

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE



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

DIAGNOSTIKA MEZI-ZÁVITOVÉHO ZKRATU VEKTOROVĚ ŘÍZENÉHO PMS MOTORU POMOCÍ UI

INTER TURN SHORT-CIRCUIT DETECTION IN VECTOR CONTROLLED PMS MOTOR USING AI

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Lukáš Zezula

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Petr Blaha, Ph.D.

BRNO 2020

Bakalářská práce

bakalářský studijní program **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

Student: Lukáš Zezula

ID: 203380

Ročník: 3

Akademický rok: 2019/20

NÁZEV TÉMATU:

Diagnostika mezi-závitového zkratu vektorově řízeného PMS motoru pomocí UI

POKYNY PRO VYPRACOVÁNÍ:

1. Seznamte se s matematickými modely synchronního motoru s permanentními magnety (PMS). Zaměřte se na matematický popis mezi-závitového zkratu.
2. Vytvořte model PMS motoru, který umožňuje simulovat mezi-závitový zkrat v prostředí MATLAB Simulink a Simscape.
3. Realizujte vektorové řízení tohoto modelu.
4. Seznamte se se současným stavem poznání v oblasti diagnostiky mezi-závitového zkratu PMS motoru metodami umělé inteligence.
5. Odsimulujte chování PMS motoru řízeného vektorovým řízením při mezi-závitovém zkratu. Získaná data označte a zpracujte metodami umělé inteligence.
6. Podobně zpracujte data z reálných experimentů na PMS motoru s možností emulace poruch.
7. Zhodnoťte dosažené výsledky.

DOPORUČENÁ LITERATURA:

- [1] Ertel, W.: Introduction to Artificial Intelligence. Springer International Publishing, 2017. ISBN 978-3-319-58-87-4.
- [2] Sul, S. K.: Control of Electric Machine Drive Systems, Wiley-IEEE Press, 2011, ISBN: 978-0-470-59079-9

další dle doporučení vedoucího práce

Termín zadání: 3.2.2020

Termín odevzdání: 8.6.2020

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

doc. Ing. Václav Jirsík, CSc.
předseda rady studijního programu

UPOZORNĚNÍ:

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

ABSTRAKT

Tato práce se zabývá diagnostikou mezizávitových zkratů na vektorově řízeném synchronním motoru s permanentními magnety. Mezizávitové zkraty jsou detekovány pomocí přepracované konvoluční neuronové sítě GoogLeNet na základě obrazů vzniklých předzpracováním signálů fázových proudů, výstupních napětí střídače a elektrické úhlové rychlosti. Předzpracování signálů mimo jiné zahrnuje digitální filtraci, převzorkování a Vlnkovou transformaci. Pro účely učení sítě je vytvořen model pohonu schopný simulovat mezizávitové zkraty. Síť je poté učena na datech ze simulace a validována na datech měřených na skutečném pohonu, který je schopný emulovat poruchy. Výsledky diagnostiky jsou poté společně s hlavními problémy prezentovány v závěru práce.

KLÍČOVÁ SLOVA

synchronní motor s permanentními magnety, mezizávitové zkraty, vektorové řízení, konvoluční neuronové sítě, diagnostika poruch elektromotorů, Vlnková transformace

ABSTRACT

This thesis deals with the diagnostics of inter turn faults in a vector controlled synchronous motor with permanent magnets. Inter turn faults are detected by the pretrained convolution neural network GoogLeNet from adequately preprocessed signals of phase currents, inverter voltages and electrical angular velocity. Signal preprocessing includes, but is not limited to digital filtration, resampling and Wavelet transform. For the purpose of network training a drive system model is created, capable of simulating inter turn faults. The network is then trained on the simulated data and later validated with data measured on a real drive system, capable of emulating faults. The results of the diagnostics, together with the main problems are presented in the conclusion.

KEYWORDS

permanent magnet synchronous motor (PMSM), Inter turn faults (ITF), Inter turn short-circuits, vector control, convolutional neural networks (CNN), electric motor fault diagnostics, Wavelet transform

ZEZULA, Lukáš. *Diagnostika mezi-závitového zkratu vektorově řízeného PMS motoru pomocí UI*. Brno, 2020, 60 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce: doc. Ing. Petr Blaha, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Diagnostika mezi-závitového zkratu vektorově řízeného PMS motoru pomocí UI“ jsem vypracoval samostatně pod vedením vedoucího bakalářské 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 bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

podpis autora

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu doc. Ing. Petru Blahovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci. Velký dík patří i panu Ing. Matúši Kozovskému za poskytnutí možnosti měření dat na reálném pohonu, pomoc s identifikací parametrů motoru a podnětné rady k problematice modelování mezizávitových zkratů.

Obsah

Úvod	11
1 Literární rešerše	13
1.1 Matematické modely synchronního motoru s permanentními magnety	13
1.1.1 <i>abc</i> model motoru	13
1.1.2 <i>dq</i> model motoru	15
1.1.3 Matematický popis mezizávitového zkratu v <i>abc</i> modelu . . .	15
1.1.4 Modelování mechaniky rotoru	18
1.2 Vektorové řízení	19
1.2.1 Signálové transformace	20
1.3 Vlnková transformace	22
1.4 Hluboké učení	24
1.4.1 Architektura konvolučních neuronových sítí	24
1.4.2 GoogLeNet	27
2 Řešení práce	28
2.1 Úvod do diagnostiky mezizávitových zkratů	28
2.2 Tvorba modelu	30
2.2.1 Model motoru v prostředí Simscape Component	31
2.2.2 Model mechaniky rotoru v prostředí Simscape a Simulink . . .	32
2.2.3 Návrh řízení v prostředí Simulink	33
2.2.4 Srovnání simulací s výsledky měření na reálném pohonu . . .	35
2.3 Předzpracování signálů	37
2.3.1 Filtrace a převzorkování signálů	37
2.3.2 Signálové obálky a Vlnková transformace	40
2.3.3 Kompozice obrazu	42
2.4 Diagnostika zkratů pomocí GoogLeNet	43
2.4.1 Obrazový set	43
2.4.2 Modifikace originální sítě	44
2.4.3 Učení modifikované sítě	45
2.4.4 Výsledky diagnostiky	48
Závěr	49
Literatura	51

Seznam symbolů, veličin a zkratk	53
Značky a symboly	53
Indexy	54
Zkratky	55
Seznam příloh	56
A Přílohy ke kapitole Tvorba modelu	57
B Přílohy ke kapitole Předzpracování signálů	59

Seznam obrázků

1	Blokové schéma znázorňující nástin řešení práce	12
1.1	IPMSM s mezizávitovým zkratem na fázi 'a'	16
1.2	Blokové schéma vektorového řízení	19
1.3	Rozmístění statorových fází v $\alpha\beta$ souřadném systému	20
1.4	Mateřská vlnka Mexican hat a z ní odvozené dceřinné vlnky	23
1.5	Příklad 2-D diskrétní konvoluce	25
1.6	Funkce ReLU	26
1.7	Optimální lokální architektura sítě GoogLeNet - Inception module . .	27
1.8	Architektura konvoluční neuronové sítě GoogLeNet	27
2.1	Model mechaniky rotoru v programu MATLAB Simulink a Simscape	32
2.2	Odezva proudové smyčky na skok řízení a akční zásah napětí	34
2.3	Odezva otáčkové smyčky na skok řízení a akční zásah proudu q složkou	34
2.4	Srovnání výsledku fázových proudů při ITF s naměřenými daty	36
2.5	Frekvenční odezva použitého FIR filtru	38
2.6	Ukázka filtrace q složky proudu	38
2.7	Ukázka proudové a napěťové obálky	40
2.8	Ukázka škálogramu proudové a napěťové obálky	41
2.9	Průběh učení sítě na diskrétní krok 5 zkratovaných závitů	46
2.10	Průběh učení sítě pro detekci hloubky zkratu	47
2.11	Průběh učení sítě pro detekci přítomnosti zkratu	47
A.1	Architektura použitých regulátorů	58
A.2	Architektura bloku zrušení vazeb	58
B.1	Ukázka obrazů pro vybrané hodnoty u , T_{load} a ω_e	60

Seznam tabulek

2.1	Parametry zkoumaného pohonu s PMSM	30
2.2	Navržené parametry regulátorů	35
2.3	Rozdíl mezi modelem a měřením vyjádřený pomocí chyby	36
2.4	Výsledky diagnostiky pro vybrané stavy na skutečném pohonu	48

Seznam výpisů

2.1	Parkova transformace realizovaná MATLAB kódem	33
2.2	Kompozice obrazu na základě škálogramů obálek a matice rychlosti .	42
2.3	Modifikace originální GoogLeNet sítě v programu MATLAB	44
2.4	Definice parametrů učení a učení sítě v programu MATLAB	45
A.1	Část definice PMSM s ITF v programu Simscape Component	57
B.1	Určení koeficientů filtru a filtrace implementovaná jako MATLAB funkce	59
B.2	Filtrace a převzorkování proudových signálů	59

Úvod

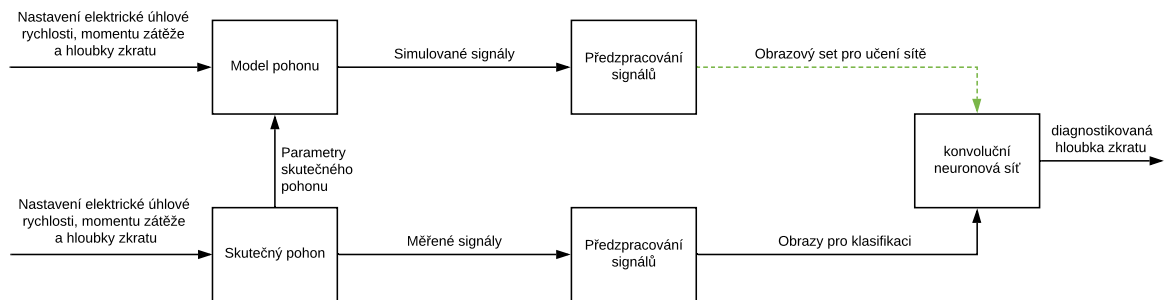
Jak již název napovídá, cílem práce je diagnostikovat mezizávitové zkraty na vektorově řízeném synchronním motoru s permanentními magnety. Mezizávitové zkraty jako takové vznikají obvykle v důsledku vniknutí vlhkosti do vnitřku motoru, tepelným průrazem izolace mezi závitů a nebo zaražením kousku nějakého tělesa (například rozpadlého ložiska) mezi závitů statorového vinutí. Díky zkratu vzniká relativně velký zkratový proud, který je doprovázen vysokým ohřevem vinutí a podílí se na dalším šíření zkratu. Toto může vyústit až ve vznícení motoru.

Na první pohled se může zdát zbytečné používat umělou inteligenci pro detekci vzrůstu proudu. Nesmí být ovšem opomenut fakt, že motor je nějakým způsobem řízen. Většina řídicích algoritmů (včetně použitého vektorového řízení) vychází z předpokladu symetrie motoru. Jakmile je tato symetrie porušena (například vlivem mezizávitového zkratu), začne řídicí algoritmus na nesymetrii pohlížet jako na poruchu, kterou se snaží kompenzovat. Rozhodnout o závažnosti zkratu již není tak jednoduché, neboť se na tvaru průběhů proudů podílí další množství faktorů. Mezi nimiž je nutno vyzdvihnout zátěžný moment a úhlovou rychlost motoru. Je-li uvážena aplikace pohonu například v elektromobilech, kde dochází k častým změnám úhlové rychlosti a zátěžného momentu, nemusejí být deterministické metody vždy použitelné. Aby tedy aplikace umělé inteligence nabyla na významu, bude diagnostika v rámci této práce nejenom vypovídat o závažnosti zkratu, ale bude také nezávislá na hodnotách úhlové rychlosti a zátěžného momentu. Toho bude dosaženo kvantifikováním vlivů zmíněných veličin a zahrnutím do diagnostiky.

Jako nástroj pro klasifikaci hloubky zkratu bude použita konvoluční neuronová síť. Tento typ sítě byl zvolen na základě rozšířené softwarové podpory. K dispozici je celá škála předučných konvolučních neuronových sítí, které stačí pro účely diagnostiky zkratů pouze mírně modifikovat. Dalším z faktorů volby tohoto typu sítě je vysoká přesnost, které tyto sítě dosahují v oblasti klasifikace obrazu.

Zkraty budou diagnostikovány na základě vhodně předzpracovaných signálů fázových proudů, PWM (pulse width modulation) napětí a elektrické úhlové rychlosti. Na základě těchto signálů budou vytvářeny obrazy, se kterými bude dále pracovat CNN (convolutional neural network). Pro účely učení sítě bude vytvořen model vektorově řízeného PMSM (permanent magnet synchronous motor), který je schopný simulovat průběhy signálů při ITF (inter turn fault). Zmíněný model bude použitelný pro SPMSM (surface-mounted permanent magnet synchronous motor) i IPMSM (interior permanent magnet synchronous motor). Simulované signály budou vhodně předzpracovány a vzniklý obrazový set bude použit pro učení neuronové sítě. Toto předzpracování bude zahrnovat mimo jiné i digitální filtraci pomocí FIR (finite impulse response) filtru, převzorkování, Vlnkovou transformaci

a kompozici obrazu. Obdobným způsobem budou předzpracovány signály měřené na skutečném pohonu a pomocí vzniklých obrazů budou detekovány závažnosti zkratu. Blokové schéma znázorňující nástin řešení celé práce lze vidět na obrázku 1.



Obr. 1: Blokové schéma znázorňující nástin řešení práce

Celá práce je rozdělena na dvě části. První část (viz kapitola 1) obsahuje teoretické zázemí pro modelování mezizávitových zkratů, vektorové řízení, Vlnkovou transformaci použitou pro předzpracování signálů a konvoluční neuronové sítě. V druhé části práce (viz kapitola 2) je popsáno řešení práce jako takové. V rámci této části je více popsán vliv mezizávitových zkratů na signály pohonu s PMSM, modelování pohonu v programu MATLAB Simulink a Simscape, předzpracování signálů a diagnostika zkratů pomocí předučené konvoluční neuronové sítě GoogLeNet.

1 Literární rešerše

Tato část práce bude zaměřená na úvod do problematiky střídavých regulačních pohonů, Vlnkové transformace a hlubokého učení. Budou probrány matematické modely PMSM a PMSM s mezizávitovým zkratem. Bude rozebrán princip vektorového řízení a signálové transformace pro převod mezi modely motoru. Následovat bude kapitola o Vlnkové transformaci se zaměřením na spojitou Vlnkovou transformaci a detailní popis vlnky použité v předzpracování signálů. V poslední kapitole této části bude probráno hluboké učení a konvoluční neuronové sítě. Detailně bude rozebrána CNN GoogLeNet, která je v práci použita pro diagnostiku zkratů.

1.1 Matematické modely synchronního motoru s permanentními magnety

V několika posledních letech se výrazně rozšiřují aplikace synchronních motorů, především potom motorů s permanentními magnety. Tyto se staly standardní pohonnou jednotkou v průmyslových servopohonech a v současné době se jejich uplatnění rychle rozšiřuje i v oblasti dopravní techniky. Mezi jejich výhody patří menší objem, hmotnost a moment setrvačnosti. Dále potom dostatečná momentová přetížitelnost a vysoká účinnost.

Obecně se používají tři modely motoru: třífázový model motoru abc , model ve statorovém souřadném systému $\alpha\beta$ a model v rotorovém souřadném systému dq . Převádět modely mezi sebou je možné na základě transformačních matic nebo na základě komplexního prostorového vektoru. Více se o odvozování modelů zmiňuje Sul v [1]. V této práci bude používán abc model motoru, protože diagnostika zkratů probíhá na základě fázových proudů a PWM napětí. Další výhodou je potom fakt, že popis mezizávitových zkratů lze nejnadhěji realizovat v abc modelu. Naproti tomu dq model motoru má přímou vazbu na vektorové řízení a lépe vystihuje chování motoru z pohledu regulace.

1.1.1 abc model motoru

Tento model popisuje PMSM z pohledu napětí a proudů statorovými fázemi. Obecně existuje několik variant tohoto modelu lišících se zejména definicí magnetického toku. Definice magnetického toku závisí na způsobu, jakým jsou vinuty fáze ve vzduchové mezeře reálného motoru. V této práci uvažujeme definici magnetického toku korespondující s transformacemi popsány v kapitole 1.2.1. Výsledný matematický model je poté sestaven na základě [4] a [5].

Napětové rovnice lze zapsat v maticovém tvaru pomocí rovnice 1.1.

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \mathbf{R}_s \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \mathbf{L}_s \frac{d}{dt} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix} \quad (1.1)$$

kde

$$\mathbf{R}_s = \begin{bmatrix} R_s & 0 & 0 \\ 0 & R_s & 0 \\ 0 & 0 & R_s \end{bmatrix}$$

$$\mathbf{L}_s = \begin{bmatrix} L_{sm} - L_{fl} \cos(2\theta_e) & -\frac{L_{sm}}{2} - L_{fl} \cos(2\theta_e - \frac{2\pi}{3}) & -\frac{L_{sm}}{2} - L_{fl} \cos(2\theta_e + \frac{2\pi}{3}) \\ -\frac{L_{sm}}{2} - L_{fl} \cos(2\theta_e - \frac{2\pi}{3}) & L_{sm} - L_{fl} \cos(2\theta_e + \frac{2\pi}{3}) & -\frac{L_{sm}}{2} - L_{fl} \cos(2\theta_e) \\ -\frac{L_{sm}}{2} - L_{fl} \cos(2\theta_e + \frac{2\pi}{3}) & -\frac{L_{sm}}{2} - L_{fl} \cos(2\theta_e) & L_{sm} - L_{fl} \cos(2\theta_e - \frac{2\pi}{3}) \end{bmatrix}$$

$$\begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix} = \frac{d\mathbf{L}_s}{dt} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \omega_e \psi_{pm} \begin{bmatrix} -\sin(\theta_e) \\ -\sin(\theta_e - \frac{2\pi}{3}) \\ -\sin(\theta_e + \frac{2\pi}{3}) \end{bmatrix}$$

R_s je elektrický odpor statorové fáze, ψ_{pm} je magnetický tok od permanentních magnetů, θ_e je elektrický úhel natočení a ω_e je elektrická úhlová rychlost. Dále se v modelu vyskytují indukčnosti L_{sm} a L_{fl} . Indukčnost L_{sm} zahrnuje vlastní a vzájemnou indukčnost statorové fáze. Indukčnost L_{fl} vyjadřuje amplitudu indukčností fluktuace vznikající rotací rotoru. Obecně lze tyto indukčnosti měřit, je to ale velice nepraktické. Výhodné je měřit indukčnost v příčné a podélné ose a L_{sm} a L_{fl} dopočítat pomocí rovnice 1.2.

$$\begin{aligned} L_{sm} &= \frac{L_q + L_d}{3} \\ L_{fl} &= \frac{L_q - L_d}{3} \end{aligned} \quad (1.2)$$

Pro SPMSM, který má magnety připevněné na povrchu rotoru, jsou indukčnosti v podélné a příčné ose shodné a fluktuace se v abc modelu neprojeví.

Generovaný krouticí moment lze v abc modelu vypočítat jako parciální derivaci celkové energie dle elektrického úhlu natočení násobenou počtem pólových dvojic. Generovaný moment se potom skládá z části generovanou permanentními magnety a z části generovanou fluktuací indukčnosti. Matematicky lze toto zapsat pomocí rovnice 1.3.

$$T = p \left(\begin{bmatrix} i_a & i_b & i_c \end{bmatrix} \frac{\partial \mathbf{L}_s}{\partial \theta_e} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} i_a & i_b & i_c \end{bmatrix} \psi_{pm} \begin{bmatrix} -\sin(\theta_e) \\ -\sin(\theta_e - \frac{2\pi}{3}) \\ -\sin(\theta_e + \frac{2\pi}{3}) \end{bmatrix} \right) \quad (1.3)$$

1.1.2 *dq* model motoru

Výsledkem aplikace transformačních matic, popsanych v kapitole 1.2.1, na vektory fázových napětí a proudů *abc* modelu je *dq* model. Tento model představuje náhradu PMSM stejnosměrným motorem s cizím buzením. Model v rotorovém souřadném systému lze potom stanovit na základě elektrického odporu statorové fáze R_s , magnetického toku od permanentních magnetů ψ_{pm} , indukčnosti v příčné ose L_q a indukčnosti v podélné ose L_d . Indukčnosti v příčné a podélné ose je výhodné měřit, protože v sobě přenáší informaci nejenom o vlastní a vzájemné indukčnosti statorové fáze, ale i o fluktuaci indukčnosti s úhlem natočení rotoru. Více o měření elektrických parametrů PMSM pojednává [3]. Napěťové rovnice *dq* modelu potom nabývají dle [1] tvaru:

$$\begin{aligned}v_d &= R_s i_d + L_d \frac{di_d}{dt} - \omega_e i_q L_q \\v_q &= R_s i_q + L_q \frac{di_q}{dt} + \omega_e (i_d L_d + \psi_{pm})\end{aligned}\tag{1.4}$$

Generovaný krouticí moment je potom dán rovnicí 1.5.

$$T = \frac{3}{2} p (i_q \psi_{pm} + i_q i_d (L_d - L_q))\tag{1.5}$$

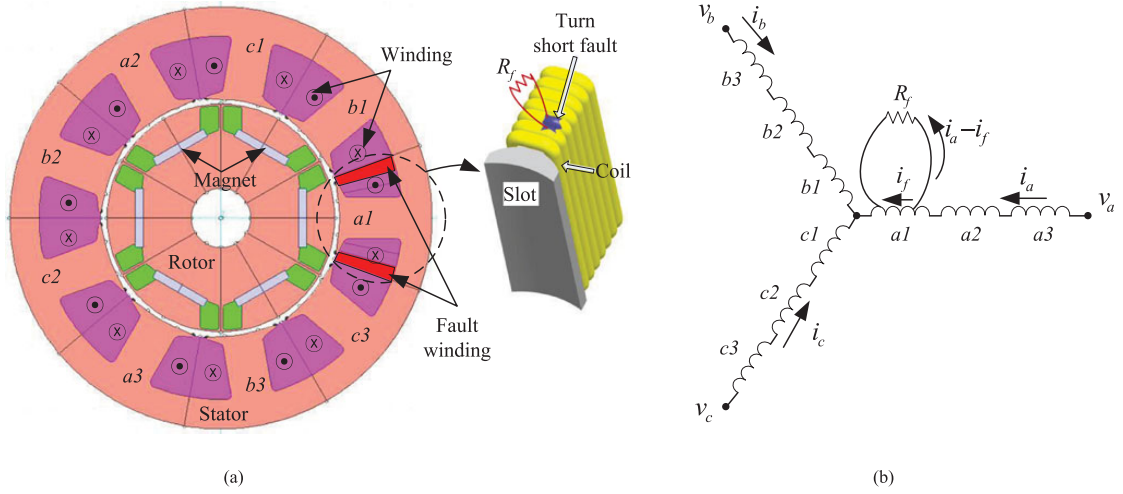
V případě, že se jedná o SPMSM platí, že $L_d = L_q$. V tomto případě nedochází k fluktuaci indukčnosti s úhlem natočení rotoru a generovaný moment je závislý pouze na *q* složce proudu.

1.1.3 Matematický popis mezizávitového zkratu v *abc* modelu

Pro simulaci ITF se ukazuje být nejhodnější použití modelů motoru vytvořených pomocí metody konečných prvků. Ve srovnání s matematickými modely jsou FEM (finite element method) modely daleko přesnější, protože pracují s geometrií konkrétního motoru. Cílem této práce je však vytvořit model, který bude možné parametrizovat na základě měřených dat a výrobní dokumentace zkoumaného motoru. Toto vylučuje použití FEM modelů.

Většina konvenčně užívaných matematických popisů ITF, které jsou odvozeny od „zdravých“ modelů motoru, dosahují jen malé přesnosti. Toto je dáno tím, že „zdravé“ modely, ze kterých jsou popisy ITF odvozeny, vycházejí z předpokladu symetrie motoru. Tento předpoklad dovoluje v modelu zanedbat například fakt, že statorová fáze reálného motoru se skládá z několika sériově/paralelně řazených cívek rozmístěných v motoru například tak, jak ukazuje obrázek 1.1 a). Mimo rozvážení magnetických vazeb mezi fázemi, dojde při ITF také k rozvážení

magnetických vazeb mezi cívkami na jedné fázi. Zahnutí tohoto faktu do modelu významně zvýší přesnost simulace. Schéma třífázového motoru, který má fáze složeny ze tří cívek a obsahuje mezizávitový zkrat, lze vidět na obrázku 1.1 b).



Obr. 1.1: a) Příčný řez IPMSM (3 fáze, 9 slotů, 6 pólů), b) Sériově řazená vinutí na fázích motoru s ITF na fázi 'a' [5]

Výsledný matematický model byl odvozen na základě [5] a [6]. Pro popis hloubky zkratu slouží parametr u , který udává poměr počtu zkratovaných závitů na jedné cívce ku celkovému počtu závitů jedné cívky. Parametr n udává počet sériově řazených cívek na jedné fázi a parametr γ kvantifikuje magnetické vazby mezi cívkami na jedné fázi. Více se o γ zmiňuje Gu v [6]. Tento parametr lze získat na základě měření indukčností, FEM simulace, nebo magneticky ekvivalentního modelu motoru. Magneticky ekvivalentnímu modelu PMSM se více věnuje [7]. Dalším významným parametrem je poté odpor zkratu R_f , který udává hodnotu elektrického odporu zkratu na izolaci. Význam ostatních parametrů je stejný jako u „zdravého“ modelu motoru (viz kapitola 1.1.1).

V této práci je uvažován model, který je schopný simulovat mezizávitový zkrat na fázi 'a' v rozsahu jedné celé zkratované cívky. Maximální hodnota zkratu $u=1$ tedy odpovídá zkratování $\frac{1}{n}$ celého fázového vinutí. Napěťové rovnice poškozeného motoru potom nabývají tvaru dle rovnice 1.6.

$$\begin{bmatrix} v_a \\ v_b \\ v_c \\ 0 \end{bmatrix} = \mathbf{R}_s \begin{bmatrix} i_a \\ i_b \\ i_c \\ i_f \end{bmatrix} + \mathbf{L}_s \frac{d}{dt} \begin{bmatrix} i_a \\ i_b \\ i_c \\ i_f \end{bmatrix} + \begin{bmatrix} e_a \\ e_b \\ e_c \\ e_f \end{bmatrix} \quad (1.6)$$

Kde

$$\mathbf{R}_s = \begin{bmatrix} R_{s11} & 0 & 0 & R_{s14} \\ 0 & R_{s22} & 0 & 0 \\ 0 & 0 & R_{s33} & 0 \\ R_{s41} & 0 & 0 & R_{s44} \end{bmatrix} \quad \mathbf{L}_s = \begin{bmatrix} L_{s11} & L_{s12} & L_{s13} & L_{s14} \\ L_{s21} & L_{s22} & L_{s23} & L_{s24} \\ L_{s31} & L_{s32} & L_{s33} & L_{s34} \\ L_{s41} & L_{s42} & L_{s43} & L_{s44} \end{bmatrix}$$

$$R_{s11} = R_s - \frac{u}{n}R_s + R_f; \quad R_{s14} = R_{s41} = -R_f; \quad R_{s22} = R_{s33} = R_s; \quad R_{s41} = \frac{u}{n}R_s + R_f$$

$$L_{sxy} = \begin{cases} k_{xy}(L_{sm} - L_{fl} \cos(2\theta_e)) & (x, y) = (1, 1); (1, 4); (4, 1); (4, 4) \\ k_{xy}(L_{sm} - L_{fl} \cos(2\theta_e + \frac{2\pi}{3})) & (x, y) = (2, 2) \\ k_{xy}(L_{sm} - L_{fl} \cos(2\theta_e - \frac{2\pi}{3})) & (x, y) = (3, 3) \\ k_{xy}(L_{sm} + 2L_{fl} \cos(2\theta_e)) & (x, y) = (2, 3); (3, 2) \\ k_{xy}(L_{sm} + 2L_{fl} \cos(2\theta_e - \frac{2\pi}{3})) & (x, y) = (1, 2); (2, 1); (2, 4); (4, 2) \\ k_{xy}(L_{sm} + 2L_{fl} \cos(2\theta_e + \frac{2\pi}{3})) & (x, y) = (1, 3); (3, 1); (3, 4); (4, 3) \end{cases}$$

$$k_{11} = 1 + \frac{u^2}{n(1-\gamma)} + \frac{2u(\gamma-1)}{n(1-\gamma)};$$

$$k_{12} = k_{21} = k_{13} = k_{31} = -\frac{1}{2}\left(1 - \frac{u}{n}\right)$$

$$k_{24} = k_{42} = k_{34} = k_{43} = -\frac{1}{2}\frac{u}{n};$$

$$k_{14} = k_{41} = -\frac{u\gamma}{n(1-\gamma)} + \frac{u(1-u)}{n(1-\gamma)}$$

$$k_{22} = k_{33} = 1; \quad k_{23} = k_{32} = -\frac{1}{2};$$

$$k_{44} = \frac{u^2}{n(1-\gamma)}$$

$$\begin{bmatrix} e_a \\ e_b \\ e_c \\ e_f \end{bmatrix} = \frac{d\mathbf{L}_s}{dt} \begin{bmatrix} i_a \\ i_b \\ i_c \\ i_f \end{bmatrix} + \omega_e \Psi_{pm} \begin{bmatrix} -(1 - \frac{u}{n}) \sin(\theta_e) \\ -\sin(\theta_e - \frac{2\pi}{3}) \\ -\sin(\theta_e + \frac{2\pi}{3}) \\ -\frac{u}{n} \sin(\theta_e) \end{bmatrix}$$

Výpočet generovaného krouticího momentu je obdobný jako u klasického *abc* modelu s tím rozdílem, že moment nyní negativně ovlivňuje zkrat na vinutí. Matematicky lze výpočet momentu popsat rovnicí 1.7.

$$T = p \left(\begin{bmatrix} i_a & i_b & i_c & i_f \end{bmatrix} \frac{\partial \mathbf{L}_s}{\partial \theta_e} \begin{bmatrix} i_a \\ i_b \\ i_c \\ i_f \end{bmatrix} + \begin{bmatrix} i_a & i_b & i_c & i_f \end{bmatrix} \Psi_{pm} \begin{bmatrix} -(1 - \frac{u}{n}) \sin(\theta_e) \\ -\sin(\theta_e - \frac{2\pi}{3}) \\ -\sin(\theta_e + \frac{2\pi}{3}) \\ -\frac{u}{n} \sin(\theta_e) \end{bmatrix} \right) \quad (1.7)$$

Použitý model je platný pro IPMSM i SPMSM. Pro SPMSM platí, že $L_{fl} = 0$ a zpětně indukované napětí v rovnici 1.6 je dáno pouze časovou derivací magnetického toku od permanentních magnetů. Adekvátně k tomu generovaný moment je daný pouze změnou energie od permanentních magnetů s elektrickým úhlem natočení.

1.1.4 Modelování mechaniky rotoru

Základem mechaniky rotoru je vztah mezi mechanickou úhlovou rychlostí ω_m a mechanickým úhlem natočení θ_m . Platí, že derivace úhlu natočení podle času je rovna úhlové rychlosti. Toto popisuje rovnice 1.8.

$$\omega_m = \frac{d\theta_m}{dt} \quad (1.8)$$

U synchronních strojů je mechanická úhlová rychlost ω_m , se kterou se otáčí hřídel motoru, přímo úměrná elektrické úhlové rychlosti ω_e , se kterou rotuje magnetické pole uvnitř vzduchové mezery motoru. Na základě rovnice 1.8 lze tuto úvahu rozšířit i na elektrický a mechanický úhel natočení. Tyto závislosti vyjadřuje rovnice 1.9.

$$\begin{aligned} \omega_e &= p \omega_m \\ \theta_e &= p \theta_m \end{aligned} \quad (1.9)$$

Kde p vyjadřuje počet pólových dvojic motoru.

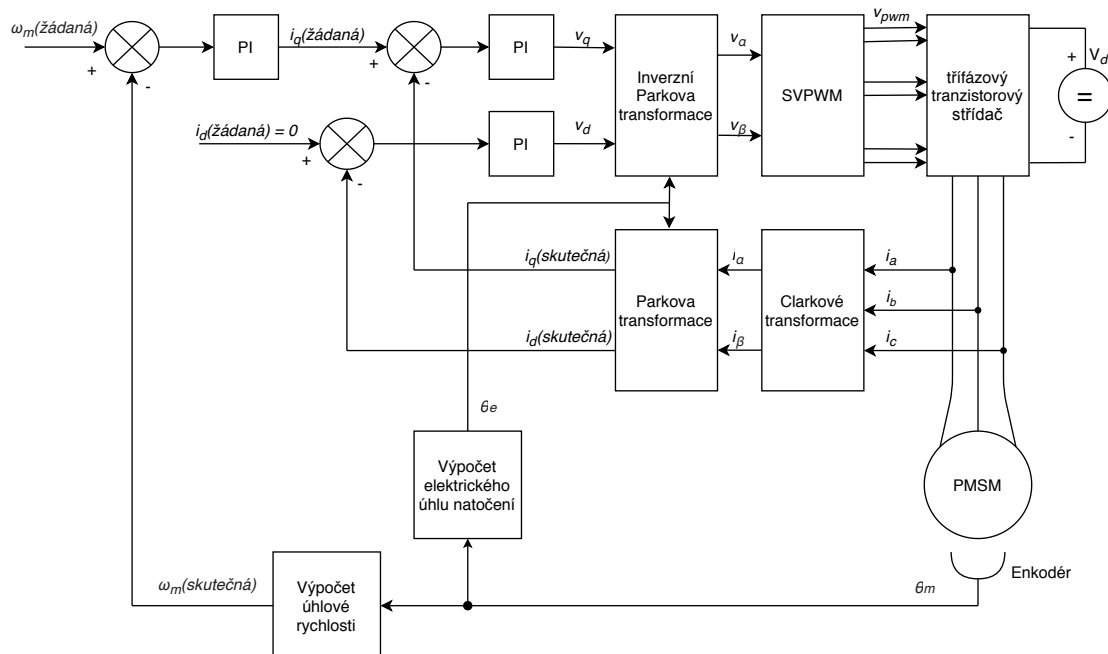
Aby bylo možné kvalitně simulovat chování elektrického pohonu, musí být zohledněna vzájemná interakce mezi hnacím elektromotorem a hnaným pracovním mechanismem. Touto interakcí se zabývá pohybová rovnice motoru. Ta říká, že těleso rotující kolem své ustálené osy je v dynamické rovnováze, je-li výslednice všech momentů působících na těleso rovna nule. Obecně na motoru působí generovaný moment T , který se snaží kompenzovat vliv dynamického momentu, statického tření, dynamického tření a zátěžového momentu. Dynamický moment je dán zrychlením a celkovým momentem setrvačnosti J na hřídeli. Tento se skládá z momentu setrvačnosti motoru a momentu setrvačnosti pracovního mechanismu. Statické tření udává hodnotu momentu, která musí být překonána, aby se motor roztočil. Dynamické tření udává hodnotu momentu, která musí být překonána, aby se motor točil požadovanými mechanickými otáčkami ω_m . Dynamické tření potom sestává ze suchého a viskózního tření. Suché tření, na rozdíl od viskózního, nezávisí na rychlosti otáčení hřídele. Zátěžový moment zahrnuje všechny ostatní vlivy. Je to obecně libovolná funkce, která může motor buď brzdit (v případě, že působí proti směru otáčení) nebo roztáčet. V této práci bude uvažován zátěžový moment, který je konstantní a působí proti směru otáčení. Pohybová rovnice mechanické části poté nabývá následujícího tvaru:

$$T = J \frac{d\omega_m}{dt} + \text{sign}(\omega_m) (B \text{abs}(\omega_m) + T_{stat}) + T_{load} \quad (1.10)$$

Kde T_{stat} představuje suché tření, B je koeficient viskózního tření a T_{load} je zátěžový moment. Statické tření není uvažováno.

1.2 Vektorové řízení

Vektorové řízení vychází z modelu PMSM v rotorovém souřadném systému dq (viz kapitola 1.1.2). Díky vektorovému řízení lze dle [1] efektivně řídit okamžité hodnoty magnetických toků a momentu stroje. Principiální schéma vektorového řízení lze vidět na obrázku 1.2.



Obr. 1.2: Blokové schéma vektorového řízení

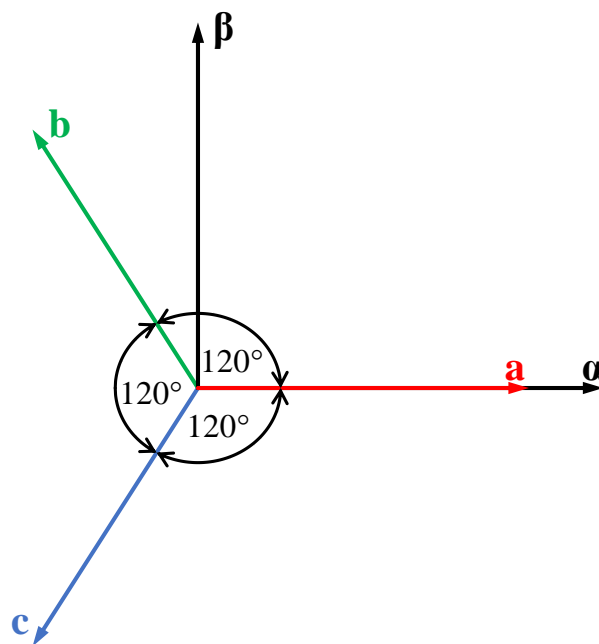
Generovaný moment je dominantně závislý na momentotvorné q složce proudu. Magnetický tok je daný permanentními magnety a tokotvornou d složkou proudu, která se majoritně používá k odbuzování motoru pro dosažení vyšších otáček. Při klasickém momentovém řízení PMSM je q složkou proudu moment řízen na požadovanou hodnotu a d složka proudu je řízena do nuly. Řízení d složky proudu do nuly zajistí, že motor nebude ani přibuzován ani odbuzován a magnetický tok bude tvořen pouze částí od permanentních magnetů. Díky tomuto je však pevně stanoven rozsah otáček, kterých je motor schopen dosáhnout. Pro řízení proudů se ve většině případů používají PI (proportional-integral), respektive PS (proportional-summation), regulátory s akčním zásahem napětí v dq osách.

V rámci této práce bylo momentové řízení rozšířeno o PS regulátor otáček s akčním zásahem proudu q složkou. Toto umožňuje řídit PMSM na požadované otáčky. Generovaný moment potom v ustáleném stavu kompenzuje třecí momenty a moment zátěže.

Aby bylo možné reálný motor vektorově řídit, musí být proudové signály transformovány do rotorového souřadného systému. Tomuto účelu slouží Clarkové a Parkova transformace, o kterých více pojednává kapitola 1.2.1. Napětí se na motor připojuje pomocí třífázového tranzistorového střídače ovládaného signály z SVPWM (space vector pulse width modulation). Tranzistorový střídač a SVPWM je více rozebrána v [8].

1.2.1 Signálové transformace

Aby bylo možné vektorově řídit reálný motor, který vždy svými vstupy a výstupy odpovídá abc modelu, je nutné adekvátním způsobem transformovat vstupní a výstupní signály. K tomuto účelu slouží Clarkové a Parkova transformace. Tvary těchto transformací jsou svázány s orientací satorového souřadného systému $\alpha\beta$ a díky tomu i s definicí magnetického toku v abc modelu motoru. V rámci této práce je satorová fáze 'a' pevně spřažena s reálnou osou satorového souřadného systému α . Magnetický tok je potom definován tak, aby při elektrickém úhlu natočení $\theta_e = 0^\circ$ byl tok fází 'a' maximální a při $\theta_e = 90^\circ$ nulový. Rozložení fází ve satorovém souřadném systému lze vidět na obrázku 1.3. Příslušný tvar Clarkové a Parkovy transformace vychází z [2].



Obr. 1.3: Rozmístění satorových fází v $\alpha\beta$ souřadném systému

Clarkové transformace

Průchod třífázového proudu třemi fixně umístěnými fázemi motoru způsobí rotaci magnetického pole ve vzduchové mezeře. Clarkové transformace vychází z poznatku, že stejné rotace magnetického pole lze dosáhnout průchodem dvoufázového proudu dvěma fázemi. Tyto fáze jsou vůči sobě v rovině posunuty o 90° . Aplikací Clarkové transformace na třífázový proud vzniká proud dvoufázový, který má stejné magnetické účinky jako původní třífázový proud. Nově vzniklé proudy se značí indexy α a β . Matematicky lze Clarkové transformaci popsat pomocí rovnice 1.11.

$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} \quad (1.11)$$

Tuto transformaci lze použít i v inverzním tvaru pro převod dvoufázového proudu na třífázový. Toto vyjadřuje rovnice 1.12

$$\begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} \quad (1.12)$$

Parkova transformace

Průchodem dvoufázového střídavého proudu dvěma fixně umístěnými fázemi vzniká rotující magnetické pole, které rotuje elektrickou úhlovou rychlostí ω_e . Stejně magnetické pole lze vytvořit i průchodem stejnosměrného proudu dvěma fázemi, které rotují úhlovou rychlostí ω_e . Parkova transformace tedy realizuje přepočít ze statického statorového souřadného systému do rotorového souřadného systému, který rotuje elektrickou úhlovou rychlostí motoru. Posun mezi statorovým a rotorovým souřadným systémem lze v daném čase vyjádřit elektrickým úhlem natočení θ_e . Aplikací Parkovy transformace na vyvážený dvoufázový proud vznikají proudy stejnosměrné. Tyto se poté označují indexy d a q . Matematicky lze Parkovu transformaci popsat pomocí rovnice 1.13.

$$\begin{bmatrix} i_d \\ i_q \end{bmatrix} = \begin{bmatrix} \cos \theta_e & \sin \theta_e \\ -\sin \theta_e & \cos \theta_e \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} \quad (1.13)$$

V inverzním tvaru lze tuto transformaci použít pro převod dvou stejnosměrných proudů na střídavý dvoufázový proud. Toto vyjadřuje rovnice 1.14.

$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \begin{bmatrix} \cos \theta_e & -\sin \theta_e \\ \sin \theta_e & \cos \theta_e \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} \quad (1.14)$$

1.3 Vlnková transformace

Vlnková transformace je stejně jako Furierova transformace integrální transformací umožňující získat časově-frekvenční popis signálu. Oproti Fourierově transformaci jsou báze funkce (vlnky) Vlnkové transformace lokální. Toto umožňuje do jisté míry lokalizovat jevy v čase. Následující kapitola vychází z [9].

Báze funkce Vlnkové transformace se říká mateřská vlnka. Mateřskou vlnkou může být libovolná funkce, která splňuje následující podmínky:

Mateřská vlnka musí oscilovat. Toto popisuje rovnice 1.15.

$$\int_{-\infty}^{\infty} \Psi(t) dt = 0 \quad (1.15)$$

Mateřská vlnka musí mít konečnou energii. Toto popisuje rovnice 1.16.

$$\int_{-\infty}^{\infty} |\Psi(t)|^2 dt \leq \infty \quad (1.16)$$

Ψ značí mateřskou vlnku.

Z mateřské vlnky se změnou měřítka (škálováním) a posunem vytvoří dceřinné vlnky $\Psi_{s,\tau}(t)$, kde s vyjadřuje škálovací parametr (měřítka) a τ vyjadřuje posun v čase. Matematicky lze odvození dceřinné vlnky z mateřské popsat pomocí následující rovnice:

$$\Psi_{s,\tau}(t) = \frac{1}{\sqrt{s}} \Psi\left(\frac{t - \tau}{s}\right) \quad (1.17)$$

Množina dceřinných vlnek tvoří poté základ Vlnkové transformace. Funkce určující měřítka se nazývá otcovská vlnka Φ . V této práci bude uvažována vždy lineární změna měřítka dceřinných vlnek.

Vlnková transformace se poté dělí na diskrétní a spojitou Vlnkovou transformaci. Spojitá Vlnková transformace uvažuje spojitě dceřinné vlnky a používá se převážně pro vykreslení časově-frekvenčního spektra signálu. Diskrétní Vlnková transformace uvažuje diskrétní (respektive diskretizované) dceřinné vlnky a používá se převážně pro digitální filtraci signálů. Aby bylo možné spojitou Vlnkovou transformaci používat na výpočetních zařízeních, musí být převedena do diskrétního tvaru. Hlavní rozdíl mezi diskretizovanou spojitou Vlnkovou transformací a diskrétní Vlnkovou transformací je potom ve způsobu, jakým je diskretizován škálovací parametr s . Tomuto tématu se podrobněji věnuje [10]. V této práci je Vlnková transformace používána pro vykreslení časově-frekvenčního spektra signálů, proto se další část této kapitoly věnuje pouze spojitě Vlnkové transformaci.

Spojité Vlnková transformace

Výstupem spojitě Vlnkové transformace aplikované na signál $x(t)$ je matice koeficientů $\mathbf{C}(s, \tau)$ s řádky odpovídajícími škálám dceřinných vlnek s a sloupci odpovídajícími posunům dceřinných vlnek τ . Matematicky lze tuto transformaci vyjádřit pomocí rovnice 1.18.

$$\mathbf{C}(s, \tau) = \int_{-\infty}^{\infty} x(t) \Psi_{s, \tau}^*(t) dt \quad (1.18)$$

Symbol * označuje komplexně sdruženou funkci k dceřinné vlnce. V rámci této práce bude jako mateřská vlnka použita vlnka Mexican hat.

Vlnka Mexican hat

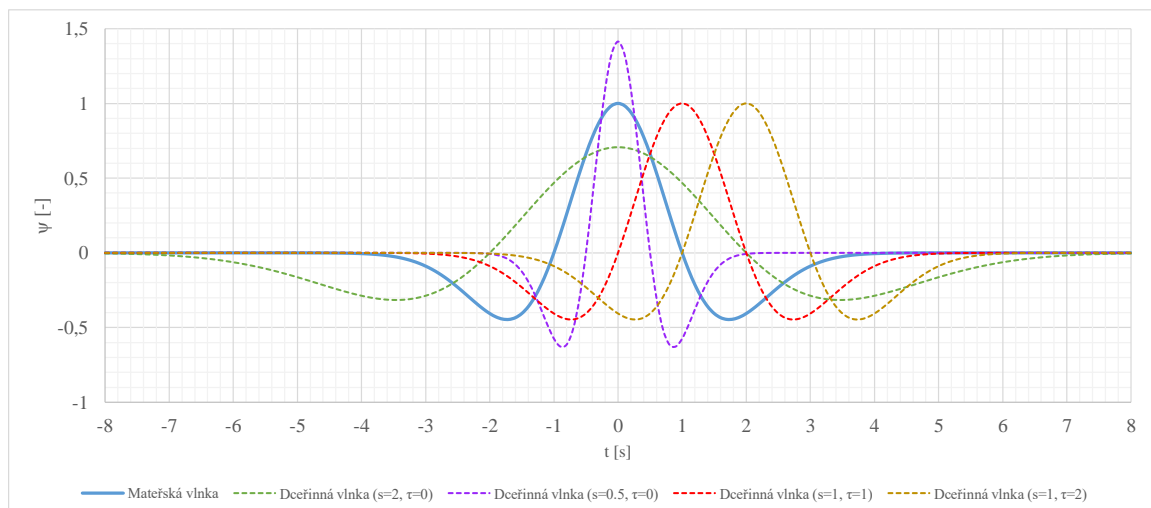
Vzhledem k charakteru signálů, na které je aplikována Vlnková transformace, byla zvolena mateřská vlnka s názvem Mexican hat. Jedná se o záporně vzatou druhou derivaci Gaussova rozložení. Všechny derivace Gaussova rozložení mohou být použity jako mateřské vlnky a jedná se o jeden z nejvíce užívaných tvarů analytických vlnek. Definicí Mexican hat vlnky lze vidět v rovnici 1.19.

$$\Psi(t) = (1 - t^2)e^{-t^2/2} \quad (1.19)$$

Škálováním a posunem potom vznikají dceřinné vlnky ve tvaru:

$$\Psi_{s, \tau}(t) = \frac{1}{\sqrt{s}} \left[1 - \left(\frac{t - \tau}{s} \right)^2 \right] e^{-\frac{1}{2}[(t - \tau)/s]^2} \quad (1.20)$$

Vlnku Mexican hat a z ní odvozené dceřinné vlnky lze vidět na obrázku 1.4.



Obr. 1.4: Mateřská vlnka Mexican hat a z ní odvozené dceřinné vlnky

1.4 Hluboké učení

Hluboké učení je metoda strojového učení, kde je za pomoci mnohvrstevných nelineárních výpočetních modelů získávána užitečná informace přímo z dat. Vyšší počet vrstev indikuje, že se jedná o hlubokou neuronovou síť (řádově desítky vrstev) a je klíčový pro extrakci drobnějších detailů ze vstupních dat. Například při zpracování obrazu mohou spodní vrstvy (vrstvy blíže ke vstupu) identifikovat hrany a vyšší vrstvy mohou identifikovat pojmy relevantní pro člověka jako jsou číslice, písmena nebo tváře. V klasifikačních úlohách dosahují hluboké neuronové sítě vysoké přesnosti, která může předčít lidské schopnosti. Nejvíce užívanou variantou hlubokých neuronových sítí jsou konvoluční neuronové sítě pro klasifikaci obrazu.

Přesnost CNN závisí z velké části na množství dat použitých k jejímu učení. Naučení celé sítě z výchozího nastavení vyžaduje ohromné množství trénovacích dat, protože je třeba nastavit miliony váhových koeficientů. Toto s sebou nese velmi dlouhou dobu učení (obvykle dny až týdny na výkonných počítačích). Takto naučená síť však dosahuje velmi vysoké přesnosti. Alternativním přístupem k učení „od nuly“ je využití „předučené“ CNN, která již sama extrahuje charakteristické rysy ze vstupních dat. Tento přístup je označován termínem transfer learning a vychází z poznatku, že úlohy klasifikace objektů v obrazových datech jsou v prvních krocích (ve spodních vrstvách sítě) v zásadě shodné. Nejprve se z obrazu extrahují jednoduché prvky (hrany, přechody) a poté složitější struktury, až ke konkrétním objektům. K přizpůsobení sítě konkrétnímu problému tedy postačí doučit pouze několik posledních vrstev, které již rozlišují mezi konkrétními objekty. Více hluboké učení a transfer learning popisuje Jirkovský v [11]. O architektuře konvolučních neuronových sítí pojednává kapitola 1.4.1.

1.4.1 Architektura konvolučních neuronových sítí

V základu se konvoluční neuronové sítě svou architekturou dělí do dvou částí. První část sítě zajišťuje extrakci charakteristických rysů pomocí konvolučních vrstev. Druhá část sítě poté realizuje klasifikaci na základě extrahovaných rysů. V této části se používají klasické klasifikační algoritmy strojového učení, jako jsou rozhodovací stromy a podobně. Nejčastěji se přechod mezi těmito vrstvami realizuje vytvořením vektoru příznaků z výstupů posledních konvolučních vrstev.

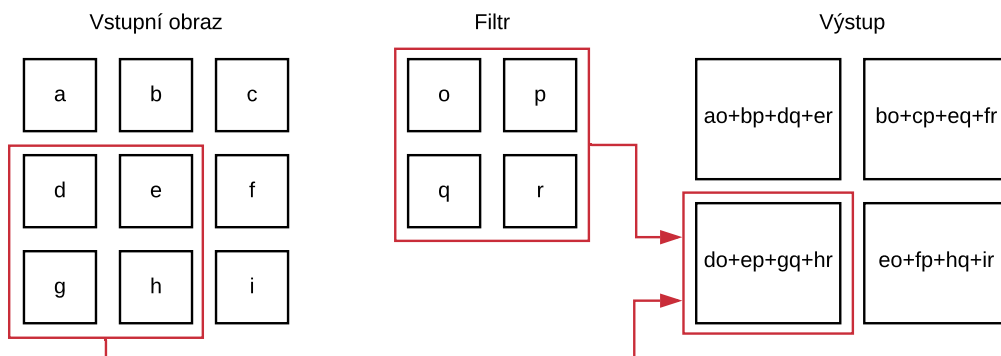
Konvoluční vrstvy

Jak již název napovídá, tyto vrstvy jsou založeny na matematické operaci konvoluci (tato se značí pomocí operátoru $*$). Konvoluce je operace mezi dvěma vstupními

signály. Výstupem této operace je třetí signál, který vyjadřuje jak je jeden ze vstupních signálů modifikován druhým. V konvolučních neuronových sítích se využívá 2-D diskrétní konvoluce (viz rovnice 1.21).

$$y(i, j) = (F * x)(i, j) = \sum_m \sum_n x(i + m, j + n)F(m, n) \quad (1.21)$$

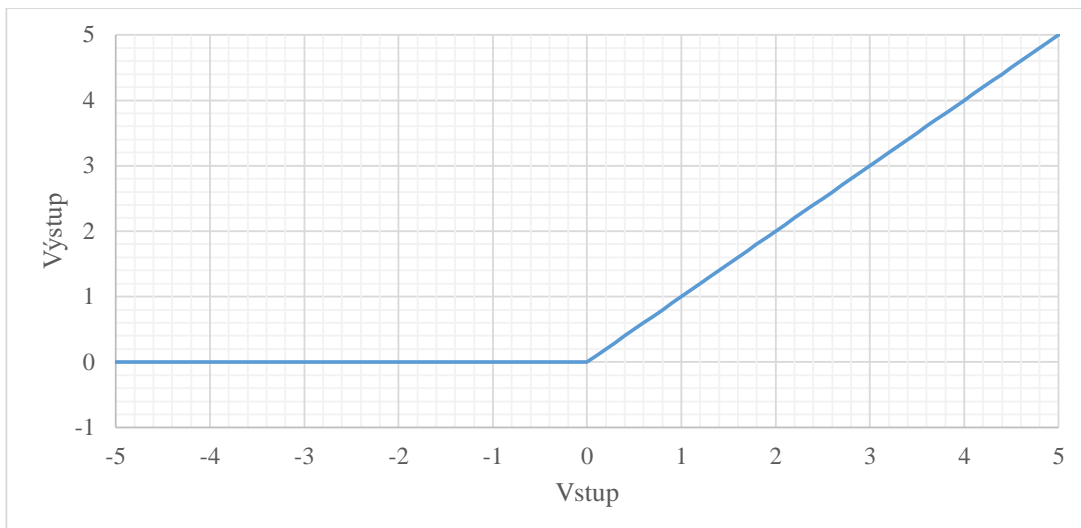
Funkce $x(i, j)$ je vstupní obraz konvoluce a $F(i, j)$ je filtr (konvoluční jádro). Koeficienty m a n potom vyjadřují posuny v rámci řádků a sloupců obrazu. Výstupní obraz $y(i, j)$ tedy vyjadřuje, jak byl vstupní signál modifikován filtrem. Graficky lze 2-D diskrétní konvoluci znázornit na obrázku 1.5.



Obr. 1.5: Příklad 2-D diskrétní konvoluce

Konvoluce využívá tři důležité myšlenky, které mohou zlepšit systém strojového učení: **řídke interakce**, **sdílení parametrů** a **ekvivariantní reprezentace**. **Řídké interakce** je dosaženo tím, že je konvoluční jádro menší než obraz. Například pro detekci hran v obrazu, který má miliony pixelů, stačí filtr jen se stovkami pixelů. Při výpočtu výstupu se potom provede méně operací (zlepšení efektivity) a ukládá se méně parametrů (snížení požadavků na paměť). **Sdílení parametrů** odkazuje na použití stejného parametru pro více než jednu funkci v modelu. Každý člen konvolučního jádra je použit při výpočtu na každé pozici vstupního obrazu. To znamená, že místo učení separátního setu parametrů pro každou pozici je učen pouze jeden set parametrů (konvoluční jádro). Konvoluční vrstva má také vlastnost zvanou **ekvivariance** k posuvu. Toto znamená, že pokud je detekovaný objekt posunut ve vstupním obrazu, je jeho reprezentace ve výstupu posunuta stejně.

Průchod obrazu konvoluční vrstvou má obecně tři etapy. V první etapě je provedeno několik paralelních konvolucí k vytvoření sady lineárních aktivací. V druhé etapě je na lineární aktivace aplikována nelineární aktivační funkce. Nejvíce užívanou aktivační funkcí v konvolučních neuronových sítích je ReLU (rectified linear function). Tuto funkci lze vidět na obrázku 1.6.



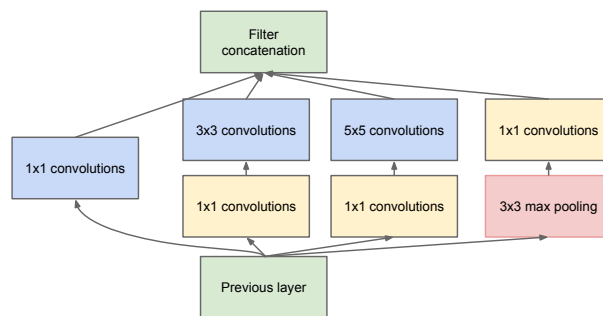
Obr. 1.6: Funkce ReLU

Oproti sigmoidě nebo tangentě hyperbolické, které jsou využívány jako aktivační funkce v běžných neuronových sítích, nemá funkce ReLU omezený výstup a konverguje až 6x rychleji. V poslední etapě je použita sdružující funkce. Tato funkce nahradí výstupy vrstvy v určitém místě obrazu sumarizovanou statistikou přilehlých výstupů. Obvykle se tato funkce využívá zároveň s podvzorkováním (určitý počet výstupů je nahrazen jedním, který obsahuje sumarizovanou statistiku). Obvykle užívané sdružující funkce jsou „max pooling“ (sdružení na základě maximální hodnoty výstupů) nebo sdružení na základě váženého průměru (váhy jsou potom odvozeny od vzdálenosti od centrálního pixelu). V každém případě sdružení napomáhá učinit výstup vrstvy nezávislý na malých posunech objektu ve vstupním obrazu. Tyto etapy lze použít v rámci jedné konvoluční vrstvy nebo jako separátní vrstvy, kde konvoluční vrstva zahrnuje pouze konvoluci jako takovou. Rozdíl mezi těmito přístupy je potom v rozměrech konvolučního jádra. Podrobněji architekturu a princip konvolučních neuronových sítí popisuje [12].

Aby při učení konvoluční neuronové sítě nedošlo k přeučení, používá se tzv. metoda dropout. Tato zajišťuje, že se při učení náhodně ignorují výstupy vrstev. Zpravidla se tato metoda používá až v posledních vrstvách sítě, kde jsou aplikovány klasické algoritmy strojového učení, nikoliv konvoluční vrstvy. Použití této metody uměle „mate“ síť. S každým průběhem se vrstva tváří, že má jiný počet uzlů. Toto zajišťuje, že se v naučené síti klasifikátor neupne pouze na jeden konkrétní neuron, ale pracuje se širokým spektrem. O metodě dropout a klasifikačních vrstvách konvolučních neuronových sítí více pojednává [12]. Metoda dropout se používá pouze při učení sítě. V rámci této práce bude použita předučená CNN GoogLeNet, o které pojednává kapitola 1.4.2.

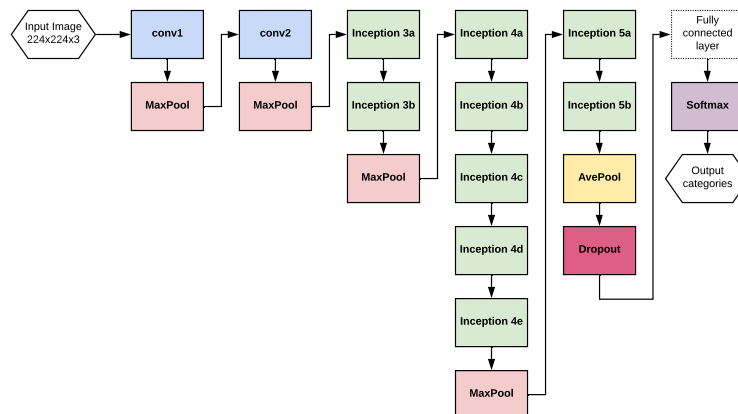
1.4.2 GoogLeNet

GoogLeNet je převedená konvoluční neuronová síť hluboká 22 vrstev (počítají se pouze vrstvy s parametry). Síť jako taková má kolem 7 milionů parametrů a je schopná rozeznávat 1000 kategorií různých objektů, jako jsou klávesnice, myši, tužky a spousta druhů zvířat. Hlavní ideou této sítě je nalézt optimální lokální architekturu a následně ji opakovat v celé síti. Optimální lokální architekturu, která nese název „Inception module“, lze vidět na obrázku 1.7. Podrobněji volbu této architektury a celou síť GoogLeNet popisuje Szegedy v [13]. Učení sítě GoogLeNet nové klasifikační úloze je popsáno v [14].



Obr. 1.7: Optimální lokální architektura sítě GoogLeNet - Inception module [13]

Architekturu celé GoogLeNet sítě lze potom vidět na obrázku 1.8.



Obr. 1.8: Architektura konvoluční neuronové sítě GoogLeNet

Funkce „Softmax“ normalizuje vektor vstupních příznaků mezi 0 a 1 tak, aby součet výstupů odpovídal hodnotě 1. Aby bylo možné tuto síť univerzálně učit novým úlohám, je před funkcí softmax doplněna ještě jedna plně propojená vrstva s počtem neuronů odpovídajícím počtu kategorií.

2 Řešení práce

V této části bude popsán přínos práce v oblasti diagnostiky mezizávitových zkratů. Nejprve bude shrnut vliv ITF na vektorově řízený PMSM a bude objasněna volba signálů, na základě kterých probíhá diagnostika. V následující kapitole bude popsána tvorba modelu vektorově řízeného PMSM, který je schopný simulovat ITF, v programu MATLAB Simulink a Simscape. V rámci této kapitoly bude také řeč o implementaci řídicího algoritmu do reálného pohonu a srovnání naměřených dat se simulací. Další z kapitol se bude věnovat předzpracování signálů pro hluboké učení. Toto bude zahrnovat filtraci signálů, řešení problému proměnného balíku dat, Vlnkovou transformaci a kompozici obrazu. Poslední z kapitol bude pojednávat o diagnostice zkratů pomocí převedené konvoluční neuronové sítě GoogLeNet. Tato kapitola bude zaměřena zejména na popis obrazového setu, který byl vytvořen pro učení sítě. Následovat bude popis modifikace architektury originální sítě a učení takto pozměněné sítě novým klasifikačním úlohám. Nakonec budou zhodnoceny dosažené výsledky v oblasti diagnostiky mezizávitových zkratů.

2.1 Úvod do diagnostiky mezizávitových zkratů

Mezizávitové zkraty se vyskytují v důsledku porušení izolace na statorovém vinutí. Po poruše ložiska se jedná o jednu z nejvíce se vyskytujících poruch na elektrických motorech vůbec. Včasná diagnostika těchto zkratů je klíčová zejména proto, že včas nepodchycený mezizávitový zkrat může vyústit až ve vznícení motoru.

U neřízeného PMSM se mezizávitový zkrat projeví hlavně vzrůstem proudu danou fází motoru. Dojde zejména ke vzrůstu třetí harmonické složky proudu. V klasických algoritmech pro detekci poruch se potom tento vzrůst zkoumá vhodně nastavenými komparátory. Z pohledu umělé inteligence lze místo komparátorů použít systém rozpoznávání vzorů tak, jak je popsáno v [15].

Při vektorovém řízení PMSM s ITF dochází rovněž k vzrůstu proudu danou fází motoru. Na rozdíl od neřízeného motoru se však tento vzrůst snaží kompenzovat regulátory proudů zkrášením akčního zásahu napětí. Při velice nízkých elektrických rychlostech motoru (přibližně do 100 rad/s) je kompenzace velice účinná a i pro velké závažnosti zkratu jsou proudy v rovnováze. Pokud se elektrická úhlová rychlost zvyšuje, regulátory „přestanou stíhat“ kompenzovat zkrat. V tomto případě dochází jak ke zkrášení akčního zásahu napětí, tak k rozvážení fázových proudů. Toto rozvážení je potom závislé nejenom na elektrické úhlové rychlosti, ale i na architektuře a parametrech regulátorů proudů a regulátoru otáček. Obecně nejpodstatnějším parametrem regulátorů proudů je zesílení. Při vyšších hodnotách zesílení je kompenzace zkratu „agresivnější“ a dochází k většímu zkrášení akčního

zásahu napětí. Při vyšších otáčkách má toto za následek znatelné nelineární zkreslení průběhů fázových proudů. Z pohledu diagnostiky lze potom zkrat dobře rozeznat. Problémem vysokého zesílení je však fakt, že regulační smyčka sklouzne k nestabilitě na znatelně nižších hodnotách hloubky zkratu než při nižším zesílení. Dokonce lze vhodnou volbou zesílení docílit stability i při zkratování celé fáze motoru. Více se o vlastním nastavení regulátorů zmiňuje kapitola 2.2.3. Další podstatný vliv na průběhy napětí a proudů mají poruchové signály regulační smyčky. Konkrétně jde o zpětně indukované napětí a zátěžný moment. Vliv zpětně indukovaného napětí je přímo závislý na elektrické úhlové rychlosti a hloubce zkratu, proto lze kvantifikovat na základě těchto veličin. Vliv zátěžného momentu lze stanovit z fázového posunu mezi napětím a proudem na dané fázi. Posledním podstatným vlivem je elektrický odpor zkratu R_f . Pokud je R_f mnohem větší než odpor statorové fáze R_s , nedojde ke vrůstu proudu na dané fázi. Pokud je R_f mnohem menší než odpor statorové fáze, veškerý proud teče zkratem a vzrůst proudu danou fází je přímo závislý na hloubce zkratu. Problém nastává, když se hodnoty odporu zkratu a odporu fáze blíží, neboť rozvážení proudů a napětí je nyní závislé i na poměru těchto odporů.

Obecně užívané přístupy diagnostiky zkratů na pohonu s PMSM nezahrnují komplexnost celého problému a snaží se závěr vyvodit pouze z průběhů fázových proudů, případně zpětně indukovaného napětí. Takto pracující algoritmy jsou však použitelné pouze pro konkrétní hodnotu zátěžného momentu a elektrické úhlové rychlosti. O aplikaci umělé inteligence v tomto odvětví více pojednává [16] a [17]. Aby bylo možné přesně určit hloubku zkratu u , musí být ostatní vlivy buď zahrnuty do diagnostiky nebo zanedbány. Uváží-li se neměnná architektura a parametry regulátorů, lze vliv řídicí struktury zanedbat. V této práci je rovněž zanedbán vliv odporu zkratu R_f , pro který platí $R_f \ll R_s$. Zkoumaným parametrem potom bude hloubka zkratu u . Tato je diagnostikována na základě signálů fázových proudů, PWM napětí a elektrické úhlové rychlosti. Průběhy PWM napětí vychází z akčních zásahů regulátorů proudů, a zkrat je na nich zřetelný i při nízkých elektrických rychlostech motoru. Průběhy fázových proudů jsou do diagnostiky zařazeny pro zpřesnění diagnóz a kvantifikování zátěžného momentu, který je dán fázovým posunem mezi napětím a proudem. Vliv elektrické úhlové rychlosti (a zpětně indukovaného napětí) je stanoven na základě přímého měření rychlosti a zahrnutí tohoto signálu do diagnostiky. Aby bylo možné získat z těchto signálů maximum užitečných informací a následně je předat konvoluční neuronové síti, musí být adekvátním způsobem předzpracovány. Tomuto tématu se věnuje kapitola 2.3.

2.2 Tvorba modelu

Pro vytvoření modelu bylo nejprve potřeba identifikovat parametry skutečného pohonu. V rámci této práce je uvažován třífázový PMSM, ale dostupný motor má šest statorových fází. Pro účely této práce byl jeden ze subsystémů motoru vyzkratován a s motorem se pracovalo jako s třífázovým. Stanovené parametry pohonu jsou shrnuty v tabulce 2.1.

Tab. 2.1: Parametry zkoumaného pohonu s PMSM

Parametr	Hodnota
Parametry motoru	
Počet pólových dvojic p	21
Odpor statorového vinutí R_s	0.606Ω
Indukčnost v příčné ose L_q	1.84 mH
Indukčnost v podélné ose L_d	1.99 mH
Magnetický tok od permanentních magnetů ψ_{pm}	9.89 mWb
Koeficient Gamma γ	0.316
Parametry rotoru a pracovního mechanismu	
Celkový moment setrvačnosti J	11.2 gm^2
Suché tření T_{stat}	0.226 Nm
Koeficient viskózního tření B	14.64 mNm s/rad
Parametry řídicího systému	
Napájení střídače V_{dc}	55 V
Maximální dovolený proud I_{max}	8 A
Vzorkovací perioda řídicího systému T_{vz}	100 μ s

Veškeré parametry motoru krom parametru γ byly převzaty z poskytnutých informací o motoru. Parametr γ byl stanoven na základě měření vlastních a vzájemných indukčností cívek na jednom vinutí. Následně byla vypočtená hodnota zpřesněna na základě srovnání simulace s měřením na různých hodnotách úhlové rychlosti motoru. Celkový moment setrvačnosti byl vypočten z momentu setrvačnosti motoru a pracovního mechanismu, který tvoří pružné spojky, snímač momentu a dynamometr. Koeficienty tření byly určeny na základě měření momentové zátěže pro různé rychlosti otáčení hřídele motoru pomocí dynamometru. Naměřené body byly proloženy přímkou pomocí metody nejmenších čtverců. Z parametrů této přímky byl následně stanoven koeficient viskózního tření B a suché tření T_{stat} .

2.2.1 Model motoru v prostředí Simscape Component

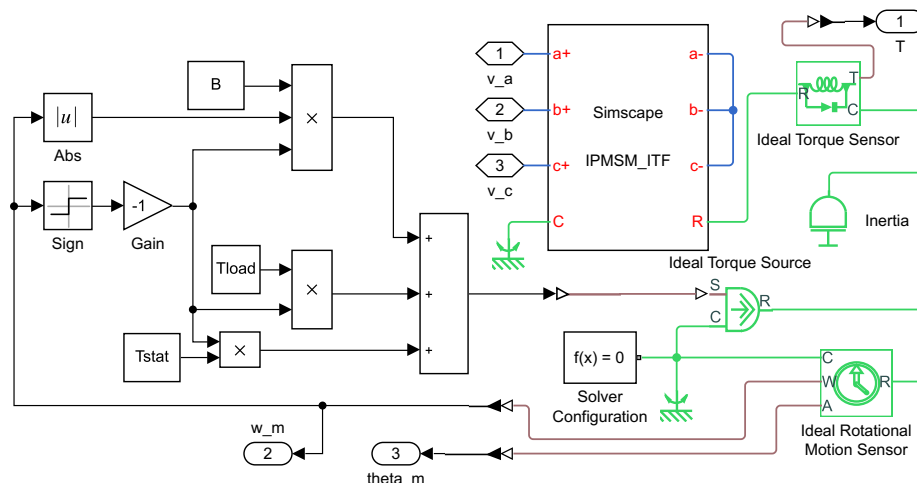
Simscape Component je nástavba programu MATLAB Simulink. Model je v tomto programu implementován na základě fyzikálního chování reálného prvku. Toto chování je obvykle popsáno soustavou matematických rovnic. Po vytvoření komponenty v tomto jazyce s ní lze v Simulinku pracovat jako s klasickým blokem.

Definice komponenty sestává z definic **uzlů**, **větví**, **parametrů**, **proměnných** a **rovníc**. **Uzly** reprezentují vstupy a výstupy komponenty a jsou přímo vázány na fyzikální signály, s kterými Simscape pracuje. Každé fázi motoru náleží dva elektrické uzly. Toto umožňuje přepínat mezi zapojením motoru do hvězdy a do trojúhelníka bez zásahu do vnitřního kódu komponenty. Dále model obsahuje dva uzly mechanické rotace (jeden pro nulovou úhlovou rychlost a druhý pro úhlovou rychlost motoru. Propojení uzlů realizují **větve**, kterým je přiřazena veličina, která jimi teče. Každá fáze motoru tedy představuje jednu z větví, kterou teče proud dané fáze. Uzly mechanické rotace jsou propojeny větví, kterou teče krouticí moment. Toto uspořádání je voleno zejména proto, aby bylo možné mechanický výstup motoru skloubit s mechanikou rotoru popsanou v kapitole 2.2.2. Fyzikální veličiny, které tečou větvemi musí být definovány v sekci **proměnné**. Proměnné jsou veličiny měnící se v čase, jejichž průběhy jsou hledány simulací. Jelikož se jedná o fyzikální modelování, musí mít každá proměnná přiřazenou platnou fyzikální jednotku. To samé platí i pro **parametry**. Parametry jsou v podstatě konstanty modelu, jejichž hodnota je předem známá. Pro zkoumaný pohon jsou parametry motoru uvedeny v tabulce 2.1 včetně fyzikálních jednotek. V sekci **rovnice** je potom definovaný matematický model PMSM s mezizávitovým zkratem popsaný v kapitole 1.1.3. V této sekci jsou definovány i rozdíly potenciálů mezi uzly (fázová napětí pro elektrické uzly a mechanická úhlová rychlost pro uzly mechanické rotace). Při implementaci rovnic motoru do Simscape Component musela být analyticky určena derivace matice indukčností podle elektrického úhlu natočení (pro výpočet generovaného krouticího momentu) a podle času (pro výpočet napěťových rovnic motoru). Část definice PMSM s ITF, sloužící zejména jako ukázka definice modelu v Simscape Component, lze vidět na výpisu kódu A.1.

Na vstupní (elektrické) uzly takto definovaného modelu jsou připojena PWM napětí. Tato představují výstupy tranzistorového střídače. Pro zrychlení simulace bylo zvoleno řešení, kdy SVPWM má na svém výstupu místo spínacích časů tranzistorů přímo efektivní hodnotu napětí, která má být v daném čase přivedena na motor. Tranzistorový střídač byl poté z modelu zcela vypuštěn. Výstupní uzly (uzly mechanické rotace) jsou použity dále v modelování mechaniky rotoru (viz kapitola 2.2.2). Z pohledu těchto uzlů se motor chová jako zdroj krouticího momentu, který je ovládaný elektrickými signály.

2.2.2 Model mechaniky rotoru v prostředí Simscape a Simulink

Modelování mechaniky rotoru a pracovního mechanismu vychází z poznatků shrnutých v kapitole 1.1.4. Konkrétněji se model snaží obsáhnout rovnicí 1.10. Pro modelování tření je v této rovnici použita funkce signum, což není velice vhodné. Tento přístup je v práci zvolen proto, že diagnostika probíhá na základě ustálených průběhů signálů. Není tedy potřeba dokonale obsáhnout dynamiku tření, která by zvýšila výpočetní nároky simulace. Modelování tření a zátěžného momentu je realizované v Simulinku. Převod signálů ze Simscape do Simulinku a naopak realizují speciální bloky pro převod fyzikálních signálů na signály, s kterými pracuje Simulink. Součtem třecích momentů a zátěžného momentu vzniká moment, který musí být v ustáleném stavu kompenzován motorem, aby se točil konstantní rychlostí. Tento je přiveden do Simscape části modelu pomocí ideálního zdroje momentu a působí proti generovanému momentu motoru. Obecně existuje analogie mezi elektrickými a mechanickými modely v Simscape. V mechanických obvodech teče krouticí moment větve obdobně jakou proud v elektrických. Rozdíl potenciálů v uzlech elektrického obvodu je analogií rozdílu úhlových rychlostí mezi body se společnou rychlostí mechanického obvodu. Této analogii potom odpovídá i princip zapojení snímačů. Snímače momentu se zapojují sériově do větve, kde má být moment měřen a snímače rychlosti se zapojují paralelně k této větvi. Do uzlu, ve kterém se střetává generovaný moment s momenty zátěže a tření je připojena setrvačná hmota. Výhoda Simscape je v tom, že po připojení setrvačné hmoty automaticky počítá i s dynamickým momentem. Hodnoty parametrů předaných do modelu jsou uvedeny v tabulce 2.1. V této práci je uvažován konstantní zátěžný moment T_{load} . Model mechaniky rotoru lze vidět na obrázku 2.1.



Obr. 2.1: Model mechaniky rotoru v programu MATLAB Simulink a Simscape

2.2.3 Návrh řízení v prostředí Simulink

Pro vektorové řízení PMSM bylo použito klasické momentové řízení s proudem i_d řízeným do nuly (viz obrázek 1.2). Část, která spadá do této kapitoly, začíná Clarkové transformací a končí SVPWM. Celá tato část pracuje diskrétně se vzorkovací periodou $T_{vz} = 100 \mu s$. Clarkové a Parkova transformace jsou do modelu řízení implementovány pomocí bloku „MATLAB Function“. Tento umožňuje popsat funkci mezi vstupy a výstupy bloku pomocí MATLAB kódu. Příklad implementace Parkovy transformace do bloku „MATLAB Function“ je vidět na výpisu kódu 2.1.

Výpis 2.1: Parkova transformace realizovaná MATLAB kódem

```
1 function [iq, id] = Park_tr(theta_el, ialpha, ibeta)
2 %#codegen
3 % This block simulates behavior of Park transformation.
4 id = ialpha*cos(theta_el) + ibeta*sin(theta_el);
5 iq = -ialpha*sin(theta_el) + ibeta*cos(theta_el);
```

Obdobným způsobem je realizována i SVPWM. Rovnice popisující chování SVPWM jsou převzaty z [8] a jsou implementovány do bloku „MATLAB Function“. O architektuře a návrhu regulátorů se více zmiňuje následující podkapitola.

Návrh regulátorů

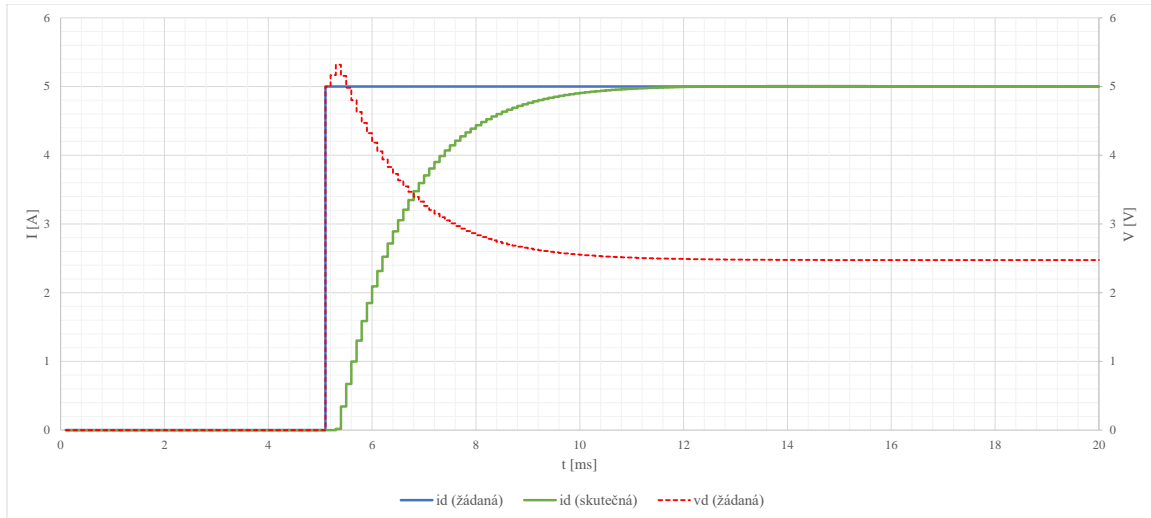
Pro řízení proudové i otáčkové smyčky byly použity PS regulátory. Chování těchto regulátorů popisuje diferenční rovnice 2.1.

$$u(k) = K \left(w(k) - y(k) + \frac{T_{vz}}{T_i} \sum_{i=0}^k (w(i) - y(i)) \right) \quad (2.1)$$

Kde $u(k)$ udává akční zásah v kroku k a $w(k) - y(k)$ je regulační odchylka v kroku k . Windup jev byl ošetřen anti windup metodou „clamping“. Akční zásahy regulátorů jsou omezeny dostupným napětím a proudem (viz tabulka 2.1). Výsledná architektura regulátorů lze vidět na obrázku A.1. Původně byl do řízení zařazen i blok pro zrušení vazeb, který měl linearizovat dq model motoru (viz kapitola 1.1.2) ve smyslu stav - vstup (viz obrázek A.2). Použitá architektura od vazbení však neuvážovala proměnlivost parametrů reálného motoru a při měření na skutečném pohonu tento blok nežádoucím způsobem zkresloval signály pro diagnostiku ITF. Proto byl odstraněn a následně odebrán i z modelu, aby se řízení shodovalo.

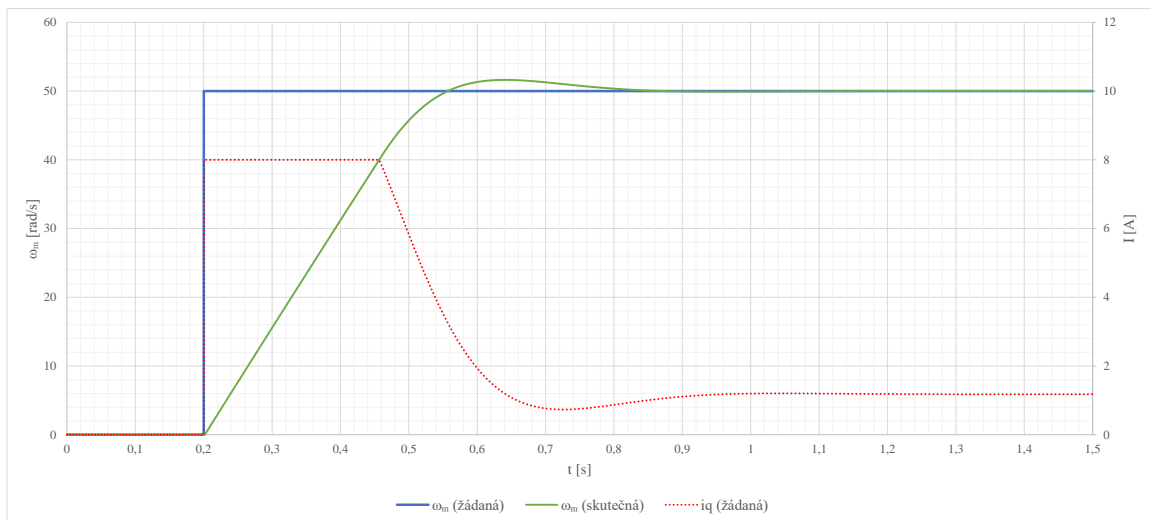
Nastavení regulátorů probíhalo na základě optimální odezvy systému na skok řízení s ohledem na akční zásah. V proudové smyčce byla zvolena pomalejší dynamika a přechodový děj bez překmitů. Toto má za následek stabilitu řídicí

smyčky i pro vysoké závažnosti zkratu, ale detekce ITF je obtížnější. Výsledná odezva proudové smyčky na skok řízení lze vidět na obrázku 2.2.



Obr. 2.2: Odezva proudové smyčky na skok řízení a akční zásah napětí

Na parametrech regulátoru otáčkové smyčky při ITF až tak nezáleží. Aby bylo možné rychle simulovat změny rychlosti, byla zvolena rychlejší dynamika otáčkové smyčky i za cenu překmitu. Odezva otáčkové smyčky na skok řízení lze vidět na obrázku 2.3.



Obr. 2.3: Odezva otáčkové smyčky na skok řízení a akční zásah proudu q složkou

Přechodové charakteristiky jsou vykresleny pro nulové poruchy. Respektive v otáčkové smyčce je nulový zátěžný moment T_{load} , ale jsou uvažovány poruchové signály vzniklé třením. Parametry regulátorů jsou shrnuty v tabulce 2.2.

Tab. 2.2: Navržené parametry regulátorů

Parametr	Hodnota
Zesílení otáčkové smyčky K_ω	0.8
Integrační časová konstanta otáčkové smyčky T_ω	0.1 s
Zesílení proudové smyčky $K_d = K_q$	1
Integrační časová konstanta proudové smyčky $T_d = T_q$	0.003 s

2.2.4 Srovnání simulací s výsledky měření na reálném pohonu

Řízení, popsané v kapitole 2.2.3, bylo implementováno i do reálného pohonu s PMSM. Data byla na reálném pohonu měřena pro hodnoty hloubek zkratu $u = 0/25, 3/25, 6/25, 14/25, 25/25$. Hodnota 25/25 odpovídá třetině fázového vinutí, které je složeno ze tří sériově řazených cívek. Každá z těchto cívek má celkem 25 závitů. Chování pohonu při zkratu bylo měřeno pro zátěžné momenty T_{load} 0 a 0.5 Nm a elektrické úhlové rychlostí ω_e 0 až 2000 rad/s s krokem 200 rad/s. Při hodnotách ω_e vyšších než 1600 rad/s přestávalo být navržené řízení stabilní, proto byla hodnota 1600 rad/s stanovena jako hranice, nad kterou se již chování motoru nebude zkoumat. Hranice udávající maximální zátěžný moment byla později stanovena na základě simulace jako 1 Nm.

Shoda simulace s naměřenými daty je zkoumána na základě rozvážení fázových proudů při dané elektrické úhlové rychlosti a zátěžném momentu. Pro popis shody simulace s naměřenými daty slouží chyba rozvážení δ . Výpočet chyby rozvážení lze vidět v rovnici 2.2.

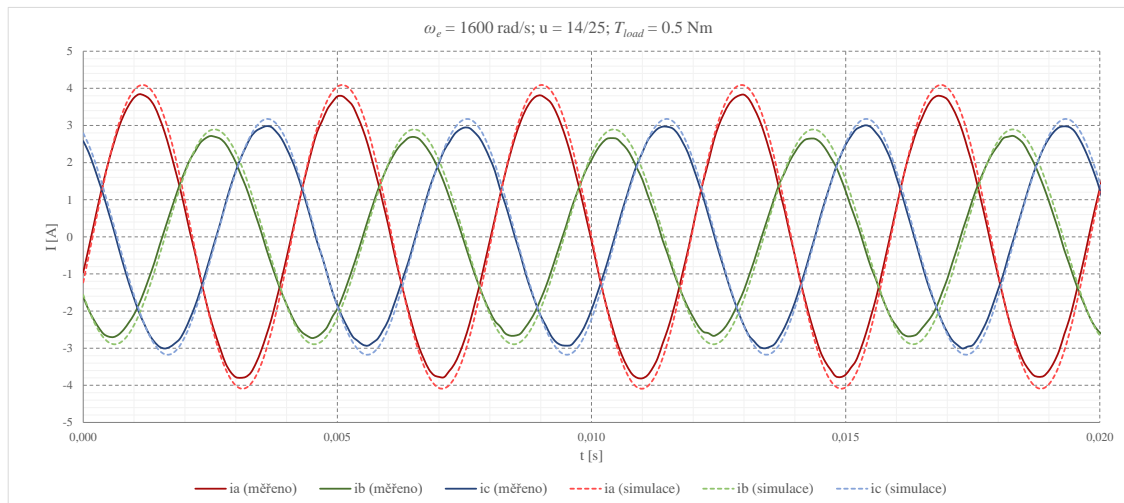
$$\delta = \left[\text{abs} \left(\frac{\frac{I_b(\text{měřeno})}{I_a(\text{měřeno})} - \frac{I_b(\text{simulace})}{I_a(\text{simulace})}}{\frac{I_b(\text{měřeno})}{I_a(\text{měřeno})}} \right) + \text{abs} \left(\frac{\frac{I_c(\text{měřeno})}{I_a(\text{měřeno})} - \frac{I_c(\text{simulace})}{I_a(\text{simulace})}}{\frac{I_c(\text{měřeno})}{I_a(\text{měřeno})}} \right) \right] 100 \quad (2.2)$$

Hodnoty I_a , I_b a I_c označují amplitudy harmonických signálů fázových proudů v ustáleném stavu. Vztažením výpočtů dílčích chyb k amplitudě proudu fází 'a' je zajištěna určitá netečnost vůči hodnotám amplitud jako takovým. Počítat chybu pouze na základě hodnot amplitud měřených proudů a simulace není vhodné. Na amplitudy měřených proudů má například vliv účinnost motoru, která není do simulace zahrnuta. Velká část takto vypočtené chyby by poté závisela na rozdílu amplitud způsobeném účinností motoru a ostatními vlivy. Chyby rozvážení stanovené pro nulový zátěžný moment lze vidět v tabulce 2.3.

Tab. 2.3: Rozdíl mezi modelem a měřením vyjádřený pomocí chyby pro $T_{load} = 0 \text{ Nm}$

Hloubka zkratu u :	3/25	14/25	25/25
Elektrická úhlová rychlost ω_e [rad/s]	δ [%]	δ [%]	δ [%]
1600	6.82	6.97	9.81
1400	7.37	7.25	8.05
1200	4.72	9.98	8.91
1000	7.36	8.53	8.89
800	7.28	8.62	8.68
600	3.76	7.11	9.20
400	6.48	9.08	9.35
200	6.01	8.68	8.35

Z tabulky 2.3 je zřejmé, že pro nižší hloubky zkratu je chyba menší. Toto je způsobeno pravděpodobně nesplněním podmínky $R_f \ll R_s$. Velikost odporu zkratu ovlivňuje rozvážení zejména na vyšších hodnotách u . Dalším z faktorů, které navyšují chybu rozvážení jsou třecí momenty, které jsou aproximovány lineární funkcí a modelovány pouze za pomoci funkce signum (viz kapitola 2.2.2). Snížit chybu způsobenou třecími momenty lze zařazením „velkého“ zátěžného momentu, který bude mít dominantní vliv. Toto může snížit chybu až o 2 %. Graficky lze srovnání měřených dat se simulací vidět na obrázku 2.4. Obecně potom platí, že se chyba rozvážení pro použitý model pohybuje mezi 5 až 10 %.



Obr. 2.4: Srovnání výsledku simulace fázových proudů při ITF s naměřenými daty

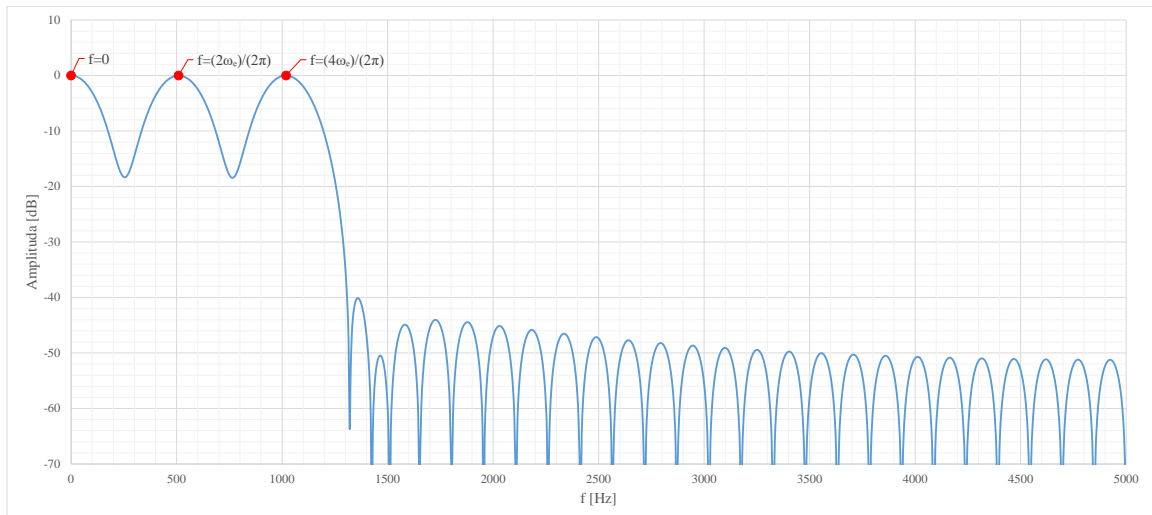
2.3 Předzpracování signálů

V této kapitole bude rozebrán postup, jakým jsou vytvořeny obrazy na základě měření fázových proudů, PWM napětí a úhlové rychlosti. Celý tento proces lze rozdělit do dílčích částí, o kterých pojednávají následující podkapitoly.

2.3.1 Filtrace a převzorkování signálů

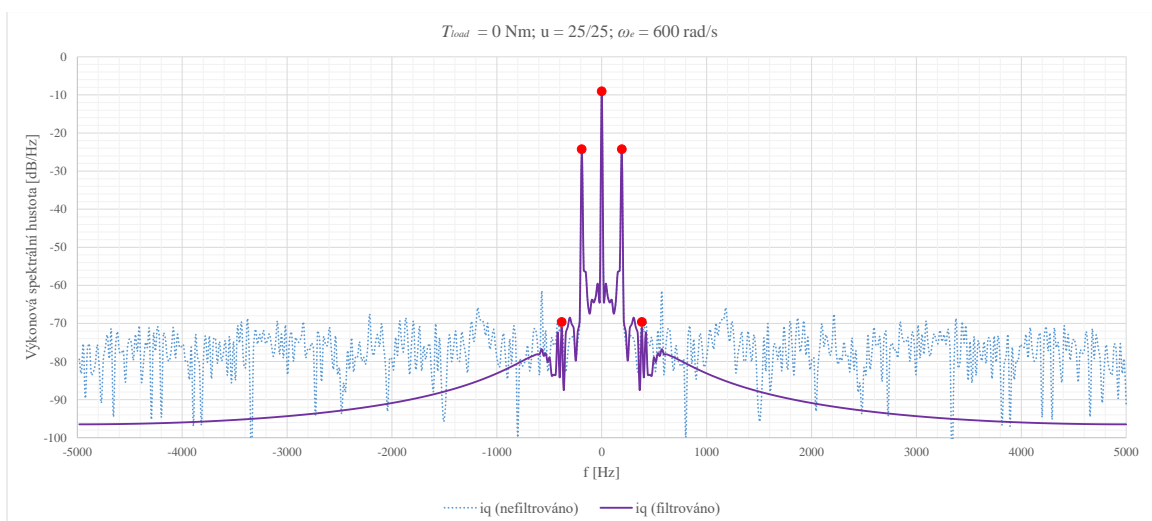
Celý proces předzpracování signálů začíná sběrem dat pro pět elektrických otáček motoru. Počet otáček, pro který jsou data sbírána, byl stanoven s ohledem na náhodné jevy a fakt, že část dat bude v rámci předzpracování odstraněna, aby se minimalizovalo zkreslení při aplikaci Vlnkové transformace. Počátek a konec sběru dat určuje elektrický úhel natočení $\theta_e = \frac{\pi}{2}$. Na vyšších rychlostech není možné, aby se vzorkovač trefil přesně do tohoto úhlu. Jako hraniční hodnoty se poté použijí hodnoty θ_e , které jsou úhlu $\frac{\pi}{2}$ nejbližší. Toto vede k mírným fázovým posunům signálů, proti kterým jsou však konvoluční neuronové sítě odolné. Elektrický úhel natočení se stanoví na základě mechanického pomocí rovnice 1.9.

Jedním z problémů takto vzniklého datového balíku je náhodný šum, který zkresluje průběhy fázových proudů a PWM napětí obdobně jako zkrat. Pokud by signály nebyly filtrovány, bylo by velice obtížné (ne-li nemožné) korektně stanovit hloubku zkratu, protože by rozvážení signálů krom zkratu ovlivňoval i šum. Výhodná se ukazuje být filtrace signálů v dq souřadnicích. V dq lze totiž snadno určit, které frekvenční složky signálů vypovídají o hloubce zkratu. V ustáleném stavu se jedná o frekvence náležící dvojnásobku a čtyřnásobku elektrické úhlové rychlosti ω_e . Fázové proudy a PWM napětí jsou do dq převedeny pomocí rovnic popsaných v kapitole 1.2.1. Za účelem filtrace signálů byl vytvořen filtr typu konečná impulsní odezva, jehož koeficienty se mění v závislosti na elektrické úhlové rychlosti. Úkolem filtru je propustit stejnosměrnou složku a složky odpovídající $2\omega_e$ a $4\omega_e$. Limitujícím parametrem je poté řád filtru. Řád filtru se volí s ohledem na počet vzorků signálu a platí, že počet vzorků musí být alespoň 3x větší než je řád filtru. Při limitní rychlosti $\omega_e = 1600$ rad/s náleží pěti elektrickým otáčkám přibližně 197 vzorků. Řád filtru je poté roven 65. Filtr tohoto řádu zvládne dobře potlačit vyšší frekvence, nicméně problémem je vyfiltrovat pásma ležící mezi zájmovými frekvenčními složkami. Tento problém se ukazuje být limitujícím faktorem pro přesné určení hloubky zkratu. Frekvenční odezva navrženého filtru s vyznačenými zájmovými složkami lze vidět na obrázku 2.5.



Obr. 2.5: Frekvenční odezva použitého FIR filtru

Určení koeficientů filtru a filtrace je implementována jako MATLAB funkce (viz výpis kódu B.1). Pro filtrace je použita funkce *filtfilt*. Tato provádí digitální filtrace s nulovým fázovým posunem. Toho je dosaženo filtrováním vstupních dat jak ve směru dopředném (od nejmenšího vzorku k největšímu) tak i ve směru zpětném (od největšího vzorku k nejmenšímu). Ukázkou filtrace lze i s vyznačenými zájmovými frekvencemi vidět na obrázku 2.6.



Obr. 2.6: Ukázkou filtrace q složky proudu

Jak je z obrázku 2.6 zřejmé, potlačení pásem mezi zájmovými frekvencemi je značně neuspokojivé. Toto vede k menší přesnosti diagnóz.

Dalším problémem, který je potřeba řešit, je proměnný balík dat. Jak již bylo zmíněno dříve, balík dat zapouzdřuje pět elektrických otáček motoru. Toto však znamená, že při různých elektrických rychlostech se počet vzorků v balíku různí (s uvažováním konstantní periody vzorkování $T_{vz} = 100 \mu s$). Například při $\omega_e = 200 \text{ rad/s}$ balík obsahuje 1571 vzorků a při $\omega_e = 1600 \text{ rad/s}$ obsahuje pouze 197 vzorků. Počet vzorků má poté zásadní vliv na Vlnkovou transformaci. Velký počet vzorků vede k vysoké přesnosti časově-frekvenční reprezentace signálu, ale doba výpočtu Vlnkové transformace je nepřiměřeně dlouhá. Malý počet vzorků naproti tomu vede k rychlému výpočtu, ale přesnost je mizivá. Proměnlivost balíku dat rovněž zbytečně mate síť, která se více než na detaily obrazu soustředí na zkreslení způsobené rozdílným počtem vzorků při různých úhlových rychlostech. Řešením tohoto problému je stanovení počtu vzorků, na který budou signály převzorkovány. Bylo zjištěno, že optimální počet vzorků je 1000. Při tomto počtu je vysoká přesnost časově-frekvenční reprezentace signálu a relativně krátká doba výpočtu.

Při převzorkování signálu elektrické úhlové rychlosti je nejprve vypočtena střední hodnota ω_e za 5 elektrických otáček. Následně je vytvořen vektor o 1000 prvcích obsahujících vypočtenou střední hodnotu rychlosti. Jelikož diagnostika probíhá v ustáleném stavu (při konstantní rychlosti) je tímto postupem dosaženo jak převzorkování tak filtrace signálu ω_e . Pro převzorkování PWM napětí a fázových proudů se ukazuje být výhodné vycházet z $\alpha\beta$ reprezentace filtrovaných signálů. V $\alpha\beta$ lze převzorkovat pouze 4 signály (2 napětí a 2 proudy). Pokud by se převzorkování provádělo v dq musel by být navíc převzorkován i signál θ_e (celkem tedy 5 signálů) a pokud by se provádělo v abc počet signálů pro převzorkování by vzrostl až na 6. Převzorkování je potom provedeno pomocí MATLAB funkce *resample*. Pokud je počet vzorků signálu větší než je žádaný počet, jsou přebývající vzorky odstraněny. Naproti tomu pokud je počet vzorků signálu menší než je žádaný počet, jsou chybějící vzorky vytvořeny pomocí interpolace. Jelikož se po aplikování filtru a převodu do $\alpha\beta$ souřadného systému jedná o harmonické signály, dosahuje tato interpolace poměrně vysoké přesnosti.

Po převzorkování jsou signály z $\alpha\beta$ převedeny zpět do abc souřadnic. Při tomto převodu se již znovu nepočítá PWM napětí pomocí SVPWM algoritmu, ale je provedena pouze inverzní Clarkové transformace obdobně jako u proudových signálů. Tímto postupem jsou tedy získány filtrované signály pro 5 elektrických otáček motoru, které mají vždy 1000 vzorků bez ohledu na elektrickou úhlovou rychlost motoru. Část kódu realizující filtraci a převzorkování proudových signálů lze vidět ve výpisu B.2. Ve výpisu B.2 je již uvažováno, že se jedná o balík dat odpovídající 5 elektrickým otáčkám.

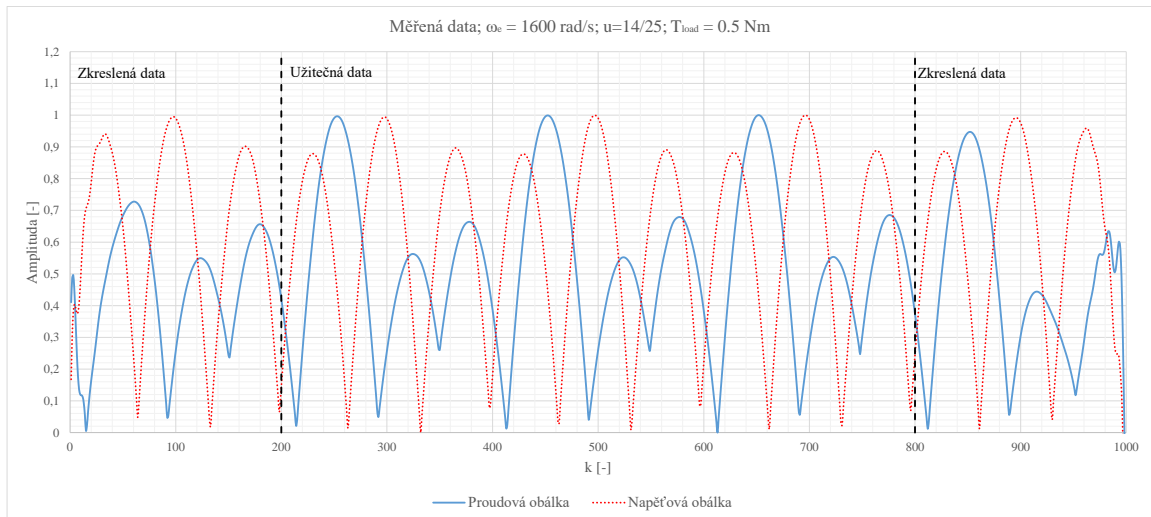
2.3.2 Signálové obálky a Vlnková transformace

Po filtraci a převzorkování je k dispozici celkem 7 signálů. Jedná se o 3 fázové proudy, 3 PWM napětí a 1 signál elektrické úhlové rychlosti. Aby bylo možné provést kompozici obrazu na základě 3 složek (R, G a B), musí být zredukován počet signálů na 3. Tyto jednorozměrné signály musí být následně převedeny do dvourozměrné reprezentace, která bude použita jako jedna ze složek obrazu.

Pro redukci počtu signálů se ukazuje být výhodné zakódovat signály PWM napětí a fázových proudů do takzvaných obálek. Tyto obálky jsou spočteny jako maximum napětí/proudů v daném vzorku a mají dostatečnou vypovídající schopnost o rozvážení napětí/proudů. Výpočet proudové obálky na základě fázových proudů prezentuje rovnice 2.3.

$$i_{ob}(k) = \max(i_a(k), i_b(k), i_c(k)) \quad (2.3)$$

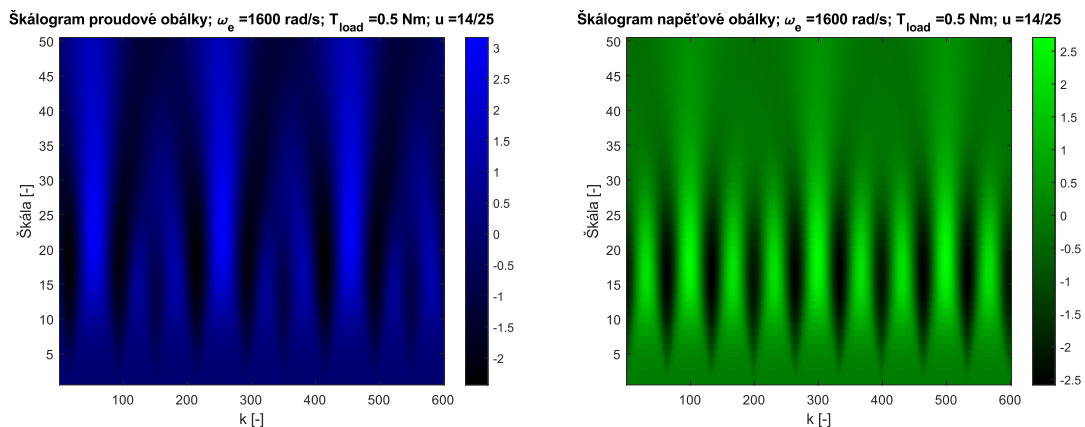
Obdobný výpočet je použit i pro stanovení napěťové obálky na základě PWM napětí. Aby se další část předzpracování učinila nezávislá na konkrétních hodnotách napětí/proudů, jsou obálky normalizovány mezi hodnotu 0 (odpovídající minimu) a 1 (odpovídající maximu). Normalizace sice posune hodnoty obálek do definovaného rozsahu, ale zachová informaci o rozvážení, která je z pohledu diagnostiky klíčová. Ukázku proudové a napěťové obálky lze vidět na obrázku 2.7.



Obr. 2.7: Ukázka proudové a napěťové obálky

Z obrázku 2.7 je zřejmé, že první a poslední perioda signálů obálek je zkreslena. Toto zkreslení vzniká vlivem filtrace a převzorkování a data z této oblasti již nelze dále považovat za validní. Rovněž lze v obrázku 2.7 vidět zřetelný fázový posun mezi napětím a proudem závislý na zátěžném momentu.

Pro převod jednorozměrných obálek na dvourozměrné signály je použita spojitá Vlnková transformace s mateřskou vlnkou „Mexican hat“ (viz kapitola 1.3). Otcovská vlnka (škálovací funkce) je nahrazena vektorem škál od 1 do 50 s krokem 1. Při aplikaci diskretizované verze spojité Vlnkové transformace na diskrétní signál dochází ke zkreslení koeficientů odpovídajících počátečním a konečným vzorkům signálu. Toto zkreslení vzniká v důsledku definovaného konce a počátku signálu. Dceřinným vlnkám v těchto oblastech chybí informace o vzorcích, které se nachází před počátkem a nebo za koncem signálu. Aby byl zachován co největší balík užitečných dat, je Vlnková transformace aplikována na celou obálku (i na část se zkreslenými daty). Toto zapříčiní, že se zkreslení koeficientů (vlivem Vlnkové transformace) projeví na vzorcích, které odpovídají zkresleným datům (vlivem převzorkování a filtrace). Následně jsou koeficienty odpovídající prvním a posledním 200 vzorkům odstraněny. Takto vzniká matice koeficientů s 600 sloupci (validní vzorky) a 50 řádky (škály), která obsahuje maximum užitečné informace a žádné nežádoucí zkreslení. Vykreslením této matice do 2-D prostoru vzniká takzvaný škálogram. Ukázkou škálogramu proudové a napěťové obálky lze vidět na následujícím obrázku.



Obr. 2.8: Ukázka škálogramu proudové a napěťové obálky

V obrázku 2.8 je volena barevná skladba škálogramů korespondující s barevnými složkami výsledného obrazu.

Signál elektrické úhlové rychlosti je do 2-D podoby převeden na základě již dříve stanovené střední hodnoty rychlosti. Je vytvořena matice s koeficienty rovnými střední hodnotě elektrické úhlové rychlosti a rozměry odpovídajícími rozměrům škálogramů. Na základě těchto tří 2-D signálů (škálogram proudové obálky, škálogram napěťové obálky a matice rychlostí) jsou skládány obrazy použité pro diagnostiku. O tomto více pojednává kapitola 2.3.3.

2.3.3 Kompozice obrazu

Výsledný obraz je skládán z R (red), G (green) a B (blue) složky. R složka je spočtena na základě matice rychlostí. Koefficienty této matice (odpovídající střední hodnotě elektrické úhlové rychlosti) jsou vyděleny deseti a zaokrouhleny na celé číslo. Na zkoumaném rozsahu 0 až 1600 rad/s toto odpovídá hodnotám červené 0 až 160. G složka je vypočtena ze škálogramu napěťové obálky. Matice koeficientů je normalizována mezi hodnoty 0 až 255 a jednotlivé koeficienty jsou zaokrouhleny na celá čísla. Obdobným způsobem je vypočtena B složka na základě škálogramu proudové obálky. Jednotlivé složky jsou sloučeny do jednoho obrazu, který nabývá rozměrů 50x600x3. Aby mohl být obraz použit při práci s konvoluční neuronovou sítí GoogLeNet, musí být rozměrově přizpůsoben vstupu sítě (224x224x3). K tomuto účelu je použita funkce *imresize*, která změní rozměr obrazu na požadovanou hodnotu. Tato funkce v základu pro změnu rozměrů používá bikubickou interpolaci. Obrazy převzorkované touto interpolací jsou vyhlazenější a obsahují menší počet interpolačních artefaktů. Část kódu realizující kompozici obrazu lze vidět ve výpisu 2.2.

Výpis 2.2: Kompozice obrazu na základě škálogramů obálek a matice rychlosti

```
1 %Red part
2 R = round(abs(w_e_mat/10));
3 R = uint8(R);
4 % Blue part
5 B = round(255.*(CWT_I_mat-min(min(CWT_I_mat)))/...
6 (max(max(CWT_I_mat))-min(min(CWT_I_mat))));
7 B = uint8(B);
8 % Green part
9 G = round(255.*(CWT_V_mat-min(min(CWT_V_mat)))/...
10 (max(max(CWT_V_mat))-min(min(CWT_U_mat))));
11 G = uint8(G);
12 % Composing image
13 IMG = cat(3, R, G, B);
14 IMG = imresize(IMG, [224 224]);
```

Obrazy pro vybrané hodnoty hloubky zkratu, momentu zátěže a elektrické úhlové rychlosti lze vidět na obrázku B.1.

2.4 Diagnostika zkratů pomocí GoogLeNet

Tato kapitola bude zaměřena na použití konvoluční neuronové sítě GoogLeNet pro diagnostiku mezizávitových zkratů. Bude probrán obrazový set, který byl vytvořen pro učení sítě. Následovat bude podkapitola o modifikaci originální GoogLeNet sítě a učení této modifikované sítě rozpoznávat hloubku zkratu. Na závěr budou shrnuty výsledky diagnostiky pro data ze simulace a data měřená na skutečném pohonu.

2.4.1 Obrazový set

Jak již bylo zmíněno dříve, přesnost s jakou se je konvoluční neuronová síť schopna naučit nové klasifikační úloze, je přímo závislá na obrazovém setu, na který je síť učena. Velký obrazový set vede sice k větší přesnosti diagnóz, ale doba učení sítě se může vyšplhat až na několik dní. Z tohoto důvodu je třeba vytvořit obrazový set, který dostatečně obsáhne celé zkoumané pásmo stavů a zároveň nenaroste do příliš velkých rozměrů.

Klíčovým parametrem zkoumaným neuronovou sítí je hloubka zkratu u . Tato je zkoumána na rozsahu 0 až 25 zkratovaných závitů na jedné cívce, která má celkem 25 závitů. Aby se diagnostika učinila nezávislá na elektrické úhlové rychlosti a momentu zátěže, musí být změna těchto veličin rovněž zahrnuta do obrazového setu. Zkoumané pásmo ω_e se pohybuje od 50 do 1600 rad/s a pásmo zkoumaných zátěžných momentů nabývá rozsahu 0 až 1 Nm. Data jsou potom simulována pro hloubky zkratu s krokem 1 zkratovaný závit, elektrické úhlové rychlosti s krokem 50 rad/s a zátěžné momenty s krokem 0.1 Nm. Toto ve finále vede k 9152 simulacím a jim příslušícím obrazům, které vznikly předzpracováním signálů popsáném v kapitole 2.3. Každý tento obraz má vypovídající schopnost o hloubce zkratu, momentu zátěže a elektrické úhlové rychlosti. Jak již bylo zmíněno dříve, uvažuje se neměnná architektura a parametry regulátorů a odpor zkratu mnohem menší než je odpor fázového vinutí. Zvolený krok pro T_{load} a ω_e se ukázal být dostačující pro to, aby síť bez problému rozpoznala hloubku zkratu i v hodnotách T_{load} a ω_e , které leží v pásmech mezi danými kroky (např. stav $u=10$; $\omega_e = 125$ rad/s; $T_{load} = 0.250$ Nm).

Problém s takto vytvořeným obrazovým setem nastal až při učení neuronové sítě (viz kapitola 2.4.3). Ukázalo se, že pokud je daný stav reprezentován pouze jedním obrazem, tak se síť nezvládne naučit na lepší přesnost než je 80 %. Tento problém byl vyřešen vytvořením pěti identických obrazů pro každý stav namísto jednoho. Toto zvedlo přesnost sítě až na hodnoty blízké 100 % (pro simulovaná data). Výsledný obrazový set tedy čítá 45 760 obrazů, které jsou rozděleny do kategorií podle hloubky zkratu. V rámci jedné kategorie se tedy obrazy liší na základě ω_e a T_{load} .

2.4.2 Modifikace originální sítě

Konvoluční neuronová síť GoogLeNet je implementovaná jako addon do MATLAB Deep Learning toolboxu, proto lze pro práci s touto sítí s výhodou využít prostředí MATLAB. V rámci této práce zůstala originální architektura CNN GoogLeNet (viz kapitola 1.4.2) téměř beze změny. Pouze poslední vrstvy této sítě jsou zaměněny za nové, které svým rozměrem odpovídají počtu kategorií, na které je modifikovaná síť učena. Konkrétně se jedná o plně propojenou vrstvu a klasifikační vrstvu. Počet neuronů v plně propojené vrstvě musí korespondovat s počtem kategorií a počet vstupů klasifikační vrstvy musí korespondovat s počtem výstupů plně propojené vrstvy (tedy počtem neuronů v této vrstvě). Klasifikační vrstva potom realizuje funkci „Softmax“ a na základě výstupů této funkce rozhodne o třídě, do které spadá vstupní obraz. Výstupem klasifikační vrstvy je potom třída s nejvyšší pravděpodobností a vektor příznaků udávající pravděpodobnost, že vstupní obraz spadá do dané třídy. Další upravenou vrstvou je dropout vrstva. Původní dropout vrstva ignoruje 50 % výstupů konvolučních vrstev. Tato vrstva byla modifikována tak, aby ignorovala 60 % těchto výstupů. Ignorovat více výstupů se ukázalo být důležité, protože obrazový set obsahuje vždy 5 totožných obrazů pro jeden stav. Síť je tedy náchylnější k přeučení. Část kódu realizující modifikaci originální GoogLeNet sítě lze vidět ve výpisu 2.3.

Výpis 2.3: Modifikace originální GoogLeNet sítě v programu MATLAB

```
1 % Uložení originální sítě do workspace jako net
2 net = googlenet;
3 lgraph = layerGraph(net); %vytvoření grafu vrstev
4 % Nová dropout vrstva
5 Dropout = dropoutLayer(0.6, 'Name', 'new_Dropout');
6 lgraph = replaceLayer(lgraph, 'pool5-drop_7x7_s1', Dropout);
7 % Nová plně propojená vrstva vytvořená na základě
8 % počtu kategorií obrazového setu pro učení sítě
9 C_num = numel(categories(imgsTrain.Labels));
10 New_L = fullyConnectedLayer(C_num, 'Name', 'new_fc', ...
11     'WeightLearnRateFactor', 5, 'BiasLearnRateFactor', 5);
12 lgraph = replaceLayer(lgraph, 'loss3-classifier', New_L);
13 % Nová klasifikační vrstva
14 New_C = classificationLayer('Name', 'new_classoutput');
15 lgraph = replaceLayer(lgraph, 'output', New_C);
```

2.4.3 Učení modifikované sítě

Pro učení modifikované CNN je použita optimalizační metoda SGDM (stochastic gradient descent with momentum). Tato metoda si pamatuje změnu vah při každé iteraci a novou změnu vah vypočítá jako lineární kombinaci gradientu a předchozí změny. Dalším podstatným parametrem učení je „Mini batch size“. Tento parametr určuje kolik obrazů je síti předloženo v rámci jedné iterace a má podstatný vliv na učení sítě. V rámci této práce je výsledná síť učena na několik etap a „Mini batch size“ se v každé etapě různí. Koeficient učení je volen jako 10^{-4} . Ukázka definice parametrů učení a učení sítě v programu MATLAB lze vidět ve výpisu 2.4.

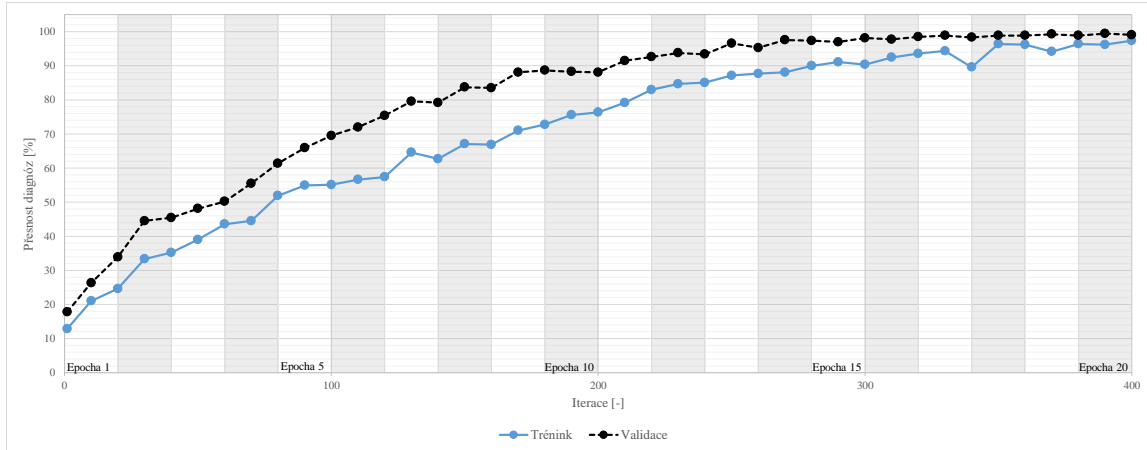
Výpis 2.4: Definice parametrů učení a učení sítě v programu MATLAB

```
1 options = trainingOptions('sgdm',...
2     'MiniBatchSize',572,...
3     'MaxEpochs',10,...
4     'InitialLearnRate',1e-4,...
5     'ValidationData',imgsValidation,...
6     'ValidationFrequency',10,...
7     'Verbose',1,...
8     'ExecutionEnvironment','cpu',...
9     'Plots','training-progress');
10 rng default
11 trainedGN = trainNetwork(allImages,lgraph,options);
```

Při pokusu učit modifikovanou síť rozpoznávat zkrat s krokem 1 zkratovaný závit se síť učila nepřiměřeně dlouhou dobu. Řádově se přesnost diagnóz zlepšovala o 5 % za 4 hodiny. Z tohoto důvodu byly sníženy nároky na diagnostiku zkratů na 3 zkratované závity v jedné kategorii. Kategorie poté nabyly tvaru 0/25, 1-3/25, 4-6/25, 7-9/25, 10-12/25, 13-15/25, 16-18/25, 19-21/25 a 22-25/25. Síť obsahující na výstupu tyto kategorie bude dále označována jako síť pro detekci hloubky zkratu. Zmíněná kategorizace sítě umožňuje rozeznat zkrat s krokem 4 % celého vinutí.

Aby bylo učení ještě více urychleno, byla síť pro detekci hloubky zkratu učena ve dvou etapách. V první etapě je originální GoogLeNet síť modifikována tak, aby plně propojená vrstva obsahovala 6 neuronů a výstupní kategorie byly dány diskretním krokem 5 zkratovaných závitů (kategorie: 0/25, 5/25, 10/25, 15/25, 20/25, 25/25). Učení sítě s tímto diskretním krokem zapříčiní to, že jsou nejprve hrubým způsobem nastaveny váhové koeficienty v konvolučních vrstvách. Toto zajistí, že síť zvládne rozpoznat základní charakteristiky obrazů, které vznikly předzpracováním signálů. Výhodné je učit síť na etapy zejména proto, že v první etapě je obrazový set daleko

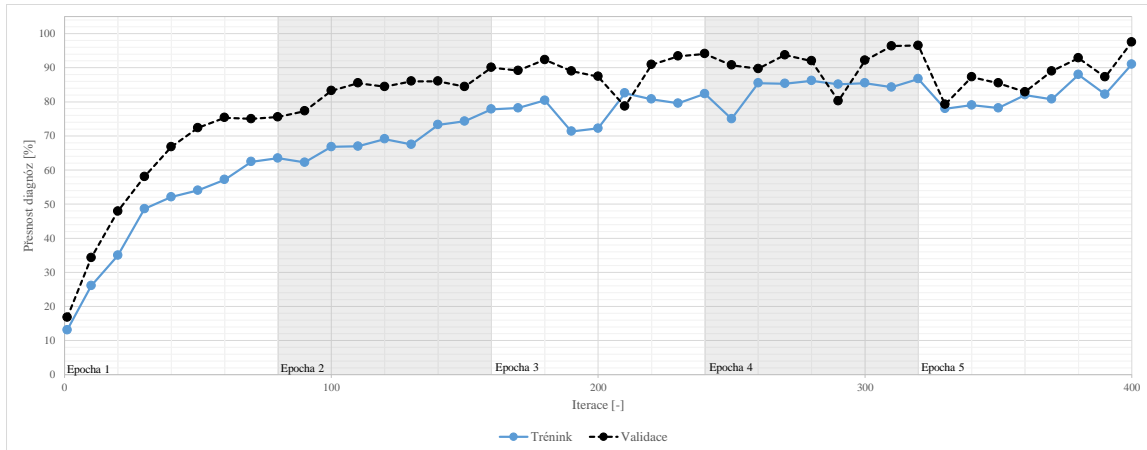
menší (pouze 10 560 obrazů). Velikost „Mini batch size“ je v první etapě volena jako 528 obrazů. Jedna epocha (předložení celého obrazového setu) potom odpovídá 20 iteracím (předložení obsahu „Mini batch“). Průběh učení sítě na diskretní krok 5 zkratovaných závitů lze vidět na obrázku 2.9.



Obr. 2.9: Průběh učení sítě na diskretní krok 5 zkratovaných závitů

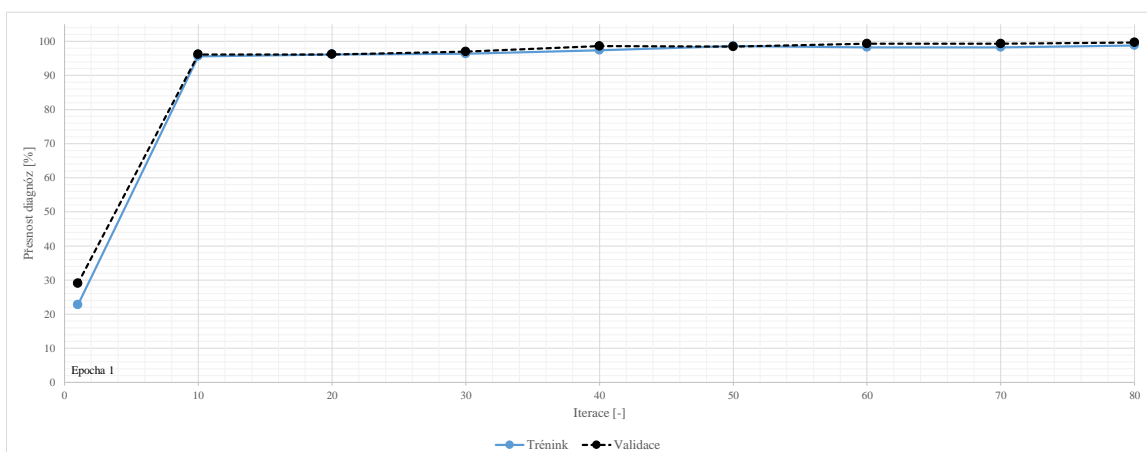
Pro validaci sítě jsou použity simulované stavy, na které síť není učena (např. stav $u = 5/25$; $\omega_e = 125 \text{ rad/s}$ a $T_{load} = 0.325 \text{ Nm}$). Další rozdíl mezi validací a tréninkem je v tom, že při validaci není aktivní dropout vrstva. To znamená, že jsou aktivní všechny výstupy konvolučních vrstev. Naproti tomu při tréninku je aktivních pouze náhodně zvolených 40 % těchto výstupů. Učení sítě trvalo 8 hodin a 21 minut. Výsledná přesnost se pohybuje okolo 99 %.

Síť vytvořená v první etapě je použita jako základ sítě v druhé etapě (namísto originální sítě GoogLeNet). Plně propojená vrstva je modifikována, aby obsahovala 9 neuronů a výstupní kategorie nabývají tvaru, jaký byl zvolen na začátku této podkapitoly (kategorie: 0/25, 1-3/25, ...). V tomto případě je již pro učení sítě použit celý obrazový set (45 760 obrazů). Velikost „Mini batch size“ je volena jako 572 obrazů a jedna epocha odpovídá 80 iteracím. Síť se byla schopna naučit na přesnost lepší než 97 % za 5 epoch. Doba učení byla 9 hodin a 8 minut. Pro validaci jsou, obdobně jako v předchozím případě, použity simulované stavy, na které síť není učena. Průběh učení sítě pro detekci hloubky zkratu lze vidět na obrázku 2.10.



Obr. 2.10: Průběh učení sítě pro detekci hloubky zkratu

Pro posouzení přítomnosti zkratu na motoru byla vytvořena speciální síť, která na svém výstupu obsahuje pouze dvě kategorie: *Healthy* a *ITF*. Tato síť bude dále označována jako síť pro detekci přítomnosti zkratu. Základem této sítě je síť pro detekci hloubky zkratu. Vycházet ze sítě pro detekci hloubky zkratu se ukazuje být klíčové. Při použití originální CNN GoogLeNet pro základ sítě, bylo problematické určit „zdravé“ chování motoru. Toto bylo způsobeno tím, že síť nezvládla odlišit náhodný šum, který se nepodařilo potlačit filtrem, od vlivu nízkých hodnot hloubky zkratu. Pokud se pro základ použije síť pro detekci hloubky zkratu, kde jsou konvoluční vrstvy nastaveny na velice jemné detaily, zvládne nově vzniklá síť odlišit šum od zkratu častěji. Průběh učení sítě pro detekci přítomnosti zkratu lze vidět na obrázku 2.11. Nastavení parametrů učení je stejné jako při učení sítě pro detekci hloubky zkratu.



Obr. 2.11: Průběh učení sítě pro detekci přítomnosti zkratu

2.4.4 Výsledky diagnostiky

Jak již bylo zmíněno v předchozí části, síť pro detekci hloubky zkratu dosahuje při diagnostice zkratů na modelu velmi kvalitních výsledků. Diagnostika hloubky zkratu je nezávislá na elektrické úhlové rychlosti a zátěžném momentu a dosahuje přesnosti lepší než 97 % a to i pro stavy, na které síť nebyla učena. Obecně má síť problém rozeznat přechody mezi kategoriemi na nízkých hodnotách hloubky zkratu a elektrické úhlové rychlosti. Řešením by mohlo být rozšířit obrazový set o více dat z oblasti nízkých rychlostí a hloubek zkratu. Síť pro detekci přítomnosti zkratu dosahuje na modelu přesnosti blízké 100 %. Obdobně jako u sítě pro detekci hloubky zkratu je problém s přechodem mezi kategoriemi. Síť má problém odlišit 1 zkratovaný závit (spadající do kategorie *ITF*) od 0 zkratovaných závitů (spadajících do kategorie *Healthy*) při nízkých elektrických úhlových rychlostech.

Při použití sítí, které byly učeny na modelu, pro diagnostiku zkratů na skutečném pohonu nastává problém s přesností modelu. Chyba rozvážení se ukazuje být významnou překážkou pro fungování sítě pro detekci hloubky zkratu na reálném pohonu. Pokud má být zkrat diagnostikován s přesností 4 % zkratovaného vinutí nelze pro učení použít model, ve kterém se chyba rozvážení pohybuje mezi 5 až 10 %. Síť pro detekci přítomnosti zkratu naproti tomu naráží na problém kvality filtrace a nesymetrie motoru. Skutečný motor nemá vždy přesně stejně dlouhá fázová vinutí, což může způsobit miniaturní rozvážení, které už síť detekuje jako zkrat. Stejný problém nastává s náhodným šumem na motoru, který není v pásmech mezi zájmovými frekvencemi dokonale potlačen. Řešením těchto problémů by mohlo být posunutí hranice udávající, že na motoru zkrat je (například na 3 zkratované závity). Tabulka 2.4 udává výsledky diagnostiky pro vybrané stavy na skutečném pohonu.

Tab. 2.4: Výsledky diagnostiky pro vybrané stavy na skutečném pohonu

Stav: (u ; ω_e ; T_{load})	Hloubka	Přítomnost	Stav: (u ; ω_e ; T_{load})	Hloubka	Přítomnost
(0/25; 400 rad/s; 0 Nm)	0/25	<i>Healthy</i>	(6/25; 400 rad/s; 0.5 Nm)	4-6/25	<i>ITF</i>
(0/25; 800 rad/s; 0 Nm)	0/25	<i>Healthy</i>	(6/25; 800 rad/s; 0.5 Nm)	1-3/25	<i>ITF</i>
(0/25; 1200 rad/s; 0 Nm)	1-3/25	<i>ITF</i>	(6/25; 1600 rad/s; 0.5 Nm)	1-3/25	<i>ITF</i>
(0/25; 1600 rad/s; 0 Nm)	1-3/25	<i>ITF</i>	(14/25; 600 rad/s; 0 Nm)	10-12/25	<i>ITF</i>
(3/25; 200 rad/s; 0 Nm)	4-6/25	<i>ITF</i>	(14/25; 1000 rad/s; 0 Nm)	10-12/25	<i>ITF</i>
(3/25; 800 rad/s; 0 Nm)	1-3/25	<i>ITF</i>	(14/25; 1600 rad/s; 0 Nm)	7-9/25	<i>ITF</i>
(3/25; 1600 rad/s; 0 Nm)	1-3/25	<i>ITF</i>	(14/25; 800 rad/s; 0.5 Nm)	10-12/25	<i>ITF</i>
(3/25; 800 rad/s; 0.5 Nm)	1-3/25	<i>ITF</i>	(14/25; 1400 rad/s; 0.5 Nm)	7-9/25	<i>ITF</i>
(3/25; 1600 rad/s; 0.5 Nm)	1-3/25	<i>ITF</i>	(25/25; 800 rad/s; 0 Nm)	19-21/25	<i>ITF</i>
(6/25; 200 rad/s; 0 Nm)	1-3/25	<i>ITF</i>	(25/25; 1600 rad/s; 0 Nm)	16-18/25	<i>ITF</i>
(6/25; 800 rad/s; 0 Nm)	4-6/25	<i>ITF</i>	(25/25; 800 rad/s; 0.5 Nm)	13-15/25	<i>ITF</i>
(6/25; 1600 rad/s; 0 Nm)	4-6/25	<i>ITF</i>	(25/25; 1600 rad/s; 0.5 Nm)	19-21/25	<i>ITF</i>

Závěr

Shrnutí práce

V této práci byl realizován model třífázového PMSM, který je schopný simulovat přítomnost mezizávitového zkratu na statorové fázi 'a'. Pro tento model bylo následně navrženo vektorové řízení. Návrh řízení zahrnoval i návrh architektury a parametrů regulátorů. Následně byl řídicí algoritmus implementován do skutečného pohonu s PMSM a byly srovnány výsledky simulace s měřenými daty.

Další část práce se zabývala předzpracováním signálů pro hluboké učení. Pro diagnostiku mezizávitových zkratů byly použity signály elektrické úhlové rychlosti, fázových proudů a napětí PWM. Předzpracování těchto signálů zahrnovalo mimo jiné i návrh FIR Filtru, řešení problému proměnného balíku dat, aplikaci spojitě Vlnkové transformace a kompozici obrazu.

Poslední část práce se zabývala použitím předučené konvoluční neuronové sítě GoogLeNet pro diagnostiku mezizávitových zkratů. Na základě simulací byl vytvořen obrazový set pro učení sítí. Následně byly modifikací originální CNN GoogLeNet vytvořeny dvě nové sítě. První z těchto sítí slouží pro detekci hloubky zkratu a druhá pro detekci přítomnosti zkratu. Tyto sítě byly použity pro detekci mezizávitových zkratů na skutečném pohonu a na modelu. Výsledky diagnostiky byly srovnány a byly stanoveny vlivy udávající přesnost diagnostiky.

Dosažené výsledky a zjištěné problémy

Shoda modelu pohonu se skutečným pohonem byla vyjádřena pomocí chyby rozvážení δ . Tato určuje jak se liší rozvážení proudů v modelu od rozvážení proudů na skutečném pohonu. Chyba rozvážení je pro některé stavy uvedena v tabulce 2.3. Obecně platí, že se chyba rozvážení pro použitý model pohybuje mezi 5 až 10 %. Dominantní vliv na tuto chybu mají nepřesně určené parametry skutečného pohonu, nesplnění podmínky $R_f \ll R_s$ a způsob modelování mechaniky rotoru. Nesplnění podmínky $R_f \ll R_s$ vyústí ve vzrůst chyby rozvážení zejména při vyšších hodnotách hloubky zkratu, kde má poměr odporů dominantní vliv na rozvážení proudů. Mechanika rotoru je modelována lineární závislostí v kombinaci s funkcí signum. Toto není zcela vhodné, protože se třecí momenty obecně lineárně nechovají. Model mechaniky rotoru by se dal vylepšit naměřením tabulky třecích momentů v závislosti na úhlové rychlosti motoru a následnou interpolací mezi těmito body. Aby však bylo možné chybu rozvážení zcela minimalizovat, musel by být použit model uvažující geometrii konkrétního motoru.

Hlavním problémem v oblasti předzpracování signálů je schopnost navrženého FIR filtru potlačit náhodný šum ve frekvenčních pásmech, které leží mezi zájmovými

frekvencemi. Potlačení těchto pásem se odvíjí od řádu filtru, který je spřažen s minimálním počtem vzorků signálu. Použitý filtr je 65. řádu. Aby bylo potlačení dostatečně účinné, musel by být použit filtr alespoň 100. řádu. Toto by ovšem vedlo na nutnost delšího sběru dat (bylo by potřeba více vzorků signálu). Dalším problémem této části je Vlnková transformace, která nepatrně zkresluje informaci o lokalitě jevů v čase a frekvenci. Toto má za následek mírné zkreslení výsledného obrazu oproti vstupním signálům.

Sítě vytvořené pro diagnostiku hloubky a přítomnosti zkratu byly učeny na základě dat ze simulací a dosahují na modelu přesnosti blízké 100 %. Toto platí i pro stavy, na které daná síť nebyla učena. Při použití sítí pro diagnostiku mezizávitových zkratů na skutečném pohonu přesnost rapidně klesá. Přesnost diagnostiky hloubky zkratu je velice ovlivněna přesností použitého modelu. Ukazuje se, že nelze diagnostikovat zkrat s krokem 4 % vinutí, pokud je chyba rozvážení 5 až 10 %. Přesnost diagnostiky přítomnosti zkratu je naproti tomu zejména ovlivněna schopností filtru potlačit pásma mezi zájmovými frekvencemi. Potlačení šumu je klíčové pro stanovení přechodu mezi průběhy signálů „zdravého“ motoru a motoru s ITF. Výsledky diagnostiky pro vybrané stavy lze vidět v tabulce 2.4. Aby byla přesnost diagnostiky na skutečném pohonu zvýšena, musí být minimalizována chyba rozvážení a vylepšena filtrace.

Největším úspěchem této práce je potom schopnost sítě určit hloubku zkratu bez ohledu na hodnotu úhlové rychlosti motoru a zátěžného momentu.

Směry dalšího postupu

Jak vyplývá z předchozích kapitol, jedním z hlavních problémů je vysoká chyba rozvážení. Chyba rozvážení by se dala snížit například použitím FEM modelu motoru a vylepšením modelu mechaniky rotoru. Toto by šlo provést například naměřením hodnot momentu tření při různých rychlostech a interpolací mezi naměřenými body. Dalším problémem navrženého řešení je výpočetní náročnost algoritmu předzpracování signálů a průchodu obrazu CNN. Vylepšením by mohlo být použití LSTM (long short-term memory) sítě. Tento typ sítí se používá pro diagnostické úlohy na základě 1-D signálů. Díky tomuto typu sítí by také odpadla nutnost provádět převod signálu do 2-D (Vlnková transformace a kompozice obrazu). Celý algoritmus by se tedy mnohonásobně zrychlil. Dalším vylepšením potom může být použití komplexnější varianty vektorového řízení, která zahrnuje propracovanější architekturu odvážení a filtr úhlové rychlosti. Nakonec by mohli být do diagnostiky zahrnuti i přechodové děje a hodnota odporu zkratu, která by mohla být určena na základě průběžné identifikace R_s .

Literatura

- [1] SUL, S. K. *Control of electric machine drive system*. Hoboken, N.J.: Wiley-IEEE Press, 2011. ISBN 978-0-470-59079-9.
- [2] Cypress Semiconductor Corporation. *Coordinate Transform in Motor Control*. [online]. San Jose, California, ©2011-2017 [cit. 2019-11-17]. Dostupné z: <https://www.cypress.com/file/222111/download>
- [3] Freescale Semiconductor, Inc. *PMSM Electrical Parameters Measurement*. [online]. ©2013 [cit. 2019-11-17]. Dostupné z: <https://www.nxp.com/docs/en/application-note/AN4680.pdf>
- [4] KRAUSE, P. C., O. WASYNCZUK, S. D. SUDHOFF a S. PEKAREK. *Analysis of electric machinery and drive systems*. Third edition. Hoboken, New Jersey: Wiley, 2013. ISBN 978-1-118-02429-4.
- [5] GU, B. G. Study of IPMSM Interturn Faults Part I: Development and Analysis of Models With Series and Parallel Winding Connections. *IEEE Transactions on Power Electronics*. 2016, **31**(8), 5931-5943. DOI: 10.1109/TPEL.2015.2496142. ISSN 0885-8993.
- [6] GU, B. G., J. H. CHOI a I. S. JUNG. Development and Analysis of Interturn Short Fault Model of PMSMs With Series and Parallel Winding Connections. *IEEE Transactions on Power Electronics*. 2014, **29**(4), 2016-2026. DOI: 10.1109/TPEL.2013.2265400. ISSN 0885-8993.
- [7] SHEIKH-GHALAVAND, B., S. VAEZ-ZADEH a A. HASSANPOUR ISFAHANI. An Improved Magnetic Equivalent Circuit Model for Iron-Core Linear Permanent-Magnet Synchronous Motors. *IEEE Transactions on Magnetics*. 2010, **46**(1), 112-120. DOI: 10.1109/TMAG.2009.2030674. ISSN 0018-9464.
- [8] RATHNAKUMAR, D., J. LAKSHMANAPERUMAL a T. SRINIVASAN. A New Software Implementation of Space Vector PWM. *Proceedings. IEEE SoutheastCon, 2005*. IEEE, 2005, s. 131-136. DOI: 10.1109/SECON.2005.1423232. ISBN 0-7803-8865-8.
- [9] ADDISON, P. S. *The illustrated wavelet transform handbook: introductory theory and applications in science, engineering, medicine and finance*. New York: Taylor & Francis, c2002. ISBN 0-7503-0692-0.

- [10] The MathWorks, Inc. *Wavelet Toolbox™ Getting Started Guide*. [online]. Natick, Massachusetts, ©1997–2020 [cit. 2019-04-10]. Dostupné z: https://www.mathworks.com/help/pdf_doc/wavelet/wavelet_gs.pdf
- [11] JIRKOVSKÝ, J. Strojové učení metodou Deep learning. *MM Průmyslové spektrum* [online]. 8.03.2017 [cit. 2020-04-15]. Dostupné z: www.mmspektrum.com/clanek/strojove-uceni-metodou-deep-learning.html
- [12] GOODFELLOW, I., Y. BENGIO a A. COURVILLE. *Deep Learning* [online]. MIT Press, 2016 [cit. 2020-01-02]. Dostupné z: <http://www.deeplearningbook.org>
- [13] SZEGEDY, C., W. LIU, Y. JIA, et al. Going deeper with convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015, s. 1-9. DOI: 10.1109/CVPR.2015.7298594. ISBN 978-1-4673-6964-0.
- [14] PAWARA, P., E. OKAFOR, O. SURINTA, L. SCHOMAKER a M. WIERING. Comparing Local Descriptors and Bags of Visual Words to Deep Convolutional Neural Networks for Plant Recognition. *Proceedings of the 6th International Conference on Pattern Recognition Applications and Methods*. SCITEPRESS - Science and Technology Publications, 2017, s. 479-486. DOI: 10.5220/0006196204790486. ISBN 978-989-758-222-6.
- [15] CIRA, F., M. ARKAN a B. GUMUS. A new approach to detect stator fault in permanent magnet synchronous motors. *2015 IEEE 10th International Symposium on Diagnostics for Electrical Machines, Power Electronics and Drives (SDEMPED)*. IEEE, 2015, s. 316-321. DOI: 10.1109/DEMPED.2015.7303708. ISBN 978-1-4799-7743-7.
- [16] MOOSAVI S. S., A. DJERDIR, Y. AIT-AMIRAT, D. A. KHABURI. ANN based fault diagnosis of permanent magnet synchronous motor under stator winding shorted turn. *Electric Power Systems Research* [online]. **128**, s. 67-82. [cit. 2020-04-18]. ISSN 0378-7796. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S0378779615000917?via%3Dihub>
- [17] HADEF M. *PMSM Stator Winding Short Circuit Fault Diagnosis via Neural Network* [online]. **12**, s. 25-37. [cit. 2020-04-18]. Dostupné z: https://www.researchgate.net/publication/332974720_PMSM_Stator_Winding_Short_Circuit_Fault_Diagnosis_via_Neural_Network

Seznam symbolů, veličin a zkratek

Značky a symboly

B	[Nm s/rad]	Koeficient viskózního tření
C	[-]	matice koeficientů Vlnkové transformace
e	[-]	Eulerovo číslo
e	[V]	zpětně indukované napětí
f	[Hz]	frekvence
$F(m,n)$	[-]	Filtr (konvoluční jádro)
i	[A]	elektrický proud
j	[-]	imaginární jednotka
J	[kg m ²]	moment setrvačnosti
k	[-]	krok/vzorek
K	[-]	zesílení
L	[H]	indukčnost
p	[-]	počet pólových dvojic
R	[Ω]	elektrický odpor
s	[-]	škálovací parametr mateřské vlnky
t	[s]	čas
T	[N m]	kroučící moment
T	[s]	perioda/časová konstanta
u	[-]	závažnost zkratu
$u(k)$	[-]	akční zásah v kroku k
v, V	[V]	elektrické napětí
$w(k) - y(k)$	[-]	regulační odchylka v kroku k
$x(t)$	[-]	jednorozměrný vstupní signál Vlnkové transformace
$x(i,j)$	[-]	dvourozměrný vstupní signál konvoluce
$y(i,j)$	[-]	dvourozměrný výstupní signál konvoluce
δ	[%]	chyba rozvážení
θ	[rad]	úhel natočení
τ	[-]	posun mateřské vlnky
Φ	[-]	otcovská vlnka
ψ	[Wb]	spřažený magnetický tok
Ψ	[-]	mateřská vlnka
ω	[rad/s]	úhlová rychlost

Indexy

<i>a</i>	statorová fáze 'a'
<i>b</i>	statorová fáze 'b'
<i>c</i>	celková
<i>c</i>	statorová fáze 'c'
<i>d</i>	reálná osa rotorového souřadného systému
<i>dc</i>	stejnoseměrné
<i>e</i>	elektrický
<i>f</i>	mezi-závitového zkratu
<i>fl</i>	fluktuace
<i>i</i>	integrační
<i>load</i>	zátěžný
<i>m</i>	mechanický
<i>max</i>	maximální
<i>ob</i>	obálky
<i>pm</i>	permanentních magnetů
<i>q</i>	imaginární osa rotorového souřadného systému
<i>s</i>	statorem
<i>s, τ</i>	označení dceřinné vlnky
<i>sm</i>	vlastní a vzájemná
<i>stat</i>	suché
<i>vz</i>	vzorkovací
α	reálná osa statorového souřadného systému
β	imaginární osa statorového souřadného systému
ω	otáčkové smyčky

Zkratky

B	blue
CNN	convolutional neural network
FEM	finite element method
FIR	finite impulse response
G	green
IPMSM	interior permanent magnet synchronous motor
ITF	inter turn fault
LSTM	long short-term memory
PI	proportional-integral
PMSM	permanent magnet synchronous motor
PS	proportional-summation
PWM	pulse width modulation
R	red
ReLU	rectified linear function
SGDM	stochastic gradient descent with momentum
SPMSM	surface-mounted permanent magnet synchronous motor
SVPWM	space vector pulse width modulation

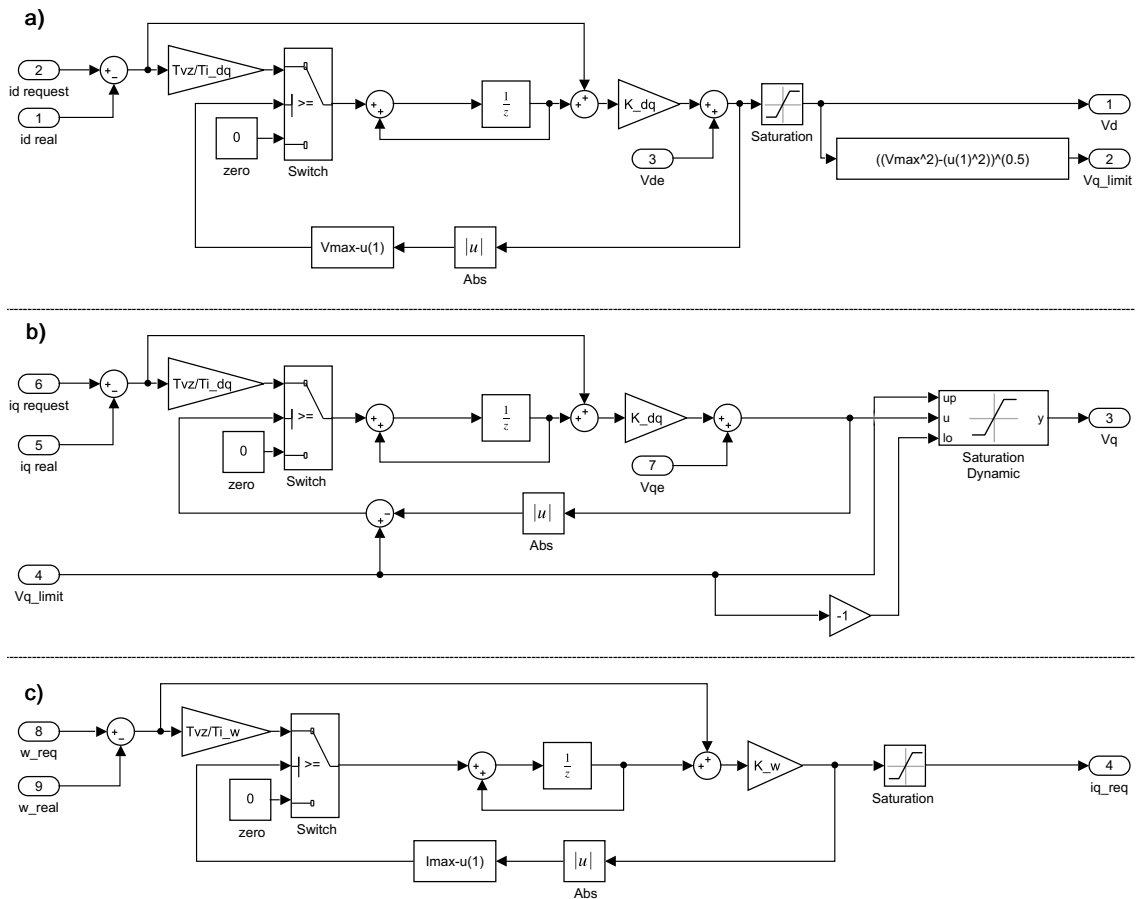
Seznam příloh

A Přílohy ke kapitole Tvorba modelu	57
B Přílohy ke kapitole Předzpracování signálů	59

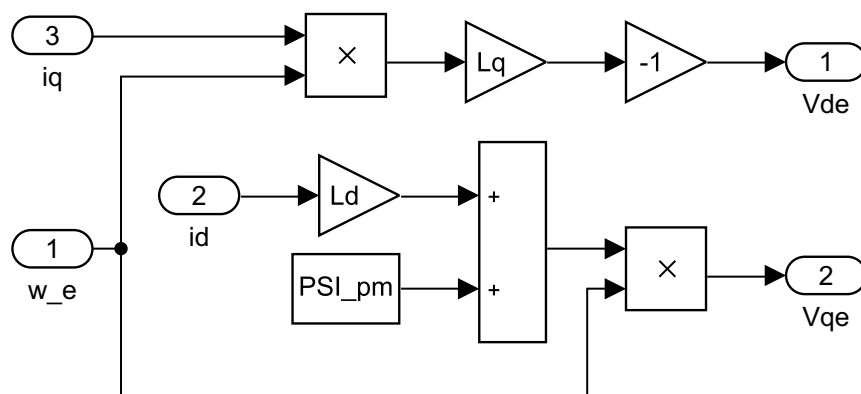
A Přílohy ke kapitole Tvorba modelu

Výpis A.1: Část definice PMSM s ITF v programu Simscape Component

```
1 component IPMSM_ITF
2 parameters
3 Rs = {0, 'Ohm'}; % Series Resistance of healthy phase Rs
4 Ld = {0, 'H'}; % Inductance in direct axis Ld
5 Lq = {0, 'H'}; % Inductance in quadrature axis Lq
6 ...
7 end
8 nodes
9 a_p = foundation . electrical . electrical ;
10 a_n = foundation . electrical . electrical ;
11 R = foundation.mechanical.rotational.rotational;
12 C = foundation.mechanical.rotational.rotational;
13 ...
14 end
15 variables
16 Va = {0, 'V'}; %Phase 'a' voltage
17 ia = {0, 'A'}; %Phase 'a' current
18 T = {0, 'N*m'}; %Generated torque
19 w = {0, 'rad/s'}; %mechanical angular velocity
20 ...
21 end
22 branches
23 ia : a_p.i -> a_n.i;
24 T : C.t -> R.t;
25 ...
26 end
27 equations
28 Va == a_p.v-a_n.v;
29 w == R.w - C.w;
30 ...
31 V == R*i + L_der_t*i + L*i_der + e; %Voltage equation
32 T == p*(0.5*i'*L_der*i+i'*PSI_der); %Generated torque
33 ...
34 end
35 end
```



Obr. A.1: a) PS regulátor d složky proudu, b) PS regulátor q složky proudu, c) PS regulátor rychlosti



Obr. A.2: Architektura bloku zrušení vazeb

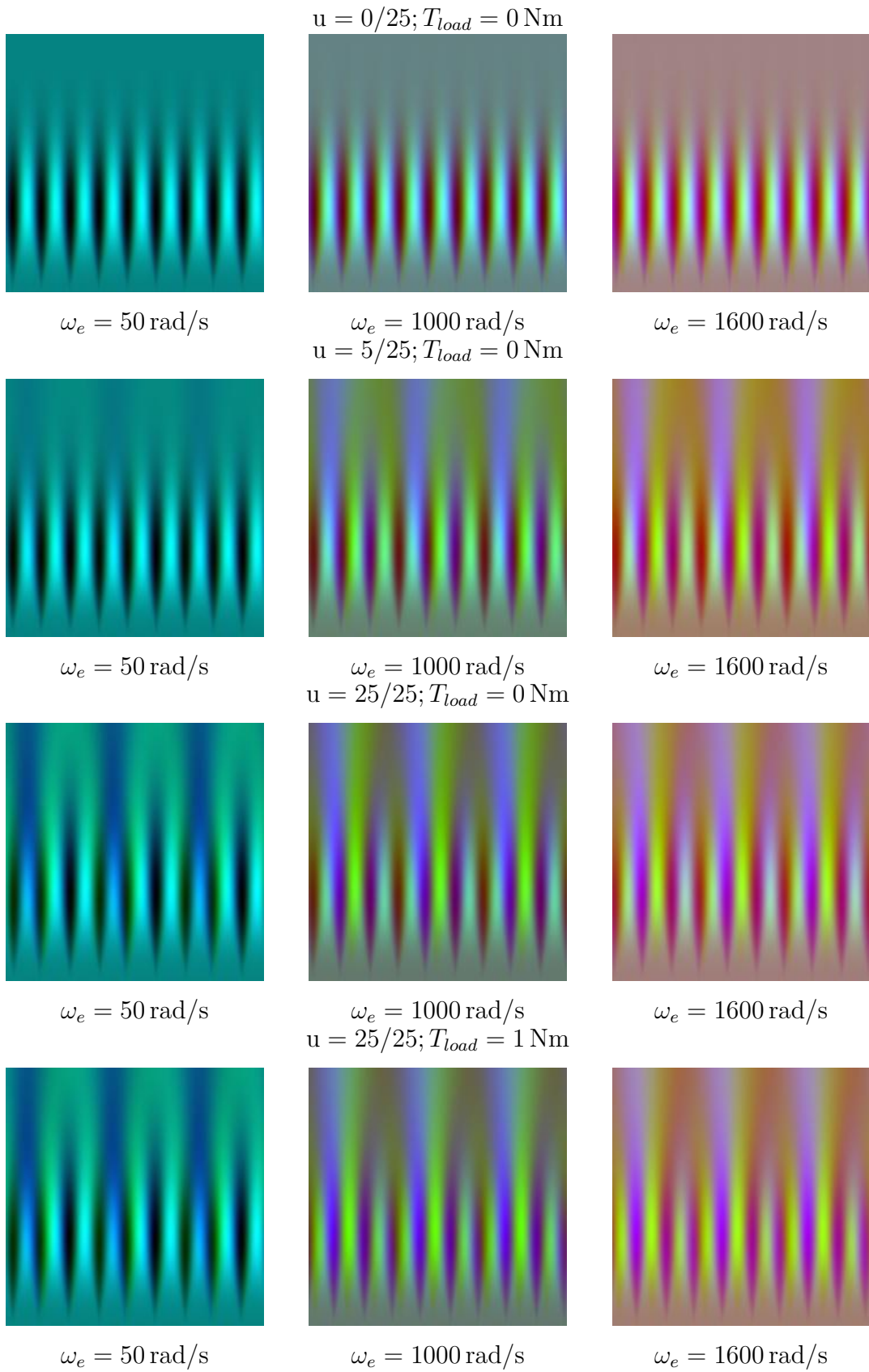
B Přílohy ke kapitole Předzpracování signálů

Výpis B.1: Určení koeficientů filtru a filtrace implementovaná jako MATLAB funkce

```
1 function [Output] = F_filter(Input,w_e)
2 Input=double(Input);           %přetypování na double
3 freq = w_e/(2*pi);             %výpočet frekvence
4 ord = 65;                       %řád filtru
5 Fs = 10000;                     %vzorkovací kmitočet
6 bnd = [(freq/50) (2*freq-freq/50)...
7         (2*freq+freq/50) (4*freq-freq/50)...
8         (4*freq+freq/50)]./(Fs/2); %norm. frekv. pásma
9 bW = fir1(ord,bnd,'DC-1');      %vytvoření filtru
10 Output = filtfilt(bW, 1, Input); %filtrace signálu
11 end
```

Výpis B.2: Filtrace a převzorkování proudových signálů

```
1 Ialpha = sqrt(2/3).*(Ia - 0.5.*Ib-0.5.*Ic);
2 Ibeta = sqrt(2/3).*((sqrt(3)/2).*Ib-(sqrt(3)/2).*Ic);
3 Iq = -Ialpha.*sin(theta_e) + Ibeta.*cos(theta_e);
4 Id = Ialpha.*cos(theta_e) + Ibeta.*sin(theta_e);
5 Iq_f = F_filter(Iq,mean(w_e));   %volání filtrace
6 Id_f = F_filter(Id,mean(w_e));   %volání filtrace
7 Ialpha_f= Id_f.*cos(theta_e)-Iq_f.*sin(theta_e);
8 Ibeta_f = Id_f.*sin(theta_e)+Iq_f.*cos(theta_e);
9 Ialpha_f = resample(Ialpha_f,1000,length(Ialpha_f));
10 Ibeta_f = resample(Ibeta_f,1000,length(Ibeta_f));
11 Ia_f = sqrt(2/3).*Ialpha_f;
12 Ib_f = (sqrt(2/3).*(-Ialpha_f + sqrt(3).*Ibeta_f))./2;
13 Ic_f = (sqrt(2/3).*(-Ialpha_f - sqrt(3).*Ibeta_f))./2;
```



Obr. B.1: Ukázka obrazů pro některé hodnoty u , T_{load} a ω_e