

**ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA ELEKTROTECHNICKÁ**

**KATEDRA ELEKTROMECHANIKY A VÝKONOVÉ
ELEKTRONIKY**

BAKALÁŘSKÁ PRÁCE

Rozbor složek zkratového proudu

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta elektrotechnická

Akademický rok: 2017/2018

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Václav MAYER**

Osobní číslo: **E15B0075P**

Studijní program: **B2612 Elektrotechnika a informatika**

Studijní obor: **Elektrotechnika a energetika**

Název tématu: **Rozbor složek zkratového proudu**

Zadávající katedra: **Katedra elektromechaniky a výkonové elektroniky**

Z á s a d y p r o v y p r a c o v á n í :

1. Proveďte analýzu střídavé a stejnosměrné složky z dodaného průběhu zkratového proudu.
2. Analýzu řešte pomocí skriptu (*.m file) v jazyce MATLAB.
3. Okomentujte jednotlivé kroky, princip a postup řešení ve skriptu.
4. Specifikujte závěrem podmínky, kdy navržený postup bude pracovat správně a kdy selže.

Rozsah grafických prací: **podle doporučení vedoucího**

Rozsah kvalifikační práce: **30 - 40 stran**

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

- 1. Student si vhodnou literaturu vyhledá v dostupných pramenech podle doporučení vedoucího práce.**

Vedoucí bakalářské práce:

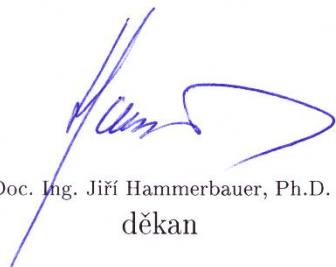
Doc. Ing. Bohumil Skala, Ph.D.

Katedra elektromechaniky a výkonové elektroniky

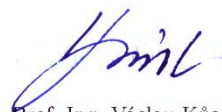
Datum zadání bakalářské práce: **10. října 2017**

Termín odevzdání bakalářské práce: **7. června 2018**

Doc. Ing. Jiří Hammerbauer, Ph.D.
děkan



Prof. Ing. Václav Kús, CSc.
vedoucí katedry



V Plzni dne 10. října 2017

Abstrakt

Tato bakalářská práce je zaměřena na přechodové jevy při blízkých zkratech na synchronních generátorech. Úkolem práce je popsat používané synchronní stroje, jejich výhody a nevýhody. Dalším úkolem je napsat skript v jazyce MATLAB, který bude zpracovávat změřené časové průběhy a analyzovat jejich jednotlivé složky, a popsat jeho chování, způsob výpočtu a popsat okolnosti, za jakých daný skript bude vracet chybné výsledky.

Klíčová slova

Synchronní stroj, zkrat, točivé magnetické pole, analýza, MATLAB, stejnosměrná složka, střídavá složka

Abstract

This bachelor thesis is focused on transient phenomena in near short circuits on synchronous generators. The aim of the thesis is to describe used synchronous machines, their advantages and disadvantages. Another task is to write a MATLAB script that will process measured timescales and analyze their individual components and describe its behavior, calculation method, and describe the conditions when the script will return erroneous results.

Key words

Synchronous machine, short circuit, rotating magnetic field, analysis, MATLAB, DC component, AC component

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně, s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této bakalářské práce.

Dále prohlašuji, že veškerý software, použitý při řešení této bakalářské práce, je legální.

.....
podpis

V Plzni dne 30.5.2018

Jméno příjmení

Poděkování

Tímto bych rád poděkoval vedoucímu diplomové práce doc. Ing. Bohumilovi Skalovi, Ph.D. za cenné profesionální rady, připomínky, metodické vedení práce a doporučení při výběru literatury.

Obsah

OBSAH	8
ÚVOD	9
SYNCHRONNÍ STROJE	11
1.1 KONSTRUKCE SYNCHRONNÍCH STROJŮ.....	11
1.1.1 Stroj s hladkým rotorem.....	11
1.1.2 Stroje s vyniklými póly.....	12
1.2 GENERÁTOROVÝ CHOD.....	13
1.3 BUDÍCÍ SYSTÉMY SYNCHRONNÍCH STROJŮ.....	15
1.4 PŘECHODNÉ DĚJE NA SYNCHRONNÍCH STROJÍCH.....	16
1.4.1 Střídavá složka zkratu.....	17
1.4.2 Stejnoseměrná složka zkratu.....	18
1.4.3 Celkový zkratový proud.....	19
1.5 MĚŘENÍ PRŮBĚHŮ ZKRATOVÝCH PROUDŮ.....	20
1.5.1 Rozdělení průběhu na jednotlivé složky.....	21
2 ANALÝZA ZKRATOVÉHO PROUDU	26
2.1 FOURIEROVA TRANSFORMACE.....	27
2.1.1 FFT jako funkce MATLABu.....	28
2.2 POPIS VYPRACOVANÉHO SKRIPTU.....	29
2.2.1 Nastavení základních proměnných skriptu.....	29
2.2.2 Vstupní data a zjištění jejich základních údajů.....	30
2.2.3 Vzorkovací frekvence.....	31
2.2.4 Postup při záporné aperiodické složce.....	32
2.2.5 Vytvoření časových oken.....	32
2.2.6 Rozbor pomocí FFT.....	34
2.2.7 Oprava fázového posunu.....	36
2.2.8 Proložení.....	36
2.2.9 Zpětná rekonstrukce složek a celkového průběhu.....	39
3 ZÁVĚR	42
SEZNAM LITERATURY A INFORMAČNÍCH ZDROJŮ	43
PŘÍLOHY	1
PŘÍLOHA A – SKRIPT.....	1

Úvod

Synchronní stroje tvoří důležitou skupinu strojů především na poli výroby elektrické energie a jako synchronní kompenzátory jalového výkonu. Mezi hlavní přednosti tohoto typu strojů patří to, že se rotor při běhu otáčí stejnými otáčkami jako statorové pole.

Při rychlých změnách chodu stroje dochází k přechodovým jevům a změnám reaktancí stroje. Tyto děje mohou vyústit až ve ztrátu synchronismu stroje a tím k jeho zastavení. Mezi velice rychlé přechodné děje patří také zkraty na statoru, které vyvolávají následný zkratový proud statorem. Tyto proudy fyzicky extrémně namáhají jednotlivé části vinutí, a to především silově. Proto je třeba znát, jak se takový stroj při zkratu v daném okamžiku chová, abychom mohli předvídat vlivy na vinutí. [1]

Tato bakalářská práce se pak ve své praktické části zabývá vytvořením skriptu, kterým by bylo možné změřený zkratový proud zanalyzovat a zjistit některé užitečné informace o stroji, na kterém toto měření bylo provedeno. Mezi tyto informace patří například časové konstanty jednotlivých složek proudu, velikosti amplitud daných složek a další.

Seznam symbolů a zkratk

ω	úhlová rychlost rotoru
α_0	úhel odpovídající okamžiku zkratu
X_d	podélná synchronní reaktance statorového vinutí
X'_d	přechodná reaktance statorového vinutí
X''_d	rázová reaktance statorového vinutí
$X_{1\sigma}$	rozptylová reaktance statoru
X_{ad}	nenасыená podélná reaktance reakce kotvy
$X_{b\sigma}$	rozptylová reaktance budícího vinutí
$X_{t\sigma d}$	rozptylová podélná reaktance tlumícího vinutí
T'_d	přechodná časová konstanta zkratu
T''_d	rázová časová konstanta zkratu
T''_a	aperiodická časová konstanta zkratu
U_i	elektromotorické napětí fáze statoru
i_{st}	střídavá složka zkratu
i_{ap}	stejnoseměrná složka zkratu
i_k	celkový zkratový proud

Synchronní stroje

Synchronní stroje spadají do kategorie strojů, které pracují při stejné rychlosti rotoru, jako je rychlost statorového pole. Tyto stroje jsou nejzastoupenějším typem při výrobě elektřiny v elektrárnách, kde dosahují i přes 1000 MW dodávaného výkonu do sítě. Dále se tyto stroje používají jako synchronní kompenzátory pro kompenzaci induktivního jalového výkonu v síti. V dnešní době se synchronní motory prosazují také u menších pohonů a v trakci, kde se často objevují v provedení s permanentními magnety. [1, 2]

1.1 Konstrukce synchronních strojů

Stroj se skládá ze statoru, rotoru a ostatních částí, jako jsou ložiska, svorkovnice atd. Stator je vyroben z jednotlivých navzájem izolovaných plechů, ve kterých jsou po vnitřní straně vystřiženy jednotlivé drážky, ve kterých je uloženo trojfázové statorové vinutí.

Rotor se v provedení s vyniklými póly vyrábí výhradně z plechů, které ale nemusí být od sebe izolované, neboť se vůči rotoru žádné magnetické pole netočí. Z plechů bývá vyroben hlavně proto, že je to konstrukčně jednodušší a přesnější než rotor vyrábět z jednoho kusu materiálu. U provedení s hladkým rotorem může být i v provedení z jednoho kusu. Na rotoru se nachází budící vinutí, které vytváří potřebné statické magnetické pole. Budící vinutí je napájeno buď z kroužků, nebo budiče na hřídeli stroje. Toto vinutí je v moderních strojích postupně nahrazováno permanentními magnety, kde pak odpadá potřeba zdroje pro budící proud, sběracích kroužků a vinutí samotného. Stroje s permanentními magnety se pak vyrábí až do jmenovitého výkonu cca 5000 kVA. Dále se na rotoru větších strojů nachází typicky tlumící vinutí, které se nepodílí na přenosu energie, ale má za úkol utlumovat kývání rotoru, které vzniká při rychlých změnách zatížení. [3]

1.1.1 Stroj s hladkým rotorem

Stroje s hladkým rotorem pracují převážně při vyšších otáčkách a jsou napojeny přímo na parní a spalovací turbíny v elektrárnách (zpravidla 3000 a 3600 otáček/min). Tomuto typu se také říká „turbostroje“.

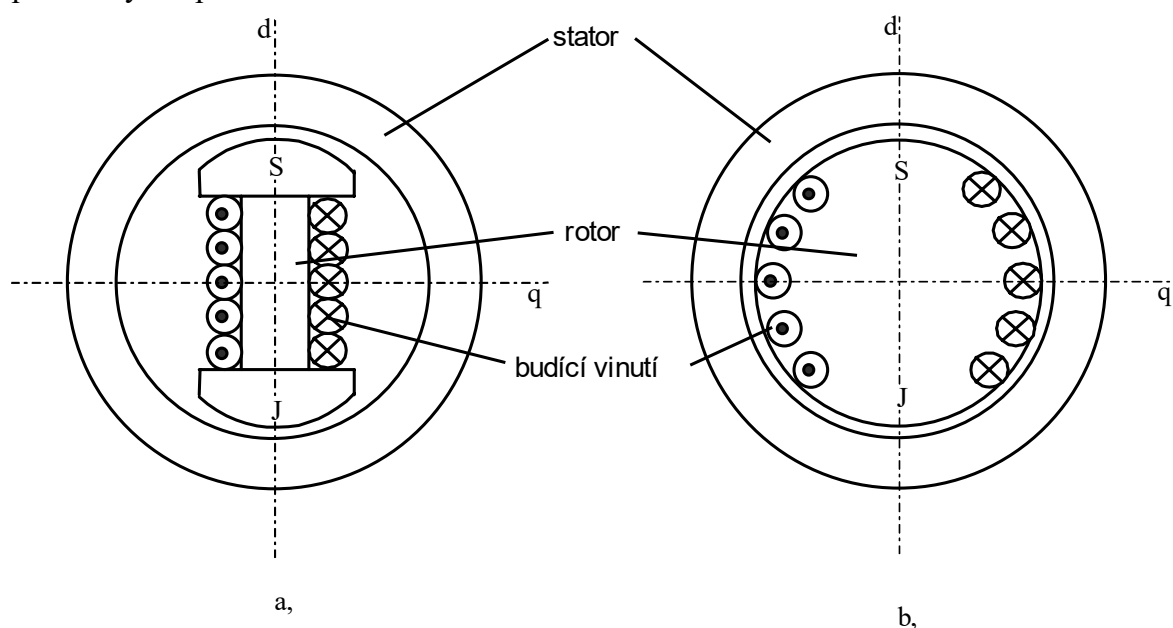
Rotor bývá v provedení jednoho dlouhého válce s průměrem do jednoho metru, na kterém je navinuto budící vinutí nejčastěji v jednopólovém provedení. Při dostatečně velkém stroji

je možno použít i 4pólové provedení. Tlumič vinutí je realizováno tyčemi s lichoběžníkovými profily zasunutými do drážek pro vinutí na vnější straně tak, aby zároveň bránily posunu budícího vinutí, na které při otáčení působí odstředivá síla. [2, 3]

1.1.2 Stroje s vyniklými póly

Stroje s vyniklými póly se používají tam, kde je třeba vyrábět elektřinu z pomaloběžného zdroje. Příkladem takového zdroje mohou být například vodní turbíny. Proto se tomuto typu také říká „hydrostroje“.

Rotor je ve tvaru krátkého válce s velkým průměrem, okolo něhož jsou po obvodu jednotlivé póly. Velký průměr rotoru je zde dán také tím, že při pomalých otáčkách a stejném výkonu zde musí být použita hřídel s větším průměrem než u turbostrojů (větší přenášený moment). Okolo každého pólu je umístěno budící vinutí. U tohoto typu stroje se tlumič vinutí realizuje pomocí tyčí v pólech.



Obrázek 1: a) stroj s vyniklými póly, b) stroj s hladkým rotorem [2]

1.2 Generátorový chod

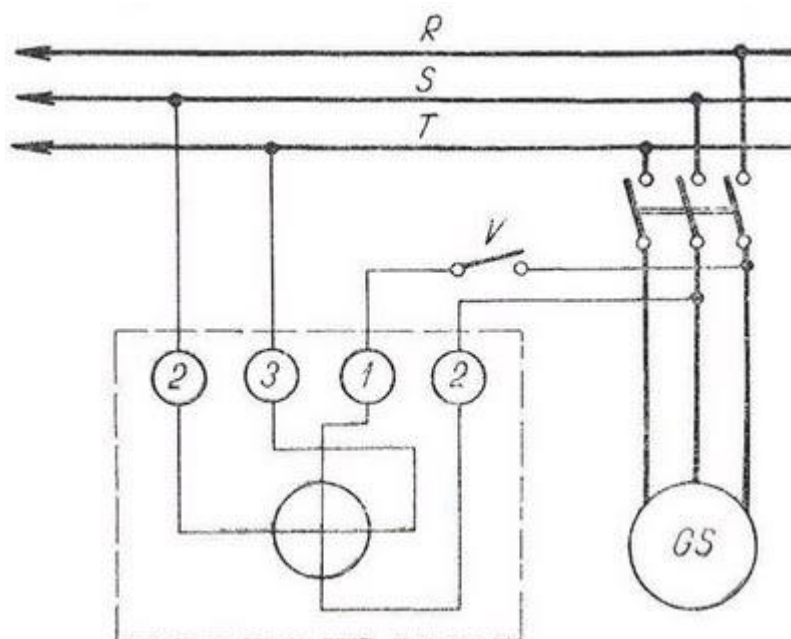
V generátorickém chodu stroje se rozlišují dva základní stavy.

Prvním stavem je, když stroj pracuje v ostrovním režimu. Tímto režimem se rozumí, že stroj není připojen k veřejné distribuční nebo přenosové síti, nepracuje více strojů do téže sítě a vyrábí elektrickou energii jen pro lokální zátěž. V tomto režimu nejsou tak velké nároky na udržování velikosti a kmitočtu výstupního napětí. Jako příklad těchto sítí se dají považovat například náhradní agregátory v nemocnicích, kde je požadavek na elektrickou energii i v případě výpadku veřejné sítě. [3, 4]

Druhým provozním stavem je, když stroj dodává výkon do tvrdé sítě. Do sítě obvykle pracuje velké množství zdrojů, které zajišťují konstantní velikost napětí na straně sítě, její konstantní frekvenci a dostatečně malou impedanci. V tomto režimu nelze nijak výrazně měnit parametry sítě a jediné, co stroj může dělat, je dodávat nebo odebírat výkon ze sítě. Aby mohl stroj na tvrdé síti pracovat, je nutno ho k ní připojit. Připojit jej ale lze, jen pokud jsou dodrženy následující podmínky:

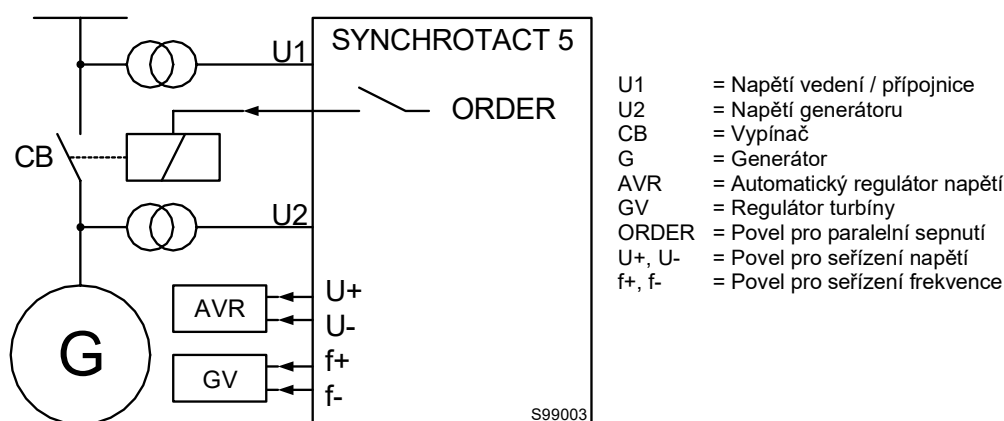
- Stejná frekvence sítě i generátoru. Pokud nebude dodrženo, síť vnutí svojí frekvenci statorovému poli a dojde k prudkému urychlení, nebo naopak zpomalení rotoru a k velkým proudům na statoru. Tento náraz pak významně mechanicky namáhá celé soustrojí.
- Stejná velikost statorového napětí, jako je napětí na blokovém transformátoru, když je připojen k síti. Velikost napětí statoru lze v nepřipojeném stavu měnit pomocí velikosti budícího proudu.
- Minimální fázový posun mezi napětími stroje a napětími sítě. Pokud bude rozdíl fázového posunu malý, dojde k přechodovému ději, kdy se začne kývat rotor, z důvodu průtoku vyrovnávacího proudu statorem. Při velkém rozdílu posunu dochází ke zkratovému proudu, odpovídajícímu danému rozdílovému napětí.
- Stejný sled fází. Při nedodržení dochází ke zkratovým proudům, které mohou vést k destrukci stroje.

Pro fázování stroje se dříve používaly speciální fázovací přístroje, tzv. synchronoskopy. Tyto přístroje se skládají ze dvou cívek, kdy se jedna připojí na stranu sítě a druhá na generátor. Potom, pokud je rozdílná frekvence zdrojů, se ručička přístroje otáčí na jednu nebo druhou stranu podle toho, která frekvence je vyšší. Na přístroji se také nachází značka značící nulový fázový posun. [5]



Obrázek 2: Schéma zapojení synchronoskopu [5]

V dnešní době se pro fázování používají spíše digitální přístroje, které měří jak frekvence, napětí, fázový posuv, tak jsou schopné vydávat signály pro řízení napětí generátoru a otáček turbíny a ve vhodném okamžiku vydat řídicí povel k sepnutí daného spínacího prvku a generátor přifázovat na síť. Tento povel je nutné vydávat s předstihem, neboť vypínačům trvá nějakou dobu, než přejdou do sepnutého stavu.



Obrázek 3: Zapojení digitálního fázovacího přístroje (ABB SYNCHROACT® 5) [16]

1.3 Budicí systémy synchronních strojů

Budicí systémy tvoří důležitou část stroje. Tyto systémy mají přímý vliv na magnetické pole vytvářené rotorem. Tyto systémy se skládají z několika základních komponent, kdy pravděpodobně nejdůležitější komponentou je výkonový usměrňovač, na který jsou kladeny požadavky na velký proud. Nejčastěji se skládá z můstkového třífázového zapojení tyristorů. Další komponentou jsou ochranné obvody, jejichž úkolem je ochránit usměrňovač a jednotlivá vinutí stroje proti přepětí a nadproudu. Typicky jsou složeny z odporníků, kondenzátorů a k nim příslušných tyristorů. Dále se skládají z odbuzovače, jehož hlavním úkolem je omezovat přepětí, které vzniká při odlehčení zátěže a při odepínání budicího obvodu. Typicky se může jednat také o odporníky. [3, 6]

Další neméně důležitou komponentou je regulátor budicího proudu. Jedná se o počítač s příslušným softwarem. Základním úkolem regulátoru je řídit budicí proud, jehož řízením lze omezovat statorové a rotorové proudy, hlídat mez podbuzení (hlídat minimální budicí proud, aby nedošlo ke ztrátě synchronismu) a také řídit účinník celého stroje.

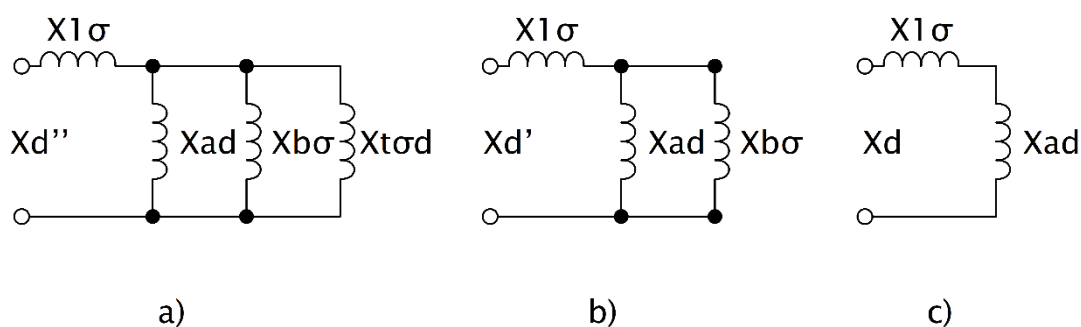
Podle zdroje energie lze budicí systémy rozdělit na systémy závislé, které jako zdroj budicího proudu využívají samotný generátor a výstupní napětí usměrňovače je tudíž závislé na svorkovém napětí generátoru, a na systémy nezávislé, které jsou napájeny buď z budiče na hřídeli, nebo z jiného naprosto nezávislého zdroje. [3, 6]

Průběh blízkého zkratu lze rozdělit podle doby od okamžiku zkratu do tří částí. První část, která začíná hned v okamžiku zkratu, se nazývá „rázová“, za ní následuje „přechodová“ a poté následuje ustálený stav v chodu nakrátko.

Jelikož se při zkratu vytvoří stejnosměrná a střídavá složka proudu, tak je patrné, že každá složka vytvoří ve stroji také příslušné magnetické pole. Střídavá složka vytvoří pole točící se synchronně s rotorem, kdežto stejnosměrná složka vytvoří v prostoru netočivé pole. Obě tyto pole působí na uzavřené části stroje, ve kterých se vybudí příslušný proud tak, aby pole od nich vyvolaná působila proti jejich smyslu. [1, 7]

1.4.1 Střídavá složka zkratu

Pro pole vyvolané střídavou složkou platí, že se hned po okamžiku zkratu uzavřou proudy v tlumícím i budícím vinutí a rázová reaktance X_d'' je proto podstatně menší než podélná synchronní reaktance. V dalším časovém okamžiku se zmenšují proudy tlumícím i budícím vinutím. Proud tlumícím vinutím klesá s menší časovou konstantou. Jak klesá proud tlumícím vinutím, tak okamžitá reaktance přechází velikostně přes přechodnou reaktanci X_d' až do velikosti synchronní podélné reaktance X_d , která odpovídá stavu, ve kterém se na nulu utlumily jak proudy tlumícím vinutím, tak vyvolané proudy budícím vinutím. [1, 8]



Obrázek 5: Změna podélné statorové reaktance stroje; a) rázová reaktance, b) přechodná reaktance, c) synchronní reaktance

Z náhradních schémat lze pak vyjádřit jednotlivé reaktance následovně:

$$X_d'' = X_{1\sigma} + \frac{1}{\frac{1}{X_{ad}} + \frac{1}{X_{b\sigma}} + \frac{1}{X_{t\sigma d}}}, \quad (1.1)$$

$$X_d' = X_{1\sigma} + \frac{1}{\frac{1}{X_{ad}} + \frac{1}{X_{b\sigma}}}, \quad (1.2)$$

$$X_d = X_{1\sigma} + X_{ad}. \quad (1.3)$$

Celkovou střídavou složku průběhu pak lze vyjádřit dle [1] jako součet rázové a přechodné složky a ustáleného stavu zkratového proudu následovně:

$$i_{st} = -\sqrt{2} \cdot U_i \cdot \left[\left(\frac{1}{X_d''} - \frac{1}{X_d'} \right) \cdot e^{-\frac{t}{T_d''}} + \right. \\ \left. + \left(\frac{1}{X_d'} - \frac{1}{X_d} \right) \cdot e^{-\frac{t}{T_d'}} + \frac{1}{X_d} \right] \cdot \cos(\omega t + \alpha_0). \quad (1.4)$$

1.4.2 Stejnoseměrná složka zkratu

V prvním okamžiku zkratu vzniká stejnosměrná složka takové velikosti, aby byla rovna okamžitá hodnota proudu před okamžikem zkratu a okamžitá hodnota proudu po vzniku zkratu. Tato složka je netočivá, a tudíž je u strojů s vyniklými póly výrazně ovlivňována okamžitým natočením rotoru a jeho příčnou a podélnou reaktancí. Tyto reaktance se vůči netočivému poli mění s dvojnásobnou frekvencí. [1]

Stejnoseměrnou složku lze vyjádřit dle [1] jako:

$$i_{ap} = \frac{\sqrt{2}}{2} \cdot U_i \cdot \left[\left(\frac{1}{X_d''} + \frac{1}{X_q''} \right) \cdot \cos(\alpha_0) + \right. \\ \left. + \left(\frac{1}{X_d''} - \frac{1}{X_q''} \right) \cdot \cos(2\omega t + \alpha_0) \right] \cdot e^{-\frac{t}{T_d''}}. \quad (1.5)$$

Pro stroje s hladkým rotorem, kde X_d'' je přibližně rovna X_q'' , pak lze zanedbat druhý člen rovnice a zjednodušit pouze na tvar:

$$i_k = \sqrt{2} \cdot U_i \cdot \frac{1}{X_d''} \cdot \cos(\alpha_0) \cdot e^{-\frac{t}{T_d''}}. \quad (1.6)$$

1.4.3 Celkový zkratový proud

Celkový průběh proudu při trojfázovém zkratu je dán aritmetickým součtem střídavé a stejnosměrné složky z rovnic 1-4 a 1-5 a pro stroj s vyniklými póly má tedy následující tvar:

$$\begin{aligned} i_k = i_{st} + i_{ap} = & -\sqrt{2} \cdot U_i \cdot \left[\left(\frac{1}{X_d''} - \frac{1}{X_d'} \right) \cdot e^{-\frac{t}{T_d''}} + \left(\frac{1}{X_d'} - \frac{1}{X_d} \right) \cdot e^{-\frac{t}{T_d}} + \frac{1}{X_d} \right] \cdot \\ & \cdot \cos(\omega t + \alpha_0) + \frac{\sqrt{2}}{2} \cdot U_i \cdot \left[\left(\frac{1}{X_d''} + \frac{1}{X_q''} \right) \cdot \cos(\alpha_0) + \right. \\ & \left. + \left(\frac{1}{X_d''} - \frac{1}{X_q''} \right) \cdot \cos(2\omega t + \alpha_0) \right] \cdot e^{-\frac{t}{T_d''}}. \end{aligned} \quad (1.7)$$

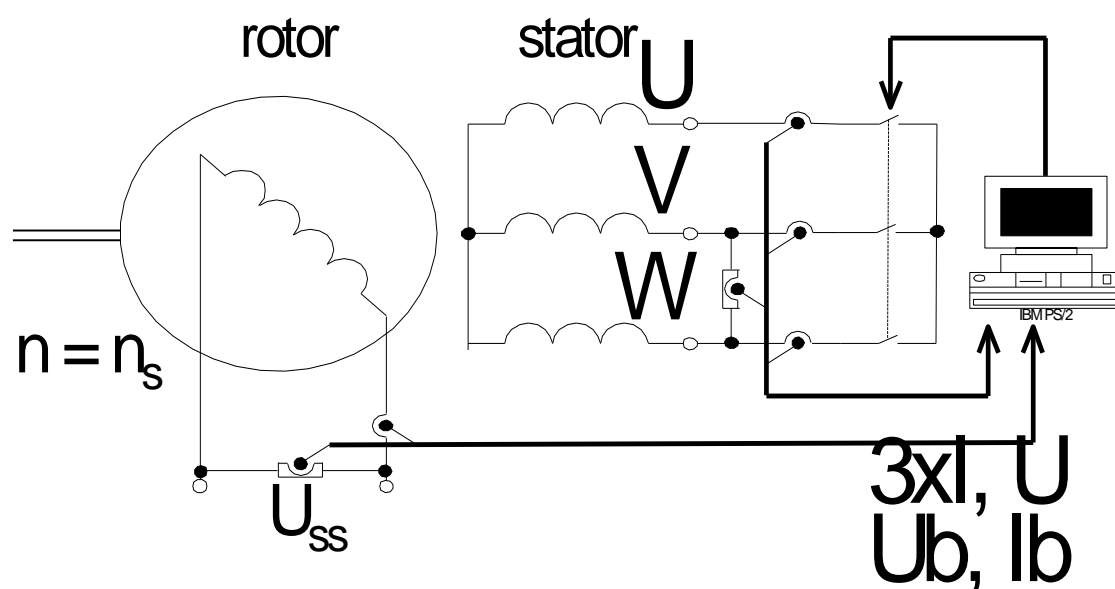
Tak jako je možné rozdělit zkratový proud i_k do složek na rázový zkratový proud, přechodný zkratový proud a ustálený zkratový proud, lze rozdělit také dobu trvání na tato období.

Dobu od začátku zkratu do $3T_d''$ je možné nazvat rázovým obdobím, dobu od $3T_d''$ do $3T_d'$ lze nazvat přechodným obdobím a dobu od $3T_d'$ do vypnutí zkratu lze nazvat ustáleným obdobím. [1]

1.5 Měření průběhů zkratových proudů

Abychom mohli zkratový proud vůbec vyhodnotit, je nutné jej nejdříve změřit. Měření zkratového proudu lze získat spoustu informací o stroji. Mezi tyto informace patří především rázová, přechodová a synchronní podélná reaktance a dále časové konstanty proudů daných složek.

Následující popis je určen pro stroj s vyniklými póly a buzením realizovaným pomocí baterie, nebo dynama s cizím buzením. Stroj je roztočen na synchronní rychlost. Je vhodné mít stroj nabuzen tak, aby ani předpokládaná hodnota proudu nepřekročila hodnotu jmenovitého proudu. Při měření je velmi vhodné spustit měření tak, aby stykač vyzkratoval stator až poté, co se začne zaznamenávat, aby byl s jistotou vidět nárůst proudu do maximální hodnoty. Měření je možné ukončit teprve tehdy, pokud odezněly veškeré přechodové děje a zkratový proud je ustálený, neboť s hodnotou ustáleného proudu se dále pracuje při vyhodnocování. [8]



Obrázek 6: Schéma zapojení pro záznam zkratového proudu [8]

1.5.1 Rozdělení průběhu na jednotlivé složky

Po změření se stanoví dvě obalové křivky amplitud průběhu. Tyto křivky začínají na maximální a minimální hodnotě první periody a spojují vrcholové hodnoty dalších period. Křivky nesmí být v těchto souřadnicích extrapolovány do okamžiku zkratu, protože pak by v tomto okamžiku vznikala chyba.

Horní obalovou křivku označíme i_1 a spodní jako i_2 .

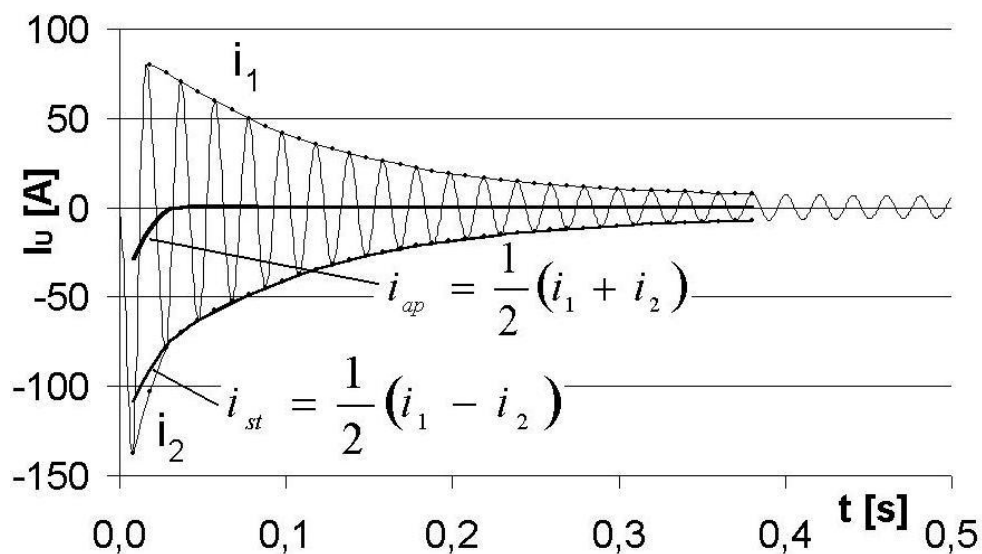
Z obalových křivek pak lze stanovit aperiodickou složku jako:

$$i_{ap} = \frac{1}{2}(i_1 + i_2) \quad (1.8)$$

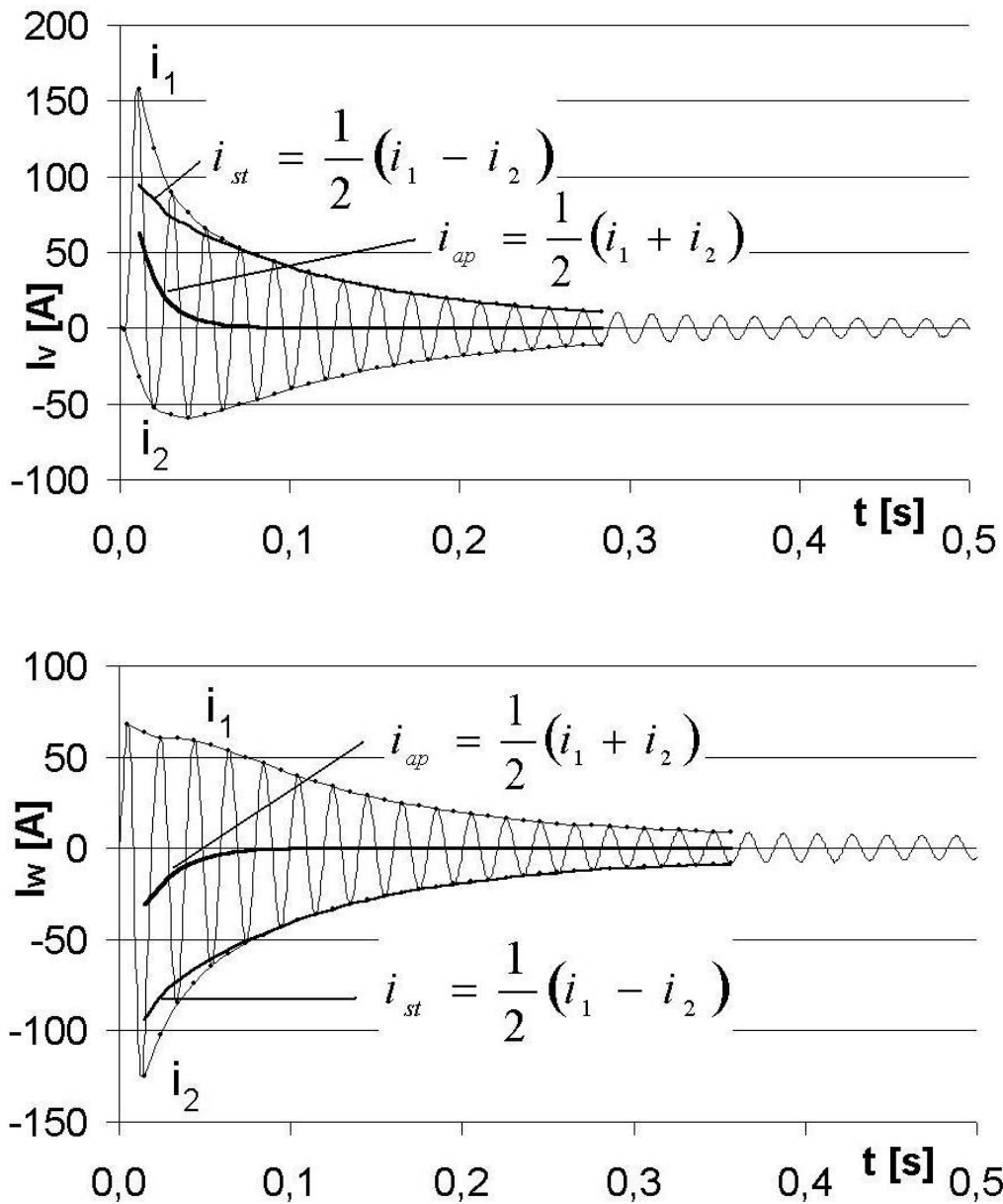
a střídavou složku jako:

$$i_{per} = \frac{1}{2}(i_1 - i_2) \quad (1.9)$$

Průběhy s obalovými křivkami a rozdělenými složkami pak lze pro každou fázi vykreslit do grafu (obrázky 7 a 8).



Obrázek 7: Průběh proudu, obalových křivek, aperiodické a střídavé složky pro fázi U [8]

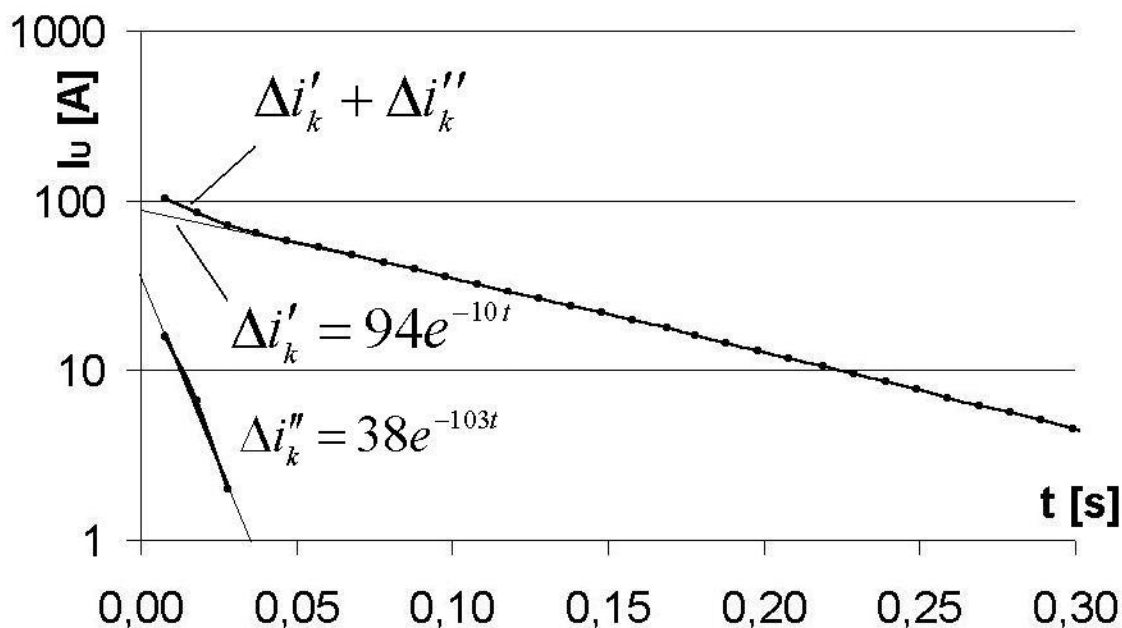


Obrázek 8: Průběh proudu, obalových křivek, aperiodické a střídavé složky pro fáze V a W [8]

V následující části se od periodické části každého průběhu odečte příslušný průběh ustáleného zkratového proudu a získá se tím součet přechodové a rázové složky. Při nestejných hodnotách ustáleného zkratového proudu uvažujeme jejich střední hodnotu.

$$i_k'' + i_k' = i_{per} - I_k = \frac{1}{2}(i_1 - i_2) - I_k \quad (1.10)$$

Dále se průběh tohoto součtu vyneseme do grafu s logaritmickou osou velikosti proudu a lineární osou času (obrázek 9).



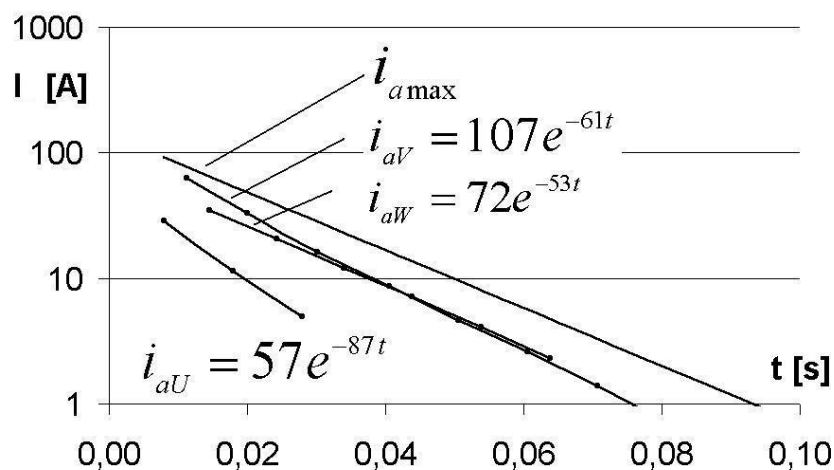
Obrázek 9: Rázová, přechodová složka a jejich součet [8]

Jak je patrné z výše uvedeného grafu, průběh součtu složek má od určitého okamžiku lineární průběh. Tuto lineární část je možné extrapolovat do počátku a oddělit tak rázovou složku i''_k (rozdíl součtového průběhu a extrapolovaného) a přechodovou složku i'_k (extrapolovaný průběh). Přímkou rázové složky lze také extrapolovat do počátku, čímž získáme druhou fiktivní hodnotou proudu, tentokrát pro rázovou složku.

Z těchto průběhů pak lze stanovit velikosti amplitud po půlperiodě a zároveň časové konstanty obou složek, které zjistíme tak, že určíme okamžik, kdy daná složka klesne na

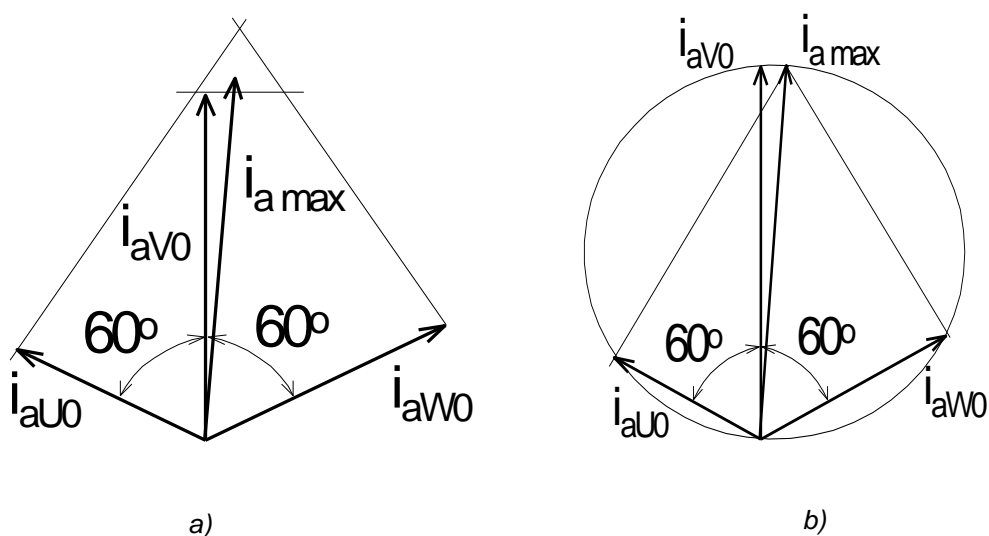
$$\frac{1}{e} = 0,368. \quad (1.11)$$

Dále v těchto semilogaritmických souřadnicích vyneseme průběhy aperiodických složek všech fází do společného grafu. Bude se jednat o tři přímky, u nichž určíme počáteční hodnotu v okamžiku zkratu a příslušné časové konstanty. Z časových konstant dále počítáme s tou nejdéle trvající a tedy nejnepříznivější.



Obrázek 10: Aperiodické složky všech fází [8]

Z hodnot v okamžiku zkratu lze také získat velikost maximální možné velikosti aperiodické složky. A to tak, že umístíme vektor maximální hodnoty a s úhlem 60° na každou stranu fázor zbylých dvou hodnot. Z konců všech fázorů následně spustíme kolmice, které se v ideálním případě protnou v jednom bodě. Pokud se v jenom bodě neprotnou, bereme jako průsečík těžiště vzniklého trojúhelníku. Všechny vektory by teoreticky měly ležet na společné kružnici, jejímž středem je právě poloviční vektor $i_{ap\ max}$. Průběh této hodnoty zakreslíme do předchozího grafu, kde bude se sklonem oné nejnepríznivější časové konstanty. Pak se opět určí hodnota po půlperiodě $i_{ap\frac{1}{2}}$.

Obrázek 11: Konstrukce $i_{ap\ max}$. a) v praxi b) ideálně [8]

Reaktance je pak možné určit následovně:

$$X_d = \frac{U\sqrt{2}}{\sqrt{3} I_k}, \quad (1.12)$$

$$X'_d = \frac{U\sqrt{2}}{\sqrt{3} (I_k + i'_{k0})}, \quad (1.13)$$

$$X''_d = \frac{U\sqrt{2}}{\sqrt{3} (I_k + i'_{k0} + i''_{k0})}. \quad (1.14)$$

Pokud zkouška probíhala za sníženého napětí, je nutné brát v úvahu, že tyto reaktance jsou závislé na aktuálním sycení obvodu stroje a jsou tedy větší než při jmenovitém napětí.

Co se týče proudu, tak pokud uvážíme nejnepriznivější moment zkratu, tj. ten, když zkrat nastane při nulovém napětí a maximálním okamžitým proudem, tak jeho maximální možná velikost nastává půl periody po vzniku zkratu, tedy v čase $t = 0,01$ s pro 50 Hz. Proto pro maximální okamžitou hodnotu platí:

$$i_{k \max} = I_k + i'_{k\frac{1}{2}} + i''_{k\frac{1}{2}} + i_{ap \max} \frac{1}{2} \quad (1.15)$$

Je zde však, stejně jako u reaktancí, nutno počítat s tím, že pokud bylo napětí při zkoušce snížené, tak jsou reaktance větší a tedy $i_{k \max}$ je menší.

2 Analýza zkratového proudu

V této části se budu zabývat výhradně praktickou částí práce. Hlavním úkolem práce je zanalyzovat dodaný průběh zkratového proudu. Analýzu je nutno řešit pomocí skriptu v programu MATLAB.

MATLAB je vysokoúrovňový programovací jazyk vyvinutý společností MathWorks, který je často využíván k různým vědeckým výpočtům a je udělán tak, aby byl uživatelsky přívětivý i k lidem, kteří se nezabývají výhradně programováním. Obsahuje spoustu předpřipravených knihovních funkcí dostupných k využití, nástrojů pro vizualizaci spočítaných dat, simulace a další. MATLAB je rozdělen do základních částí, tzv. toolboxů, které jsou pak dostupné pod samostatnými licencemi a obsahují dodatečné funkce, které jsou daleko více specifikovány na konkrétní problémy.

Spolu s programovacím jazykem MATLAB vyvíjí MathWorks dodatečný nástroj Simulink, který hlavnímu programu sekunduje. Simulink, jak již název napovídá, je určený k simulaci fyzikálních modelů, ale může být pomocí něj například řízen robot, či prováděno zpracování obrazu a jiné.

Celá analýza je založená na postupném rozboru průběhu Fourierovou transformací tak, aby šlo průběh rozložit na jednotlivé složky, tj. 50Hz střídavou složku, 100Hz složku (která se, jak je výše popsáno, vyskytuje u strojů s vyniklými póly) a na stejnosměrnou složku. Skript je navržen tak, aby ale bylo možné analyzovat i stroje pracující na 60Hz síti, a proto v něm lze zadat základní frekvenci. V daném případě pak rozklad probíhá na aperiodickou složku, 60Hz složku a 120Hz složku.

Vzhledem k tomu, že je celý program založen na Fourierově transformaci, budu nejdříve popisovat samotnou transformaci a až za ní bude následovat popis zbylého programu.

2.1 Fourierova transformace

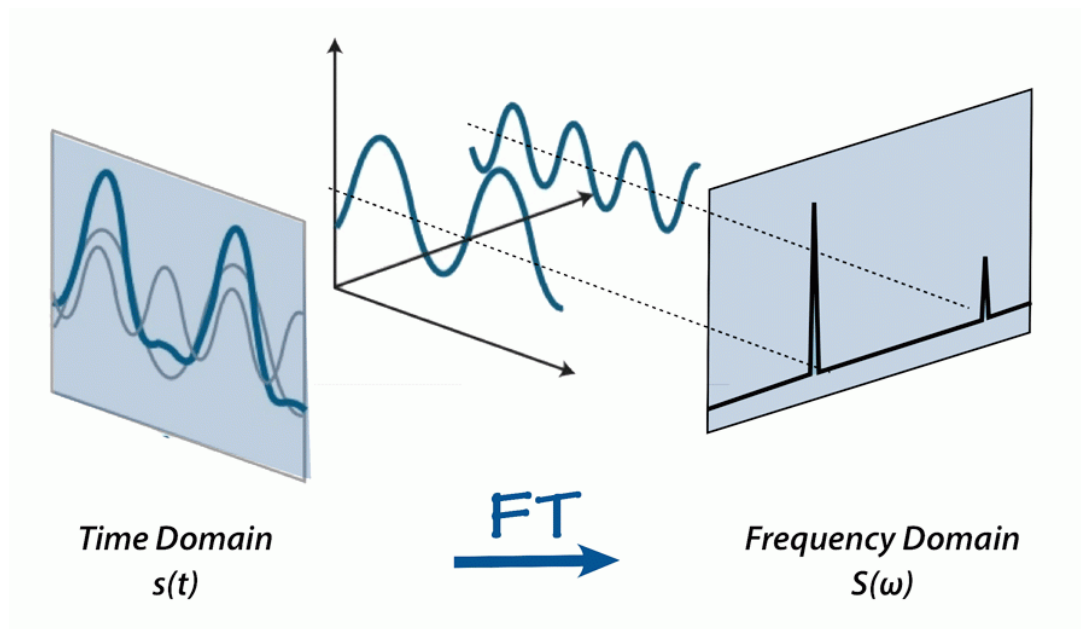
Jedná se o matematickou operaci, jejímž hlavním úkolem je rozložit průběh funkce závislé na čase na funkci závislou na frekvenci. Výsledkem je pak soubor komplexních exponenciál, z nichž každou lze přepsat pomocí Eulerova vzorce (2.1) na funkce sinus a cosinus.

$$e^{ix} = \cos(x) + i \cdot \sin(x) \quad (2.1)$$

Jako opak této transformace lze provádět operaci zvanou Inverzní Fourierova transformace, která, jak je patrné z názvu, převádí funkci závislou na frekvenci na funkci závislou na čase.

Fourierova transformace nachází své využití ve velmi širokém spektru oblastí. Lze ji využít například k obecnému popisu signálů, přenosu dat, zpracování zvuku, fotografií, videí, nebo odhalení rezonančních frekvencí mechanických součástí.

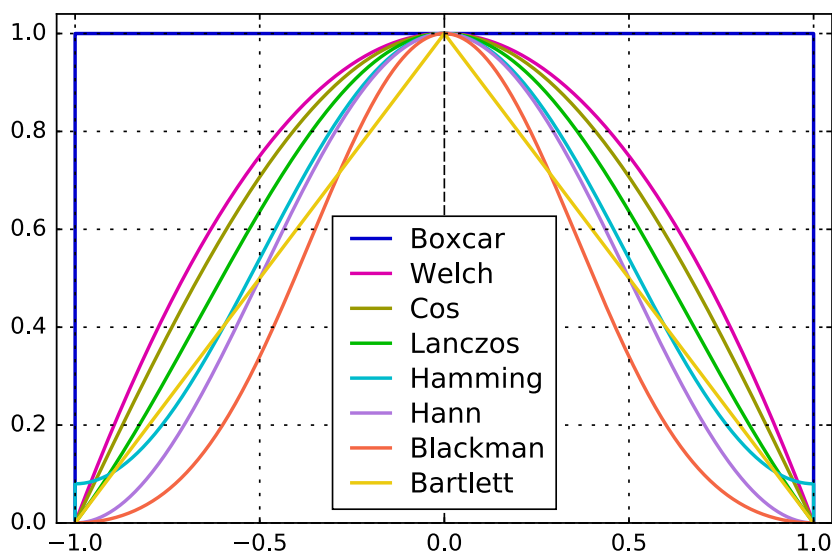
Jako vstupní signál pro transformaci lze mít data spojitá, nebo diskrétní. Diskrétní data jsou pak taková data, která vznikla vzorkováním zdrojového signálu a každý vzorek má tedy svůj přesný časový okamžik, ve kterém byl změřen. Dále lze vstupní signály dělit na periodické a aperiodické. [11,13]



Obrázek 12: Grafické vyjádření funkce Fourierovy transformace [12]

Existuje více typů této transformace a mezi nejznámější se řadí Rychlá Fourierova transformace (FFT – Fast Fourier transform). Vstupní signál v tomto případě musí být diskrétní a periodický. Vzhledem k tomu, že analyzované průběhy jsou často neperiodické, je nutné provést jejich periodizaci. Ta se provádí tak, že se určí časový výřez signálu, který je následně analyzován. V tomto případě ale může nastat případ, kdy se počáteční hodnota nerovná koncové hodnotě a dochází tedy ke skoku a tím ke vzniku nežádoucích harmonických složek, které ale v původním signálu obsaženy nebyly.

Tyto skoky lze minimalizovat použitím některé z váhových funkcí. Vstupní okno signálu je touto funkcí vynásobeno a analyzuje se výsledek. V tomto případě je ale nutné brát v úvahu, že spektra jednotlivých váhových funkcí se projeví i ve výsledném spektru, které je tím zkresleno. Jak je vidět z obrázku 13, tak váhové funkce si zachovávají hodnoty blízké jedné ve středu analyzovaného okna, a naopak potlačují krajní hodnoty. [11,13]



Obrázek 13: Nejvíce používané váhové funkce [12]

2.1.1 FFT jako funkce MATLABu

V programu MATLAB je dostupná Rychlá Fourierova transformace, které je přístupná přes funkci `fft` se základní syntaxí $Y = \text{fft}(X, n, \text{dim})$. Nutným vstupním parametrem jsou vstupní data X . Ty mohou být funkci předložena jako vektor, nebo jako matice s tím, že v maticovém tvaru se `fft` chová tak, že každý sloupec považuje za rozdílný průběh. Volitelným parametrem

n upravujeme délku transformace, která musí být rovna kladnému celému číslu. Typicky se může jednat o mocniny čísla 2. Parametrem *dim* pak můžeme, v případě že jsou vstupní data zadána jako matice, upravit směr rozkladu, kdy 1 odpovídá výchozímu směru po sloupcích a 2 odpovídá rozkladu po řádcích.

Výstupními daty funkce je poté vektor komplexních čísel, který odpovídá dvoustrannému spektru vstupu. Tento vektor je před dalším použitím nutno vydělit délkou transformace, abychom dostali konečné dvoustranné spektrum. Na jednostranné amplitudové spektrum signálu je možné se potom dostat tak, že vezmeme absolutní hodnotu spodní poloviny a vynásobíme dvěma, neboť dvoustranné spektrum má oproti jednostrannému poloviční amplitudy. Stejnoseměrnou složku, která leží na první pozici, není třeba násobit dvěma.

K výstupnímu vektoru pak lze také přiřadit vektor frekvencí, aby bylo možné spektrum zobrazit v grafu. Tento vektor frekvencí lze spočítat jako vzorkovací frekvenci dělenou délkou transformace a vynásobenou vektorem od nuly do poloviny délky transformace. [14,15]

2.2 Popis vypracovaného skriptu

V následujících odstavcích se nachází již popis samotného programu, jež je praktickou částí k mé práci. Skript zde budu popisovat postupně tak, jak jdou jednotlivé bloky za sebou, a v každé části bude vysvětlen princip daného bloku skriptu a problémy, s nimiž jsem se v dané chvíli potýkal.

2.2.1 Nastavení základních proměnných skriptu

Skript jako takový obsahuje několik proměnných, jež ovlivňují následné zpracování průběhu. Definice těchto proměnných je uvedena hned na začátku tak, aby nebylo nutné je hledat mezi řádky a byly snadno dostupné.

Proměnná *plotnout* určuje, jestli se na konci zpracování bude zobrazovat změřený zkratový proud spolu s jednotlivými zjištěnými složkami, kdy každá složka je zobrazena v odděleném grafu.

Proměnná *plotnout_celkovy* nechává uživatele zvolit, jestli chce nechat vytvořit samostatné okno se změřeným proudem a do stejného grafu zobrazit součet všech složek. Toto zobrazení může sloužit jako jakási vizuální kontrola toho, jestli bylo vše správně spočteno.

Jako další volba je zde proměnná *ulozit*, která stejně jako obě výše zmíněné může nabývat hodnot 0 a 1, kdy 0 znamená, že se daný úsek neprovede, a 1, že se úsek programu má provést. Tato volba umožňuje zvolit, jestli se má vektor zjištěných složek, příslušného časového vektoru a jednotlivé časové konstanty uložit do samostatného *.mat souboru.

Dalším požadavkem, tentokrát určujícím charakter vstupních dat, je nastavení proměnné s názvem *frekvence*. Tato proměnná určuje jednotlivé harmonické složky, které se budou zjišťovat z Fourierovy transformace a zároveň také frekvenci, s jakou se budou zjištěné amplitudy složek prokládat danou harmonickou funkcí (v tomto případě cosinus). Typicky se bude jednat o frekvence odpovídající 50 Hz a 60 Hz, nicméně je možnost zvolit i jinou frekvenci. Frekvence uložená v této proměnné by samozřejmě šla spočítat jako převrácená hodnota rozdílu času jednotlivých maxim průběhu, nicméně si myslím, že bude jednodušší, když uživatel specifikuje frekvenci vstupního průběhu, neboť při měření může docházet ke změnám otáček.

Poslední ze vstupních proměnných, které je nutné zadat před zpracováním, je proměnná *t_zk*, která specifikuje, kdy ve zpracovávaném průběhu dochází ke zkratu. Podle této proměnné pak dochází k výpočtu fiktivních hodnot rázové a přechodové složky v okamžiku zkratu, které normálně není možné nijak změřit. Pokud tento čas uživatel neví, tak je dobré zanalyzovat průběh, s nastavením proměnné rovné nule, z grafu odečíst příslušný čas, zadat jej do skriptu a průběh zanalyzovat znovu.

2.2.2 Vstupní data a zjištění jejich základních údajů

Vstupní data, která mi byla poskytnuta, jsou uložena jako *.mat soubor, který obsahuje dva vektory. První vektor obsahuje časy vzorků, kdy se provádělo měření a druhý vektor obsahuje samotné změřené okamžité hodnoty proudu. Nejedná se tedy o trojfázový průběh, ale o průběh zkratového proudu pouze v jedné fázi.

K načítání průběhů jsem využil funkce *uiopen*, která otevře dialogové okno, v němž je možné vybrat vstupní soubor. U této funkce jsem zvolil u parametru *type* hodnotu *load*, která

dovoluje načtení pouze souborů typu *.mat. Dále lze tento parametr nastavit do *matlab*, jež dovoluje načíst *.m soubory, *figure* pro načtení obrazových souborů a *editor* pro otevírání v editoru upravitelných souborů. Dále lze také zvolit výchozí jméno načítaného souboru, ale této možnosti jsem nevyužil, neboť názvy v budoucnu analyzovaných souborů neznám. [17]

Ihned po načtení dat dochází k prověření, že vstupní data jsou v pracovním prostoru MATLABu. Uživatel totiž může v načítacím dialogu kliknout na storno, čímž se pochopitelně žádá data nenačtou a program by zbytečně vypisoval chyby o chybějících proměnných. V souboru s daty také mohou být vstupní vektory uloženy pod jinými názvy, a to by vedlo k těm samým chybám. Tento úsek proto řeší, že pokud nejsou v pracovním prostoru vektory I_k a t , dojde k vypsání chybové hlášky, vyčištění proměnných a ukončení skriptu.

V další části dochází k nalezení nejbližšího bodu ve vektoru času k tomu, co zadal uživatel na začátku programu pod proměnnou t_{zk} a jedná se o čas, kdy ve vstupních datech dochází ke zkratu. Dochází zde k oříznutí všech nepotřebných vstupních dat, jež se nachází před okamžikem zkratu. Vzhledem k tomu, že ze vstupních dat lze odebrat pouze celočíselný počet vzorků od začátku průběhu, dochází k nalezení indexu ve vektoru času, při kterém je odchylka od zadaného času nejmenší.

2.2.3 Vzorkovací frekvence

Na následujícím řádku dochází k výpočtu průměrné vzorkovací frekvence, jakožto převrácené hodnoty průměru rozdílu sousedních prvků vektoru času. Tento řádek také definuje jednu z podmínek, kdy skript může selhat. Je totiž nutné, aby vzorkovací frekvence byla v průběhu měření pokud možno konstantní a s co nejmenší odchylkou od průměru. Z této vzorkovací frekvence se totiž v průběhu skriptu dopočítává délka Fourierovy transformace pro každé analyzované časové okno a pokud by vzorkovací frekvence v průběhu měření kolísala, skript by nespočítal správně požadované harmonické složky v těch periodách, kde by se aktuální vzorkovací frekvence velmi lišila od průměrné. Pro mnou analyzovaný průběh se jednalo o vzorkovací frekvenci 100 kHz, která se od této frekvence odchylovala maximálně v řádu 10^{-6} Hz, tudíž jsem ji mohl považovat zcela jistě za konstantní po celou dobu měření. Ve skriptu je pak výpočet řešen na 25. řádce následovně: $F_s = 1/\text{mean}(\text{diff}(t))$.

Po zjištění vzorkovací frekvence dochází k ujištění, že vektory I_k a t jsou sloupcové vektory. Pokud se o sloupcové vektory nejedná, dochází k jejich transpozici. Tuto část bylo nutné přidat, neboť v průběhu zpracování se zde pracuje i s jednotlivými vzorky a bylo tedy nutné se ujistit, že vstupní data budou vždy ve stejném formátu.

2.2.4 Postup při záporné aperiodické složce

Následně se zjišťuje, jestli dodaný průběh má kladnou0 či zápornou aperiodickou složku. Když jsem tento skript vypracovával, tak pro kladnou aperiodickou složku pracoval relativně v pořádku, nicméně pokud jsem zkusil analyzovat průběh se zápornou aperiodickou složkou (například po vynásobení -1), tak docházelo ke špatnému časovému zasazení fázového posunu jednotlivých složek. Ani po vícero pokusech jsem však nemohl přijít na to, do jakého časového okamžiku zasadit tyto získané hodnoty. Proto jsem zde zavedl to, že když je minimum průběhu větší v absolutní hodnotě než jeho maximum, tak se jedná o zápornou aperiodickou složku. V tomto případě tedy dochází k vynásobení vstupních dat -1 a jejich analýze, jako kdyby se jednalo o kladnou aperiodickou složku. Na konci zpracování, pokud došlo k otočení na začátku, pak dochází k ještě jednomu otočení tak, aby výsledek odpovídal správné polaritě aperiodické složky.

2.2.5 Vytvoření časových oken

Pro správnou funkci Fourierovy transformace je nutné zajistit, aby zpracovávala jen určité časové oblasti průběhu. Jak je napsáno v kapitole 2.1, tak je pro zajištění minimálních vyšších harmonických složek nejlepší, když se signál analyzuje po úsecích, které jdou snadno převést na periodické. Tomu nejlépe odpovídá analyzování po celých periodách. Z tohoto důvodu jsem zvolil, že analyzované časové úseky budou mezi jednotlivými dílčími maximy průběhu.

Pro zjištění vektoru maxim je zde použita funkce *findpeaks*, která je součástí Signal Processing Toolboxu. Tato funkce se používá pro obecné signály k nalezení lokálních maxim a obsahuje spoustu parametrů, jimiž lze upravovat její detekci. Dá se například nastavovat minimální a maximální velikost maxima, minimální rozdíl ve velikosti oproti sousedním maximům, nebo například minimální rozdíl v ose x , jehož jsem využil i ve skriptu.

Tento parametr je funkci zadán jako parametr *MinPeakDistance* a určuje minimální vzdálenost dvou maxim. Jedná se o parametr vyjadřující index zpracovávaných dat a jeho velikost jsem nastavil na 80 % periody základní frekvence. Tento parametr bylo nutné přidat v okamžiku, kdy jsem si vygeneroval několik sad testovacích dat a v některých byla 100Hz složka tak velká, že se začala vytvářet lokální maxima i mimo základní 50Hz složku.

Výstupními parametry funkce *findpeaks* je pak vektor hodnot maxim a vektor jejich indexů ve vstupních datech. [18]

Ve skriptu tato funkce analyzuje až data po maximální hodnotě, kterou je pak nutno do výsledného vektoru přidat. Takto je to z toho důvodu, aby když uživatel nezná okamžik zkratu a zadá $t_{zk}=0$, tak aby funkce nedetekovala i maxima před zkratem.

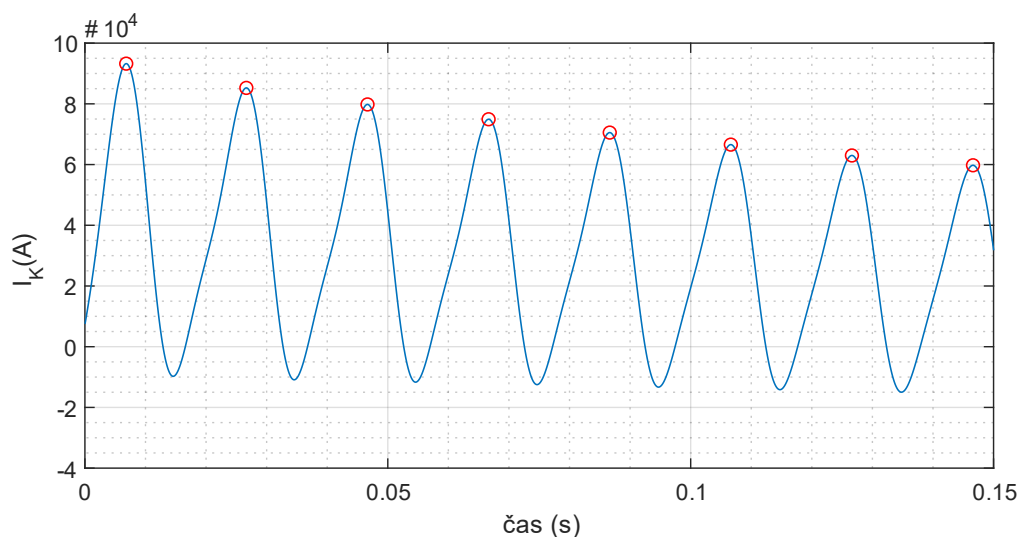
Tento výpočet je pak realizován následovně:

```

47 [pks_max_fp,locs_max_fp]=findpeaks(i_mes(i_mes_max_id:end),'MinPeakDistance',tol*Fs);
49 locs_max_fp=locs_max_fp+i_mes_max_id-1;
50 locs_max=[i_mes_max_id; locs_max_fp];
51 pks_max=[i_mes_max; pks_max_fp];

```

Tuto funkci by šlo velmi snadno nahradit funkcí *islocalmax*, která nenáleží žádnému toolboxu. Nicméně ta je v MATLABu obsažena až od verze R2017b, a tak jsem se rozhodl ji nevyužít z důvodu možnosti použití skriptu na starších verzích programu.



Obrázek 14: Výsledek funkce *findpeaks* se zobrazením změřeného zkratového proudu

2.2.6 Rozbor pomocí FFT

Po tom, co se vytvoří jednotlivá časová okna, lze jednotlivé periody zanalyzovat pomocí FFT. Nejdříve dochází k vytvoření vektorů jednotlivých složek tak, aby do nich následně mohly být uloženy hodnoty. Následně se ve for cyklu analyzují jednotlivé periody. Dochází zde k vytvoření vektoru proudu a času, jež se bude analyzovat, a k volání funkce *rozborFFT*, jež je umístěna na konci skriptu. Samotné volání funkce se provede, s uložením všech výstupních proměnných, takto:

```
75 [aper(index-1),amp50(index-1),amp100(index-1),...  
76 ps50(index-1), ps100(index-1)]=...  
77 rozborFFT(i_analyz,t_analyz,Fs,frekvence);
```

Výstupem funkce *rozborFFT* jsou amplitudy a fázové posuny pro aperiodickou, 50Hz a 100Hz složku průběhu.

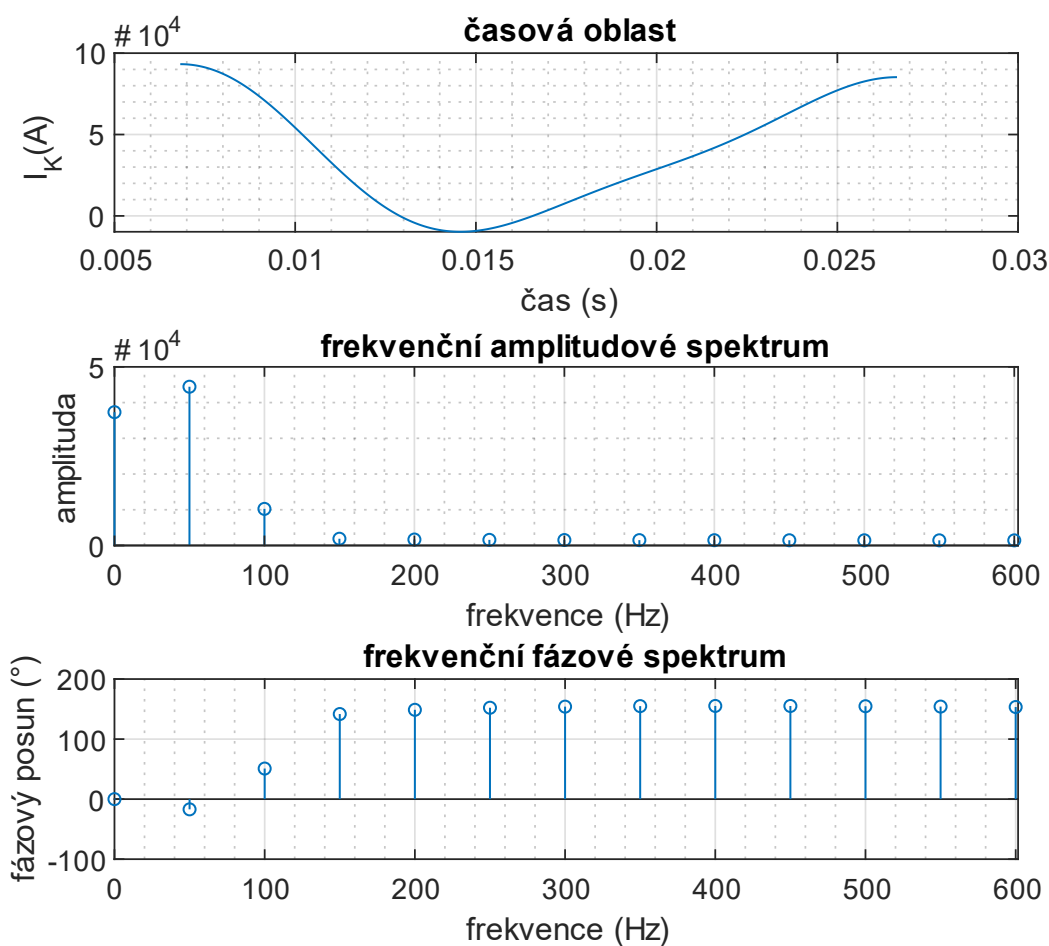
Funkce samotná pak zajišťuje výpočet délky transformace, transformaci samotnou a zpracování výstupu.

Délku transformace lze zvolit, což nám umožňuje zajistit, aby na výstupu Fourierovy transformace byly jednotlivé harmonické složky rozděleny s přesně definovaným rozestupem. V našem případě chceme, aby jednotlivé harmonické měly rozestup daný proměnnou *frekvence*. To lze zajistit tak, že délku transformace zvolíme jako vzorkovací frekvenci vydělenou základní harmonickou (v našem případě 50, nebo 60 Hz). Délku transformace je nutno volit jako sudé číslo, neboť při dalším zpracování dochází k převodu dvoustranného spektra na jednostranné a v případě lichého čísla délky transformace bychom dostali i lichý počet výsledků dvoustranného spektra. To by vedlo k tomu, že by docházelo k neceločíselným indexům prvků výsledku. Délka transformace je pak počítána na řádku 266: $L=2.*\text{round}(Fs/(2.*\text{frekvence}))$.

Výsledek FFT je pak vektorem komplexních čísel, ze kterých lze spočítat jejich absolutní velikost a fázové natočení. V případě velikosti je nutné je vydělit délkou transformace, následně vzít spodní polovinu spektra a tu vynásobit dvěma (od indexu 2 výše). V případě fázového posunu je nutné pouze počítat s tím, že v první polovině je fázový posun záporný a v druhé polovině kladný. [19]

Následně, pokud je správná délka transformace, je ve výsledném vektoru na první pozici stejnosměrná (aperiodická) složka, a na druhé a třetí pozici pak složky odpovídající první a druhé harmonické složce.

Funkce je pak doplněna o možnost zobrazení výsledného spektra. Toto zobrazení lze zapnout nastavením proměnné `plotnout_fft` rovno jedné. Nicméně je vhodné brát v úvahu fakt, že se spektrum zobrazí pro každou periodu v samostatném okně a výsledné zobrazení tedy může být dosti náročné pro počítač, na kterém je analýza prováděna. Přepočítání vektoru frekvence je k těmto grafům prováděn pouze, pokud je zapnut grafický výstup.



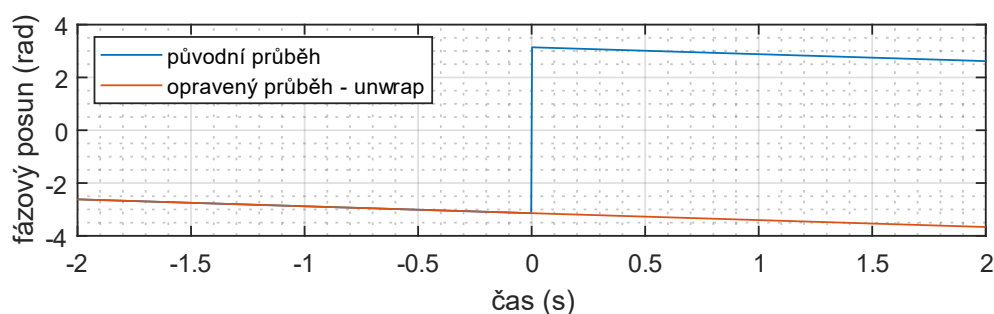
Obrázek 15: Grafický výstup funkce `rozborFFT` aplikované na první periodu dodaného průběhu

V předchozím kroku jsme tedy získali amplitudy a fázové posuny pro jednotlivé složky. Tyto hodnoty jsou závislé na čase a pro každou periodu je hodnota jiná. Jako další krok se

tedy provede proložení všech získaných hodnot tak, aby bylo možné získat časové průběhy o stejné vzorkovací frekvenci jako má zdrojový průběh.

2.2.7 Oprava fázového posunu

Ještě před samotným proložení všech hodnot je nutné provést korekci frekvenčních průběhů. Funkce FFT totiž vrací komplexní čísla, na ta je následně aplikována funkce *angle* pro zjištění fázového úhlu. Tato funkce vrací úhel v rozsahu $-\pi$ až π . Vzhledem k tomu, že se tento úhel může v čase měnit, je nutno brát v úvahu, že, než ho proložíme, musíme opravit přeskoky, které mohly nastat při změnách úhlu z $-\pi$ na π (malá fyzická změna, ale výpočetně změna o necelých 2π). Pro tento účel se skvěle hodí vestavěná funkce *unwrap*, která tyto přeskoky opraví. Vstupním parametrem této funkce je vektor fázového posunu dané složky a volitelný parametr, kterým lze změnit hodnotu, při které se opravuje. Tato hodnota se volí jako polovina maximálního rozdílu a ve výchozím nastavení je rovna π . Pokud tedy vstupní vektor udělá mezi dvěma sousedními hodnotami skok větší než π , dochází pak na následujících hodnotách k odečtení, nebo naopak přičtení celočíselného násobku 2π . [20]



Obrázek 16: Ukázka funkce *unwrap* (nejedná se o analyzovaný průběh)

2.2.8 Proložení

K samotnému proložení amplitud je ve skriptu využita funkce *fit*. Tato funkce je součástí Curve Fitting Toolboxu a používá se k obecnému proložení hodnot zvolenou funkcí. Vstupem je zde vektor hodnot osy x (časy), osy y (amplitudy získané z FFT) a dále parametr zvaný *fittype*, kterým je možné zvolit si rovnici, jímž se data mají prokládat.

Složka 50 Hz se skládá z rázové a přechodové složky. Proto od amplitud této složky je v prvním kroku odečten poslední prvek. Pokud je proud na konci měření ustálený, pak se

tento prvek rovná velikosti amplitudy ustáleného zkratového proudu. Tímto odečtením tak získáme součet rázové a přechodové složky, který pak lze proložit. U tohoto průběhu jsem zvolil proložení s parametrem *exp2*, což odpovídá rovnici $y = a \cdot e^{bx} + c \cdot e^{dx}$. Jedná se tedy o součet dvou exponenciál, k jejímž konstantám *a* až *d* lze přistupovat. Parametry *a* a *c* pak odpovídají amplitudám pro $x = 0$ (okamžik zkratu při správně nastaveném *t_zk*) a konstanty *b* a *c* odpovídají zápornému násobku převrácených hodnot časových konstant. Tyto časové konstanty jsou pak v programu porovnány a větší z nich se bere jako přechodová konstanta T'_d a menší jako rázová časová konstanta T''_d . Podle toho, která konstanta náleží které exponenciále, je pak rozhodnuto i o příslušnosti amplitud k té dané složce. Proložení samotné pak probíhá takto:

```
102 razprech_fit=fit(fft_locs(1:end-1),razprech(1:end-1),'exp2');
105 konstanta1=-1./razprech_fit.b;
106 konstanta2=-1./razprech_fit.d;
107 [td_razova,index_raz]=min([konstanta1, konstanta2]);
108 td_prechodna=max([konstanta1, konstanta2]);
```

Podobné proložení je pak provedeno i pro 100Hz složku a aperiodickou složku. V těchto dvou případech je ale proložení jinou rovnicí. Konkrétně se jedná o typ *exp1*, který odpovídá rovnici $y = a \cdot e^{bx}$. Podobně jako v předchozím případě jsou pak z rovnic zjištěny jednotlivé časové konstanty a amplitudy. [21]

Vzhledem k tomu, že výstupem funkce *fit* je symbolická funkce (nejedná se o vektor hodnot, ale čistě jen o funkční předpis), je nutné pro každou výstupní funkci spočítat hodnoty. Výpočet hodnot se pak provádí funkcí *feval*, která má jako vstupní parametry funkci, pro níž chceme spočítat hodnoty, a rozsah hodnot, v jakých chceme výpočet provést. Jako rozsah počítaných hodnot je zde volen vektor časů mezi absolutním maximem průběhu a koncovým časem. Výstupem je pak vektor hodnot, pro každý prvek časového vektoru. Jako příklad výpočtu hodnot lze uvést příkaz na řádce 119, který provádí výpočet pro součet rázové a přechodové složky: `razprech_eval=feval(razprech_fit,t(i_mes_max_id:end))`. [22]

Fázové posuny zjištěné pomocí FFT jsou pak proloženy funkcí *spline*, která provede výpočet kubické spline pro dané hodnoty. Spline jsem zvolil proto, že dopředu nevím, jaký typ průběhu očekávat. Tato funkce pak má jako vstupní data hodnoty fázových posunů, jejich

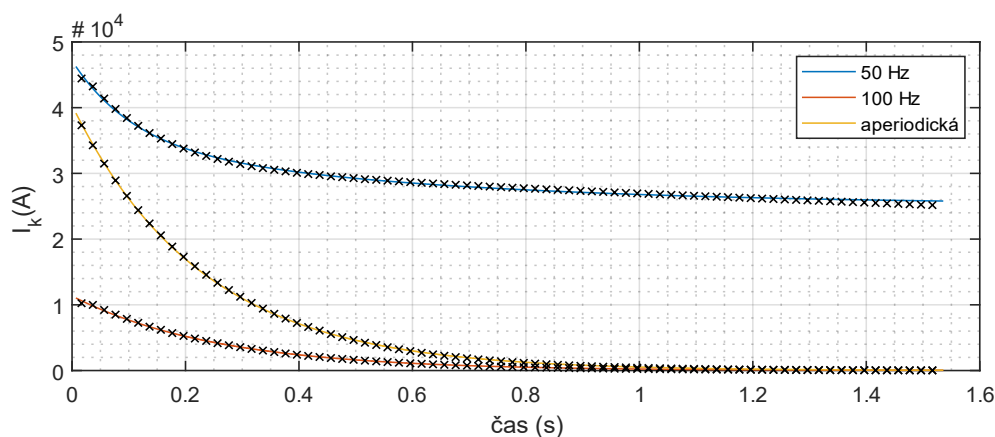
časové umístění a vektor časů, pro který se spline počítá. Výstupem je pak vektor proložených hodnot. [23]

2.2.8.1 Zjištěné informace o dodaných datech

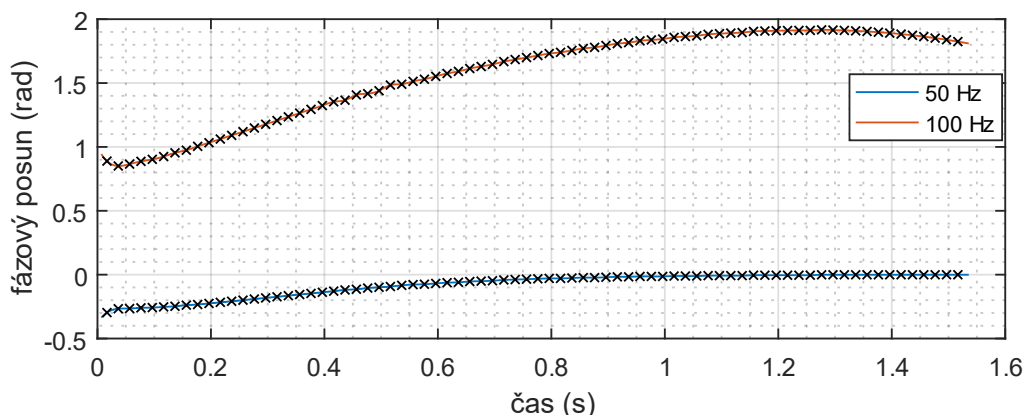
Pokud analýzu provedeme až do tohoto bodu, lze z dodaného průběhu zkratového proudu získat informaci o jeho složkách a základních vlastnostech. Pro dodaný průběh jsou pak získané informace následující:

	Časové konstanty:	Amplitudy v okamžiku zkratu
Rázová složka	$T_d'' = 104,82 \text{ ms}$	9,724 kA
Přechodová složka	$T_d' = 556,90 \text{ ms}$	12,252 kA
100Hz složka	$T = 256,72 \text{ ms}$	11,339 kA
Aperiodická složka	$T_a'' = 230,47 \text{ ms}$	40,348 kA

Velikost amplitudy ustáleného zkratového průběhu je 25,175 kA. U všech doposud uváděných amplitud se jedná o maximální hodnoty, a ne o hodnoty efektivní.



Obrázek 17: Proložení amplitud s vyznačenými prokládanými body



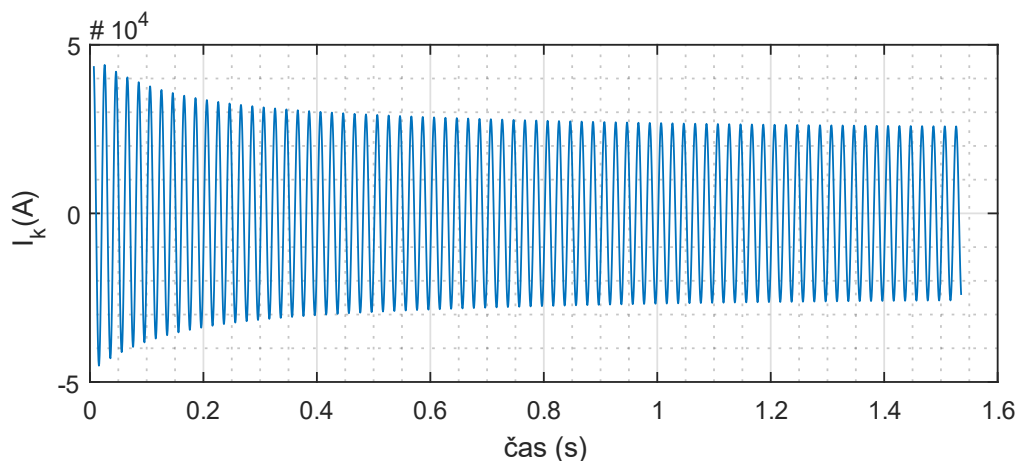
Obrázek 18: Proložení fázového posunu složek s vyznačenými prokládanými body

2.2.9 Zpětná rekonstrukce složek a celkového průběhu

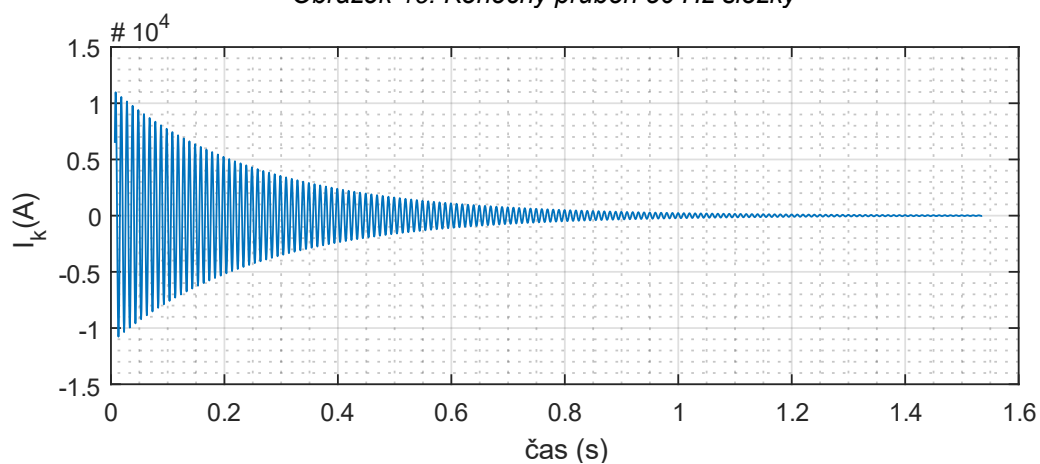
V tomto okamžiku o průběhu víme tedy jeho časové průběhy amplitud a fázových posunů všech základních složek. Nic tedy nebrání tomu, aby s těchto hodnot byly vypočteny okamžité hodnoty jednotlivých složek a následně sečtením těchto složek zrekonstruován obraz původního signálu.

Ve skriptu jsou nejdříve vytvořeny vektory daných složek a následně dochází ve *for* cyklu k výpočtu. Složka 50 Hz se počítá pomocí vektoru příslušných amplitud vynásobených funkcí *cosinus* s argumentem dané frekvence a vektoru fázového posunu. 100 Hz složka je počítána stejným způsobem, ovšem s dvojnásobnou frekvencí. Aperiodická složka nepotřebuje přepočítávat, neboť nekmitá s časem. Dále je pak vypočítána celková DC (netočivá) složka, jakožto součet aperiodické a 100 Hz složky. Nakonec je spočítán celkový průběh zkratového proudu, který odpovídá součtu DC a AC (50 Hz) složky.

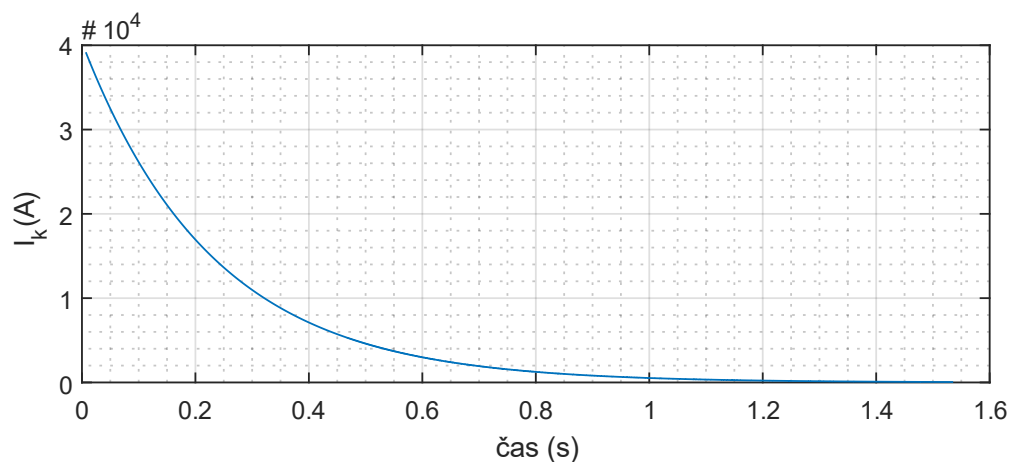
Pokud byl průběh se zápornou aperiodickou složkou (byl tedy otočen na začátku skriptu), dochází v této části také k otočení všech jeho složek.



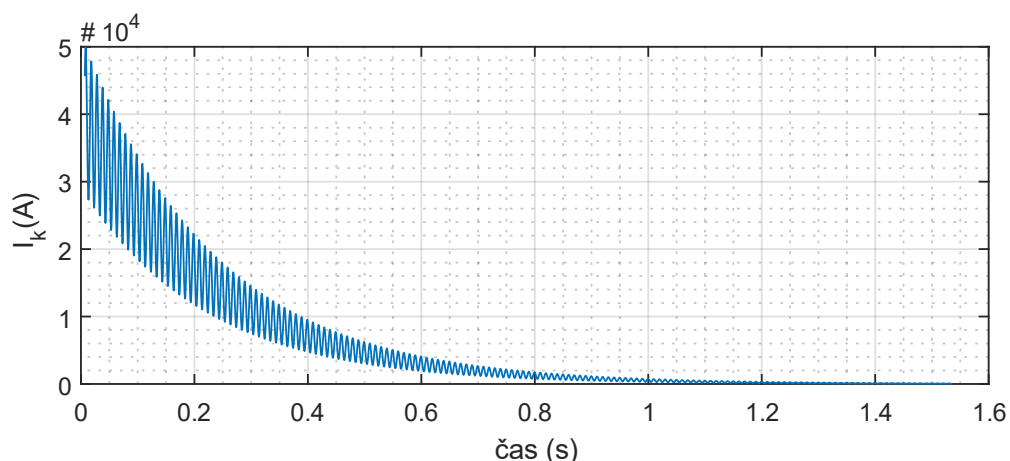
Obrázek 19: Konečný průběh 50 Hz složky



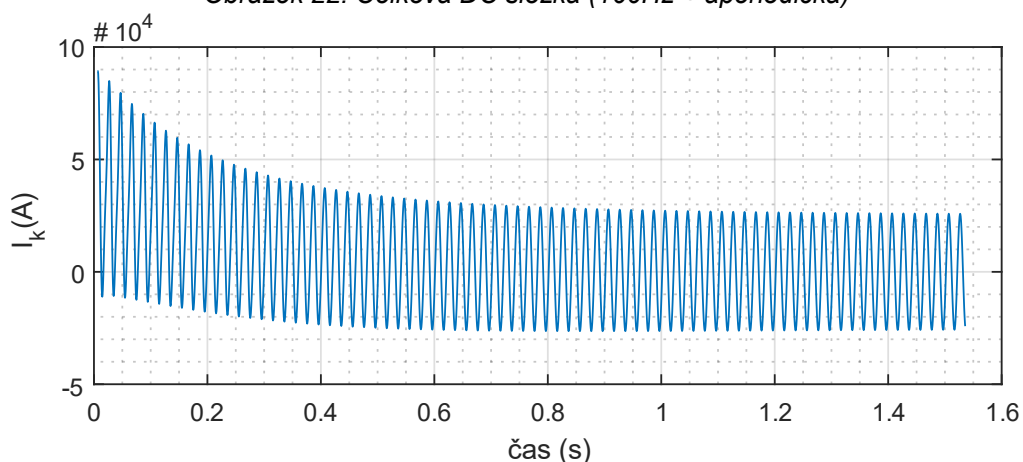
Obrázek 20: Konečný průběh 100 Hz složky



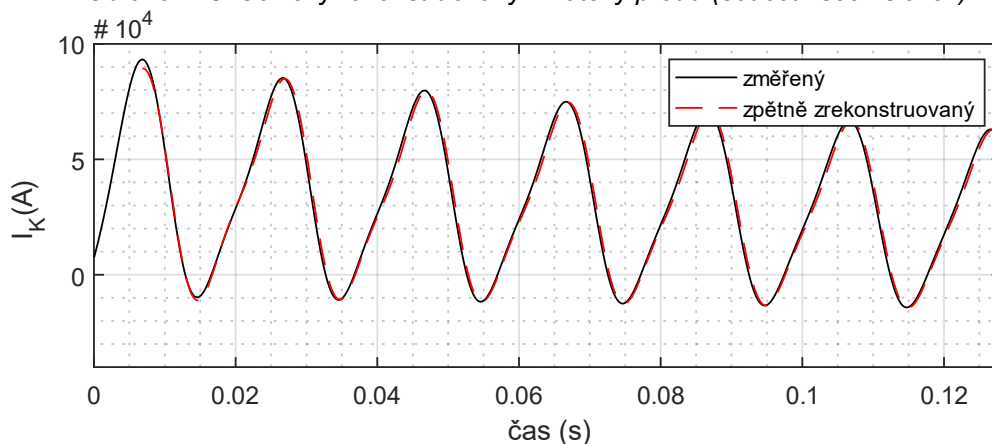
Obrázek 21: Průběh aperiodické složky



Obrázek 22: Celková DC složka (100Hz + aperiodická)



Obrázek 23: Celkový rekonstruovaný zkratový proud (součet všech složek)



Obrázek 24: Porovnání rekonstruovaného a změřeného proudu na několika periodách

Skript na svém konci pak vypíše základní zjištěné informace o průběhu. V tomto textovém výstupu jsou vypsány absolutní maxima okamžitých hodnot jednotlivých složek, jejich časové zasazení, časové konstanty složek a nakonec velikost amplitudy (maximální hodnota) každé složky v okamžiku zkratu.

3 Závěr

Tato bakalářská práce se zabývá problematikou zkratových proudů na synchronních strojích s vyniklými póly. Práce je rozdělena přibližně na polovinu, kde první polovina se zabývá teoretickými znalostmi synchronních strojů a rozbořem zkratů a druhá polovina se zabývá popisem vypracovaného skriptu a problémy, jež vznikaly při jeho vypracování.

Skript jako takový provádí analýzu dodaného průběhu pomocí rozboru Fourierovou transformací na jednotlivé harmonické složky, které jsou pak dále zpracovávány. Při zpracovávání skriptu jsem si všimnul drobné odchylky, jež při analyzování vznikala. Na dodaném průběhu trochu neseďela velikost amplitud v době, kdy už byl průběh ustálen. Pozdějším zkoumáním jsem dospěl k závěru, že je to nejspíše způsobeno tím, že frekvence základní harmonické v tomto okamžiku neodpovídala přesně zadaným 50Hz, ale pohybovala se okolo 49,975 Hz. Toto pravděpodobně vedlo k nepatrné odchylce při výpočtu pomocí FFT.

Podmínky na správné zpracování skriptem jsou uvedeny vždy u dané sekce, jež se přímo týkají. Celkově by se tyto podmínky daly shrnout následovně:

- neměnná vzorkovací frekvence s minimální odchylkou,
- správně zvolená frekvence, jež mají data mít,
- správně nastavený okamžik zkratu t_{zk} (pouze pro výpočet složek v okamžiku zkratu),
- průběh musí být přibližně periodický, a musí alespoň připomínat teoretický průběh (vychýlený periodický průběh s klesajícími amplitudami),
- průběh musí být na konci měření ustálený, jinak dochází ke špatnému výpočtu časových konstant a k nesprávnému proložení.

Seznam literatury a informačních zdrojů

- [1] PETROV, Georgij Nikolajevič. Elektrické stroje 2. Praha : Academia, 1982.
- [2] Synchronní stroje [online]. 2004. [cit. 10. 3. 2018]. Dostupné z: http://fei1.vsb.cz/kat420/vyuka/Bakalarske_FS/prednasky/sylab_synchronni%20stroje_bc%20FS.pdf
- [3] Bartoš, Václav. Elektrické stroje. Plzeň: Západočeská univerzita, 2006. ISBN 80-7043-444-9.
- [4] Proelektrotechniky.cz. Víte, co to je a jak funguje ostrovní systém? [Online]. [cit. 12. 3. 2018]. Dostupné z: <http://www.proelektrotechniky.cz/vzdelavani/20.php>
- [5] Mikeš, Jan. Profi elektrika.cz. [Online] 18. 2. 2016. [cit. 19. 03 2018] Dostupné z: <https://elektrika.cz/data/clanky/synchronoskop>. ISSN 1212-9933.
- [6] Noháč, Ing. Karel. [Online] [Citace: 23. březen 2018] Dostupné z: <http://home.zcu.cz/~nohac/E2/Elektrarny2-cast06-v2.pdf>
- [7] Moravec, Jan. Zkrat v elektrických sítích. [Online]. 31. 20. 2015. [cit: 23. 3. 2018]. Dostupné z: <http://oenergetice.cz/elektrina/zkrat-v-elektrickych-sitich/>
- [8] Bartoš, Václav a Skala, Bohumil. Měření na elektrických strojích. Plzeň: Západočeská univerzita, 2006. ISBN 80-7043-447-3.
- [9] Bartoš, Václav. Teorie elektrických strojů. Plzeň: Západočeská univerzita, 2009. ISBN 978-80-7043-509-0.
- [10] Fázování synchronního generátoru. [Online] 31. 3 2004, [cit: 12. 3. 2018]. Dostupné z: http://fei1.vsb.cz/kat410/studium/studijni_materialy/se/cast_B_el_stroje/se_es_c2_fazovani.pdf
- [11] Fourierova transformace. [Online], [cit: 21. 4. 2018]. Dostupné z: <https://aavos.eu/glossary/fourier-transform/>
- [12] Geek3 CC BY-SA 4.0, Váhové funkce [Online], [cit: 21. 4. 2018] Dostupné z: https://commons.wikimedia.org/wiki/File:Mplwp_window-functions-symmetric.svg
- [13] Stanislav Mašláň, Rychlá Fourierova transformace (FFT) pro AVR. [Online] 28.2.2012, [cit: 21.4.2018]. Dostupné z: <http://elektronika.kvalitne.cz/ATMEL/necoteorie/transformation/AVRFFT/AVRFFT.html>
- [14] MathWorks, Fast Fourier transform [Online], [cit: 21.4.2018] Dostupné z: <https://www.mathworks.com/help/matlab/ref/fft.html>

- [15] Gaussianwaves.com - Signal Processing Simplified [Online] 16. 6. 2014, [cit. 21. 4. 2018]. Dostupné z: <https://www.gaussianwaves.com/2014/07/how-to-plot-fft-using-matlab-fft-of-basic-signals-sine-and-cosine-waves/>
- [16] Synchronact 5 [Online] [cit. 23.3.2018]. Dostupné z: http://www.vfservis.cz/files/000392_901067czb.pdf
- [17] MathWorks, Uiopen [Online], [cit: 28. 4. 2018] Dostupné z: <https://www.mathworks.com/help/matlab/ref/uiopen.html>
- [18] MathWorks, Findpeaks [Online], [cit: 4. 5. 2018]. Dostupné z: <https://www.mathworks.com/help/signal/ref/findpeaks.html>
- [19] MathWorks, FFT [Online], [cit: 4. 5. 2018]. Dostupné z: <https://www.mathworks.com/help/matlab/ref/fft.html>
- [20] MathWorks, Unwrap [Online], [cit: 4. 5. 2018]. Dostupné z: <https://www.mathworks.com/help/matlab/ref/unwrap.html>
- [21] MathWorks, Fit [Online], [cit: 12. 5. 2018]. Dostupné z: <https://www.mathworks.com/help/curvefit/fit.htm>
- [22] MathWorks, Feval [Online], [cit: 12. 5. 2018]. Dostupné z: <https://www.mathworks.com/help/matlab/ref/feval.html>
- [23] MathWorks, Feval [Online], [cit: 12. 5. 2018]. Dostupné z: <https://www.mathworks.com/help/matlab/ref/spline.html>

Přílohy

Příloha A – skript

```
1 close all;
2 clear;
3
4 %vstupni volby
5 plotnout=1;          %(0/1)
6 plotnout_celkovy=1;  %(0/1)
7 ulozit=0;           %(0/1)
8 frekvence=50;       %default=50 (Hz)
9 t_zk=0.9643; %okamzik zkratu - pro extrapolaci slozek
10 %(pri chybe analyzy nejprve zkusit nastavit spravny cas)'
11
12 %-----Nacitani dat-----
13 uiopen('load');
14 if (exist('Ik')&&exist('t'))==0 %osetreni pri spatnem vstupu, nebo zavreni okna
15     disp('Neplatná, nebo chybějící data. ');
16     clear;
17     return;
18 end
19
20 %Oriznuti prubehu o data pred zkratem
21 [~,t_zk_index]=min(abs(t-t_zk)); %Nalezeni nejbližsiho vzorku
22 t=t(t_zk_index:end)-t(t_zk_index); %Posun dat
23 Ik=Ik(t_zk_index:end);
24
25 Fs=1/mean(diff(t));          %Vypocet prumerne vzorkovaci frekvence
26
27 % t      -> vektor casu
28 % Ik     -> vektor vstupniho proudu
29 % Fs     -> vzorkovaci frekvence
30
31 %-----Prvotni rozbor-----
```

```
32
33 if isrow(Ik)==true %-> Sjednoceni na sloupceve vektory
34     Ik=Ik';
35     t=t';
36 end
37 i_mes=Ik;
38
39 otoceno=0;
40 if abs(max(i_mes))<abs(min(i_mes)) %zaporna max(abs) vychylka -> otoceni
41     i_mes=-i_mes;
42     otoceno=1;
43 end
44
45 tol=0.8./frekvence; %min rozdil v case dvou maxim
46 [i_mes_max,i_mes_max_id]=max(abs(i_mes));% Nalezeni absolutního maxima
47                                     [pks_max_fp,locs_max_fp]      =
findpeaks(i_mes(i_mes_max_id:end),'MinPeakDistance',tol*Fs);
48 % Nalezeni dilcich lokalnich maxim
49 locs_max_fp=locs_max_fp+i_mes_max_id-1; %posun o index startu findpeaks
50 locs_max=[i_mes_max_id; locs_max_fp]; %pridani absolutniho maxima do vektoru
dilcich
51 pks_max=[i_mes_max; pks_max_fp];
52
53
54 % plot(t,Ik,t(locs_max),pks_max,'ro')
55
56 % Promenne zmenene/pridane po tomto useku:
57 % pks_max, locs_max -> maxima prubehu (+indexy)
58
59 % -----FFT-----
60 % zanalyzovani vstupniho signalu rozdeleneho na jednotlivé periody
61 % (peak ->peak)
62
63 % vytvoreni vektoru
```

```
64 amp50=zeros(1,length(locs_max)-1)';
65 amp100=zeros(1,length(locs_max)-1)';
66 aper=zeros(1,length(locs_max)-1)';
67 ps50=zeros(1,length(locs_max)-1)';
68 ps100=zeros(1,length(locs_max)-1)';
69
70 % vypočet amplitud a fází složek po jednotlivých časových oknech
71 for index = 2:length(locs_max)
72     i_analyz=i_mes(locs_max(index-1):locs_max(index)); % vytvoření časového okna
73     t_analyz=t(locs_max(index-1):locs_max(index));
74
75     [aper(index-1),amp50(index-1),amp100(index-1),...
76      ps50(index-1), ps100(index-1)]=...
77     rozborFFT(i_analyz,t_analyz,Fs,frekvence);
78 end
79
80 ps50=unwrap(ps50); % Opravi preskoky faze fft -pi<->+pi
81 ps100=unwrap(ps100);
82
83 L_fft=length(locs_max)-1;
84 fft_locs=zeros(1,L_fft)'; % v polovine analyzovane periody
85 for index=1:L_fft
86     fft_locs(index) = (t(locs_max(index))+t(locs_max(index+1)))/2;
87 end
88
89 if(L_fft)<25
90     warning('Počet period je menší, než 25. Casove konstanty nemusí být správné.
91     Zkontrolujte, zdali je průběh na konci ustaleny.');
```

```
97 % -----Proloženi získaných dat-----
98
99 razprech=amp50-amp50(end); %Oriznutí ustaleneho proudu
100
101 %Proloženi součtu rázové a přechodné složky  $a \cdot \exp(b \cdot x) + c \cdot \exp(d \cdot x)$ 
102 razprech_fit=fit(fft_locs(1:end-1),razprech(1:end-1),'exp2');
103
104 % Zjistění časových konstant 50Hz složky
105 konstanta1=-1./razprech_fit.b;
106 konstanta2=-1./razprech_fit.d;
107 [td_razova,index_raz]=min([konstanta1, konstanta2]);
108 td_prechodna=max([konstanta1, konstanta2]);
109
110 % Zjistění amplitud rázové a přechodné složky v okamžiku zkratu
111 if index_raz==1
112     i_prech_t0=abs(razprech_fit.a);
113     i_raz_t0=abs(razprech_fit.c);
114 else
115     i_prech_t0=abs(razprech_fit.c);
116     i_raz_t0=abs(razprech_fit.a);
117 end
118
119     razprech_eval=feval(razprech_fit,t(i_mes_max_id:end)); %Výpočet proložených
hodnot
120     amp50_eval=razprech_eval+amp50(end); %Přičtení ustaleneho proudu a získání
průběhu amplitud 50Hz
121     ps50_eval=spline(fft_locs,ps50,t(i_mes_max_id:end)); %Proloženi fázového posunu
122
123 % Proloženi amplitud aperi. a 100Hz složky tvarem  $a \cdot \exp(b \cdot x)$ 
124 amp100_fit=fit(fft_locs,amp100,'exp1');
125 aper_fit=fit(fft_locs,aper,'exp1');
126
127 % Zjistění aperiodické konstanty
128 ta_aper=-1./aper_fit.b;
```



```
129
130 % Vypocet prolozenych hodnot
131 amp100_eval=feval(amp100_fit,t(i_mes_max_id:end));
132 aper_eval=feval(aper_fit,t(i_mes_max_id:end));
133 ps100_eval=spline(fft_locs,ps100,t(i_mes_max_id:end));
134
135
136 % amp50_eval, amp100_eval, aper_eval, ps50_eval, ps100_eval
137 % -> prolozeny od maxima do konce
138
139 % -----Rekonstrukce puvodniho signalu-----
140 L_rek=length(t(i_mes_max_id:end)); %Pocet bodu k vypoctu
141
142 % Vytvoreni vektoru pro ulozeni
143 slozka_50_hz=zeros(1,L_rek)';
144 slozka_100_hz=zeros(1,L_rek)';
145 total_dc=zeros(1,L_rek)';
146 total_sc=zeros(1,L_rek)';
147
148 slozka_ap=aper_eval; %aperiodicka slozka nekmita -> nemusi se s ni nic delat
149
150 for index=1:L_rek % Vytvoreni kmitavych 50Hz a 100Hz slozek
151     slozka_50_hz(index) =amp50_eval(index) .*cos(2.*pi.*frekvence
.*(t(index+i_mes_max_id-1)-t(i_mes_max_id)) -ps50_eval(index));
152     slozka_100_hz(index) =amp100_eval(index)
.*cos(2.*pi.*2*frekvence.*(t(index+i_mes_max_id-1)-t(i_mes_max_id))
-ps100_eval(index));
153 end
154 for index=1:L_rek %soucet
155     total_dc(index)=slozka_100_hz(index)+slozka_ap(index);
156     total_sc(index)=slozka_50_hz(index)+total_dc(index);
157 end
158 if otoceno==1 % V pripade, ze byl prubeh na zacatku otocen (zaporna ap slozna),
otocime znovu
```

```
159     slozka_50_hz=-slozka_50_hz;
160     slozka_100_hz=-slozka_100_hz;
161     slozka_ap=-slozka_ap;
162     total_dc=-total_dc;
163     total_sc=-total_sc;
164 end
165
166 % -----Vystupy-----
167 if ulozit==1
168     t_analyzed=t(i_mes_max_id:end);
169     total=total_sc;
170     uisave({'t_analyzed' 'total' 'slozka_50_hz' 'slozka_100_hz' 'slozka_ap' 'td_razova'
'td_prechodna' 'ta_aper'},'export');
171     clear t_analyzed total
172 end
173
174 % Spolecne grafy
175 if plotnout==1
176     subplot(2,2,1)
177     set(gcf,'units','normalized','outerposition',[0 0 1 1])
178     plot(t,Ik,'k',...
179         t(i_mes_max_id:end),total_sc,'r--');
180     title('Časový průběh vstupního signálu');
181     legend('změřený','zpětně zrekonstruovaný');
182     xlabel('čas (s)');
183     ylabel('I_K (A)');
184     grid on;
185     grid minor;
186
187     subplot(2,2,2)
188     plot(t(i_mes_max_id:end),slozka_50_hz,"");
189     title1=sprintf('Celková AC (%d Hz) složka průběhu',frekvence);
190     title(title1);
191     xlabel('čas (s)');
```

```
192   ylabel('i_s_t (A)');
193   grid on;
194   grid minor;
195
196   subplot(2,2,3)
197   plot(t(i_mes_max_id:end),slozka_100_hz,'r',...
198        t(i_mes_max_id:end),slozka_ap,'b');
199   title2=sprintf('%dHz a aperiodická složka DC složky,2*frekvence);
200   title(title2);
201   legend2_1=sprintf('%dHz slozka',2*frekvence);
202   legend(legend2_1,'aperiodická složka');
203   xlabel('čas (s)');
204   ylabel('i_a_p (A)');
205   grid on;
206   grid minor;
207
208   subplot(2,2,4)
209   plot(t(i_mes_max_id:end),total_dc);
210   title('Celková DC složka');
211   xlabel('čas (s)');
212   ylabel('i_D_C (A)');
213   grid on;
214   grid minor;
215 end
216 % Celkovy Ik se zpetne rekonstruovanim
217 if plotnout_celkovy==1
218   figure();
219   plot(t,Ik,'k',...
220        t(i_mes_max_id:end),total_sc,'r--');
221   title('Časový průběh vstupního signálu');
222   legend('puvodni','zpetne rekonstruovany');
223   xlabel('čas (s)');
224   ylabel('I_K (A)');
225   grid on;
```

```
226 grid minor;
227 ax=gca;
228 end
229
230
231 % Finalni vypis ziskanych informaci
232 Ik_max=i_mes_max; %Celkove maximum
233 Ik_max_t=t(i_mes_max_id);
234 [i50_max,i50_max_id]=max(abs(slozka_50_hz)); %Maximum 50Hz slozky
235 i50_max_t=t(i50_max_id+i_mes_max_id);
236 [i100_max,i100_max_id]=max(abs(slozka_100_hz)); %Maximum 100Hz slozky
237 i100_max_t=t(i100_max_id+i_mes_max_id);
238
239 shrnuti_s=sprintf('Maxima:\nCelkovy proud:\t%.2e\t%.2f ms\n50Hz
slozka:\t%.2e\t%.2f ms\n100Hz slozka:\t%.2e\t%.2f ms\naper. slozka:\t%.2e',...
240 Ik_max, 1000*Ik_max_t, i50_max, 1000*i50_max_t, i100_max, 1000*i100_max_t,
abs(slozka_ap(1)));
241 konstanty_s=sprintf('Konstanty jednotlivých složek:\nRázová složka:\t%.2f
ms\nPřechodná složka:\t%.2f ms\nAperiodická složka:\t%.2f ms\n',...
242 td_razova*1000, td_prechodna*1000, ta_aper*1000);
243 pocatecni_hodnoty_s=sprintf('Počáteční hodnoty složek (maximální
hodnoty):\nRázová složka:\t%.2f\nPřechodová složka:\t%.2f\nAperiodická
složka:\t%.2f',i_raz_t0,i_prech_t0,aper_fit.a);
244 disp(shrnuti_s);
245 disp(konstanty_s);
246 disp(pocatecni_hodnoty_s);
247
248 function [ aper, amplituda50, amplituda100, ps50, ps100 ] = rozborFFT(
i,t,Fs,frekvence )
249 % Vstupni parametry:
250 % i = vektor proudu
251 % t = vektor casu
252 % Fs = vzorkovaci frekvence
253 % frekvence = Zjistovana prvni harmonicka slozka (50/60 Hz)
```

```
254 %
255 % Output arguments:
256 % aper      = aperiodická složka
257 % amplituda50 = První harmonická složka
258 % amplituda100= Druhá harmonická složka
259 % https://www.mathworks.com/help/matlab/ref/fft.html
260
261 plotnout_fft=0;
262 % pro každou periodu zvlášť(Pozor při >10 periodách,
263 % velmi náročné na vykreslení a zpracování (každá perioda na samostatný graf)) (0/1)
264
265 %Delka transformace musí být celé číslo -> zaokrouhlení + zajištění sudosti
266 L=2.*round(Fs/(2.*frekvence));
267 fourier=fft(i,L);
268
269 %Vytvoření vektoru amplitud
270 P2=abs(fourier/L);      % Absolutní hodnota/delka transformace
271 P1=P2(1:L/2+1);      % -> jednostranné spektrum
272 P1(2:end-1) = 2*P1(2:end-1); % přenasobení 2
273
274
275 % Vytvoření vektoru fázových posunů
276 PP2=angle(fourier/L); % časový fázový posun
277 PP1=-PP2(1:L/2+1);   % jednostranné spektrum
278
279 aper=P1(1);          % aperiodická
280 amplituda50=P1(2);   % 1. harmonická
281 amplituda100=P1(3);  % 2. harmonická
282 ps50=PP1(2);
283 ps100=PP1(3);
284
285 if plotnout_fft==1
286     f = Fs*(0:(L/2))/L; % Vektor frekvencí (x souřadnice)
287     figure;
```

```
288 set(gcf,'units','normalized','outerposition',[0 0 1 1])
289 subplot(3,1,1);
290 plot(t,i);
291 title('casova oblast');
292 xlabel('t (s)');
293 ylabel('amplituda')
294 grid on;
295 subplot(3,1,2);
296 stem(f,P1);
297 title('f spektrum');
298 xlabel('f (Hz)');
299 ylabel('amplituda');
300 subplot(3,1,3)
301 stem(f,PP1*180/pi);
302 title('f spektrum');
303 xlabel('f (Hz)');
304 ylabel('fazovy posun');
305 end
306 end
```