

Univerzita Pardubice
Dopravní fakulta Jana Pernera

Implementace a testování SW modulů pro návrh
nastavování zkratových ochran v mřížových
trakčních sítích 25 kV 50 Hz

2025

Kryštof Prejs

Univerzita Pardubice
Dopravní fakulta Jana Pernera
Akademický rok: 2023/2024

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Bc. Kryštof Prejs**
Osobní číslo: **D23449**
Studijní program: **N0788A040001 Dopravní technika**
Specializace: **Elektrická trakce a elektromobilita**
Téma práce: **Implementace a testování SW modulů pro návrh nastavování zkratových ochran v mřížových trakčních sítích 25 kV 50 Hz**
Zadávající katedra: **Katedra elektrotechniky, elektroniky a zabezpečovací techniky v dopravě**

Zásady pro vypracování

1. Rozpracujte algoritmy vybraných programových modulů pro nastavování zkratových ochran v trakčních mřížových sítích 25 kV 50 Hz
2. Implementujte vybrané SW moduly pro nastavování uvedených zkratových ochran v prostředí Matlab
3. Proveďte a vyhodnoťte testování implementovaných SW modulů
4. Zpracujte základní dokumentaci k implementaci a testování uvedených SW modulů

Rozsah pracovní zprávy: **dle zadání**
Rozsah grafických prací: **dle zadání**
Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam doporučené literatury:

Mohan, N., Undeland, T. M., Robbins, W. P.: *Power Electronics – Converters, Applications, and Design*, John Wiley & Sons, New York, 2003
Firemní podklady Hitachi
Firemní podklady k distančním ochranám

Vedoucí diplomové práce: **prof. Ing. Jaroslav Novák, CSc.**
Katedra elektrotechniky, elektroniky a zabezpečovací
techniky v dopravě

Datum zadání diplomové práce: **6. května 2024**
Termín odevzdání diplomové práce: **9. května 2025**

L.S.

doc. Ing. Ladislav Řeutil, Ph.D.
děkan

Ing. Vítězslav Krčmář, Ph.D.
vedoucí katedry

V Pardubicích dne 6. května 2024

Prohlašuji:

Práci s názvem Implementace a testování SW modulů pro návrh nastavování zkratových ochran v mřížových trakčních sítích 25 kV 50 Hz jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze 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, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 7/2019 Pravidla pro odevzdávání, zveřejňování a formální úpravu závěrečných prací, ve znění pozdějších dodatků, bude práce zveřejněna prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne

Kryštof Prejs v.r.

Rád bych využil této příležitosti a poděkoval své rodině. Dále děkuji firmě Elektrizace železnic Praha a.s. za podporu během mého studia.

Zvláštní poděkování patří prof. Ing. Jaroslavu Novákovi CSc. za jeho velmi užitečné rady při programování a psaní této diplomové práce.

Anotace

Diplomová práce se v první části zabývá popisem měničových technologií používaných či navrhovaných k použití pro napájení trakčních sítí 25 kV 50 Hz. Také vysvětluje impedanční ochrany a jejich použití s tímto druhem napájení. Druhá část obsahuje požadavky na simulační program pro výpočet zkratů v mřížových sítích 25 kV 50 Hz. Ve třetí části je popis logického jádra, které je součástí backendu simulačního programu. Tato část dále obsahuje popis testování logického a také výpočetního jádra, které není součástí této diplomové práce. V poslední části se zabývá návrhem logického jádra pro automatické nastavení zón impedančních ochran.

Klíčová slova

simulace, program, impedanční ochrana, distanční ochrana, měničová technologie, trakční napájecí stanice, MATLAB, zkrat

Title [EN]

Implementation and testing of SW modules for the design of short-circuit protection tuning in grid traction networks 25 kV 50 Hz

Annotation [EN]

The first part of the thesis deals with the description of the inverter technologies used for the supply of 25 kV 50 Hz traction networks. It also explains impedance protection devices and their use with this type of power supply. The second part contains the requirements for a simulation program for the calculation of short circuits in 25 kV 50 Hz grids. This part is followed by a description of the logic core that is part of the backend of the simulation program. The third part also describes the testing of the logic core and also the computational core, which is not part of this thesis. The last section deals with the design of the logic kernel for automatic adjustment of impedance protection zones.

Keywords [EN]

simulation, program, impedance protection, distance protection, inverter technology, substation, MATLAB, short circuit

Obsah

Úvod.....	9
1) Technologie pro jednofázovou soustavu 25 kV 50 Hz.....	11
1.1) Měničové technologie pro napájecí stanice 25 kV 50 Hz	11
1.2) Technologie měničové napájecí stanice a trakčního vedení.....	12
1.3) Impedanční ochrany v mřížových sítích 25 kV 50 Hz	15
1.4) Výpočet zkratových poměrů trakčního vedení	17
2) Program pro výpočty v trakčních sítích 25 kV 50 Hz	18
2.1) Popis komplexního programu z uživatelského pohledu	18
2.2) Základní popis programu z vývojářského pohledu.....	19
2.3) Porovnání programu Univerzity Pardubice s konkurenčními programy pro trakční a energetické výpočty	22
3) Program logického jádra pro výpočet jednotlivého zkratu.....	24
3.1) Popis programu se zjednodušeným vývojovým diagramem	24
3.2) Vstupní tabulka zadání, popis dat tabulky	25
3.2.1) Číslo traťového úseku	25
3.2.2) Číslo počátečního a koncového uzlu	25
3.2.3) Délka traťového úseku.....	26
3.2.4) Vlastní poměrná impedance traťového úseku.....	26
3.2.5) Vazba mezi traťovými úseky a poměrná impedance vazby	26
3.2.6) Napájecí stanice, její uzel, impedance a napětí	26
3.2.7) Proudové omezení a charakteristika NS	27
3.2.8) Traťový úsek se zkratem, vzdálenost a impedance zkratu	27
3.3) Popis hlavních celků programu.....	27
3.3.1) Metodika úpravy vstupních dat pro výpočetní jádro – vložení zkratu	28
3.3.2) Metodika úpravy vstupních dat pro výpočetní jádro – zkratový režim SFC	29
3.3.3) Větvení logického jádra na základě podmínek – vzdálenost zkratu.....	30

3.3.4)	Větvení logického jádra na základě podmínek – vazební traťové úseky.....	31
3.4)	Popis jednotlivých částí programu.....	33
3.4.1)	Počet nových traťových úseků i s vázanými traťovými úseky	33
3.4.2)	Vytvoření nových traťových úseků	34
3.4.3)	Přiřazení uzlu a parametrů zkratovanému traťovému úseku	35
3.4.4)	Přiřazení uzlů a parametrů vázaným traťovým úsekům	36
3.4.5)	Vytvoření nové napájecí stanice na místě zkratu.....	38
3.4.6)	Přenesení programových proměnných do výstupní tabulky a její export.....	38
3.4.7)	Kontrola a zpřesnění výsledků z výpočetního jádra	41
3.5)	Testování součinnosti logického a výpočetního jádra na základě výsledků simulací porovnaných s výsledky zkratových zkoušek.....	45
3.5.1)	Představení použité tratě pro testování programu.....	45
3.5.2)	Nejzajímavější testovací příklady jednotlivých zkratů pro logické jádro.....	46
4)	Logické jádro pro automatické nastavení zón impedančních ochran	55
4.1)	Požadavky a vstupní podmínky logického jádra	55
4.2)	Zjednodušený vývojový diagram pro logické jádro automatického nastavení zón impedančních ochran	55
4.3)	Návrh výstupní tabulky pro automatické nastavení zón.....	58
Závěr		60
Zdroje.....		62
Seznam zkratk		64
Seznam obrázků.....		65
Seznam tabulek		66
Seznam příloh		67

Úvod

S příchodem lepší polovodičové techniky nachází jednotlivé společnosti různé cesty, jak zlepšit napájení různých provozů, které ke své funkci vyžadují vysoké napětí. Jedním z těchto provozů je právě železnice. Konvenční napájení bylo realizováno na každé napájecí stanici dvěma jednofázovými transformátory napájenými ze dvou sdružených napětí. V dnešní době na železnici velmi řešíme zatíženost jednotlivých fází a také podíl jednotlivých harmonických, které se projevují na soustavu o vyšším napětí. V neposlední řadě se řeší i účinník odebíraného proudu. Na místo transformátoru byly díky lepším možnostem dosazeny nepřímé měniče, které s těmito parametry umí velmi dobře pracovat.

Postupné nasazování nepřímých měničů pro napájení železniční trakční infrastruktury přináší celému systému železnice nespornou řadu výhod. Největším přínosem pro železniční provoz je zrušení neutrálních úseků, které mohou způsobovat provozní problémy dopravcům, na něž se klade čím dál větší nárok komfortu cestujících. Při projíždění neutrálním úsekem ztratí hnací vozidlo napájení, tudíž nejsou napájeny pomocné spotřeby vlaku a ani vozy, které potřebují udržovat dodávku energie ke klimatizování, topení a k jiným spotřebám, a to víceméně nepřetržitě. Není to ale jen komfort, který se díky vyřazení neutrálních úseků zlepšuje. Druhou nespornou výhodou je možnost napájet jednotlivé úseky oboustranně, respektive obecně řečeno vícestanně, kdy se výstupní napětí nepřímých měničů dokáže na sebe synchronizovat. Další dvě výhody nesouvisí s napájením železnice, ale s vlivy na nadřazenou distribuční soustavu, kdy při konvenčním napájení nebylo zcela možno symetrizovat odběr. Protože je u nepřímých měničů použit stejnosměrný meziobvod, odběr je víceméně symetrický. Druhou výhodou pro nadřazenou síť může být i fakt, že měniče spolu dokáží spolupracovat na základě nastavení charakteristiky měničů. Správným nastavením charakteristik měničů můžeme dosáhnout toho, že železnice bude dodavatelem jalové energie. Tyto procesy se mohou dít zcela automaticky a tím napomáhat ke stabilitě distribuční soustavy v daných lokalitách.

S měničovou napájecí technologií ovšem souvisí i další změny v napájení. Jedná se hlavně o změnu druhu sítě. S měničovou technologií přecházíme na síť mřížové, u kterých už nelze chránit pouze kritériem proudu, ale i napětí. V tomto případě tedy nelze použít nadproudové ochrany, protože napájení je oboustranné. Přechází se tedy na ochrany distanční (impedanční),

které nám poskytují vyšší komfort v nastavení, a tedy omezení zbytečných výpadků napájení. Tímto krokem se samozřejmě zvyšuje celková komplexnost systémů napájení, než byla dosud.

Jasným ukazatelem smysluplnosti nepřímých měničů nejsou jen jejich výhody, ale i praktické nasazení této technologie. Prvním místem v ČR se stala Trakční napájecí stanice Otrokovice. Další měnič byl nasazen v Trakční napájecí stanici Říkovice, a to z důvodu vyzkoušení dvoustranného napájení. Plánované nasazení nepřímých měničů od firmy Hitachi je v Kladně a také na hraničním přechodu CZ – SK, tedy Střelná – Lysá pod Makytou. Měničová technologie od firmy Siemens byla naplánována do Brna, kde bude těchto měničů několik.

Prvním cílem této diplomové práce je seznámit čtenáře s problematikou použití měničových technologií pro napájení železniční infrastruktury a předat mu tak informace potřebné k pochopení následujících řešení problémů ohledně chránění mřížových sítí.

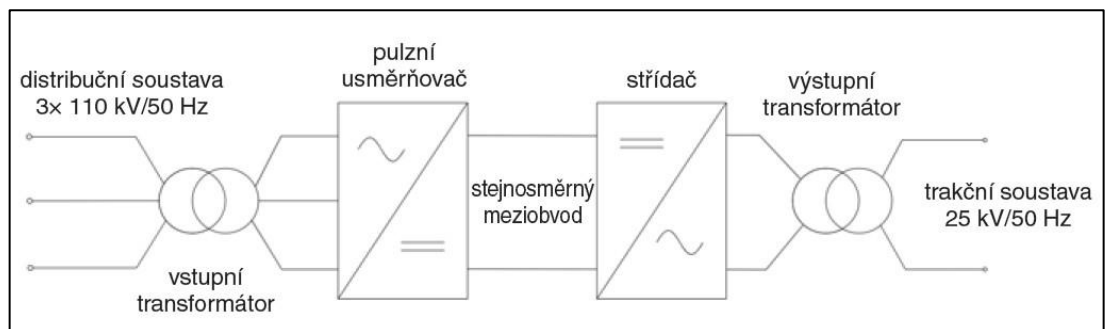
Druhým úkolem, který spadá do praktické části diplomové práce, je navrhnout a napsat program v softwarovém prostředí MATLAB. Tento program by se měl chovat jako logické jádro pro výpočet jednorázového zkratu. Měl by ošetřovat všechny stavy, které mohou při parametrizaci jednorázového zkratu nastat. V neposlední řadě by se mělo logické jádro otestovat. V rámci testování logického jádra bude testováno i výpočetní jádro, které je pouze funkcí v logickém jádře. Výpočetní jádro není mým výtvozem.

Třetím a posledním úkolem je shrnout poznatky z programu logického jádra pro jednotlivé zkraty a navázat na ně dalším logickým jádrem, které bude dotvořeno až po diplomové práci. Druhé logické jádro by mělo být složitější, a to z důvodu automatického návrhu impedančních zón. Za úkol se tedy pro tuto část programu dává, aby byla navržena struktura druhého logického jádra, případně nakreslen zjednodušený vývojový diagram, který sice nebude ukazovat detaily, ale prozradí, jak by mělo být toto logické jádro strukturováno a na základě jakých hodnot by se mělo rozhodovat o jednotlivých zónách. Měly by být také navrženy výstupní tabulky, protože výstupních hodnot z této části automatického nastavení zón bude mnoho. Tento třetí úkol navazuje na diplomovou práci Ing. Ladislava Cermana, který pro tato zadání zkoumal, jak by měl návrh zón z teoretického hlediska vypadat.

1) Technologie pro jednofázovou soustavu 25 kV 50 Hz

1.1) Měničové technologie pro napájecí stanice 25 kV 50 Hz

První měničovou technologií jsou nepřímé měniče. V dnešní době je pro aplikace v trakčních sítích 25 kV 50 Hz na Evropském trhu nabízí hlavně tři společnosti, které se liší svými řešeními. Těmito společnostmi jsou Hitachi, Siemens a General Electric. Těmto měničům se jinak přezdívá SFC (Static frequency converters). Zcela jinou technologii měniče vyvíjí firma Elektrotechnika a.s., která má ve své dílně maticové měniče.



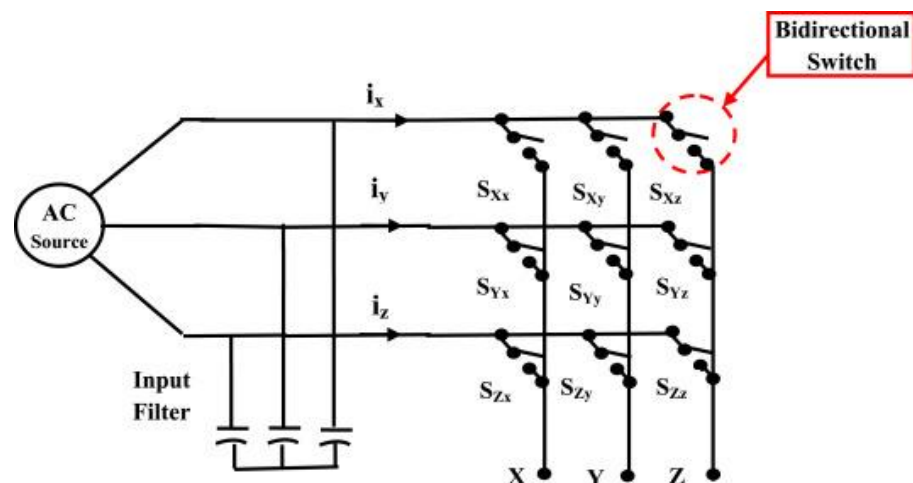
Obrázek 1 - Blokovaná struktura měničové napájecí stanice 25 kV 50 Hz [1]

Blokové schéma na obrázku 1 představuje topologii měniče, kdy z distribuční soustavy 110 kV připojíme snížovací transformátor, který napájí pulzní usměrňovač. Následuje typická topologie pro nepřímé měniče, kdy se ve stejnosměrném obvodu uchovává energie. Ze stejnosměrného napětí vytvoříme za pomoci střídače střídavé napětí, které následně zvyšovacím výstupním transformátorem zvýšíme na 25 kV. Různé vnitřní topologie nepřímých měničů můžeme například vyčíst zde [2].

Nevýhodou nepřímých měničů je již zmíněný stejnosměrný meziobvod. První velmi složitou záležitostí je řízení napětí v meziobvodu. Pokud vezmeme v potaz dynamicky se měnící odběr energie s možnými chvilkovými poruchovými stavy (například do vypnutí vypínače hlavního vozidla) nebo naopak rekuperaci hnacích vozidel na trati, řízení napětí ve stejnosměrném meziobvodu se ještě o to více komplikuje. Se stejnosměrným meziobvodem také souvisí sady elektrolytických kondenzátorů. Jedinou nevýhodou elektrolytických kondenzátorů je jejich vysychání. Pro údržbu bude tedy po čase provozu náročnější tyto kondenzátory diagnostikovat a případně měnit. Na druhou stranu by se měl výrobce i provozovatel snažit držet tyto části v dobrých klimatických podmínkách, aby se co nejvíce prodlužovala životnost celého zařízení.

I přes nevýhody nepřímého měniče byla právě tato technologie zvolena pro napájení trakčních sítí. Technologie je již v praxi vyzkoušená, a to nejen v ČR, ale i v zahraničí. Velmi oblíbené jsou v Evropě, a to obzvláště v zemích, kde je pro napájení trakčního vedení použita frekvence 16,7 Hz (Německo, Švýcarsko, Rakousko). Tato technologie je v těchto zemích používána více než 20 let. O zkušenostech při zprovoznování otrokovického a říkovického měniče pro systém 25 kV 50 Hz pojednává příspěvek na odborném semináři [3]. [4]

Druhou technologií, v železničním napájení ještě nepoužitou, jsou maticové měniče. Maticové měniče vznikly právě díky nevýhodám nepřímých měničů. Při pohledu na topologii na Obrázek 2 si můžeme všimnout, že tyto měniče nemají stejnosměrný meziobvod. Nevýhoda je zde ale zcela jasná – složitost řízení spínání jednotlivých fází maticového měniče. V případě železničního použití pro systém 25 kV 50 Hz by bylo řízení nejspíše o trochu jednodušší, a to z důvodu zmenšeného počtu výstupních fází. Pro železnici by byla výstupní pouze jedna fáze, přičemž druhý vývod měniče by byl uzemněn a ukolejněn. Ovšem i přesto bychom si při řízení museli dávat pozor na to, abychom nezkratovali vůči sobě jednotlivé fáze. Maticový měnič má ještě takovou vlastnost, že jeho výstupní napětí je maximálně 0,866násobkem vstupního napětí. Další nevýhodou, kterou uvádí diplomová práce [6], oproti nepřímému měniči je ta, že při napájení bychom měli používat tzv. FKZ, tedy filtračně-kompenzační zařízení, jež bude kompenzovat pouze jednotlivé harmonické. Účinník si maticový měnič dokáže řídit sám. [5]

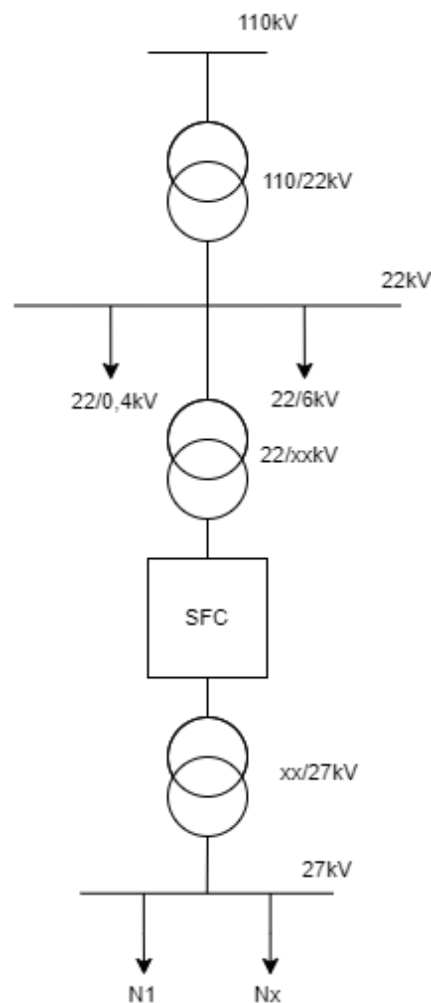


Obrázek 2 - Jedna z možných topologií maticového měniče [3]

1.2) Technologie měničové napájecí stanice a trakčního vedení

Trakční napájecí stanice (dále jen TNS) je rozložením pro napájení nepřímým měničem víceméně stejná jako konvenční napájecí stanice. Přívod je většinou realizován distribuční sítí

o napětí 110 kV, ovšem může být použita i linka 22 kV, protože hned na vstupu do TNS je použit snižovací transformátor 110/22 kV. Napětí 22 kV je rozvedeno přípojnici například i pro vlastní spotřebu napájecí stanice, tedy pro transformátor 22/0,4 kV. Dále zde můžeme najít i vývody pro transformátory, které napájí liniovým rozvodem zabezpečovací zařízení, tedy 22/6 kV. Z přípojnice 22 kV jsou napájeny i vývody pro snižovací transformátory nepřímých měničů. Následuje blokové schéma, které je zobrazeno na Obrázek 1. Poté, co je napětí z měniče zvýšeno na hladinu 27 kV dalším transformátorem, je vývodovým vypínačem přivedeno na přípojnici 27 kV, na které najdeme již jednotlivé napaječe směrem na trakční vedení. Druhý vývod transformátoru je uzemněn a ukolejněna. Vodiče, které vedou k tomuto vývodu zpětné cesty, můžeme nazývat PEN vodiči.

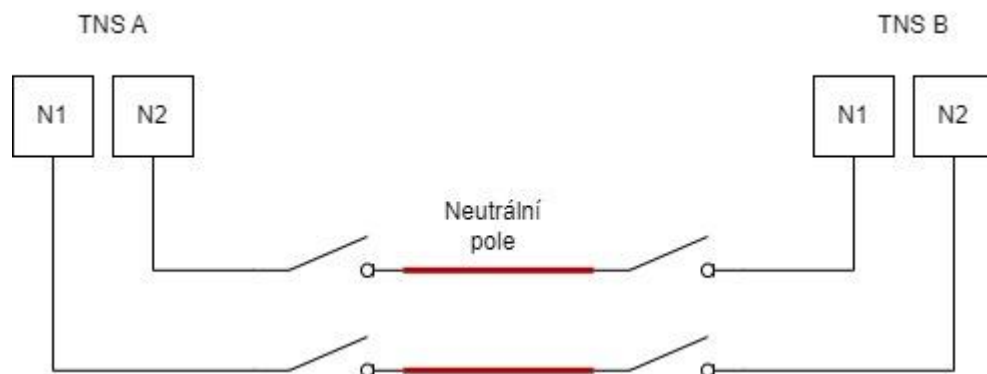


Obrázek 3 - Přehledové schéma TNS

Jak již bylo zmíněno v úvodu, nepřímé měniče mohou synchronizovat svá výstupní napětí tak, aby bylo možno napájet sousední úseky oboustranně či vícestranně. Synchronizace je možno dosáhnout za pomoci správného řízení měničů. Nejdříve je ale potřeba určit, který měnič je master a který slave. To můžeme určit například tím, že si měniče pošlou

informaci o tom, zda jsou již synchronizovány s jiným zařízením, či ne. Pokud budou, je následně jasné, že se měnič v ostrovním režimu bude synchronizovat na měnič již sesynchronizovaný. Poté mezi těmito měniči započne komunikace s aktuálními hodnotami efektivního napětí a úhlu, která se v synchronizujícím se měniči bude porovnávat s aktuální hodnotou. Na základě řízení tohoto měniče se efektivní hodnota i úhel co nejvíce přiblíží k hodnotám druhého měniče a vydá se povolení ke spojení těchto dvou soustav do sebe. Sesynchronizují se tedy fázory výstupních napětí dvou či více sousedících SFC měničů. [6]

Z přechozího odstavce vyplývá, že neutrální pole zcela nevymizí, ale že je lze za určitých podmínek propojit. Propojení neutrálních úseků realizujeme buď za pomoci recloserů, které mohou být umístěny přímo na sloupu, či za pomoci typické spínací stanice. Vypínače či reclosery jsou zpravidla umístěny z každé strany napájení do neutrálního pole. V jedné napájecí stopě jsou umístěny dva vypínače v sérii tak, jak je znázorněno na Obrázek 44. Červeně je znázorněn úsek neutrálního pole.



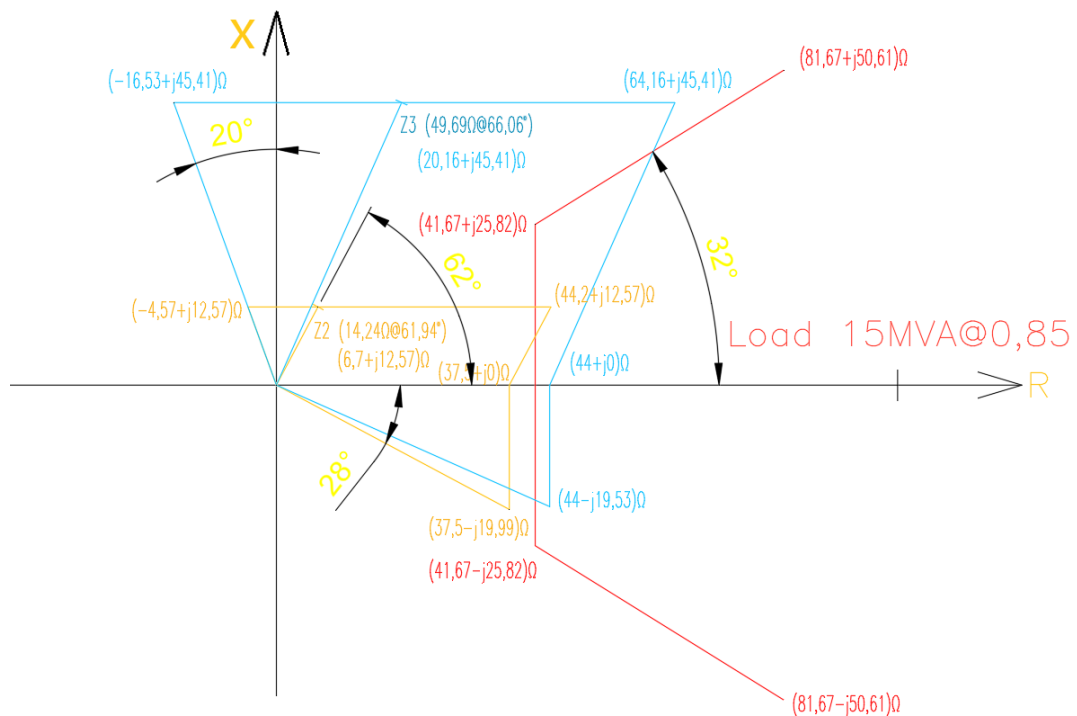
Obrázek 4 - Možnost zapojení recloserů k neutrálnímu úseku a TNS

Neutrální pole můžeme těmito vypínači spojit pouze ve třech případech. Prvním případem je splnění podmínky synchronizace. Druhým případem je možnost jednostranného napájení i do druhého úseku k další napájecí stanici. Tento případ je podmíněn vypnutím vypínače příslušné stopy ve vedlejší napájecí stanici, kdy bude na druhé straně neutrálního pole zcela beznapěťový stav. Třetí možností, a to pouze ve výjimečných případech, je možnost napájet neutrální pole jednostranně. Nespojíme tak dvě soustavy do sebe, ovšem v neutrálním úseku bude napětí pouze z jedné strany. Zde se můžeme dostat do problému na druhém neseprnutém vypínači, a to z důvodu izolační schopnosti vypínače. Pokud totiž budou napětí z TNS v protifázi, fázorově otočené o 180° , bude součet napětí na vypínači roven hodnotě okolo 54 kV a může dojít k proražení izolační bariéry a nechtěnému spojení těchto dvou soustav do sebe. Je tedy potřeba dávat na tento výjimečný stav pozor a učinit opatření již při návrhu

neutrálního pole. Tohoto jednostranného napájení neutrálního pole můžeme využít například tehdy, pokud nějaký vlak v neutrálním poli uvázne.

1.3) Impedanční ochrany v mřížových sítích 25 kV 50 Hz

Impedanční ochrana na základě změřeného proudu a napětí vypočítá v reálném čase fázor impedance. Je tedy jasné, že není důležitá pouze hodnota těchto dvou veličin, ale i úhly ve fázorovém vyjádření.



Obrázek 5 - Příklad nastavení impedanční ochrany napaječe

Na Obrázek 5 vidíme dohromady 3 impedanční zóny. Na ose R je znázorněna reálná hodnota impedance a na ose X imaginární hodnota impedance. První zóna, označena žlutě a popisem Z2, je druhou zónou, jež příslušná ochrana vypíná. Pokud se tedy vypočítaná impedance dostane fázorovým vyjádřením do této plochy, ochrana vypne vypínač. To samé platí i pro zónu označenou modře s popisem Z3. Speciálností ohledně digitálních impedančních ochran je možnost nastavení zóny zátěže, jež je na obrázku označena červeně. Zóna zátěže nám může pomoci s odstraněním falešného vypínání u některých druhů zátěží. Například zde zóna zátěže ořezává jak třetí, tak i druhou zónu, a to právě z toho důvodu, že se u novějších typů lokomotiv předpokládá odběr energie s účínkem okolo hodnoty 1. Tyto lokomotivy se tedy mohou dostat do druhé i první zóny, aniž by ochranu aktivovaly. Přesněji řečeno, pokud by proud a napětí

byly ve fázi a ochrana by vypočítala hodnotu impedance $Z = (42+j0) \Omega$, ochrana by nevypla i když se impedance nachází v zóně pro vypínání.

Zónování impedanční ochrany přináší řadu výhod, jež je velmi dobré v praxi používat. První výhodou je možnost nastavení času vypnutí zóny. Pokud se naměřená impedance dostane do zóny, která má například vypnout až po 250 ms, ochrana celou dobu kontroluje, zda se impedance nachází v určené ploše. Pokud se ale vypočítaná impedance dostane mimo tuto plochu, odpočet se přeruší a vypínač se nevypne. Z toho vyplývá, že krátké přechodné poruchy nemusí znamenat výpadek napájení. Druhou výhodou je možnost nastavení opětovného zapnutí (dále jen OZ) pouze pro určité zóny. Například pro první zónu, kde se většinou nastavuje relativně malá impedance, která je na začátku chráněného úseku, se OZ zcela zruší, tedy neprovede se. Pro další zóny je již OZ použit. Po vypnutí vypínače ochrana na železničním napaječi odpočítává dobu patnácti sekund. Po uplynutí této doby vypínač znovu zapne. Pokud do cca pěti sekund nevznikne nová porucha, OZ se resetuje a při dalším výpadku nastane tato operace znovu. Poslední výhodou, která souvisí s časovým nastavením, je možnost vytvářet tzv. záložní ochrany. Obecně můžeme říct, že vytváříme jakousi časovou selektivitu pro jednotlivé vypínače v různě vzdálených místech od poruchy, do které mohou přispívat. V líniovém, respektive mřížovém, a obecně v jakémkoli napájení jde hlavně o to, abychom vypnuli co nejmenší úsek a omezili tak co nejméně vlaků/odběratelů.

Oproti konvenčnímu napájení se při napájení synchronizovaných SFC měničů dostávají ochrany do pozice, kdy chrání tzv. mřížové sítě. Principem mřížových sítí je faktor vícestranného napájení, kdy do případné poruchy může téci proud z více zdrojů. Právě z tohoto důvodu nemůžeme používat nadproudové ochrany. Pokud by totiž porucha vznikla uprostřed chráněného úseku, oba zdroje by dodávaly přesně poloviční proud. Tím pádem by se tento poruchový proud pro obě ochrany choval jako normální zátěž a žádná z ochran by poruchu nevypla. Frekvenční ochrany také nelze použít, protože jednotlivé příspěvky zdrojů do poruchy nemusí být natolik velké, aby docházelo ke zmenšení frekvence napájecího napětí při poruše.

[1]

To, že použijeme pro chránění distanční ochrany, nám stále nezjednodušuje návrh chránění. Infeed factor, tedy příspěvek do poruchy z jiného zdroje, zde máme stále. Proud, který teče do poruchy přes ochranu jedné TNS, je stále menší a napětí díky oboustrannému napájení

vzrostlo. Celková impedance, kterou ochrana počítá, je tedy chybně vyšší. Pokud bychom jednotlivé zóny špatně navrhli, může se stát, že ochrana bude mít tak velkou odchylku od skutečné impedance, že nebude vypínat. Ke každému výpočtu tedy vždy přidáváme jistotu zóny (zvětšujeme ji) tak, abychom si byli jistí, že ochrana vypne. Navíc dle metodiky nastavení impedančních ochran máme některé zóny, které vypínají poruchy i za přípojnicí vedlejší napájecí stanice. Je tedy velmi nepravděpodobné, že alespoň jedna ochrana v trase poruchového proudu nevypne. [1]

Dalším faktorem, který přispívá k bezpečnosti vypnutí je komunikace mezi ochranami. Ochrany mezi sebou komunikují optickým vláknem výhradně určeným pro komunikaci diferenciálních ochran, která se v trakčních sítích samozřejmě nepoužívá. Přes toto komunikační rozhraní si dvě ochrany na jedné stopě posílají informace o zapnutém či vypnutém stavu. Dále signál k vypnutí, aby nedocházelo k příspěvku do poruchy, kterou ochrana na druhé straně teoreticky nemusí vidět (má malý příspěvek do poruchy). Posílají si i případnou připravenost k sepnutí OZ a zda vůbec bude OZ účinkovat. Všechny tyto informace jsou velmi potřebné k bezproblémovému chodu oboustranně napájených úseků trakčního vedení. Pokud bychom chtěli být ve vypínání ještě přesnější, mohou si ochrany posílat buď vypočtené impedance, či měřené hodnoty proudu a napětí. Z těchto hodnot na druhém konci vedení by se dopočítával celkový proud, který vtéká směrem do poruchy. Na základě těchto informací by ochrana taktéž mohla vypínat.

1.4) Výpočet zkratových poměrů trakčního vedení

Výpočet zkratových poměrů byl do nedávné doby realizován hlavně za pomoci služební rukověti SR34 [7], která jasně udávala pro jednotlivé trakční sestavy reálnou i imaginární složku impedance v jednotkách Ω/km . Stále je mezi ochranáři velmi oblíbenou příručkou, jež jasně udává nastavení starších distančních kruhových ochran. Ovšem v této době a obzvláště s novým vícestranným napájením i na střídavé soustavě ji již nelze tak lehce na nové typy ochran aplikovat. Ve firmě Elektrizace železnic Praha a.s. se projektanti shodli, že potřebují lepší nástroje k výpočtům nastavení, nebo alespoň lepší pomůcku, jež jim pomůže rozhodovat o parametrech nastavení. Vznikl tedy projekt, ze kterého by měl vzejít program pro výpočet zkratových poměrů, nastavení impedančních zón a také výpočet provozního scénáře.

2) Program pro výpočty v trakčních sítích 25 kV 50 Hz

2.1) Popis komplexního programu z uživatelského pohledu

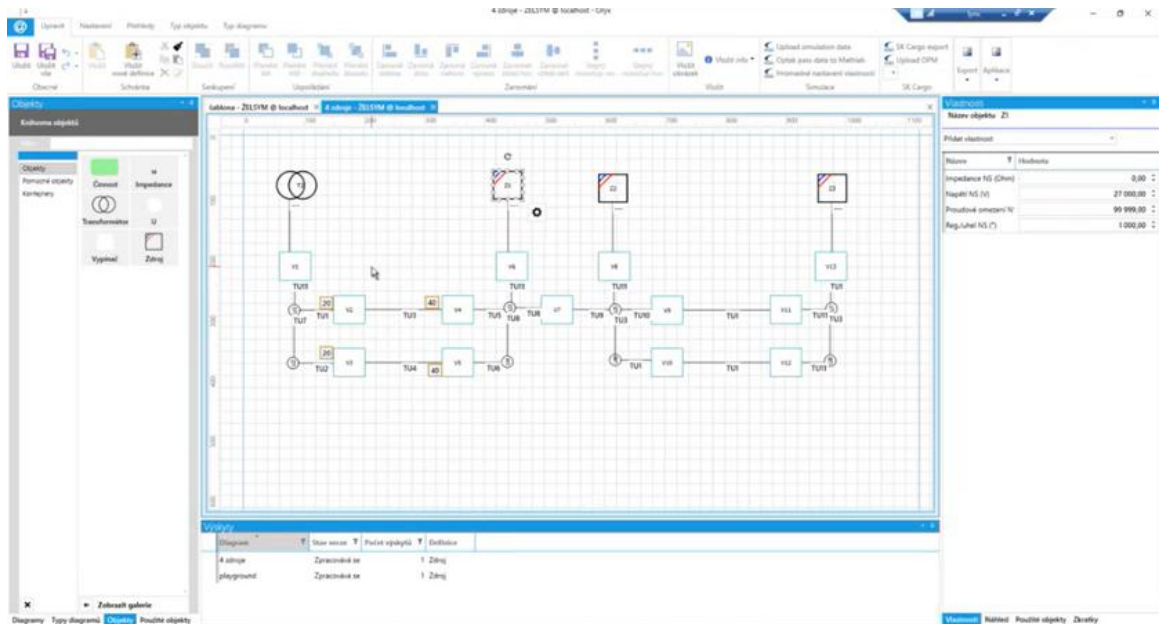
Celý program je obecně zaměřen na výpočty v jednofázových trakčních sítích 25 kV 50 Hz, jež se v České republice používají pro napájení hnacích vozidel na železnici. Z uživatelského pohledu se program skládá ze tří hlavních částí, které může uživatel libovolně používat dle svých potřeb. Musí ovšem znát vstupní parametry výpočtů, které si popíšeme v dalších kapitolách. Částmi programu jsou:

- a) Výpočet hodnot pro jednotlivý zkrat, který si uživatel určí
- b) Automatické navržení nastavení zón impedančních ochran (nadstavba jednotlivého zkratu)
- c) Výpočet provozního scénáře s vlaky na určené trati, kdy si program sám může měnit nastavení charakteristik měničů tak, aby bylo napájení co nejefektivnější

Jedná se tedy o komplexní program, který je pomůckou jak při návrhu chránění sítě, tak i ověření tohoto nastavení v reálném provozu, který se na trati může vyskytovat. I když je ale program složitý, pro uživatele se stále tváří jako jediná aplikace, do které zadává vstupní hodnoty.

Zadávání probíhá za pomoci grafického rozhraní, kde uživatel kreslí schéma trati. K dispozici má základní stavební prvky – traťové koleje, měničové napájecí stanice (SFC), konvenční napájecí stanice, napaječové vypínače a vypínače zdroje. Z těchto základních stavebních prvků si v grafickém rozhraní vytvoří celý přehled trati, na kterém bude chtít počítat jakoukoli částí programu. Po rozkreslení trati přichází na řadu parametrizace jednotlivých položek. Uživatel zadává měrné impedance jednotlivých traťových kolejí, jejich délku, impedance všech zdrojů a jejich jmenovitá napětí a proud. Vypínače následně slouží jakožto měřící body pro ochrany a také jako body rozpojení, kdy si uživatel za pomoci vypínačů může zkonfigurovat napájení trati stejně, jako to může v reálném provozu udělat elektrodispečer. Poté, co uživatel naparametrizuje všechny položky ve schématu, může začít s jakýmkoli výpočtem. Pro každý druh výpočtu musí určit ještě další vstupní proměnné, či musí použít další základní stavební prvek v grafickém zobrazení. Grafické rozhraní je zobrazeno na Obrázek 6.

Pro výpočet jednotlivého zkratu využije uživatel další základní prvek, a to zkrat. Umístí ho ke koleji, na které by mělo dojít k výpočtu zkratu, uvede jeho impedanci a vzdálenost od začátku traťového úseku. Poté může začít výpočet. Pro automatické navržení nastavení zón impedančních ochran pouze uživatel zvolí, k jakým vypínačům bude chtít tento výpočet provést. U výpočtu provozního scénáře už bude muset uživatel přímo určit jízdní řád. Zadávání jízdního řádu není předmětem této práce.



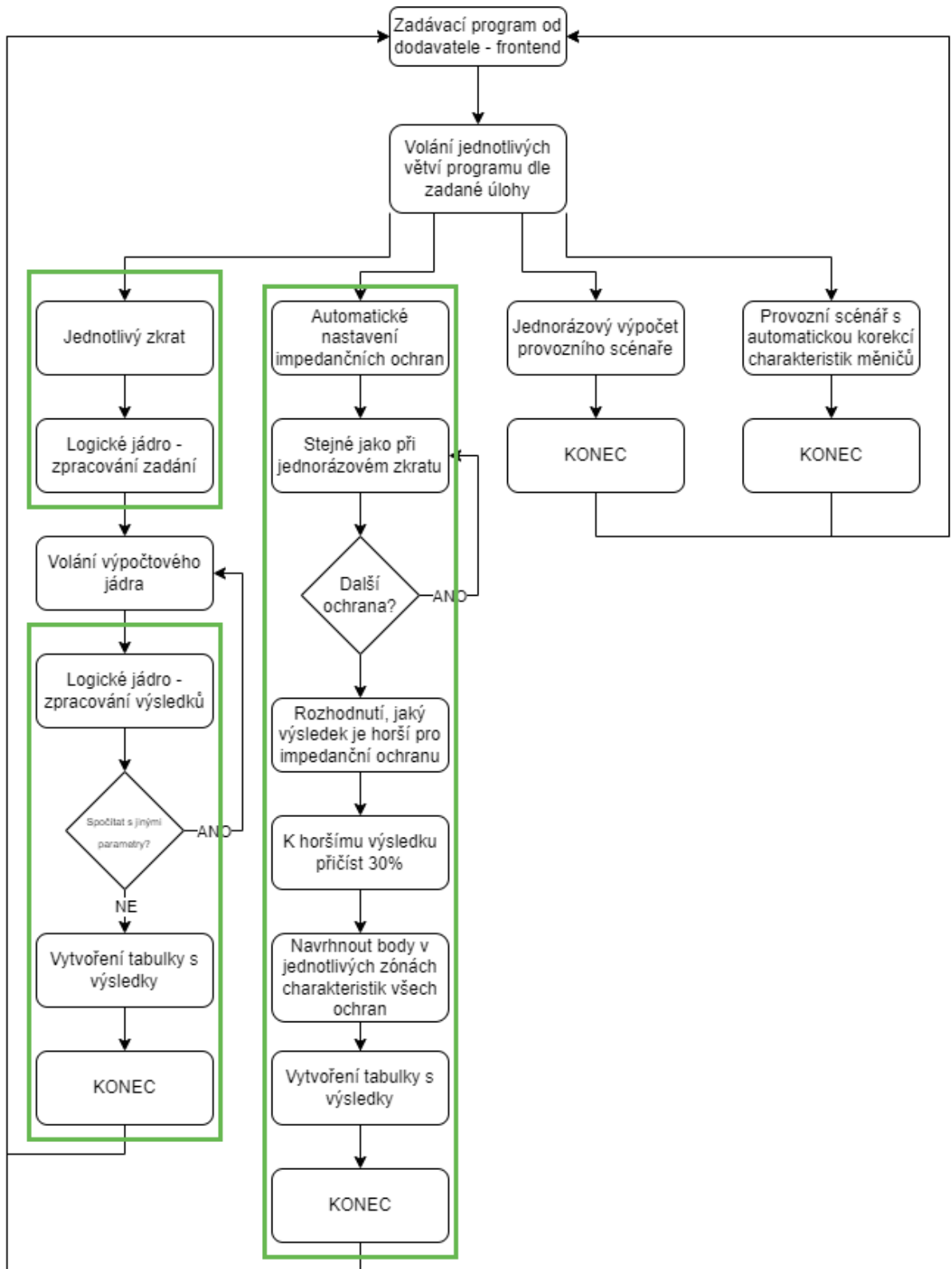
Obrázek 6 - Grafické rozhraní programu pro výpočty v trakčních sítích 25 kV 50 Hz [12]

2.2) Základní popis programu z vývojářského pohledu

Z pohledu vývojářů dělíme program na čtyři obecné části, které si spolu předávají informace. Těmito částmi jsou:

- Grafické uživatelské rozhraní
- Logické jádro rozdělující úlohy pro jednotlivý zkrat a automatické nastavení zón impedančních ochran
- Logické jádro pro výpočet provozního scénáře s možností automatické korekce nastavení charakteristik měničů
- Výpočetní jádro

Na Obrázek 7 je zobrazen zjednodušený vývojový diagram, který má za cíl přiblížit funkce jednotlivých částí. Jak si můžeme všimnout, na vrcholu je vždy zadávací program, tedy grafické rozhraní, do kterého uživatel zadává všechny potřebné údaje k výpočtům. Od tohoto zadávacího programu se již rozvětvují jednotlivé úlohy do různých logických jader, která mají za úkol řešit zpracování dat zadání a přeložit je výpočetnímu jádru do srozumitelné podoby, kterou výpočetní jádro vyžaduje. Následně mají všechna logická jádra za úkol i zpracování dat z výpočetního jádra do podoby, které naopak bude rozumět grafické rozhraní. Logická jádra navíc mohou na základě výpočtů rozhodovat o výsledcích. Tím je myšleno, že například u automatického nastavení zón impedančních ochran logické jádro rozhoduje o velikosti a tvaru jednotlivých zón právě na základě výpočtů z výpočetního jádra. Logická jádra mají tedy samostatné rozhodovací nástroje.



Obrázek 7 - Zjednodušený vývojový diagram pro upřesnění funkcí jednotlivých částí programu

Postup při programování komplexního celku byl takový, že se nejdříve vytvořilo výpočetní jádro. Nejjednodušší programování výpočetního jádra bylo stanoveno v softwaru MATLAB, který je pro výpočetní úlohy určen. Abychom nemuseli během dalších prací řešit komunikaci

mezi výpočetním a logickými jádry, zvolil se software MATLAB i pro logická jádra. Jednotlivé skripty tak na sebe mohou navazovat. Spolupráce, tedy předávání dat k výpočtům, mezi logickými a výpočetním jádrem probíhá pomocí přesně definovaných tabulek. Logická jádra vždy volají výpočetní jádro, aby na základě předložených tabulek provedlo výpočet. Výpočetní jádro je tedy v jazyku MATLAB pouze funkcí.

2.3) Porovnání programu Univerzity Pardubice s konkurenčními programy pro trakční a energetické výpočty

V aktuální době se ke slovu dostávají i jiné výpočetní programy pro trakční výpočty, které jsou na trhu již delší dobu a prokázaly svou nenahraditelnou existencí, že softwary tohoto typu mají smysl. Mezi nejvyspělejší programy pro trakční výpočty patří program od společnosti Siemens s názvem Sitras Sidytrac. Na stejném místě je i program od společnosti IFB Institut für Bahntechnik GmbH, který nese název OpenPowerNet. Oba programy nabízí možnost simulací AC i DC systémů. Jejich nespornou výhodou oproti našemu programu je, že při simulaci jízd hnacích vozidel berou v potaz i vzájemné elektrické interakce nejen mezi vozidly, ale i interakce trakčního vedení.

Celý program OpenPowerNet se skládá z jednotlivých zásuvných modulů, jež fungují jako podprogramy. Modul s názvem OpenPowerTrack je silně specializován na trakční výpočty pro jedno či dvoufázové soustavy. Velká nevýhoda tohoto programu tkví právě v těchto modulech, které nám celé zadávání rozděluje celkem do tří částí. Abychom mohli vypočítat úlohu, musíme nejdříve v modulu OpenPowerTrack nakreslit strukturu tratě. Následně přejdeme do základního modulu OpenPowerNet, kde určíme všem nakresleným částem parametry. Poté se znovu musíme vrátit do OpenPowerTracku a vyplnit jízdní řád. Výhodou programu je naopak přesnost ve vyjádření vazeb mezi sousedními kolejemi, kdy na základě jejich vzdáleností a magnetického toku dopočítává vzájemnou indukčnost trakčního vedení. Zpětné vedení už ale není simulováno tak komplexně, protože je za zpětnou cestu dosazen pouze jeden vodič se specifickou impedancí. Toto nahrazení nemusí dokonale postihnout komplexnost složení zpětné cesty. [8]

Program Sitras Sidytrac je oproti OpenPowerNetu kompaktním programem. Podporuje jednofázové i vícefázové výpočty. Díky kompaktnosti má intuitivnější rozhraní pro zadávání parametrů trati. Součástí programu je tzv. MKP, které program používá pro detailnější výpočet

zpětné cesty. Může tak sice dosahovat vyšší přesnosti ve výpočtech, ovšem komplexnost MKP zvyšuje nároky na zadávání parametrů. Sitras Sidytrac není tak přesný při vzájemném ovlivňování jedoucích vozidel, ale má větší míru komplexnosti v nastavení parametrů napájecích stanic. [9]

Program Univerzity Pardubice nabízí pouze výpočty pro jednofázovou soustavu 25 kV 50 Hz. V detailnosti napájecích stanic se velmi blíží programu od Siemensu, a to z důvodu možnosti nastavení více zdrojů a jejich parametrů. Navíc dokáže řešit charakteristiky nepřímých měničů, což v žádném z přechozích programů nelze. Také si umí charakteristiky upravovat na základě aktuálních odběrů vlaků zadaných v jízdním řádu. Všechny vzájemné vazby jsou zde řešeny pouze zadáním vzájemných impedancí. Nepoužívá ke zpřesnění vzájemných vazeb simulaci magnetických toků, ovšem i přes to může dosahovat velmi velké přesnosti, a to na základě právě zadaných parametrů od uživatele, který si může vzájemné ovlivňování různých trakčních sestav změřit a určit parametrem poměrnou vzájemnou indukčnost. Nespornou výhodou je také možnost automatického návrhu zón impedančních ochran.

Zpětná cesta v programu Univerzity Pardubice se na základě simulací provedených v diplomové práci [1] nezanedbává. Byla ale použita zjednodušující náhrada, kdy se impedance zpětné cesty sečte s impedancí trakčního vedení. Tímto zjednodušujícím krokem dostaneme v metodě uzlových napětí méně rovnic a celý výpočet proběhne rychleji. [1]

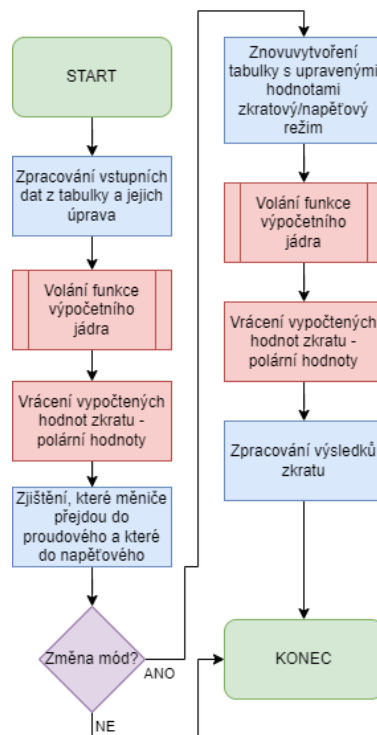
3) Program logického jádra pro výpočet jednotlivého zkratu

Úvodem této kapitoly je třeba zdůraznit, že je ústřední částí diplomové práce a představuje tak hlavní přínosy k řešené problematice programu logického jádra pro výpočet jednotlivého zkratu.

V popisu programu pro logické jádro jednotlivého zkratu si nejdříve popíšeme zjednodušeným vývojovým diagramem základní funkci programu. Následně budou popsány vstupy do logického jádra. Poté si vysvětlíme základní principy změn v zadávací tabulce a také rozdělení logického jádra do tří větví. Bude také popsán celý zbytek programu dopodrobna tím, že se z celkového vývojového diagramu budou vybírat menší funkční celky, které budou zobrazeny a popsány. Posledním cílem této kapitoly je představení testování této části programu na reálných příkladech, kde byly provedeny zkratové zkoušky. Výsledky simulačního programu se porovnají s výsledky zkratových zkoušek a na základě těchto porovnání se vyhodnotí, zda je funkční i výpočetní jádro.

3.1) Popis programu se zjednodušeným vývojovým diagramem

Na Obrázek 8 vidíme zjednodušený vývojový diagram. Logické jádro pro jednotlivé zkraty se zavolá poté, co uživatel v grafickém rozhraní zadá všechny potřebné parametry pro traťové koleje, napájecí stanice, nakonfiguruje trať a přidá objekt zkratu na místo, kde chce zkrat vytvořit. Logické jádro následně dle zadaných parametrů v tabulce, která je taktéž vygenerována grafickou nadstavbou, upraví tyto parametry pro výpočtové jádro tak, aby mohlo správně vypočítat veškeré veličiny, které si bude chtít uživatel zobrazit. Následuje volání výpočtového jádra, které vrátí vypočítané výsledky taktéž v tabulce. Logické jádro má následně za úkol zjistit, zda nějaké z SFC přejde do zkratového režimu. Pokud ne, program se ukončí a grafické rozhraní použije tabulku z výpočtového jádra k zobrazení výsledků. Když dojde na základě dále zmíněných podmínek ke změně režimu měniče, musí se celé zadání přepočítat, protože změna režimu měniče velmi ovlivňuje výsledek výpočtu.



Obrázek 8 - Zjednodušený vývojový diagram popisující hlavní funkce

3.2) Vstupní tabulka zadání, popis dat tabulky

Grafická nadstavba, říkáme jí frontend, vygeneruje zadávací tabulku. Tato tabulka se z důvodu použití metody uzlových napětí musí upravit právě do takové podoby, aby bylo možno tuto výpočetní metodu ve výpočetním jádře použít. Pojdme si popsat jednotlivá data v zadávací tabulce.

3.2.1) Číslo traťového úseku

K jasné identifikaci jednotlivých traťových kolejí je použit právě tento sloupec. Ke každému řádku se následně váže specifické číslo. Začínáme vždy od čísla jedna a postupuje se po jedné k vyšším číslům. Uživatel v grafickém rozhraní nemá k tomuto číslování přístup. Toto číslování je čistě interní a slouží k identifikaci pro výsledky, které se vážou přímo k traťovým kolejím.

3.2.2) Číslo počátečního a koncového uzlu

Stejně jako číslo traťového úseku slouží čísla jednotlivých uzlů k identifikaci. Počáteční a koncový uzel jsou dohromady dva sloupce dat. Díky tomuto zadávání můžeme jednotlivým traťovým úsekům určit i směr, aneb od počátečního uzlu ke koncovému uzlu. To je důležité hlavně pro vzdálenost místa zkratu, která se do tabulky zadává právě od počátku traťového

úseku. Další důležitou vlastností je i to, že víme, které traťové úseky na sebe navazují a tím můžeme dát dohromady celou zadanou trať, která je kreslena ve frontendu.

3.2.3) Délka traťového úseku

Délka traťového úseku je vztažena vždy k určitému řádku (číslu) traťového úseku. Jedná se tedy pouze o identifikaci délky, která se následně ve výpočtu váže ke všem poměrným impedancím, které jsou zadány v základní jednotce Ω/km . Délka traťového úseku je zadávána v kilometrech.

3.2.4) Vlastní poměrná impedance traťového úseku

Každý traťový úsek může mít zcela odlišnou poměrnou impedanci. V zadávací tabulce je tedy zřízen tento sloupec, který každému traťovému úseku poměrnou impedanci přiřazuje. Tuto hodnotu má možnost uživatel měnit v parametrizaci traťových úseků. Impedance je zadávána ve formátu $R + jX$.

3.2.5) Vazba mezi traťovými úseky a poměrná impedance vazby

Další dva sloupce určují, mezi jakými traťovými úseky vzniká impedanční vazba. Identifikátorem pro vazbu jsou čísla traťových úseků, která jsou určena frontendem. Uživatel musí tuto vazbu mezi traťovými úseky nastavit a určit také poměrnou impedanci vazby, která je zadána taktéž v Ω/km a formátu $R + jX$. Důležité je také zmínit, že pokud chce uživatel nějaké traťové úseky svázat indukční vazbou, musí tyto úseky být kilometricky stejně dlouhé a také musí mít stejný směr kilometráže. Další informace si program bere z ostatních polí.

3.2.6) Napájecí stanice, její uzel, impedance a napětí

Další čtyři sloupce v tabulce se věnují parametrizaci jednotlivých napájecích stanic. Pro jasnou identifikaci je prvním sloupcem číslo napájecí stanice (dále jen NS). Abychom věděli, kde se NS nachází, je zde určen její uzel. Z důvodu použití metody uzlových napětí nelze NS umístit na traťovou kolej. Toto omezení lze obejít tím způsobem, že se traťová kolej rozdělí uzlem, do kterého již lze NS umístit. Impedance NS určuje její vnitřní odpor, který je do výpočtu také zahrnut. Impedance zde není poměrná, tudíž je v jednotkách Ω , ale znovu se vyžaduje použití formátu $R + jX$. Napětí NS je napětí naprázdno, tedy nezatížené

NS, jež se do výpočtu zahrnuje. Uživatel tak může měnit tuto hodnotu a převést výpočty do reálného prostředí, kde mohou být napětí naprázdno rozdílná.

3.2.7) Proudové omezení a charakteristika NS

Proudové omezení a charakteristika NS se vztahují pouze k zadání, kdy je jako zdroj použit SFC. Proudové omezení i použitou charakteristiku SFC si může uživatel definovat dle svých vlastních potřeb. Tyto veličiny se zahrnují do výpočtu a ovlivňují jeho konečný výsledek.

3.2.8) Traťový úsek se zkratem, vzdálenost a impedance zkratu

Na dalším listu zadávací tabulky se nachází jeden sloupec, ve kterém jsou určeny tři hodnoty. První hodnotou je určeno místo zkratu. Místo se zapisuje pomocí čísla traťového úseku, na kterém byl zkrat vyvolán. Logické jádro také potřebuje vědět, v jaké vzdálenosti od počátku traťového úseku je zkrat umístěn. K tomu slouží druhá hodnota, která udává místo zkratu na traťovém úseku v kilometrech. Poslední hodnotou je impedance zkratu, která je zadávána v Ω . Typický formát v celém programu pro jakoukoli impedanci zůstává stále stejný, tedy $R + jX$. Všechny tyto 3 hodnoty určuje uživatel v grafickém rozhraní. Tyto tři parametry je možno najít v objektu zkratu.

3.3) Popis hlavních celků programu

Jak již bylo zmíněno, hlavním vstupem do programu je tabulka vygenerovaná frontendovou aplikací. Abychom s těmito daty mohli pracovat, musíme si nejdříve všechny proměnné v tabulce převést na správné druhy proměnných, které následně budeme při běhu používat. Program si nejdříve načte první list tabulky, která obsahuje všechna data ohledně zadaných tratí, napájecích stanic a jejich parametrů. Následně si načte i druhý list, kde jsou uvedeny parametry zkratu.

Následuje určení konstant, aby se mohlo s celým zdrojovým kódem lépe pracovat. Jsou zde popsána umístění výstupního souboru z logického jádra, umístění výpočetního jádra a také umístění výsledků z výpočetního jádra. Dále zde najdeme napětí nové NS, proud nové NS, charakteristiku nové NS, počáteční polohu zadání vlaku, zkratové napětí NS, zkratovou impedanci NS a zkratové proudové omezení NS. V neposlední řadě je zde určeno i podpětí sítě. Všechny tyto proměnné jsou právě na začátku z důvodu přehlednosti a lehké upravitelnosti.

Dále zde najdeme definici proměnných a jejich přetypování na určitý druh proměnné. Názvy proměnné víceméně odpovídají názvům jednotlivých sloupců v tabulce tak, aby bylo s daty pracováno co nejpřehledněji.

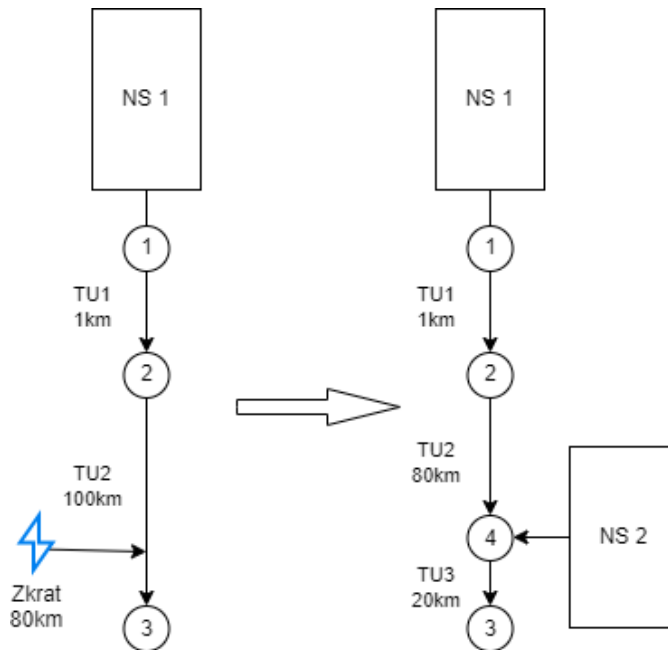
V posledním kroku před začátkem logických funkcí jsou ještě vytvořeny sloupce, které potřebuje ke správnému zadání výpočetní jádro. Jedná se hlavně o vytvoření fiktivního vlaku na trati ve třiceti časových intervalech. Jeho poloha je určena číslem traťového úseku, který je aktuálně stanoven vždy na traťový úsek číslo jedna. Navíc se ještě zadává jeho rychlost, která je 0 km/h, výkon, který je 0,001 kW a také účinník, jež je nastaven na 0,99. Tento fiktivní vlak nemá žádný vliv na jakýkoli zkrat, ale výpočetní jádro ho vyžaduje. Pokud nebude vlak v zadávací tabulce uveden, výpočetní jádro vytvoří chybovou hlášku.

Nejdříve si podrobně popíšeme metodiku úprav, které je potřeba udělat, aby výpočetní jádro rozumělo zadání a mohlo vypočítat všechny neznámé, které budou moci být zobrazeny v grafickém rozhraní programu.

3.3.1) Metodika úpravy vstupních dat pro výpočetní jádro – vložení zkratu

Nejdůležitější úprava vstupních dat se týká vložení zadání zkratu do takové podoby, aby tomuto zadání výpočetní jádro (dále jen backend) správně rozumělo. Z pohledu tabulek nám do logického jádra vstupují z grafického rozhraní tabulky dvě. Jedna zahrnuje celé složení tratě se všemi vloženými elementy, přičemž druhá určuje pouze parametry zkratu. Tyto parametry zkratu se tedy musí správně vložit do zadávací tabulky backendu.

Abychom správně pochopili tuto úpravu, ukažme si jí na příkladu jednokolejné trati. Na Obrázek 9 vlevo najdeme schéma původní trati. Místo budoucího zkratu je označeno bleskem, který vkládá uživatel. Na stejném obrázku, tentokrát vpravo se již nachází upravené zadání pro podobu backendu.



Obrázek 9 - Příklad úpravy zadávací tabulky na jednokolejně trati

Jednokolejná trať se skládá z napájecí stanice NS 1, dvou traťových úseků TU_x a tří uzlů. V uzlech vždy začínají a končí jednotlivé traťové úseky. Zároveň se do uzlů vždy vkládají napájecí stanice. Pro potřeby výpočtového jádra je nutno tuto jednokolejnou trať upravit tak, že v místě zkratu na původním TU2 vytvoříme nový uzel s označením čtyři. Protože jsme trať TU2 rozdělili na dva úseky, musí vzniknout nový traťový úsek, který je pojmenován TU3. Všimněme si také toho, že traťové úseky jsou rozděleny i kilometrově. Výpočetní jádro neumí pracovat s ničím jiným než s těmito prvky, proto se do nového uzlu vkládá napájecí stanice označená NS2, která reprezentuje zkrat. Do tabulky se dále pro novou napájecí stanici vyplní napětí, které bude o hodnotě 0 V a také impedance, která bude shodná se zadanou impedancí zkratu od frontendu. Dále se musí vyplnit proudové omezení, které bude nastavené na hodnotu 99999 A. Posledním parametrem je charakteristika NS, která bude určena na hodnotu 0.

3.3.2) Metodika úpravy vstupních dat pro výpočetní jádro – zkratový režim SFC

Druhou úpravou, která probíhá ve zjednodušeném vývojovém diagramu, ale až po výpočtu, je převedení jedné, či více napájecích stanic do takzvaného zkratového režimu. Zkratový režim SFC znamená, že ve skutečnosti měnič nedokáže udržet takové napětí, aby dodával tak velký proud. V tu chvíli se z SFC místo napěťového zdroje stává zdroj proudový.

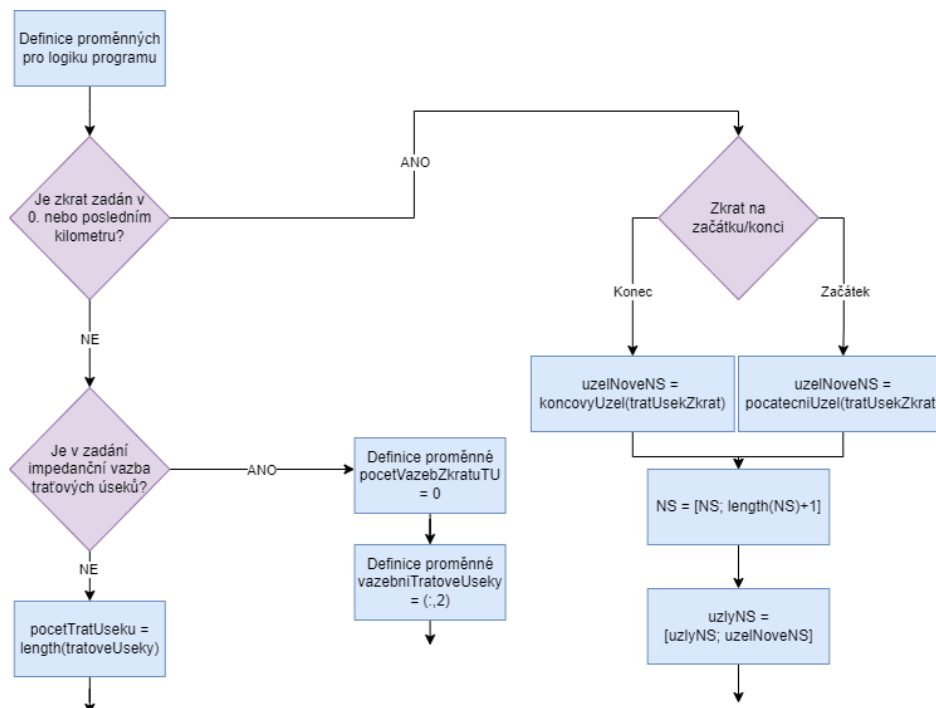
Abychom tuto změnu mohli správně do výpočtu vyjádřit, musíme i z napájecí stanice udělat tvrdý proudový zdroj. Změna se uskuteční tak, že do dotčených napájecích stanic se zapíše do sloupce napětí hodnota 8 MV a do sloupce impedance 10 k Ω . Pokud tyto hodnoty vydělíme dle Ohmova zákona, výsledkem bude proud o hodnotě 800 A. Tímto jsme do výpočtu zařadili velmi tvrdý zdroj.

Z důvodu, že nevíme jak vypadá průběh změny z napěťového do zkratového režimu, se tento přechodový děj ve výpočtu zanedbává. Na základě dostupných informací jsme určili, že změna režimu nastává tehdy, pokud napětí na svorkách měniče poklesne pod hodnotu 19,5 kV. S touto hodnotou tedy počítáme jako s referenční pro počátek přechodu režimů.

Znamená to tedy, že po prvním výpočtu musí logické jádro zkontrolovat výsledky. Pokud na nějaké z napájecích stanic, která je typu SFC, dojde k poklesu napětí pod hodnotu 19,5 kV, musí se celý příklad přepočítat s jedním, nebo s více novými proudovými zdroji.

3.3.3) Větvení logického jádra na základě podmínek – vzdálenost zkratu

Celé logické jádro pro výpočet jednotlivého zkratu se může svým chodem dělit na tři různé průchody. Tato větvení jsou zde hlavně z toho důvodu, že díky některým typům zadání se mohou, respektive musí vynechat určité kroky úprav.



Obrázek 10 - Větvení běhu programu

Na Obrázek 10 vidíme na základě určitých podmínek celkem 3 větvení. Tato větvení se nachází hned na začátku programu poté, co jsou zpracována všechna data ze vstupních tabulek.

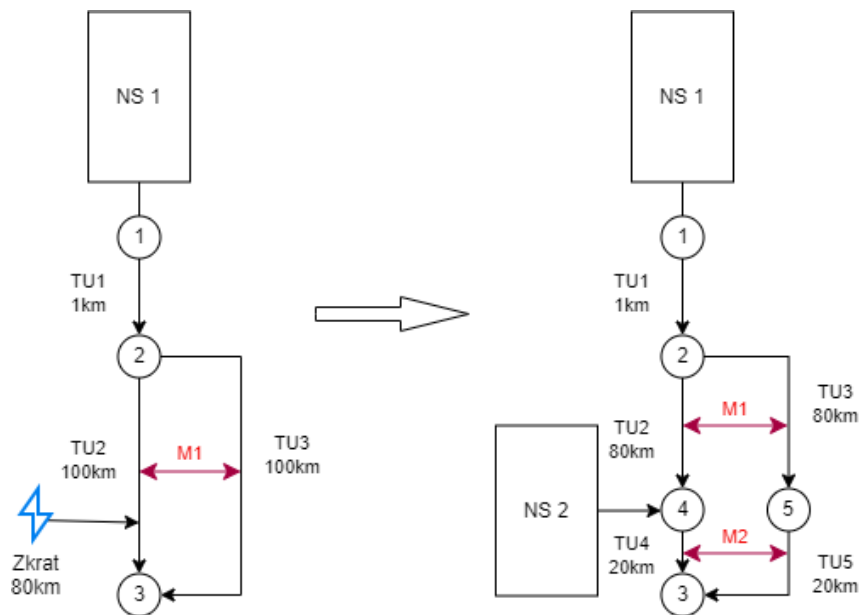
První větvení se vztahuje na kilometrickou polohu zkratu. Pokud je totiž zkrat umístěn na začátku nebo na konci traťového úseku, znamená to, že je umístěn v uzlu. Nemusíme tak tvořit nový uzel, ani rozdělovat původní traťový úsek na dvě části. Toto rozhodování se děje pouze na základě informace z druhé tabulky. V parametrech zkratu je jasně určený kilometr, ve kterém se zkrat nachází. Znamená to tedy, že pokud je zkrat zadán v nulovém kilometru, přecházíme rovnou na větev, která následuje za ANO. Druhou podmínkou pro rozhodnutí ANO je, že se zkrat nachází na konci traťového úseku. Zde se ale musí zkontrolovat, že kilometr zkratu je shodný s kilometry zkratovaného traťového úseku. Tato podmínka je s první podmínkou dohromady složena booleovskou logikou za pomoci NEBO:

```
if vzdalenostZkrat == 0 || vzdalenostZkrat == delkaTratUseku(tratUsekZkrat)
```

3.3.4) Větvení logického jádra na základě podmínek – vazební traťové úseky

Ke druhému větvení se během programu může dostat jen tehdy, pokud byl zkrat zadán mezi uzly, tedy na nějakém z traťových úseků. Druhé větvení je zde z důvodu impedanční vazby. Impedanční vazby mezi traťovými úseky jsou složitější záležitostí s ohledem na vytváření nového traťového úseku po rozdělení novým uzlem za účelem zkratu. Abychom lépe pochopili,

jak má nový traťový úsek s impedanční vazbou vznikat, uvedu na příkladu pro dvoukolejnou trať, která má impedanční vazbu mezi traťovými kolejemi.



Obrázek 11 - Příklad úpravy dvoukolejné trati

Na Obrázek 11 můžeme vidět, jak se oproti tratím, kde nejsou žádné vazební impedance, komplikuje celá úprava. V původním obrázku, který nakreslí uživatel, je pouze vložena vazba mezi traťové úseky TU2 a TU3. Při úpravě, kterou již bezpečně známe, opět vložíme na místo zkratu nový uzel. Rozdělíme trať na dva traťové úseky, přičemž druhému úseku přiřadíme nové číslo, tedy $TU_{\max} + 1$. Protože výpočetní jádro používá metodu uzlových napětí, musíme to stejné udělat i na vázaných kolejích k původnímu traťovému úseku TU2. Proto se na původním úseku TU3 vytvoří nový uzel v kilometrické poloze stejně, jako na TU2. Nesmíme zapomínat ani na vzájemnou impedanční vazbu, kdy mezi novými traťovými úseky TU4 a TU5 vznikne vazba se stejnou poměrnou impedancí, jakou má vazba M1.

Díky tomu, že v některých zadáních nejsou použity vazby, můžeme zjednodušit běh programu. Nemusíme řešit vytváření nových uzlů na vázaných traťových kolejích. Navíc není potřeba řešit ani vzájemné vazby, které díky tvoření nových uzlů a traťových úseků zcela narušíme. Proto vzniklo toto dělení, které automaticky odstraní některé prvky z běhu programu.

Rozhodnutí o tom, zda je v zadání vazba probíhá pouze na základě informace prázdného sloupce určení vazby. Pokud je sloupec pro určení vazby prázdný, ve vývojovém diagramu

na Obrázek 10 přejde program do větve NE. Pokud sloupec není prázdný, přechází do větve ANO. Podmínka pro rozhodování je vytvořena nativní funkcí MATLABu isempty():

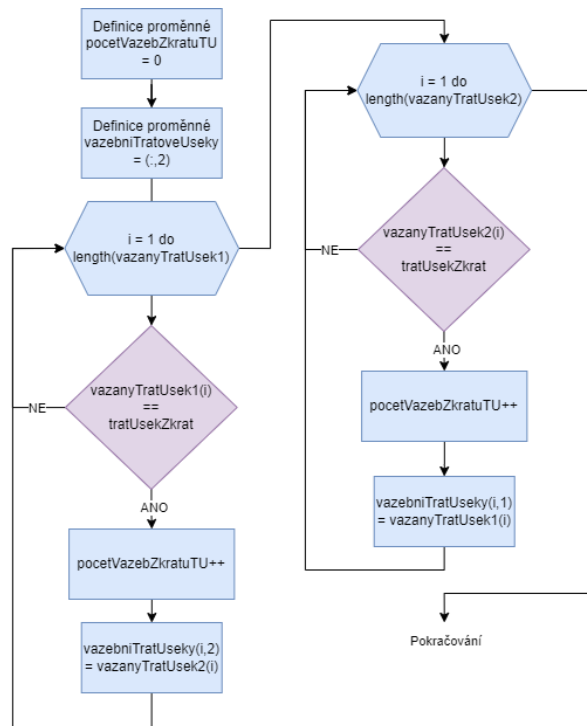
```
if isempty(vazanyTratUsek1)
```

3.4) Popis jednotlivých částí programu

Na základě určení jednotlivých funkcí a možností zadání byl napsán program logického jádra takovým způsobem, aby dokázal všechny tyto typy zadání zpracovat. Z důvodu různých typů zadání jsou tedy některé větve zjednodušeny a jsou jim ubrány některé funkční algoritmy, které logické jádro ve svém běhu používá. Abychom zbytečně nepopisovali jednotlivé větve, popíšeme pouze větev nejsložitější, která obsahuje všechny použité algoritmy. K nejsložitější větvi se program dostane pouze tehdy, pokud je zkrat umístěn mezi uzly na traťovém úseku a zároveň jsou v zadání použity vazební traťové úseky, kdy zkratovaný traťový úsek je navíc vázán vazbou k jinému traťovému úseku nebo jejich většímu počtu.

3.4.1) Počet nových traťových úseků i s vázanými traťovými úseky

Pokud se logické jádro dostane do stavu, kdy jsou v zadání i vazební traťové úseky, musí se nejdříve zjistit, kolik nových traťových úseků se bude vytvářet. Základní počet nových traťových úseků je jeden, který náleží zkratované traťové koleji. S každou další vazbou, která se díky použitému algoritmu zjistí, tento počet přibývá právě o jeden.

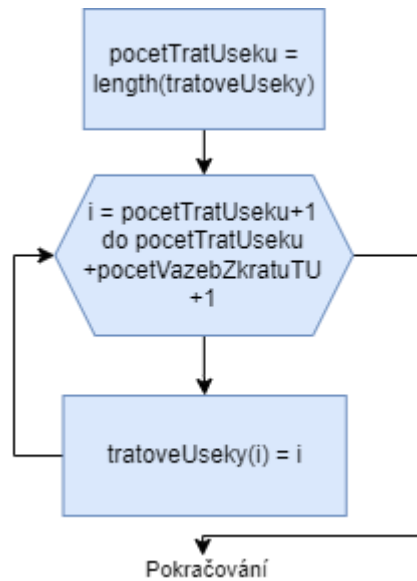


Obrázek 12 - Logika počtu traťových úseků i s vazbami

Zjištění počtu vázaných traťových úseků ke zkratovanému traťovému úseku probíhá tak, že se nejdříve kontroluje první sloupec pro čísla vázaných traťových úseků. Pokud se v tomto sloupci vyskytuje číslo zkratovaného úseku, inkrementuje se proměnná pocetVazebZkratuTU. Zároveň se do druhého sloupce proměnné vazebniTratUseky запиše číslo ve druhém sloupci. Toto číslo značí, s jakým číslem úseku je zkratovaný traťový úsek ve vazbě. Protože vazba může být čísla traťových úseků i obráceně, musí se zkontrolovat i druhý sloupec. Tentokrát se při objevení čísla zkratovaného traťového úseku zapisuje do prvního sloupce proměnné vazebniTratUseky. Jaký má tato dvourozměrná proměnná smysl si vysvětlíme v následujících odstavcích.

3.4.2) Vytvoření nových traťových úseků

Poté, co zjistíme nový počet traťových úseků, můžeme upravit první sloupec tabulky, tedy čísla jednotlivých traťových úseků. K úpravě použijeme jednoduchý algoritmus, který je zobrazen na vývojovém diagramu pod tímto odstavcem.

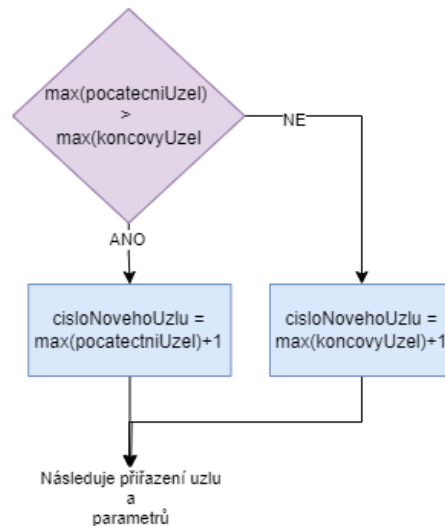


Obrázek 13 - Algoritmus pro nové traťové úseky

Protože v proměnné `tratoveUseky` jsou již data ze zadávací tabulky, zapíšeme do ní pouze nová čísla traťových úseků. Nová čísla jsou určena for cyklem, který začíná počítat od starého počtu traťových úseků + 1. Končí tehdy, když se dosáhne čísla, které je dáno součtem starého počtu traťových úseků, počtem vazeb ke zkratovanému traťovému úseku a +1 z důvodu nového traťového úseku vlivem rozdělení kvůli zkratu.

3.4.3) Přiřazení uzlu a parametrů zkratovanému traťovému úseku

Poté, co se vytvoří nové traťové úseky, můžeme přiřadit zkratované koleji nový uzel. Protože již víme, jaké číslo bude nést nový traťový úsek vlivem rozdělení, uzpůsobíme i tento nový úsek navazující na starý přes nový uzel.



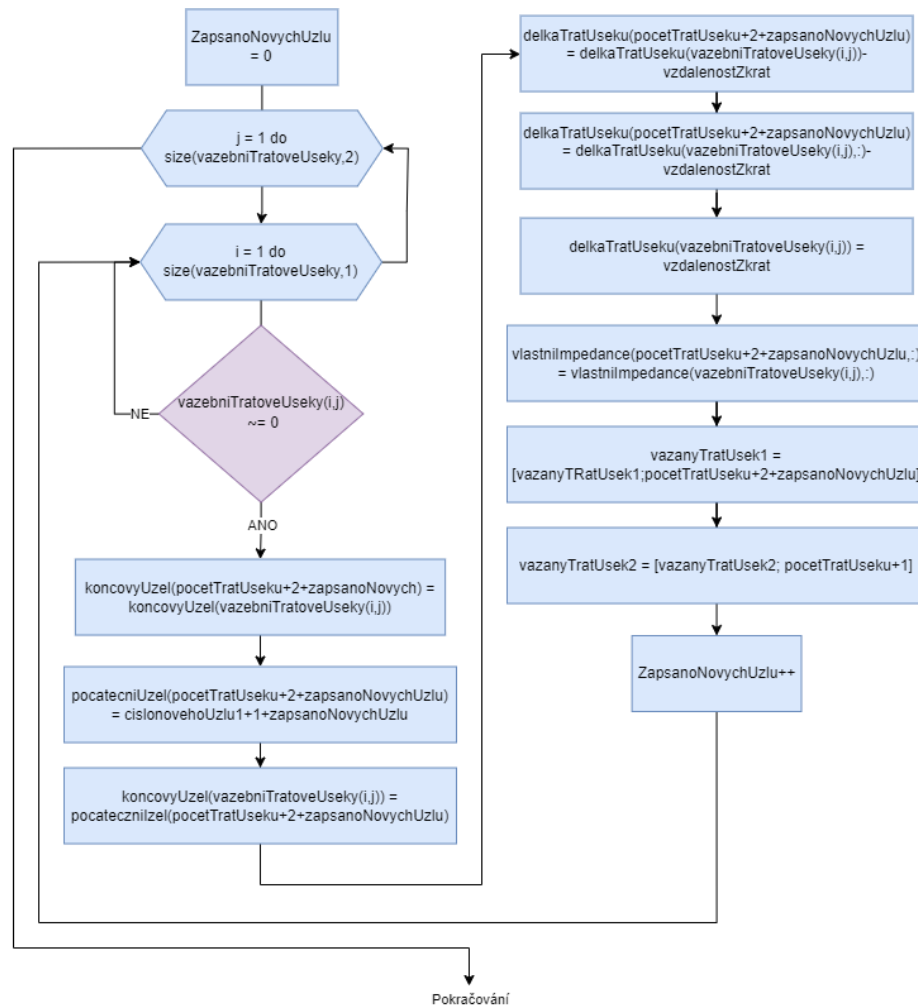
Obrázek 14 - Zjištění počtu uzlů ze zadání a následné přiřazení parametrů traťovému úseku

Nejdříve musíme zjistit nejvyšší číslo, které bylo uzlům přiřazeno. Protože nevíme, zda se nejvyšší číslo nachází v počátečních či koncových uzlech jednotlivých úseků, musíme zkontrolovat oba tyto sloupce. K nejvyššímu číslu uzlu se následně přičte jednička. Tím je nové číslo uzlu určeno.

Následuje proces, kdy je novému traťovému úseku přiřazen počáteční uzel, který nese číslo nového uzlu. Dále je mu přiřazen koncový uzel, který býval koncovým uzlem zkratovaného traťového úseku. Následně se ještě musí přepsat koncový uzel starého traťového úseku, který končí v nově vytvořeném uzlu. Poté proběhne přepsání délek jak zkratovaného, tak nového traťového úseku. Novému traťovému úseku se запиše délka starého traťového úseku, od kterého se odečte vzdálenost ke zkratu. Délka starého zkratovaného úseku je samozřejmě vzdálenost ke zkratu. Posledním parametrem je měrná impedance, která se pouze ze starého úseku překopíruje do nového úseku. Tento algoritmus je dobře viditelný na příkladu dvou vázaných kolejí, který je zobrazen na Obrázek 11.

3.4.4) Přiřazení uzlů a parametrů vázaným traťovým úsekům

Jeden z nejsložitějších algoritmů pro zprovoznění bylo rozdělení všech vázaných traťových úseků, které jsou přidruženy ke zkratovanému traťovému úseku. V době, kdy byl tento algoritmus programován, se ještě nevědělo, jaký limit bude pro vázané traťové úseky. Algoritmus je tedy psán obecně, aby zvládnul prakticky neomezené množství těchto vázaných traťových úseků.



Obrázek 15 - Přiřazení uzlů a parametrů vázaným traťovým úsekům

Protože v tomto i v dalším algoritmu potřebujeme vědět, kolik bylo zapsáno nových uzlů, je před začátkem algoritmu vytvořena proměnná ZapsanoNovychUzlu. V tomto algoritmu se dostává do hry proměnná vazebniTratoveUseky, která byla vytvořena a naplněna algoritmem pro vyhledávání vazeb se zkratovaným traťovým úsekem.

Celý algoritmus prohledává proměnnou vazebniTratoveUseky tak, že vezme nejdříve sloupec a zkontroluje všechny řádky. Při kontrole jedné proměnné v tomto poli se ověří, zda není buňka prázdná. Pokud není prázdná, postupuje se v algoritmu na ANO pod touto rozhodovací podmínkou. Následují jednotlivá přiřazení nových uzlů k traťovému úseku. Číslo tohoto úseku známe právě z proměnné vazebniTratoveUseky, kam jsme čísla těchto úseků zapisovali. Podle této proměnné víme, na kterém řádku se traťový úsek nachází, a můžeme ho libovolně upravovat.

Nejdříve se zapíše koncový uzel pro nový traťový úsek, který známe z informace o předchozím traťovém úseku. Následuje zapsání i počátečního uzlu pro nový traťový úsek, což je právě nový uzel, který byl vytvořen. Zbývá již jen zapsat nový koncový uzel i pro starý traťový úsek, který je číslem stejný, jako počáteční uzel nového traťového úseku. Následuje úprava délek traťových úseků, kdy starý traťový úsek bude mít délku ke zkratu a nový bude mít zbytek délky. Poté je potřeba zkopírovat i vlastní impedanci ze starého do nového traťového úseku. Protože se nacházíme v úpravě vazebních traťových úseků, je potřeba upravit i vazby. To probíhá tak, že se vezme nově vytvořený traťový úsek vazby a přidá se vazba k nově vytvořenému traťovému úseku na zkratované koleji.

Po skončení všech úprav vázaného traťového úseku a jeho vazby k nově vytvořenému zkratovanému úseku se ještě inkrementuje proměnná `ZapsanoNovychUzlu`. Tím dáme dalšímu průběhu smyčky, ve kterém bude potvrzena další vazba, informaci o tom, že má zvýšit o jedno číslo jak další nově vytvořený uzel, tak i nový traťový úsek. Tím docílíme toho, že se identifikátory pro traťové úseky a uzly nebudou opakovat.

3.4.5) Vytvoření nové napájecí stanice na místě zkratu

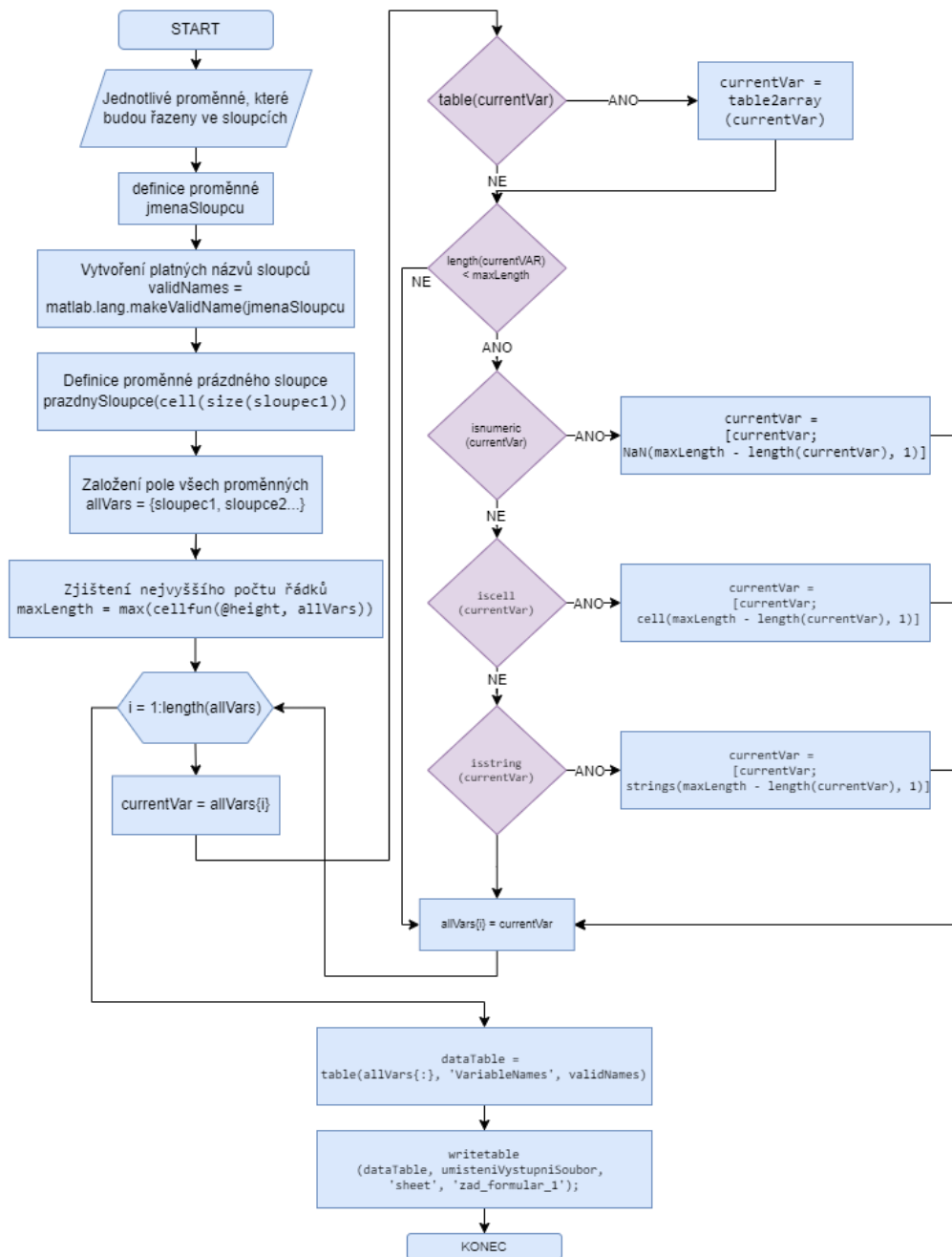
Poslední úpravou, která se v první části programu děje, je vložení nové napájecí stanice do uzlu zkratu. Jak již bylo zmíněno, tato napájecí stanice má nulové napětí a její impedance je rovna impedanci zkratu. Slouží tedy pouze pro tyto účely a její parametry jasně vyjadřují, že trať nenapájí.

Pro zapsání nové napájecí stanice a jejích parametrů si nejdříve musíme zjistit, kolik NS se již v zadávací tabulce nachází. Tento počet inkrementujeme a určíme tak identifikační číslo pro novou NS. Následně zapíšeme její impedanci, tedy impedanci zkratu. Dalším parametrem je napětí, které bude rovno nule. Proudové omezení je stanoveno na 99999 A. Posledním parametrem je charakteristika zdroje, kterou nastavíme na hodnotu nula.

3.4.6) Přenesení programových proměnných do výstupní tabulky a její export

Transformace jednotlivých proměnných do výstupní tabulky je napsána v souboru funkce, který nese název `VytvorTabulku`. Do této funkce se přenáší proměnné všech sloupců, které jsou buď definovány jako konstanty v logickém jádře, naimportovány nebo upraveny

logickým jádrem. Také se do této funkce posílá proměnná, která definuje umístění a název výstupního souboru ve formátu xlsx.



Obrázek 16 - Vývojový diagram funkce VytvorTabulku

Funkce vytvoř tabulku se spustí až po zavolání logickým jádrem. Jedinou proměnnou, která je v této funkci definována, je jmenaSloupce. Z těchto jmen se musí vytvořit platné názvy pro tyto sloupce, a to z důvodu vnitřních algoritmů MATLABu. MATLAB totiž nedovoluje mít v těchto názvech háčky, čárky a další speciální znaky. Posledním krokem před hlavním

algoritmem této funkce je založení pole, které obsahuje všechny proměnné. Tato proměnná se nazývá `allVars`, aby bylo z názvu patrné, co obsahuje. Poté se zjistí i maximální počet obsazených řádků ve všech proměnných a ten se zapíše do proměnné `maxLength`.

Hlavní algoritmus této funkce je tvořen cyklem `for`. V principu cyklus prochází jednotlivé sloupce a doplňuje do nich „prázdné“ řádky. Na začátku cyklu se do proměnné `currentVar` vybere jeden sloupec z `allVars`. Následuje podmínka, že pokud je aktuální proměnná (`currentVar`) tabulka, musíme ji z důvodu následujícího doplnění prázdných buněk převést na vektor. Poté, co se tento sloupec převede, nebo nepřevede, následuje podmínka, která zjistí počet řádků v aktuálně vybrané proměnné. Pokud je těchto řádků méně než v nejdelším sloupci, následují podmínky `if` a `elseif`, které mají za úkol tyto prázdné řádky doplnit. Chybějící řádky do jednotlivých sloupců jsou doplněny, protože při převádění proměnné `allVars` na proměnnou typu `table` musí být všechny sloupce pojmenované a také stejně dlouhé. Pokud by sloupce nebyly stejně dlouhé, nikdy bychom nemohli proměnnou převést na tabulku.

Pokud funkce zjistí, že sloupci chybí řádky, začne se dotazovat na typ proměnné, která je ve sloupci uložena. Dohromady máme zobecnění na tři typy proměnných. Když funkce zjistí, že je proměnná číselného typu, dosadí do chybějících řádku sloupce konstantu `NaN`. Tím se zajistí, že výpočetní jádro nebude s těmito čísly pracovat. Kdybychom totiž uložili do prázdných řádků nulu, mohlo by výpočetní jádro s nulou počítat jako s dalším číslem. Druhým typem proměnné je obecná buňka, což souvisí s tím, že jsme tabulku převedli na vektor. Zde se na chybějící řádky pouze doplní prázdná buňka. Posledním typem proměnné je `string`. K těmto prázdným řádkům se doplní prázdná proměnná typu `string`, kdy taktéž není potřeba ošetřovat, aby výpočetní jádro s těmito prázdnými proměnnými nepracovalo. Tento algoritmus se opakuje do té doby, dokud neprojde všechny sloupce, a následně skončí.

Posledními operacemi této funkce je převedení všech sloupců na proměnnou typu `table`. Funkci `table` se musí určit jakou proměnnou je potřeba převést. Dále je také nutné určit parametr „`VariableNames`“, který určuje jména jednotlivých sloupců. Za tento atribut dosadíme proměnnou, ve které jsou uložena validní jména, což je proměnná `validNames`. Nakonec se vyexportuje tabulka ve formátu `.xlsx`, která bude uložena na místo definované logickým jádrem a její list se bude jmenovat `zad_formular_1`. Tím se funkce ukončí a program se vrátí

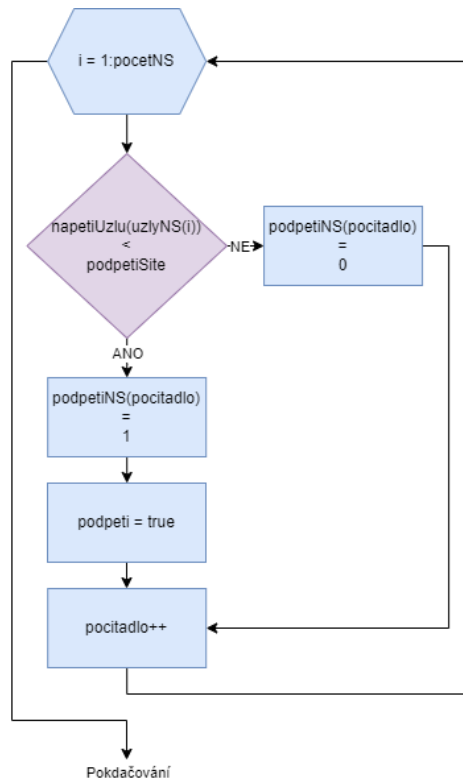
svým během zpět do logického jádra. V logickém jádře následně proběhne už jen volání výpočetního jádra, které je definováno pouze jeho umístěním v konstantách.

Tato funkce není zcela dokonalá. V budoucnu by se mohla dočkat nějakých úprav, a to hlavně z pohledu její univerzálnosti. Příkladem může být i to, že by se do funkce musela posílat i jména jednotlivých sloupců. Pokud by počet sloupců neseseděl s počtem jmen těchto sloupců, mohla by funkce generovat chybu, která by se mohla programátorovi zobrazit. Další krok v univerzálnosti funkce by mohl být i ten, že by se do ní mohlo posílat neomezené množství sloupců, které by taktéž proběhly kontrolou s počtem jmen sloupců. Funkce aktuálně není zcela samostatná a pokud se upraví výstupní uspořádání výstupní tabulky, musí se upravit nejen logické jádro, ale i tato funkce.

3.4.7) Kontrola a zpřesnění výsledků z výpočetního jádra

Po ukončení procesu výpočetního jádra je potřeba zkontrolovat jednotlivá napětí měničových napěťových stanic, zda u nich nedošlo k podpětí a tedy k přechodu do zkratového režimu. Nejdříve si načteme tabulku výsledku a poté inicializujeme proměnné.

První proměnnou, abychom mohli vyčítat z tabulky výsledků, je počet sloupců, které je potřeba přeskočit a vůbec nečíst. Tyto sloupce mají konstantní počet, protože obsahují informace o vlaku, který je jasně definován upraveným vstupním souborem a také určité konstantní informace. Druhou proměnnou, kterou definujeme, je aktuální počet uzlů, tedy už i s uzly, které předtím přidalo samotné výpočetní jádro. Znovu rozhodujeme na základě podmínky, jako jsme to dělali u určení celkového počtu uzlů ze zadání, tedy na Obrázek 14.



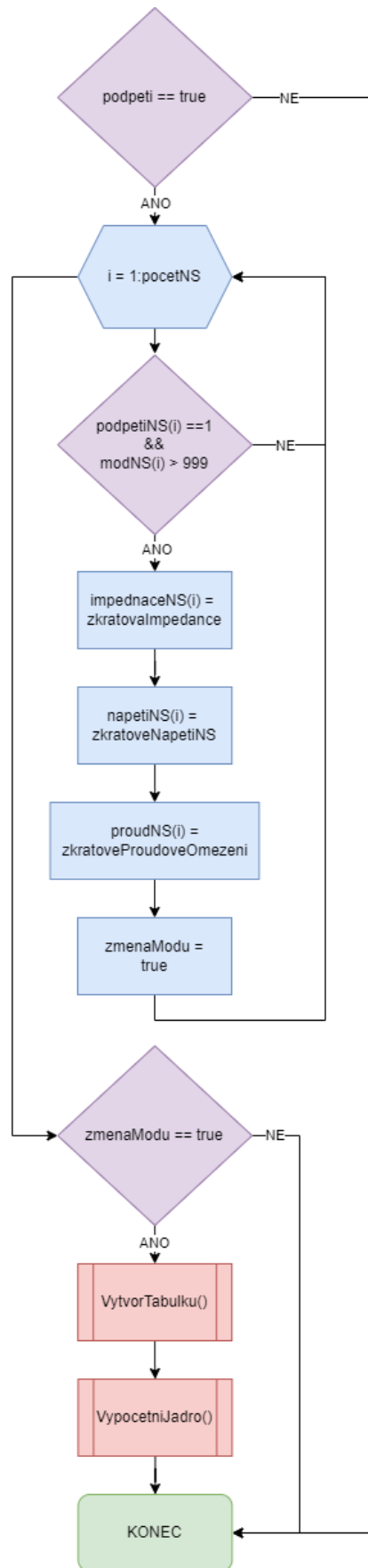
Obrázek 17 - Vývojový diagram rozhodování o podpětí NS

Dalším krokem je již čtení výsledků napětí uzlů, ve kterých jsou umístěny jednotlivé NS. Tento algoritmus je popsán na Obrázek 17. V tomto kroku se ale ještě neví, zda jsou to NS s měničovou technologií, či jen transformátory, které samozřejmě do zkratového režimu nepřechází. Jak můžeme vidět, jednotlivá čísla uzlů, ve kterých jsou umístěny NS, jsou zapsána v proměnné uzlyNS. Naopak v proměnné napetiUzlu jsou zapsána jednotlivá vypočtená napětí. Pokud je napětí v daném uzlu menší než konstanta podepetiSite, do proměnné podpetiNS se zapíše číslo 1. Následně se změní příznak podpeti na hodnotu TRUE a počítadlo se inkrementuje. Pokud na dané NS nebylo podpětí, zapíše se do dané proměnné hodnota 0 a počítadlo se taktéž inkrementuje.

Po rozhodnutí o podpětí v uzlu NS může buď program skončit, nebo provést zpřesnění výsledku výpočtů na základě dalších rozhodovacích podmínek. Pokud předchozí algoritmus na Obrázek 17 vyhodnotil, že na některé z NS podpětí vzniklo, počáteční podmínka v algoritmu na Obrázek 18 rozhodne o provedení cyklu. Jak je vidět na prvním rozhodovacím symbolu v cyklu, podmínky pro přejítí do zkratového režimu jsou takové, že u dané NS muselo vzniknout podpětí a zároveň číslo módu pro danou NS musí být větší, než 999. Těmito podmínkami se vyloučí přepsání parametrů transformátorů, které z podstaty funkce

nemohou do zkratového režimu přejít. Pokud daná NS splní obě tyto podmínky, přepíše se její impedance na zkratovou impedanci, její napětí na zkratové napětí a také její proudové omezení na zkratové proudové omezení. Poslední akcí po rozhodnutí o změně módu NS se také přepíše pomocná proměnná, která indikuje změnu módu některé z napájecích stanic.

Další podmínka je navázána na indikátor změny módu NS, který rozhoduje o tom, zda bude či nebude znovu proveden výpočet pro zpřesnění výsledků. Pokud bude, vytvoří se tabulka s novými parametry pro NS a znovu se zavolá výpočetní jádro. Po tomto výpočtu už logické jádro nic nekontroluje a rovnou skončí. Výsledky z výpočetního jádra jsou uloženy do tabulky, kterou dostane k dispozici i frontend, který tyto výsledky zobrazí.



Obrázek 18 - Algoritmus pro zpřesnění výsledků na základě vyhodnocení podpětí z přechozího algoritmu

3.5) Testování součinnosti logického a výpočetního jádra na základě výsledků simulací porovnaných s výsledky zkratových zkoušek

Testování probíhalo v době, kdy ještě nebyl k dispozici frontend. Součinnost frontendu a backendu bude vyzkoušena až někdy v budoucnu, kdy se do frontendové aplikace budou kreslit zadání z tohoto testování a výsledky se budou porovnávat taktéž s výsledky nynějších testů.

Prvním krokem bylo tedy vždy vytvořit zadání takovým způsobem, že jsem musel vyplnit vstupní tabulku do logického jádra, kterou při normálních operacích generuje frontend. Jaké informace tabulka obsahuje, již bylo v této práci zmíněno. Poté se musela odladit spolupráce mezi logickým a výpočetním jádrem. Při některých zadáních se stávalo to, že některé proměnné byly zadané ve špatném formátu vlivem přepisování vstupních hodnot logickým jádrem. Dalším problémem bylo spouštění výpočetního jádra, protože když byl spouštěn jako další program, prostředí MATLABu vymazalo všechny proměnné z logického jádra. Problém byl vyřešen tak, že výpočetní jádro bylo přepsáno pouze na funkci.

Testování probíhalo nejdříve na jednoduchých příkladech typu zkratu na jednokolejných tratích. Poté se přecházelo na dvoukolejné tratě s vazební poměrnou impedancí těchto dvou kolejí. Tyto příklady nebudu do této diplomové práce uvádět, protože na nich byl odladěn běh programu a součinnost obou aplikací. Posledním testováním, které do této práce zahrnu, bylo vytvoření reálných tratí do zadání. Na těchto reálných tratích byly provedeny zkratové zkoušky, které nám pomohou určit přesnou odchylku od reálně změřených hodnot. Také nám pomohou dokázat, že obě části fungují správně dle předpokladů programátorů.

3.5.1) Představení použité tratě pro testování programu

Pro testování byla použita trať Říkovice – Nedakonice, která byla rozdělena na dvě části ve stanici Otrokovice. Na trati Otrokovice – Říkovice se nachází dva zdroje typu SFC, které mohou být vlivem synchronizace výstupních napětí spojeny do sebe. Tohoto spojení se ale při zkratových zkouškách nevyužívalo a testovala se pouze jednotlivá SFC jakožto jednostranné napájení.

Na trati Otrokovice – Nedakonice se také nachází dva zdroje, ale v Nedakonicích je jednofázový transformátor napájený ze dvou sdružených napětí. Na výstupní napětí transformátoru se dokáže měnič také synchronizovat. Toho se u zkratových zkoušek využilo a byly provedeny zkraty na oboustranně napájeném úseku.

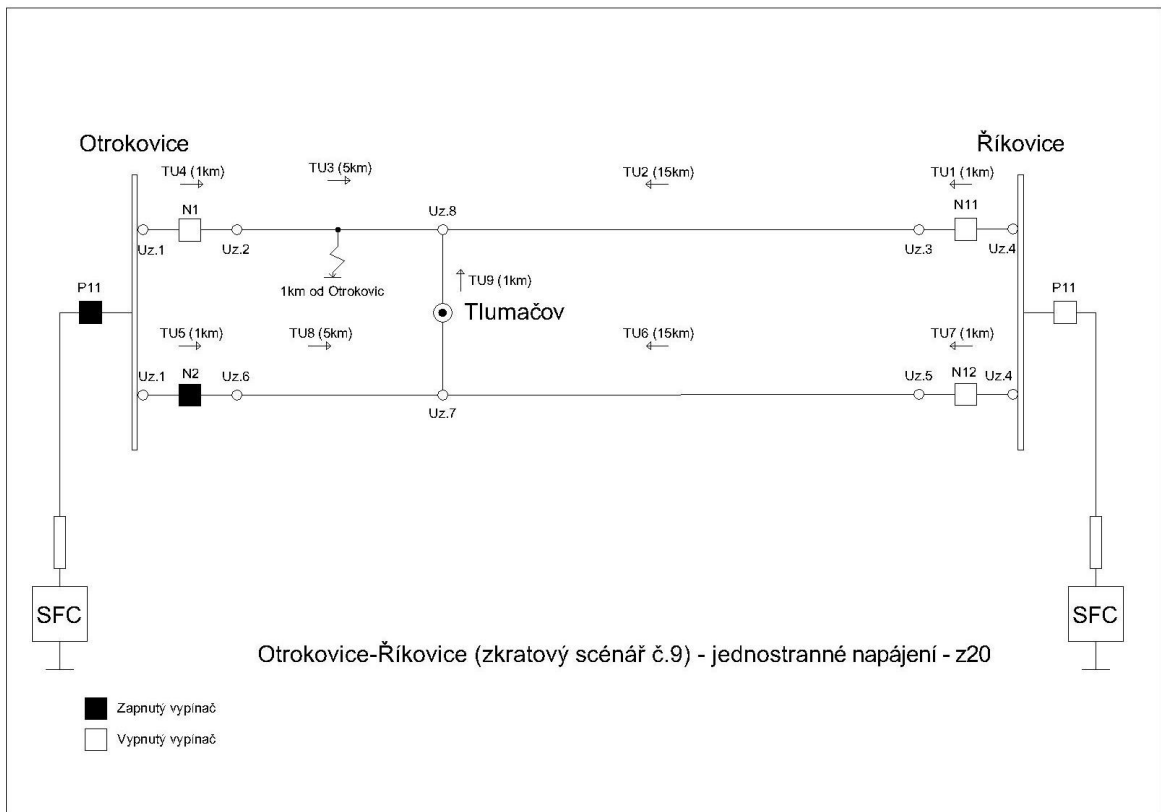
Dohromady byl program otestován na jedenácti zkratech. Následně se všechny hodnoty z výpočetního jádra porovnal s hodnotami změřenými u zkratových zkoušek. V této práci uvedu pouze nejzajímavější zkraty z pohledu programu, jako celku. Vynechám jednoduché úlohy, kdy je zdroj přímo připojený přes jeden vypínač do zkratovaného traťového úseku.

3.5.2) Nejzajímavější testovací příklady jednotlivých zkratů pro logické jádro

Mezi první testované příklady patří zkraty na trati Otrokovice – Říkovice, kde jsou vždy dvě traťové koleje, které jsou po celou kilometrickou délku vedle sebe. Dále se v pátém kilometru od stanice Otrokovice ve stanici Tlumačov nachází příčné dělení, které bylo při zkratových zkouškách využito. Z této části tratě vybírám pouze dvě zkratové zkoušky, které jsou ideálním reprezentantem pro trochu složitější testování logického jádra.

Testování probíhalo tak, že se nejdříve musela vytvořit tabulka zadání, kde jsou přesně definovány traťové úseky, uzly a jejich poměrné či vzájemné impedance. V diplomové práci Ing. Ladislava Cermana se již přímo těmito tratěmi zabýval, tudíž jsem musel taktéž použít pro hodnoty impedancí z příručky SR34 [7] korekční konstantu, která pro tuto trať činí 0,7. Tato korekční konstanta je zde z důvodu použití atypické sestavy trakčního vedení. Jak se dopočítat korekční konstanty také popisuje diplomová práce [1]. Impedance traťových kolejí ve všech zadáních byla stanovena na $0.182+0.385j \Omega$. Ještě je nutné zmínit to, že pokud je v traťovém úseku vypínač vypnutý, zapisuje se do tohoto traťového úseku hodnota impedance $1000000+0j \Omega$. Naopak pokud je vypínač v daném TU zapnutý, zapisuje se hodnota impedance $0.001+0j$.

Výsledky testování můžeme určit ze dvou úhlů pohledu. První pohled cílí pouze na logické jádro, kdy se díváme na korektnost úpravy zadaných úseků. Druhým pohledem je již výsledek z backendu, který má jen číselně vyjádřené hodnoty. Oba tyto pohledy byly v každém případě zkontrolovány. Zajímavější je ale pouze pohled na přesnost výpočetního jádra, kdy se zároveň testují i algoritmy logického jádra.



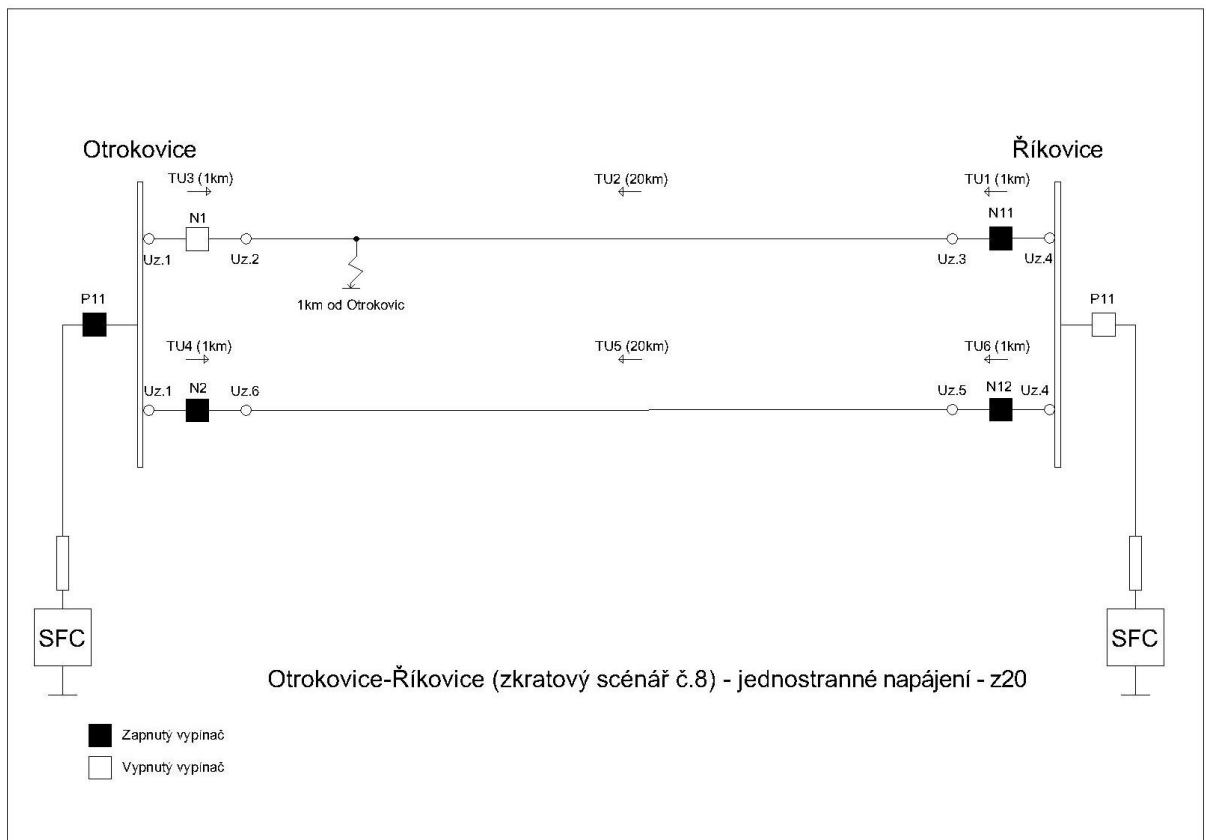
Obrázek 19 - Zkratový scénář č. 9 na trati Otrokovice – Říkovice [10]

Ve zkratovém scénáři na Obrázek 19 vidíme, že napájení probíhalo z TNS Otrokovice a zkrat byl vyvolán přes příčné dělení na druhou traťovou kolej. Jak je dále patrné, zkratový proud by měl tedy téci z uzlu 1 přes TU5, TU8, TU9 a TU3 směrem ke zkratu. Aby byl zkrat zaznamenán backendem, musí logické jádro rozdělit TU3 na 2 traťové úseky, kdy mezi TU3 a nový TU10 vloží nový uzel s číslem 10. Do tohoto uzlu je vložena nová NS s číslem 3, která simuluje zkrat.

Tabulka 1 - Výsledky impedance Otrokovice-Říkovice, zkratový scénář č. 9

Výsledky ze zkratových zkoušek EŽ	
$ Z $ [Ω]	4,17
ϕz [$^\circ$]	64,06
Výsledky ze simulace	
$ Z $ [Ω]	2,91994226
ϕz [$^\circ$]	59,0844257
Odchylka vypočtených hodnot	
$ Z $ [%]	30,0
ϕz [%]	7,8

Nedílnou součástí této úlohy jsou právě induktivně vázané traťové úseky. Tyto induktivní vazby, které jsou určeny uživatelem programu, zpřesňují výpočet. U tohoto příkladu byla ale odchylka nejvyšší, a to ne v úhlu impedance, ale hlavně v modulu impedance, který je o 30 % menší než skutečná naměřená hodnota. Protože tento příklad je jediným zkratem, pro který bylo použité příčné propojení traťových kolejí, je možné, že na odpojovačích v tomto dělení byly velké přechodové odpory. Dalšími možnostmi pro nezpřesněné výsledky mohou být také špatně zapsané hodnoty vzdálenosti jednotlivých traťových kolejí do protokolu [10], kdy máme následně takto velkou odchylku od reálně naměřené impedance. Na základě tak velké odchylky byla zkontrolována výstupní tabulka z logického jádra, která neobsahovala žádné chyby. Všechny směry traťových kolejí, které jsou impedančně vázané, byly na stejnou stranu. Všechny impedančně vázané úseky měly správně definovanou vazební indukčnost. Hlavním řešitelem projektu bylo stanoveno, že maximální dosažená odchylka může být 30 %, což i v tomto zvláštním případě program zcela splňuje. Důležité je také zmínit, že konečnou přesnost simulovaného výpočtu určují hlavně parametry, které musí uživatel správně zadat. Odchylka pro úhel impedance činila pouze 7,8 %, což je hluboko pod hranicí dovolené odchylky.



Obrázek 20 - Zkratový scénář č. 8 na trati Otrokovice – Říkovice [10]

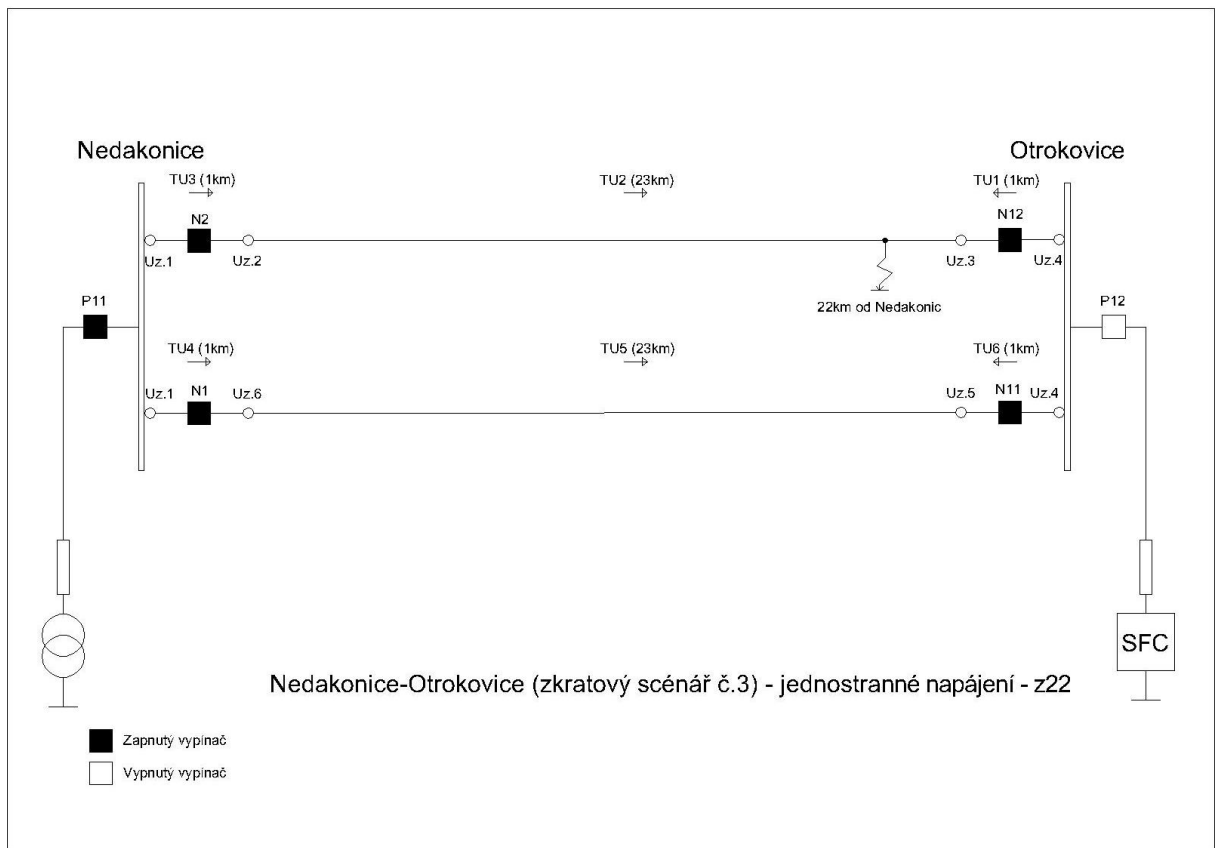
Další příklad se taktéž nachází na trati Otrokovice – Říkovice. Zde jsme ale rozepnuli příčné dělení ve stanici Tlumačov a sepnuli vypínače N11 a N12 v TNS Říkovice. Zkratový proud by měl téci právě přes přípojnicí vedlejší stanice. Vypínač P11 v Říkovicích je vypnutý, tudíž znovu máme pouze jednostranné napájení.

V zadávacím souboru byly odmazány uzly 7 a 8. Také se již v zadávacím souboru nenachází TU č. 3 a 8, kdy se TU č. 3 pouze přesunul na vypnutý vypínač N11. Dále se TU2 prodloužilo na celých 20 km délky stejně, jako se TU5 přesunul z vypínače na širou trať. Muselo být přepsáno i zadání zkratu, kdy je jeho umístění na TU2 v 19. kilometru. Při tomto zkratu byl pořízen snímek, který je umístěn v příloze A.

Tabulka 2 - Výsledky impedance Otrokovice – Říkovice, zkratový scénář č. 8

Výsledky ze zkratových zkoušek EŽ	
Z [Ω]	11,18
φz [°]	64,01
Výsledky ze simulace	
Z [Ω]	12,1
φz [°]	59,87
Odchylka vypočtených hodnot	
Z [%]	-8,2
φz [%]	6,5

V tomto scénáři je již odchylka od změřených hodnot zkratovými zkouškami velmi malá. Co se týče velikosti impedance, simulace udává o 8,2 % vyšší impedanci než změřenou. Naopak vypočítaný úhel impedance je nižší o 6,5 %. Tento příklad je tak zcela v souladu s požadavky na přesnost. Navíc jsme vyzkoušeli, že si logické jádro s výpočetním jádrem rozumí, i když proud teče přes více traťových úseků, a to dokonce i přes takové traťové úseky, které začínají ve stejném uzlu, ale končí v jiných uzlech. Pokud mají dva či více traťových úseků stejný počáteční nebo koncový uzel, můžeme tento uzel většinou nazývat přípojnicí napájecí stanice či jiného objektu.



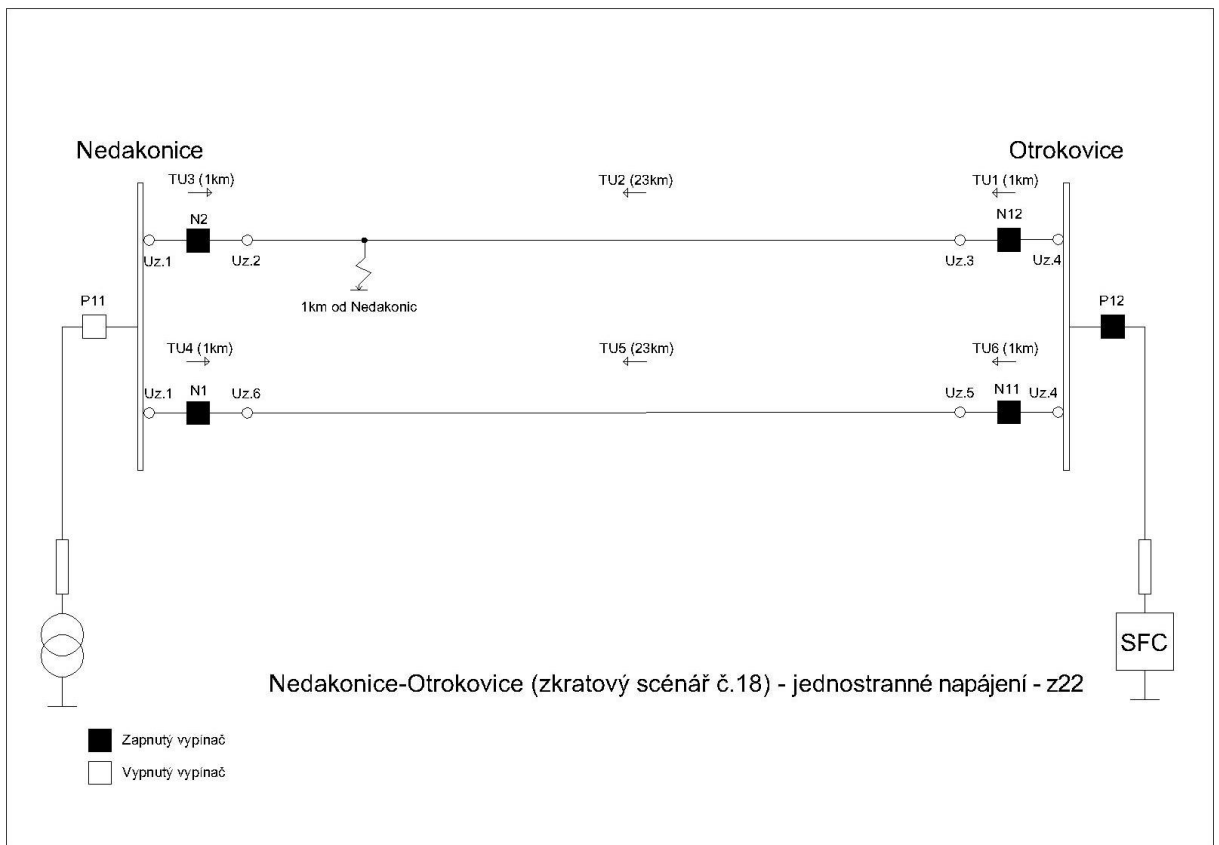
Obrázek 21 - Zkratový scénář č. 3 na trati Nedakonice – Otrokovice [11]

Všechny další, pro logické a výpočetní jádro, zajímavé zkraty se uskutečnily na trati Nedakonice – Otrokovice. Zkratový scénář č. 3 byl zkonfigurován tak, že probíhalo jednostranné napájení z TNS Nedakonice, kdy napájel jednofázový transformátor. Aby byla nižší impedance ke zkratu, napájely se obě koleje přes vypínače N1 a N2. Navíc bylo napětí prosmyčkováno přes přípojnicí v Otrokovicích, kde byly sepnuty vypínače N11 a N12. Ke zkratu tekla proud oběma kolejemi, přičemž proud z TU5 tekla dále přes přípojnicí TNS Otrokovice. Zkrat se nachází ve 22. kilometru TU2. U tohoto zkratu nemůže dojít k přechodu do zkratového režimu, protože transformátor zkratový režim nemá.

Tabulka 3 - Výsledky impedance Nedakonice – Otrokovice, zkratový scénář č. 3

Výsledky ze zkratových zkoušek EŽ	
$ Z $ [Ω]	12,15
ϕz [$^\circ$]	75,25
Výsledky ze simulace	
$ Z $ [Ω]	11,9
ϕz [$^\circ$]	66,21
Odchylka vypočtených hodnot	
$ Z $ [%]	2,0
ϕz [%]	12,0

Po ověření výstupní tabulky logického jádra, kdy nemůže dojít k přechodu do zkratového režimu se mohly znovu revidovat výsledky výpočtu výpočetního jádra. Tentokrát můžeme velikost impedance prohlásit za zcela shodnou, i když je odchylka 2 %. V tomto případě, kdy má velký vliv na úhel impedance impedanční vazba, se dostáváme vždy na větší odchylku právě v úhlu impedance. Úhel je znovu v simulaci nižší, tentokrát o 12 %.



Obrázek 22 - Zkratový scénář č. 18 na trati Nedakonice – Otrokovice [11]

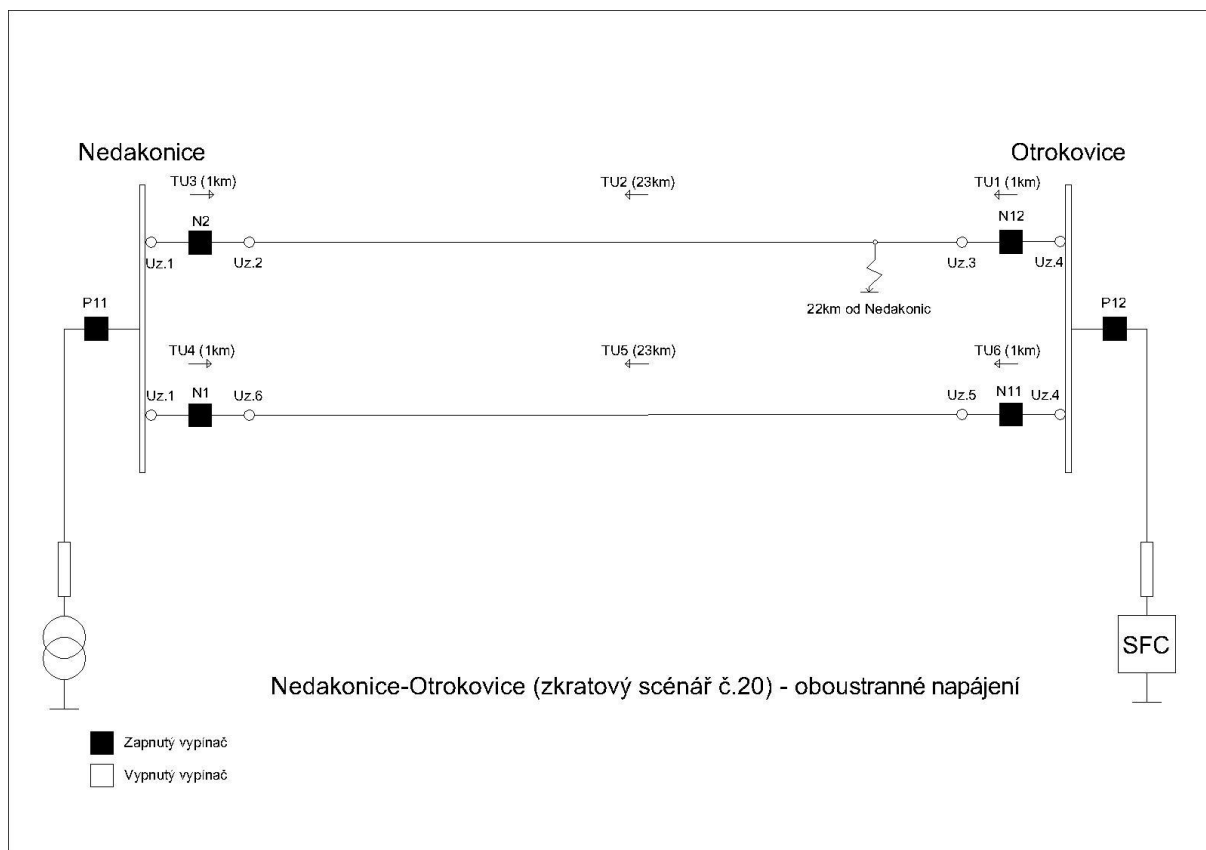
Zkratový scénář č. 18 je velmi podobný předchozímu zkratu. Na rozdíl od minulého scénáře se napájelo z TNS Otrokovice, tedy z SFC. Znovu byla trať konfigurována tak, že napětí bylo prosmyčkováno přes přípojnicí v TNS Nedakonice, kde byly sepnuty vypínače N1 a N2. V tomto scénáři také nejde o dvoustranné napájení traťových úseků, z důvodu vypnutého vypínače P11. Tímto scénářem vyzkoušíme, že není problém, ani když proud teče oboustranně ke zkratu z SFC.

Tabulka 4 - Výsledky impedance Nedakonice – Otrokovice, zkratový scénář č. 18

Výsledky ze zkratových zkoušek EŽ	
Z [Ω]	12,655
ϕ_z [$^\circ$]	70,9523
Výsledky ze simulace	
Z [Ω]	11,9
ϕ_z [$^\circ$]	66,21
Odchylka vypočtených hodnot	
Z [%]	6,0
ϕ_z [%]	6,7

Jak si můžeme všimnout, výsledek simulace je úplně stejný jako v přechodím scénáři. Prakticky je to pro simulaci ten stejný zkrat, jen z druhé strany. Je tedy zřejmé, že SFC do zkratového režimu nepřešlo, tudíž se chovalo stejně jako transformátor. Obecně chování SFC by se mělo zlepšit, na základě možnosti měnit charakteristiku SFC před simulací jednotného zkratu. Tudíž by výsledky neměly být totožné, pokud se zásadně změní charakteristika měniče.

Odchylka od reálných hodnot je v tomto příkladu pro velikost impedance větší než v příkladu předchozím. Naopak odchylka úhlu impedance se zmenšila. Protože hodnoty simulace porovnáváme s hodnotami změřenými, tyto změřené hodnoty nemusí být vždy 100% přesné, stejně jako naše simulace. Je tedy jasné, že nejdůležitější změnou v procentuálním vyjádření odchylky hrají v tomto případě právě změřené hodnoty.



Obrázek 23 - Zkratový scénář č. 20 na trati Nedakonice – Otrokovice [11]

Posledním a zároveň nejzajímavějším příkladem, který představím, je zkrat při oboustranném napájení. Vlivem stále se zvětšujícího počtu měničů v síti bude toto spojení častější. Někdy se bude jednat pouze o oboustranné napájení jedné trati, kdy na druhé straně napájení není SFC, ale transformátor. V druhém případě mohou napájet dva či více měničových zdrojů proti sobě. Protože ještě neproběhly žádné zkratové zkoušky pro nafázovaná SFC, nemáme simulované hodnoty s čím porovnat.

Tabulka 5 - Výsledky impedance Nedakonice – Otrokovice, zkratový scénář č. 20

Výsledky ze zkratových zkoušek EŽ	
$ Z $ [Ω]	12,017
ϕz [$^\circ$]	75,45
Výsledky ze simulace	
$ Z $ [Ω]	12,65
ϕz [$^\circ$]	53,84
Odchylka vypočtených hodnot	
$ Z $ [%]	-5,3
ϕz [%]	28,6

Výsledky zkratových zkoušek jsou velmi podobné výsledkům ze zkratového scénáře č. 3. I dle Obrázek 23 - Zkratový scénář č. 20 na trati Nedakonice – Otrokovice byl zkrat umístěn na stejném místě, akorát byl navíc sepnut vypínač P12 ve stanici Otrokovice. Vlivem oboustranného napájení se nám ale velmi pokazila procentuální odchylka úhlu impedance. Při jednostranném napájení z SFC byl úhel dle simulace $66,21^\circ$. Při oboustranném napájení tento úhel při simulaci poklesl na hodnotu $53,84^\circ$. To může být způsobeno tím, že zkrat je velmi blízko měničovému zdroji, kdy se tolik neuplatňuje imaginární složka impedance, ale spíše převažuje reálná impedance. Díky tomu se sníží i úhel impedance. I přes všechny tyto nesrovnalosti se stále dostáváme pod maximální odchylku 30 %, což je skvělá zpráva.

Díky tomuto testování jasně víme, že logické i výpočetní jádro pracují dobře. Pouze v jednom případě byla odchylka tak velká, že dosahovala hodnoty 30 %. Tuto odchylku ale není možné ověřit, protože příčné propojení ve zkratovém scénáři Říkovice – Otrokovice č. 9 bylo změřeno pouze jednou. Většinou se odchylka pro velikost impedance pohybuje okolo jednotek procent. To stejné platí i pro odchylku úhlu impedance. Celkově vzato je tímto způsobem program funkční a spolupráce logického jádra s výpočetním funguje na výbornou.

4) Logické jádro pro automatické nastavení zón impedančních ochran

Logické jádro pro automatické nastavení zón již není pouze algoritmiizační úlohou o uspořádání výstupní tabulky, kde následně na základě jedné další proměnné z výpočtu rozhoduje o provedení druhého výpočtu s jinými parametry zdroje. Tentokrát slouží také jako hlavní článek pro návrh jednotlivých zón v impedančních ochranách. Díky této úloze se logické jádro pro tuto činnost velmi zesložituje, a to hlavně z pohledu automatizace všech kroků, které dělá technik na základě výpočtů z jednotlivého zkratu.

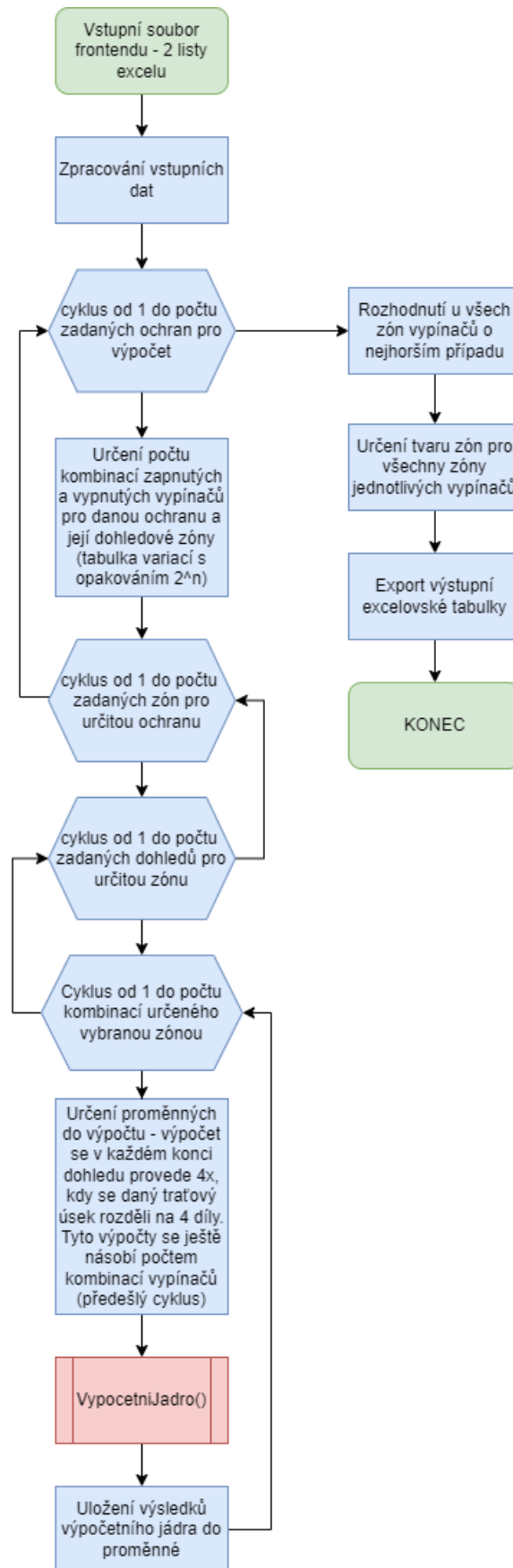
4.1) Požadavky a vstupní podmínky logického jádra

Hlavním požadavkem projektantů bylo, aby program ověřil správnost svých výpočtů s nastavením zón. Tento požadavek byl nakonec přenesen na uživatele, protože kdyby i tento krok mělo dělat logické jádro, výpočet by trval velmi dlouho. Domluvilo se nakonec to, že logické jádro udělá pro každý dohled, kam impedanční ochrana uvidí, čtyři výpočty, kdy poslední traťový úsek rozdělí na čtyři části a každou takovou část pošle se zkratem do výpočtu. Všechny výpočty, které byly za běhu logického jádra provedeny, se nakonec zobrazí ve frontendu, kde uživatel přímo uvidí, jaké hodnoty do výpočtu vstoupily. Uvidí také výsledné impedance zkratů vykreslených do příslušné zóny v grafu.

Aby výpočet netrval vždy tak dlouho, může uživatel pro některé ochrany tento výpočet zakázat. I tak je ale čistě na uživateli, kolik zón bude chtít navrhnout a zda si bude vědom toho, že výpočty s přibývajícím počtem zón, dohledů těchto zón a nakonec i ochran stoupají exponenciálně. Navíc můžeme předpokládat, že si uživatel vždy nakreslí celou trať, kde může být i přes 100 uzlů a například 6 napájecích stanic. S počtem uzlů, traťových úseků a zdrojů taktéž stoupá čas výpočtu, protože výpočetní jádro počítá metodou uzlových napětí.

4.2) Zjednodušený vývojový diagram pro logické jádro automatického nastavení zón impedančních ochran

V rámci konzultací s hlavními řešiteli jsem navrhnul přibližnou strukturu logického jádra, které by se na základě tohoto vývojového diagramu mohlo začít programovat. Ve vývojovém diagramu určitě chybí rozepsání nejdůležitějších kroků, a to například pro příkaz „Určení tvaru zón“. Tento algoritmus bude pro logické jádro nejsložitějším na naprogramování.



Obrázek 24 - Vývojový diagram pro logické jádro automatického nastavení zón impedančních ochran

Vstupní tabulka do logického jádra obsahuje 2 listy, přičemž jsou na prvním listu informace o trati, stejně jako u vstupní tabulky do logického jádra pro výpočet jednotlivého zkratu. Druhý list obsahuje seznam všech vypínačů. K těmto vypínačům jsou přidruženy hodnoty 0 či 1, které znamenají, zda se mají navrhnout impedanční zóny pro tento vypínač přidružený k ochraně.

Dále se již dostáváme do složitějšího cyklu, kde je potřeba mít dohromady čtyři instance. První cyklus určuje, pro jaký vypínač budou následující výpočty probíhat, druhý určuje, pro jakou zónu výpočet probíhá, třetí označuje, který dohled zóny bude vypočítán a čtvrtý určuje jednotlivé kombinace zapnutých a vypnutých vypínačů pro určitý dohled. Při každém průchodu se spustí výpočetní jádro tolikrát, kolik je na trati dohledů, přičemž tento počet spuštění se násobí čtyřmi kvůli požadavkům hlavního řešitele. Po každém výpočtu se výsledky uloží do tabulky, která bude popsána v následujících odstavcích této práce.

Poté, co proběhnou všechny výpočty, může dojít k určení nejhoršího stavu, tedy k posouzení pro každou zónu, který výsledek je impedančně nejvyšší, a to i s úhlem. Největší smysl nám zatím dává, že nejvyšší výsledek vyjde pro jednostranné napájení, kdy bude zkrat umístěn na konci daného traťového úseku, respektive na nejvzdálenějším dohledu zóny. Vyhodnocení jednotlivé zóny proběhne prostým porovnáním velikosti impedance, kdy se spolu budou srovnávat jednotlivé dohledy i s jednotlivými možnostmi sepnutí vypínačů. Na základě toho se do jiné proměnné napíše nejvyšší možná hodnota, na kterou bude navržena daná zóna. Tímto se určí i tvar zóny, přičemž od uživatele již budeme znát jednotlivé úhly kreslení zón, které přesahují jak do druhého, tak čtvrtého kvadrantu. Pokud bude zóna zpětná, bude zasahovat hlavně do třetího kvadrantu, přičemž tyto přesahy budou definovány do druhého a čtvrtého kvadrantu.

Po určení bodů v jednotlivých zónách bude vygenerována výstupní tabulka, kterou bude program končit. Na základě takto určených bodů budou vykresleny v grafickém rozhraní jednotlivé zóny do jednoho obrázku pro jednu ochranu (vypínač). Tyto výsledky by mělo být možné exportovat z grafického rozhraní a zapisovat je do protokolu společně se simulovanými hodnotami.

4.3) Návrh výstupní tabulky pro automatické nastavení zón

Hlavní myšlenkou výstupní tabulky je, že musí obsahovat téměř všechna data, se kterými logické jádro pracovalo, respektive počítalo. Hlavní úlohou této tabulky je přenést všechna vypočtená data směrem k frontendu. Už teď je jasné, že výstupní tabulky budou dvě, přičemž jedna bude použita na výsledky ze všech výpočtů a druhá na body pro vykreslení zón.

V zadávací tabulce od grafického jádra pro vypínače jsou uvedeny i názvy těchto vypínačů. Možná bude dodáno i jejich identifikační číslo, či obecně identifikační znaky. Aby byla výstupní tabulka výsledků přehlednější a srozumitelnější, bude třírozměrná. Názvy jednotlivých listů budou určovat, k jakému vypínači a jaké zóně dané výsledky přiléhají, např. N11_Z1 značí vypínač N11 a první zónu. V jednotlivých řádcích budou vypsány dohledy. Zadavatelem bylo určeno, že dohledů pro jednu zónu může být maximálně deset, proto jsou řádky označeny D1-D10. Ve sloupcích se opakují stále dokola ty samé proměnné, kterých je dohromady šest: velikost proudu, úhel proudu, napětí, velikost impedance a úhel impedance. Posledním určující parametr je, v kolikátém kilometru byl zkrat vytvořen.

Tabulka 6 - Návrh výstupní tabulky pro výsledky výpočtů jednotlivých dohledů

	I [A]	I [°]	U [kV]	Z [ohm]	Z [°]	[km]
D1						
D2						
D3						
D4						
D5						
D6						
D7						
D8						
D9						
D10						

Druhá tabulka bude obsahovat pouze body pro jednotlivé zóny, přičemž třetí rozměr bude taktéž list v excelu, který bude pojmenován názvem daného vypínače, případně jeho identifikačním číslem. V řádcích najdeme jednotlivé zóny a ve sloupcích vždy souřadnice bodu tak, že se sloupce budou po dvou opakovat. První sloupec bude vždy velikost na ose R a druhý sloupec bude velikost na ose X. Grafické rozhraní jen tyto body pospojuje přímkami a tím vznikne obrazec zóny. Maximální počet počítaných zón je určen na 5.

Tabulka 7 - Návrh výstupní tabulky pro vykreslení impedančních zón

	R [ohm]	X [ohm]
Z1		
Z2		
Z3		
Z4		
Z5		

Závěr

Byl zpracován přehled měničových technologií, které se buď používají v drážním provozu, nebo jsou pro toto napájení ve vývoji. Do budoucna můžeme předpokládat ještě větší rozvoj této technologie napájení, protože výhody použití převažují nad nevýhodami, a to hlavně pro provozovatele. Dále byl také zpracován přehled o technologiích, které s použitím SFC mohou souviset. Tento přehled obsahuje i umístění SFC do struktury obecné napájecí stanice, tedy tak, jak se ve skutečnosti aktuálně měniče v TNS používají. Posledním teoretickým podkladem v této práci jsou informace o impedančních ochranách, abychom pochopili záměr celého simulačního programu, který je taktéž v této práci popsán.

Tato diplomová práce není jediným výstupem, který během studia vznikl. Hlavním výstupem je program logického jádra pro zpracování a úpravu zadávací tabulky, který přenáší upravená data do výpočetního jádra programu. Následně kontroluje výsledky, kdy za určitých podmínek může vyslat pokyn k opětovnému přepočítání, tentokrát ale s jinými parametry. V diplomové práci byly popsány všechny použité algoritmy, které jsou nedílnou součástí logického jádra pro jednotlivý zkrat. Algoritmy jsou vždy popsány jak vývojovým diagramem, tak doplňujícím psaným výkladem, který upřesňuje běh programu.

Navrhnutý program byl i úspěšně otestován na spoustě příkladů, přičemž ty nejzajímavější z nich byly popsány v předložené práci. Popis testování nejdříve určuje, za jakým účelem celý program testujeme, a následně shrnuje výsledky. Výsledky jsou popsány v tabulkách, které obsahují reálně změřená data při zkratových zkouškách a také výsledky ze simulací. Tato data jsou mezi sebou porovnávána procentuální odchylkou, která určuje přesnost výpočtu oproti změřenému výsledku. Toto porovnání někdy není zcela prokazatelné, a proto se používá i metoda porovnání přibližně stejných zadání, kterou se rovněž prokázalo, že program počítá správně.

Se změnami, které se pravidelně do výpočtového jádra zanášejí, není ale program logického jádra zcela hotový. Do výstupní tabulky pro výpočetní jádro přibyl list, který určuje pro jednotlivé zdroje také jejich charakteristiky. Další změnou jsou i některé nepotřebné sloupce, které jsou nahrazeny právě druhým listem v zadávací tabulce. Z těchto důvodů by se měla přepsat funkce VytvorTabulku() na univerzálnější funkci, jež bude na tyto změny připravena.

V neposlední řadě některá stanoviska mění i hlavní řešitel, pro kterého je program tvořen. Takovým hlavním stanoviskem je možnost změny parametrů pro zkratový režim SFC, kdy si uživatel bude moci určit trvalý výstupní proud. Dále je to i napěťová podmínka zkratového režimu, kdy přechod u jiných druhů měničů nemusí začínat na 19,5 kV, ale na jiné hodnotě napětí.

Další změny spíše souvisejí s rozvojem algoritmů. Hlavními změnami by mohly projít některé algoritmy, které používají ke své funkci cykly for. Zjednodušení těchto algoritmů by spočívalo ve zmenšení počtu cyklů, kdy není potřeba používat dva cykly, ale lze všechny kroky těchto dvou cyklů sloučit pouze do jednoho cyklu. Tyto úpravy přispějí ke spolehlivosti a rychlosti přepsání tabulky logickým jádrem.

Posledním splněným úkolem diplomové práce bylo navržení zjednodušeného vývojového diagramu pro logické jádro automatického nastavení zón impedančních ochran. Tento vývojový diagram byl navrhnout na základě požadavků zadavatele úlohy. Byly navrhнутy i výstupní tabulky, které obsahují všechny informace pro vykreslení a zobrazení. Tyto výstupní tabulky byly taktéž navrhнутy na základě požadavků od zadavatele.

Zdroje

- [1] CERMAN, Ladislav. *Vyšetřování zkratových poměrů v trakčních obvodech 25 kV 50 Hz se spojitým napájením a s měničovými napájecími stanicemi*. Diplomová práce. Pardubice: Univerzita Pardubice, Dopravní fakulta Jana Pernera, Katedra elektrotechniky, elektroniky a zabezpečovací techniky, 2023.
- [2] HASAN, Nor Shahida, Norzanah ROSMIN, Dygku. Asmanissa Awg. OSMAN and Aede Hatib MUSTA'AMAL@JAMAL. Reviews on multilevel converter and modulation techniques. *Renewable and Sustainable Energy Reviews* [online]. 2017, 80, 163-174 [cit. 2025-04-17]. ISSN 13640321. DOI: 10.1016/j.rser.2017.05.163. Dostupné z: <http://dx.doi.org/10.1016/j.rser.2017.05.163>
- [3] KAŠPAR, Jindřich. Zkušenosti s realizací pilotního projektu statických měničů (Nedakonice – Říkovice). *Odborný seminář Nová Elektrická zařízení*. 2022, roč. 2021-2022, č. 1, s. 1-15.
- [4] *Rail Static Frequency Converter (SFC) solutions: / MORE POWER FOR RAILWAY LINES DYNAMIC SHUNT COMPENSATION AND STATIC FREQUENCY CONVERSION*. Online. In: HITACHI. Hitachi energy. 2021. Dostupné z: <https://publisher.hitachienergy.com/preview?DocumentID=8DBS000001&LanguageCode=en&DocumentPartId=&Action=launch>. [cit. 2025-04-17].
- [5] MAHESWARI, K.T.; BHARANIKUMAR, R. a MANIVANNAN, S. Comparative evaluation of Z source / quasi Z-source direct and indirect matrix converters for PMG based WECS. Online. *ScienceDirect*. 2024, roč. 2024, č. 22, s. 2. Dostupné z: <https://doi.org/10.1016/j.rineng.2024.102219>. [cit. 2025-04-17].
- [6] BUREŠ, Jan. *Modely aktivních trakčních napaječů železnic pro analýzu zpětných vlivů na distribuční síť*. Online, Diplomová práce, vedoucí Jan Klusáček. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2024. Dostupné z: <https://dspace.vut.cz/items/554d07b2-6afb-48d5-b402-4e79591999d9>. [cit. 2025-04-21].

- [7] *Nastavovanie, prevádzka a údržba reléových ochrán trakčného napájacieho obvodu: SR 34 (E).* .pdf. Nakladatelství dopravy a spojů Praha, 1970.
- [8] IFB INSTITUT FÜR BAHNTECHNIK GMBH. *OpenPowerNet*. Online. 2024. Dostupné z: <https://www.openpowernet.de/>. [cit. 2025-04-22].
- [9] *Sitras Sidytrac*. Online. SIEMENS MOBILITY. Sitras Sidytrac. 2025. Dostupné z: <https://www.mobility.siemens.com/global/en/portfolio/digital-solutions-software/infrastructure/rail-electrification/sidytrac.html>. [cit. 2025-04-22].
- [10] VOJKŮVKA, Aleš, Richard LUŽNÝ a NEŠKODNÝ. Zpráva ze zkratových zkoušek TNS Otrokovice - TNS Říkovice. Praha: ELEKTRIZACE ŽELEZNIC PRAHA, 2022.
- [11] VOJKŮVKA, Aleš, Richard LUŽNÝ a NEŠKODNÝ. Zpráva ze zkratových zkoušek TNS Otrokovice - TNS Nedakonice. Praha: ELEKTRIZACE ŽELEZNIC PRAHA, 2022.
- [12] *Interní materiály firmy Elektrizace železnic Praha a.s.*

Seznam zkratek

TNS	Trakční napájecí stanice
FKZ	Filtračně-kompenzační zařízení
SFC	Static frequency convertor
NS	Napájecí stanice
OZ	Opětovné zapnutí
Frontend	Grafické rozhraní pro uživatele
Backend	Výpočetní jádro
TU	Trat'ový úsek

Seznam obrázků

Obrázek 1 - Bloková struktura měničové napájecí stanice 25 kV 50 Hz [1].....	11
Obrázek 2 - Jedna z možných topologií maticového měniče [3].....	12
Obrázek 3 - Přehledové schéma TNS	13
Obrázek 4 - Možnost zapojení recloserů k neutrálnímu úseku a TNS	14
Obrázek 5 - Příklad nastavení impedanční ochrany napaječe	15
Obrázek 6 - Grafické rozhraní programu pro výpočty v trakčních sítích 25 kV 50 Hz [12]....	19
Obrázek 7 - Zjednodušený vývojový diagram pro upřesnění funkcí jednotlivých částí programu	21
Obrázek 8 - Zjednodušený vývojový diagram popisující hlavní funkce	25
Obrázek 9 - Příklad úpravy zadávací tabulky na jednokolejné trati	29
Obrázek 10 - Větvení běhu programu.....	31
Obrázek 11 - Příklad úpravy dvoukolejné trati.....	32
Obrázek 12 - Logika počtu traťových úseků i s vazbami	34
Obrázek 13 - Algoritmus pro nové traťové úseky	35
Obrázek 14 - Zjištění počtu uzlů ze zadání a následné přiřazení parametrů traťovému úseku	36
Obrázek 15 - Přiřazení uzlů a parametrů vázaným traťovým úsekům	37
Obrázek 16 - Vývojový diagram funkce VytvorTabulku.....	39
Obrázek 17 - Vývojový diagram rozhodování o podpětí NS	42
Obrázek 18 - Algoritmus pro zpřesnění výsledků na základě vyhodnocení podpětí z přechodného algoritmu.....	44
Obrázek 19 - Zkratový scénář č. 9 na trati Otrokovice – Říkovice [10].....	47
Obrázek 20 - Zkratový scénář č. 8 na trati Otrokovice – Říkovice [10].....	48
Obrázek 21 - Zkratový scénář č. 3 na trati Nedakonice – Otrokovice [11].....	50
Obrázek 22 - Zkratový scénář č. 18 na trati Nedakonice – Otrokovice [11].....	51
Obrázek 23 - Zkratový scénář č. 20 na trati Nedakonice – Otrokovice [11].....	53
Obrázek 24 - Vývojový diagram pro logické jádro automatického nastavení zón impedančních ochran.....	56

Seznam tabulek

Tabulka 1 - Výsledky impedance Otrokovice-Říkovice, zkratový scénář č. 9.....	47
Tabulka 2 - Výsledky impedance Otrokovice – Říkovice, zkratový scénář č. 8.....	49
Tabulka 3 - Výsledky impedance Nedakonice – Otrokovice, zkratový scénář č. 3	51
Tabulka 4 - Výsledky impedance Nedakonice – Otrokovice, zkratový scénář č. 18	52
Tabulka 5 - Výsledky impedance Nedakonice – Otrokovice, zkratový scénář č. 20	53
Tabulka 6 - Návrh výstupní tabulky pro výsledky výpočtů jednotlivých dohledů.....	58
Tabulka 7 - Návrh výstupní tabulky pro vykreslení impedančních zón	59

Seznam příloh

Příloha A - Fotografie zkratovacího zařízení při zkratové zkoušce.....	68
Příloha B - Fotografie přívodového transformátoru k SFC	69
Příloha C - Fotografie SFC od firmy Hitachi	69
Příloha D - Fotografie vývodového transformátoru SFC	70
Příloha E - Fotografie rozvaděče 27 kV	70
Příloha F - Vývojový diagram logického jádra pro výpočet jednotlivého zkratu	71



Priloha B - Fotografie privodoveho transformatoru k SFC



Priloha C - Fotografie SFC od firmy Hitachi



Příloha D - Fotografie vývodového transformátoru SFC



Příloha E - Fotografie rozvaděče 27 kV



Příloha F - Vývojový diagram logického jádra pro výpočet jednotlivého zkratu

