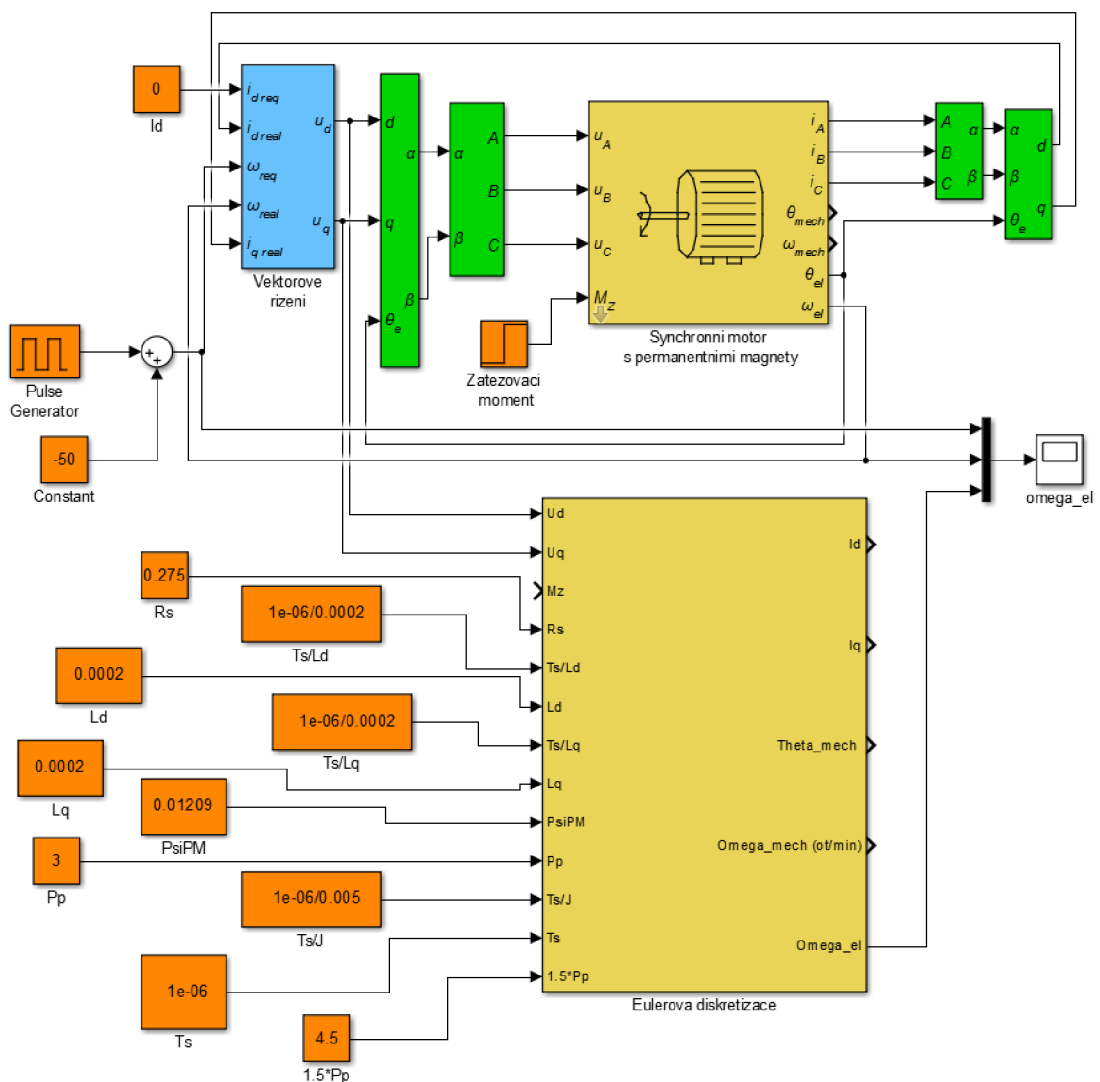
Pro přehlednost uvádím tyto hodnoty pro testování chování modelů v Simulinku i v tabulce.

Název parametru	Značení	Jednotka	Hodnota
Indukčnost d složky	$L_d$	[H]	0,0002
Indukčnost q složky	$L_q$	[H]	0,0002
Odpor statorového vinutí	$R_s$	[ $\Omega$ ]	0,275
Konstanta permanentního magnetu	$\Psi_f$	[Wb]	0,01209
Moment setrvačnosti	$J$	[kg m <sup>2</sup> ]	0,005
Počet pólových dvojic	$P_p$	[-]	2

Tab. 3: Parametry PMSM motoru pro testování modelů v Simulinku

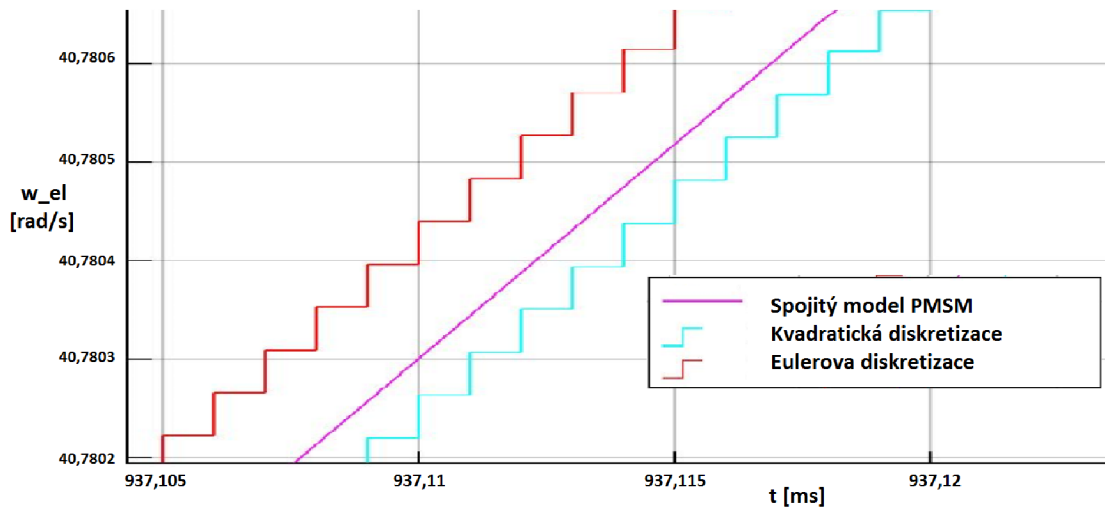
K testování nových modelů používám následující schéma s již ověřeným spojitým modelem PMSM motoru řízeným pomocí vektorového řízení. Transformace mezi vektorovým řízením a modelem mi umožňují testování modelů v dq,  $\alpha\beta$  i abc systému.

Z vektorového řízení si do testovaného modelu přivádím požadovaná vstupní napětí a sleduji, jestli se výstupy shodují s řízeným spojitým modelem. Při všech testováních v Simulinku nastavuji požadovanou elektrickou úhlovou rychlost  $\omega_{el}$  jako obdélkový signál s rozsahem mezi 50 a -50 rad/s a s periodou 6s, to znamená, že každé 3 sekundy dochází ke skokové změně žádané hodnoty úhlové rychlosti.



Obr. 26 : Testovací schéma modelů v Simulinku

Popsaným způsobem jsem úspěšně otestoval oba způsoby diskretizace spojitého modelu v dq souřadném systému. Na přiblíženém průběhu úhlové rychlosti je vidět, že chyba diskretizovaných průběhů je v obou případech téměř zanedbatelná, je také pěkně patrný schodovitý průběh způsobený diskretizací o periodě vzorkování  $T = 1\mu\text{s}$ .



Graf 2: Porovnání diskretizací v dq souřadnicích se spojitým průběhem

Podobného úspěchu jsem bohužel nedosáhl s diskrétním ani se spojitým průběhem modelu v abc souřadnicích, který se s průběhy spojitého modelu v dq souřadnicích neshoduje.

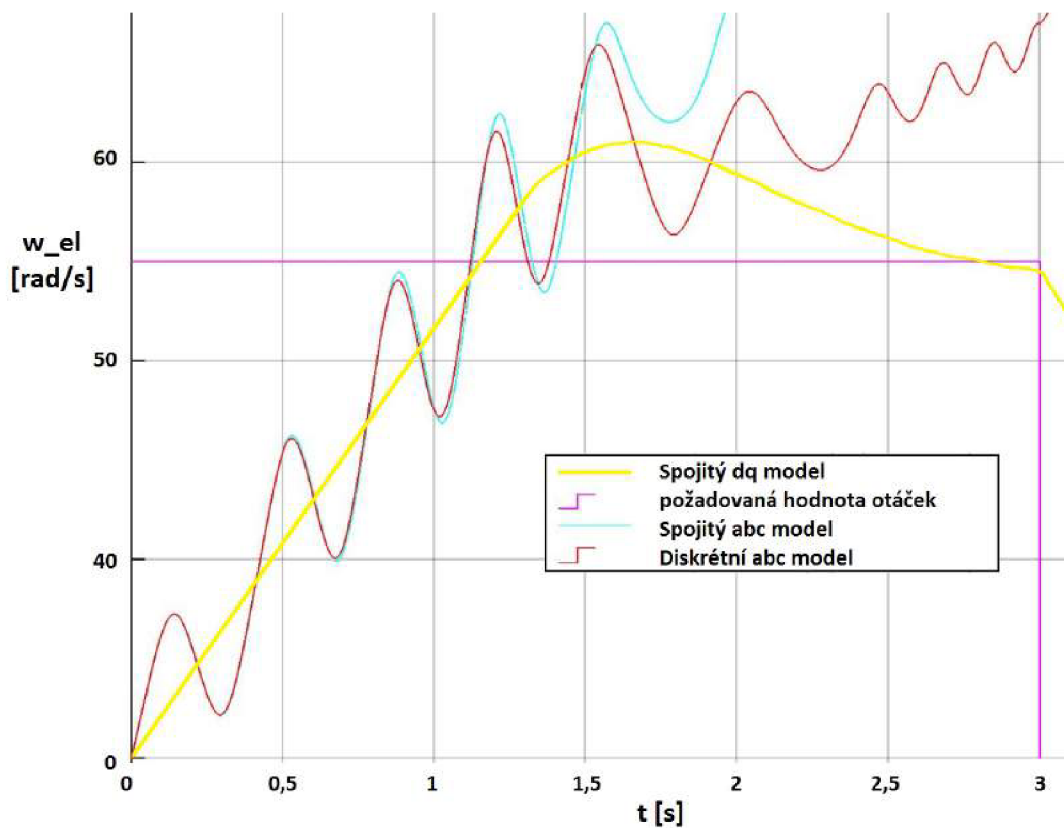
U modelu v abc souřadnicích jsem pouze experimentálně zkoušel nastavovat hodnoty indukčností  $L$ ,  $L_{ls}$  a  $L_{\Delta}$ . Vzhledem k širokému rozsahu zkoušených hodnot myslím mohu vyloučit, že by chyba tohoto průběhu byla způsobena pouze jejich volbou.

Testováním modelu s postupně přiváděnými proudy, polohou i otáčkami z modelu v dq souřadnicích do zpětných vazeb modelu abc, stejně jako opakovanými výpočty mohu vyloučit i výpočetní chybu způsobenou přehlédnutím nebo špatně provedenými matematickými operacemi.

Průběhy abc modelu tak zřejmě neodpovídají ověřeným průběhům dq modelu kvůli výrazné závislosti abc modelu na aktuální hodnotě polohy motoru, kterou nelze počítat zcela přesně a může způsobovat velkou chybu v chování celého modelu.

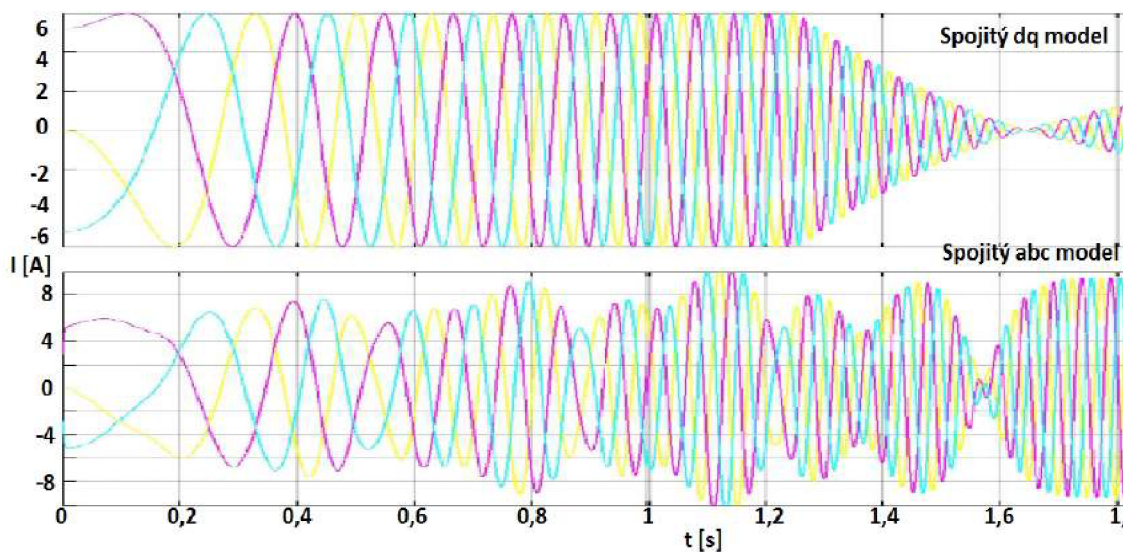
Srovnání diskrétního i spojitého průběhu abc modelu s modelem v dq souřadnicích je vidět v následujícím grafu. Průběhy se alespoň přibližně drží společně s dq modelem pouze prvních pár vteřin, pak se již od požadovaného průběhu vzdalují.





Graf 3: Srovnání úhlové rychlosti modelů v dq a v abc souřadnicích

Srovnání transformovaných dq proudů a proudů abc modelu pak vypadá jako na grafu 4. Tyto průběhy dávají tušit, že stavové rovnice pro proudy nebudou špatně namodelované, ale v amplitudách proudům dq modelu přesto neodpovídají.



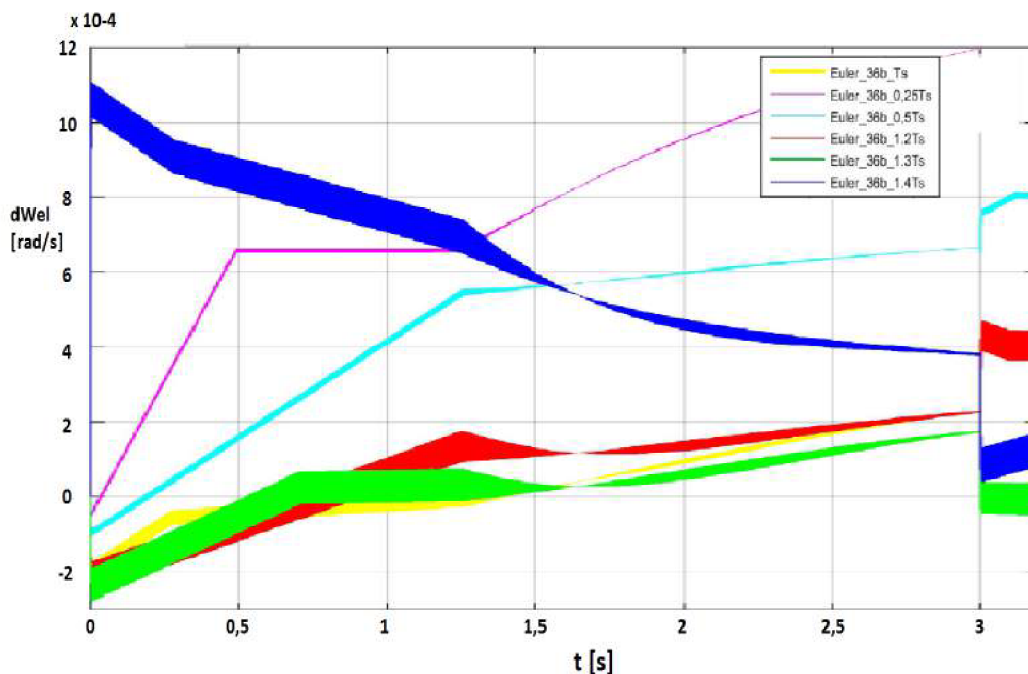
Graf 4: Srovnání proudů modelů v dq a v abc souřadnicích

## 7.3 Volba periody vzorkování

Problematika volby periody vzorkování se může zdát na první pohled jednoduchá. Rozum velí, že čím menší periodu vzorkování zvolím, tím přesněji se bude výsledek podobat průběhu spojitému. Vzhledem k omezené přesnosti mezivýpočtů v modelu to však pro příliš malé hodnoty přestává platit. Zaokrouhlovací chyba pak převáží chybu vzorkovací. Budu se tedy v této části práce snažit najít ideální kompromis, při kterém bude docházet k co nejmenší chybě.

K testování volby periody vzorkování jsem si zvolil model vypočtený Eulerovou diskretizací. Přesnosti proměnných jsem u všech výpočtů nastavil na 36-bitová čísla a poté na 54-bitová čísla (volba rozsahů bude popsána v následující kapitole) a zkoumal jsem hodnoty okolo předpokládané ideální hodnoty periody vzorkování  $T_s = 1$  us.

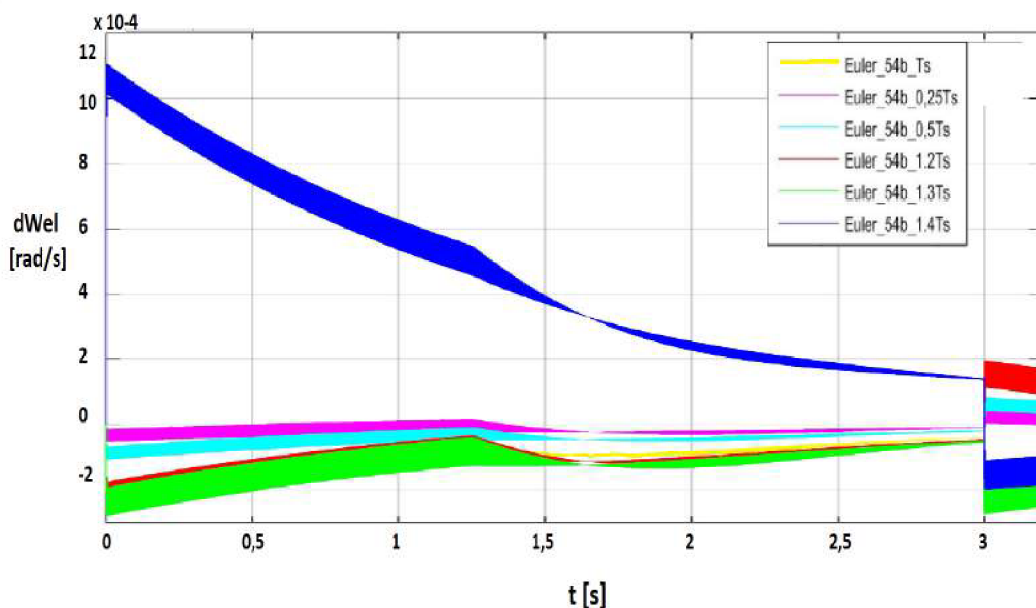
Přesnost jednotlivých modelů vyhodnocuji podle velikosti rozdílu elektrických otáček diskrétního modelu od elektrických otáček spojitého modelu. Spojitý model je u všech testování řízen obdélníkovým signálem pro žádanou hodnotu elektrických otáček o amplitudě 50 rad/s a periodě 6 sekund, to znamená, že po každých 3 sekundách dojde ke skokové změně žádané hodnoty.



Graf 5: Chyba úhlové rychlosti při různých hodnotách  $T_s$ , 36-bitové rozsahy

Je vidět, že rozdíly v chybách jsou opravdu malé v řádech  $10^{-4}$  a menších. Obdobně situace vychází i pro 54-bitové rozsahy. Při volbě většího rozpětí testovaných period už průběhy z důvodu přetečení proměnných v mezivýpočtech modelu neodpovídaly.

Jediným rozdílem v těchto dvou grafech je schopnost na přesnějších 54-bitových rozsazích lehce zpřesnit výsledek volbou menších period vzorkování než je  $1 \mu\text{s}$ .



Graf 6: Chyba úhlové rychlosti při různých hodnotách  $T_s$ , 54-bitové rozsahy

Z grafů je patrné, že pro aktuální nastavení proměnných je nejvhodnější perioda vzorkování okolo  $1 \mu\text{s}$  až  $1,3 \mu\text{s}$ . Větší periody vzorkování nedostatečně přesně sledují spojitý průběh modelu a menší hodnoty už vykazují větší chybu způsobenou nastavenými rozsahy výpočtů.

Samozřejmě by se nastavení Fixed pointů dalo změnit a upřednostnit tak výpočty na jiné vzorkovací periodě, pro mou práci jsem ale jako nejvhodnější zvolil právě toto nastavení proměnných a tedy periodu vzorkování  $T_s = 1 \mu\text{s}$ .

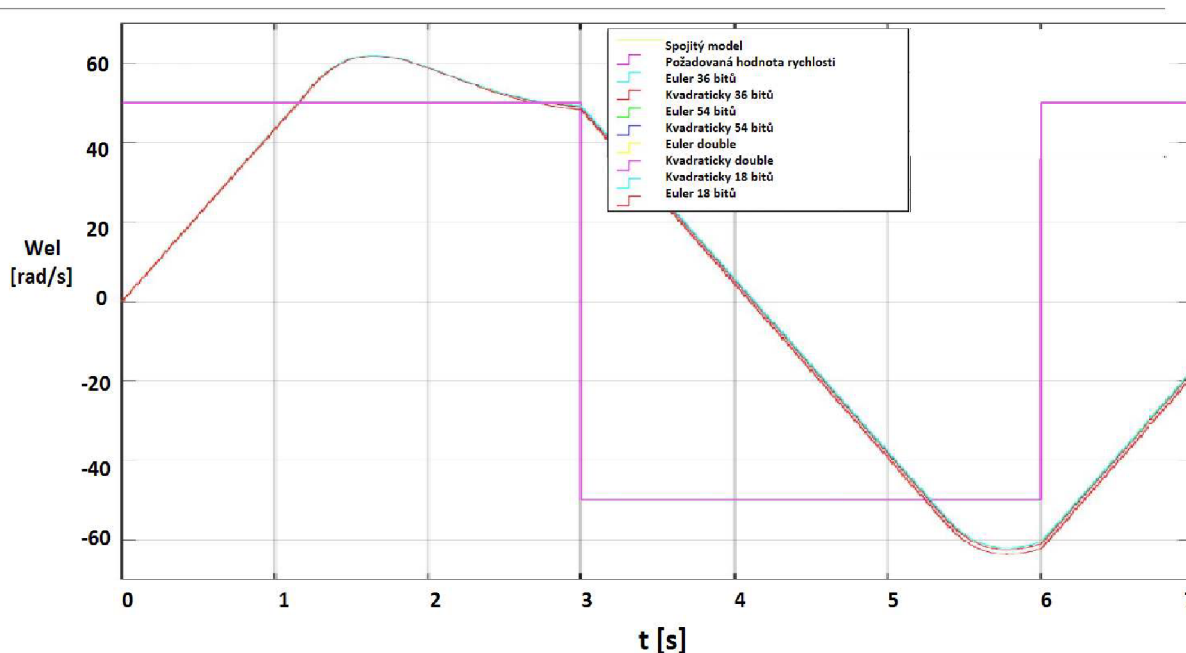
## 7.4 Nastavení Fixed Pointů

Pro správný převod do HDL kódu musí mít model v každém místě přesně nadefinované rozsahy proměnných. K tomu v Simulinku slouží nastavování Fixed pointů. Je třeba vhodně zvolit bitový rozsah všech mezivýpočtů a správně v každém bodě rozdělit celou a desetinnou část čísel pro požadovanou funkčnost celé aplikace.

Volba rozsahů má velmi velký vliv na využití výpočetního výkonu CompactRIA. Jedna násobička v jeho hradlovém poli pracuje vždy jen se dvěma vstupy a vždy s párem bytů (8 bitů + 1 znaménkový bit), to znamená dohromady 18 bitů.

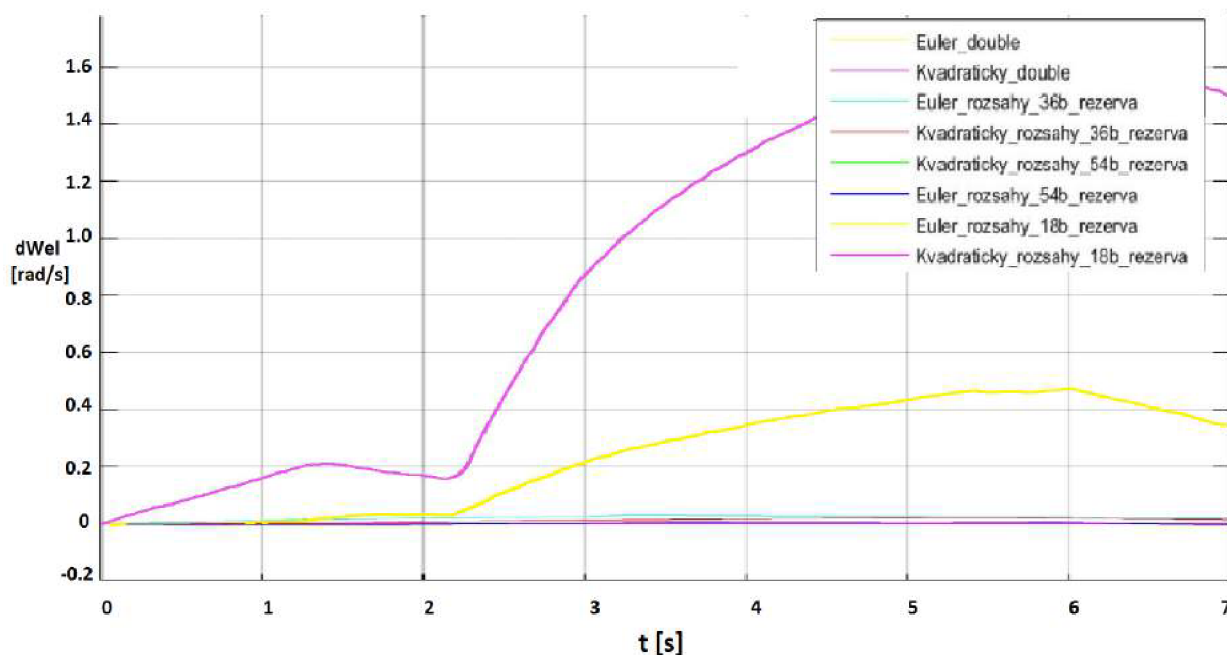
Pokusím se tedy nyní v Matlabu otestovat vliv na přesnost modelů při zkoušení násobků těchto 18 bitů (18, 36 a 54 bitů). Testování přesnosti provádím pro oba způsoby diskretizace modelu PMSM motoru v dq souřadnicích.

Rozdělení bitového rozsahu na desetinnou a celou část provádím zvlášť v každém kroku výpočtu modelu s jistou rezervou podle velikosti průchozího signálu.



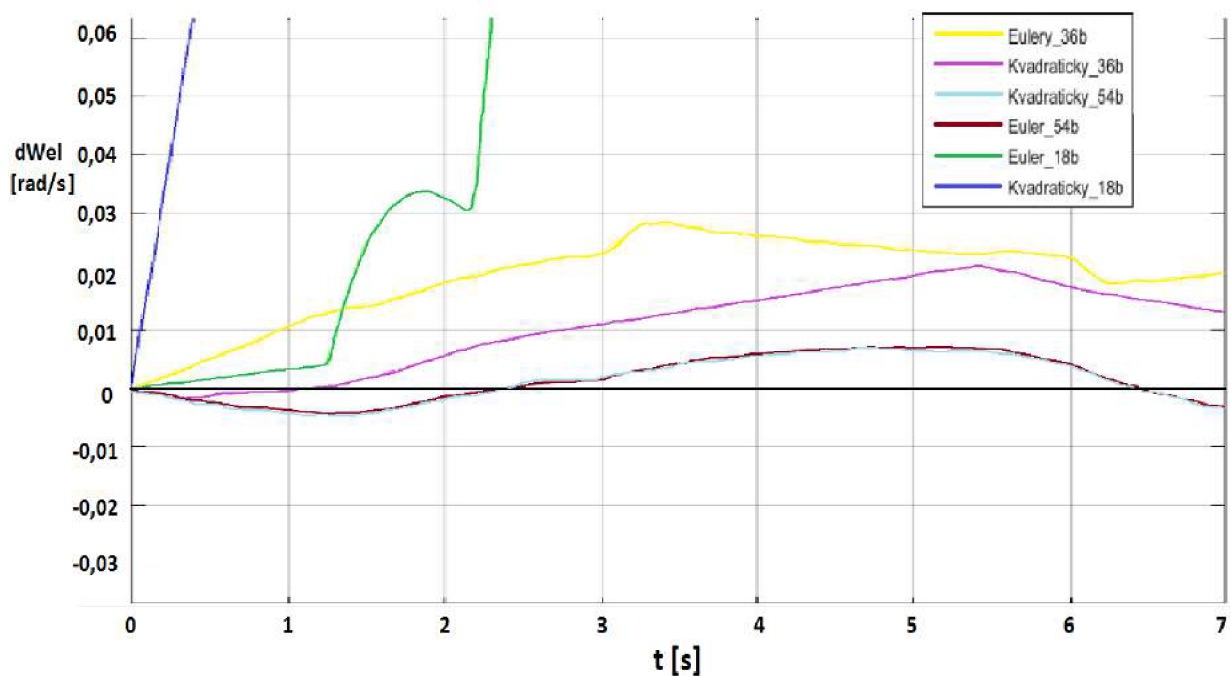
Graf 7: Testování Fixed pointů - náhled

Z náhledu na průběhy elektrické úhlové rychlosti vidíme, že všechny druhy modelů přibližně odpovídají modelu spojitému. Abych lépe rozpoznal chyby jednotlivých způsobů, budu nyní odčítat úhlovou rychlost diskretních modelů od úhlové rychlosti spojitého modelu.



Graf 8: Testování Fixed pointů – chyby diskretních modelů

Na tomto grafu jsou patrné výrazné chyby u modelů s 18-bitovými rozsahy, které jsou řádově vyšší než u ostatních modelů, proto si situaci ostatních modelů ještě přiblížím.



Graf 9: Testování Fixed pointů – chyby diskretních modelů 2

Zatímco u 18-bitových rozsahů se chyba pohybovala v řádech až několika procent, u 36-bitových je tato chyba více než desetkrát menší. Modely s 54-bitovými rozsahy jsou pak ještě přesnější, ale nejde už o tak výraznou změnu jako u předešlého srovnání.

Hodnoty chyb jednotlivých variant uvádím ještě v následující tabulce. Jedná se o chybu vyhodnocenou pouze z prvních 7 vteřin průběhu úhlové rychlosti, hodnoty tedy slouží pouze jako orientační srovnání jednotlivých metod.

<b>Pojmenování modelu</b>	<b>Největší chyba [rad/s]</b>	<b>Procentuální chyba [%]</b>
Euler 18 bitů	0,49	0,294
Euler 36 bitů	0,028	0,017
Euler 54 bitů	0,007	0,004
Kvadratický 18 bitů	1,72	1,032
Kvadratický 36 bitů	0,021	0,013
Kvadratický 54 bitů	0,007	0,004

Tab. 4: Chyby úhlové rychlosti při různých rozsazích diskretních modelů

Z procentuálních chyb je patrné, že rozdíl v přesnosti mezi kvadratickou a Eulerovou diskretizací není tak zásadní, aby převážil nevýhodu podstatně složitějšího modelu. Jako nejvhodnější se tedy jeví Eulerovou metodou diskretizovaný model s Fixed pointy nastavenými na rozsah 36 bitů. Rozdíly ve výpočetní náročnosti jednotlivých modelů vyčíslím v následující kapitole zabývající se realizací úlohy v programu Labview.

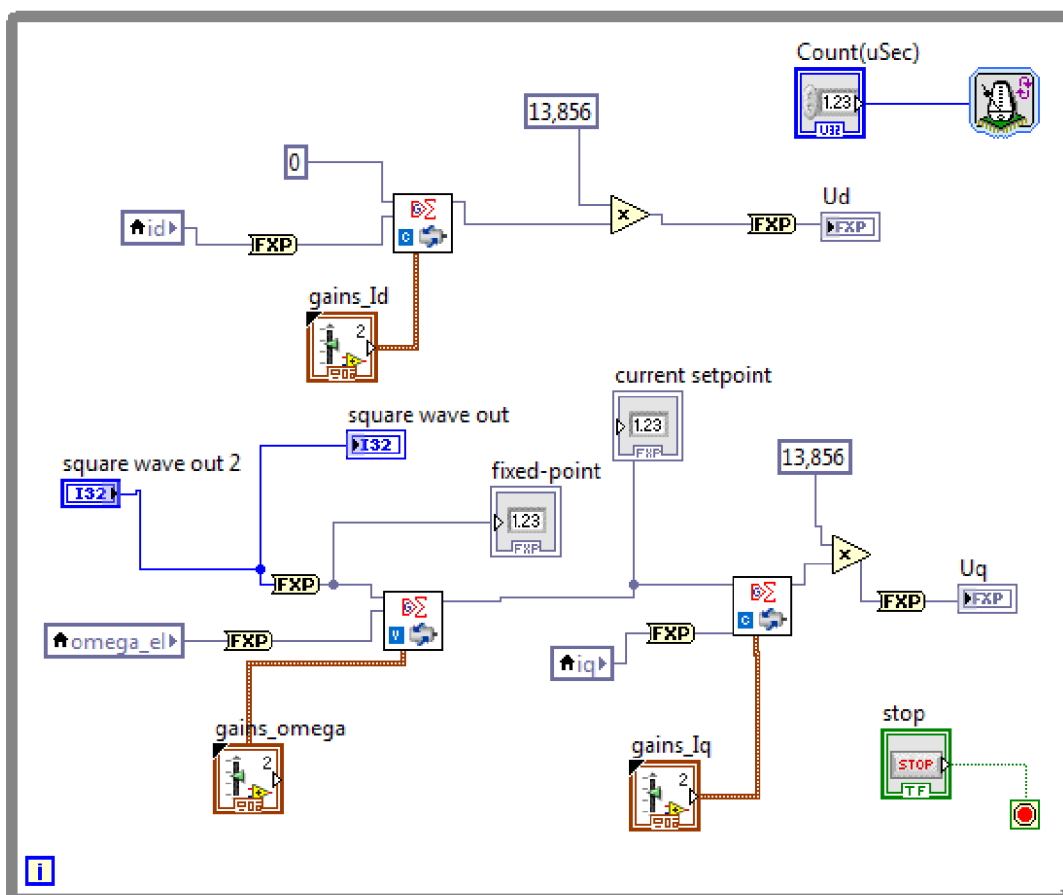
## 8 REALIZACE LABVIEW

V této kapitole popíši program vytvořený v programu Labview pro nahrání modelu a jeho řízení do hradlového pole CompactRIA a pro jeho následné ovládání a testování.

Budu se zde zabývat i problematikou zatížení CompactRIA a na jeho základě volbou vhodného finálního modelu pro testování s reálným motorem. Popíši také implementaci Parkovy a Clarkové transformace, abych mohl v testovacím programu pracovat s proudy a napětími v abc i dq souřadnicích.

### 8.1 FPGA část s modelem a řízením

Po správném nastavení projektu a spárování programu a CompactRIA vytvořím v FPGA části projektu souvor VI, do kterého nahraji testovaný model a vytvořím zde vektorové řízení, abych mohl zkoušet přímo v CompactRIU funkčnost těchto modelů.



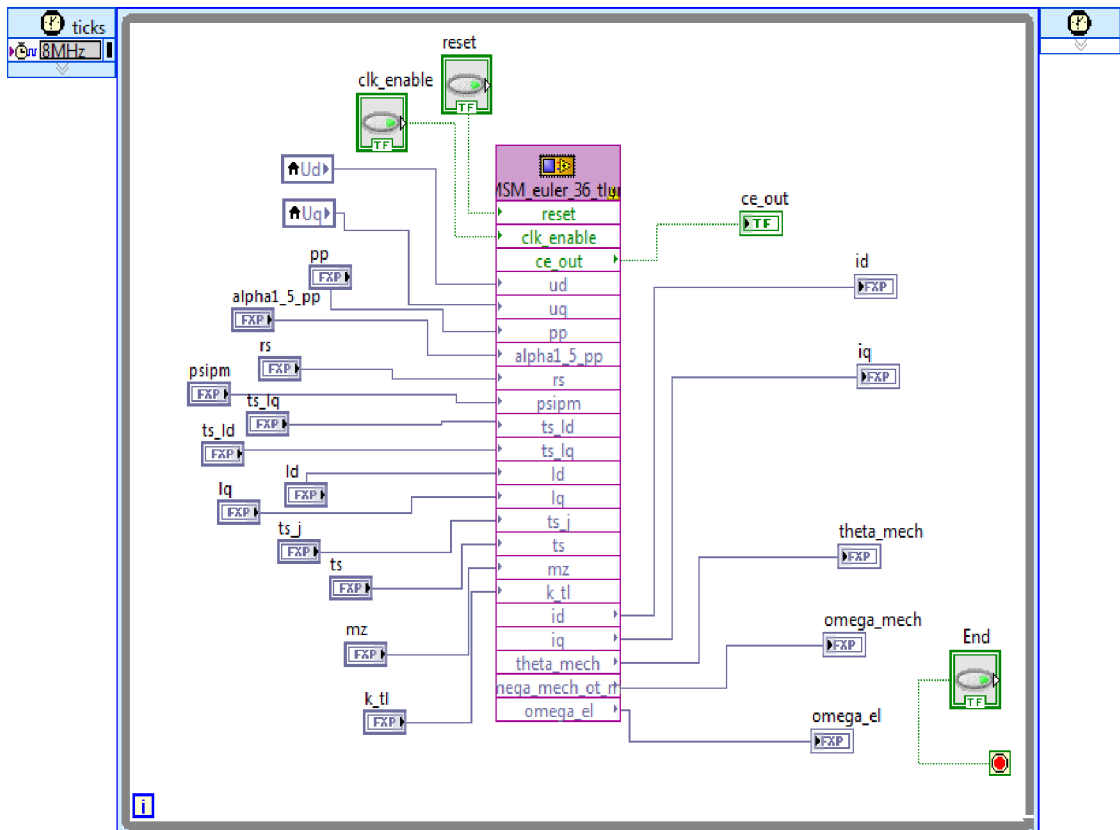
Obr. 27 : VI soubor v FPGA, část s vektorovým řízením

Na předchozím obrázku je ukázána první část VI souboru nahraného v FPGA s vektorovým řízením pro testovaný model. Celé řízení je umístěno uvnitř časované smyčky, jejíž perioda je nastavována v mikrosekundách přímo za běhu programu proměnou Count(uSec).

Vektorové řízení sestavuji z předprogramovaných PI regulátorů proudů a otáček, které lze nalézt v knihovně funkcí NI SoftMotion. Aplikace vektorového řízení ze Simulinku převodem přes HDL kód by příliš zvýšila výpočetní náročnost úlohy.

Stejně jako u vektorového řízení v Simulinku řídím proud  $i_d$  na nulovou hodnotu, aktuální hodnoty proudu  $i_q$  a úhlové rychlosti pak vyčítám v každém kroku z modelu motoru. Výstupní hodnoty PI regulátorů proudu mají na výstupu napětí v rozsahu pouze od -1 V do 1 V, proto tyto hodnoty na výstupu regulátorů ještě pronásobuji DC bassem o hodnotě  $\frac{24}{\sqrt{3}} = 13,856$  a teprve pak tyto hodnoty posílám jako vstupní proměnné do modelu PMSM motoru.

Nastavování konstant regulátorů i žádané hodnoty úhlové rychlosti provádím přímo v běžícím programu, abych tak mohl testovat různé možnosti řízení.

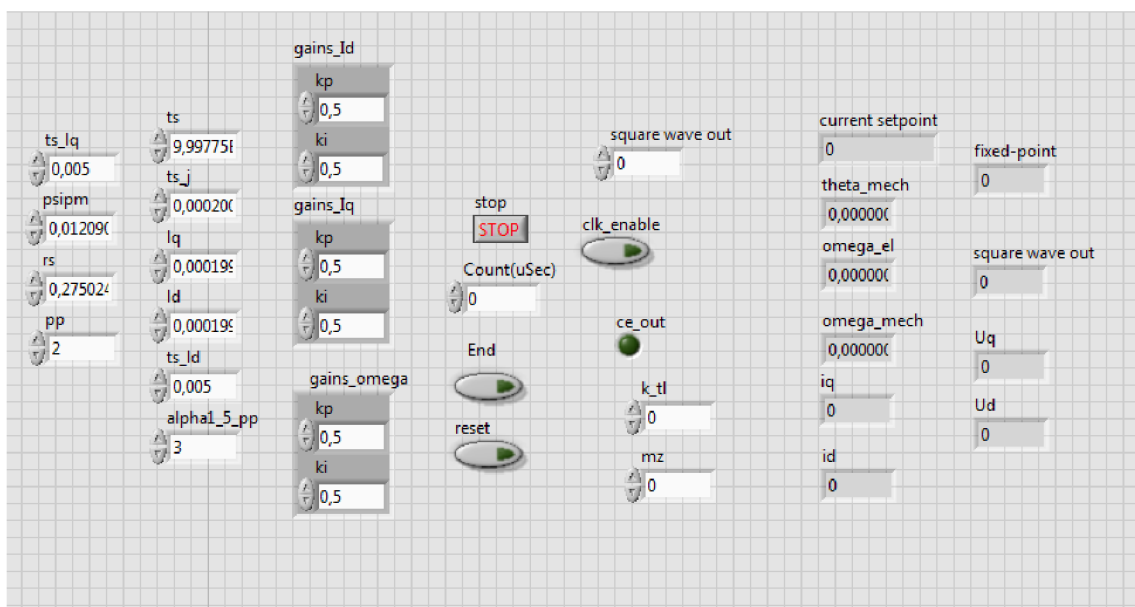


Obr. 28 : VI soubor v FPGA, část s modelem PMSM motoru



V druhé části VI souboru nahraného v FPGA je opět v časované smyčce, tentokrát nastavené přímo na 8MHz, model motoru v HDL kódu nahraný v IP Nodu. Tento IP Node má vyvedeny na tlačítka hodnoty resetu a clk\_enable a jsou v něm správně nastavené všechny vstupní i výstupní proměnné a především jejich bitové rozsahy celé i desetinné části.

Veškeré konstanty motoru jsou nastavovány až během chodu programu, abych mohl testovat jejich případnou změnu. Výstupní hodnoty pak posílám do VI nahraného v real-time části CompactRIA, kde probíhá jejich vykreslování do grafů, tuto část popíší v dalších kapitolách.



Obr. 29 : VI soubor v FPGA, ovládání

Ovládání tohoto VI souboru je rozděleno přehledně na čtyři části. Zleva se jedná o nastavování konstant motoru, v tomto případě shodně nastavených jako při simulaci v Simulinku, dále nastavování regulátorů proudů a otáček, řízení simulace s nastavováním požadovaných hodnot otáček, tlumení a zátěžného momentu a úplně vpravo je pak umístěno zobrazování výsledků simulace.

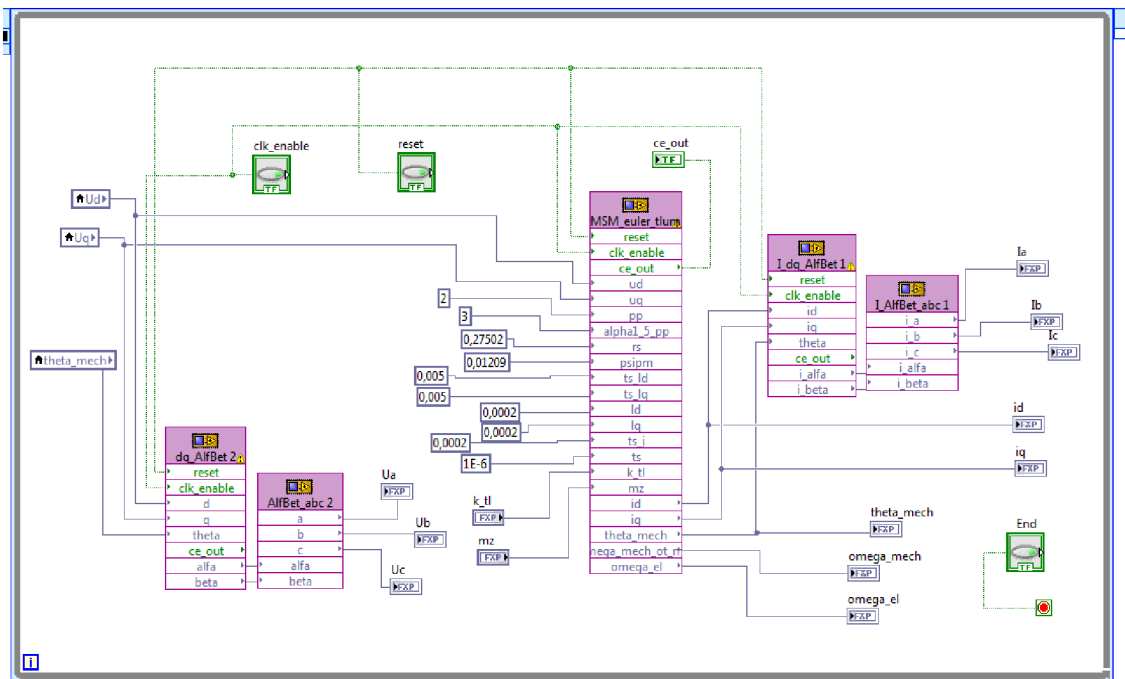
Výsledky simulace modelu tímto programem jsou k vidění v kapitole RT testování modelu.

## 8.2 Transformace souřadného systému v LabVIEW

Abych mohl sledovat aktuální hodnoty proudů i napětí v abc i dq souřadném systému, přidám nyní k programu v FPGA ještě transformace mezi těmito souřadnými systémy. Mimo sledování hodnot v různých souřadných osách mi tento krok umožní i testovat modely v jiném než dq systému.

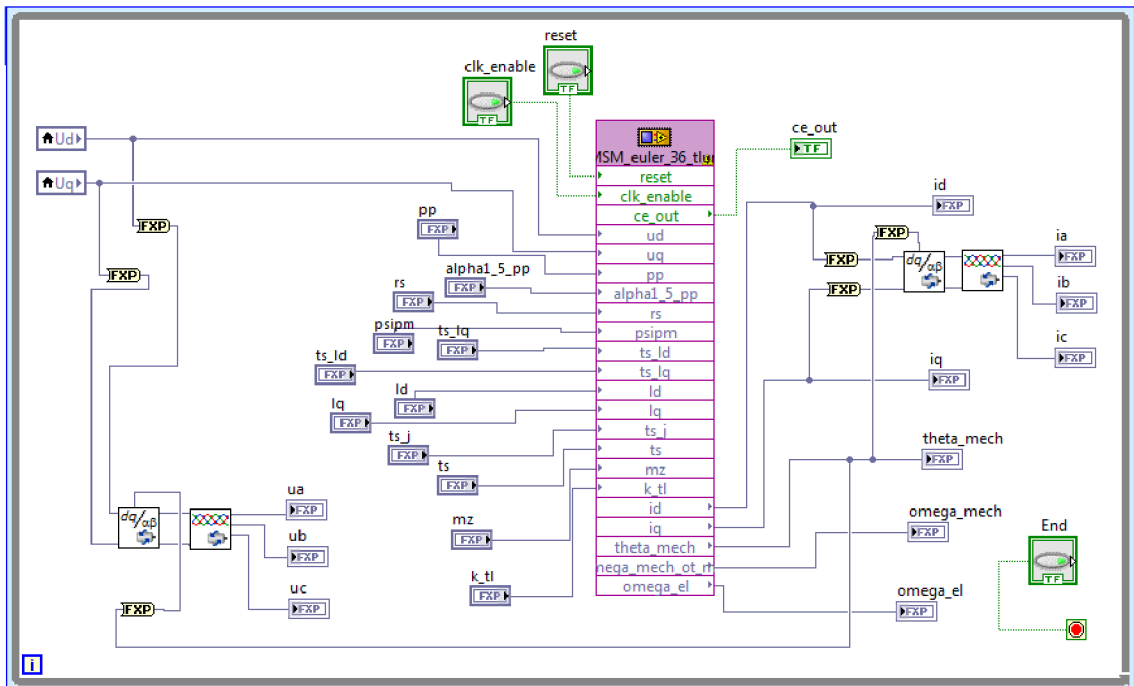
Nabízí se opět dvě varianty pro tvorbu těchto transformací, převést vytvořené transformace ze Simulinku nebo použít již připravené transformace z knihovny funkcí NI SoftMotion. Vzhledem k přítomnosti trigonometrických funkcí v těchto transformacích by převod přes HDL kód znamenal výrazné zpomalení programu.

U trigonometrických funkcí přímo v LabVIEW navíc mohu nastavovat počet cyklů v kolika se výpočet provede a tedy počet pipeline potřebných k tomuto přepočtu, jak je popsáno na začátku této práce v kapitole Implementace funkce sinus, v Přípravných úlohách.



Obr. 30 : Transformace souřadných systémů v LabVIEW, pomocí IP Nodů

Na obrázku 30 je vidět část VI v FPGA s transformacemi převedenými ze Simulinku. Na obrázku 31 pak stejný program s transformacemi použitými z knihovny NI SoftMotion.



Obr. 31 : Transformace souřadných systémů v LabVIEW, knihovna NI SoftMotion

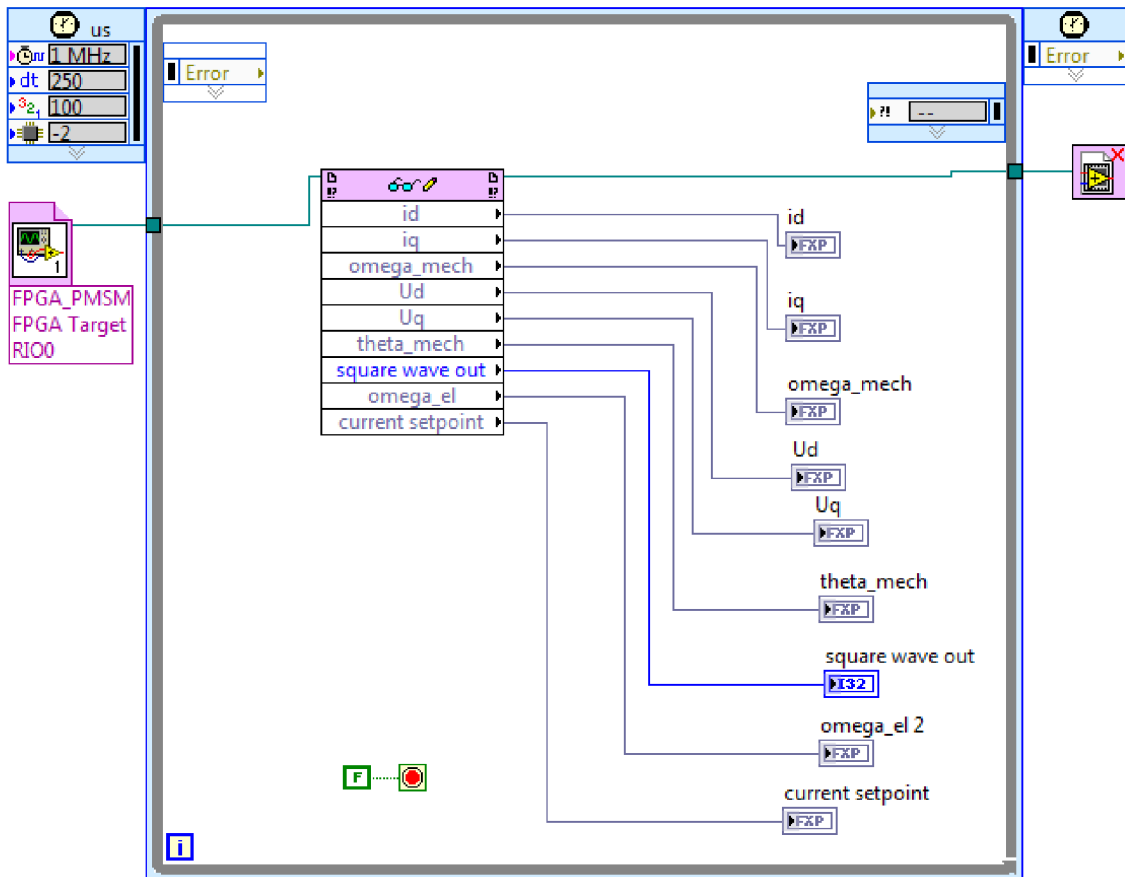
Z knihovny NI SoftMotion jsem zde použil bloček inverzní Parkovy transformace pro převod z  $dq$  do  $\alpha\beta$  systému a poté Space Vektor Modulation SVM pro převod z  $\alpha\beta$  do  $abc$  souřadnic. Transformace provádím jak pro vstupní napětí, tak i pro výstupní proudy.

Výsledky těchto transformací jsou zobrazeny společně s ostatními výsledky RT testování v LabVIEW v další kapitole RT testování modelu.

## 8.3 RT testování modelu

Vedle VI souboru nahraného v FPGA části mám v projektu ještě VI v real-time části projektu, které mi slouží k testování modelu a k vykreslování jednotlivých proměnných do grafů.

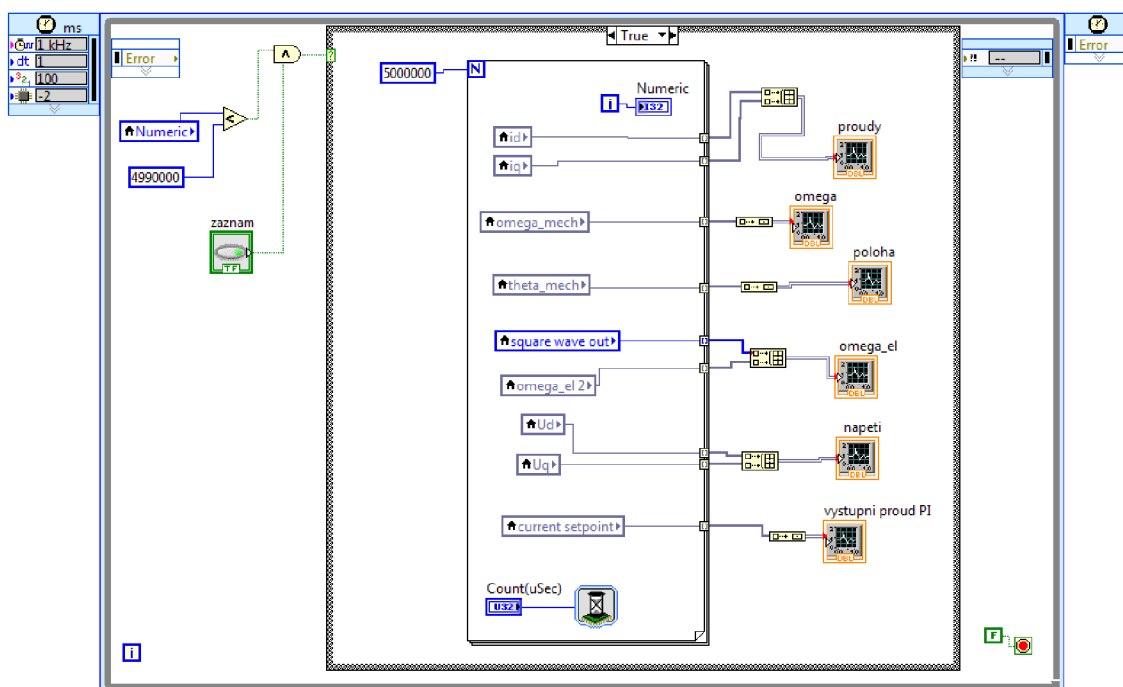
V jeho první části vyčítám aktuální hodnoty proměnných z hradlového pole v časované smyčce. V bločku FPGA Target nastavím soubor, z kterého chci proměnné sbírat, a postupně si všechny požadované povytahuji do proměnných v tomto VI.



Obr. 32 : RT VI pro testování modelu a vykreslování grafů, první část

Tyto proměnné pak posílám do druhé časované smyčky v tomto programu, kde je vykresluji do grafů.

Vnitřní smyčkou této části programu je for cyklus vykonávající nastavený počet kroků v závislosti na délce těchto kroků udaných v  $\mu\text{s}$  v proměnné Count(uSec), v této smyčce se zaznamenává určitý časový úsek hodnot proměnných. Okolo for cyklu je další smyčkou while cyklus, ten je v hodnotě true, dokud neskončí vnitřní for cyklus a dokud je aktivní tlačítko záznamu. V tomto cyklu jsou naměřená data vkládána do datových řad a vykreslována v grafech.



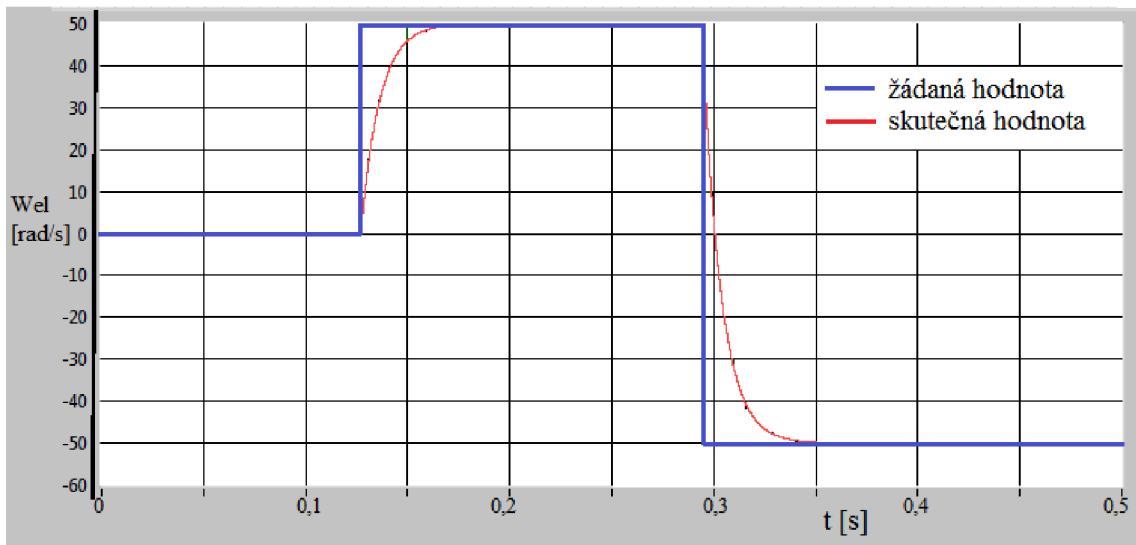
Obr. 33 : RT VI pro testování modelu a vykreslování grafů, druhá část

Ve Front panelu pak není nic jiného než nastavování času jednoho cyklu for smyčky, tlačítko záznamu a samotné grafy pro vybrané proměnné modelu.

Ukáži zde výsledky vybraných grafů kvadraticky diskretizovaného modelu a modelu s Eulerovou diskretizací spolu s transformacemi proudů do abc systému.

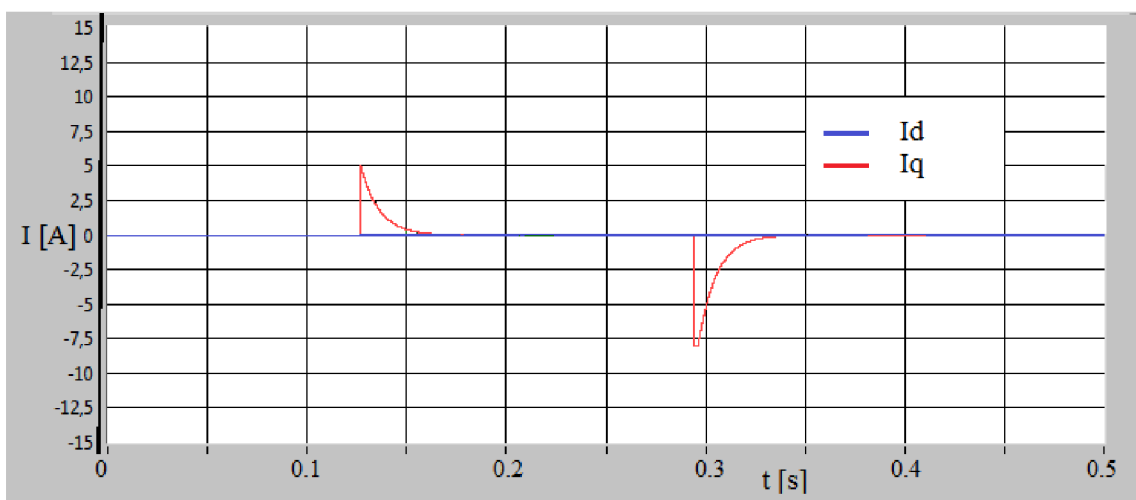
U obou modelů nastavuji skokové změny žádané hodnoty elektrických otáček a sleduji hodnoty jednotlivých proměnných.

Průběh regulovaných elektrických otáček kvadraticky diskretizovaného modelu sledujících změny požadované hodnoty je ukázán v grafu č. 10.



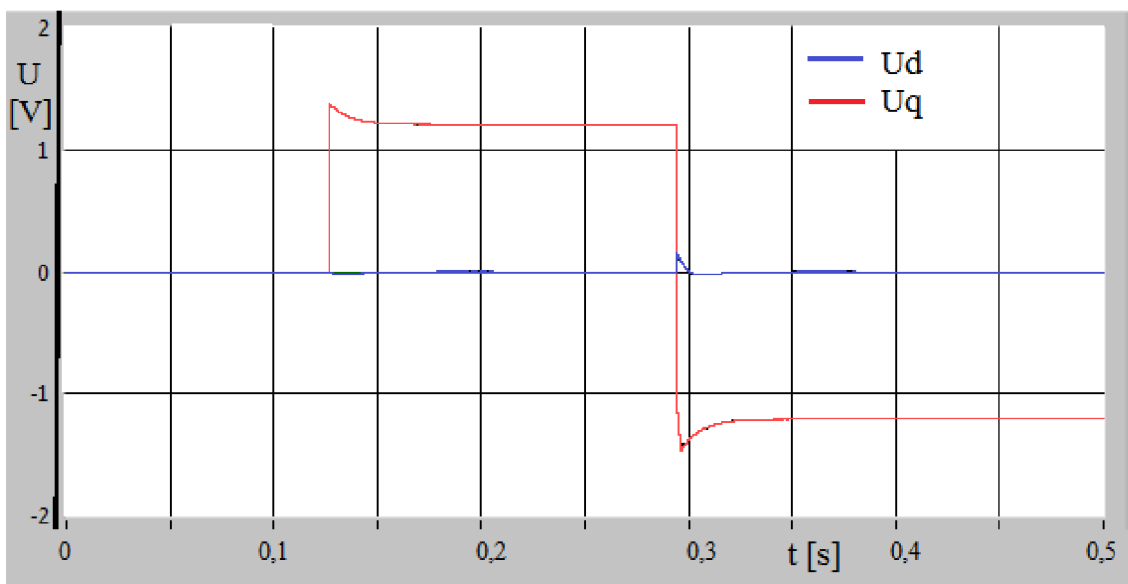
Graf 10: RT testování kvadratického modelu – graf elektrických otáček

Pro stejný model i otáčky je na dalším grafu ukázán průběh proudů  $i_q$  a  $i_d$  na výstupu testovaného modelu. Proud  $i_d$  je neustále regulován okolo nulové hodnoty, proud  $i_q$  pak reaguje na změny žádané hodnoty otáček v závislosti na velikosti této změny.



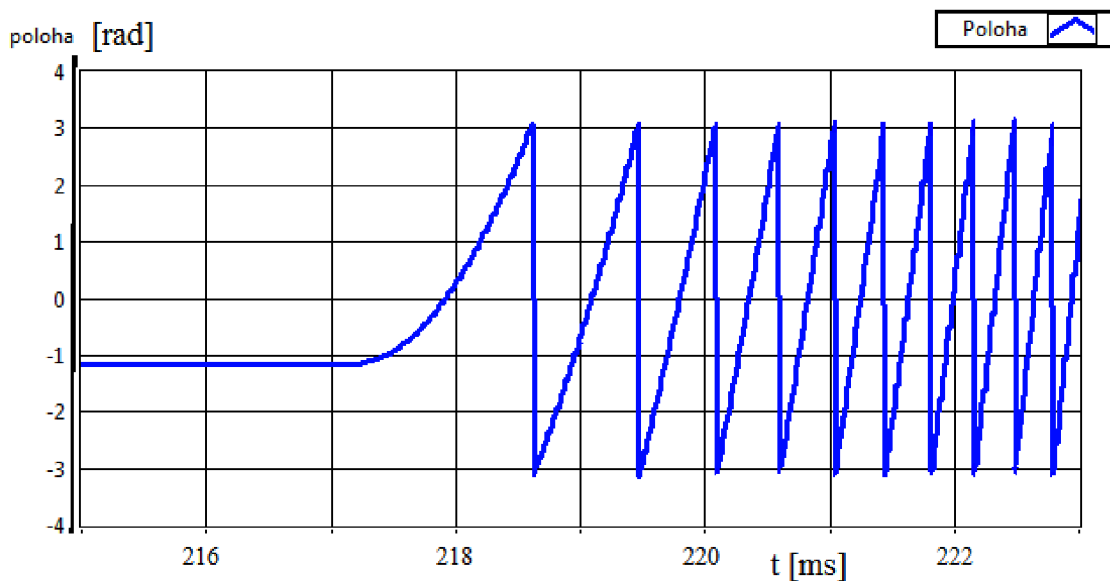
Graf 11: RT testování kvadratického modelu – graf proudů

Na tyto hodnoty reagují PI regulátory proudů. Výstupní hodnoty napětí  $u_d$  a  $u_q$  z těchto regulátorů, které vstupují do kvadratického modelu PMSM motoru, jsou ukázány na další straně.



Graf 12: RT testování kvadratického modelu – graf napětí

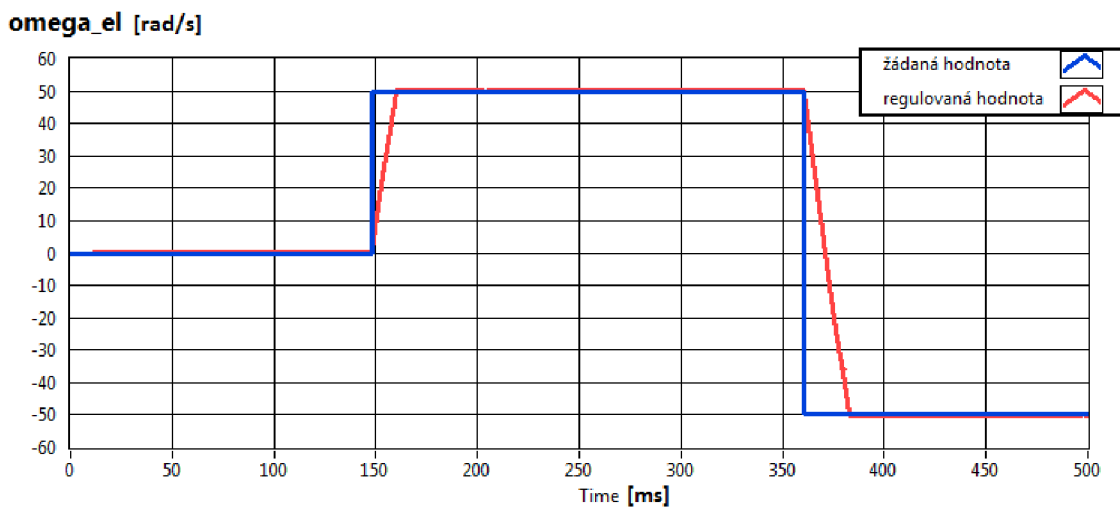
Nakonec pro tento model ukáží výřez z grafu aktuální polohy motoru při rozjezdu tohoto modelu z klidové polohy.



Graf 13: RT testování kvadratického modelu – výřez z grafu polohy

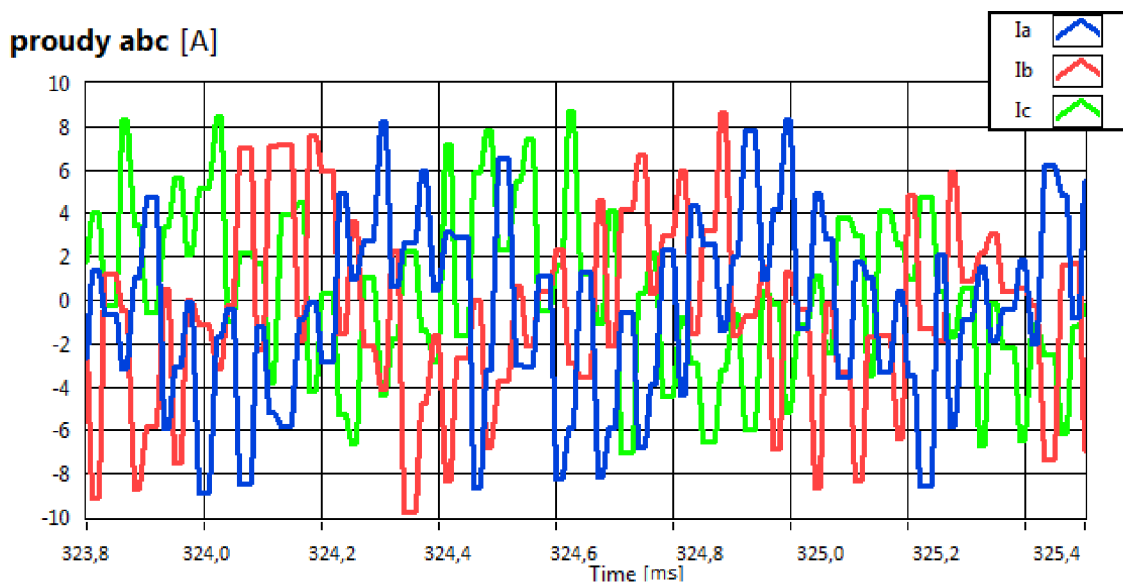
Obdobně jsem testoval i ostatní vytvořené modely s různými hodnotami konstant regulátorů a s různými tvary požadovaných hodnot otáček.

Na další straně je graf elektrických otáček modelu diskretizovaného Eulerovou metodou s 36-bitovými rozsahy proměnných, kde hodnota otáček modelu sleduje skokové změny žádané hodnoty.



Graf 14: RT testování Eulerova modelu – graf elektrických otáček

Posledním zobrazeným grafem je průběh proudů v abc souřadném systému na výstupu modelu PMSM motoru získaný transformacemi z dq souřadného systému. Na grafu je zobrazen pouze krátký časový úsek tohoto průběhu, na kterém je patrné vzájemné rozkmitání těchto proudů při konstantní hodnotě elektrických otáček 50 rad/s způsobené pravděpodobně nepřesným výpočtem aktuální polohy motoru, potřebné pro výpočty transformací.



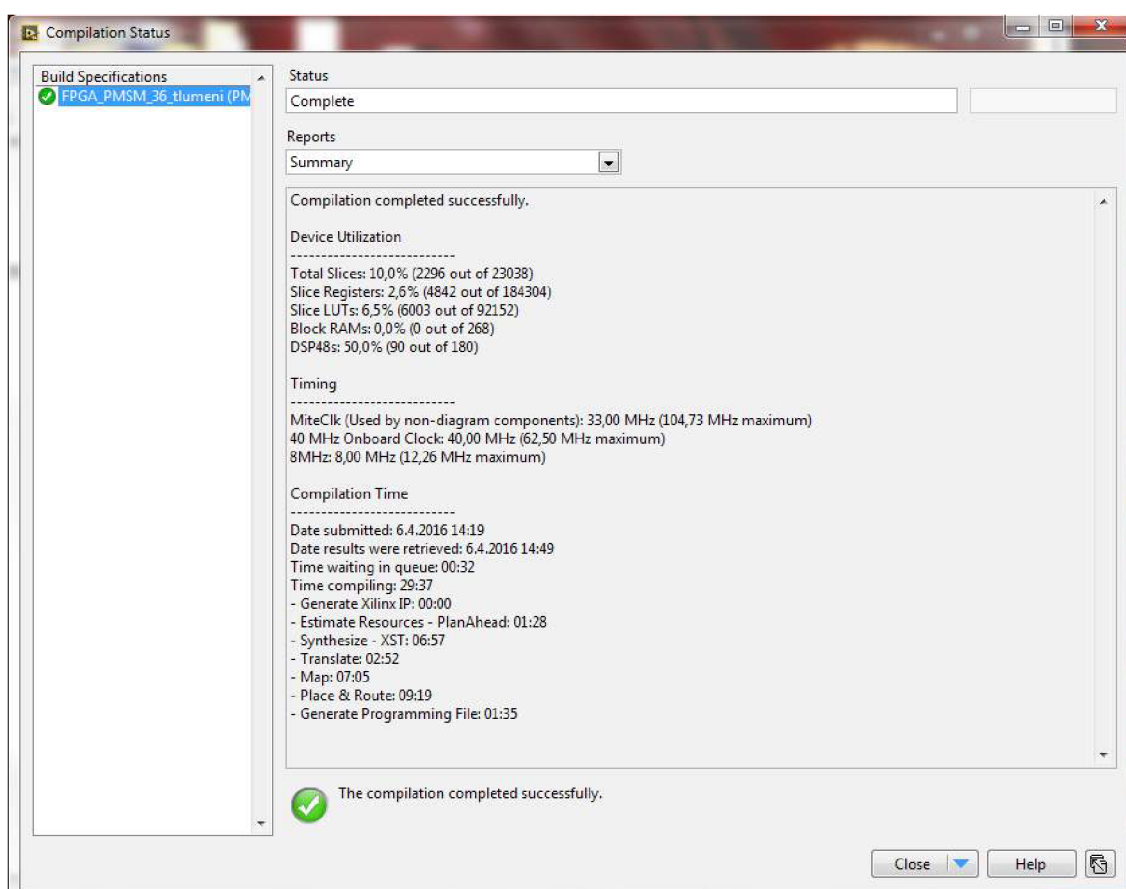
Graf 15: RT testování kvadratického modelu – graf transformovaných proudů  $i_{abc}$



## 8.4 Srovnání zatížení CompactRIA

Po zkompilování a nahrání programu do FPGA CompactRIA se objeví shrnutí kompilace a všechny statistiky nahraného kódu. Z těchto statistik pak lze jednoduše zjistit celkové zatížení CompactRIA nahranou aplikací.

Na shrnující obrazovce kompilace nalezneme vytižení hradlového pole, údaje o použitých hodinových signálech a na závěr časy jednotlivých částí kompilace. Ukázka jedné úspěšně dokončené kompilace je zde na obrázku 34.



Obr. 34 : Výpis shrnujících informací o proběhlé kompilaci

Kompilace programu se skládá z několika částí. První z nich je předplánování, tedy odhad zdrojů, které budou pro danou úlohu využity. Dalšími kroky jsou syntéza Xilinx Synthesis Technology (XST), která zpracovává VHDL soubory a překlad celého programu před nahráním do FPGA. Zpravidla časově nejnáročnější jsou pak procesy mapování a umístění s propojováním hradlového pole. Na závěr se ještě generují programovací soubory a statistiky kompilace.

U statistik vytížení CompactRIA pak jde především o počty a procenta využitých logických elementů (Slices) hradlového pole, které se dále dělí na Slice Registers pro ukládání bitových hodnot, Slice LUTs například pro funkce typu multiplexoru, Block RAM pro ukládání proměnných a DSP48s pro operace s násobením.

Nyní tedy můžeme jednotlivé modely s různými bitovými rozsahy proměnných a s různými způsoby diskretizace navzájem srovnat na základě jejich výpočetní náročnosti.

<b>Druh modelu</b>	<b>Bitový rozsah</b>	<b>Total Slices</b>	<b>Slice Registers</b>	<b>Slice LUTs</b>	<b>Block RAMs</b>	<b>DSP48s</b>
Euler-dq	18	8,1%	0,0%	0,1%	0,0%	22,8%
Euler-dq	36	10,0%	2,6%	6,5%	0,0%	50,0%
Euler-dq	54	13,7%	2,9%	9,5%	0,0%	78,9%
Kvadrat.-dq	18	10,5%	2,6%	7,2%	0,0%	49,4%
Kvadrat.-dq	36	42,6%	3,8%	37,6%	0,0%	100,0%

Tab. 5: Vytížení hradlového pole CompactRIA s různými modely

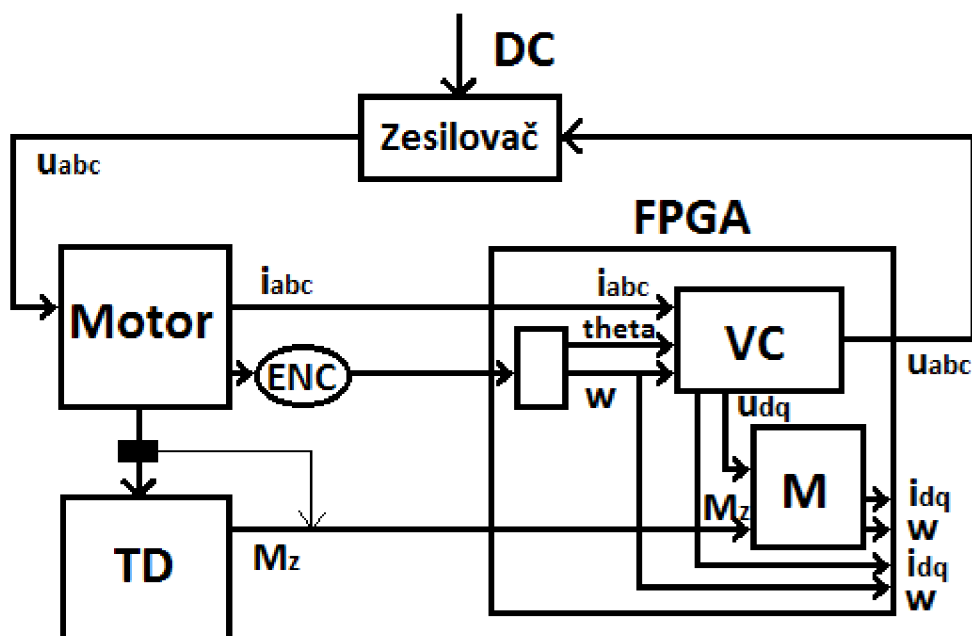
Z tabulky je dobře vidět, že zvýšení o další pár 9-bitů spotřebuje zhruba dvakrát tak velký počet násobiček zakódovaných v DSP48s. Modely získané kvadratickou diskretizací jsou výrazně náročnější a nevýhodnější vzhledem k ne výrazně přesnějším výsledkům, které poskytují. Jako nejvhodnější kompromis mezi zatížením CompactRIA a přesností diskretizovaného průběhu jsem tedy vybral model diskretizovaný Eulerovou metodou s 36-bitovými rozsahy, který jsem i použil pro testování shody modelu s reálným motorem.

## 9 POROVNÁNÍ MODELU S REÁLNÝM MOTOREM

Důležitou úlohou této práce je porovnání vytvořeného modelu s reálným PMSM motorem. Pro tento účel jsem si vybral model diskretizovaný Eulerovou metodou v dq souřadnicích i s doplněným tlumením stroje a s 36-bitovými rozsahy čísel.

### 9.1 Logické schéma porovnání modelu s reálným motorem

Na následujícím schématu je popsáno pracoviště s reálným PMSM motorem sloužící k otestování tohoto motoru s vytvořeným modelem. V hradlovém poli (FPGA) CompatRIA použitého v této úloze je uloženo vektorové řízení stroje i s příslušnými transformacemi mezi souřadnými systémy a také testovaný model. Dále na pracovišti najdeme PMSM motor propojený s tachodynamem, které dokáže simulovat různé hodnoty momentu zátěže, a také zesilovač s připojeným DC napájecím napětím.



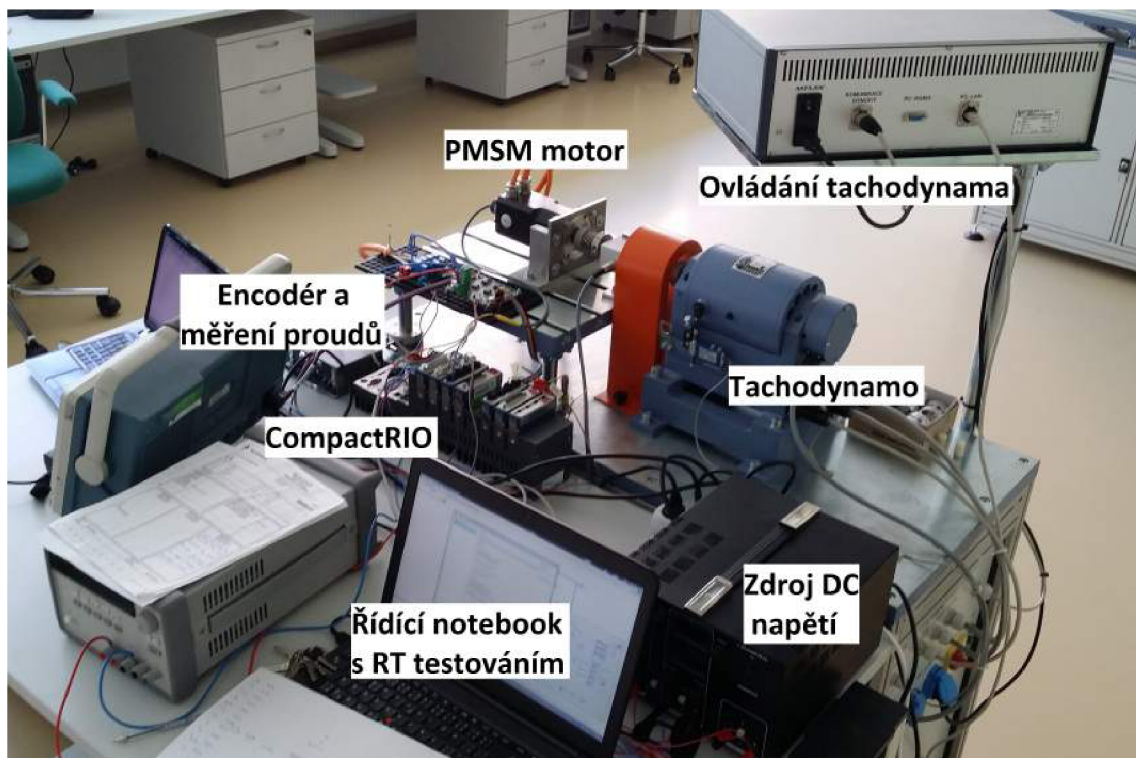
Obr. 35 : Porovnání modelu s reálným motorem – schéma zapojení

Otáčky na motoru jsou snímány enkodérem, který zároveň udává i informaci o poloze motoru. Obě tyto veličiny jsou zpracovány v FPGA a následně využity ve vektorovém řízení. Pro vektorové řízení musíme samozřejmě z motoru snímat i informaci o proudech. Výstupními napětími z vektorového řízení je řízen zesilovač udávající hodnoty vstupním napětím do motoru. Zároveň jsou tyto výstupní napětí transformovány do dq systému a posílány jako vstup do testovaného modelu.

Pro přesnější shodu motoru s jeho modelem posíláme do modelu i informaci o zátěžném momentu motoru, který i při nulové přídavné zátěži není zcela zanedbatelný. Informaci o momentu zátěže můžeme vyčíst buď přímo z displeje tachogenerátoru nebo přesněji ze snímače momentu na hřídeli motoru.

V RT části CompactRIO pak sledujeme aktuální hodnoty otáček vyčtené enkodérem na motoru a jeho přetransformované proudy v dq souřadnicích. Hodnoty těchto veličin zde porovnáváme s výstupními hodnotami otáček a proudů testovaného modelu.

Na fotografii pracoviště jsou vidět i s popisky jednotlivá zařízení použitá v této úloze. Všechna tato zařízení jsou popsána v příslušné kapitole této práce (2.2 Použitý hardware).



Obr. 36 : Porovnání modelu s reálným motorem – fotografie pracoviště s popisky

Na další fotografii je pohled na pracoviště z opačné strany, kde je lépe vidět motor s tachogenerátorem i jeho ovládáním.

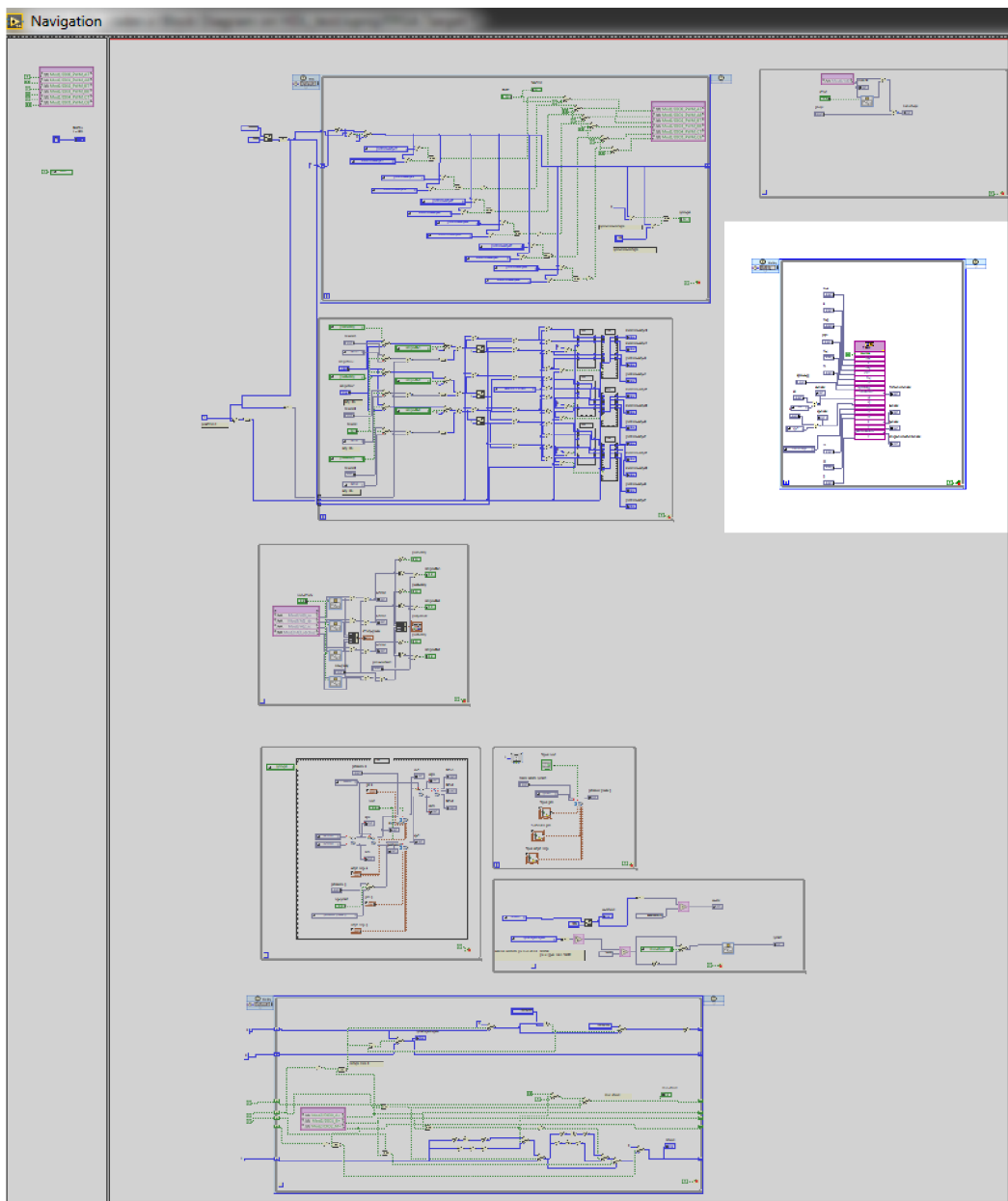


Obr. 37 : Porovnání modelu s reálným motorem – fotografie pracoviště

## 9.2 Nastavení parametrů modelu a regulátorů

Testování jsem prováděl v LabView z již připraveného programu pro ovládání reálného PMSM motoru. Do tohoto programu jsem pouze vložil část s modelem stroje, jak je vidět na obrázku na další straně, nastavil jsem všechny potřebné parametry regulátorů a modelu a testoval zda je chování PMSM motoru a jeho modelu shodné.

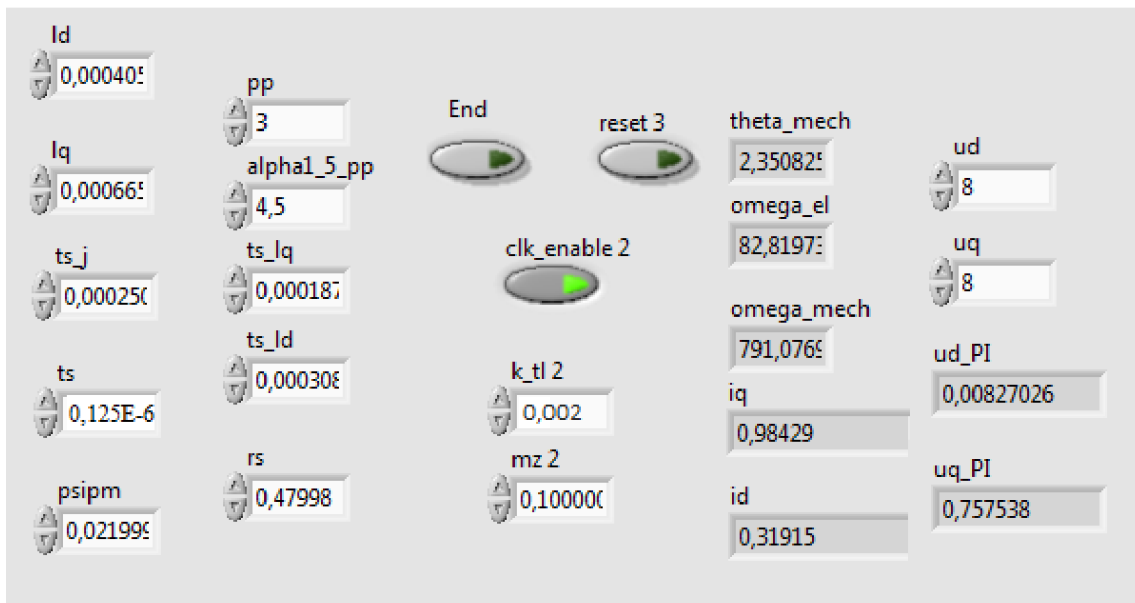
Při nastavování správných parametrů modelu pro toto testování jsem vycházel ze zkušeností vedoucího mé práce a z opakovaného experimentálního zkoušení. Regulátory proudu jsem nastavoval zvlášť, nezávisle na zbytku regulačního schématu, pomocí osciloskopu. Regulátor otáček jsem pak dolad'oval přímo podle chování reálného motoru.



Obr. 38 : Náhled testovacího VI nahraného v FPGA s modelem PMSM motoru

V první polovině Front panelu tohoto VI souboru se nastavují hodnoty všech tří regulátorů vektorového řízení a také kompenzace zátěžného momentu motoru. Můžeme zde sledovat aktuální hodnoty proudů, otáček motoru i jeho polohu. Také se odtud dá řídit žádaná hodnota otáček motoru nebo zvlášť žádané hodnoty pro regulátory proudů.

V druhé části ovládání tohoto VI souboru jsem pak prováděl nastavování parametrů modelu, jeho restartování a kontrolu hodnot jeho výstupních parametrů. Náhled na tuto část je na obrázku 39.



Obr. 39 : Ovládání testovacího FPGA VI s nastavením modelu PMSM motoru

Vedle parametrů modelu zde nastavuji také hodnoty zátěžného momentu  $M_Z$ , odhadovaného tlumení motoru  $K_{TL}$  a DC bass pro zesílení výstupních napětí regulátorů proudu. Zde uvádím všechny tyto hodnoty přehledně v tabulce 6.

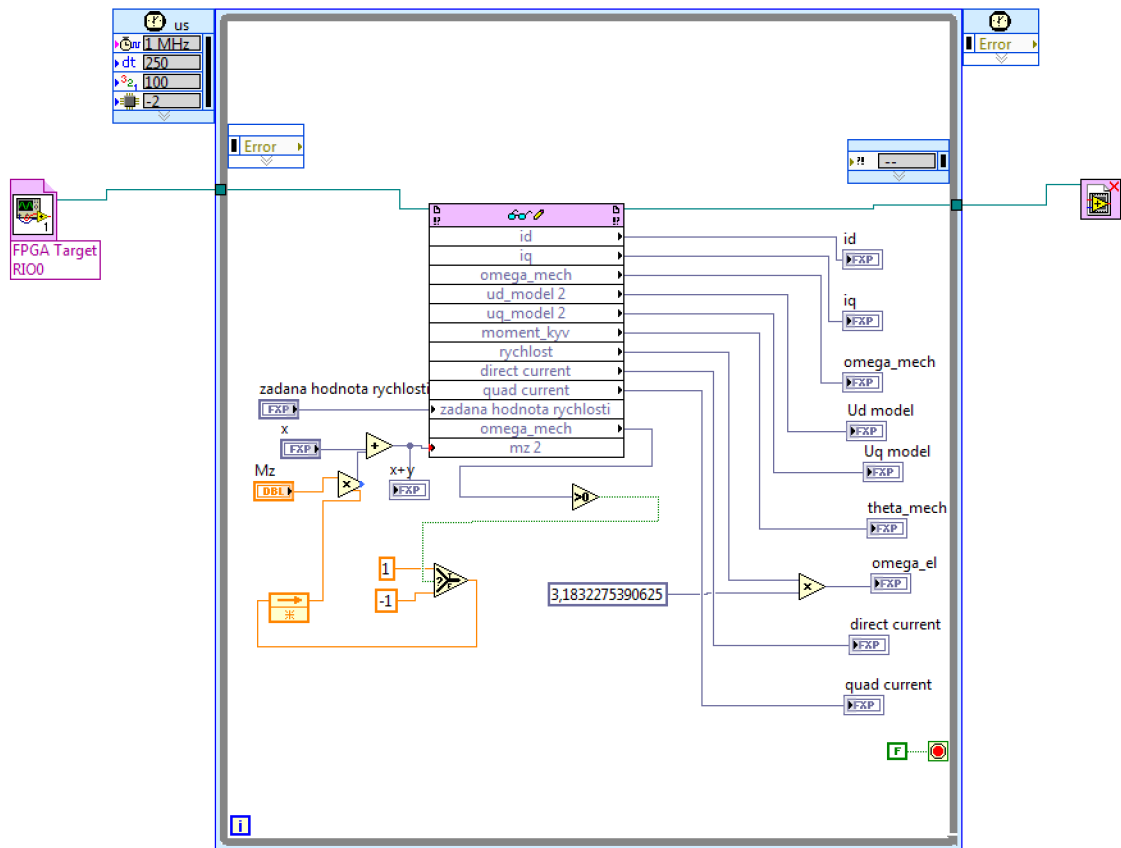
Název proměnné	Značení	Jednotka	Hodnota
Indukčnost d složky	$L_d$	[mH]	0,405
Indukčnost q složky	$L_q$	[mH]	0,665
Perioda vzorkování	$T_s$	[ $\mu$ s]	0,125
Podíl vzorkovací periody a $L_d$	$T_s/L_d$	[s/H]	0,000308
Podíl vzorkovací periody a $L_q$	$T_s/L_q$	[s/H]	0,000187
Odpor statorového vinutí	$R_s$	[ $\Omega$ ]	0,47998
Konstanta permanentního magnetu	$\Psi_f$	[Wb]	0,022
Moment setrvačnosti	$J$	[kg m <sup>2</sup> ]	0,0005
Podíl vzorkovací periody a $J$	$T_s/J$	[s/(kg m <sup>2</sup> )]	0,00025
Počet pólových dvojic	$P_p$	[-]	3
1,5 násobek počtu pólových dvojic	$1,5 \times P_p$	[-]	4,5
Zátěžný moment	$M_Z$	[Nm]	0,1
Tlumení motoru	$K_{TL}$	[kg m <sup>2</sup> s <sup>-1</sup> ]	0,002
DC bass napětí v d složce	Bass_u <sub>d</sub>	[-]	8
DC bass napětí v q složce	Bass_u <sub>q</sub>	[-]	8

Tab. 6: Nastavení hodnot modelu při testování s reálným PMSM motorem

### 9.3 Výsledky testování shody modelu a reálného motoru

Pro testování shody modelu a motoru jsem si vytvořil VI, které jsem nahrál do real-time části CompactRIA. Sem si posílám z FPGA důležité proměnné a následně je zde vykresluji do grafu. Proces sbírání proměnných z FPGA je vidět na obrázku níže.

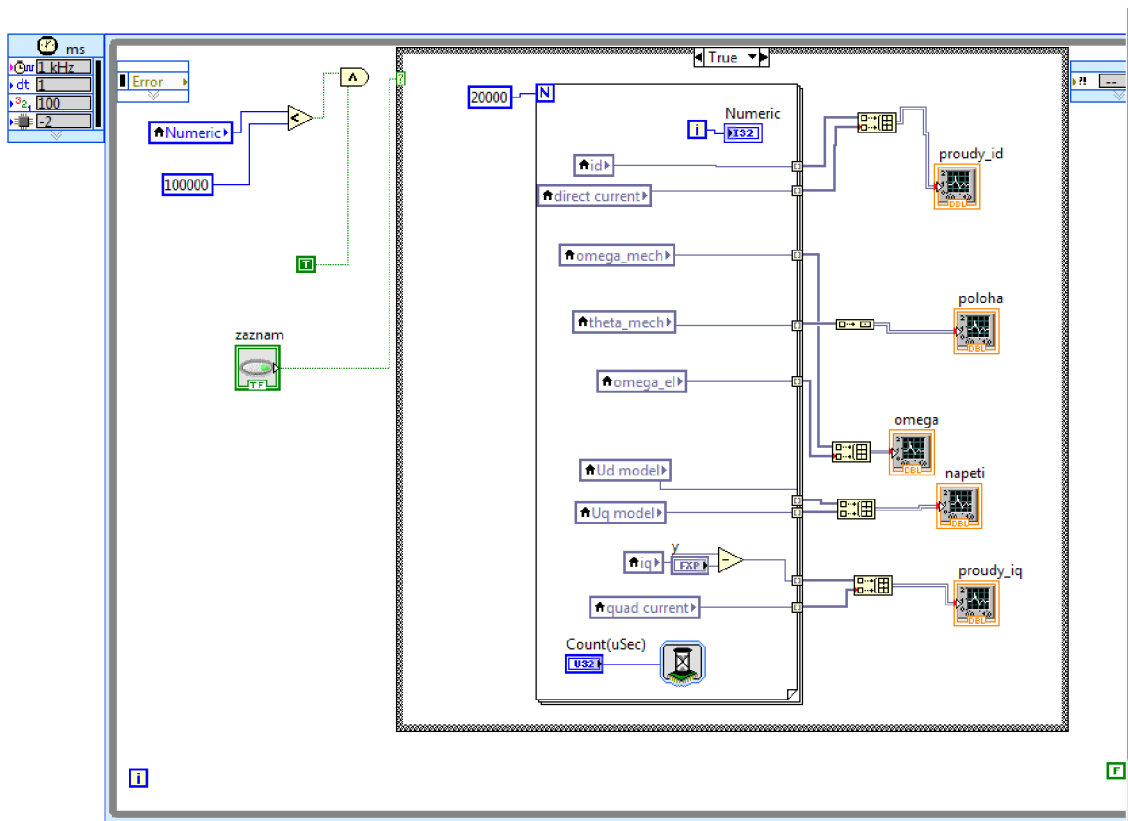
Před samotným testováním jsem ještě musel vyřešit problém se znaménkem před zátěžným momentem modelu. Je potřeba, aby tento moment působil vždy proti směru točení motoru, to znamená, aby měnil své znaménko zároveň se znaménkem úhlové rychlosti. Toho jsem dosáhl v RT části programu porovnáváním hodnoty úhlové rychlosti s nulovou hodnotou, jak je ukázáno na obrázku 40.



Obr. 40 : RT VI pro testování shody PMSM motoru a jeho modelu, první část

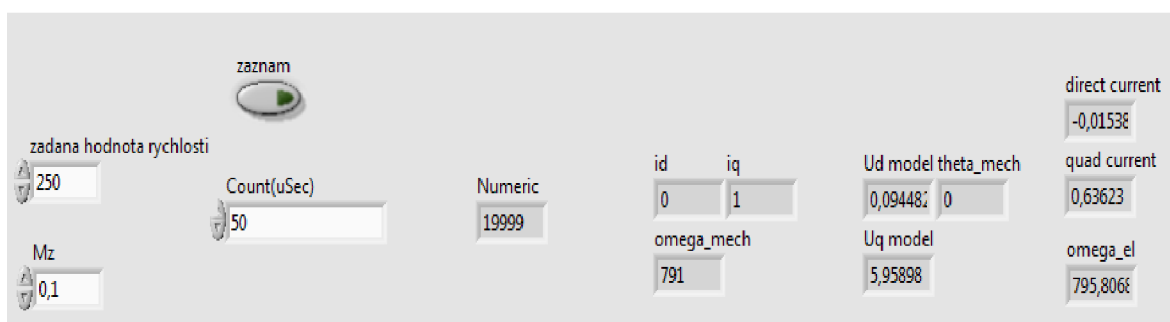
V druhé části tohoto VI souboru probíhá vykreslování proměnných do grafu. Zobrazují hodnoty vstupních napětí, úhlových rychlostí modelu i motoru a proudy v dq osách vypočtené z hodnot odečtených na motoru v porovnání s proudy v modelu.





Obr. 41 : RT VI pro testování shody PMSM motoru a jeho modelu, druhá část

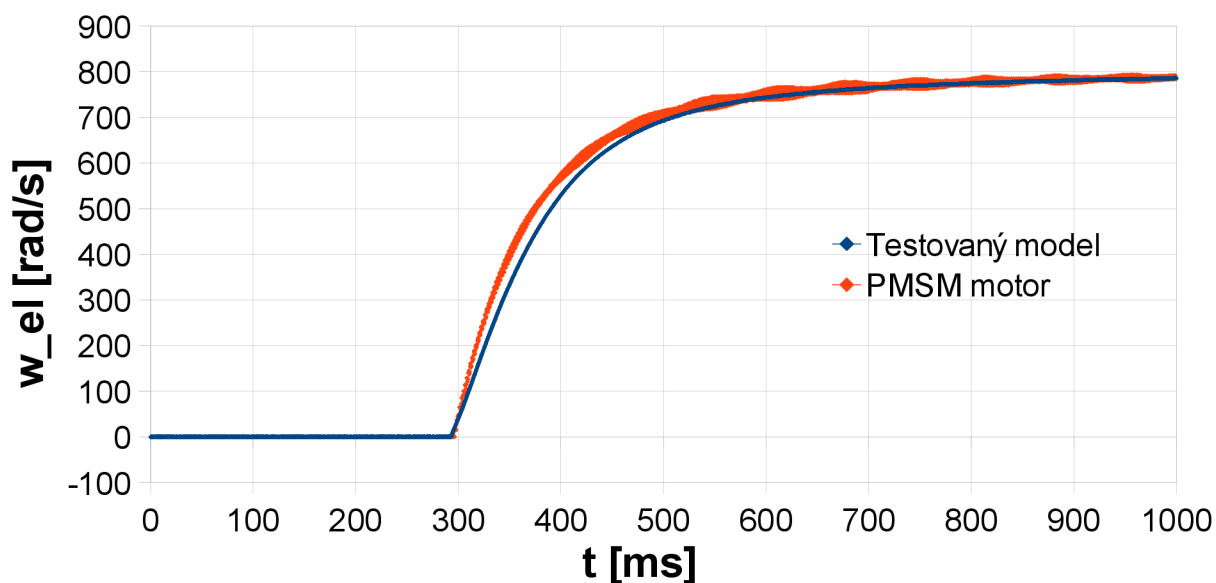
Do grafů vykresluji 20 000 hodnot po 50  $\mu$ s při sepnutém tlačítku záznam, to znamená, že vykreslený úsek je vždy přesně 1 sekundu dlouhý. Přimo v ovládání RT VI pak ještě nastavuji hodnotu zátěžného momentu a žádanou hodnotu úhlové rychlosti motoru i jeho modelu.



Obr. 42 : RT VI pro testování shody PMSM motoru a jeho modelu, ovládání

Nejdůležitější proměnnou, podle které jsem se snažil nastavovat hodnoty konstant modelu a zesílení u regulátorů, pro mě byla elektrická úhlová rychlost motoru. Z grafu porovnání této rychlosti u reálného motoru a jeho modelu je patrné jejich téměř shodné chování.

## Graf porovnání otáček PMSM motoru a jeho modelu



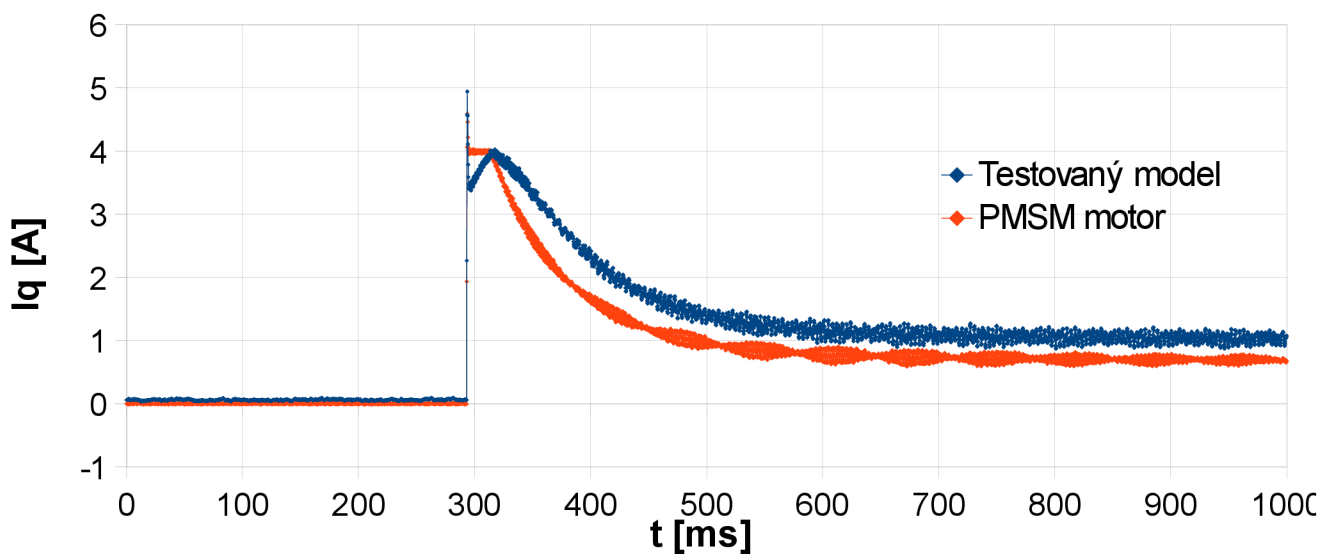
Graf 16: Porovnání otáček PMSM motoru a jeho modelu

Dalšími sledovanými proměnnými byly výstupní proudy modelu a reálného motoru. Jak již bylo zmíněno v kapitole o vektorovém řízení, regulují proud v d složce na nulovou hodnotu, momentotvorným proudem je tedy ten v q složce.

V dalším grafu č. 17 je vidět porovnání těchto  $i_q$  proudů. Je zde patrné shodné chování proudu v modelu i v reálném motoru. I v pravé části grafu, kde došlo k jisté aditivní chybě o hodnotě okolo 0,4 A modelovaného  $i_q$  proudu, je stále patrné shodné chování s reálným motorem, což je vidět v grafu 18, který je výřezem z grafu 17.

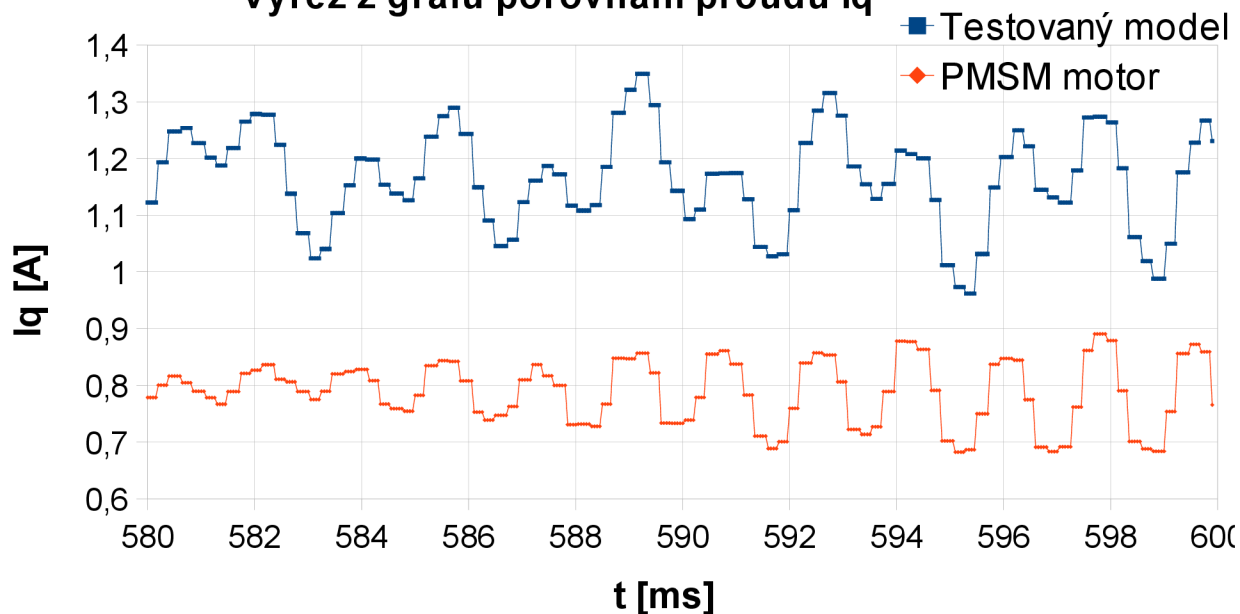
Ve výřezu jsou pěkně viditelné shodné nástupné a sestupné hrany proudu u motoru i jeho modelu, stejně jako vrcholové hodnoty ve shodných časech.

### Porovnání proudu $i_q$ PMSM motoru a jeho modelu



Graf 17: Porovnání proudu  $i_q$  PMSM motoru a jeho modelu

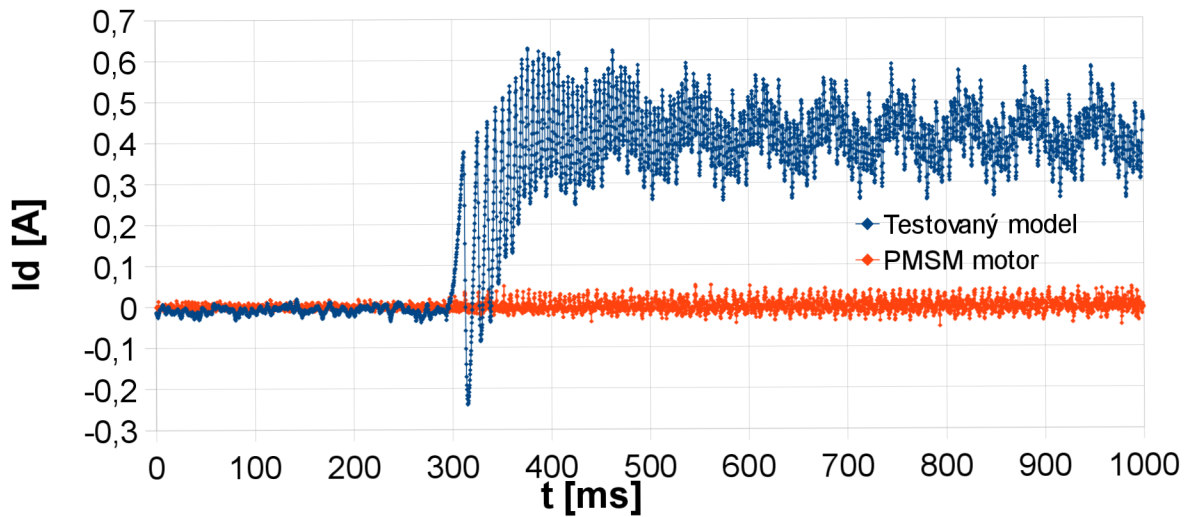
### Výřez z grafu porovnání proudu $i_q$



Graf 18: Výřez z grafu porovnání proudu  $i_q$

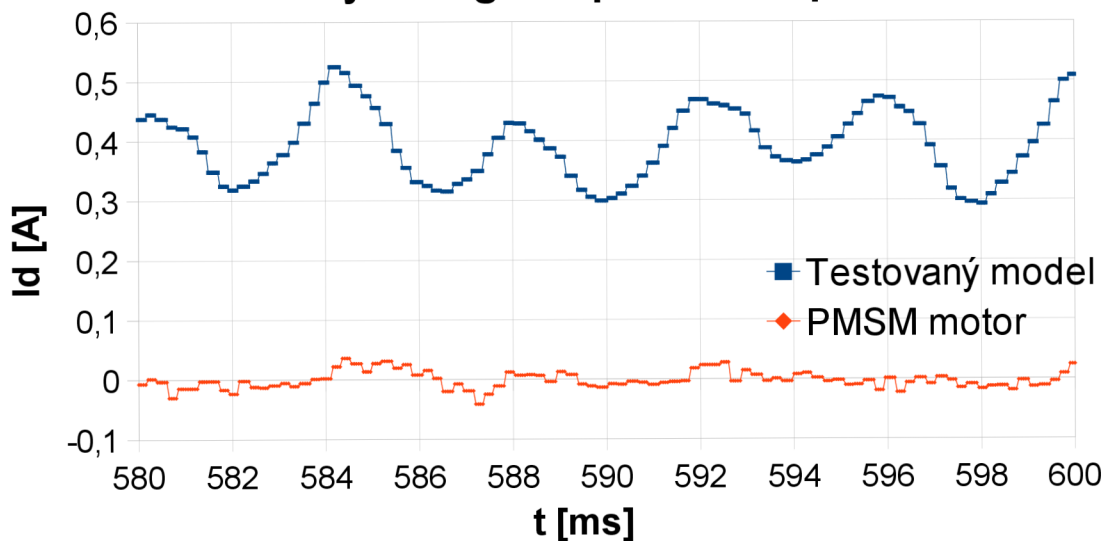
Obdobně jako proud  $i_q$  testoval jsem i shodu u proudu  $i_d$ . Po skokové změně žádané hodnoty úhlové rychlosti i zde došlo k jisté aditivní chybě o hodnotě asi 0,4 A. I přes tuto chybu se však stále motor chová obdobně jako jeho model.

### Porovnání proudu $I_d$ PMSM motoru a jeho modelu



Graf 19: Porovnání proudu  $i_d$  PMSM motoru a jeho modelu

### Výřez z grafu porovnání proudu $I_d$



Graf 20: Výřez z grafu porovnání proudu  $i_d$

I zde je na výřezu z grafu porovnání proudu  $i_d$  patrné stále stejné chování motoru i jeho modelu. Ačkoliv to zde již není tak dobře patrné, nástupné a sestupné hrany proudu v motoru a v jeho modelu stále víceméně odpovídají.

Na přesnost těchto průběhů má velký vliv volba konstant PMSM motoru, jejichž přesná identifikace nebyla součástí této práce. Snažil jsem se tedy tyto hodnoty nastavovat pouze experimentálně a na základě údajů udávaných v parametrech skutečného PMSM motoru.

# ZÁVĚR

V diplomové práci jsem popsal konkrétní postup modelování elektrických pohonů na platformě CompactRIO s využitím FPGA, popsal jsem rozdělení a jednotlivé typy elektrických pohonů stejně jako různé souřadné systémy pro jejich popis a transformace mezi těmito systémy.

Pro svou práci jsem si vybral synchronní motor s permanentními magnety PMSM. Odvodil jsem stavové rovnice pro model tohoto motoru v dq souřadném systému spojeném s rotorem stroje i přímo v abc souřadném systému, u kterého je možné zkoumat i různé nesymetričnosti a zkraty ve vedení motoru.

Zabýval jsem se možnostmi diskretizace těchto spojitých modelů jak z hlediska typu diskretizace, tak z hlediska volby vhodné periody vzorkování. Otestoval jsem diskrétní modely získané Eulerovou i kvadratickou diskretizací a porovnal jejich přesnost i výpočetní náročnost.

V programu Simulink jsem pak tyto diskrétní modely namodeloval a vyzkoušel si všechna pravidla pro automatické generování HDL kódu z tohoto programu. Testoval jsem také různé nastavení přesnosti mezivýpočtů modelu pomocí Fixed pointů, abych dosáhl ideálního kompromisu mezi přesností vygenerovaného kódu a místem, které kód zabere v hradlovém poli CompactRIA.

Pro otestování správné funkčnosti vytvořených dq modelů motoru jsem vytvořil projekt v programu Labview ověřující vlastnosti modelů pomocí real-time testování na platformě CompactRIO. V tomto projektu jsem si také vyzkoušel možnosti transformací souřadnic zpět do abc souřadného systému.

Chování vybraného nejvhodnějšího modelu PMSM motoru (model v dq souřadnicích získaný diskretizací prvního řádu, pracující s 36-bitovými rozsahy čísel) jsem pak úspěšně porovnal s chováním reálného motoru tohoto typu.

Zkoumal jsem reakce motoru a modelu na skokové změny žádané hodnoty otáček a také pomocí tachogenerátoru připojeného k reálnému motoru vliv změny zátěžného momentu stroje a modelu.

V závěrečné kapitole zabývající se tímto testováním je na grafech patrná shoda chování reálného motoru a mnou vytvořeného modelu. Pro ještě přesnější emulaci elektrického pohonu by byla zapotřebí dokonalejší identifikace jednotlivých parametrů testovaného PMSM motoru, která již nebyla součástí této práce.

# SEZNAM OBRÁZKŮ

Obr. 1 : Logické schéma úlohy.....	12
Obr. 2 : Fotografie platformy CompactRIO-9082 [6].....	13
Obr. 3 : Elektromotor – stator, rotor [9].....	15
Obr. 4 : Dělení elektrických pohonů [2].....	16
Obr. 5 : Clarkové transformace [2].....	19
Obr. 6 : Parkova transformace [2].....	21
Obr. 7 : Schéma sčítačky - Simulink.....	25
Obr. 8 : Definování délky slov pomocí Fixed pointů.....	26
Obr. 9 : Sčítačka – strom projektu v Labview.....	26
Obr. 10 : Sčítačka – Block diagram.....	27
Obr. 11 : Sčítačka – spuštěná aplikace.....	27
Obr. 12 : Trigonometrická funkce - Matlab.....	28
Obr. 13 : Implementace funkce sinus - Labview.....	29
Obr. 14 : Nastavení trigonometrické funkce - Labview.....	30
Obr. 15 : Porovnání výpočetní náročnosti trig. funkcí s a bez využití pipeline.....	30
Obr. 16 : Schéma vektorového řízení.....	31
Obr. 17 : Regulační schéma spojitého modelu s VC a transformacemi.....	32
Obr. 18 : Vektorové řízení - PI regulátor.....	32
Obr. 19 : Zobrazení souřadnicových systémů [2].....	34
Obr. 20 : Namodelovaná matice indukčností pro abc model v Simulinku.....	41
Obr. 21 : Namodelovaná inverze matice indukčností pro abc model.....	42
Obr. 22 : Náhled diskrétního modelu Eulerovou diskretizací v dq systému.....	51
Obr. 23 : Náhled diskrétního modelu kvadratickou diskretizací v dq systému.....	52
Obr. 24 : Náhled diskrétního modelu Eulerovou diskretizací v abc systému.....	53
Obr. 25 : Subsystem modelu PMSM motoru se vstupními hodnotami.....	54
Obr. 26 : Testovací schéma modelů v Simulinku.....	55
Obr. 27 : VI soubor v FPGA, část s vektorovým řízením.....	63
Obr. 28 : VI soubor v FPGA, část s modelem PMSM motoru.....	64
Obr. 29 : VI soubor v FPGA, ovládání.....	65

Obr. 30 : Transformace souřadných systémů v LabVIEW, pomocí IP Nodů.....	66
Obr. 31 : Transformace souřadných systémů v LabVIEW, knihovna NI SoftMotion67	
Obr. 32 : RT VI pro testování modelu a vykreslování grafů, první část.....	68
Obr. 33 : RT VI pro testování modelu a vykreslování grafů, druhá část.....	69
Obr. 34 : Výpis shrnujících informací o proběhlé kompilaci.....	73
Obr. 35 : Porovnání modelu s reálným motorem – schéma zapojení.....	75
Obr. 36 : Porovnání modelu s reálným motorem – fotografie pracoviště s popisky.....	76
Obr. 37 : Porovnání modelu s reálným motorem – fotografie pracoviště.....	77
Obr. 38 : Náhled testovacího VI nahraného v FPGA s modelem PMSM motoru.....	78
Obr. 39 : Ovládání testovacího FPGA VI s nastavením modelu PMSM motoru.....	79
Obr. 40 : RT VI pro testování shody PMSM motoru a jeho modelu, první část.....	80
Obr. 41 : RT VI pro testování shody PMSM motoru a jeho modelu, druhá část.....	81
Obr. 42 : RT VI pro testování shody PMSM motoru a jeho modelu, ovládání.....	81

## SEZNAM TABULEK

Tab. 1: Využití slotů cRIA 9082 při testování s reálným motorem [6].....	13
Tab. 2: Souřadné systémy v různých komplexních rovinách [1].....	18
Tab. 3: Parametry PMSM motoru pro testování modelů v Simulinku.....	54
Tab. 4: Chyby úhlové rychlosti při různých rozsazích diskrétních modelů.....	62
Tab. 5: Vytížení hradlového pole CompactRIA s různými modely.....	74
Tab. 6: Nastavení hodnot modelu při testování s reálným PMSM motorem.....	79



# SEZNAM GRAFŮ

Graf 1: Regulovaný průběh otáček sledující skokové změny žádané hodnoty.....	33
Graf 2: Porovnání diskretizací v dq souřadnicích se spojitým průběhem.....	56
Graf 3: Srovnání úhlové rychlosti modelů v dq a v abc souřadnicích .....	57
Graf 4: Srovnání proudů modelů v dq a v abc souřadnicích .....	57
Graf 5: Chyba úhlové rychlosti při různých hodnotách $T_s$ , 36-bitové rozsahy.....	58
Graf 6: Chyba úhlové rychlosti při různých hodnotách $T_s$ , 54-bitové rozsahy.....	59
Graf 7: Testování Fixed pointů - náhled.....	60
Graf 8: Testování Fixed pointů – chyby diskrétních modelů.....	61
Graf 9: Testování Fixed pointů – chyby diskrétních modelů 2.....	61
Graf 10: RT testování kvadratického modelu – graf elektrických otáček.....	70
Graf 11: RT testování kvadratického modelu – graf proudů.....	70
Graf 12: RT testování kvadratického modelu – graf napětí.....	71
Graf 13: RT testování kvadratického modelu – výřez z grafu polohy.....	71
Graf 14: RT testování Eulerova modelu – graf elektrických otáček.....	72
Graf 15: RT testování kvadratického modelu–graf transformovaných proudů $i_{abc}$ ....	72
Graf 16: Porovnání otáček PMSM motoru a jeho modelu.....	82
Graf 17: Porovnání proudu $i_q$ PMSM motoru a jeho modelu.....	83
Graf 18: Výřez z grafu porovnání proudu $i_q$ .....	83
Graf 19: Porovnání proudu $i_d$ PMSM motoru a jeho modelu.....	84
Graf 20: Výřez z grafu porovnání proudu $i_d$ .....	84

# LITERATURA

- [1] NEBORAK, I.: *Modelování a simulace elektrických regulovaných pohonů*, Ostrava 2002. ISBN 80-248-0083-7
- [2] VESELÝ, L.: *Algoritmy bezsnímačového řízení synchronního motoru s permanentními magnety*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2013. 122s. Vedoucí disertační práce byl doc. Ing. Pavel Václavek, Ph.D.
- [3] NATIONAL INSTRUMENTS: *Vývoj aplikací v LabVIEW pre meracie úlohy s CompactRIO systémami-LabVIEW Core 1-3 Course manual*. Centrum pro rozvoj výzkumu pokročilých řídicích a senzorických technologií, Bratislava 2012. CZ.1.07/2.3.00/09.0031
- [4] NATIONAL INSTRUMENTS: *Vývoj aplikací v LabVIEW pre meracie úlohy s CompactRIO systémami-LabVIEW Core 1 Exercises*. Centrum pro rozvoj výzkumu pokročilých řídicích a senzorických technologií, Bratislava 2012. CZ.1.07/2.3.00/09.0031
- [5] TG DRIVES: *Servomotors-AC brushless servomotors TGT a TGH*. [online] 2015. Dostupné z: [http://www.tgdrives.cz/fileadmin/user\\_upload/download/TG\\_Drives\\_2014\\_Katalog\\_servomotory\\_TGH%2BTGT\\_CZ.pdf](http://www.tgdrives.cz/fileadmin/user_upload/download/TG_Drives_2014_Katalog_servomotory_TGH%2BTGT_CZ.pdf)
- [6] NATIONAL INSTRUMENTS: *CompactRIO NI cRIO-9081/9082*. [online] National Instruments Corporation, 2011. 59s. Dostupné z: [http://www.ni.com/pdf/manuals/376904a\\_02.pdf](http://www.ni.com/pdf/manuals/376904a_02.pdf)
- [7] OCEN, D.: *Direct Torque Control of a Permanent Magnet synchronous Motor*. [online] 2005. Dostupné z: <http://www.diva-portal.org/smash/get/diva2:582454/FULLTEXT01.pdf>
- [8] VALA, R.: *RIO–revoluce v měření, řízení a automatizaci*. [online] Automa - časopis pro automatizační techniku. 4/2016. Dostupné z: [http://automa.cz/index.php?id\\_document=30937](http://automa.cz/index.php?id_document=30937)

- [9] ČERMÁK, J.: *Elektrické točivé stroje-rozdělení a základní vlastnosti*. [online] 4/2015. Oenergetice.cz. Dostupné z: <http://oenergetice.cz/technologie/elektroenergetika/elektricke-tocive-stroje-rozdeleni-a-zakladni-vlastnosti/>
- [10] BLAHA, P.: *Diskretizace spojitých systémů*. [online] 2015. 51 str. MTDS-Teorie dynamických systémů. Dostupné z: <https://sites.google.com/site/teoriedynamickyxsystemu/prednasky>
- [11] TOLIYAT, H.A., CAMPBELL, S.G.: *DSP-Based Electromechanical Motion Control*. CRC Press, 2003. 360s. ISBN 0203486331
- [12] PADALÍK, M.: *Algoritmy odbuzování při řízení střídavých motorů*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2008. 59s. Vedoucí diplomové práce doc. Ing. Pavel Václavek, Ph.D.
- [13] MALIŠ, M.: *Algoritmy bezsnímačového řízení pohonu se synchronním motorem*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, 2012. 110 s. Vedoucí práce byl doc. Ing. Pavel Václavek, Ph.D.

# SEZNAM PŘÍLOH

Příloha 1. CD s diplomovou prací a všemi zdrojovými soubory