



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

MIL SIMULACE ELEKTRICKÝCH MOTORŮ  
V REÁLNÉM ČASE

MIL REAL TIME SIMULATION OF ELECTRICAL MOTORS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Ondřej Bartík

VEDOUCÍ PRÁCE

SUPERVISOR

Doc. Ing. Petr Blaha, Ph.D.

BRNO 2017

# Diplomová práce

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

**Student:** Bc. Ondřej Bartík

**ID:** 146782

**Ročník:** 2

**Akademický rok:** 2016/17

**NÁZEV TÉMATU:**

## MIL simulace elektrických motorů v reálném čase

### POKYNY PRO VYPRACOVÁNÍ:

1. Nastudujte problematiku modelování a řízení střídavých motorů pomocí programovatelných hradlových polí v SoC (System on Chip) ZYNQ-7000.
2. Vytvořte modely BLDC motoru a asynchronního motoru, které poběží v FPGA
3. Oba modely doplňte modelem mechanické části.
4. Otestujte v simulaci Model In the Loop (MIL) s regulátorem běžícím ve vybrané části SoC (procesor ARM nebo FPGA). Jako testovací platformu použijte vývojovou desku ZedBoard nebo ZYBO.

### DOPORUČENÁ LITERATURA:

[1] Sul, S.K.: Control of Electric Machine Drive Systems. February 2011, Wiley-IEEE Press. ISBN: 978-0-470-59079-9.

Firemní literatura firmy Xilinx, další dle doporučení vedoucího.

**Termín zadání:** 6. 2. 2017

**Termín odevzdání:** 15.5.2017

**Vedoucí práce:** doc. Ing. Petr Blaha, Ph.D.

**Konzultant:**



**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 práce se zabývá implementací modelů dvou různých typů střídavých elektrických motorů v obvodu ZYNQ-7000 pro potřeby MIL simulace v reálném čase. Jedná se konkrétně o BLDC motor a asynchronní motor. V této práci je popsáno, jak byly jejich matematické modely upraveny pro potřebu implementace v obvodu ZYNQ-7000 a jak tyto modely byly v obvodu implementovány. V závěrečné části této práce jsou popsány tři různé MIL simulace s těmito motory.

## **Klíčová slova**

MIL, BLDC, asynchronní, motor, ZYNQ-7000, FPGA, real-time

## **Abstract**

The goal of this thesis is how to implement the two different types of the electric alternate motors in ZYNQ-7000 device for MIL real-time simulation purposes. The chosen types of motors are BLDC motor and AC induction motor. Mathematics models of these motor, the necessary changes for implementation purposes and the way how the models were implemented in ZYNQ-7000 device are described in this work. Three different experimental MIL simulation, using these motors are described at the end of this thesis.

## **Keywords**

MIL, BLDC, asynchronous, motor, ZYNQ-7000, FPGA, real-time

## **Bibliografická citace:**

BARTÍK, O. *MIL simulace elektrických motorů v reálném čase*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2017. 70s.  
Vedoucí práce: Doc. Ing. Petr Blaha, Ph.D.



## **Prohlášení**

„Prohlašuji, že svou závěrečnou práci na téma MIL simulace elektrických motorů v reálném čase 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 závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne **11. května 2017**

.....

podpis autora

## **Poděkování**

Děkuji vedoucímu diplomové práce Doc. Ing. Petru Blahovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

Tato práce byla podpořena projektem CIDAM - Centrum inteligentních pohonů a pokročilého řízení strojů TE02000103, který je financovaný Technologickou agenturou České republiky.

V Brně dne **28. dubna 2017**

.....  
podpis autora

# Obsah

1	Úvod .....	1
2	Software a Hardware prostředky pro tvorbu a implementaci modelů .....	2
2.1	MIL simulace.....	2
2.2	Prostředí Matlab/Simulink .....	3
2.3	Programovatelná hradlová pole FPGA .....	3
2.4	Architektura ARM Cortex-A9 .....	4
2.5	Rodina obvodů ZYNQ-7000.....	4
2.6	Prostředí Vivado HLS .....	5
2.7	Prostředí Vivado .....	6
2.8	Prostředí Xilinx SDK .....	7
2.9	Vývojový kit Zed Board .....	8
2.10	Závěr kapitoly SOFTWARE a HARDWARE prostředky pro tvorbu a implementaci modelů .....	8
3	Metodika implementace modelů v hradlovém poli FPGA.....	9
3.1	Metoda diskretizace pomocí Eulerovy metody.....	9
3.2	Aritmetika výpočtů s pevnou desetinnou čárkou .....	10
3.3	Použití měřítek v modelování dynamických systémů.....	11
3.4	Vytvoření implementace modelu pomocí Vivado HLS.....	12
3.5	Závěr kapitoly metodika implementace modelů v hradlovém poli FPGA .	13
4	Implementace modelu BLDC motoru .....	14
4.1	Základní popis BLDC motoru .....	14
4.2	Matematický model BLDC motoru .....	15
4.3	Úprava modelu pro implementaci .....	18
4.4	Implementace modelu v ZYNQ-7000.....	21
4.5	Závěr kapitoly implementace modelu BLDC motoru .....	24
5	Implementace modelu asynchronního motoru .....	25
5.1	Základní popis asynchronního motoru.....	25
5.2	Matematický model asynchronního motoru.....	26
5.3	Úprava modelu pro implementaci .....	30
5.4	Implementace modelu v ZYNQ-7000.....	36
5.5	Závěr kapitoly Implementace asynchronního motoru .....	39
6	Experimentální MIL simulace .....	40
6.1	Základní popis experimentu.....	40
6.2	Matematický model experimentální simulace .....	40
6.3	Úprava modelu pro implementaci .....	44
6.4	Implementace experimentu v ZYNQ-7000 .....	48
6.5	Závěr kapitoly Experimentální MIL simulace .....	49

7 Závěr .....	50
Literatura .....	51
Seznam zkratk, symbolů a veličin.....	54
Seznam příloh.....	56

## Seznam obrázků

Obr. 1: Příklad modelu v prostředí Matlab/Simulink .....	3
Obr. 2: Architektura obvodů ZYNQ-7000 [8] .....	5
Obr. 3: Hierarchie C simulace .....	6
Obr. 4: Hierarchie Co simulace.....	6
Obr. 5: Vývojový kit Zed Board[18].....	8
Obr. 6: Diskretizace Eulerovou metodou .....	10
Obr. 7: Velikost chyby v závislosti na kroku h .....	10
Obr. 8: Osmi bitová proměnná s pevnou desetinnou čárkou [19] .....	11
Obr. 9: Struktura IP core vytvořené v prostředí Vivado HLS .....	13
Obr. 10: Model BLDC motoru.....	17
Obr. 11: Průběh elektrických otáček a proudu vinutím .....	17
Obr. 12: Průběh mechanické polohy rotoru .....	17
Obr. 13: Přepracované modelovací schéma BLDC motoru .....	19
Obr. 14: Srovnání otáček a proudu mezi referenčním a upraveným modelem .....	21
Obr. 15: Srovnání mechanické polohy mezi referenčním a upraveným modelem..	21
Obr. 16: Srovnání otáček a proudu mezi upraveným modelem a simulacemi v prostředí Vivado HLS.....	22
Obr. 17: Srovnání mechanické polohy mezi upraveným modelem a simulacemi v prostředí Vivado HLS.....	23
Obr. 18: Diagram procesorového systému pro testování modelu BLDC motoru.....	23
Obr. 19: Srovnání otáček a proudu mezi upraveným modelem, Co-simulací v prostředí Vivado HLS a výpočtem na FPGA části obvodu ZYNQ-7000 .....	24
Obr. 20: Srovnání mechanické polohy mezi upraveným , Co-simulací v prostředí Vivado HLS a výpočtem na FPGA části obvodu ZYNQ-7000.....	24
Obr. 21: Model asynchronního motoru v $\alpha\beta$ rovině .....	29
Obr. 22: Průběhy statorových proudů v $\alpha\beta$ složkách modelu asynchronního motoru .....	29
Obr. 23: Průběhy rotorových toků v $\alpha\beta$ složkách modelu asynchronního motoru.	29
Obr. 24: Průběh elektrických otáček modelu asynchronního motoru .....	30
Obr. 25: Upravené modelovací schéma modelu asynchronního motoru .....	34
Obr. 26: Srovnání průběhů statorových proudů referenčního modelu a upraveného modelu asynchronního motoru.....	35
Obr. 27: Srovnání průběhů rotorových toků referenčního modelu a upraveného modelu asynchronního motoru.....	35
Obr. 28: Srovnání průběhů otáček referenčního modelu a upraveného modelu asynchronního motoru .....	35

Obr. 29: Srovnání proudů mezi upraveným modelem a simulacemi v prostředí Vivado HLS.....	37
Obr. 30: Srovnání toků mezi upraveným modelem a simulacemi v prostředí Vivado HLS.....	37
Obr. 31: Srovnání otáček mezi upraveným modelem a simulacemi v prostředí Vivado HLS.....	37
Obr. 32: Diagram procesorového systému pro testování modelu asynchronního motoru.....	38
Obr. 33 : Srovnání proudů mezi upraveným modelem, Co-simulací v prostředí Vivado HLS a výpočtem na FPGA části obvodu ZYNQ-7000.....	38
Obr. 34: Srovnání toků mezi upraveným modelem, Co-simulací v prostředí Vivado HLS a výpočtem na FPGA části obvodu ZYNQ-7000.....	39
Obr. 35: Srovnání otáček mezi upraveným modelem, Co-simulací v prostředí Vivado HLS a výpočtem na FPGA části obvodu ZYNQ-7000.....	39
Obr. 36: Uspořádání mechanické pružné spojky s dvěma tuhostmi.....	41
Obr. 37: Mode mechanické pružné spojky.....	41
Obr. 38: Spojení pružné spojky a BLDC motoru.....	42
Obr. 39: Spojení pružné spojky a asynchronního motoru.....	42
Obr. 40: Průběh otáček $\omega$ na výstupu mechanické spojky v případě BLDC (vlevo) motoru a asynchronního motoru (vpravo).....	43
Obr. 41: Spojení PI regulátoru proudu a BLDC motoru.....	44
Obr. 42: Průběh regulovaného proudu BLDC motoru.....	44
Obr. 43: Upravené modelovací schéma mechanické pružné spojky.....	46
Obr. 44: Upravené modelovací schéma PI regulátoru.....	47
Obr. 45: Srovnání otáček referenčním a upraveným modelem BLDC motoru (vlevo) a asynchronního motoru (vpravo).....	47
Obr. 46: Srovnání výstupu proudu referenčního modelu a upraveného modelu BLDC motoru s připojeným PI regulátorem.....	48
Obr. 47: Srovnání průběhů otáček mechanické spojky mezi referenčním modelem, upraveným modelem pro implementaci a reálnou simulací BLDC motoru (vlevo) a asynchronního motoru (vpravo).....	49
Obr. 48: Srovnání průběhů proudů mezi referenčním modelem, upraveným modelem pro implementaci a reálnou simulací BLDC motoru s připojeným PI regulátorem.....	49

## Seznam tabulek

Tab. 1: Parametry motoru.....	17
Tab. 2: Vstupní parametry modelu BLDC motoru .....	20
Tab. 3: Výstupní datový matematických operací modelovacího schématu BLDC motoru.....	20
Tab. 4: Výstupní parametry modelu BLDC motoru.....	21
Tab. 5: Hodnocení implementace modelu BLDC motoru v prostředí Vivado HLS..	22
Tab. 6: Hodnocení implementace procesorového systému v prostředí Vivado .....	23
Tab. 7: Parametry modelu asynchronního motoru.....	28
Tab. 8: Vstupní parametry modelu asynchronního motoru.....	33
Tab. 9: Datové typy výstupů matematických operací uvnitř modelu .....	34
Tab. 10: Popis výstupních parametrů modelu asynchronního motoru.....	34
Tab. 11: Hodnocení implementace modelu asynchronního motoru v prostředí Vivado HLS.....	36
Tab. 12: Hodnocení implementace procesorového systému v prostředí Vivado.....	38
Tab. 13: Parametry mechanické pružné spojky .....	43
Tab. 14: Popis vstupů a výstupů upraveného modelu mechanické spojky .....	47
Tab. 15: Popis vstupů a výstupů upraveného schématu PI regulátoru.....	48

## Seznam rovnic

Rovnice 1: Obecná diferenciální úloha prvního řádu.....	9
Rovnice 2: Nahrazení diferencí.....	9
Rovnice 3: Integrace Eulerovou metodou .....	9
Rovnice 4: Rovnice setrvačného členu prvního řádu.....	12
Rovnice 5: Bezrozměrné veličiny pro měřítkování setrvačného členu prvního řádu .....	12
Rovnice 6: Vztahy mezi bezrozměrnými veličinami a veličinami setrvačného členu .....	12
Rovnice 7: Měřtkovaný popis setrvačnosti prvního řádu.....	12
Rovnice 8: Rovnice napětí statorových vinutí BLDC motoru.....	15
Rovnice 9: Rovnice spřažených toků BLDC motoru .....	15
Rovnice 10: Rovnice vnitřního magnetického momentu BLDC motoru.....	15
Rovnice 11: Příklad hodnot napětí a proudů pro danou polohu BLDC motoru .....	15
Rovnice 12: Elektrická rovnice BLDC motoru s uvažovanými spřaženými toky .....	16
Rovnice 13: Elektrická rovnice BLDC motoru s uvažovanou konstantou stroje.....	16
Rovnice 14: Vztah mezi elektrickými a mechanickými otáčky BLDC motoru .....	16
Rovnice 15: Mechanická rovnice BLDC motoru .....	16
Rovnice 16: Rovnice mechanické polohy BLDC motoru .....	16
Rovnice 17: Diskretizovaná napěťová rovnice BLDC motoru .....	18
Rovnice 18: Diskretizovaná mechanická rovnice BLDC motoru.....	18
Rovnice 19: Diskretizovaná rovnice mechanické polohy BLDC motoru .....	18
Rovnice 20: Napěťová rovnice BLDC motoru doplněna o měřítko.....	18
Rovnice 21: Mechanická rovnice BLDC motoru doplněna o měřítko .....	18
Rovnice 22: Bezrozměrné veličiny pro měřítkování BLDC motoru .....	19
Rovnice 23: Měřítka modelu BLDC motoru .....	19
Rovnice 24: Skluz asynchronního motoru.....	26
Rovnice 25: Statorová, proudová rovnice složky $\alpha$ modelu asynchronního motoru .....	26
Rovnice 26: Statorová, proudová rovnice složky $\beta$ modelu asynchronního motoru .....	26
Rovnice 27: Rotorová, toková rovnice složky $\alpha$ modelu asynchronního motoru ....	26
Rovnice 28: Rotorová, toková rovnice složky $\beta$ modelu asynchronního motoru ....	26
Rovnice 29: Vztah mezi elektrickými a mechanickými otáčkami asynchronního motoru .....	27
Rovnice 30: Substituce $K_r$ .....	27
Rovnice 31: Substituce $K_l$ .....	27
Rovnice 32: Substituce $T_r$ .....	27



Rovnice 33: Vnitřní magnetický moment asynchronního motoru v $\alpha\beta$ rovině .....	27
Rovnice 34: Mechanická rovnováha momentů.....	27
Rovnice 35: Mechanická rovnice modelu asynchronního motoru v $\alpha\beta$ rovině .....	28
Rovnice 36: Diskretizovaná statorová, proudová rovnice složky $\alpha$ .....	30
Rovnice 37: Diskretizovaná statorová, proudová rovnice složky $\beta$ .....	30
Rovnice 38: Diskretizovaná rotorová, toková rovnice složky $\alpha$ .....	30
Rovnice 39: Diskretizovaná rotorová, toková rovnice složky $\beta$ .....	30
Rovnice 40: Diskretizovaná mechanická rovnice modelu asynchronního motoru v $\alpha\beta$ rovině.....	30
Rovnice 41: Měřítkovaná statorová, proudová rovnice složky $\alpha$ .....	31
Rovnice 42: Měřítkovaná statorová, proudová rovnice složky $\beta$ .....	31
Rovnice 43: Měřítkovaná rotorová, toková rovnice složky $\alpha$ .....	31
Rovnice 44: Měřítkovaná rotorová, toková rovnice složky $\beta$ .....	31
Rovnice 45: Měřítkovaná mechanická rovnice modelu asynchronního motoru v $\alpha\beta$ rovině.....	32
Rovnice 46: Bezrozměrné veličiny pro měřítkování modelu asynchronního motoru .....	32
Rovnice 47: Měřítko modelu asynchronního motoru .....	32
Rovnice 48: Rovnosti měřítek .....	32
Rovnice 49: Odvození důkazu rovnosti měřítek pro indukčnost a $K_l$ .....	32
Rovnice 50: Důkaz rovnosti měřítek pro indukčnost a $K_l$ .....	33
Rovnice 51: Odvození důkazu rovnosti měřítek pro odporu a $K_r$ .....	33
Rovnice 52: Důkaz rovnosti měřítek pro odporu a $K_r$ .....	33
Rovnice 53: První rovnice mechanické pružné spojky.....	40
Rovnice 54: Druhá rovnice mechanické pružné spojky .....	40
Rovnice 55: Zjednodušený ekvivalent první rovnice mechanické spojky .....	42
Rovnice 56: Zjednodušený ekvivalent druhé rovnice mechanické spojky .....	42
Rovnice 57: Rovnice PI regulátoru .....	43
Rovnice 58: Diskretizovaný ekvivalent první rovnice mechanické spojky .....	45
Rovnice 59: Diskretizovaný ekvivalent druhé rovnice mechanické spojky .....	45
Rovnice 60: Diskretizovaná první rovnice mechanické spojky rozšířená o měřítko .....	45
Rovnice 61: Diskretizovaná druhá rovnice mechanické spojky rozšířená o měřítko .....	45
Rovnice 62: Bezrozměrné členy upraveného modelu mechanické pružné spojky .	45
Rovnice 63: Měřítko modelu pružné mechanické spojky.....	45
Rovnice 64: Diskretizovaná rovnice PI regulátoru .....	46
Rovnice 65: Měřítkovaná rovnice PI regulátoru .....	46
Rovnice 66: Bezrozměrné členy pro měřítko PI regulátoru .....	46

Rovnice 67: Měřítko PI regulátoru .....	46
---	----

# 1 ÚVOD

V praxi může být mnohdy problém provádět určité experimenty na reálných zařízeních. Neodladěné a neozkoušené postupy a procesy mohou mít za následek zničení zařízení a to s sebou přináší problémy, jako jsou prodloužení doby činnosti, zvýšení finančních nákladů a jiné. Pro vyhnutím se těmto rizikům je vhodné pracovat s modely reálných zařízení, které se nedají „zničit“. Velkou nevýhodou modelů je fakt, že jsou jen nepřesným odhadem skutečnosti a nelze je brát jako dokonalou náhradu reálného zařízení. Nicméně, pro hrubý návrh všech procesů a postupů, jsou dostačující.

Věrnost modelu vůči reálnému zařízení, je přímo úměrná komplexnosti modelu. A závisí na následujících aspektech: s jak velkým pracovním rozsahem reálného zařízení model pracuje, zda se jedná o model, který vystihuje pouze lineární chování reálného zařízení nebo model, který respektuje i nelinearity. V neposlední řadě záleží na numerické přesnosti výpočtu modelu vůči skutečnému zařízení. Všechno tyto aspekty mají vliv na výpočetní náročnost modelu, a tedy i na dobu celého výpočtu. V praxi může být požadováno aby určitý proces, který proběhne na reálném zařízení proběhl za stejnou dobu i v případě modelu. Toto chování bývá označeno jako proces dějící se v reálném čase, protože výsledky procesu jsou validní v okamžiku, kdy je to požadováno. Tato vlastnost je požadována například v případě *Model In the Loop* (MIL) simulacích, kde je simulace složena z nezávislých simulačních bloků, které mohou být implementovány na různých platformách a s různou výpočetní náročností. Simulace *Hardware In the Loop* (HIL) je další skupinou, kde je toto chování požadováno. V tomto případě je model doplněn o fyzické rozhraní tak, aby mohl přímo nahradit reálné zařízení v konkrétním procesu. Stále se jedná pouze o model.

Tato práce se zabývá návrhem a implementací modelů elektrických motorů pro MIL simulace v reálném čase. V této práci jsou uvažovány dva elektrické motory: Prvním je *BrushLess DC* (BLDC) motor a druhým je asynchronní motor. V práci je také uvažován model mechanické pružné spojky dvou tuhostí. Modely jsou implementovány na obvod ZYNQ-7000 firmy Xilinx. Konkrétně je zde využito FPGA části tohoto obvodu.

Práce je členěna dle následující osnovy: Nejprve jsou popsány *Software* a *Hardware* prostředky pro tvorbu a implementaci modelů. Následně je zde popsána metodika s jakou byly modely motorů implementovány. Potom následují jednotlivé sekce popisující implementaci modelu konkrétního motoru. Na závěr je v této práci popsána experimentální MIL simulace BLDC motoru s připojenou mechanickou pružnou spojkou, asynchronního motoru s připojenou mechanickou pružnou spojkou a MIL simulace BLDC motoru s proudovým PI regulátorem.

## 2 SOFTWARE A HARDWARE PROSTŘEDKY PRO TVORBU A IMPLEMENTACI MODELŮ

Vytvoření a implementace modelu pro *model in the loop* (dále jen MIL) simulaci zahrnuje několik kroků, jako je sestavení a simulace modelu, připravení pro implementaci na konkrétní hardware (dále jen HW) platformě a provedení samotné simulace. Pro každý takovýto krok je vhodný jiný nástroj, a to například v závislosti na dostupném simulačním nástroji a zvolené HW platformě.

Tato kapitola popisuje MIL simulace, jejich podstatu a využití. Dále popisuje nástroje Matlab/Simulink, které byly zvoleny jako nástroje pro simulaci modelů. Následuje popis skupiny nástrojů společnosti Xilinx, která distribuuje programovatelná hradlová pole (*Field Programmable Gate Arrays*, dále jen FPGA). Mezi tyto nástroje patří: Vivado HLS, Vivado a Xilinx SDK. Všechny tyto nástroje jsou nezbytné pro implementaci modelů. Jako cílová HW platforma byl zvolen vývojový kit ZED Board, který je osazen obvodem XC7Z20 rodiny obvodů ZYNQ-7000 společnosti Xilinx. Popis a vlastnosti této HW platformy jsou taktéž popsány v této kapitole.

### 2.1 MIL simulace

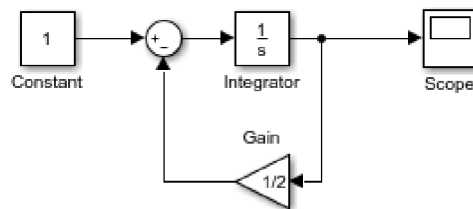
MIL simulace je simulace několika samostatných, na sobě nezávislých simulovaných bloků, které jsou vhodně propojeny tak, aby byly splněny požadavky simulace [1]. Mezi tyto požadavky patří například správné časování jednotlivých iterací výpočtů, nebo celková doba simulace.

V obvodech FPGA je možné implementovat různé logické bloky a obvody, jejichž činnost je prováděna nezávisle na sobě. Toto je možné díky nativnímu paralelismu, se kterým se logické bloky implementují [2].

Výhodou MIL simulací je provedení simulace v rámci jedné platformy, například v prostředí Matlab/Simulink, nebo jako implementace v FPGA. Výhodou FPGA implementace je vysoký výkon platformy, který umožňuje provádět výpočty v reálném čase. Mezi další výhody MIL simulace patří i to, že není potřeba velké množství externího HW, které je potřeba pro běžný provoz simulované činnosti. Stačí se omezit pouze na jedinou platformu. V neposlední řadě MIL simulace také zkracuje dobu vývoje simulované činnosti. V případě, že je potřeba upravit něco v simulované činnosti, je toto nutno udělat pouze v modelu a ne v reálném prostředí.

## 2.2 Prostředí Matlab/Simulink

Matlab je komerční SW nástroj společnosti Mathworks. Jeho největší užití je pro inženýrské výpočty [3]. Zahrnuje několik nástrojů pro různé odvětví činnosti. Jeho rozšíření Simulink pak umožňuje tvořit a spouštět simulace vytvořených modelů. Jejich tvorba se děje pomocí propojování základních bloků jako jsou: integrátory, násobící členy, sčítací členy, bloky pro výpočet matematické funkce, bloky pro zobrazení průběhů proměnných v čase a jiné. Na Obr. 1 je vidět příklad jednoduchého modelu setrvačného členu prvního řádu vytvořeného v prostředí Matlab/Simulink.



Obr. 1: Příklad modelu v prostředí Matlab/Simulink

Prostředí Matlab Simulink bylo v této práci použito pro prvotní realizace modelů motorů.

## 2.3 Programovatelná hradlová pole FPGA

Programovatelná hradlová pole FPGA jsou integrované obvody, které umožňují díky jejich konfigurovatelné struktuře vytvořit širokou škálou číslicových obvodů [4]. Jejich struktura obsahuje několik prvků, jejichž kombinací pak vzniká kombinační, nebo sekvenční logika. Jejich architektura se liší podle použité rodiny obvodů. Tato práce uvažuje použití obvodů firmy Xilinx sedmé generace, a proto zde bude tato architektura stručně popsána [5]. Struktura těchto obvodů se skládá z konfiguračních logických bloků (*configurable logic blocks*, dále jen CLB), které jsou propojeny do propojovací matice (*routing matrix*). Ta umožňuje vzájemné propojení více CLB dohromady. CLB pak obsahují šestnáct kopných obvodů D (*D flip-flop*, dále jen DFF) a osm šesti vstupových generátorů logických funkcí (*Look Up Table*, dále jen LUT). Ke každé LUT je připojena dvojice DFF. Dále obsahují paměťové bloky RAM (dále jen BRAM) pro uchovávání dat a digitální signálové procesní (dále jen DSP) bloky. DSP bloky slouží pro implementaci matematických operací jako je násobení a dělení.

Díky vysoké integraci těchto obvodů (největší FPGA obvody obsahují více než 1 420 000 CLB) jsou FPGA vhodné pro velké spektrum aplikací. Mezi tyto patří serverové a síťové aplikace, zpracování audio a video signálů, medicínské a vojenské aplikace. Své uplatnění nacházejí i ve vesmírné technice.

Díky možnosti implementovat matematické operace, jsou FPGA obvody vhodné i pro implementaci modelů a jiných matematicky náročných algoritmů.

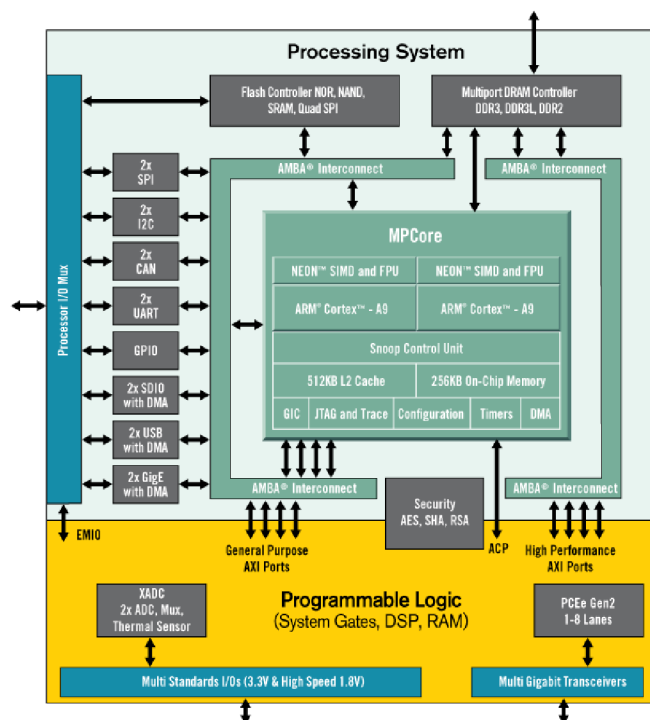
Nevýhodou hradlových polí je jejich komplexnost a velká rozdílnost v architektuře mezi jednotlivými rodinami obvodů. Ta má za následek horší přenositelnost implementace z hlediska optimálního chodu. Různé rodiny obvodů mohou mít různý počet DFF v CLB, nebo i různý počet vstupů u LUT, to má za následek nutnost použití jiných optimalizačních přístupů. Nebo i nového přístupu při implementaci požadovaného logického obvodu.

## 2.4 Architektura ARM Cortex-A9

ARM Cortex-A9 je 32bitový mikrokontroler, jehož architektura je vytvořena společností ARM [6]. Jeho hlavní určení je pro nízkoenergetické aplikace. Výrobce nabízí čtyř i více jádrové řešení, kde každé jádro je uloženo v klastru. V obvodech ZYNQ-7000 je použita dvou jádrová variace. Komunikace mezi jádry a sběrnicovým rozhraním architektury je zajištěna pomocí *Snoop Control Unit* (SCU). V každém klastru kromě jádra Cortex-A9 je uložena jednotka pro výpočty s plovoucí desetinnou čárkou (*Floating Point Unit*, dále jen FPU) s rozsahem 32 bitů a jednotka pro jedнокrokovou instrukci násobení, jak pro celočíselné operace, tak i pro operace s desetinnými čísly. Dále pak instrukční a datovou, rychlou paměť cache. Pro všechna jádra je pak společný řadič přerušení a sběrnicové rozhraní pro periferie a externí paměť dat a programu. Přes sběrnicový systém je možné adresovat prostor o velikosti 4G 32 bitových slov. Tento prostor je společný jak pro data tak i pro instrukce.

## 2.5 Rodina obvodů ZYNQ-7000

Jak už bylo zmíněno výše, obvody SOC ZYNQ-7000 obsahují dvou jádrovou variaci mikrokontroleru ARM Cortex-A9 a hradlové pole FPGA společnosti Xilinx rodiny obvodů sedmé generace. Mimo to také obsahuje fyzické periferie připojené na sběrnicový systém mikrokontroleru ARM. Tyto periferie pak jsou: sériové komunikační rozhraní SPI, I2C, CAN, UART. Rozhraní pro práci s porty GPIO a jiná [7]. Na Obr. 2 lze vidět architekturu obvodu ZYNQ-7000.



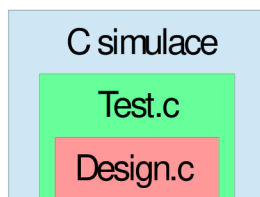
Obr. 2: Architektura obvodů ZYNQ-7000 [8]

FPGA část obvodu ZYNQ 7000 je spojena s architekturou ARM přes sběrnicové rozhraní, podobně jako s fyzickými periferiemi. Toto spojení pak umožňuje vytvoření vlastních paměťově mapovaných periferií uvnitř hradlového pole FPGA. Tyto periferie jsou pak ze strany ARM přístupné stejně, jako kterékoliv jiné fyzické periferie uvnitř obvodu ZYNQ-7000. Tato skutečnost umožňuje využít obvody ZYNQ-7000 ve velkém množství aplikací, kde je potřeba vytvoření vlastních periferií, protože dostupná mikrokontrolerová řešení nemusejí těmito periferiemi disponovat.

## 2.6 Prostředí Vivado HLS

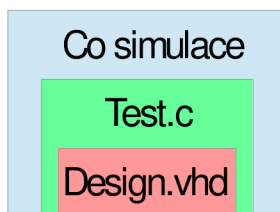
Prostředí Vivado HLS je nástroj pro vygenerování VHDL, popřípadě Verilog kódu, pro FPGA implementaci z algoritmů definovaných ve vyšších programovacích jazycích jako jsou C/C++ [9]. Prostředí disponuje knihovnami pro práci s proměnnými s pevnou desetinnou čárkou a s proměnnou délkou, které jsou vhodné pro implementaci matematických algoritmů v hradlových polích FPGA. A to díky své nižší náročnosti na výpočetní výkon. Prostředí Vivado HLS umožňuje použít následující postup [9]: Vytvořit aplikaci v C/C++ jazyce, která reprezentuje výslednou implementaci v hradlovém poli. Dále pak vytvořit testovací aplikaci v jazyce C/C++ pro implementovanou aplikaci a tím ověřit správnost algoritmu

pomocí *C simulace*. Hierarchie této simulace lze vidět na Obr 3. Následně spustit překlad C/C++ kódu do VHDL/Verilog jazyka.



**Obr 3: Hierarchie C simulace**

Po tomto kroku je k dispozici výkonové hodnocení implementace. Toto hodnocení obsahuje časovou náročnost chodu implementace a její využití zdrojů v obvodech FPGA. Poté je možné aplikovat implementační optimalizace, které mají dopad na zmíněné výkonové hodnocení. Tyto optimalizace jsou například: nahrazení vnitřních smyček pomocí pipeline, nebo definování protokolů pro vstupní a výstupní rozhraní implementovaného algoritmu. Po přeložení a aplikaci všech optimalizací, je možné pustit *Co-simulaci*. Ta umožňuje na základě testovací aplikace, která byla použita pro *C simulaci*, vytvořit sadu simulačních signálů (dále jen stimulů) a pomocí těchto stimulů otestovat chování implementace. Díky tomuto je možné aplikovat stejný test jako v případě *C simulace* a to se stejnými výstupy. Na Obr. 4 lze pozorovat hierarchii *Co simulace*.



**Obr. 4: Hierarchie Co simulace**

Po provedení všech výše popsaných kroků je možné exportovat výslednou implementaci v podobě samostatného bloku pro implementaci v komplexnějších aplikacích. Tyto bloky se nazývají *IP core (Intelligent Property core)*.

## 2.7 Prostředí Vivado

Vývojové prostředí Vivado [10] firmy Xilinx je prostředí pro tvorbu a editaci kódu v některém z popisovacích jazyků pro programování hradlových polí FPGA. Nebo také tvorbu aplikaci sestavenou z *IP core* bloků. Prostředí také umožňuje provést



nezbytné operace pro vytvoření konfiguračního souboru pro hradlová pole. Ten reprezentuje logiku, která má být implementována.

Nezbytné operace pro vytvoření konfiguračního souboru jsou syntéza a implementace. Syntéza je proces, který překládá textový kód do hradové reprezentace [11]. Logika je reprezentována jako zapojení kombinačních a sekvenčních logických prvků. Operace syntézy neprovádí žádné optimalizace v překládané logice. Může být jen optimalizována z hlediska spotřebovaného času nebo operační paměti počítače, na kterém je vykonána. Žádné ze zvolených optimalizací neovlivňují výsledek operace, pouze její dobu trvání. Operace implementace pak překládá výstupy syntézy na jinou hradlovou reprezentaci [12]. Funkčnost je zachována, ale překlad je proveden s ohledem na vnitřní strukturu FPGA obvodu. Během procesu implementace se provádí optimalizace překládané logiky. Tato optimalizace se zaměřuje na velikost použitých zdrojů (LUT, DFF, BRAM atd.), nebo na datovou propustnost implementované logiky. Výsledek je vždy kompromisem mezi těmito dvěma aspekty. Pro dosažení maximální úrovně optimalizace je nutné poskytnout informace o pracovních taktech a o použitých vývodech hradlového pole. Proces implementace rozhoduje, kde fyzicky uvnitř struktury hradlového pole bude logika implementována. Při tomto kroku se projeví i metodika, s jakou byl napsán kód, popisující chování výsledné logiky. Pokud je například definován klopný obvod typu D s resetovacím vstupem aktivním v úrovni logické jedničky a struktura hradlového pole disponuje pouze klopnými obvody typu D s resetovacím vstupem aktivním v úrovni logické nuly, musí při procesu implementace do výsledné logiky přidat některé prvky navíc, aby byla docílena žádaná konfigurace. Po úspěšném provedení procesu implementace je vytvořen již zmíněný soubor pro konfiguraci hradlového pole.

Prostředí Vivado také umožňuje simulaci vytvořených logických obvodů a to ve dvou úrovních. První úroveň je pro logiku před procesem implementace, a tedy bez optimalizací a druhá je pro logiku po procesu implementace, a tedy i s provedenými optimalizacemi.

Práce s FPGA není jediným aspektem vývojového prostředí Vivado. Umožňuje pracovat i s obvody ZYNQ-7000, kde je procesor ARM reprezentován neměnnou částí, která je propojena s vytvořenou částí pro hradlové pole. A vytvářet tak procesorové systémy. Část ARM je pak vyloučena z procesu syntézy a implementace.

## 2.8 Prostředí Xilinx SDK

Prostředí Xilinx SDK (*Software Development Kit*) slouží pro tvorbu zdrojového kódu v jazycích C/C++ pro procesorová jádra [13]. Těmito jádry se myslí jak fyzická jádra ARM v obvodech ZYNQ-7000 tak i tzv. *soft-core* procesorová jádra,

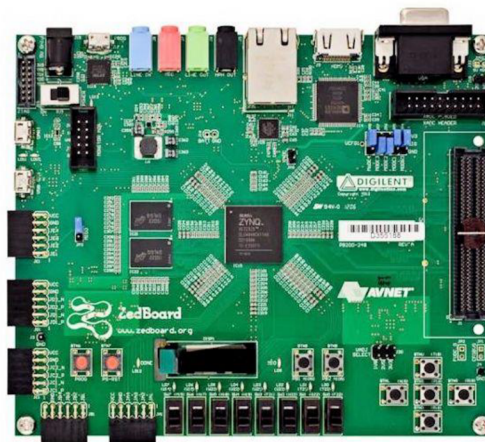
kteřá jsou implementována v FPGA obvodech. Mezi tyto patří například [14], [15] jądra *PicoBlaze* a *MicroBlaze*, obě distribuována společností Xilinx.

Prostředí Vivado SDK disponuje standardními nástroji pro tvorbu kódu jako jsou: textový editor, kompilátor C/C++ programovacích jazyků, nástroj pro ladění (*debug*), nástroj pro správu proměnných a paměťového prostoru a jiné.

Nástroj také umožňuje importovat procesorový systém implementovaný v FPGA a pracovat s ním jako se standardním reálným procesorovým systémem. Toto prostředí také zajišťuje nahrání programu do procesoru, případně i nahrání konfiguračního souboru pro FPGA [13].

## 2.9 Vývojový kit Zed Board

Pro implementaci a testování modelů byl zvolen vývojový kit Zed Board [16]. Tento kit je kromě běžných periférií (LED diody, přepínače, rozhraní ethernet, univerzální konektory) osazen obvodem XC7Z020CLG484-1[17]. Jedná se o obvod rodiny obvodů ZYNQ-7000 a skládá se ze dvoujádřového kontroléru s jádry ARM Cortex-A9 a FPGA částí, která architekturou odpovídá sedmé generaci obvodů FPGA rodiny ARTIX-7. FPGA část tohoto obvodu obsahuje 510000 DFF, 680000 LUT, 4,9Mb BRAM a 220 DSP bloků. Na Obr. 5 je vyobrazen vývojový kit Zed Board.



Obr. 5: Vývojový kit Zed Board[18]

## 2.10 Závěr kapitoly SOFTWARE a HARDWARE prostředky pro tvorbu a implementaci modelů

V této kapitole byla popsána podstata a význam MIL simulace. Dále zde byly popsány SW prostředky pro tvorbu modelů a také SW prostředky pro práci s hradlovými poli společnosti Xilinx. Dále zde byla popsána HW platforma použita pro implementaci modelů elektrických motorů. Jednalo se o obvod XC7Z020CLG484-1 rodiny obvodů ZYNQ-7000 společnosti Xilinx.

## 3 METODIKA IMPLEMENTACE MODELŮ V HRADLOVÉM POLI FPGA

V této kapitole a jejich jednotlivých podkapitolách bude popsána metodika tvorby a implementace modelů dvou typů elektrických motorů pro obvod XC7Z020CLG484-1. Nejprve budou v této kapitole popsány techniky použité při návrhu modelu pro implementaci. Mezi tyto techniky patří metoda Eulerovy diskretizace, využití výpočtů s pevnou desetinnou čárkou a využití měřítkování. Na závěr zde bude popsána tvorba *IP core* periferie v prostředí Vivado HLS.

### 3.1 Metoda diskretizace pomocí Eulerovy metody

Eulerova metoda patří mezi jednokrokové metody pro řešení diferenciálních rovnic. Její odvození vychází z následující obecné úlohy prvního řádu [18]:

$$\frac{dy}{dt} = f(x_i, y(x_i)) \quad (1)$$

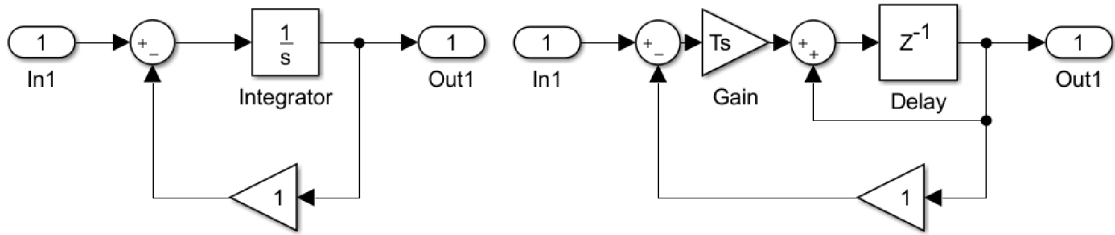
Kde  $x_i$  je pravidelná řada hodnot s krokem  $h$ . Pak lze derivace na levé straně přibližně nahradit diferencí [18]:

$$\frac{y(x_{i+1}) - y(x_i)}{h} \cong f(x_i, y(x_i)) \quad (2)$$

Jednoduchou úpravou lze tento vztah pro derivaci upravit na vztah pro integraci [18]:

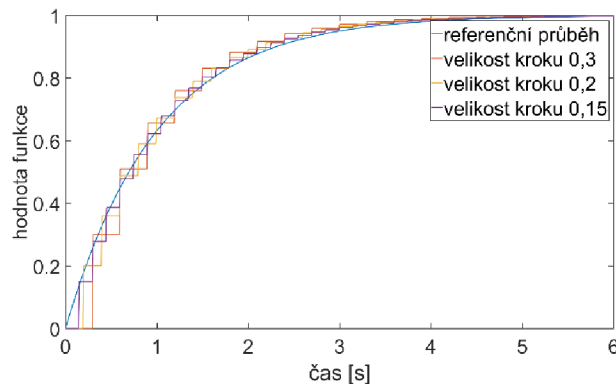
$$y(x_i) \cong hx_i + y(x_{i-1}) \quad (3)$$

Tento vztah lze pak v modelovaných soustavách použít jako náhradu za člen spojité integrace. Chyba této jednoduché metody vůči skutečnému řešení je přímo úměrná velikosti kroku  $h$ . V praxi při uvažování vzorkovací periody  $T_s$  to znamená, že čím je tato perioda vzorkování nižší, tím je nižší i odchylka této metody od skutečné hodnoty. Na Obr. 6 je vyobrazeno simulační schéma v prostředí Matlab/Simulink jednoduchého setrvačného členu prvního řádu, kde místo použití členu spojité integrace je použito náhrady Eulerovou metodou.



Obr. 6: Diskretizace Eulerovou metodou

Při volbě velikosti vzorkovací periody  $T_s$  je nutné dodržet vzorkovací teorém, který zní: vzorkovací frekvence, respektive perioda vzorkování, musí být alespoň dvakrát vyšší, respektive menší, než největší frekvence obsažena v soustavě, respektive perioda. Pro tento případ platí, že perioda vzorkování  $T_s$  musí být menší než hodnota 0,32. na Obr. 7 je znázorněna závislost velikosti kroku  $h$  (periody vzorkování  $T_s$ ) na odchylce od skutečného průběhu výstupu systému z Obr. 6 vlevo.



Obr. 7: Velikost chyby v závislosti na kroku  $h$

Metoda je jednoduchá na implementaci i v méně výkonných systémech a při použití dostatečně malé periody vzorkování  $T_s$ , je tato metoda vhodnou náhradou za výpočet spojité integrace.

### 3.2 Aritmetika výpočtů s pevnou desetinnou čárkou

Aritmetika výpočtů s pevnou desetinnou čárkou využívá proměnné rozdělené na dvě části, kde první část proměnné reprezentuje celočíselnou část hodnoty uložené v proměnné a druhá část pak reprezentuje desetinnou část hodnoty uložené v proměnné. Jako příklad je zde uvedena neznaménková osmi bitová proměnná s pěti bity reprezentující celočíselnou část a třemi bity reprezentující desetinnou část na Obr. 8.

<b>Pozice bitu</b>	<b>8</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>
<b>Váha bitu</b>	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$	$2^{-1}$	$2^{-2}$	$2^{-3}$
<b>Desítková váha bitu</b>	16	8	4	2	1	0,5	0,25	0,125

**Obr. 8: Osmi bitová proměnná s pevnou desetinnou čárkou [19]**

V případě uvažování znaménkové proměnné, se musí nejvyšší bit vyhradit pro signalizaci polarity čísla. Vlastnosti proměnné s pevnou desetinnou čárkou můžeme popsat pomocí dvou parametrů: číselný rozsah a rozlišení [19]. Kde číselný rozsah je dán rozdílem maximální reprezentovatelné hodnoty a minimální reprezentovatelné hodnoty. Pro příklad na Obr. 8 je tato hodnota 37,75. Rozlišení proměnných s pevnou desetinnou čárkou je rovno absolutní hodnotě nejnižší reprezentovatelné hodnoty. Opět v případě příkladu na Obr. 8 se jedná o číslo 0,125. Oproti proměnným uloženým ve formátu s plovoucí desetinnou čárkou mají proměnné s pevnou desetinnou čárkou výhodu v nižších nárocích na výpočetní výkon. Další výhodou oproti proměnným s plovoucí desetinnou čárkou je ta, že proměnné s pevnou desetinnou čárkou mohou mít vyšší rozlišení při stejném počtu bitů na proměnnou, ale samozřejmě v takovém případě nedosáhnou stejného číselného rozsahu. Přibližné rozlišení proměnné s počtem bitů 32 a s plovoucí desetinnou čárkou je  $1,2 \cdot 10^{-7}$ . Kde pak proměnná s pevnou desetinnou čárkou může dosáhnout na stejném počtu bitů rozlišení  $4,7 \cdot 10^{-10}$ . Obě proměnné jsou uvažované jako znaménkové. Nevýhodou proměnných s pevnou desetinnou čárkou je, že je nutné zvolit velikost obou částí, tak aby číselný rozsah a rozlišení vyhovovalo konkrétní aplikaci. V praxi se často používají tzv. Q-formáty (například Q15 a Q31). Tyto formáty pracují s omezeným rozsahem -1 až 1, ale využívají velkého rozlišení. Pro šestnácti bitové proměnné je to  $3,1 \cdot 10^{-5}$  a pro dvaatřicetibitové proměnné je to již zmíněná hodnota  $4,7 \cdot 10^{-10}$ .

Způsobů zápisů proměnných s pevnou desetinnou čárkou existuje několik. Ve zbývajícím textu, této práce bude použit následující způsob: „sCCfDD“ respektive „uCCfDD“. Kde „s“ respektive „u“ symbolizuje, zda se jedná o znaménkovou hodnotu respektive o neznaménkovou. Pak „CC“ je číslo vyjadřující celkový počet bitů proměnné, včetně znaménkového bitu. A nakonec „DD“ je číslo udávající počet bitů reprezentující desetinnou část hodnoty uloženou v proměnné. Pomocí tohoto zápisu by se proměnná z příkladu na Obr. 8 zapsala následovně: „u8f3“.

### 3.3 Použití měřítek v modelování dynamických systémů

Měřítka neboli normy, se v minulosti používaly pro určení vztahů mezi veličinami reálných systémů a signálů jejich modelů na analogovém počítači [18]. Dnes při

použití výpočetní techniky a datových typů s plovoucí desetinnou čárkou, které mají mnohem větší rozsahy hodnot než analogové signály na analogových počítačích, toto ztrácí na významu. Avšak při použití datových typů proměnných s pevnou desetinnou čárkou, je možné pomocí nich dosáhnout maximálního rozlišení proměnné při konkrétní bitové šířce.

Pro popis problematiky měřítek je zde uveden příklad jednoduchého dynamického systému. Jedná se o setrvační člen prvního řádu popsany jednou diferenciální rovnicí:

$$\frac{dx_1}{dt} = x_1 + u \quad (4)$$

Kde  $x_1$  je stavová proměnná, která je zároveň i výstupem systému. Proměnná  $u$  je vstupní veličinou systému. Dále lze definovat číselné konstantní hodnoty  $N_x$  a  $N_u$  takové, že hodnota stavové proměnné  $x_1$  bude vždy nižší než hodnota  $N_x$  a hodnota vstupní veličiny  $u$  bude vždy nižší než hodnota  $N_u$ . Nyní lze definovat bezrozměrné veličiny:

$$\left(\frac{x_1}{N_x}\right), \left(\frac{u}{N_u}\right) \quad (5)$$

Pro tyto veličiny platí, že jejich absolutní hodnoty nepřekročí hodnotu 1. Pro uvedené veličiny lze zavést následující vztahy:

$$x_1 = \frac{x_1}{N_x} N_x, u = \frac{u}{N_u} N_u \quad (6)$$

Dosazením bezrozměrných ekvivalentů za stavové veličiny v rovnici (6) vznikne:

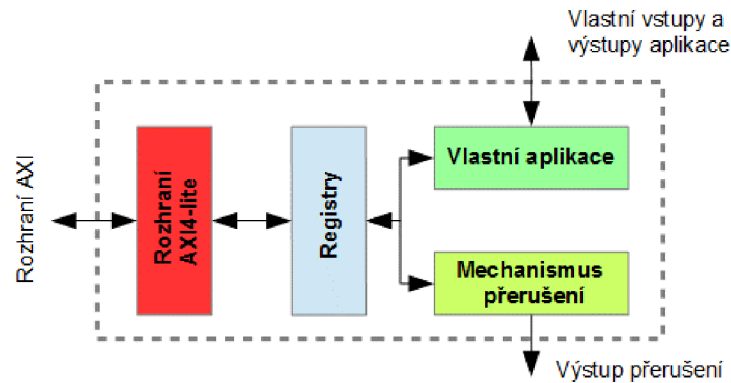
$$\frac{d}{dt} \left(\frac{x_1}{N_x}\right) = \left(\frac{N_x}{N_x}\right) \frac{x_1}{N_x} \quad (7)$$

Tato úprava zajišťuje, že absolutní hodnota žádné z veličin nepřekročí hodnotu 1.

### 3.4 Vytvoření implementace modelu pomocí Vivado HLS

Jak bylo popsáno v kapitole 2, pomocí prostředí Vivado HLS je možné simulovat chování implementované aplikace. V tomto případě modelu dynamického systému. Po úspěšném dokončení (výsledky odpovídají požadavkům) obou simulací (C simulace a Co-simulace) je možné vytvořit *IP core* periférii. Během procesu tvorby periférie prostředí k periférii automaticky přidá *AXI4-lite* sběrnice rozhraní a

definované vstupní parametry, implementované funkce definuje jako paměťově mapované registry. Dále pak vytvoří mechanismus řízení periferie pomocí přidaného řídicího registru. následně přidá mechanismus přerušení a soubor ovladačů pro přístup ke všem registrům a k ovládání mechanismu přerušení. Na Obr. 9 je vidět struktura takto vytvořené *IP core* periferie.



Obr. 9: Struktura *IP core* vytvořené v prostředí Vivado HLS

Takto vytvořenou periferii lze pak použít při tvorbě procesorového systému v prostředí Vivado.

### 3.5 Závěr kapitoly metodika implementace modelů v hradlovém poli FPGA

V této kapitole byly popsány metody použité pro implementaci modelu dynamického systému. Nejdříve zde byla popsána metoda Euelerovy diskretizace a její nedostatky. Následně pak aritmetika výpočtů s proměnnými s pevnou desetinnou čárkou a její výhody a nevýhody oproti aritmetice s plovoucí desetinnou čárkou. Poté zde bylo popsáno použití měřítek v modelování dynamických systému a na závěr zde bylo popsáno jak je konkrétní model implementován pomocí prostředí Vivado HLS.

## 4 IMPLEMENTACE MODELU BLDC MOTORU

V této kapitole jsou popsány základní vlastnosti BLDC motoru. Je zde odvozen jeho matematický model, který je implementován v prostředí Matlab/Simulink. Dále je zde popsáno jak je tento model upraven pro implementaci v obvodu ZYNQ-7000. Jsou zde také srovnány výstupy simulací z obvodu ZYNQ-7000 a prostředí Matlab/Simulink.

### 4.1 Základní popis BLDC motoru

BLDC motor si lze představit jako „převrácený“ stejnosměrný motor s mechanickým komutátorem [19]. Takovýto motor má permanentní magnety na statoru a na rotoru je umístěno vinutí. Motor obsahuje mechanický komutátor, který je tvořen buďto sběrnými lamelami, nebo sběrnými kartáči. Tyto pak zaručují obrácení polarizace magnetického pole tvořeného v rotorovém vinutí tak, aby toto pole působilo proti poli tvořeného permanentními magnety ve statoru. Tímto je zajištěn rotační pohyb stejnosměrného motoru s mechanickým komutátorem. BLDC motor má permanentní magnety na rotoru a stator je tvořen vynutím. Kvůli tomuto „obrácení“ se v označení tohoto typu motoru nechává zkratka „DC“. Funkci mechanického komutátoru v případě BLDC motoru přebírá výkonový tranzistorový střídač, který zajišťuje elektrickou komutaci. Kvůli tomuto se BLDC motory také někdy označují písmeny EC (Electric Commutated) motor, neboli elektricky komutovaný motor. Funkcí výkonového střídače je budit statorová vinutí tak, aby magnetické pole indukované vinutím působilo proti magnetickému poli tvořeného permanentními magnety na rotoru. Pro tuto elektrickou komutaci je nutný snímač polohy rotoru, podle které jsou střídavě buzeny statorová vinutí. Tento snímač polohy je tvořen Hallovými snímači, které indukují zda v jejich blízkosti je magnetické pole o určité polaritě, tedy zda je v jejich blízkosti pól magnetu. Výše popsaný mechanismus pak způsobuje rotační pohyb motoru, podobně jako u stejnosměrného motoru s mechanickým komutátorem.

Velkou výhodou BLDC motoru oproti stejnosměrnému motoru s mechanickým komutátorem je právě fakt, že ke komutaci nedochází pomocí sběrných lamel nebo kartáčů. Protože mezi sběrnými lamelami nebo sběrnými kartáči a rotorem stejnosměrného motoru může docházet k jiskření, nebo dokonce ke vzniku elektrického oblouku na komutátoru. Tyto jevy zhoršují možnost přesného měření elektrického proudu vinutím rotoru a tím pádem zhoršují parametry případného řídicího systému toho typu motoru.



## 4.2 Matematický model BLDC motoru

Pro odvození matematického modelu můžeme uvažovat tyto zjednodušující podmínky: Uvažujeme umístění permanentních magnetů rotoru motoru na povrchu. Průběh magnetické indukce  $B$  ve vzduchové mezeře je pravoúhlý. Průběh spřaženého magnetického toku  $\psi$  je pravoúhlý. Průběh zpětného indukovaného napětí je lichoběžníkový. Dalším zjednodušením je, že reluktanční moment je zanedbán. Poslední zjednodušující podmínkou je, že hodnoty odporu a indukčnosti jednotlivých vinutí jsou navzájem stejné a vzájemné indukčnosti jsou zanedbány [19].

Pro jednotlivá statorová vinutí platí následující vztahy:

$$\begin{aligned}u_x(t) &= Ri_x(t) + \frac{d\psi_x(t)}{dt} \\u_y(t) &= Ri_y(t) + \frac{d\psi_y(t)}{dt} \\u_z(t) &= Ri_z(t) + \frac{d\psi_z(t)}{dt}\end{aligned}\quad (8)$$

Kde  $u_x, u_y, u_z$  jsou fázová napětí,  $i_x, i_y, i_z$  jsou elektrické proudy vinutím,  $\psi_x, \psi_y, \psi_z$  jsou spřažené toky a  $R$  je odpor jednotlivých vinutí. Pro jednotlivé spřažené toky  $\psi_{xyz}$  platí:

$$\begin{aligned}\psi_x(t) &= Li_x + \psi_{Bx}(\theta) \\ \psi_y(t) &= Li_x + \psi_{By}(\theta) \\ \psi_z(t) &= Li_x + \psi_{Bz}(\theta)\end{aligned}\quad (9)$$

Kde  $L$  je indukčnost jednotlivých vinutí. Dále je nutné uvést vztah pro vnitřní magnetický moment:

$$m_i(t) = i_x(t) \frac{d\psi_{Bx}(\theta)}{d\theta} + i_y(t) \frac{d\psi_{By}(\theta)}{d\theta} + i_z(t) \frac{d\psi_{Bz}(\theta)}{d\theta}\quad (10)$$

Pro elektricky komutovaný motor platí pro danou polohu rotoru například:

$$\begin{aligned}i_x &= -i_y = i \\ i_z &= 0 \\ u_x - u_y &= u\end{aligned}\quad (11)$$

Pak lze soustavu rovnic (8) převést do tvaru:

$$u(t) = 2Ri(t) + 2L \frac{di(t)}{dt} + \omega_{el}(t) \frac{d\psi_{Bx}(\theta)}{d\theta} - \omega_{el}(t) \frac{d\psi_{By}(\theta)}{d\theta} \quad (12)$$

Kde  $\omega_{el}$  jsou elektrické otáčky stroje. V pracovní oblasti, kdy platí vztahy (Rovnice 8), mají funkce  $\psi_{Bx}(\theta)$  a  $\psi_{By}(\theta)$  lineární průběh, takže jejich derivace jsou konstanty. Absolutní hodnoty jsou stejné, ale mají opačné znaménko. Tuto absolutní hodnotu můžeme nazvat konstantou stroje  $C_e$ . Výsledný vztah bude vypadat následovně:

$$u(t) = 2Ri(t) + 2L \frac{di(t)}{dt} + 2C_e \omega_{el}(t) \quad (13)$$

Tato rovnice se někdy nazývá elektrickou rovnicí modelu elektricky komutovaného stejnosměrného motoru a popisuje vztah mezi elektrickými otáčkami motoru  $\omega$  a jeho svorkovým napětím  $u$ . Vztah mezi elektrickými a mechanickými otáčkami, je dán počtem pól párů dle následující rovnosti:

$$\omega_{el}(t) = \omega_{mech}(t)P \quad (14)$$

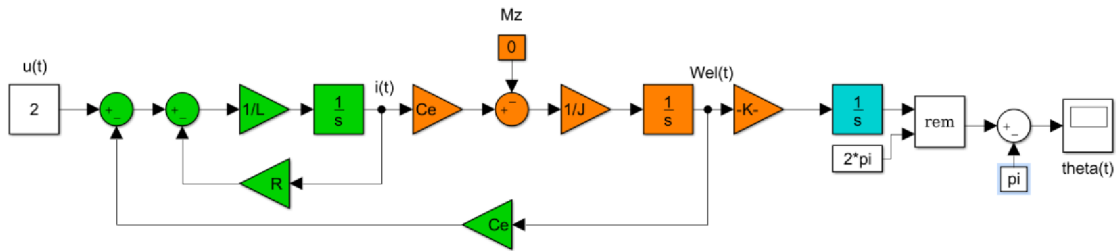
Kde  $P$  je počet pólových dvojic na rotoru. Pro úplnost ještě chybí přidat rovnici popisující mechanickou rovnováhu momentů a vztah mezi mechanickým momentem a elektrickým proudem. Tyto dva vztahy lze zapsat do jedné rovnice:

$$2C_e i(t) = J \frac{d\omega_{el}(t)}{dt} + M_z(t) \quad (15)$$

Kde  $J$  je celkový moment setrvačnosti a  $M_z$  je moment působící zátěže. Tyto dvě rovnice tvoří zjednodušený matematický model BLDC motoru. Model může být doplněn rovnicí vyjadřující mechanickou polohu rotoru.

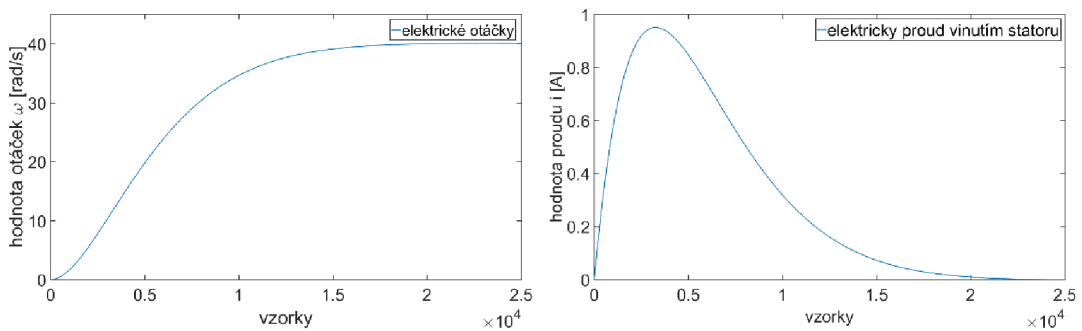
$$\frac{d\theta(t)}{dt} = \frac{\omega_{mech}(t)}{P} \quad (16)$$

Kompletní odvození modelu BLDC motoru lze dohledat v [19]. Na Obr. 10 je vyobrazena implementace modelu v prostředí Matlab/Simulink a na Obr. 11 jsou vyobrazeny průběhy otáček a proudu motoru. Na Obr. 12 je pak vyobrazen průběh mechanické polohy rotoru. Parametry motoru jsou uvedeny v Tab. 1.

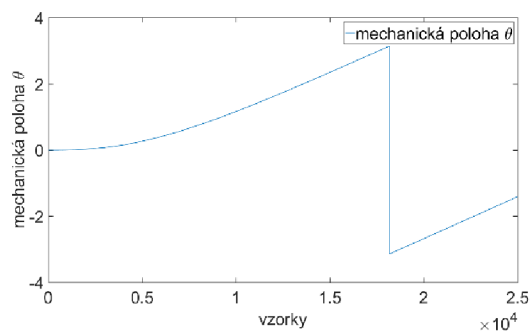


Obr. 10: Model BLDC motoru

Do simulačního schématu byla přidána funkce „zbytek po dělení *REM*“ spolu s odečtením hodnoty  $-\pi$  pro zajištění zobrazení polohy periodicky v rozsahu  $\pm\pi$ . Pro lepší orientaci ve schématu byly určité části modelu barevně označeny. Bloky označené zelenou barvou odpovídají rovnici (13). Bloky zvýrazněné oranžovou barvou, pak odpovídají rovnici (14). Nakonec blok vyznačený světle modrou barvou odpovídá rovnici (16). Simulace modelu byla spuštěna po dobu 0,025 s při periodě vzorkování  $1 \mu\text{s}$ . Pro běh simulace bylo zvoleno numerické metody *ode3 Bogacki-Shampine*.



Obr. 11: Průběh elektrických otáček a proudu vinutím



Obr. 12: Průběh mechanické polohy rotoru

Tab. 1: Parametry motoru

Název parametru	hodnota
Indukčnost $L$	2,6 mH
Odpor vinutí $R$	1,5 $\Omega$

Konstanta stroje $C_e$	0,05 $Wb$
Moment setrvačnosti $J$	$9,6 \cdot 10^{-6} \text{ Kg} \cdot \text{m}^2$
Počet pól párů	2
Velikost budícího napětí $u$	2 $V$

### 4.3 Úprava modelu pro implementaci

Model na Obr. 10 je realizován v prostředí Matlab/Simulink a toto prostředí samo řeší aproximaci za integrace a navíc má k dispozici nástroje pro práci s datovými typy s plavoucí desetinnou čárkou. Při vlastní implementaci je nutné tyto mechanismy určitým způsobem zahrnout.

Pro nahrazení operace integrace byla zvolena Eulerova metoda popsána v kapitole 3.1. Pro nahrazení datových typů byly zvoleny datové typy s pevnou desetinnou čárkou, které jsou popsány v téže kapitole. Pro usnadnění práce s volbou rozsahů celočíselné části proměnných s pevnou desetinnou čárkou, byl model naměřitkován tak, aby výstupy ze všech matematických operací byly v rozsahu  $\pm 1$ . Následující vztahy popisují diskretizovanou formu zjednodušeného modelu BLDC motoru.

$$u(k) = 2Ri(k) + 2L \frac{i(k+1) - i(k)}{T_s} + 2C_e \omega_{el}(k) \quad (17)$$

$$2C_e i(k) = J \frac{\omega_{el}(k+1) - \omega_{el}(k)}{T_s} + M_0(k) \quad (18)$$

$$\frac{\theta(k+1) - \theta(k)}{T_s} = \frac{\omega_{mech}(k)}{P} \quad (19)$$

Nyní následuje skupina těchto rovnic doplněna o měřítka  $N$ :

$$\frac{u(k)}{N_u} = \frac{2Ri(k)}{N_R N_i} \cdot \frac{N_R N_i}{N_u} + \frac{2L (i(k+1) - i(k)) N_{T_s}}{N_L T_s N_i} \cdot \frac{N_L N_i}{N_{T_s} N_u} + \frac{2C_e \omega_{el}(k)}{N_{C_e} N_{\omega_{el}}} \cdot \frac{N_{C_e} N_{\omega_{el}}}{N_u} \quad (20)$$

$$\frac{2C_e i(k)}{N_{C_e} N_i} = \frac{J (\omega_{el}(k+1) - \omega_{el}(k)) N_{T_s}}{N_J T_s N_{\omega_{el}}} \cdot \frac{N_J N_{\omega_{el}}}{N_{T_s} N_{C_e} N_i} + \frac{M_0(k)}{N_M} \cdot \frac{N_M}{N_{C_e} N_i} \quad (21)$$

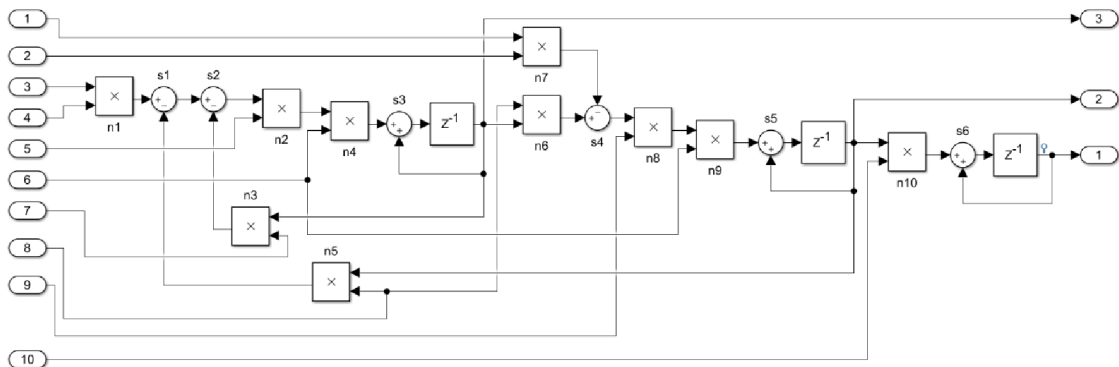
Následující členy jsou bezrozměrné a rovny hodnotě 1. Při zvolení měřítek pro napětí  $N_u$ , proud  $N_i$ , elektrické otáčky  $\omega_{el}$  a periodu vzorkování  $T_s$  je možné odvodit zbývající měřítka.

$$\frac{N_R N_i}{N_u} = 1, \frac{N_L N_i}{N_{T_s} N_u} = 1, \frac{N_{C_e} N_{\omega_{el}}}{N_u} = 1, \frac{N_J N_{\omega_{el}}}{N_{T_s} N_{C_e} N_i} = 1, \frac{N_M}{N_{C_e} N_i} = 1 \quad (22)$$

První tři měřítka mohou být zvolena například na základě katalogového listu výrobce a měřítka periody vzorkování  $N_{T_s}$  se zvolí například tisíckrát větší, než je perioda vzorkování  $T_s$ . Takto odvozená měřítka zároveň slouží jako jakési mantinely pro velikost parametrů motoru, protože vztahy mezi nimi vycházejí z fyzikálních závislostí uvnitř BLDC motoru. Je vhodné uvažovat určitou rezervu mezi měřítkem a skutečným parametrem motoru, protože toto má pak vliv na stabilitu výpočtu, tedy na náchylnost k přetékání proměnných. To je také důvod proč je hodnota měřítka periody vzorkování  $N_{T_s}$  volena vyšší než je samotná perioda vzorkování  $T_s$ . Následující vztahy ukazují odvození ostatních měřítek z bezrozměrných členů.

$$N_R = \frac{N_u}{N_i}, N_L = \frac{N_{T_s} N_u}{N_i}, N_{C_e} = \frac{N_u}{N_{\omega_{el}}}, N_J = \frac{N_{T_s} N_{C_e} N_i}{N_{\omega_{el}}}, N_M = N_{C_e} N_i \quad (23)$$

Pro rovnici vyjadřující mechanickou polohu rotoru není třeba takto vyjadřovat měřítka. Tato veličina pracuje v rozsahu  $\pm\pi$ , a proto je její měřítko  $2\pi$ . Na Obr. 13 je přepracované modelovací schéma BLDC motoru z Obr. 10. Všechny parametry modelu jsou realizované jako vstupy do modelu. Toto je kvůli lepšímu srovnání s výslednou implementací, kde model bude implementován jako funkce a všechny parametry modelu budou tvořit parametry funkce. To platí i pro výstupy modelu. Všechny parametry funkce jsou realizovány jako ukazatele.



Obr. 13: Přepracované modelovací schéma BLDC motoru

V následující tabulce jsou popsány vstupní parametry modelu včetně zvoleného datového typu ve formátu s pevnou desetinnou čárkou.

**Tab. 2: Vstupní parametry modelu BLDC motoru**

Číslo parametru	Označení	Datový typ
1	$M_z$	s18f15
2	$1/N_M$	s18f17
3	u	s18f10
4	$1/N_u$	s18f17
5	$N_L/L$	u18f18
6	$T_s/N_{T_s}$	u18f18
7	$R/N_R$	u18f18
8	$C_e/N_{C_e}$	u18f18
9	$N_J/J$	u18f18
10	$2T_s/(N_{T_s}\pi)$	u18f18

Následující tabulka popisuje výstupní datový typ jednotlivých matematických operací uvnitř modelovacího schématu BLDC motoru. Jednotlivé operace jsou popsány indexy  $n$  v případě násobení a indexem  $s$  v případě sčítání.

**Tab. 3: Výstupní datový matematických operací modelovacího schématu BLDC motoru**

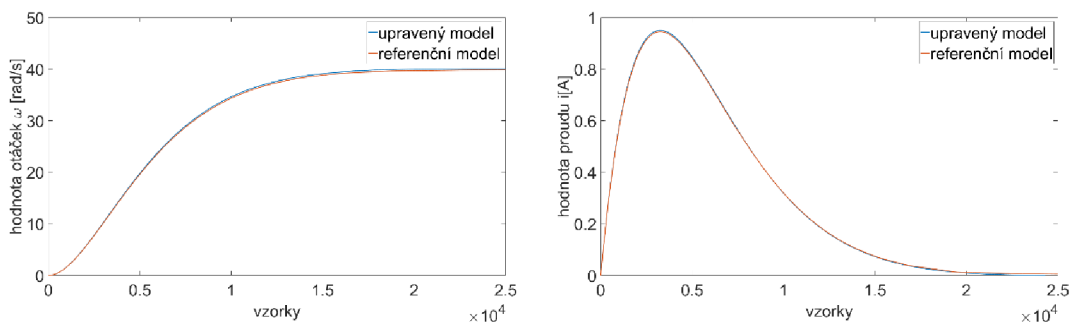
Číselné označení	Násobení/sčítání	Datový typ
n1	násobení	s18f17
n2	násobení	s25f24
n3	násobení	s25f24
n4	násobení	s25f24
n5	násobení	s25f24
n6	násobení	s18f17
n7	násobení	s18f17
n8	násobení	s25f24
n9	násobení	s25f24
n10	násobení	s25f24
s1	sčítání	s18f17
s2	sčítání	s18f17
s3	sčítání	s25f24
s4	sčítání	s18f17
s5	sčítání	s25f24
s6	sčítání	s25f24

Poslední tabulka. popisuje výstupní parametry, podobně jako Tab. 2.

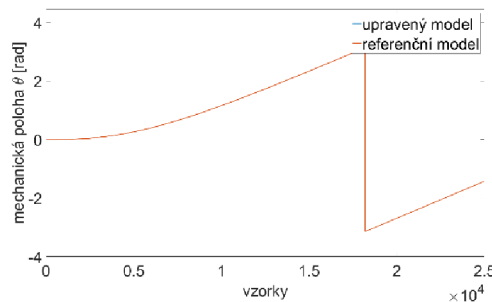
**Tab. 4: Výstupní parametry modelu BLDC motoru**

Číslo parametru	Označení	Datový typ
1	$\theta_{mech}$	s25f24
2	$\omega_{el}$	s25f24
3	i	s25f24

Na následujících obrázcích (Obr. 14, Obr. 15) jsou zobrazeny průběhy původního referenčního modelu a upraveného modelu BLDC motoru.



**Obr. 14: Srovnání otáček a proudu mezi referenčním a upraveným modelem**



**Obr. 15: Srovnání mechanické polohy mezi referenčním a upraveným modelem**

Rozdíly mezi průběhy jsou dány jednak chybou aproximace pomocí Eulerovy metody a jednak snížením přesnosti použitím datových typů s pevnou desetinnou čárkou, hlavně použitím s18f17 datového typu.

#### 4.4 Implementace modelu v ZYNQ-7000

Pro implementaci modelu BLDC motoru je nutné přetransformovat model do podoby kódu v C/C++ jazyce, který podporuje prostředí Vivado HLS. Nabízí se i možnost použít vestavěný překladač v prostředí Matlab/Simulink, ale výstupní kód je hůře čitelný pro člověka, a navíc kód popisující chování modelu není nijak složitý na vytvoření. Je možné jej dohledat v elektronické příloze této práce.

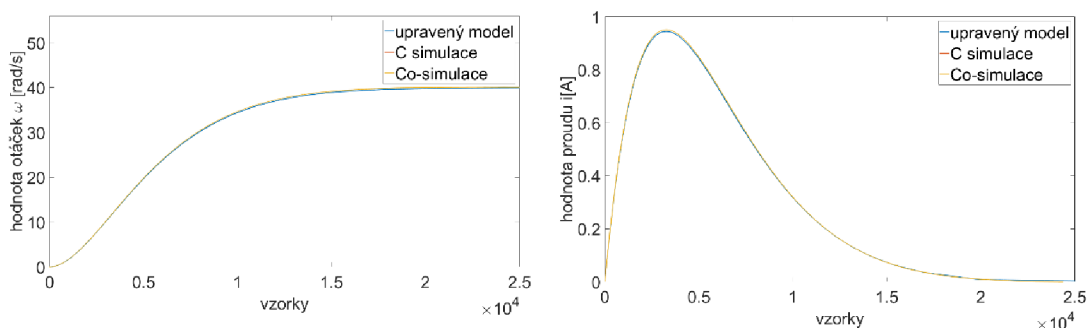
Funkce v jazyce C/C++ obsahuje výpočet třech stavových veličin dle rovnic (20), (21) a (16). Dále se počítá s implementací kódu do podoby *IP core* periferie připojitelné k jádru ARM Cortex-A9 pomocí sběrnice rozhraní AXI4-lite. Funkce také zahrnuje přepínací mechanismus, který je ovládán skrze registr přístupný přes sběrnice rozhraní jádra procesoru. Tento přepínací mechanismus přepíná to, zda mají být vstupní data (vstupní napětí  $u$  a zátěžný moment  $M_z$ ) načítána pro výpočet ze sběrnice rozhraní, a tedy poskytnuta procesorem nebo skrze fyzické porty periferie, a tedy poskytnuta z jiné periferie, která je přímo připojena k periférii reprezentující model BLDC motoru.

Následující Tab. 5 udává procentuální vytíženost prostředků uvnitř FPGA části obvodu *xc7z020clg484-1* a časovou náročnost na jednu iteraci výpočtu periferie zobrazenou jako počet period hodinového signálu (latence). Frekvence hodinového signálu je  $100\text{MHz}$ . Tyto údaje byly vyčteny přímo z prostředí Vivado HLS.

**Tab. 5: Hodnocení implementace modelu BLDC motoru v prostředí Vivado HLS**

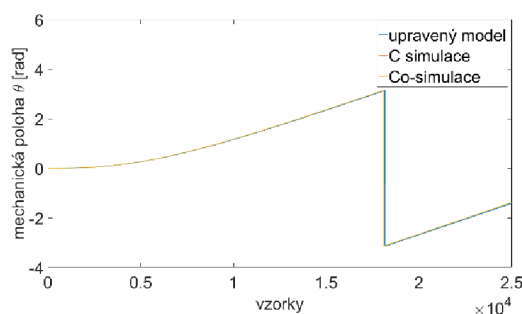
Údaj	hodnocení
Vytíženost LUT	2%
Vytíženost DFF	1%
Vytíženost DSP	9%
Latence	18

Jak už bylo popsáno v kapitole 2.6, prostředí Vivado HLS umožňuje dva typy simulací pro otestování chování implementované aplikace. První je C simulace a další je Co-simulace. Na Obr. 16 a Obr. 17 jsou zobrazeny průběhy stejných veličin jako na Obr. 14 a Obr. 15, ale jedná se o srovnání výstupů z C simulace, Co-simulace a výstupu z upraveného modelu (diskretizace, měřítko, a proměnné s pevnou desetinnou čárkou).



**Obr. 16: Srovnání otáček a proudu mezi upraveným modelem a simulacemi v prostředí Vivado HLS**

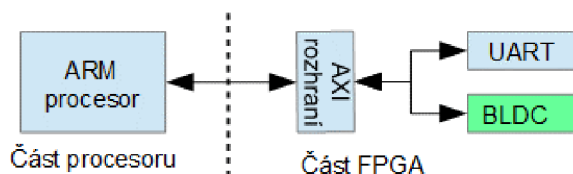




**Obr. 17: Srovnání mechanické polohy mezi upraveným modelem a simulacemi v prostředí Vivado HLS**

Při pohledu na srovnání průběhů je vidět, že simulace se navzájem shodují, ale jsou odlišné od výstupů z upraveného modelu. Tento rozdíl je dán rozdílnou aritmetikou operací s proměnnými s pevnou desetinnou čárkou v prostředí matlab/Simulink a prostředí Vivado HLS.

Po vytvoření *IP core* periferie s AXI4-lite rozhraním a SW ovladači byl vytvořen v prostředí Vivado procesorový systém, znázorněný na následujícím diagramu (Obr. 18), kde blok *BLDC* reprezentuje *IP core* periferii modelu BLDC motoru.



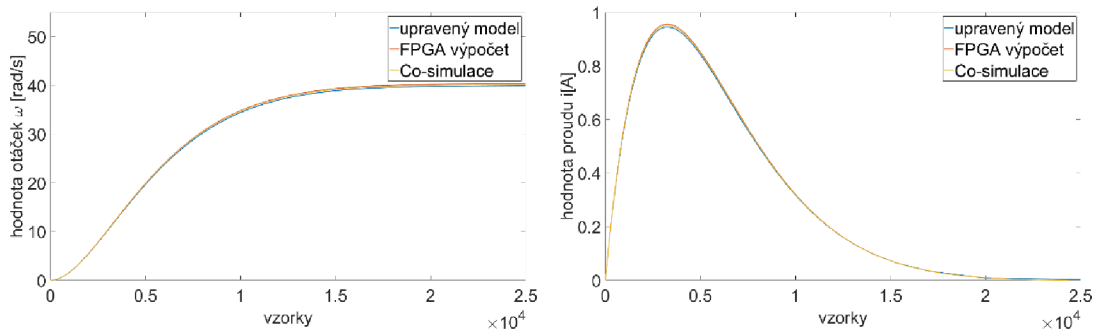
**Obr. 18: Diagram procesorového systému pro testování modelu BLDC motoru**

Sériové rozhraní UART (*Universal Asynchronous Receiver Transmitter*) zde slouží pro záznam dat ze simulace. Při implementaci tohoto procesorového systému v prostředí Vivado byly aplikovány další optimalizace pro spotřebu prostředků FPGA části. Tyto optimalizace byly v režii prostředí Vivado. V Tab. 6 je znázorněna procentuální vytíženost FPGA obvodu při implementaci celkového procesorového systému (včetně sběrnice rozhraní AXI a periferie UART).

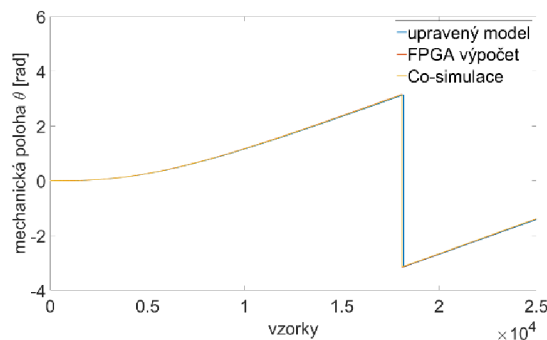
**Tab. 6: Hodnocení implementace procesorového systému v prostředí Vivado**

Údaj	hodnocení
Vytíženost LUT	2,5%
Vytíženost DFF	2%
Vytíženost DSP	9%

V procesoru ARM Cortex-A9 byla implementována aplikace pro spouštění jednotlivých kroků výpočtu modelu a to pro stejný počet vzorků jako modely v prostředí Matlab/Simulink. Při tomto nebylo použito žádné časování (například časování čítačem). Po skončení simulace byla data porovnána s výstupem upraveného modelu a výstupem Co-simulace. Výsledky jsou zobrazeny na Obr. 19 a Obr. 20.



**Obr. 19: Srovnání otáček a proudu mezi upraveným modelem, Co-simulací v prostředí Vivado HLS a výpočtem na FPGA části obvodu ZYNQ-7000**



**Obr. 20: Srovnání mechanické polohy mezi upraveným , Co-simulací v prostředí Vivado HLS a výpočtem na FPGA části obvodu ZYNQ-7000**

## 4.5 Závěr kapitoly implementace modelu BLDC motoru

V této kapitole byla popsána implementace modelu BLDC motoru v FPGA části obvodu ZYNQ-7000. Bylo poukázáno na chyby výpočtu při diskretizaci modelu pomocí Eulerovy metody. Dále bylo ukázáno na rozdílnost aritmetik pro operace s proměnnými s pevnou desetinnou čárkou v prostředích Vivado HLS a Matlab/Simulink. A v poslední části bylo poukázáno na rozdílnost mezi výpočty v prostředí Matlab/Simulink, Vivado HLS a výpočtem na FPGA části obvodu ZYNQ-7000.

## 5 IMPLEMENTACE MODELU ASYNCHRONNÍHO MOTORU

V této kapitole je popsána implementace modelu asynchronního motoru v  $\alpha\beta$  rovině. Nejdříve je zde popsán asynchronní motor a jeho vlastnosti. Dále je zde uveden matematický model, který slouží jako předloha pro tvorbu *IP core* periferie. Následně je zde popsáno, jak je model upraven pro potřeby implementace a na závěr je zde popsán postup tvorby periferie. Kapitola je doplněna výsledky jednotlivých kroků implementace modelu a jsou zde srovnány výstupy z referenčního modelu a výsledného modelu implementovaného na obvodu ZYNQ-7000.

### 5.1 Základní popis asynchronního motoru

Asynchronní motor je třífázový indukční stroj pracující pod synchronní rychlostí při motorickém režimu a nad synchronní rychlostí při generátorovém režimu. Z pohledu nákladů na výrobu jsou tyto stroje levnější než synchronní stroje. Díky velkému konstrukčnímu rozsahu pracovních výkonů (jednotky wattů až stovky kilowatt) a nízkým nárokům na údržbu jsou často používány v průmyslu. Nicméně regulace otáček těchto motorů je složitější než regulace rychlosti stejnosměrných motorů. Další nevýhodou je to, že hodnoty startovacích proudů bývají šestkrát, nebo i osmkrát vyšší než hodnoty proudů při jejich maximálním zatížení. Při nízkém zatížení tyto motory také dosahují horší výkonové účinnosti [20].

Konstrukce motoru se skládá z nehybného statoru a otáčivého rotoru. Stator je složen z cylindrického magnetického jádra. Toto jádro je složeno z tenkých ocelových plátů pro potlačení vířivých ztrát. Dále jsou v tomto magnetickém jádru vyhotoveny drážky pro uložení vinutí. Vinutí bývá zapojeno do trojúhelníkového zapojení ( $\Delta$ ) nebo do hvězdicového zapojení ( $Y$ ). Nejčastější materiál pro vinutí je měď. Rotor motoru bývá dvou různých typů: vinutý a klecový. První typ je tvořen vinutím uložených v drážkách podobně jako u statoru. Počet těchto drážek musí být neceločíselným násobek počtu drážek ve statoru. Aby nedocházelo k magnetickému semknutí „zubů“ na statoru a rotoru při rozběhu motoru. Vinutí motoru jsou vyvedena ze stroje pomocí kroužků nebo kartáčů. Druhý typ rotoru je tvořen klecí z kovového, oválného disku, který je nasunut na střed hřídele motoru. V tomto disku jsou drážky, ve kterých jsou uloženy kovové tyče (měď, mosaz). Tyto tyče jsou na svých koncích spojeny zkratovacími kruhy tyto tyče bývají rovnoběžné s osou motoru nebo i zkosené po obvodu klece a vůči ose motoru. Toto

zkosení má za následek různé velikosti proudů v sousedních tyčích a to má za následek vyšší moment než případ s rovnoběžnými tyčemi [21].

Magnetický statorový tok asynchronního motoru není pevně spřažen s magnetickým tokem rotoru. To znamená, že točivé magnetické pole statoru se točí asynchronně a s jinými otáčkami než rotor. Proto se tomuto motoru říká asynchronní. Asynchronní motor pracuje se skluzem  $s$ , který nabývá hodnot 0 až 1 a je dán poměrným rozdílem otáček dle následujícího vztahu:

$$s = \frac{\omega_s - \omega_r}{\omega_s} \quad (24)$$

Kde  $\omega_s$  jsou otáčky magnetického pole ve statoru (synchronní rychlost) a  $\omega_r$  jsou otáčky rotoru.

## 5.2 Matematický model asynchronního motoru

Pro modelování asynchronních motorů existuje velké množství přístupů. Záleží na použitém náhradním obvodu  $(\pi, T, \Gamma)$  [22]. A také zda je model odvozen v trojfázově soustavě, nebo je použito transformací do dvoufázových rovin [23]. V tomto případě také záleží na tom, zda byla transformace provedena do rotující roviny (transformace  $dq0$ ), nebo do stacionární roviny (transformace  $\alpha\beta0$ ). Dalším problémem je validace modelu asynchronního motoru. Model uvažuje indukčnost a elektrický odpor v rotoru, žel tyto parametry jsou v praxi těžko měřitelné.

V tomto textu bude uveden  $T$  model v dvoufázové stacionární rovině  $(\alpha\beta0)$ . Elektrická část modelu je popsána čtyřmi rovnicemi. Dvě vyjadřují změny proudů ve statoru v jednotlivých složkách  $(\alpha\beta)$  a další dvě vyjadřují změny toků v rotoru v jednotlivých složkách.

$$\frac{di_{s\alpha}(t)}{dt} = \frac{u_\alpha(t)}{K_l} - \frac{K_r}{K_l} i_{s\alpha}(t) + \frac{L_m R_r}{L_r^2 K_l} \psi_{r\alpha}(t) + \frac{L_m}{L_r K_l} \psi_{r\beta}(t) \omega_{el}(t) \quad (25)$$

$$\frac{di_{s\beta}(t)}{dt} = \frac{u_\beta(t)}{K_l} - \frac{K_r}{K_l} i_{s\beta}(t) + \frac{L_m R_r}{L_r^2 K_l} \psi_{r\beta}(t) - \frac{L_m}{L_r K_l} \psi_{r\alpha}(t) \omega_{el}(t) \quad (26)$$

$$\frac{d\psi_{r\alpha}(t)}{dt} = \frac{L_m}{T_r} i_{s\alpha}(t) - \frac{1}{T_r} \psi_{r\alpha}(t) - \psi_{r\beta}(t) \omega_{el}(t) \quad (27)$$

$$\frac{d\psi_{r\beta}(t)}{dt} = \frac{L_m}{T_r} i_{s\beta}(t) - \frac{1}{T_r} \psi_{r\beta}(t) + \psi_{r\alpha}(t) \omega_{el}(t) \quad (28)$$

Kde  $i_{s\alpha}$  a  $i_{s\beta}$  jsou proudy ve statorovém vinutí v jednotlivých složkách.  $u_\alpha$  a  $u_\beta$  jsou fázová napětí na jednotlivých složkách.  $L_m$  je vzájemná indukčnost mezi státorem a rotorem a je shodná pro obě složky.  $L_r$  je pak indukčnost rotoru a je taktéž společná pro obě složky. Pro odpor rotoru  $R_r$  platí to samé. Spřažené toky  $\psi_{r\alpha}$  a  $\psi_{r\beta}$  jsou toky v rotoru pro jednotlivé složky. Veličina  $\omega_{el}$  jsou elektrické otáčky asynchronního motoru. Mezi elektrickými otáčkami a mechanickými otáčkami asynchronního motoru platí stejný vztah jako v případě BLDC motoru, tedy:

$$\omega_{el}(t) = \omega_{mech}(t)P \quad (29)$$

Kde  $P$  značí počet pólových dvojic. Pro zjednodušení zápisu rovnic bylo použito substitucí  $K_l$ ,  $K_r$  a  $T_r$ . Jejich význam je dán následujícími rovnicemi:

$$K_r = R_s + \frac{L_m^2}{L_r^2} R_r \quad (30)$$

$$K_l = L_s - \frac{L_m^2}{L_r} \quad (31)$$

$$T_r = \frac{L_r}{R_r} \quad (32)$$

Kde parametr  $L_s$  je indukčnost statorového vinutí a je společná pro obě složky. Substitute  $K_r$  vyjadřuje odpor vinutí statoru ovlivněný rotorem. Substitute  $K_l$  vyjadřuje indukčnost statoru ovlivněnou rotorem. A substitute  $T_r$  pak vyjadřuje časovou konstantu obvodu rotoru. Následující rovnice popisuje vnitřní magnetický moment motoru:

$$m_i(t) = \frac{3PL_m}{2JL_r} (i_{s\beta}(t)\psi_{r\alpha}(t) - i_{s\alpha}(t)\psi_{r\beta}(t)) \quad (33)$$

Kde  $J$  je moment setrvačnosti motoru. Při uvažování následující rovnice rovnováhy momentů:

$$m_i(t) = J \frac{d\omega(t)}{dt} + M_z(t) \quad (34)$$

Kde  $M_z$  je moment zátěže. Dále dosazením za vnitřní magnetický moment  $m_i$  a úpravou, vzniká vztah pro změnu elektrických otáček stroje:

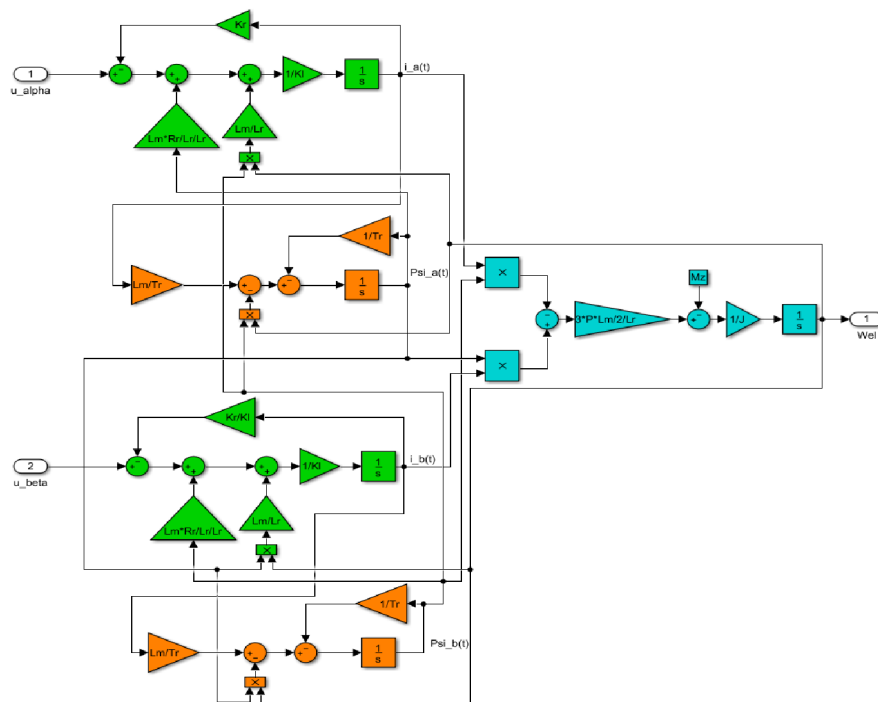
$$\frac{d\omega_{el}(t)}{dt} = \frac{3PL_m}{2JL_r} \left( i_{s\beta}(t)\psi_{r\alpha}(t) - i_{s\alpha}(t)\psi_{r\beta}(t) \right) - \frac{1}{J} M_z(t) \quad (35)$$

Těchto pět rovnic (25), (26), (27), (28) a (35) tvoří matematický model asynchronního motoru v dvoufázové, stacionární rovině  $\alpha\beta$ . Kompletní odvození modelu lze dohledat v [24]. Na Obr. 21 je implementace modelu v prostředí Matlab/Simulink. Pro přehlednost byly jednotlivé části modelu barevně odlišeny. Zeleně jsou části, které jsou popsány rovnicemi (25) a (26). Oranžově pak části popsány rovnicemi (27) a (28). Poslední světlemodrá část je pak popsána rovnicí (29). V Tab. 7 jsou popsány parametry modelu konkrétního asynchronního motoru.

**Tab. 7: Parametry modelu asynchronního motoru**

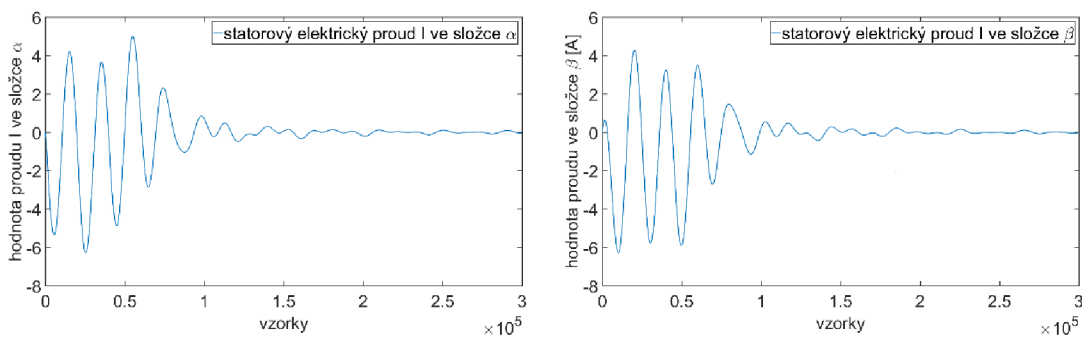
Název parametru	Hodnota parametru
Statorová indukčnost $L_s$	0,1192 H
Rotorová indukčnost $L_r$	0,1181 H
Vzájemná indukčnost $L_m$	0,112 H
Odpor vinutí na statoru $R_s$	0,894 $\Omega$
Odpor vinutí na rotoru $R_r$	0,85 $\Omega$
Počet pólových dvojic $P$	2
Moment setrvačnosti $J$	$2,5 \cdot 10^{-4} \text{ kg} \cdot \text{m}^2$

Simulace byla spuštěna s periodou vzorkování  $T_s \cdot 10^{-6} \text{ s}$  a jako numerická metoda pro výpočet integrací v modelu byla zvolena metoda *ode3 Bogacki-Shampine*.

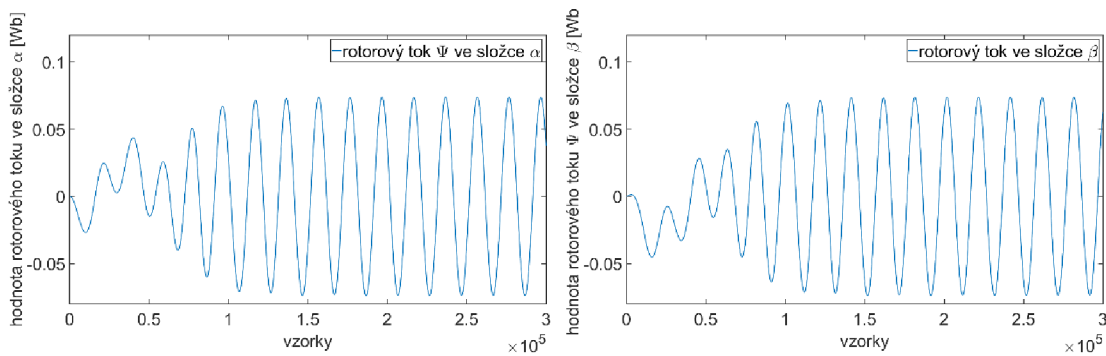


Obr. 21: Model asynchronního motoru v  $\alpha\beta$  rovině

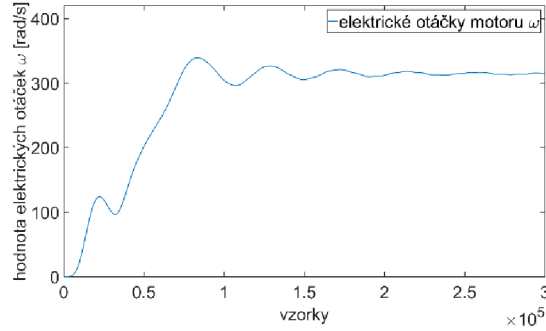
Na Obr. 22 jsou zobrazeny průběhy statorových proudů v složkách  $\alpha$  a  $\beta$ . Obr. 23 ukazuje průběhy magnetických toků v rotoru, také v obou složkách. Konečně pak Obr. 24 ukazuje průběhy elektrických otáček.



Obr. 22: Průběhy statorových proudů v  $\alpha\beta$  složkách modelu asynchronního motoru



Obr. 23: Průběhy rotorových toků v  $\alpha\beta$  složkách modelu asynchronního motoru



Obr. 24: Průběh elektrických otáček modelu asynchronního motoru

### 5.3 Úprava modelu pro implementaci

Podobně jako v kapitole 4.3 je nutné diskretizovat model pomocí Eulerovy aproximace a doplnit model o měřítka tak, aby číselný rozsah všech matematických operací uvnitř modelu byl v rozsahu  $\pm 1$ . Následující vztahy popisují diskretizovanou formu modelu asynchronního motoru:

$$\begin{aligned} \frac{i_{s\alpha}(k+1) - i_{s\alpha}(k)}{T_s} &= \frac{u_\alpha(k)}{K_l} - \frac{K_r}{K_l} i_{s\alpha}(k) + \frac{L_m R_r}{L_r^2 K_l} \psi_{r\alpha}(k) + \frac{L_m}{L_r K_l} \psi_{r\beta}(k) \omega_{el}(k) \end{aligned} \quad (36)$$

$$\begin{aligned} \frac{i_{s\beta}(k+1) - i_{s\beta}(k)}{T_s} &= \frac{u_\beta(k)}{K_l} - \frac{K_r}{K_l} i_{s\beta}(k) + \frac{L_m R_r}{L_r^2 K_l} \psi_{r\beta}(k) - \frac{L_m}{L_r K_l} \psi_{r\alpha}(k) \omega_{el}(k) \end{aligned} \quad (37)$$

$$\frac{\psi_{r\alpha}(k+1) - \psi_{r\alpha}(k)}{T_r} = \frac{L_m}{T_r} i_{s\alpha}(k) - \frac{1}{T_r} \psi_{r\alpha}(k) - \psi_{r\beta}(k) \omega_{el}(k) \quad (38)$$

$$\frac{\psi_{r\beta}(k+1) - \psi_{r\beta}(k)}{T_r} = \frac{L_m}{T_r} i_{s\beta}(k) - \frac{1}{T_r} \psi_{r\beta}(k) + \psi_{r\alpha}(k) \omega_{el}(k) \quad (39)$$

$$\frac{\omega_{el}(k+1) - \omega_{el}(k)}{T_s} = \frac{3PL_m}{2JL_r} (i_{s\beta}(k) \psi_{r\alpha}(k) - i_{s\alpha}(k) \psi_{r\beta}(k)) - \frac{1}{J} M_z(k) \quad (40)$$



Nyní následuje skupina těchto rovnic doplněna o měřítka  $N$ :

$$\begin{aligned}
& \frac{i_{s\alpha}(k+1) - i_{s\alpha}(k)}{T_s} \cdot \frac{N_{T_s}}{N_i} \\
&= \frac{u_\alpha(k)N_{K_l}}{K_l N_u} \cdot \frac{N_u N_{T_s}}{N_{K_l} N_i} - \frac{K_r N_{K_l} i_{s\alpha}(k)}{K_l N_{K_r} N_i} \cdot \frac{N_{K_r} N_i N_{T_s}}{N_{K_l} N_i} \\
&+ \frac{L_m R_r \psi_{r\alpha}(k) N_L^2 N_{K_l}}{L_r^2 K_l N_L N_R N_\psi} \cdot \frac{N_L N_R N_\psi N_{T_s}}{N_L^2 N_{K_l} N_i} + \frac{L_m \psi_{r\beta}(k) \omega_{el}(k) N_L N_{K_l}}{L_r K_l N_L N_\psi N_\omega} \\
&\cdot \frac{N_L N_\psi N_\omega N_{T_s}}{N_L N_{K_l} N_i}
\end{aligned} \tag{41}$$

$$\begin{aligned}
& \frac{i_{s\beta}(k+1) - i_{s\beta}(k)}{T_s} \cdot \frac{N_{T_s}}{N_i} \\
&= \frac{u_\beta(k)N_{K_l}}{K_l N_u} \cdot \frac{N_u N_{T_s}}{N_{K_l} N_i} - \frac{K_r N_{K_l} i_{s\beta}(k)}{K_l N_{K_r} N_i} \cdot \frac{N_{K_r} N_i N_{T_s}}{N_{K_l} N_i} \\
&+ \frac{L_m R_r \psi_{r\beta}(k) N_L^2 N_{K_l}}{L_r^2 K_l N_L N_R N_\psi} \cdot \frac{N_L N_R N_\psi N_{T_s}}{N_L^2 N_{K_l} N_i} - \frac{L_m \psi_{r\alpha}(k) \omega_{el}(k) N_L N_{K_l}}{L_r K_l N_L N_\psi N_\omega} \\
&\cdot \frac{N_L N_\psi N_\omega N_{T_s}}{N_L N_{K_l} N_i}
\end{aligned} \tag{42}$$

$$\begin{aligned}
& \frac{\psi_{r\alpha}(k+1) - \psi_{r\alpha}(k)}{T_s} \cdot \frac{N_{T_s}}{N_\psi} \\
&= \frac{L_m N_{T_r} i_{s\alpha}}{T_r N_L N_i} \cdot \frac{N_L N_i N_{T_s}}{N_{T_r} N_\psi} - \frac{N_{T_r} \psi_{r\alpha}(k)}{T_r N_\psi} \cdot \frac{N_\psi N_{T_s}}{N_{T_r} N_\psi} - \frac{\psi_{r\beta}(k) \omega_{el}(k)}{N_\psi N_\omega} \\
&\cdot \frac{N_\psi N_\omega N_{T_s}}{N_\psi}
\end{aligned} \tag{43}$$

$$\begin{aligned}
& \frac{\psi_{r\beta}(k+1) - \psi_{r\beta}(k)}{T_s} \cdot \frac{N_{T_s}}{N_\psi} \\
&= \frac{L_m N_{T_r} i_{s\beta}(k)}{T_r N_L N_i} \cdot \frac{N_L N_i N_{T_s}}{N_{T_r} N_\psi} - \frac{N_{T_r} \psi_{r\beta}(k)}{T_r N_\psi} \cdot \frac{N_\psi N_{T_s}}{N_{T_r} N_\psi} + \frac{\psi_{r\alpha} \omega_{el}(k)}{N_\psi N_\omega} \\
&\cdot \frac{N_\psi N_\omega N_{T_s}}{N_\psi}
\end{aligned} \tag{44}$$

$$\begin{aligned}
& \frac{\omega_{el}(k+1) - \omega_{el}(k)}{T_s} \cdot \frac{N_{T_s}}{N_\omega} \\
&= \frac{3PL_m \left( i_{s\beta}(k)\psi_{r\alpha}(k) - i_{s\alpha}(k)\psi_{r\beta}(k) \right) N_J N_L}{2JL_r N_L N_i N_\psi} \cdot \frac{N_L N_i N_\psi N_{T_s}}{N_J N_L N_\omega} \\
&- \frac{M_z N_J}{J N_M} \cdot \frac{N_M N_{T_s}}{N_J N_\omega}
\end{aligned} \tag{45}$$

Následující členy jsou bezrozměrné a rovny hodnotě 1. Při zvolení měřítek pro napětí  $N_u$ , proud  $N_i$ , elektrické otáčky  $\omega_{el}$  a periodu vzorkování  $T_s$  je možné odvodit zbývající měřítka.

$$\begin{aligned}
\frac{N_u N_{T_s}}{N_{K_l} N_i} = 1, \frac{N_{K_r} N_i N_{T_s}}{N_{K_l} N_i} = 1, \frac{N_L N_R N_\psi N_{T_s}}{N_L^2 N_{K_l} N_i} = 1, \frac{N_L N_\psi N_\omega N_{T_s}}{N_L N_{K_l} N_i} = 1, \frac{N_L N_i N_{T_s}}{N_{T_r} N_\psi} = 1, \\
\frac{N_\psi N_{T_s}}{N_{T_r} N_\psi} = 1, \frac{N_\psi N_\omega N_{T_s}}{N_\psi} = 1, \frac{N_L N_i N_\psi N_{T_s}}{N_J N_L N_\omega} = 1, \frac{N_M N_{T_s}}{N_J N_\omega} = 1
\end{aligned} \tag{46}$$

Některé bezrozměrné členy se nacházejí v soustavě rovnic dvakrát, ale díky stejnému rozměru je možné je vyčíslit pouze jednou a používat je ve všech rovnicích stejně. Následující vztahy ukazují odvození ostatních měřítek bezrozměrných členů.

$$\begin{aligned}
N_{K_l} = \frac{N_u N_{T_s}}{N_i}, N_{K_r} = \frac{N_{K_l}}{N_{T_s}}, N_\psi = \frac{N_L N_{K_l} N_i}{N_R N_{T_s}}, N_\omega = \frac{N_{K_l} N_i}{N_\psi N_{T_s}}, N_{T_r} = N_{T_s}, N_J \\
= \frac{N_i N_\psi N_{T_s}}{N_\omega}, N_M = \frac{N_J N_\omega}{N_{T_s}}
\end{aligned} \tag{47}$$

Podle kapitoly 3.3 jsou měřítka vázána na jednotku, tedy že měřítko pro jednu veličinu lze použít pro jinou veličinu se stejnou jednotkou [23], nepřekročí-li tato veličina maximální hodnotu danou tímto měřítkem. Podle tohoto by mělo například platit:

$$N_{K_l} = N_L, N_{K_r} = N_R \tag{48}$$

Důkaz tohoto tvrzení pro měřítko  $N_L$  vychází z následující rovnosti:

$$\frac{K_l}{N_{K_l}} = \frac{L_s}{N_L} \cdot \frac{N_L}{N_{K_l}} - \frac{L_m^2 N_L}{L_r N_L^2} \cdot \frac{N_L^2}{N_L N_{K_l}} \tag{49}$$

Z toho pro bezrozměrný člen  $\frac{N_L}{N_{K_l}}$  jasně plyne:

$$\frac{N_L}{N_{K_l}} = 1 \Rightarrow N_L = N_{K_l} \quad (50)$$

A následně pro  $N_R$  platí:

$$\frac{K_r}{N_{K_r}} = \frac{R_s}{N_R} \cdot \frac{N_R}{N_{K_r}} + \frac{L_m^2 R_r N_L^2}{L_r^2 N_L^2 N_R} \cdot \frac{N_L^2 N_R}{N_L^2 N_{K_r}} \quad (51)$$

Z toho pro bezrozměrný člen  $\frac{N_R}{N_{K_r}}$  jasně plyne:

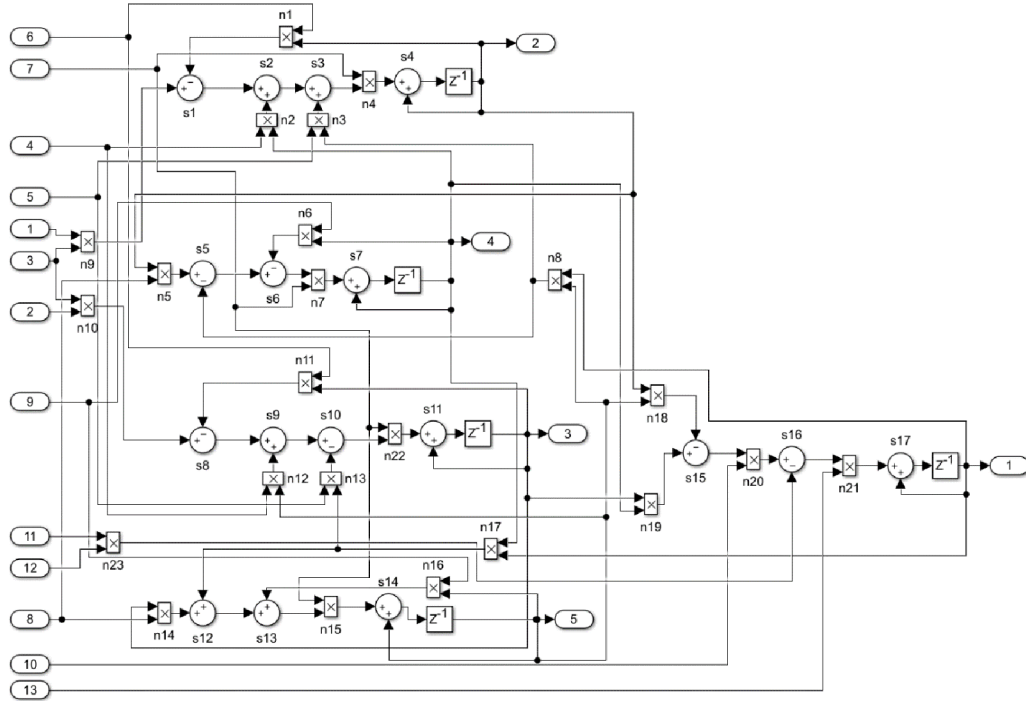
$$\frac{N_R}{N_{K_r}} = 1 \Rightarrow N_R = N_{K_r} \quad (52)$$

Na Obr. 25 je přepracované modelovací schéma asynchronního motoru v  $\alpha\beta$  rovině z Obr. 21. Všechny parametry modelu jsou realizované jako vstupy do modelu ze stejných důvodů jako v případě modelu BLDC motoru. V Tab. 8 jsou popsány vstupní parametry modelu asynchronního motoru, včetně datových typů ve formátu s pevnou desetinnou čárkou.

**Tab. 8: Vstupní parametry modelu asynchronního motoru**

Číslo parametru	Označení	Datový typ
1	$u_\alpha$	s22f14
2	$u_\beta$	s22f14
3	$N_{K_l}/K_l/N_u$	s15f14
4	$L_m R_r N_L^2 / (L_r^2 K_l N_R)$	s25f24
5	$L_m N_L / (L_r K_l)$	s25f24
6	$K_r N_{K_l} / (K_l N_{K_r})$	s25f24
7	$T_s / N_{T_s}$	s25f24
8	$L_m N_{T_r} / (T_r N_L)$	s25f24
9	$N_{T_r} / T_r$	s25f24
10	$3PL_m / (2L_r)$	s25f20
11	$M_z$	s18f16
12	$1/N_M$	s22f16
13	$T_s N_J / (J N_{T_s})$	s25f24

Tab. 9 popisuje výstupní datový typ jednotlivých matematických operací uvnitř modelovacího schématu BLDC motoru. Jednotlivé operace jsou popsány indexy  $n$  v případě násobení a indexem  $s$  v případě sčítání.



Obr. 25: Upravené modelovací schéma modelu asynchronního motoru

Tab. 10 popisuje výstupní parametry, podobně jako Tab. 8. Kvůli velikosti tabulky a opakujícím se informacím, je obsah tabulky zapsán zkráceně

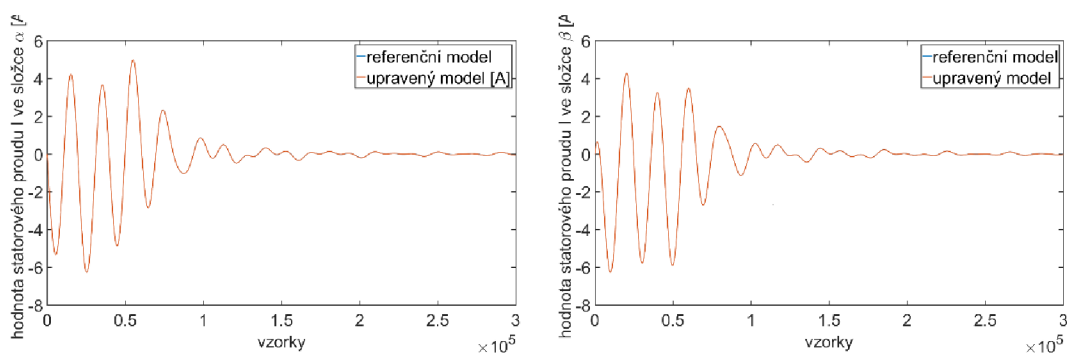
Tab. 9: Datové typy výstupů matematických operací uvnitř modelu

Číselné označení	Násobení/sčítání	Datový typ
n1 až n23	násobení	s25f24
s1 až s17	sčítání	s25f24

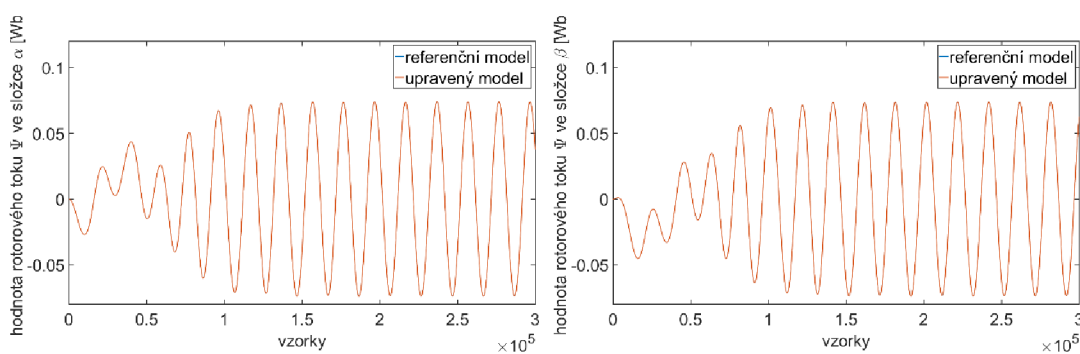
Na Obr. 26, Obr. 27 a Obr. 28 jsou zobrazeny průběhy původního referenčního modelu a upraveného modelu asynchronního motoru.

Tab. 10: Popis výstupních parametrů modelu asynchronního motoru

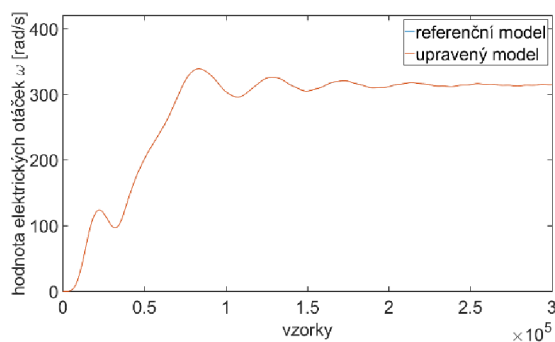
Číslo parametru	označení	Datový typ
1	$\omega_{el}$	s25f24
2	$i_{\alpha}$	s25f24
3	$i_{\beta}$	s25f24
4	$\Psi_{\alpha}$	s25f24
5	$\Psi_{\beta}$	s25f24



**Obr. 26: Srovnání průběhů statorových proudů referenčního modelu a upraveného modelu asynchronního motoru**



**Obr. 27: Srovnání průběhů rotorových toků referenčního modelu a upraveného modelu asynchronního motoru**



**Obr. 28: Srovnání průběhů otáček referenčního modelu a upraveného modelu asynchronního motoru**

Na průbězích lze vidět lepší překrytí křivek, než v případě úpravy modelu pro BLDC motor. Toto je dáno nepoužitím datových typů s18f17.

## 5.4 Implementace modelu v ZYNQ-7000

Podobně jako v případě implementace modelu BLDC motoru je nutné v případě implementace asynchronního motoru přetransformovat model v prostředí Matlab/Simulink do podoby kódu v jazyce C/C++.

Kód obsahuje výpočet 5 stavových rovnic podle rovnic (41), (42), (43), (44) a (45). Dále se počítá s implementací kódu do podoby *IP core* periferie připojitelné k jádru ARM Cortex-A9 pomocí sběrnice rozhraní AXI4-lite. Funkce také zahrnuje podobný přepínací mechanismus jako v případě modelu BLDC motoru, který je ovládán skrze registr přístupný přes sběrnice rozhraní jádra procesoru. Tento přepínací mechanismus přepíná to, zda mají být vstupní data (vstupní napětí  $u$  a zátěžný moment  $M_z$ ) načítána pro výpočet ze sběrnice rozhraní, a tedy poskytnuta procesorem, a nebo skrze fyzické porty periferie, a tedy poskytnuta z jiné periferie, která je přímo připojena k periférii reprezentující model asynchronního motoru.

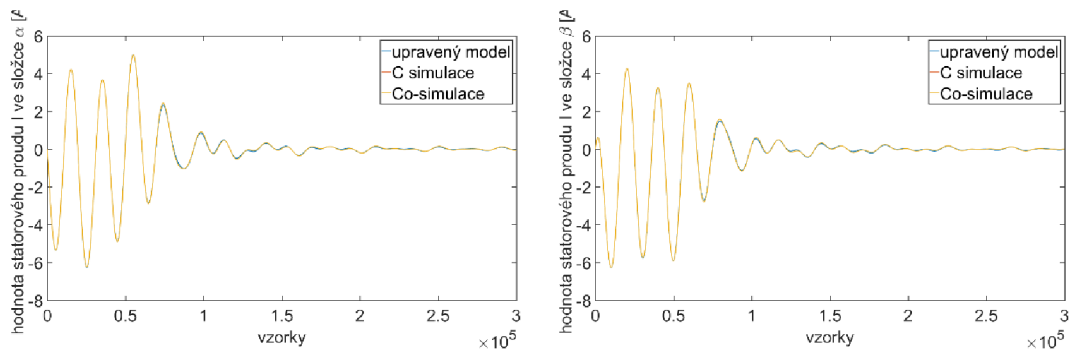
Následující Tab. 11 udává procentuální vytíženost prostředků uvnitř FPGA části obvodu *xc7z020c1g484-1* a časovou náročnost na jednu iteraci výpočtu periferie zobrazenou jako počet period hodinového signálu (latence). Frekvence hodinového signálu je  $100\text{MHz}$ . Tyto údaje byly vyčteny přímo z prostředí Vivado HLS.

**Tab. 11: Hodnocení implementace modelu asynchronního motoru v prostředí Vivado HLS**

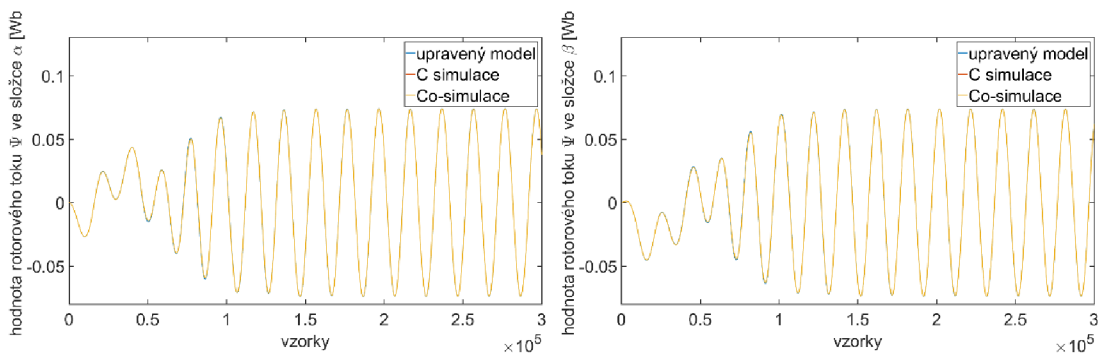
Údaj	hodnocení
Vytíženost LUT	9%
Vytíženost DFF	3%
Vytíženost DSP	19%
Latence	20

Podobně jako v případě popisu implementace BLDC motoru, je na následujících obrázcích (Obr. 29, Obr. 30 a Obr. 31) srovnání průběhů upraveného modelu a výstupů z C simulace a Co-simulace z prostředí Vivado HLS.

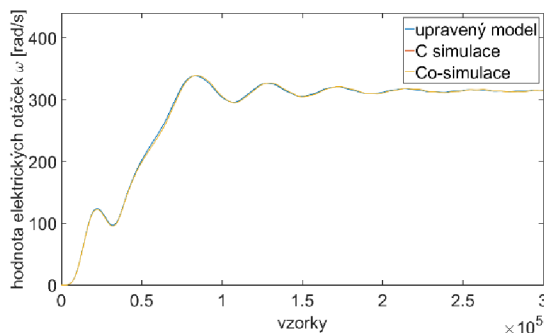
Při pohledu na srovnání průběhů je vidět, že situace je podobná jako u modelu BLDC motoru. Rozdíly jsou opět dány rozdílnou aritmetikou operací proměnných s pevnou desetinnou čárkou.



**Obr. 29: Srovnání proudů mezi upraveným modelem a simulacemi v prostředí Vivado HLS**



**Obr. 30: Srovnání toků mezi upraveným modelem a simulacemi v prostředí Vivado HLS**

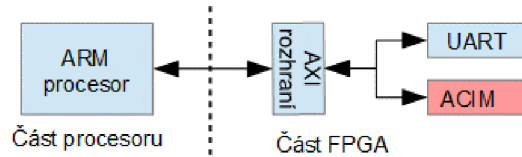


**Obr. 31: Srovnání otáček mezi upraveným modelem a simulacemi v prostředí Vivado HLS**

Následující postup overení funkčnosti implementace byl shodný s postupem v případě modelu BLDC motoru. Opět byla vytvořena *IP core* periferie a v prostředí Vivado byl vytvořen procesorový systém s procesorem ARM Cortex-A9. Na Obr. 32 je znázorněno jeho blokové schéma, kde blok *ACIM* reprezentuje *IP core* periferii modelu asynchronního motoru.

Sériové rozhraní UART (*Universal Asynchronous Receiver Transmitter*) zde opět slouží pro záznam dat ze simulace. Při implementaci tohoto procesorového systému v prostředí Vivado, byly aplikovány další optimalizace pro spotřebu prostředků FPGA části. Tyto optimalizace byly v režii prostředí Vivado. V Tab. 12 je

znázorněna procentuální vytiženosť FPGA obvodu při implementaci celkového procesorového systému (včetně sběrnice rozhraní AXI a periferie UART).

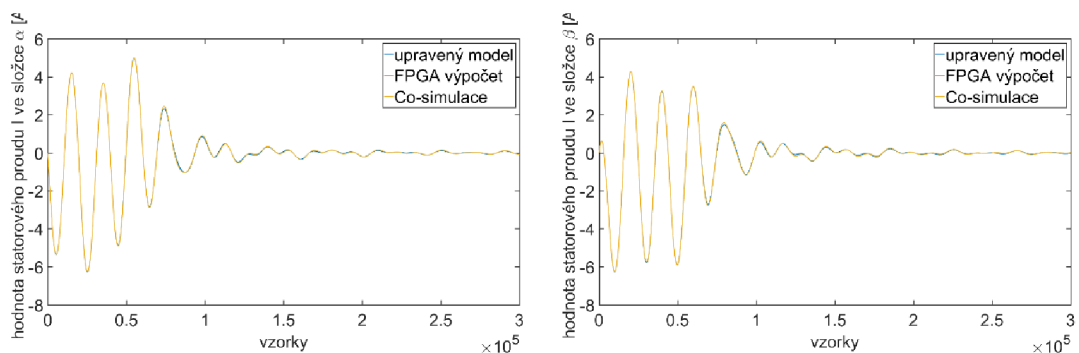


**Obr. 32: Diagram procesorového systému pro testování modelu asynchronního motoru**

**Tab. 12: Hodnocení implementace procesorového systému v prostředí Vivado**

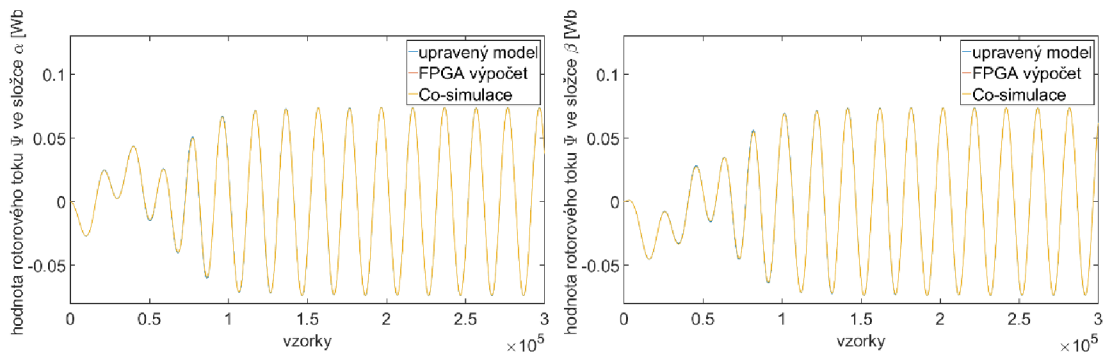
Údaj	Hodnocení
Vytiženosť LUT	4%
Vytiženosť DFF	4%
Vytiženosť DSP	20%

V procesoru ARM Cortex-A9 byla implementována podobná aplikace pro spouštění jednotlivých kroků výpočtu modelu a to pro stejný počet vzorků jako modely v prostředí Matlab/Simulink, podobně jako v případě modelu BLDC motoru. Při tomto nebylo použito žádné časování (například časování čítačem). Po skončení simulace byla data porovnána s výstupem upraveného modelu a výstupem Co-simulace. Výsledky jsou zobrazeny na Obr. 33, Obr. 34 a Obr. 35.

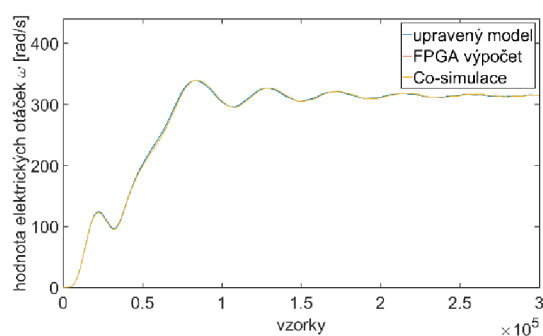


**Obr. 33 : Srovnání proudů mezi upraveným modelem, Co-simulací v prostředí Vivado HLS a výpočtem na FPGA části obvodu ZYNQ-7000**





**Obr. 34: Srovnání toků mezi upraveným modelem, Co-simulací v prostředí Vivado HLS a výpočtem na FPGA části obvodu ZYNQ-7000**



**Obr. 35: Srovnání otáček mezi upraveným modelem, Co-simulací v prostředí Vivado HLS a výpočtem na FPGA části obvodu ZYNQ-7000**

## 5.5 Závěr kapitoly Implementace asynchronního motoru

V této kapitole byla popsána implementace modelu asynchronního motoru v  $\alpha\beta$  rovině. Bylo zde ukázáno, jak byl model upraven pro potřeby implementace. Dále zde bylo popsáno, jak byla testována výsledná implementace modelu asynchronního motoru. V závěrečné části této kapitoly byly porovnány výsledky chování modelu v prostředí Matlab/Simulink a implementace v FPGA části obvodu ZYNQ-7000.

## 6 EXPERIMENTÁLNÍ MIL SIMULACE

V této kapitole jsou popsány tři experimentální MIL simulace BLDC motoru a asynchronního motoru. Ke každému motoru je připojen model mechanické pružné spojky a jsou zkoumány výstupní otáčky mechanické pružné spojky. Třetí simulací je pak kooperace modelu BLDC motoru a proudového PI regulátoru. Modely motoru a spojky běží na různých periodách vzorkování. Proudový PI regulátor pak běží na stejné periodě vzorkování jako model BLDC motoru. Jak model BLDC motoru tak i model asynchronního motoru jsou implementovány v FPGA části obvodu ZYNQ-7000. Model mechanické pružné spojky a proudového PI regulátoru jsou implementovány v Procesoru ARM Cortex-A9. Pro potřeby simulace jsou použity *IP core* periferie popsané v kapitole 4 a v kapitole 5. V této kapitole je popsán matematický model mechanické pružné spojky a přenos PI regulátoru. Dále je zde popsána jejich úprava a implementace a na závěr jsou zde porovnány výstupy z referenčních modelů v prostředí Matlab/Simulink a implementovaných modelů v obvodu ZYNQ-7000.

### 6.1 Základní popis experimentu

Experiment MIL simulace se skládá z modelu BLDC motoru, implementovaného v FPGA části obvodu ZYNQ-7000. K tomuto modelu je připojen model pružné mechanické spojky. Spojka spojuje dvě rotační hmotnosti. První je reprezentována mechanickou částí motoru a druhá konstantním momentem setrvačnosti  $J_z$ .

Algoritmus modelu je realizován jako aplikace na procesoru ARM Cortex-A9 a je tedy realizován jako samostatný simulační blok, oddělený od modelu implementovaného v FPGA části. Toto odpovídá definici MIL simulace v Kapitole 2.1.

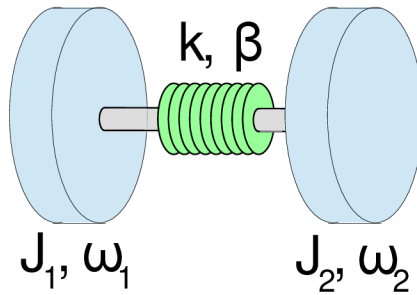
Model BLDC motoru je počítán s periodou vzorkování  $T_s = 10 \mu s$  a model mechanické části je počítán s periodou vzorkování  $T_{sm} = 50 \mu s$ .

### 6.2 Matematický model experimentální simulace

Obr. 36 znázorňuje uspořádání mechanické pružné spojky spojující dvě tuhosti. Její matematický model je definován dvěma následujícími rovnicemi (53) a (54).

$$M_0(t) = J_1 \cdot \frac{d\omega_1(t)}{dt} + k \int (\omega_1(t) - \omega_2(t)) dt + \beta(\omega_1(t) - \omega_2(t)) \quad 53$$

$$0 = J_2 \cdot \frac{d\omega_2}{dt} - k \int (\omega_1(t) - \omega_2(t)) dt - \beta(\omega_1 - \omega_2) \quad 54$$

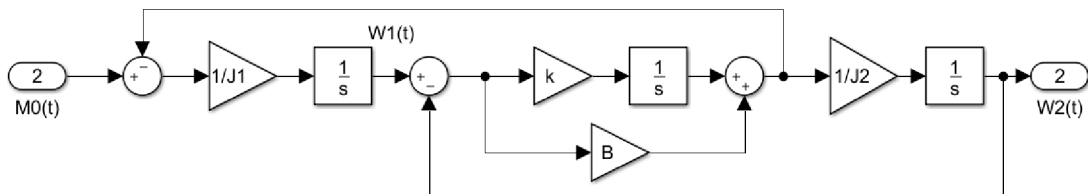


Obr. 36: Uspořádání mechanické pružné spojky s dvěma tuhostmi

Kde  $M_0$  je vstupní moment, jenž je přenášen přes mechanickou spojku. Parametr  $J_1$  respektive  $J_2$  je moment setrvačnosti hmotnosti na vstupní straně spojky respektive na výstupní. Proměnná  $\omega_1$  respektive  $\omega_2$  představuje otáčky na vstupní straně spojky, vyvolané vstupním momentem  $M_0$ , respektive na výstupní straně. Parametr  $k$  představuje pružnost spojky a parametr  $\beta$  představuje tlumení spojky. Detailní popis modelu lze dohledat [24].

Na Obr. 37 je modelovací schéma modelu pružné mechanické spojky v prostředí Matlab/Simulink.

Jak už bylo zmíněno dříve, vstupní rotační hmotnost je realizována mechanickou částí BLDC motoru a proto se model mechanické spojky zjednoduší. Vazba  $k \int (\omega_1 - \omega_2) dt + \beta (\omega_1 - \omega_2)$  má význam zátěžného momentu  $M_z$ , vstupující do modelu motoru. Rovnice (53) a (54), se pak upraví na rovnice (55) a (56).

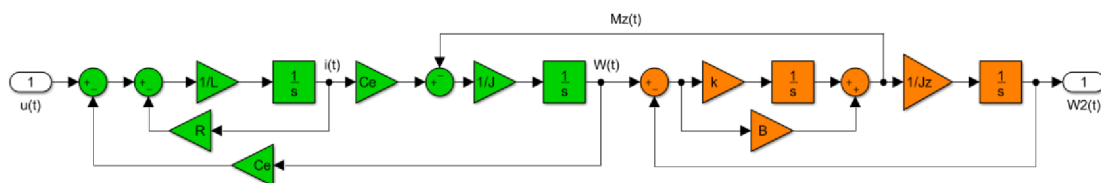


Obr. 37: Mode mechanické pružné spojky

Na Obr. 38 je modelovací schéma modelu BLDC motoru a mechanické části, s uvažovaným momentem setrvačnosti  $J$  motoru a momentem setrvačnosti na výstupu spojky  $J_z$ . V modelu bylo použito barevného označení bloků pro lepší orientaci. Zelenou barvou jsou označeny části modelu motoru a oranžovou barvou jsou vyznačeny bloky modelu mechanické pružné spojky.

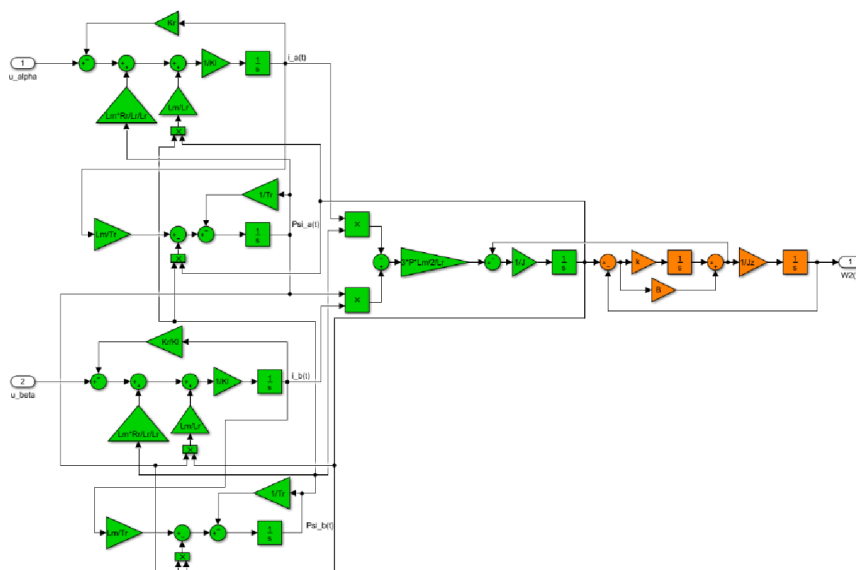
$$M_z(t) = \beta(\omega_1(t) - \omega_2(t)) + k \int (\omega_1(t) - \omega_2(t)) dt \quad 55$$

$$\omega_2(t) = \frac{1}{J} \int M_z(t) dt \quad 56$$



Obr. 38: Spojení pružné spojky a BLDC motoru

Stejné uspořádání pak platí i pro spojení asynchronního motoru s mechanickou spojkou. Na Obr. 39 je pak zobrazeno modelovací schéma asynchronního motoru s mechanickou spojkou. Podobně jako v případě modelovacího schématu s BLDC motorem, jsou bloky v tomto simulačním schématu barevně zvýrazněny. Kde bloky označené zeleně náleží modelu asynchronního motoru a bloky zvýrazněny oranžově náleží modelu mechanické pružné spojky.



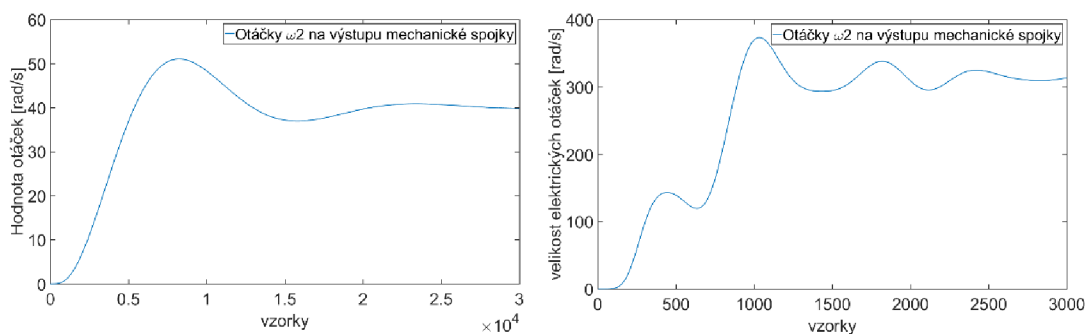
Obr. 39: Spojení pružné spojky a asynchronního motoru

V Tab. 13 jsou uvedeny parametry mechanické pružné spojky. Koeficient pružnosti byl různý pro případ BLDC motoru a pro případ asynchronního motoru (položka ACIM). Ostatní parametry mechanické pružné spojky byly shodné pro oba typy motorů. Parametry pro části motorů jsou shodné jako v kapitole 4.2 a v kapitole 5.2. Na Obr. 40 je zobrazen výstup otáček  $\omega_2(t)$  na výstupu mechanické spojky spojené s BLDC motorem a výstup otáček  $\omega_2(t)$  na výstupu mechanické spojky

spojené s asynchronním motorem. Simulace obou modelů byla spuštěna podobně jako referenční modely pro BLDC motor a asynchronní motor v prostředí Matlab/Simulink. Numerická metoda pro výpočet integrací byla zvolena opět *ode3 Bogacki-Shampine*. Délka trvání simulace pro BLDC motor byla 0,3 s a pro asynchronní motor pak 0,03 s. Rozdíl v délce trvání jednotlivých simulací je dán různou délkou přechodných dějů jednotlivých modelů.

**Tab. 13: Parametry mechanické pružné spojky**

Název parametru	hodnota
Koeficient pružnosti $k$ (BLDC)	$0,05 \text{ kg} \cdot \text{m}^2/\text{s}^2$
Koeficient pružnosti $k$ pro (ACIM)	$0,3 \text{ kg} \cdot \text{m}^2/\text{s}^2$
Koeficient tlumení $\beta$	$0 \text{ kg} \cdot \text{m}^2/\text{s}$
Moment setrvačnosti na výstupu $J_z$	$2,88 \cdot 10^{-5} \text{ kg} \cdot \text{m}^2$
Vstupní napětí BLDC motoru $u(t)$	$2 \text{ V}$
Amplituda vstupního napětí as. motoru	$22 \text{ V}$
Frekvence vstupního napětí as. motoru	$50 \text{ Hz}$

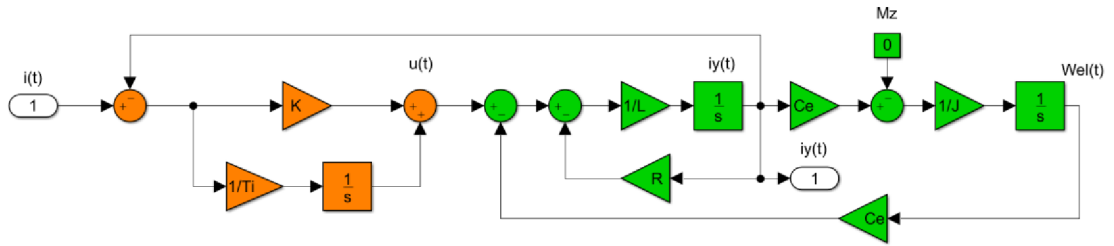


**Obr. 40: Průběh otáček  $\omega$  na výstupu mechanické spojky v případě BLDC (vlevo) motoru a asynchronního motoru (vpravo)**

Následující rovnice popisuje výstup PI regulátoru [25]:

$$y_{PI}(t) = K \left( e(t) + \frac{1}{T_i} \int e(t) dt \right) \quad 57$$

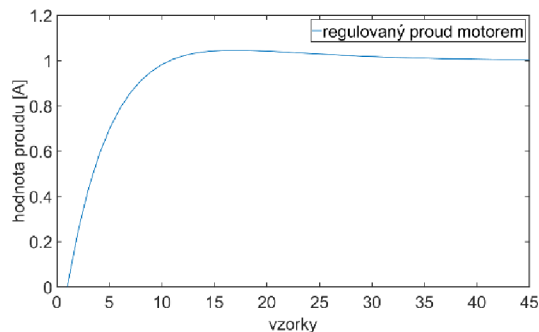
Kde  $K$  je proporcionální zesílení a  $T_i$  je integrační časová konstanta. Na Obr. 41 je zobrazeno modelovací schéma BLDC motoru s připojeným PI regulátorem, který slouží jako proudová regulace. Jednotlivé bloky jsou ve schématu opět barevně odlišený. Zeleně zvýrazněné bloky náleží modelu BLDC motoru a oranžově zvýrazněné bloky pak náleží PI regulátoru.



Obr. 41: Spojení PI regulátoru proudu a BLDC motoru

Regulátor proudu byl navrhnut s následujícími parametry: proporcionální zesílení  $K = 7$  a integrační časová konstanta  $T_i = 0,0173$ . V návrhu nebylo zahrnuto dopravní zpoždění způsobené diskretizací ani dobou výpočtu, protože se jednalo o nízké hodnoty.

Vnitřní zpětná vazba od otáček motoru do elektrické části modelu není nijak touto regulací kompenzována, ale je chápána jako porucha. Díky přítomnosti integrační složky v regulátoru je však působení této poruchy potlačeno. Na Obr. 42 je vyobrazen průběh výstupu proudu z modelu BLDC motoru. Žádaná hodnota proudu byla 1 A. Simulace byla spuštěna se stejnými parametry jako simulace s mechanickou spojkou. Akorát PI regulátor proudu pracoval se stejnou periodou vzorkování jako model BLDC motoru, tedy  $T_s = 10 \mu s$ .



Obr. 42: Průběh regulovaného proudu BLDC motoru

### 6.3 Úprava modelu pro implementaci

Stejně jako v případech úpravy modelu BLDC motoru a asynchronního motoru je nutné diskretizovat modely pomocí Eulerovy aproximace, doplnit modely o měřítko tak, aby číselný rozsah všech matematických operací uvnitř modelu byl v rozsahu  $\pm 1$ . Následující vztahy popisují diskretizovanou formu modelu mechanické pružné spojky bez připojeného BLDC motoru:

$$M_z(k) = \beta(\omega_1(k) - \omega_2(k)) + k \left( (T_{sm}\omega_1(k) + \omega_1(k-1)) - (T_{sm}\omega_2(k) + \omega_2(k-1)) \right) \quad 58$$

$$\omega_2 = \frac{1}{J} (T_{sm}M_z(k) + M_z(k-1)) \quad 59$$

Nyní následují tyto rovnice doplněné o měřítka:

$$\begin{aligned} & \frac{M_z(k)}{N_{M_z}} \\ &= \frac{\beta(\omega_1(k) - \omega_2(k))}{N_\beta N_\omega} \cdot \frac{N_\beta N_\omega}{N_{M_z}} \\ &+ \frac{k \left( \left( \frac{T_{sm}}{N_{T_s}} N_{T_s} \omega_1(k) + \omega_1(k-1) \right) - \left( \frac{T_{sm}}{N_{T_s}} N_{T_s} \omega_2(k) + \omega_2(k-1) \right) \right)}{N_k N_\omega} \\ & \cdot \frac{N_k N_\omega N_{T_s}}{N_{M_z}} \end{aligned} \quad 60$$

$$\frac{\omega_2(k)}{N_\omega} = \frac{N_J}{J N_{M_z}} \left( \frac{T_{sm}}{N_{T_s}} N_{T_s} M_z(k) + M_z(k-1) \right) \cdot \frac{N_{M_z} N_{T_s}}{N_J N_\omega} \quad 61$$

Následující členy jsou bezrozměrné a rovny hodnotě 1.

$$\frac{N_\beta N_\omega}{N_{M_z}} = 1, \frac{N_k N_\omega N_{T_s}}{N_{M_z}} = 1, \frac{N_{M_z} N_{T_s}}{N_J N_\omega} = 1 \quad 62$$

Měřítka  $N_J$ ,  $N_{M_z}$ ,  $N_\omega$  a  $N_{T_s}$  jsou shodná s měřítky použitými v části motoru za předpokladu, že číselné hodnoty parametrů nepřesáhnou maximální hodnoty měřítek. Pomocí těchto známých měřítek lze pak odvodit měřítka pro koeficient pružnosti  $k$  a koeficient tlumení  $\beta$ .

$$N_\beta = \frac{N_{M_z}}{N_\omega}, N_k = \frac{N_{M_z}}{N_\omega N_{T_s}} \quad 63$$

Nyní následuje stejný postup pro PI regulátor. Nejprve diskretizovaná rovnice popisující PI regulátor

$$y_{PI}(k) = K \left( e(k) + \frac{1}{T_i} (T_s e(k) + e(k-1)) \right) \quad 64$$

Nyní následuje tato rovnice doplněna o měřítka, kde vstup  $e$  je upraven měřítkem proudu, protože se jedná o vstup žádané hodnoty proudu a výstup  $y_{PI}$  je měřítkován měřítkem napětí, protože na výstupu je akční zásah napětí.

$$\frac{y_{PI}(k)}{N_u} = \frac{K}{N_K} \frac{e(k)}{N_i} \cdot \frac{N_K N_i}{N_u} + \frac{K}{N_K} \frac{N_{T_i}}{T_i} \frac{\left( \frac{T_s}{N_{T_s}} e(k) + e(k-1) \right)}{N_i} \cdot \frac{N_K N_i N_{T_s}}{N_{T_i} N_u} \quad 65$$

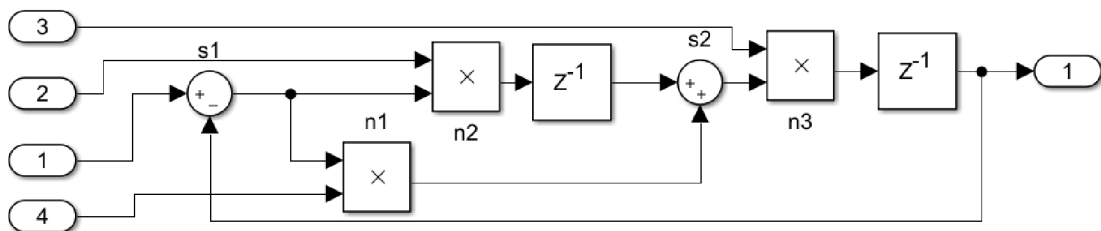
Bezrozměrné členy pro určení měřítek PI regulátoru jsou pak následující:

$$\frac{N_K N_i}{N_u} = 1, \frac{N_K N_i N_{T_s}}{N_{T_i} N_u} = 1 \quad 66$$

Z nich pak jasně vyplývá:

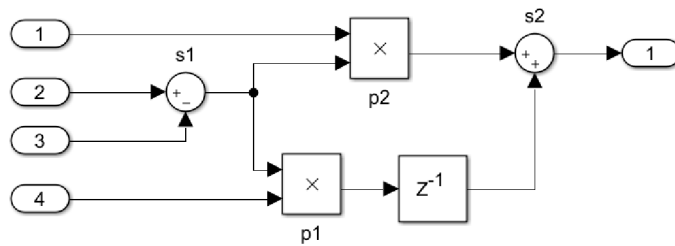
$$N_K = \frac{N_u}{N_i} = N_R, N_{T_i} = \frac{N_K N_i N_{T_s}}{N_u} = \frac{N_R N_{T_s}}{N_R} = N_{T_s} \quad 67$$

Na Obr. 43 je přepracované modelovací schéma části mechanické spojky z Obr. 38. Na Obr. 44 je přepracované schéma PI regulátoru z Obr. 40. Všechny parametry modelů jsou realizované jako vstupy do modelů. Toto je kvůli lepšímu srovnání s výslednou implementací. Kde modely budou implementovány jako funkce a všechny parametry modelů budou tvořit parametry funkcí. To platí i pro výstupy modelů. Všechny parametry funkcí jsou realizovány jako ukazatele. V tomto případě však funkce nebudou realizovány jako *IP Core* periferie, ale jen jako C funkce implementované v procesoru ARM Cortex-A9.



Obr. 43: Upravené modelovací schéma mechanické pružné spojky



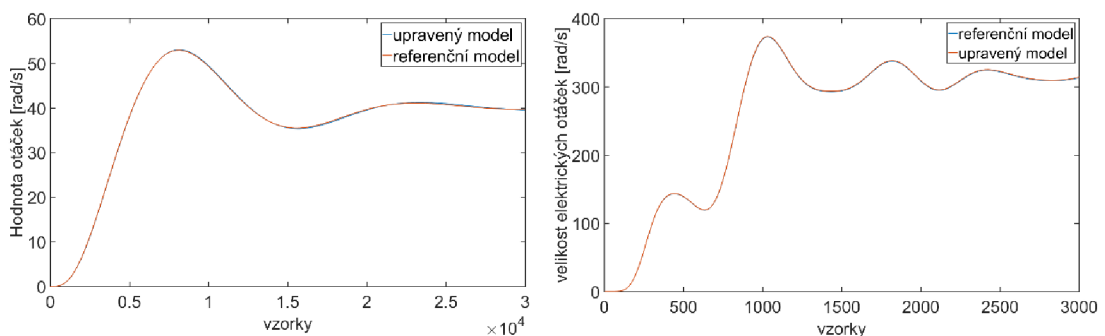


**Obr. 44: Upravené modelovací schéma PI regulátoru**

Protože je funkce realizována v jazyce C, nejsou k dispozici datové typy s proměnnou velikostí jako v případě použití knihoven v prostředí Vivado HLS, ale pouze standardní 8, 16, 32 a 64 bitové proměnné. Proto jsou všechny vstupní parametry, výstupy matematických operací a výstupní parametry mechanické pružné spojky a PI regulátoru realizované jako 32 bitové proměnné s pevnou desetinnou čárkou  $s32f31$ . V následující tabulce (Tab. 14) je popis vstupních a výstupních parametrů upraveného modelu mechanické pružné spojky. Na Obr. 45 je vyobrazeno porovnání výstupu původního modelu a upraveného modelu pro oba modely motorů.

**Tab. 14: Popis vstupů a výstupů upraveného modelu mechanické spojky**

Číslo parametru	Označení	Datový typ
Vstup 1	$\omega(k)$	$s32f31$
Vstup 2	$kT_{sm}/N_k N_{T_s}$	$s32f31$
Vstup 3	$T_{sm} N_j / J_z N_{T_s}$	$s32f31$
Vstup 4	$\beta / N_\beta$	$s32f31$
Výstup 1	$\omega_2(k)$	$s32f31$

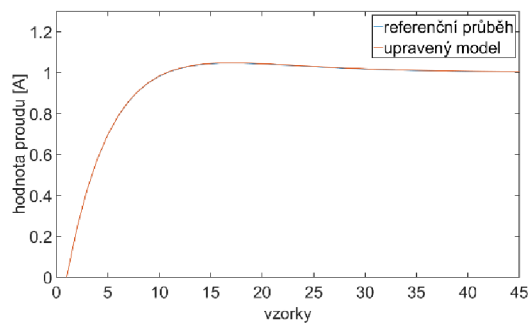


**Obr. 45: Srovnání otáček referenčním a upraveným modelem BLDC motoru (vlevo) a asynchronního motoru (vpravo)**

V následující tabulce (Tab. 15) je popis vstupních a výstupních parametrů upraveného schématu PI regulátoru. Na Obr. 46 je vyobrazeno porovnání výstupu původního modelu a upraveného modelu s PI regulátorem.

**Tab. 15: Popis vstupů a výstupů upraveného schématu PI regulátoru**

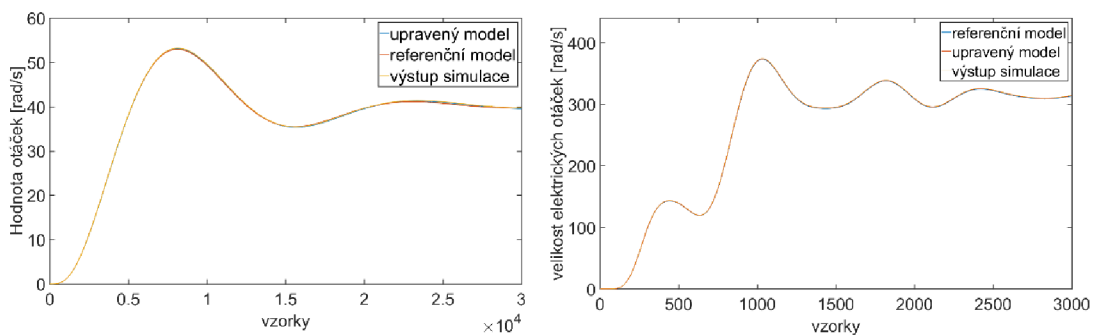
Číslo parametru	Označení	Datový typ
Vstup 1	$K/N_R$	$s32f31$
Vstup 2	$i(k)$	$s32f31$
Vstup 3	$i_y(k - 1)$	$s32f31$
Vstup 4	$T_s/T_i$	$s32f31$
Výstup 5	$u(k)$	$s32f31$



**Obr. 46: Srovnání výstupu proudu referenčního modelu a upraveného modelu BLDC motoru s připojeným PI regulátorem**

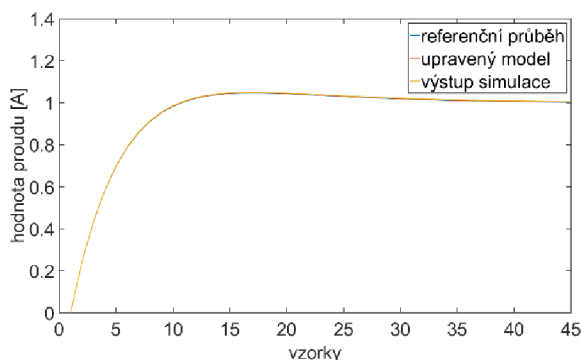
## 6.4 Implementace experimentu v ZYNQ-7000

Pro implementaci experimentální simulace s mechanickou pružnou spojkou byla použita již hotová  $I_p$  core periferie modelu BLDC motoru a asynchronního motoru. Jak již bylo zmíněno v textu výše, část mechanické pružné spojky, popsané rovnicemi (57) a (58), je realizována jako aplikace běžící na procesoru ARM Cortex-A9. Pro přesné časování spuštění simulace (perioda vzorkování  $T_s$   $10 \mu s$ ) byl použit čítač procesoru ARM. Po každém pátém spuštění výpočtu kroku simulace BLDC motoru (nebo asynchronního motoru) je spuštěn krok výpočtu modelu mechanické pružné spojky. Tímto je docíleno to, že výpočet tohoto modelu je počítán s periodou vzorkování  $T_{sm} = 50 \mu s$ . Procesorový systém pro testování modelů je pak shodný s tím, který je použit v kapitole 4.4 respektive v kapitole 5.4. Na Obr. 47 je znázorněno porovnání průběhu otáček z výstupu simulace spolu s průběhem výstupu upraveného modelu a referenčního modelu pro oba modely motorů.



**Obr. 47: Srovnání průběhů otáček mechanické spojky mezi referenčním modelem, upraveným modelem pro implementaci a reálnou simulací BLDC motoru (vlevo) a asynchronního motoru (vpravo)**

Pro testování modelu BLDC motoru s proudovým PI regulátorem bylo také použito čítače pro přesné časování spouštění simulace. Model BLDC motoru a proudového regulátoru však běžely na stejné periodě vzorkování  $T_s = 10 \mu s$ . Na Obr. 48 je pak porovnání průběhu proudu z výstupu simulace spolu s průběhem výstupu upraveného modelu a referenčního modelu BLDC motoru s PI regulátorem.



**Obr. 48: Srovnání průběhů proudů mezi referenčním modelem, upraveným modelem pro implementaci a reálnou simulací BLDC motoru s připojeným PI regulátorem**

## 6.5 Závěr kapitoly Experimentální MIL simulace

V této kapitole byly popsány tři experimentální MIL simulace využívající *IP core* periférii BLDC motoru, asynchronního motoru, model mechanické pružné spojky a PI regulátor implementovaný v procesoru ARM Cortex-A9. Každá simulace byla sestavena ze dvou bloků (BLDC motor a spojka, asynchronní motor a spojka a BLDC motor a PI regulátor), kde obě části modelu běží samostatně jako oddělené bloky, což splňuje definici MIL simulace. V této kapitole byl také popsán matematický model mechanické pružné spojky a struktury PI regulátoru. V závěru kapitoly bylo uvedeno srovnání výsledků mezi simulací v prostředí Matlab/Simulink a simulací na platformě ZYNQ-7000.

## 7 ZÁVĚR

Zadání diplomové práce MIL simulace elektrických motorů v reálném čase se skládalo ze čtyř bodů. První bod byl o prozkoumání možností obvodu ZYNQ-7000 pro implementaci modelů a řízení střídavých, elektrických pohonů. Vlastnosti a možnosti využití tohoto obvodu jsou popsány v kapitole 2. Druhým bodem zadání bylo vytvoření modelů BLDC motoru a asynchronního motoru. Tato problematika je popsána v kapitole 4 a v kapitole 5. Třetí bod zadání se zabýval přidáním mechanické části k obou modelům motorů. Postup a výsledky tohoto bodu zadání jsou popsány v kapitole 6. Poslední čtvrtý bod zadání se zabýval připojením proudového PI regulátoru k jednomu motoru. Tento bod je popsán taktéž v kapitole 6.

Diplomová práce se zabývala úpravou matematických modelů dvou vybraných typů motorů a jejich implementací v obvodu ZYNQ-7000 společnosti Xilinx. Dva vybrané typy motorů byly bezkartáčový BLDC motor a asynchronní motor. Úprava modelů spočívala v jejich diskretizaci a měřítkování pro snadné použití datových typů s pevnou desetinnou čárkou. Práce poukázala v kapitole 3.1 na vliv Eulerovy metody, která byla v této práci použita. V kapitole 3.2 byly diskutovány vlastnosti proměnných s pevnou desetinnou čárkou a byly srovnány s vlastnostmi datových typů s plovoucí desetinou čárkou. Práce také popisuje způsob, jakým byly jednotlivé modely implementovány. K tomuto bylo použito vývojových prostředí Vivado, Vivado HLS a Xilinx SDK pro tvorbu *IP core* periférií, testovacích procesorových systémů a testovacích aplikací pro otestování vlastností jednotlivých modelů. V poslední kapitole této práce jsou popsány tři experimentální MIL simulace. První experiment byl simulací propojení BLDC motoru a mechanické pružné spojky. Druhý experiment byl simulací propojení asynchronního motoru a mechanické pružné spojky s jinými parametry než v případě BLDC motoru. Třetí a poslední experiment byl simulací BLDC motoru s proudovým PI regulátorem.

Práce poukazuje na fakt, že platforma ZYNQ-7000 je použitelný nástroj pro implementaci modelů střídavých elektrických motorů pro účely real-time simulace a MIL simulace. Díky možnosti provádět simulace v reálném čase je tato platforma vhodná i pro jiné typy simulací například: PIL, SIL, HIL (*Processor In the Loop, System In the Loop, Hardware In the Loop*).

Hlavním přínosem této práce je přístup, s jakým byly modely implementovány. A to jako *IP core* periférie, které jsou vhodné pro tvorbu procesorových systémů ať už v obvodech ZYNQ-7000 spolu s procesorovými jádry ARM Cortex-A9, a nebo na samostatných FPGA obvodech, kde periférie mohou tvořit procesorový systém spolu se *softcore* procesory (Xilinx *MicorBlaze*).

# Literatura

- [1] From AUTOSAR Models to Co-Simulation for MiL-Testing in the Automotive Domain [online]. †Fraunhofer-Institute for Computer Architecture and Software Technology FIRST, Berlin, Germany, 2012 [cit. 2017-04-04].  
Dostupné z:  
<http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6200149>.  
Fraunhofer-Institute for Computer Architecture and Software Technology FIRST.
- [2] FPGAs & 3D ICs [online]. Xilinx Inc.: Xilinx, 2017 [cit. 2017-04-04]. Dostupné z: <http://www.xilinx.com/products/silicon-devices/fpga.html>
- [3] MATLAB. Mathworks.com [online]. Mathworks: Mathworks, 2017 [cit. 2017-04-04]. Dostupné z: <https://www.mathworks.com/products/matlab.html>
- [4] XILINX ARTIX-7 FPGAS: New Performance and Bandwidth Standards for Power-Limited, Cost-Sensitive Markets. Xilinx.com [online]. Xilinx Inc.: Xilinx, 2016 [cit. 2017-04-04]. Dostupné z:  
<https://www.xilinx.com/support/documentation/product-briefs/artix7-product-brief.pdf>
- [5] 7 Series FPGAs Configurable Logic Block. In: Xilinx [online]. 2016 [cit. 2017-01-05]. Dostupné z:  
[https://www.xilinx.com/support/documentation/user\\_guides/ug474\\_7Series\\_CLB.pdf](https://www.xilinx.com/support/documentation/user_guides/ug474_7Series_CLB.pdf)
- [6] ARM Cortex-A9 Technical Reference Manual. 2008. Dostupné také z:  
[http://infocenter.arm.com/help/topic/com.arm.doc.ddi0388g/DDI0388G\\_cortex\\_a9\\_r3p0\\_trm.pdf](http://infocenter.arm.com/help/topic/com.arm.doc.ddi0388g/DDI0388G_cortex_a9_r3p0_trm.pdf)
- [7] Xilinx [online]. 2017 [cit. 2017-01-05]. Dostupné z:  
<https://www.xilinx.com/products/silicon-devices/soc/zynq-7000.html>
- [8] ZYNQ-7000 architecture. In: Xilinx [online]. [cit. 2017-01-05]. Dostupné z:  
<http://www.xilinx.com/content/dam/xilinx/imgs/block-diagrams/zynq-mp-core-dual.png>
- [9] Vivado High-Level Synthesis. Xilinx.com [online]. Xilinx Inc.: Xilinx, 2017 [cit. 2017-04-04]. Dostupné z: <http://www.xilinx.com/products/design-tools/vivado/integration/esl-design.html>
- [10] Vivado user guide. In: Xilinx [online]. [cit. 2017-01-05]. Dostupné z:  
[http://www.xilinx.com/support/documentation/sw\\_manuals/xilinx2014\\_1/ug902-vivado-high-level-synthesis.pdf](http://www.xilinx.com/support/documentation/sw_manuals/xilinx2014_1/ug902-vivado-high-level-synthesis.pdf)
- [11] Vivado user Guide: Synthesis. In: Xilinx [online]. 2013 [cit. 2017-01-05].  
Dostupné z:  
[http://www.xilinx.com/support/documentation/sw\\_manuals/xilinx2012\\_2/ug901-vivado-synthesis.pdf](http://www.xilinx.com/support/documentation/sw_manuals/xilinx2012_2/ug901-vivado-synthesis.pdf)

- [12] Vivado user guide: implementation. In: Xilinx [online]. 2012 [cit. 2017-01-05]. Dostupné z:  
[http://www.xilinx.com/support/documentation/sw\\_manuals/xilinx2012\\_2/ug904-vivado-implementation.pdf](http://www.xilinx.com/support/documentation/sw_manuals/xilinx2012_2/ug904-vivado-implementation.pdf)
- [13] Xilinx Software Development Kit (XSDK). Xilinx.com [online]. Xilinx Inc.: Xilinx, 2017 [cit. 2017-04-04]. Dostupné z:  
<https://www.xilinx.com/products/design-tools/embedded-software/sdk.html>
- [14] PicoBlaze 8-bit Microcontroller. Xilinx.com [online]. Xilinx Inc.: Xilinx, 2017 [cit. 2017-04-04]. Dostupné z: <http://www.xilinx.com/products/intellectual-property/picoblaze.html>
- [15] MicroBlaze Soft Processor Core. Xilinx.com [online]. Xilinx Inc.: Xilinx, 2017 [cit. 2017-04-04]. Dostupné z: <https://www.xilinx.com/products/design-tools/microblaze.html>
- [16] ZedBoard. Zedboard.org [online]. Avnet: Avnet, 2017 [cit. 2017-04-04]. Dostupné z: <http://zedboard.org/product/zedboard>
- [17] Zynq-7000 All Programmable SoC. Xilinx.com [online]. Xilinx Inc.: Xilinx, 2016 [cit. 2017-04-04]. Dostupné z:  
[http://www.xilinx.com/support/documentation/data\\_sheets/ds187-XC7Z010-XC7Z020-Data-Sheet.pdf](http://www.xilinx.com/support/documentation/data_sheets/ds187-XC7Z010-XC7Z020-Data-Sheet.pdf)
- [18] VÁCLAVEK, Pavel a František ŠOLC. Modelování a simulace. ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY, 2017. Dostupné také z:  
[https://e7256040-a-0ffaca9c-sites.googlegroups.com/a/vutbr.cz/bmod/soubory-ke-stazeni/bmod.pdf?attachauth=ANoY7cq0Llm93Su1zTMngLaSkk5Lu6ss7JXaLc23peoy3vM1C3wKgNduQxGYBW-EJIXJDnfHcR7lwX9IK6jy0-j3uHrLVmdccB8I9M60j1vs3ZB-GKawc34a08bFuSjN\\_idcK4T438bKTU7S09VOeMMq25843HSgNu08DiQKEMEvw7PZYu6IW1Ex9tctAE9T0sugLRuxJjk7CFrQGDBDspYKTQnWfMFQ%3D%3D&attredirects=0](https://e7256040-a-0ffaca9c-sites.googlegroups.com/a/vutbr.cz/bmod/soubory-ke-stazeni/bmod.pdf?attachauth=ANoY7cq0Llm93Su1zTMngLaSkk5Lu6ss7JXaLc23peoy3vM1C3wKgNduQxGYBW-EJIXJDnfHcR7lwX9IK6jy0-j3uHrLVmdccB8I9M60j1vs3ZB-GKawc34a08bFuSjN_idcK4T438bKTU7S09VOeMMq25843HSgNu08DiQKEMEvw7PZYu6IW1Ex9tctAE9T0sugLRuxJjk7CFrQGDBDspYKTQnWfMFQ%3D%3D&attredirects=0)
- [19] Fixed point arithmetic. Root.cz [online]. iinfo.cz: Root.cz, 2017 [cit. 2017-04-14]. Dostupné z: <https://www.root.cz/clanky/fixed-point-arithmetic/>
- [20] CHEE-MUN, Ong. Dynamic Simulation of Electric Machinery. 1. Upper Saddle River, New Jersey 07458: PRENTICE HALL PTR, 1997. ISBN 0-13-723785-5.
- [21] How does an Induction Motor Work ? Learnengineering.org [online]. LearnEngineering.org & Imajey Consulting Engineers Pvt. Ltd.: LearnEngineering.org & Imajey Consulting Engineers Pvt., 2012 [cit. 2017-04-28]. Dostupné z: <http://www.learnengineering.org/2013/08/three-phase-induction-motor-working-squirrel-cage.html>
- [22] SEUNG-KI, Sul. Control of Electric Machine Drive Systems. 1. Wiley-IEEE Press: Wiley-IEEE Press, 2011. ISBN 978-0-470-5979-9.

- [23] PETERSON, Bo. Oscillations in Inverter Fed Induction Motor Drives. Lund Institute of Technology, 1991. Dissertation. Lund Institute of Technology. Vedoucí práce Supervisor Gustaf Olsson, Jaroslav Valis.
- [24] BOUTROS, Antoine, Philippe EL-JURDI, Hadi Y. KANAAN a Kamal AL-HADDAD. Modeling and Simulation of a Complex Mechanical Load Using the Multi-Mass Approach. In: 17th IEEE Mediterranean Electrotechnical Conference [online]. Beirut, Lebanon: Mediterranean Electrotechnical Conference (MELECON), 2014 17th IEEE, 2014, s. 7 [cit. 2017-04-28]. DOI: 10.1109/MELCON.2014.6820563. ISBN 978-1-4799-2337-3. ISSN 14. Dostupné z:  
<http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6820563>
- [25] BLAHA, Petr a Petr VAVŘÍN. Řízení a Regulace I. USTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY, 2014. Dostupné také z:  
<https://docs.google.com/viewer?a=v&pid=sites&srcid=ZGVmYXVsdGRvbWFpbXxyaXplbmlhcmVndWxhY2UxfGd4OjYxNTNlYmMONTJjYmViZTE>

# Seznam zkratek, symbolů a veličin

<i>HIL</i>	Hardware In the Loop
<i>BLDC</i>	Brushles DC
<i>ACIM</i>	Asynchronní motor
<i>HW</i>	Hardware
<i>SW</i>	Software
<i>FPGA</i>	Field Programmable Gate Array
<i>LUT</i>	Look Up Table
<i>BRAM</i>	Block Random Access memory
<i>DFP</i>	D Flip Flop
<i>DSP</i>	Digital Signal Process
<i>CLB</i>	Configurable Logic Block
<i>FPU</i>	Floating Point unit
$T_s$	Perioda vyorkování
<i>IP core</i>	Inteligent Property core
$u_{xyz}$	Fázové napětí BLDC motoru
$i_{xyz}$	fázové proudy BLDC motoru
$\psi_{xyz}$	Spřažené toky BLDC motoru
$\psi_{Bxyz}$	Magnetické toky permanentního magnetu statoru BLDC motoru
$\theta$	úhel natočení
<i>R</i>	Elektrický odpor vinutí
<i>L</i>	Indukčnost vinutí
$\omega_{el}$	Elektrické otáčky
$\omega_{mech}$	Mechanické otáčky
<i>P</i>	Počet pólových dvojic
<i>J</i>	Moment setrvačnosti motoru
$C_e$	Konstanta stroje BLDC motoru
$M_z$	Zátěžný moment
$N_u$	Měřítka napětí
$N_i$	Měřítka proudu
$N_R$	Měřítka odporu
$N_L$	Měřítka indukčnosti
$N_{C_e}$	Měřítka konstanty stroje
$N_{T_s}$	Měřítka periody vzorkování
$N_\omega$	Měřítka otáček
$N_M$	Měřítka momentu
<i>UART</i>	Universal Asynchronous Receiver Transmitter
<i>AXI</i>	Sběrníkové rozhraní mezi ARM a FPGA



$\omega_s$	synchronní otáčky asynchronního motoru
$\omega_r$	otáčky rotoru asynchronního motoru
$s$	skluz asynchronního motoru
$i_{s\alpha}$	statorový proud ve složce $\alpha$ asynchronního motoru
$i_{s\beta}$	statorový proud ve složce $\beta$ asynchronního motoru
$\psi_{r\alpha}$	rotorový tok ve složce $\alpha$ asynchronního motoru
$\psi_{r\beta}$	rotorový tok ve složce $\beta$ asynchronního motoru
$K_r$	substituční vztah pro model asynchronního motoru
$K_l$	substituční vztah pro model asynchronního motoru
$T_r$	substituční vztah pro model asynchronního motoru
$L_m$	vzájemná indukčnost rotoru a statoru asynchronního motoru
$R_r$	ekvivalentní odpor rotoru asynchronního motoru
$R_s$	odpor statorového vinutí asynchronního motoru
$L_s$	Indukčnost statorového vinutí asynchronního motoru
$L_r$	Ekvivalentní indukčnost rotoru asynchronního motoru
$m_i$	Elektrický moment
$N_{K_l}$	Měřítko substitučního vztahu pro model asynchronního motoru
$N_{K_r}$	Měřítko substitučního vztahu pro model asynchronního motoru
$N_{T_r}$	Měřítko substitučního vztahu pro model asynchronního motoru
$N_\psi$	Měřítko magnetického toku asynchronního motoru
$T_{sm}$	Perioda vzorkování pro mechanický model pro MIL simulaci
$k$	Koeficient pružnosti mechanické pružné spojky
$\beta$	Koeficient tlumení mechanické pružné spojky
$J_z$	Moment setrvačnosti na výstupu mechanické pružné spojky
$N_k$	Měřítko koeficientu pružnosti mechanické pružné spojky
$N_\beta$	Měřítko koeficientu tlumení mechanické pružné spojky
$\omega_1$	Vstupní otáčky mechanické pružné spojky
$\omega_2$	výstupní otáčky mechanické pružné spojky
$K$	proporcionální zesílení PI regulátoru
$T_i$	Časová integrační konstanta PI regulátoru
$N_K$	měřítko proporcionálního zesílení PI regulátoru
$N_{T_i}$	Měřítko integrační časové konstanty PI regulátoru

# Seznam příloh

Příloha 1. IP Core periferie BLDC motoru, \*.zip archiv

Příloha 2. . IP Core periferie BLDC motoru, zdrojové kódy

Příloha 3. . IP Core periferie asynchronního motoru, \*.zip archiv

Příloha 4. . IP Core periferie asynchronního motoru, zdrojové kódy

Příloha 5. . CD s elektronickou verzí práce