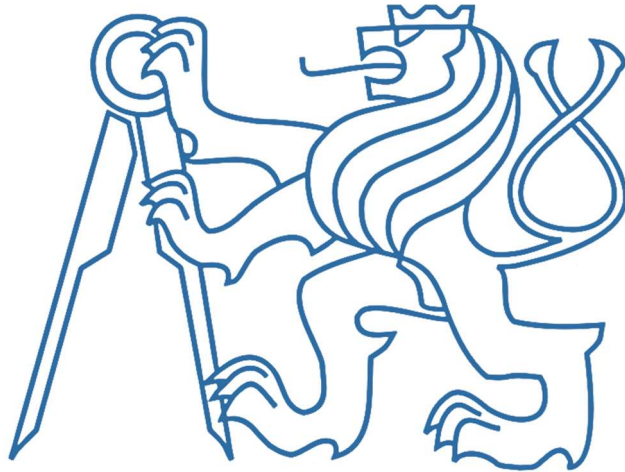


České vysoké učení technické v Praze
Fakulta dopravní
Ústav dopravní telematiky



Návrh algoritmu pro virtuální liniové řízení dopravy

DIPLOMOVÁ PRÁCE

Vypracoval: Bc. **Martin Vacenovský**
Vedoucí práce: Doc. Ing. **Zdeněk Lokaj, Ph.D., LL.M.**
Ing. **Miroslav Vaniš, Ph.D.**
Rok: **2022**

Čestné prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a použil jsem k tomu pouze zdroje uvedené na konci práce, a to v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Nemám závažný důvod proti užívání tohoto školního díla ve smyslu § 60 Zákona č.121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským.

V Praze dne:

.....

podpis

Poděkování

Děkuji Ing. Miroslavu Vanišovi, Ph.D. a doc. Ing. Zdeňku Lokajovi, Ph.D., LL.M. za vedení mé diplomové práce a za podnětné návrhy, které ji obohatily. Dále bych chtěl v tomto směru poděkovat Ing. Jiřímu Benešovi za věcné připomínky a konzultace k vypracování této práce. V poslední řadě bych chtěl poděkovat svým rodičům a blízkým za morální podporu po dobu studia.

.....

podpis

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta dopravní

děkan

Konviktská 20, 110 00 Praha 1



K614..... **Ústav dopravní telematiky**

ZADÁNÍ DIPLOMOVÉ PRÁCE (PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení studenta (včetně titulů):

Bc. Martin Vacenovský

Studijní program (obor/specializace) studenta:

navazující magisterský – ITS – Inteligentní dopravní systémy

Název tématu (česky): **Návrh adaptivního algoritmu pro virtuální liniové řízení dopravy**

Název tématu (anglicky): Design of Adaptive Algorithm for Virtual Line Traffic Control

Zásady pro vypracování

Při zpracování diplomové práce se řiďte následujícími pokyny:

- Popište systémy liniového řízení dopravy a aktuálně využívané algoritmy pro virtuální liniové řízení dopravy.
- Analyzujte možné využitelné datové zdroje pro virtuální liniové řízení dopravy.
- Analyzujte aktuální stav a možnosti využití modelů pro predikci dopravní situace.
- Na základě provedené analýzy navrhnete vhodný adaptivní algoritmus pro virtuální liniové řízení dopravy s možností predikce.
- Navrhnutý algoritmus ověřte na dopravním modelu ve vhodném simulačním SW.
- Zhodnoťte závěry práce a navrhnete další směry vývoje v dané oblasti.

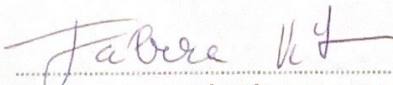


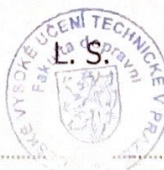
- Rozsah grafických prací: 10 - 20 obrázků
- Rozsah průvodní zprávy: minimálně 55 stran textu (včetně obrázků, grafů a tabulek, které jsou součástí průvodní zprávy)
- Seznam odborné literatury: Zelinka, T., Svítek, M.: Telekomunikační řešení pro informační systémy síťových odvětví, Grada 2009
Odborné články IEEE
Normy, standardy a odborné časopisy

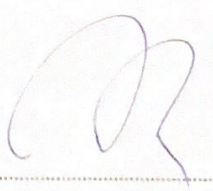
Vedoucí diplomové práce: **doc. Ing. Zdeněk Lokaj, Ph.D.**
Ing. Miroslav Vaniš, Ph.D.

Datum zadání diplomové práce: **16. července 2021**
(datum prvního zadání této práce, které musí být nejpozději 10 měsíců před datem prvního předpokládaného odevzdání této práce vyplývajícího ze standardní doby studia)

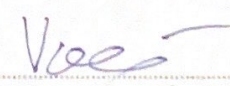
Datum odevzdání diplomové práce: **16. května 2022**
a) datum prvního předpokládaného odevzdání práce vyplývající ze standardní doby studia a z doporučeného časového plánu studia
b) v případě odkladu odevzdání práce následující datum odevzdání práce vyplývající z doporučeného časového plánu studia


doc. Ing. Vít Fábera, Ph.D.
vedoucí
Ústavu dopravní telematiky


doc. Ing. Pavel Hrubeš, Ph.D.
děkan fakulty



Potvrzuji převzetí zadání diplomové práce.


Martin Vacenovský
jméno a podpis studenta

V Praze dne.....16. července 2021

Název práce: Návrh algoritmu pro virtuální liniové řízení dopravy
Autor: Bc. Martin Vacenovský
Obor: Inteligentní dopravní systémy
Druh práce: Diplomová práce
Vedoucí práce: Doc. Ing. Zdeněk Lokaj, Ph.D., LL.M.
Ing. Miroslav Vaniš, Ph.D.
Ústav dopravní telematiky, Fakulta dopravní, České vysoké
učení technické v Praze

Abstrakt: Předmětem této diplomové práce je návrh adaptivního algoritmu pro virtuální liniové řízení dopravy na dálnicích. První kapitola obsahuje rešerši systémů liniového řízení dopravy, jež jsou aktuálně v provozu. Dále jsou v první kapitole shrnuty přístupy návrhu algoritmů systémů liniového řízení dopravy. Druhá kapitola obsahuje rešerši detekčních prvků, které mohou být využity v rámci návrhu algoritmu VLŘD. Poté následuje popis využitého simulačního SW v rámci této práce. Stěžejním zaměřením práce je vlastní návrh řídicího algoritmu a jeho následné ověření v mikroskopické simulaci dopravy.

Klíčová slova: liniové řízení dopravy, virtuální liniové řízení dopravy, řídicí algoritmus, dopravní detektory, dopravní simulace, SUMO, proměnné rychlostní limity

Title: Design of Adaptive Algorithm for Virtual Line Traffic Control

Author: Bc. Martin Vacenovský

Abstract: The subject of this diploma thesis is the design of an adaptive algorithm for virtual line traffic control on highways. The first chapter contains a search of line traffic management systems that currently are in operation. Furthermore, the first chapter summarizes the approaches to the design of algorithms for linear traffic control systems. The second chapter contains a search of detection elements that can be used in the design of the VLTC algorithm. This is followed by a description of the simulation software used in this work. The main focus of the work is the design of the control algorithm and its subsequent verification in the microscopic traffic simulation.

Key words: line traffic control, virtual line traffic control, control algorithm, traffic detectors, traffic simulation, SUMO, variable speed signs

Obsah

| | |
|--|----|
| Úvod..... | 3 |
| Kapitola 1 Systémy liniového řízení dopravy..... | 5 |
| 1.1 Liniové řízení dopravy v ČR..... | 5 |
| 1.1.1 Řídící algoritmus | 6 |
| 1.2 Liniové řízení dopravy ve světě | 10 |
| 1.2.1 Algoritmy založené na pravidlech (rule-based) | 11 |
| 1.2.2 Algoritmy založené na fuzzy logice (fuzzy-logic based)..... | 15 |
| 1.2.3 Algoritmy založené na teorii řízení (control-theory based) | 18 |
| 1.3 Shrnutí | 20 |
| Kapitola 2 Datové zdroje pro VLŘD | 21 |
| 2.1 Metoda detekce indukční smyčkou | 21 |
| 2.2 Metoda detekce FCD (Floating Car Data)..... | 23 |
| 2.3 Metoda detekce pomocí technologie Bluetooth / Wifi..... | 26 |
| 2.3.1 Analýza datových výstupů Bluetooth/Wifi detekce | 27 |
| 2.3.2 Závěr analýzy detekce pomocí technologie Bluetooth/Wifi ... | 31 |
| 2.4 Metoda detekce pomocí technologie V2X..... | 32 |
| 2.4.1 Stručný úvod do C-ITS systémů | 32 |
| 2.4.2 V2X technologie..... | 33 |
| 2.4.3 Detekce pomocí CAM zpráv | 34 |
| 2.4.4 Probe Vehicle Data..... | 35 |
| 2.4.5 Závěr analýzy detekce pomocí V2X technologie..... | 36 |
| 2.5 Závěr analýzy datových zdrojů pro algoritmus VLŘD | 37 |
| Kapitola 3 Možnosti predikce dopravní situace..... | 38 |
| 3.1 SUMO | 39 |
| 3.1.1 Funkční koncept | 39 |
| 3.1.2 Data o dopravní síti | 40 |
| 3.1.3 Dodatečné informace o dopravní infrastruktuře..... | 44 |
| 3.1.4 Dopravní požadavky..... | 44 |

| | | |
|--------------------------|--|----|
| 3.1.5 | Spuštění simulace | 47 |
| 3.2 | Možnosti predikce dopravní situace pomocí nástroje SUMO..... | 51 |
| Kapitola 4 | Návrh adaptivního algoritmu pro systém VLŘD..... | 54 |
| 4.1 | Základní schéma navrženého algoritmu..... | 54 |
| 4.2 | Popis jednotlivých částí algoritmu | 55 |
| 4.2.1 | Vstupní prvky | 55 |
| 4.2.2 | MPC (Model Predictive Control)..... | 58 |
| 4.2.3 | Výstupní prvky | 63 |
| 4.3 | Popis funkce algoritmu..... | 64 |
| 4.3.1 | Možné poruchové stavy..... | 66 |
| 4.3.2 | Provozní režimy algoritmu | 68 |
| 4.3.3 | Harmonizace rychlostních omezení | 68 |
| 4.3.4 | Prognóza trendu..... | 69 |
| Kapitola 5 | Ověření návrhu algoritmu ve zvoleném simulačním SW | 71 |
| 5.1 | Ověřovací proces | 71 |
| 5.2 | Implementace ověřovacího procesu | 72 |
| 5.3 | Ověření navrženého algoritmu | 76 |
| 5.3.1 | Situace ve volném dopravním proudu..... | 76 |
| 5.3.2 | Situace při vyšší dopravní zátěži | 78 |
| 5.4 | Zhodnocení výsledků ověřovacího procesu | 79 |
| Kapitola 6 | Další směry vývoje v dané oblasti | 80 |
| Závěr | | 81 |
| Citovaná literatura..... | | 83 |
| Seznam zkratek | | 86 |

Úvod

Dopravně telematické systémy jsou integrací informačních a telekomunikačních technologií s dopravně-inženýrskými zkušenostmi, které jsou doplněny o další poznatky ze souvisejících oborů, jakými mohou být například teorie dopravního proudu, systémová analýza a další. Tyto systémy jsou v dnešní době stále častěji využívány jak v silniční infrastruktuře, tak v jednotlivých dopravních prostředcích, kde slouží k lepší informovanosti účastníků silničního provozu, zvýšení bezpečnosti, optimalizaci nákladů spojených s dopravou, k zajištění maximální využitelnosti stávajících i nově budovaných komunikací, atd. V rámci těchto technologií jsou využívány různé metody detekce veličin, jež jsou klíčové pro řízení dopravy jakožto celku. Mezi tyto veličiny mohou patřit jak veličiny dopravního proudu, tak například meteorologické veličiny, dle kterých lze odvodit stav povrchu komunikace. Jako příklad telematické aplikace lze v současné době zařadit systém eCall, který umožňuje rychlou automatickou pomoc účastníkům silničního provozu, kteří se stali účastníkem silniční nehody, nebo liniové řízení systémy dopravy, které pomocí proměnlivých rychlostních omezení se snaží o harmonizaci dopravního toku.

Právě systému liniového řízení dopravy (LŘD) se bude tato práce dále věnovat. LŘD je jednou ze strategií managementu dálnic, která si klade za cíl předcházet pravidelně se tvořícím dopravním kongescím, v jisté míře řešit mimořádné dopravní excesy, a zvýšit bezpečnost dopravy na pozemních komunikacích. Je dokázáno, že maximální propustnosti komunikace není dosahováno při maximální rychlosti, ale při rychlostech nižších, zhruba mezi 80 – 100 km/h. Není ale smysluplné omezovat maximální povolenou rychlost ve chvílích, kdy se na komunikaci nachází volný dopravní proud. Díky proměnlivým dopravním značením je možné toto omezení dát v platnost pouze tehdy, je-li to potřeba. Spínání těchto rychlostních omezení je závislé na naměřených veličinách dopravního proudu. Logika vyhodnocování naměřených veličin a vytváření rychlostních omezení utváří v rámci systému LŘD řídicí algoritmus.

Postupem doby již je možné zavést termín virtuální liniové řízení dopravy (VLŘD), který značí vytváření rychlostních omezení ve virtuálním světě. Stejně jako LŘD se VLŘD snaží o harmonizaci dopravního proudu pomocí rychlostních omezení, které ale nejsou jako v případě LŘD zobrazovány na jednotlivých portálech, ale jsou digitální formou distribuovány účastníkům silničního provozu, případně samotným vozidlům. Vysoký potenciál tohoto systému nelze hledat v době současné, ale v době budoucí, kdy se za nástupu autonomních vozidel může systém VLŘD stát nenahraditelným nástrojem pro harmonizaci dopravního proudu.

Hlavním cílem této práce je návrh adaptivního algoritmu pro VLŘD. Návrh bude vytvořen na základě dostupných relevantních informací o systémech LŘD a jejich řídicích algoritmech. Nicméně systém VLŘD, stejně jako systém LŘD jsou systémy

komplexní, kdy je v rámci návrhu jeho řídicího algoritmu nutné brát v úvahu mnoho vstupních (ale i výstupních) faktorů, které systém ovlivňují. Tato práce si klade za cíl, aby ve výsledném návrhu bylo co nejvíce takových faktorů vzato v potaz a díky tomu by výsledný návrh mohl být použit při možné budoucí implementaci. Faktory, které se v rámci této práce nepodařilo do návrhu zařadit budou posléze zmíněny v kapitole věnující se budoucímu rozvoji, aby případné budoucí návrhy tyto faktory do své implementace zařadily.

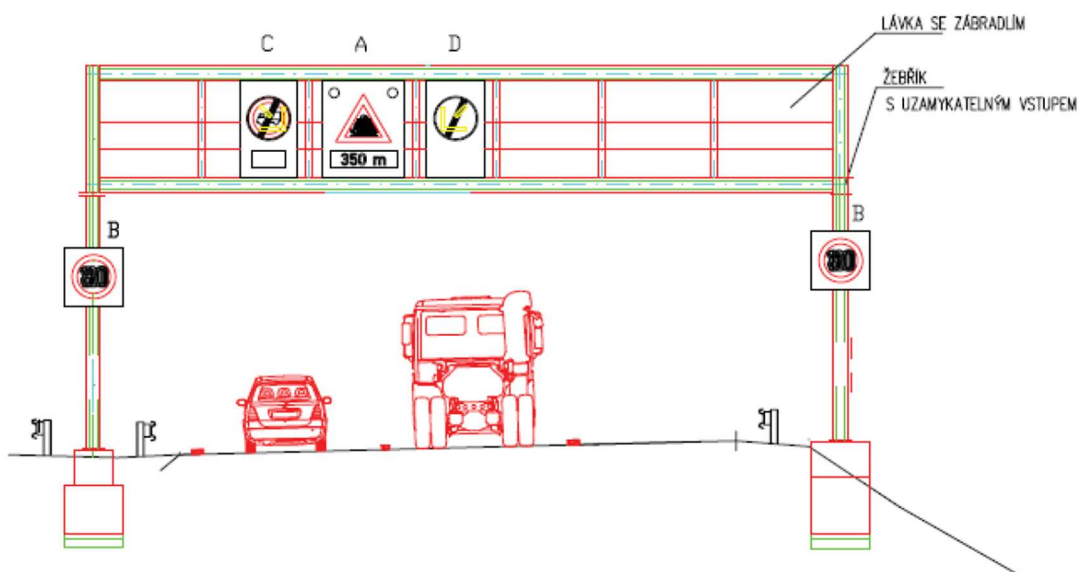
Kapitola 1

Systemy liniového řízení dopravy

Předmětem kapitoly je rešerše užívaných systémů liniového řízení dopravy ve světě či tuzemsku a jejich řídicích algoritmů. Cílem kapitoly je stručné shrnutí aktuálních trendů a výtah relevantních informací, jež by mohly přispět k cílům této práce.

1.1 Liniové řízení dopravy v ČR

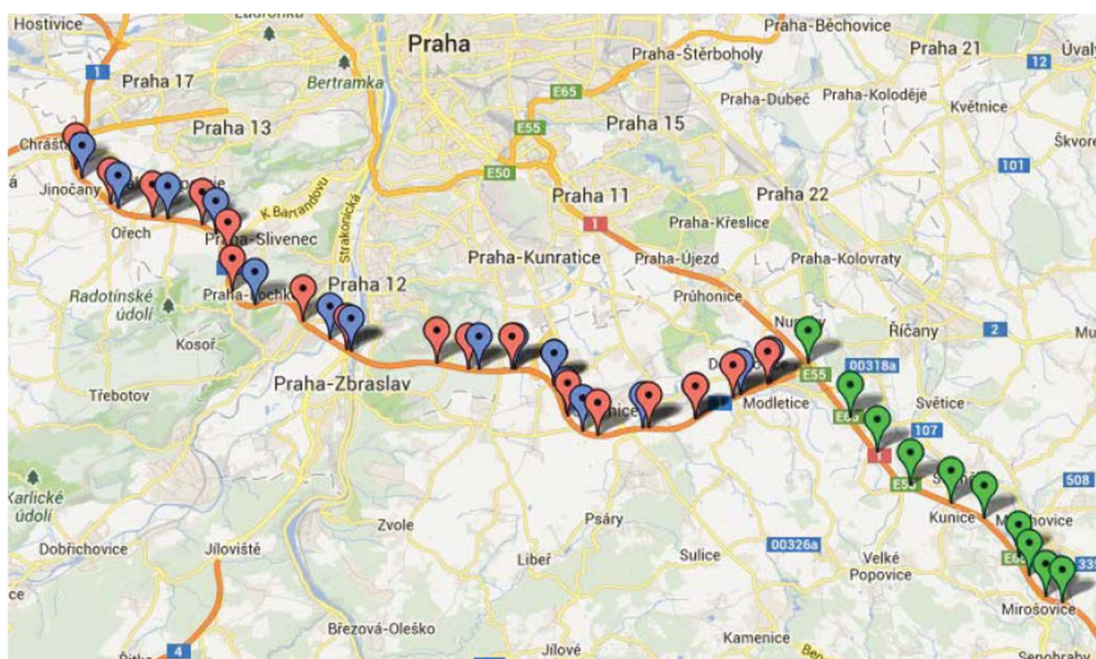
Linové řízení dopravy (dále jen LŘD) je jedna z telematických aplikací managementu dálnic, jež slouží k řízení dopravního proudu na dálnici pomocí proměnného dopravního značení (PDZ). Základním kamenem těchto systémů jsou portály rozmístěné v jednotlivých řezech komunikace. Vzdálenost mezi jednotlivými portály se pohybuje v rozmezí 1000 až 1500 metrů. Každý portál je osazen PDZ umístěným na čele portálu a po jeho bocích (viz Obrázek 1), jež předávají výstup systému řidičům. Dále jsou součástí všech portálů indukční smyčkové detektory, které jsou umístěny ve vozovce v každém jízdním pruhu. Jejich podstatou je detekce průjezdu jednotlivých vozidel. Poslední nezbytnou částí portálu LŘD je řídicí jednotka portálu (tzv. gantry server), která agreguje data naměřená smyčkami, odesílá je na centrální server a poveluje PDZ. Příklad portálu nad dvoupruhovým uspořádáním jízdního pásu je vyobrazen na Obrázku 1. [1]



Obrázek 1: Portál LŘD, dvoupruhová komunikace [1]

Dle Obrázku 1 je patrné, že na každém portálu LŘD jsou umístěny dvojice PDZ B20a (B), na kterých je zobrazována nejvyšší povolená rychlost. Hodnoty rychlostí, jež je možné zobrazit jsou 60, 80, 100 a 120 km/h. Dále je na portálu LŘD umístěna výstražná dopravní značka (A) s piktogramy Nebezpečí smyku (A8), Práce na silnici (A15), Jiné nebezpečí (A22), Kolona (A23), Náledí (A24), Mlha (A26) a nehoda (A27). Nad jízdními pruhy jsou potom umístěny PDZ, umožňující zobrazit symbol Světelná šipka (S8c a S8d) nad levým pruhem (a středním pruhem u třípruhového uspořádání). Dále umožňují zobrazit Zákaz jízdy nákladních vozidel s dodatkovou tabulkou E3 6t a značku B26 Konec všech zákazů. [1]

Typickým příkladem LŘD v České republice je úsek D1 od Mirošovic k Pražskému okruhu a dále směrem na Zličín. Portálů LŘD je na Pražském okruhu a dálnici D1 umístěno celkem 45. Mapa zobrazující jejich pozice je vyobrazena na obrázku 2.



Obrázek 2: Lokality portálů LŘD na Pražském okruhu a D1 [1]

Proměnné dopravní značky omezující rychlost jsou na portálu automaticky povolovány řídicím algoritmem modulu harmonizace dopravy, který byl vyvinut ve spolupráci Fakulty dopravní ČVUT a skupiny ELTODO v rámci projektu vědy a výzkumu s označením CG944-033-120 INEP – Systém identifikace nehod a zvýšení propustnosti liniových komunikací. [1]

1.1.1 Řídicí algoritmus

Řídicí algoritmus pro systém LŘD v ČR, jež poveluje jednotlivé portály a rozhoduje o maximální povolené rychlosti je dle Jiřího Beneše v jeho diplomové práci popsán následovně [1]:

„Algoritmus funguje na principu klasického rozhodovacího stromu. Jsou definovány podmínky pro zapínání jednotlivých omezení rychlosti – pakliže je podmínka splněna, je

aktivováno příslušné PDZ. Algoritmus obsahuje také podmínky pro vypínání, které mají posunutě hraniční hodnoty tak, aby k vypínání docházelo s mírným zpožděním (např. podmínka pro zapnutí PDZ B20a 120 km/h je intenzita vozidel > 3200 jvoz/h, podmínka pro její vypnutí intenzita vozidel < 3000 jvoz/h.

Přestože algoritmus obsahuje pouze 10 podmínek, je samotný rozhodovací strom poměrně složitý. V každém kroku je nutné zkontrolovat, které omezení má být aktivováno a zároveň, které a jestli vůbec může být deaktivováno. Výsledná struktura tak obsahuje velký počet rozhodnutí. S přidáním dalších podmínek by pak tento počet rychle narůstal (zvláště pokud by šlo o kombinace některých parametrů).“ [1]

1.1.1.1 Vstupní algoritmu a rozhodovací podmínky

Z popisu je tedy zřejmé, že hlavním rozhodovacím kritériem pro úpravu rychlostního omezení jsou naměřené hodnoty intenzit v jednotlivých řezech, které jsou posléze pro zjednodušení převedeny na hodnoty jednotkových vozidel. Detektory poskytují počty vozidel jednotlivých kategorií, a převod na jednotková vozidla je realizován formou vynásobení jednotlivých hodnot příslušnými koeficienty (např. detekce jedné hodnoty nákladního vozidla je vynásobena koeficientem s hodnotou 2) a poté jsou všechny vynásobené hodnoty sečteny. Převod je přizpůsoben tomu, aby výsledek reprezentoval svými vlastnostmi nejčastější vozidla v dopravním proudu, tedy osobní automobily [2].

Vypočtené jednotkové vstupní hodnoty intenzit jsou následně porovnány s hodnotami intenzit přednastavenými v rámci návrhu algoritmu. Tyto hodnoty jsou spočteny z charakteristik komunikace (např. hustota). Měly by představovat maximální počet vozidel, který by se měl v tom nejvíce vytíženém okamžiku na řízenou komunikaci vejít [2].

Pokud se vstupní hodnoty intenzit přiblíží hraničním hodnotám, začne algoritmus dle dalších pravidel spínat jednotlivá rychlostní omezení. V rámci algoritmu je implementována i prognóza trendu, která odhaduje koeficient vzrůstu či poklesu jednotlivých naměřených veličin. Tyto koeficienty jsou posléze využity v dalších podmínkách ústících do zvolení optimální hodnoty rychlosti [2].

Nicméně konstantně nastavené hodnoty, při kterých se má algoritmus rozhodovat, jaké rychlostní omezení zvolí, mohou začít být z hlediska dlouhodobé prognózy problematické.

Nastíněná problematičnost může být způsobena navyšováním počtu vozidel, která vstupují do dopravního proudu. Tvrzení je podloženo analýzou celostátního sčítání dopravy provedené subjektem ŘSD z let 2010 a 2016 z úseku na komunikaci D0 hranice okr. Praha-východ a Praha-západ – Jesenice, kde v roce 2010 byl součet všech vozidel RDPI ze všech dní roven hodnotě 31 069 a v roce 2016 již tento počet byl roven hodnotě 38 750. Rozdíl těchto dvou hodnot se rovná procentuálnímu nárůstu meziročně o 4 % [3] [4].

Ve sčítání dopravy z roku 2020 je hodnota RDPI všech vozidel ze všech dní na analyzovaném úseku rovna hodnotě 39 568, nicméně v době tohoto sčítání byla dopravní

situace výrazně ovlivněna pandemií COVID-19, kdy například nákladní doprava zaznamenala pokles oproti roku 2019 až o 8 %, u veřejné hromadné dopravy se uvádí pokles až 30-40 %. Je tedy pravděpodobné, že skutečný nárůst po navrácení dopravní situace do standartního stavu je mnohem vyšší, než kolik naznačují počty naměřené v tomto celostátním sčítání [5].

Z výše uvedeného vyplývá skutečnost, že kvantita vozidel v dopravním proudu v čase je dynamická, nikoliv statická. Proto návrh statických hodnot intenzit, na základě kterých se systém LŘD pokouší o homogenizaci dopravy nemusí být z hlediska dlouhodobé prognózy optimální. Po delším časovém období již algoritmus nemusí reagovat na dopravní situaci optimálně a je nutná jeho recalibrace. Je to z důvodu, že se charakteristika dopravního proudu v delším časovém horizontu významně mění, ať už do počtu vozidel nebo do jejich typů. Z provedené analýzy je navíc zřejmé, že kvantita vozidel v dopravním proudu má spíše vzestupný trend. Pokud by tomu bylo naopak, tedy že trend kvantity vozidel v dopravním proudu by byl klesající, tak by problém statických hraničních hodnot v návrhu algoritmu nebyl tolik markantní.

1.1.1.2 Funkční režimy algoritmu LŘD

Aktuální algoritmus LŘD umožňuje 4 režimy řízení. Tyto režimy jsou [2]:

- Automatický režim
- Automatický režim s potvrzením
- Poloautomatický režim
- Manuální režim

Každý režim má specifický vzorec chování. Automatický režim na základě dat z dopravních detektorů automaticky poveluje jednotlivé řezy a snaží se harmonizovat dopravní proud pomocí omezení maximální povolené rychlosti, případně zákazu vjezdu nákladních vozidel v levém jízdním pruhu. Celý tento proces probíhá automaticky bez zásahu dispečera [2].

Automatický režim s potvrzením funguje podobně jako režim automatický s tím rozdílem, že všechna opatření navrhnutá algoritmem je nutné potvrdit dispečerem, aby se opatření v řezech zobrazila. Tento režim navíc, krom dopravních detektorů vyhodnocuje i data z meteohlásek a na základě těchto dat navrhuje dispečerovi symboly o meteorologických podmínkách k zobrazení v řezech. V případě nepříznivých povětrnostních podmínek navrhuje algoritmus i úpravu rychlosti [2].

V poloautomatickém režimu nejsou zpracovávána žádná vstupní dopravní data, na základě kterých by posléze algoritmus vydával opatření. Vstupní data jsou v tomto režimu zadána dispečerem. Algoritmus na základě těchto dat vygeneruje návazná opatření a tato opatření opět předá na potvrzení dispečerovi. Po potvrzení jsou opatření zobrazena v příslušných řezech [2].

Manuální režim má ze všech výše uvedených režimů nejvyšší prioritu. V tomto režimu jsou veškeré PDZ na všech řezech plně pod kontrolou dispečera. Je tedy schopen ovládat všechny řezy a zobrazovat na nich jakékoliv dopravní značení, jaké uzná za vhodné.

Zobrazené symboly jsou navíc uzamčeny pro úpravy ze strany algoritmu a zůstanou tam do té doby, dokud je dispečer nevypne. Původní návrh počítal s tím, že tento režim bude využíván pouze ve výjimečných případech [2].

Každý z režimů má omezené množství dopravních značení, které je možné zobrazit, kromě manuálního, který má možnost zobrazit všechny možné dopravní značení. Nejmenší množství dostupných značení k zobrazení má automatický režim, poté se oprávnění k zobrazení značení každým dalším režimem postupně rozšiřují až k manuálnímu režimu, jež má oprávnění na všechna. Symboly vytvořené pomocí manuálního režimu mají také nejvyšší prioritu [2].

1.1.1.3 Výstupy algoritmu

Výstupem algoritmu je povel pro portál systému LŘD. Povelem je myšlen příkaz, které symboly zobrazit na jednotlivých PDZ umístěných na portálu. Povel může, a měl by být, pro každý portál jiný a měl by být závislý na jeho pozici v kontextu ostatních portálů. Povel obsahuje informace jaké dopravní značení a na jaké části portálu ho zobrazit. Povelovací logika funguje na principu master, slave, kdy portál je v kontextu této logiky slave a povelovací algoritmus master. Na portálu by neměla být implementována vyšší logika řízení portálu z důvodu omezení chybovosti celého systému.

Hlavním výstupem jsou v tomto systému hodnoty omezení rychlosti, které jsou zobrazovány na jednotlivých portálech systému LŘD, která mají primárně harmonizovat dopravní proud při vyšších intenzitách. Vedlejším výstupem mohou být povely obsahující dopravní značení poskytující dodatečné informace účastníkům silničního provozu. Příkladem takového povelu může být povel k zobrazení výstražné dopravní značky A27 “Nehoda“ s dodatečnou informací o lokaci, která informuje účastníky silničního provozu o dopravní nehodě před nimi. Dalším dodatečným výstupem algoritmu LŘD může být informace o meteorologických podmínkách na komunikaci předáním povelu portálu k zobrazení výstražné značky A26 “Mlha“, která informuje silniční účastníky o meteorologické situaci před nimi.

Výstupy algoritmu jsou tedy pouze povely pro jednotlivé portály. Existuje také možnost tyto povely převést do formy mezinárodního protokolu DATEX II, jenž je využíván systémem NDIC pro všeobecné dopravní informace na území ČR.

Pro ilustraci je na obrázku 3 vyobrazen příklad zobrazení povelu algoritmu LŘD na dopravním portálu. V příkladu je vyobrazeno omezení rychlosti pomocí dopravního značení B20a “Nejvyšší povolená rychlost“ na hodnotu 100 km/h a informování řidičů značením A22 “Jiné nebezpečí“ spolu se světelnou šipkou S 8c upřesňující lokaci informace [2].



Obrázek 3: Portál LŘD [6]

1.2 Liniové řízení dopravy ve světě

Tato část práce má za úkol analyzovat algoritmy LŘD užívané ve světě. Systémy LŘD jsou implementovány v řadě světových zemí, nicméně popisy jejich algoritmů a zhodnocení jejich výsledné efektivity nejsou, stejně jako je tomu u České varianty, veřejně dostupné. Mezi nejdůležitější nálezy lze zařadit rozdělení aktuálně navržených algoritmů řízení, ať už implementovaných v reálném provozu nebo zatím pouze teoretických, do 3 kategorií odvíjejících se od implementované logiky řízení. Tyto kategorie jsou [7]:

- Algoritmy založené na pravidlech (Rule based)
- Algoritmy založené na fuzzy logice (fuzzy-logic based)
- Algoritmy založené na řídicí teorii (control-theory based)

Dle dostupných informací, ve většině zemí, kde jsou systémy LŘD implementovány, jsou použity algoritmy založené na pravidlech. Hlavním rozdílem napříč světově používanými algoritmy jsou okolní podmínky a přednastavené hodnoty [7] [8].

Pro lepší pochopení problematiky a s přihlédnutím na cíl práce byla tato kapitola rozdělena na části dle kategorického rozdělení řídicích algoritmů LŘD uvedeného výše. U tzv. rule-based algoritmů budou uvedeny příklady z reálného světa a možnosti jejich výstupů. Ty se totiž na rozdíl od vstupů mohou měnit. Ostatní kategorie jsou popsány pouze v teoretické rovině a není u nich známa reálná implementace. Vstupy řídicích algoritmů jsou ve většině případů stejné [7].

1.2.1 Algoritmy založené na pravidlech (rule-based)

Tyto algoritmy se řadí mezi světově nejpoužívanější. Pro zobrazování rychlostních symbolů jsou v těchto algoritmech použity tzv. tresholdy, znamenající například přednastavené hodnoty intenzit, obsazenosti a průměrné rychlosti. Samotný řídicí algoritmus obsahuje soubor podmínek, které porovnávají naměřené hodnoty na detektorech vůči zmíněným tresholdům. Pokud naměřená hodnota přesáhne hodnotu tresholdu, pak algoritmus spustí předdefinovaný proces zobrazení rychlostního omezení pro tento treshold.

Od počtu hodnot omezení rychlosti, jež je systém LŘD schopen zobrazit se odvíjí počet tresholdů. Nejmenší hodnota omezení rychlosti by měla odpovídat nejvyšší hodnotě tresholdu. Naopak nejvyšší hodnota omezení rychlosti by měla odpovídat nejnižší hodnotě tresholdu. Tímto způsobem je zajištěno, že při nejvyšší intenzitě dopravy se budou zobrazovat nejmenší hodnoty omezení rychlosti.

Hodnoty řídicích tresholdů by měly být vypočteny z charakteristických hodnot jednotlivých úseků řízené dopravní sítě. Jelikož jsou hodnoty tresholdů základním stavebním kamenem algoritmů tohoto typu, je správně zvolená hodnota tresholdů klíčová pro správnou funkčnost.

Jak již ale bylo zmíněno při popisu algoritmu LŘD používaného v ČR, použití této metody pro řízení systému LŘD není příliš vhodné z hlediska dlouhodobé prognózy kvůli dynamicky měnící se povaze dopravy a jejích parametrů. Nicméně i přes tyto nedostatky je tento typ řídicích algoritmů hojně využíván ve většině systémů LŘD implementovaných v reálném provozu. Důvod četného využívání je jeho jednoduchá implementace [8].

1.2.1.1 Příklady systémů LŘD ve světě a jejich výstupy

Systémy LŘD jsou implementovány v řadě světových zemí. Mezi tyto země patří například: Spojené Království, Nizozemsko, Spojené státy Americké, Německo, Austrálie, Nový Zéland či Švédsko [7] [9].

Jednotlivé implementace se liší ve výstupech algoritmů. Například algoritmus implementován ve Spojeném Království umožňuje řízení jednotlivých pruhů. Nad jednotlivými pruhy může být zobrazen symbol "X", značící že daným pruhem není povolen průjezd. Porušení takového zákazu je pak ve Spojeném Království penalizováno pokutou dosahující výše až 100 liber a uzmutí 3 bodů. Při zvýšené intenzitě se tento uzavřený pruh zpřístupní, čímž se výrazně zvýší kapacita dané komunikace. Příklad portálu LŘD ve spojeném království omezující vjezd do pruhu a nejvyšší povolenou rychlost je vyobrazen na obrázku 4. [10]



Obrázek 4: Uzavření jízdního pruhu a omezení rychlosti systémem LŘD ve spojeném království [11]

Ve spojených státech se liší definice jednotlivých systémů LŘD a to v určení, zda je zobrazené rychlostní omezení povinné, či nikoliv. Tabulka, ve které jsou jednotlivé systémy LŘD a jejich definice popsány je dostupná v citaci tohoto odstavce. Dále je v tabulce uvedeno, zda je daný systém řízen automaticky, manuálně či hybridně. Také jsou zde uvedené primární funkce jednotlivých systému (řízení kongescí, přizpůsobování meteorologickým vlivům nebo pracovním zónám). Ne všechny systémy LŘD jsou tedy ve Spojených státech Amerických určeny k omezování rychlosti. Na obrázku 5 je vyobrazen příklad portálu ve Spojených státech Amerických. [12]



Obrázek 5: Příklad portálu omezující nejvyšší povolenou rychlost v USA [12]

Na následujících obrázcích jsou pro názornost vyobrazeny příklady portálů systému LŘD v dalších světových zemích. Nutno podotknout že u Švédské implementace jsou omezení rychlosti generované systémem LŘD klasifikována jako doporučená.



Obrázek 6: Příklad portálu systému LŘD ve Švédsku [13]



Obrázek 7: Příklad portálu systému LŘD ve Spolkové republice Německo [14]

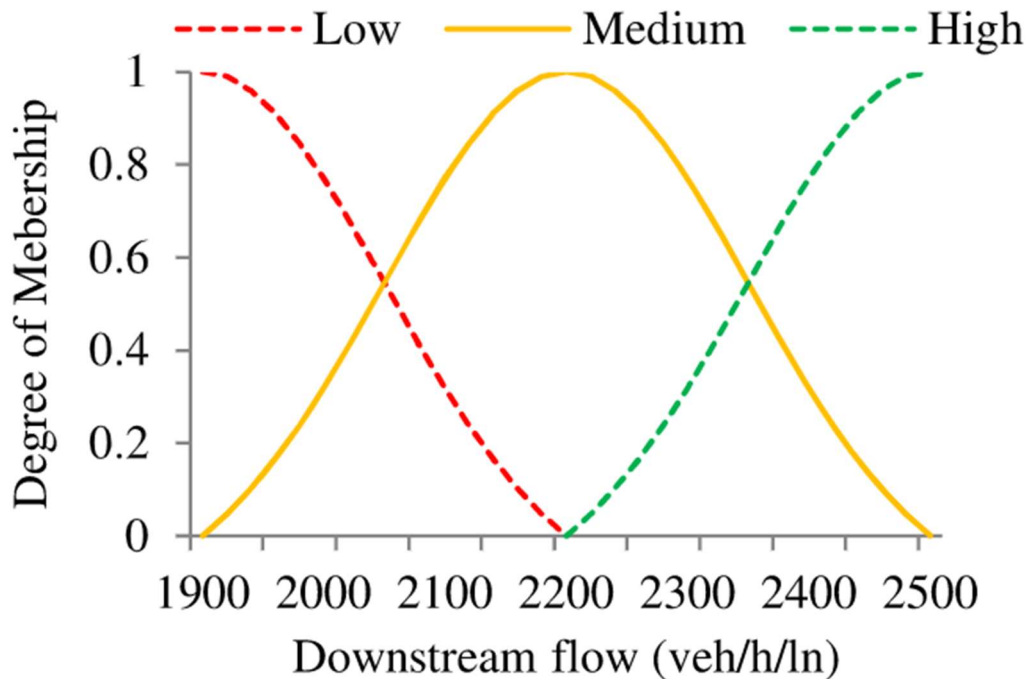
1.2.2 Algoritmy založené na fuzzy logice (fuzzy-logic based)

Jedná se o další z typů řídicích algoritmů systémů LŘD. Tento je založen na teorii fuzzy množin. Touto teorií se zabýval Jiří Beneš ve své diplomové práci, kde navrhoval řídicí algoritmus systému LŘD pro komunikaci D0 právě s využitím teorie fuzzy množin. Popis této teorie byl jím popsán takto:

„V roce 1965 byla poprvé profesorem Lofti A. Zadehem představena teorie fuzzy množin, která zevšeobecňuje klasický booleanovský přístup založený na bivalentních vstupech $\{0,1\}$, kde prvek buď do množiny patří nebo nepatří. Teorie fuzzy množin zavádí termín „míra příslušnosti“, který je přirozeným prostředkem pro formulaci vágnosti, pojmové neurčitosti. Fuzzy množiny tak umožňují matematický zápis sémantických neurčitých popisů, u kterých nejsem schopni přesně definovat hranice, jako jsou pojmy „málo“ nebo „hodně“. Míra (stupeň) příslušnosti nám umožňuje přiřadit příslušnost prvku k více množinám, a to v celém rozmezí hodnot $\langle 0,1 \rangle$.

Klasickým příkladem je posuzování sklenice, která není ani plná, ani prázdná. V klasické dvouhodnotové logice bychom takovou sklenici museli přiřadit buď do množiny „plná“, nebo „prázdná“, což by ale vedlo k velmi nepřesnému závěru, pokud by ve sklenici o objemu 0,5 litru bylo třeba 0,3 litru tekutiny. Případně by bylo možné definovat více množin s určitým rozmezím objem, kdy bude sklenic např. „poloprázdná“ v rozmezí 0,167 – 0,333 litru, což ale zvyšuje složitost rozhodování a abychom přesně vyjádřili stav naplnění sklenice, potřebovali bychom nekonečně mnoho množin s různým rozsahem objemu. V teorii fuzzy množin ale v takovém případě s výhodou použijeme intuitivní logiku a zapíšeme, že sklenice je např. S mírou příslušnosti 0,25 prázdná a s mírou příslušnosti 0,75 plná. Po vhodné defuzifikaci (zpětnému vyjádření konkrétní hodnoty) dospějeme ke skutečnosti, že ve sklenici je např. 375 ml tekutiny.“ [1]

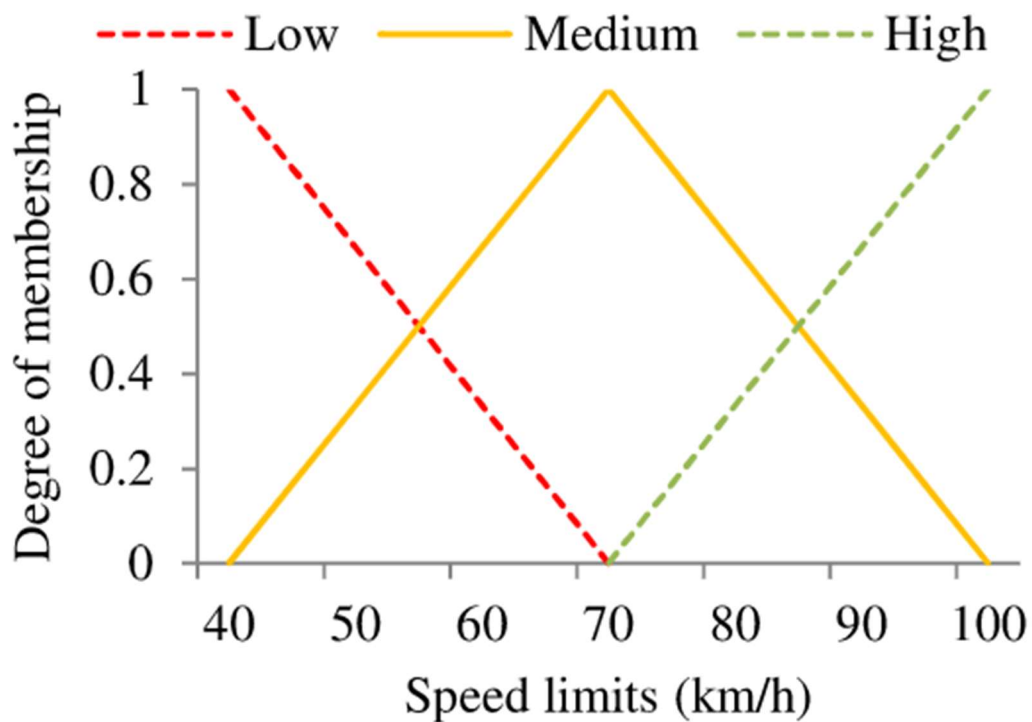
Takto je ve stručnosti popsána teorie fuzzy množin, která je využívána v řídicích algoritmech tohoto typu. Detailnější popis fuzzy logiky je možné dohledat ve skriptech systémové analýzy ČVUT [15]. Dle teorie jsou posléze jednotlivé číselné vstupy do algoritmu fuzzyfikovány a po této operaci již je řídicí logika navrhnutá nad těmito fuzzyfikovanými hodnotami. V grafu níže je vyobrazen příklad procesu fuzzyfikace ze studie z roku 2015 na datech z dálniční komunikace v Barceloně ve Španělsku.



Obrázek 8: Příklad procesu fuzzyfikace naměřených hodnot intenzit [8]

Na vodorovné ose jsou zobrazeny hodnoty hodinových intenzit, na svislé osy je vyobrazen tzv. stupeň příslušnosti. Stupeň příslušnosti udává v číselném rozmezí $\langle 0;1 \rangle$ jak moc daná hodnota přísluší definované třídě. Z grafu je zřetelné, že hodnota intenzity 1900 voz/hod přísluší fuzzy hodnotě s názvem “low“, tedy nízká, se stupněm příslušnosti 1. Jinými slovy hodnota naměřené veličiny plně odpovídá fuzzy množině “low“. Zato hodnota nacházející se na intervalu intenzity mezi 2000 voz/hod a 2100 voz/hod je na pomezí stupně příslušnosti fuzzy množin “low“ a “medium“, přičemž hodnoty blíže k hranici 2000 voz/hod mají vyšší stupeň příslušnosti ke fuzzy množině “low“ a hodnoty blíže k hranici 2100 voz/hod mají vyšší stupeň příslušnosti k fuzzy množině “medium“.

Obdobně jako vstupy, jsou v těchto typech algoritmů fuzzyfikovány i výstupy, jež jsou v tomto případě rychlostní limity. Příklad fuzzyfikace výstupu algoritmu je vyobrazen na obrázku 9. V tomto případě jsou na vodorovné ose vyobrazeny rychlostní limity, svislá osa zůstává stejná jako na obrázku 8, tedy stupeň příslušnosti.



Obrázek 9: Fuzzyfikace rychlostních omezení [8]

Nad touto fuzzyfikací je postavena řídicí logika, jenž je založena na zkušenostech operátora, expertních posudků a znalostech systému. Formát takové logiky je v podobě podmínek a reakcí na tyto podmínky, podobně jako v algoritmech typu “rule-based“. Pomocí podmínek je rychlostní omezení propojeno s naměřenými intenzitami.

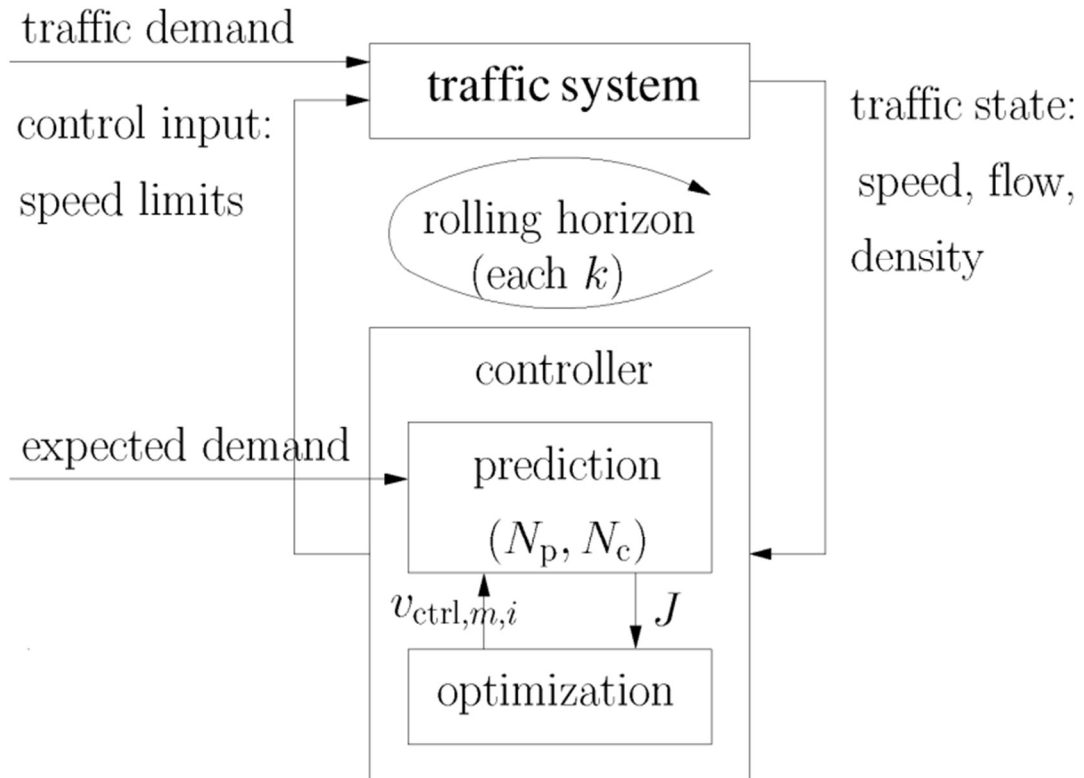
1.2.3 Algoritmy založené na teorii řízení (control-theory based)

Algoritmy založené na teorii řízení využívají k určení hodnot výstupů, jimiž jsou v případě těchto algoritmů rychlostní limity, výpočetní model, na jehož základě se snaží odhadnout budoucí vstupní hodnoty a pro tyto hodnoty určit optimální hodnotu výstupu. Princip algoritmů vychází z teorie řízení, ve které se řídicí systém snaží o regulaci řízeného prvku na základě veličin, které mu řízený prvek poskytuje. Jeden z hlavních rozdílů této metody oproti metodám, jež byly v rámci kapitoly popsány je právě způsob určení výstupních hodnot. Vstupní parametry se neliší oproti vstupním parametrům předchozích algoritmů, stále jsou jimi hodnoty intenzit, rychlostí, nebo obsazenosti, nicméně vstupní hodnoty v případě těchto algoritmů nejsou zpracovávány v podobě absolutních hodnot, kdy se na jejich základě odvodí výstupní hodnota, jako je tomu v případě "rule-based" algoritmů nebo v podobě fuzzyfikovaných hodnot, jako je tomu v případě fuzzy algoritmů. V algoritmech založených na řídicí teorii jsou vstupní hodnoty použity jako vstupní hodnoty pro výpočetní model, jež je součástí těchto algoritmů [16].

Za předpokladu použití vhodného výpočetního modelu je algoritmus schopen pohlížet na řízenou komunikaci nejen pohledem jednotlivých řezů, kde se nachází jednotlivé detektory, ale dokáže pohlížet na řízenou komunikaci jako na celek. Takový přístup může algoritmu poskytnout detailní popis řízené dopravní sítě, což mu dává oproti ostatním algoritmům víc možností pro optimalizaci řízení řízené oblasti. Nemusí být totiž zcela závislý na jednotlivých řezech, ale je schopen za předpokladu užití vhodného dopravního modelu simulovat situaci na jakémkoliv místě sledované komunikace. Vhodně zvolený dopravní model navíc umožní projekci dopravní situace v budoucím časovém horizontu a možnost řídicí reakce ještě před vznikem simulované situace [16].

Další výhodou přístupu může být předejití jistého typu nestability, která může vzniknout mezi vznikem rušivého jevu v dopravním proudu a reakcí na něj. Z teorie řízení systému je známo, že právě zpoždění mezi vznikem rušivého jevu a akcí řídicího prvku může způsobit nestabilitu. Užitím prediktivního řízení je možné těmto typům nestability předejít a řízení jistým způsobem homogenizovat.

V teorii algoritmů založených na řídicí teorii je zaveden pojem MPC (Model Predictive Control), jež je základním prvkem popisovaných algoritmů. Ilustrační schéma prvku MPC je vyobrazeno na obrázku 10. Následující odstavec obsahuje popis jednoho iteračního kroku, jak jej popsali autoři studie zabývající se optimální koordinací proměnlivých rychlostních omezení pro snížení počtu rázových vln.



Obrázek 10: Základní schéma MPC (Model Predictive Control) [16]

„Základním prvkem MPC je tzv. “controller“. Tento prvek v každém časovém kroku k spočte optimální řídicí signál na základě poskytnutých vstupních parametrů v predikčním horizontu N_p . Řídicí horizont $N_c (< N_p)$ slouží ke snížení počtu proměnných a ke zvýšení stability celého systému (Poté co byl řídicí horizont překročen, je řídicí signál obvykle brán jako konstanta). Ze spočteného optimálního řídicího signálu, pouze první vzorek $k + 1$ je aplikován do procesu. V dalším časovém kroku je vytvořena nová optimalizace a opět, pouze první prvek je použit v dalším procesu. Takto je proces opakován stále dokola.

Toto schéma, nazvané “rolling horizon“ umožňuje aktualizaci vstupních dopravních parametrů nebo aktualizaci samotného výpočetního modelu v každé iteraci. Výhodou aktualizace stavu dopravního modelu je snížení chyb predikce. Pravidelnou aktualizací modelu je možné dosáhnout adaptivního řídicího systému, jež je schopen i při velkých změnách predikované situace stále vykazovat relativně správné výsledky. Také je možné pravidelnou aktualizací dosáhnout větší efektivity modelu v budoucnosti, kdy se výpočetní model každou provedenou aktualizací více přizpůsobuje systému, pro který má být predikce spočtena.“ [16]

Citovaný článek jako výpočetní model použil makroskopický model dopravního proudu s názvem METANET a dosáhl pomocí algoritmu založeném na řídicí teorii snížení času stráveného na simulované komunikaci o 17,3 % [16].

1.3 Shrnutí

V zásadě existují 3 typy teoreticky popsaných přístupů v oblasti řízení dálničních sítí pomocí proměnných rychlostních omezení. Přístup nejvíce využíván ve realizovaných řešeních se nazývá “rule-based“, v překladu do češtiny “založen na pravidlech“. Algoritmy typu “rule-based“ načítají naměřené hodnoty dopravních parametrů, kterými mohou být např. intenzita, obsazenost, rychlost, atd. a na tyto hodnoty předdefinovaným způsobem reagují. Předdefinovaný způsob reakce jsou pravidla (proto název přístupu “rule-based“), která pouze určují, jaké zobrazit rychlostní omezení, pokud naměřené hodnoty přesáhnou předdefinovanou hodnotu. Nevýhodou tohoto přístupu je nemožnost algoritmu přizpůsobit se dynamické změně povahy dopravního proudu. Počet vozidel v dopravním proudu meziročně narůstá o 4% [3] [4], aby mohl být algoritmus typu “rule-based“ efektivní i z hlediska dlouhodobé prognózy, je nutná jeho kalibrace po delším časovém horizontu. Algoritmus tohoto typu je implementován a v provozu také v ČR.

Dalším typem algoritmů jsou tzv. “fuzzy-logic“ algoritmy. Tyto algoritmy fungují relativně podobně jako “rule-based“, ale na rozdíl od algoritmů typu “rule-based“ nepoužívají jako rozhodující mezníky číselné hodnoty, ale fuzzy množiny. Tato skutečnost utváří z algoritmů tohoto typu efektivnější řešení, kdy hodnoty vstupů jsou prakticky rozděleny do intervalů, kterým je přiřazeno slovní ohodnocení, což vyústí v efektivnější řízení dopravního proudu, nicméně stále je možné narazit na úskalí v podobě ztráty efektivity z hlediska dlouhodobější prognózy. Tím, že jsou fuzzyfikovány vstupní hodnoty naměřeny za určité časové období, je možné že fuzzyfikace bude potřeba být zopakována po určitém časovém intervalu. Nicméně samotná fuzzyfikace výrazně usnadňuje implementaci řídicí logiky a její možné budoucí revize.

Posledním typem teoreticky popsaných algoritmů jsou tzv. “control-theory“ algoritmy jež využívají k řízení predikci výpočetních modelů. Výhodou tohoto přístupu je, při užití dopravního výpočetního modelu, možnost rozhodování na základě veličin z celé sledované dopravní sítě, nikoliv pouze z jednotlivých řezů. Při správném návrhu si algoritmus provádí kalibraci sám při každé iteraci, není tedy nutné kalibraci provádět ručně, jako tomu může být u ostatních typů algoritmů. Tato skutečnost může vést k uspokojivějším výsledkům z hlediska dlouhodobé prognózy.

Obecně mají všechny algoritmy víceméně stejné vstupní parametry. Jsou jimi hodnoty naměřené dopravními detektory rozmístěnými po sledované dopravní síti. Těmito vstupy mohou být hodnoty intenzit, obsazenost detektoru, případně hodnoty rychlosti.

Výstupy jsou napříč typy algoritmů stejné. Jsou to hodnoty rychlostních omezení, které mají být zobrazeny na jednotlivých řezech.

Z provedené analýzy vyplývá jako nejlepší možnost pro návrh adaptivního algoritmu pro VLŘD využít typ algoritmu založený na řídicí teorii, jež se jeví jako optimální varianta, ze které vycházet při návrhu.

Kapitola 2

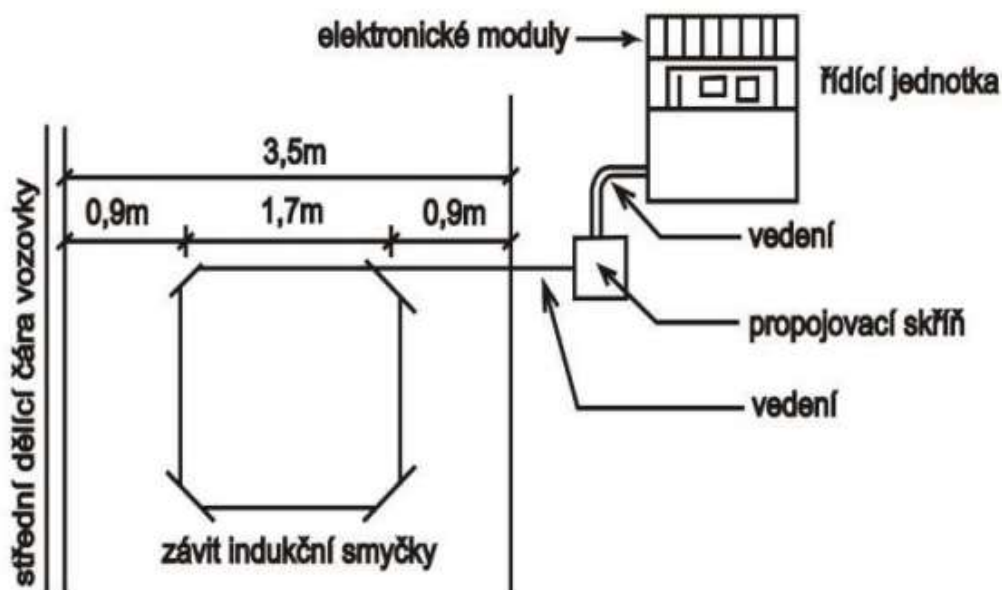
Datové zdroje pro VLŘD

Předmětem této kapitoly je rešerše aktuálně dostupných datových zdrojů, které by mohly být využity v rámci algoritmu VLŘD. Datové zdroje musí poskytovat relevantní dopravní data, která dokážou s určitou přesností reflektovat aktuální stav dopravního proudu v místě měření.

Sběr dopravních dat je prováděn různými organizacemi po celém území české republiky. Podrobné rozebrání všech detekčních prvků používaných v ČR by bylo nad rámec této práce a již na toto téma v minulosti bylo napsáno mnoho publikací, proto byla vybrána detekční metoda aktuálně nejvíce využívaná organizací Ředitelství silnic a dálnic, protože ŘSD svojí působností pokrývá největší oblast silničních komunikací České republiky.

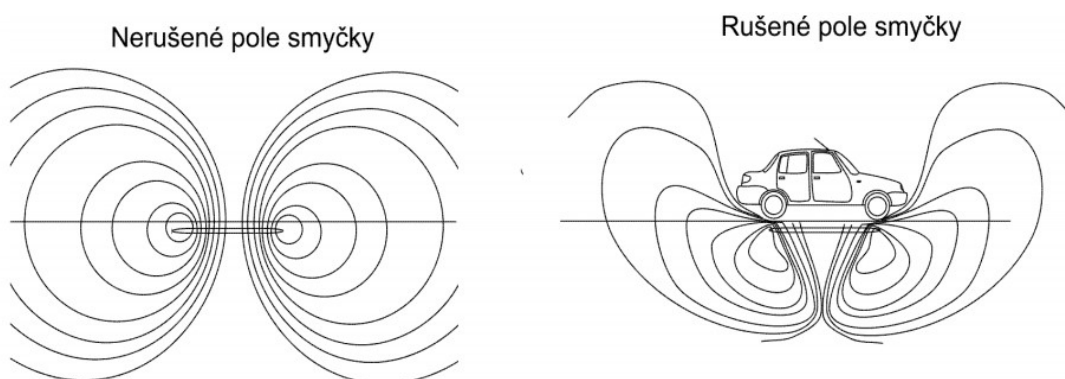
2.1 Metoda detekce indukční smyčkou

Indukční smyčky jsou destruktivní elektromagnetické detektory se stacionárním polem, které se umísťují pod povrch vozovky, a to do hloubky 30 – 60 mm [2]. Indukční detektor se skládá z jednoho nebo více závitů izolovaného metalického vodiče, Propojovací skříně (zesilovače) a řídicí jednotky. Základní konstrukce indukční smyčky je situována na obrázku níže [17].



Obrázek 11: Schéma konstrukce indukční smyčky [18]

Do závitů je pouštěn střídavý elektrický proud o frekvenci 20-150 kHz. Závit začne generovat elektromagnetické pole, a pokud nad smyčkou projede vozidlo, je toto pole ovlivněno kovovou karoserií vozidla. Narušení vozidlem poté řídicí jednotka vyhodnotí jako průjezd vozidla díky změně indukčnosti cívky. Situace je znázorněna na obrázku 12.



Obrázek 12: Narušení magnetického pole kovovou karoserií vozidla [18]

Nicméně změna indukčnosti cívky může být velmi malá, je proto nutné brát v potaz i ztráty na přívodním kabelu.

Jednotlivé detekční prvky jsou logicky řazeny za sebou na dálnicích a většině silničních komunikací 1. třídy. Použití více prvků je z důvodu zvýšení kvality snímaných informací, rozšíření snímané oblasti a získání nových veličin (např. hustota). Indukční detektor je destruktivní prvek, což značí že pro přidání nového detektoru je nutné počítat se stavebními náklady. Cena samotného detektoru se pohybuje okolo 10 tisíc Kč a ještě je nutné k nim připočítat cenu za úpravu vozovky. I přes to je tato metoda detekce považována za levnější možnost.

Tato metoda detekce je schopna detekovat následující parametry [17]:

- Intenzita vozidel [$voz \times h^{-1}$]
- Klasifikace vozidel [-]
- Obsazenost [%]

Při použití kombinace dvou či více těchto detektorů je možné detekovat ještě tyto parametry navíc [17]:

- Rozlišení směru jízdy [-]
- Rychlost [$km \times h^{-1}$]

Směr jízdy je možné pomocí tohoto detektoru naměřit, pokud je dostupný sekundární detektor. Parametry délku vozidla a výšku vozidla lze odvodit z klasifikace vozidla, kterou tento detektor poskytne, nicméně klasifikace poskytnutá detektorem nemusí odpovídat realitě. Je to způsobeno tím, že klasifikace je v tomto případě přiřazení typu vozidla změně magnetického pole. Za jistých podmínek se ale může stát, že změnu magnetického pole oklasifikovanou jako "nákladní automobil s přívěsem" vyvolá ve skutečnosti vozidlo mnohem menších rozměrů.

V rámci VLŘD je schopna tato technologie detekovat intenzitu v profilovém řezu komunikace s velmi vysokou přesností. Jedná se tedy o důležitý datový zdroj, který je schopen reflektovat aktuální situaci v řezech celé komunikace. Jak již bylo zmíněno v předchozí kapitole, indukční smyčky jsou umístěny v každém místě, kde se nachází portál systému LŘD. Proto je tento datový zdroj považován za jeden z nejdůležitějších v rámci celého navrhovaného systému VLŘD. Pomocí tohoto datového zdroje je možné kalibrovat dopravní model, ze kterého by algoritmus VLŘD vyhodnocoval budoucí stav dopravy [17].

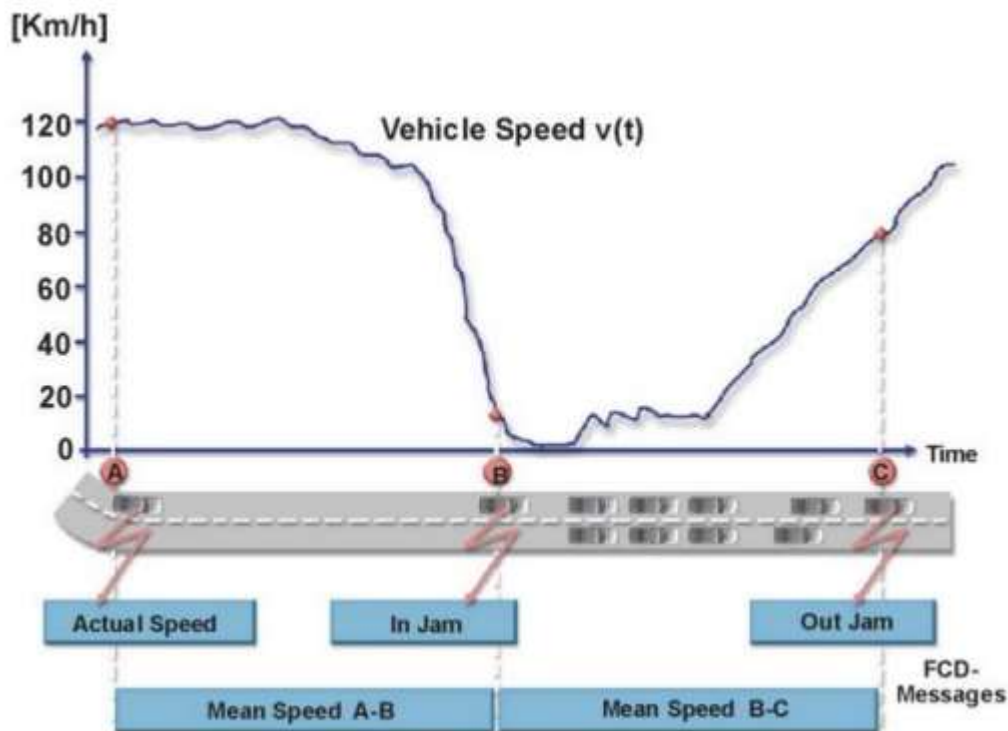
2.2 Metoda detekce FCD (Floating Car Data)

Dalším důležitým datovým zdrojem, který by mohl reprezentativně zobrazovat aktuální stav dopravního proudu je metoda detekce FCD (Floating Car Data). Pro tuto metodu detekce je charakteristické, že detektor se nenachází na silniční infrastruktuře, nýbrž uvnitř samotného vozidla.

FCD technologie je spolehlivý a nákladově efektivní způsob sběru přesných dopravních dat pro širokou dopravní síť. Je založena na sběru real-time dat ze zařízení umístěných ve vozidlech, která jsou odesílána přes datovou síť mobilních operátorů. Jelikož jsou data odesílána na centrální server, kde jsou posléze vyhodnocována, jedná se o centralizovaný systém.

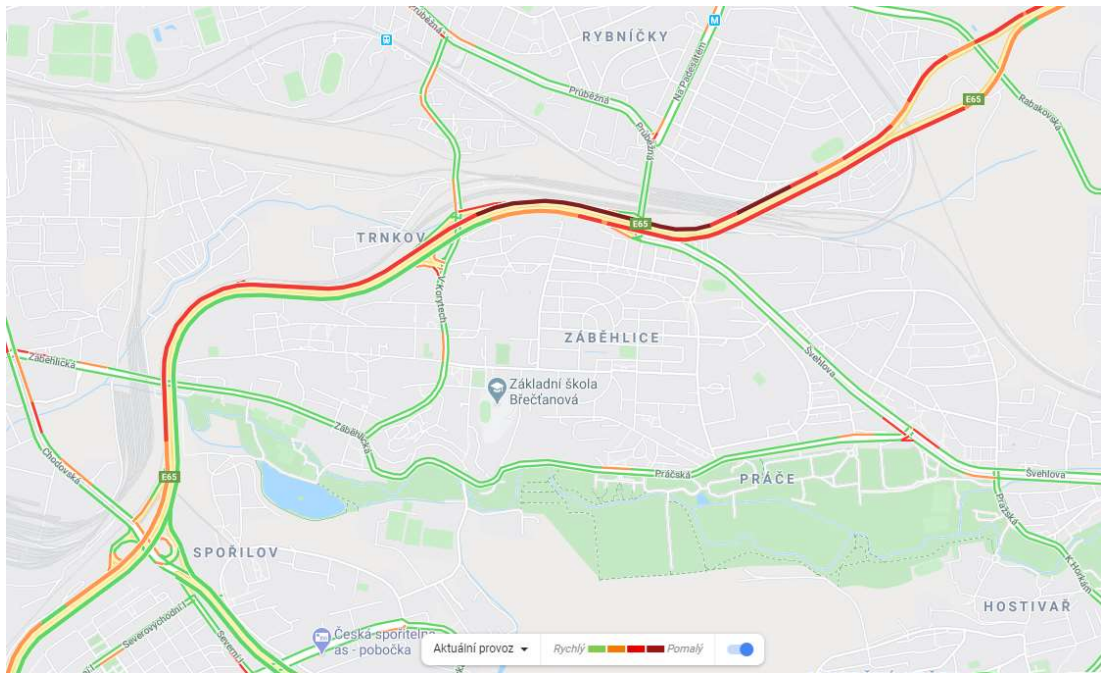
Data, která jsou odesílána na centrální server obsahují anonymizované informace o rychlosti, poloze vozidla a dojezdové časy. Vozidlo, které tato data odesílá víceméně slouží jako sonda umístěná v dopravním proudu a na základě pohybu této sondy se dopravní systém snaží odvodit, jak se v daném místě dopravní proud chová.

Pokud se podíváme do světového měřítka, zjistíme že tato metoda detekce je již ve velké míře využívána. Necht' jsou příkladem platformy google maps, či mobilní aplikace Waze. Obě tyto platformy využívají mobilních telefonů cílového uživatele, aby z nich získaly anonymizované informace o poloze a rychlosti. Pokud cílový uživatel používá navigační aplikace těchto dvou společností, automaticky se z jeho mobilního telefonu stává detekční zařízení. U obou platforem má již uživatel možnost i určité typy dopravních událostí zadávat, pokud ještě nejsou v mapových podkladech zobrazeny a systém o nich neví. Jako příklad této dopravní informace může být informace o dopravní nehodě, či uzavírce. Událost vytvořená uživatelem je poté zobrazena v mapových podkladech a je uvážena při výpočtu dojezdových dob. Z těchto informací je možné relativně určit, jak by mohlo vypadat chování dopravního proudu kolem tohoto vozidla. Pro určení chování dopravního proudu vycházíme z principu, který je zobrazen na obrázku 13 [17].



Obrázek 13: Vykreslení rychlosti v čase měřeného vozidla [19]

Na obrázku 13 je zobrazen graf závislosti rychlosti sledovaného vozidla na čase. Protože poloha vozidla je známá, systém má informaci, na jaké komunikaci se vozidlo aktuálně nachází. Dlouhodobým sledováním je určena průměrná rychlost při volném proudu na této komunikaci, v případě obrázku 13 může být průměrná hodnota 120 km/h. Pokud vozidlo přejde z volného proudu do kongesce, promítne se tento stav na aktuální rychlosti vozidla, kde se její hodnota oproti průměrné hodnotě razantně sníží. Tato událost je detekovatelná a značí změnu chování dopravního proudu. Podle velikosti odchylky od hodnoty volného proudu je možné klasifikovat stupně dopravy. Pokud se rychlost plovoucího vozidla blíží k nulové hodnotě, je možné usoudit že propustnost komunikace je velmi nízká. Tuto informaci je možné zobrazit nad mapovými podklady a prezentovat ji koncovému uživateli. Příklad prezentace dat z plovoucích vozidel je uveden na obrázku 14 [17].



Obrázek 14: Způsob prezentace FCD dat [20]

Prezentace dat je na této platformě realizována barevnou škálou, které odpovídají hodnotám rychlostí přijatých od plovoucích vozidel. Barevná škála je v tomto případě v mezích od zelené barvy po tmavě červenou a značí průměrnou rychlost aktuálního provozu, přičemž je-li barva zelená na komunikaci, je provoz rychlý, je-li červená, provoz je pomalý. Pod pojmem “Aktuální provoz“ se skrývá pouhé přiřazení barvy intervalu rychlosti. Jinými slovy, od plovoucího vozidla dorazí na centrální server informace o jeho aktuální rychlosti a poloze, systém pomocí předdefinovaných klasifikačních pravidel rychlost oklasifikuje a výsledek zobrazí na mapových podkladech pod odpovídající barvou. Pro příklad, pohybuje-li se vozidlo na komunikaci rychlostí 5 km/h a hodnota rychlosti volného proudu na této komunikaci je 80 km/h, bude stav dopravy na tomto místě zobrazen v barevné škále barvou tmavě červené.

Jak již bylo uvedeno, pro zvýšení kvality měřených veličin, je vhodné nevyužívat k detekci pouze jeden zdroj, ale využít zdrojů více. Na tuto podmínku je při použití FCD technologie nutno brát ještě větší ohled než u metod jiných. Je to z toho důvodu, že pokud by bylo použito pouze jedno plovoucí vozidlo na zjištění stavu dopravy v určitém místě, tak výsledek přijatý z tohoto vozidla nemusí nutně odpovídat skutečnému stavu dopravy. Může se stát, že plovoucí vozidlo sníží svou rychlost nezávisle na dopravním proudu kolem něj, či může dojít ke zkreslení při určování polohy vozidla satelitem GNSS. Také platí, že pokud bychom použili pouze jedno plovoucí vozidlo, není možné určit stav dopravy v jiném místě, než v jakém se vozidlo nachází. Proto je vhodné použít plovoucích vozidel tolik, kolik je systémem umožněno. U detekčních prvků umístěných na dopravní infrastruktuře je realizace více detektorů zdrojově i stavebně složitý proces. U FCD technologie tomu tak není [17].

Pokud jsou data o pohybu vozidel odesílána při využívání navigační platformy, tak počet plovoucích vozidel je roven počtu uživatelů této platformy. Například společnost Google

Inc. Na svých webových stránkách uvádí, že jejich navigační platforma Google maps, má měsíčně 1 bilion aktivních uživatelů po celém světě. To znamená, že má k dispozici 1 bilion plovoucích vozidel, které každý den odesílají až 25 milionů zpráv o svém pohybu, jež jsou centrálně zpracovávány a prezentovány [19]. S tímto množstvím zdrojů je možné poskytovat relativně přesné informace o dopravním proudu. Zároveň je na tomto příkladu demonstrováno, jak vysoké náklady jsou potřeba pro realizaci nových zdrojů dopravních dat. Náklady jsou téměř zanedbatelné. Z hlediska dopravního systému, každý nový uživatel, který si aplikaci stáhne na svůj mobilní telefon, se automaticky stává novým dopravním detektorem. Jediné, čím je v tomto případě dopravní systém omezován je přiřazený výpočetní prostor.

FCD technologie je velmi závislá na penetraci do vozidel pohybujících se v dopravním prostoru. Pokud je penetrace vysoká, je možné FCD technologii považovat za důvěryhodný zdroj dopravních informací. Je-li penetrace nízká nikoliv.

Jak již bylo zmíněno, z technologie FCD je možné získat tyto atributy dopravního proudu:

- Rychlost [km/h]
- Poloha
- Jízdní doba [s]

Tato technologie v kombinaci s technologií detekce indukční smyčkou vytváří unikátní datový zdroj, jenž je schopen ve velkém geografickém měřítku reprezentativně reflektovat aktuální stav dopravy a proto by tato technologie měla být zahrnuta v návrhu VLŘD [17].

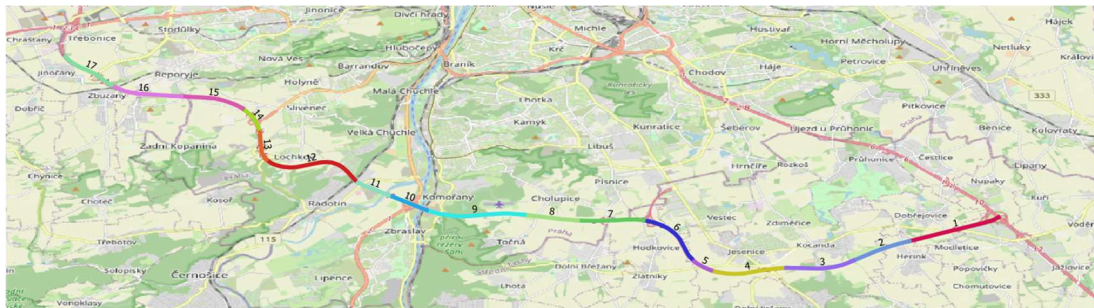
2.3 Metoda detekce pomocí technologie Bluetooth / Wifi

Jedná se o poměrně novou metodu detekce, jež nebyla dosud řádně využívána. Princip technologie spočívá v detekci průjezdu vozidla pomocí bezdrátové technologie Bluetooth/wifi. Metoda spoléhá na skutečnost, že se v projíždějícím vozidle nachází zdroj Bluetooth/wifi technologie. V praxi to může znamenat, že se ve vozidle nachází řidič, který má u sebe mobilní zařízení disponující touto technologií. Detektor je schopen tento zdroj zaměřit a vyčíst z něj sílu signálu a mac adresu. Mac adresa slouží k párování záznamu průjezdu napříč detektory. Síla signálu může sloužit k určení směru, kterým vozidlo jelo.

Na pilotní lokalitě D0 jsou tyto detektory umístěny u každé RSU jednotky, celkové je jimi pokryta v zásadě celá lokalita. Jelikož je cílem této kapitoly analyzovat vhodné datové zdroje, jež by mohly být použity v rámci návrhu adaptivního algoritmu pro VLŘD, je v rámci této kapitoly zanalyzován i tento zdroj.

2.3.1 Analýza datových výstupů Bluetooth/Wifi detekce

Pro analýzu této metody detekce byla pilotní lokalita D0 rozdělena na 17 imaginárních detekčních úseků, kde hranice jednoho úseku je právě jeden detektor. Situace rozdělení je vyobrazena na obrázku 15.



Obrázek 15: Imaginární detekční úseky na pilotní lokalitě D0

Rozdělení testovací lokality do úseků, jenž jsou uvedeny na obrázku je nutné z hlediska vlastností detekční technologie. Detektory v tomto odvětví nejsou aktuálně schopny bodové detekce, tzn. Že záznam detekce neobsahuje relevantní informace o intenzitě, obsazenosti či bodové rychlosti. Intenzitu v bodě detekce by bylo možné určit z počtu zachycených zařízení v časovém okamžiku, nicméně nelze v tomto případě označovat záchyt jako bodový z důvodu rádiusu dosahu signálu, který je u Bluetooth technologie sice nižší (cca 10 metrů u mobilních zařízení [21]) ale u wifi technologie již dosahuje relativně vysokých hodnot (cca 100 metrů [22]). Zároveň technologie nezaručuje detekci všech vozidel v dopravním proudu.

Z těchto důvodů se jeví jako lepší možnost využít technologii detekce Bt/Wifi k úsekové detekci. Proto byla pilotní lokalita rozdělena do imaginárních detekčních úseků, na kterých se počítala průměrná dojezdová doba a průměrná rychlost.

K analýze byla využita data společnosti ŘSD z měsíce únor z roku 2022. Bt/Wifi

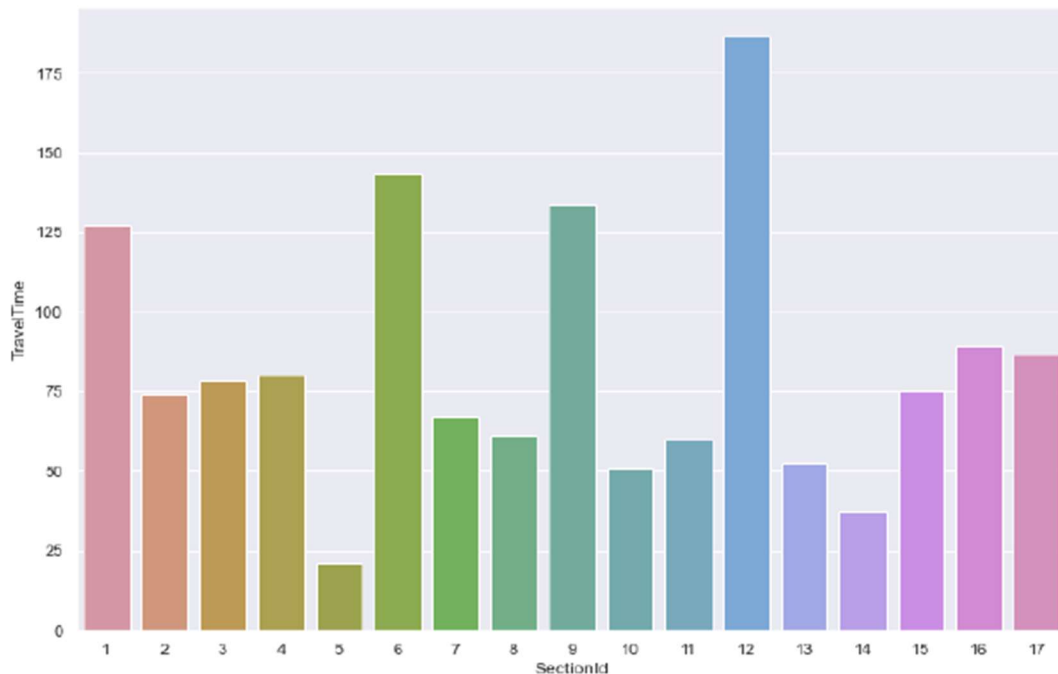
V době psaní této práce obsahoval záznam tyto relevantní informace, pomocí kterých bylo možné spočítat dopravní data pro algoritmus VLŘD:

- Čas detekce
- DeviceId (zašifrovaná MAC adresa detekovaného zařízení)
- Poloha detektoru
- Technologie

Čas detekce je nutný pro určení cestovní doby detekovaného vozidla. DeviceId je parametr užitý ke spárování záznamu detekce mezi dvěma detekčními body. Poloha detektoru je použita k přiřazení záznamu detekce do geografického kontextu. Parametr technologie určuje, kterou technologií byl záznam zachycen. Zde mohou být jen dvě hodnoty – buď wifi nebo Bluetooth.

Možný výstup Bluetooth/Wifi detekce po spárování je vyobrazen na obrázku 16, kde jsou vykresleny jednotlivé úseky na ose x a na ose y jsou vykresleny dojezdové doby. Obrázek

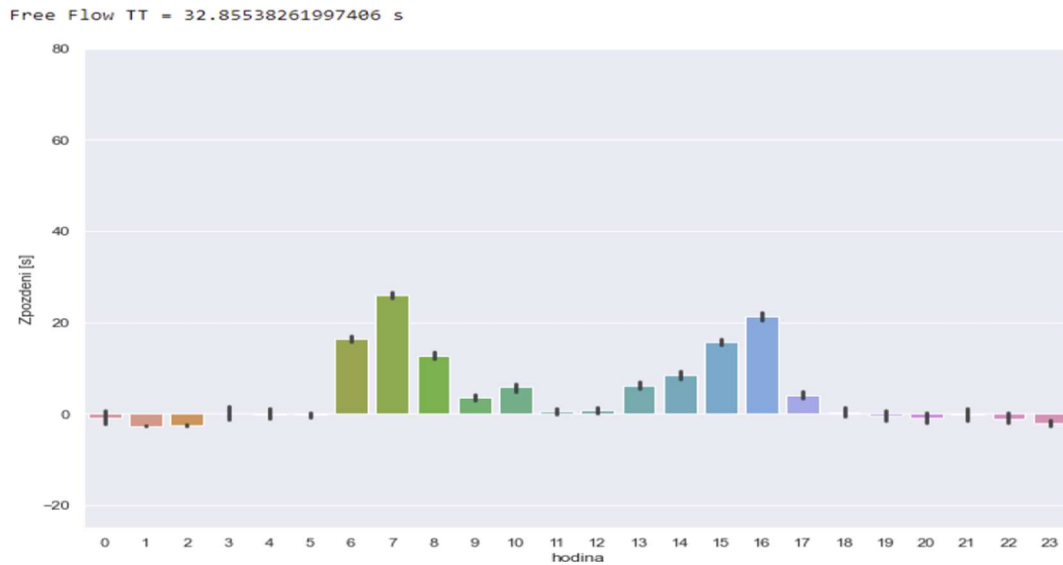
tedy zobrazuje průměrné dojezdové doby jednotlivých úseků napříč celým měsícem. Směrnost, kterou je také možné dopočítat z časových rázitek „surových“ záznamů v tomto obrázku rozdělena není. Graf tedy zobrazuje průměrnou jízdní dobu v obou směrech.



Obrázek 16: Průměrné dojezdové doby v jednotlivých úsecích za celý měsíc v obou směrech z obou detekčních technologií

Prvním pohledem na obrázek 16 a mapu úseků lze usoudit, že vyobrazené TT by mohly odpovídat realitě, protože jejich hodnota v jednotlivých úsecích víceméně odpovídá délce úseku a omezení rychlosti v něm.

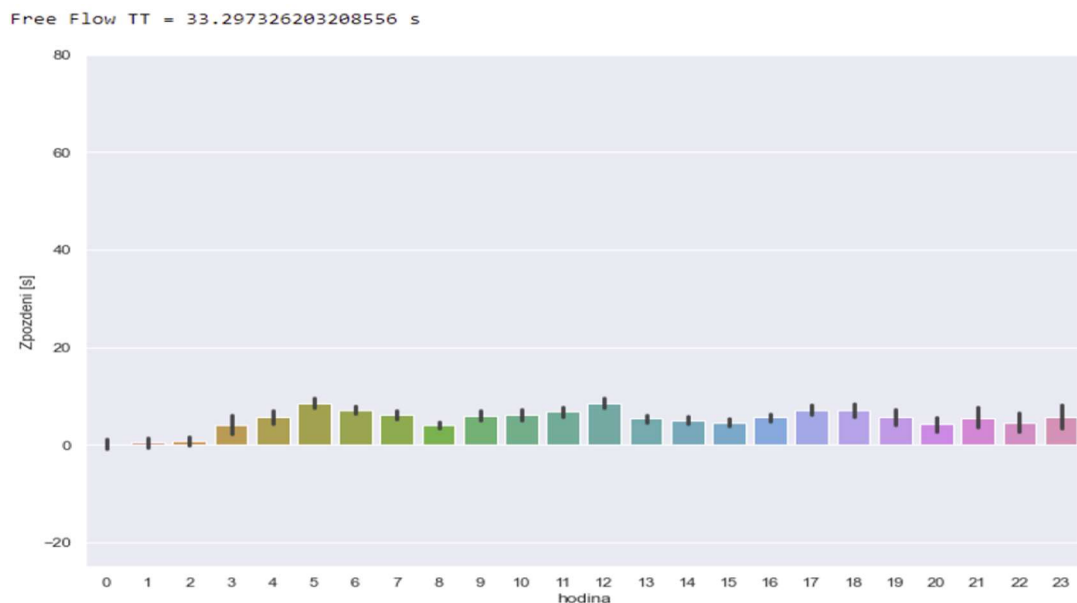
Pro další analýzu byly vybrány úseky, kde se předpokládá zvýšené, či snížené dopravní zatížení v jednotlivých směrech. Jako úsek, kde se předpokládá zvýšené dopravní zatížení byl vybrán úsek č. 14, který pokrývá část D0 s nájezdem z Barrandova. Tento úsek kvůli zmíněnému nájezdu bývá ve špičkových hodinách vytížený ve směru D1 -> D5. Naopak ve směru D5 -> D1 bývá úsek vytížený méně. Obrázek 17 vyobrazuje průměrné hodnoty zdržení v hodinách, které byly spočteny z hodnoty dojezdové doby při volném proudu. Tato hodnota byla pro potřeby analýzy spočtena jako nejčastější hodnota v hodinách, kdy není úsek zatížen. Výsledné zdržení bylo spočteno jako rozdíl průměrné hodnoty dojezdové doby oproti hodnotě dojezdové doby ve volném proudu. Hodnota dojezdové doby při volném proudu je uvedena v pravém rohu grafu. Oba výstupy zobrazeny níže jsou z obou technologií.



Obrázek 17: Průměrné zdržení na úseku č.14 ve směru D1 -> D5

Z grafu je patrné, že zdržení začne narůstat v hodinách, kdy se intenzita dopravy začíná blížit saturaci a klesat v hodinách kdy intenzita dopravy také klesá.

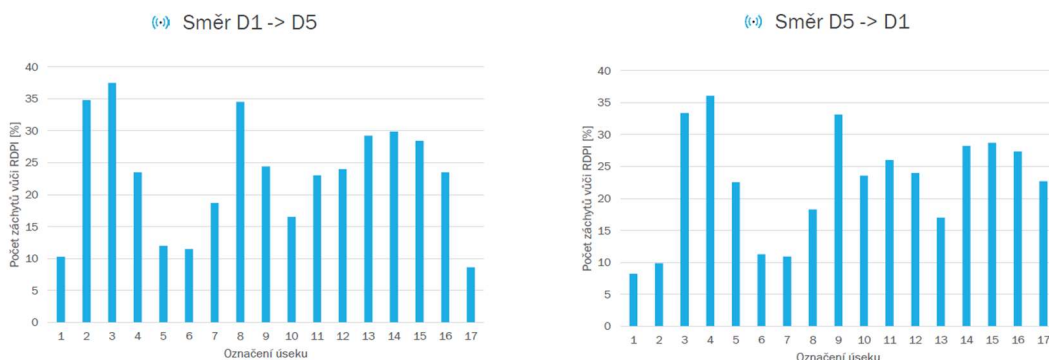
Na dalším obrázku je vyobrazen stejný úsek, avšak v druhém směru, tedy z komunikace D5 do komunikace D1.



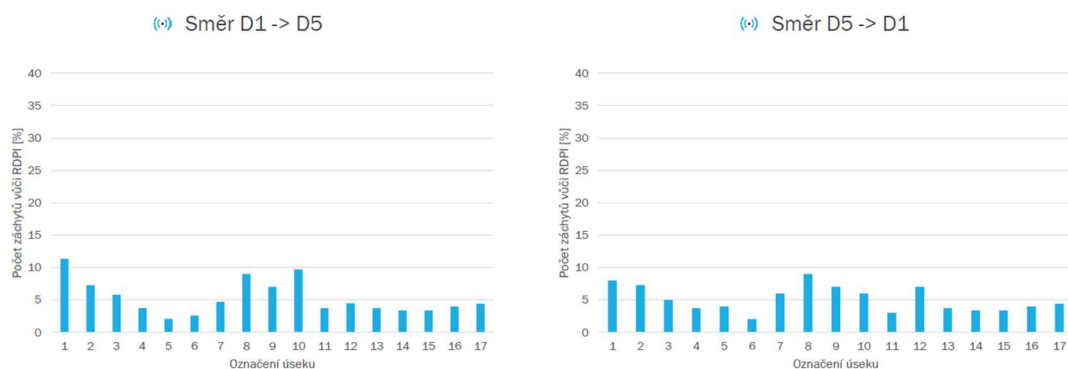
Obrázek 18: Průměrné zdržení na úseku č.14 ve směru D5 -> D1

Jak lze vidět na obou grafech, hodnota dojezdové doby při volném proudu je v obou směrech téměř ekvivalentní. Na obrázku 18 lze vidět, že zpoždění je zde napříč hodinami víceméně stejné. Nicméně hodnota dojezdové doby při volném proudu na obrázku 18 by mohla v tomto případě být zvolena lépe. I přes to ale lze konstatovat, že vyobrazené situace odpovídají předpokladu, tedy že ve směru ke komunikaci D5 je vyšší vytížení, než je tomu ve směru opačném.

Jako poslední analýza v rámci této detekční technologie bylo provedeno porovnání počtu spárovaných detekcí jednotlivých vozidel pomocí technologie Bt/Wifi oproti denním průměrným intenzitám naměřených při celostátním sčítání dopravy v roce 2020. V této analýze byly technologie rozděleny dle typu detekce a byly zprůměrovány denní počty záchytů. Následně byly počty rozděleny dle směrovosti. Výsledné hodnoty jsou zobrazeny v procentuální hodnotě ku RDPI. Výstupy jsou vyobrazeny na obrázku 22 a 19.



Obrázek 19: Počet detekovaných vozidel ku RDPI vyjádřeno v % - Bluetooth



Obrázek 20: Počet detekovaných vozidel ku RDPI vyjádřeno v % - Wifi

Z grafů je patrné, že detekcí technologií Wifi bylo detekováno menší množství projetých vozidel, než technologií Bluetooth. Lze tedy tvrdit, že detekce vozidel technologií Bluetooth je přesnější než detekce druhou technologií. Nicméně nelze v rámci této analýzy vyloučit, že na některých Wifi detektorech došlo k problému a kvůli tomu je měření zkreslené. Proto je nutné označit výše dva uvedené grafy jako orientační, nikoliv jako směrodatné. Hlubší analýza, která by přesně určila procentuální zastoupení detekce vozidel jednotlivými detektory, není předmětem této DP. Tato analýza sloužila pouze ke rámcovému určení, zda jsou data z těchto relevantní a k orientačnímu pohledu, kolik záchytů z celkového dopravního proudu, je možné touto technologií detekovat.

2.3.2 Závěr analýzy detekce pomocí technologie Bluetooth/Wifi

Z provedené analýzy vyplývá, že dopravní data získána technologií detekce Bluetooth/Wifi mohou reflektovat alespoň částečně aktuální stav dopravního proudu. Nelze ovšem konstatovat, že výstupy z této metody detekce jsou reprezentativní. To je možné dokázat pouze hlubší analýzou této problematiky, nicméně to není zadání této DP. Tato DP pouze analyzuje jejich využití v rámci algoritmu VLŘD, jehož návrh má být výstupem.

Z hlediska využití v algoritmu VLŘD je možné na základě provedené analýzy využít data z popsané technologie detekce. Nicméně není doporučeno využít tato data jako hlavní zdroj, na kterém by byl algoritmus VLŘD postaven. Důvodů k tomu je několik. Detekční data získaná touto technologií mají určité zpoždění oproti skutečnému stavu. Je to způsobeno tím, že aby mohla být vyhodnocena doba průjezdu vozidla, je nutné aby vozidlo daný úsek projelo. Až poté co vozidlo daný úsek projede je možné teprve přistoupit k vyhodnocení průjezdu.

Dalším nedostatkem je neschopnost technologie detekovat všechna vozidla dopravního proudu. Tento nedostatek je způsoben tím, že ne všechna vozidla mají uvnitř sebe Bt/Wifi vysílač/přijímač který detektor detekuje. Výsledné hodnoty počtů vozidel tedy neodpovídají realitě. Dále může technologie nabývat zkreslení, a to v případě, že se v jednom vozidle nachází více vysílačů (např. ve vozidle se nachází 4 cestující, kteří mají všichni u sebe mobilní telefon s bt/wifi vysílačem).

V budoucnu je také nutné brát v potaz vývoj Bt/Wifi vysílačů, kdy stále více výrobců preferuje z důvodu ochrany osobních údajů spotřebitelů zavádění tzv. anonymizace. Anonymizací je myšlena změna MAC adresy po každém připojení k přijímači, případně změna MAC adresy po daném časovém intervalu. Tato skutečnost by mohla v budoucnu znemožnit detekci, jelikož párování průjezdů je realizováno právě pomocí zašifrovaných MAC adres.

Nicméně i přes tyto nedostatky je aktuálně možné využít data z této detekce v rámci algoritmu VLŘD pro zpřesnění výstupů jiných detekčních technologií (ASD, FCD) jež jsou v algoritmu VLŘD využívány jako primární zdroje. Velkou výhodou této technologie je cena výstavby detekčního zařízení. Do budoucna by ale měla tato data být nahrazena daty z technologie detekce C-ITS, která je podrobně popsána v další podkapitole.

2.4 Metoda detekce pomocí technologie V2X

Jedná se o metodu, která využívá k detekci dopravních parametrů tzv. V2X technologii, jež je součástí C-ITS systému a znamená Vehicle To X. Písmenem X je myšlena proměnná, která má symbolizovat možnost, že vozidlo může komunikovat s různými prvky kolem sebe, ať už se jedná o infrastrukturu, případně ostatní vozidla. V rámci této komunikace jsou definovány typy zpráv pro různé situace. Pro tuto práci jsou relevantní hlavně zprávy typu CAM, které mohou obsahovat informace relevantní pro dopravní proud. Touto metodou detekce je možné detekovat relativně mnoho dopravních parametrů, nicméně penetrace touto technologií je v době psaní této práce nízká, proto kvantita těchto dat není příliš vysoká.

2.4.1 Stručný úvod do C-ITS systémů

C-ITS (Cooperative Intelligent Transport systém) je decentralizovaný systém, který umožňuje dopravním prostředkům komunikovat mezi sebou, případně komunikovat s prvky umístěnými na dopravní infrastruktuře a přeposílat si důležité informace o dopravním proudu, či dopravní situaci. Tato komunikace je zprostředkována ve vyhrazeném pásmu C-V2X a ITS-G5 technologiemi, které jsou povrchově popsány v dalších kapitolách.

Základním kamenem C-ITS systému je hybridní jednotka, která dokáže komunikovat ve vyhrazeném frekvenčním pásmu 5,9 GHz a zároveň je připojena k datové síti pomocí LTE sítě mobilních operátorů. Pásmo 5,9 GHz je používáno pro komunikaci s ostatními jednotkami, zatímco síť LTE je používána ke komunikaci s centrálním serverem. Jednotka je umístěná buď v dopravním prostředku nebo na infrastruktuře komunikace. Použití frekvenčního pásma 5,9 GHz je definováno normou ETSI EN 302 571.

Definice C-ITS jako decentralizovaného systému vychází ze skutečnosti, že hybridní jednotka není povinna odesílat svá data na BO tzn. je schopna fungovat bez připojení k centrálnímu serveru. Centrální server je rozdělen na dvě části. Část, která vyhodnocuje data, odesílá pokyny jednotkám a je hlavním výpočetním prvkem se nazývá Back Office (dále jen BO). Data zpracovává BO, pro příklad to mohou být polohy hybridních jednotek, informace o provozu nebo dopravní události, jsou posléze odesílána na druhou část centrálního serveru, jež se nazývá Frontend Office (dále jen FO). FO má za úkol data přijatá z BO zobrazovat nad mapovými podklady a zároveň je na této části realizováno HMI pro koncového uživatele. Jinými slovy FO zprostředkovává dispečink C-ITS systému, z něhož je možné jednotky a události jimi vytvořené spravovat. Dispečer má také možnost vytvořit nové zprávy, virtuální portály (tzv. IVI) či dojezdové doby.

C-ITS systém je popsán jednotlivými případy použití tzv. Use Case (dále jen UC). Konkrétní UC jsou podrobně popsány v C-ROADS Use Case katalogu.

V tabulce 1 jsou definovány 4 základní komunikační kategorie hybridních jednotek. Tyto kategorie jsou [17]:

Tabulka 1: Základní komunikační kategorie hybridních jednotek

| V2V | V2I | I2V | I2I |
|--------------------|---------------------------|---------------------------|----------------------------------|
| Vehicle to Vehicle | Vehicle to Infrastructure | Infrastructure to Vehicle | Infrastructure to Infrastructure |

2.4.2 V2X technologie

V2X komunikace se omezuje pouze na kategorie, kdy je zdrojem vyslané zprávy hybridní jednotka umístěná ve vozidle, jedná se tedy o typy komunikace znázorněné v tabulce 2 a jednotky typu OBU a RVU.

Tabulka 2: Komunikační kategorie technologie V2X

| V2V | V2I |
|--------------------|---------------------------|
| Vehicle to Vehicle | Vehicle to Infrastructure |

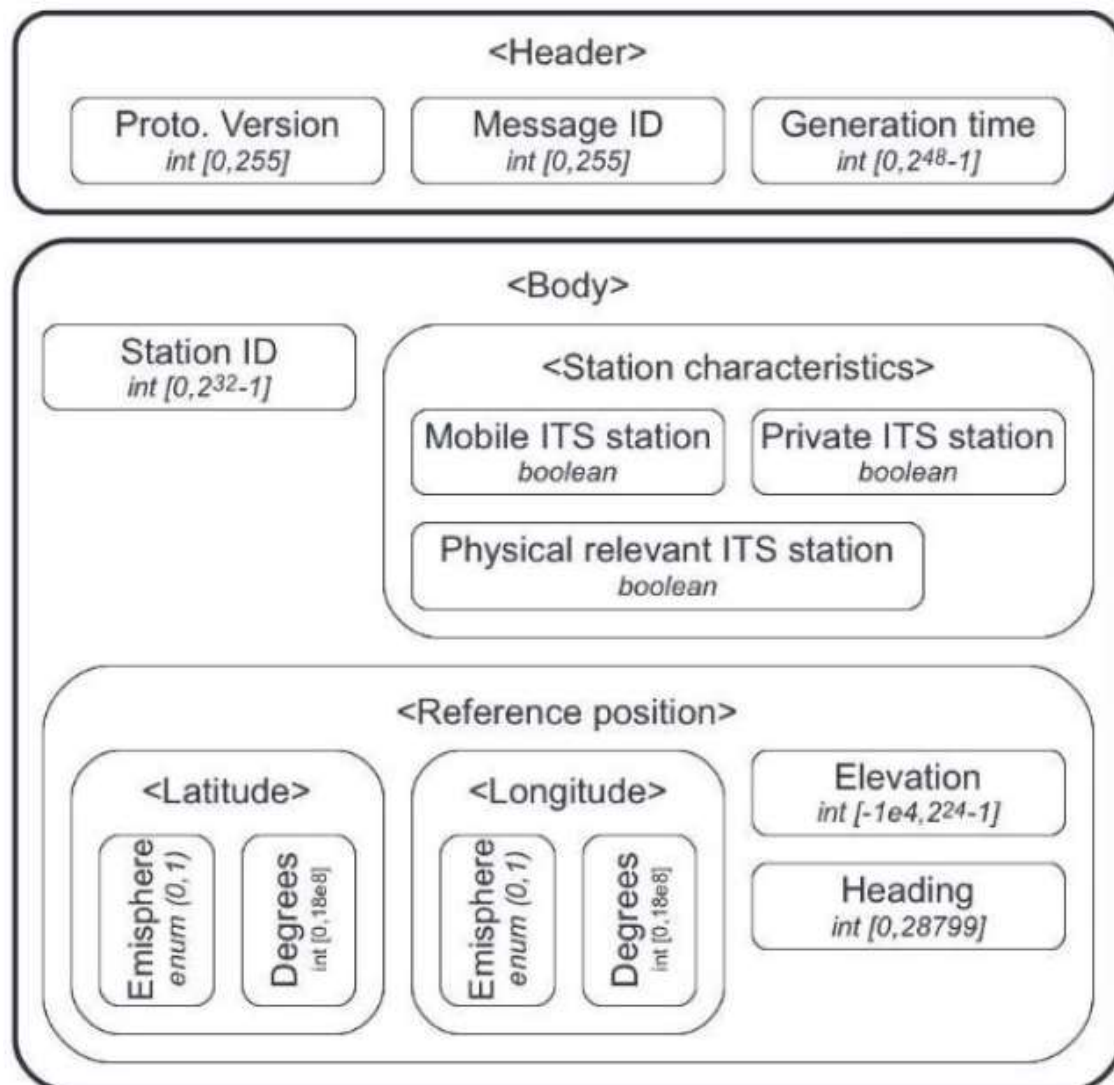
Přičemž pro potřeby této práce je stěžejní kategorie V2I, tedy Vehicle to Infrastructure. Je to z toho důvodu, že při detekci V2X technologií je RSU jednotka považována za detektor. Pokud je RSU jednotkou přijata CAM zpráva od jiné jednotky a nemá atribut StationType nastaven na RSU (což by znamenalo, že CAM zpráva byla odeslána od jiné RSU jednotky), je tato událost považována za detekci jedoucího vozidla.

Detekce dopravních parametrů je ve V2X technologii realizována pomocí RSU jednotek a mobilní datové sítě LTE. Vozidlové jednotky (tedy OBU a RVU) každou sekundu odesílají do éteru několik CAM zpráv. CAM zprávy jsou klíčové při detekci touto technologií. Obsahují totiž důležité informace o projíždějícím vozidle, ze kterých je možné získat dopravní parametry pro určení chování dopravního proudu v daném místě. CAM zprávy odeslané vozidlovými jednotkami jsou zachyceny RSU jednotkami a přeposílány na BO, kde jsou dále zpracovávány. Pokud je OBU/RVU jednotka vybavena SIM kartou a má připojení k LTE síti, jsou tyto CAM zprávy automaticky přeposílány na BO k další analýze.

Je důležité zmínit, že RSU jednotky nesbírají dopravní data nepřetržitě. Pro sběr dopravních dat z RSU je nutné na FO nastavit intervaly sběru dopravních dat (aktuálně lze nastavit dlouhodobý a krátkodobý sběr) a definovat detekční zóny. Po této činnosti začne RSU z přijatých CAM zpráv přeposílat na BO dopravní parametry [17].

2.4.3 Detekce pomocí CAM zpráv

Zprávy CAM jsou generovány a odesílány periodicky až 10x za sekundu dle okolních podmínek. Poskytují základní informace o zařízení, které je vygenerovalo (OBU/RVU/RSU jednotka). Obsahují hlášení o přítomnosti, poloze, teplotě a provozním stavu příslušného zařízení. CAM bývají často označovány jako „Here I am“ zprávy. Struktura zprávy CAM je znázorněna na následujícím obrázku.



Obrázek 21: Struktura CAM zprávy [23]

Jak již bylo zmíněno, zprávy CAM mohou být mimo jiné využívány pro výpočty dojezdových dob, zdržení, průměrných rychlostí apod. v rámci služby Probe vehicle data. Vždy první CAM zpráva, kterou RSU jednotka obdrží od konkrétní OBU jednotky, je zároveň přeposlána do C-ITS back office. Všechny další obdržené CAM zprávy jsou zpracovávány na úrovni RSU jednotky, kde se data agregují dle virtuálních detektorů. Do C-ITS serveru už jsou následně odesílána agregovaná data o provozu v oblasti dosahu příslušné RSU jednotky. Zprávy CAM přijímají také ostatní OBU jednotky v okolí

„vysílající jednotky“. Specifikace zprávy CAM je detailně popsána v předpisu ETSI TS 102 637-2 Specification of Cooperative Awareness Basic Service. Jednotlivé atributy zprávy CAM jsou popsány ve standardu ETSI TS 102 894-2 Applications and facilities layer common data dictionary. [17]

2.4.4 Probe Vehicle Data

Prostřednictvím této služby jsou správci komunikace či operátorovi MHD dodatečně poskytovány dopravní informace. RSU umístěné podél komunikace sbírají data z projíždějících vozidel (vybavených OBU/RVU) a v definovaných časových intervalech zasílají nashromážděná data do BO (a dále do nadřazených řídicích systémů). Každé RSU zařízení umístěné na infrastruktuře lze nakonfigurovat na několik virtuálních detekčních zón, tudíž je možné sledovat jedním RSU více zón najednou (vč. protisměrných). Agregace a zpracování dat nemusí probíhat pouze na RSU, ale také přímo v C-ITS BO v případech přímého propojení OBU a C-ITS BO prostřednictvím mobilních sítí. RSU umístěné na infrastruktuře provádějí krátkodobý sběr dat (pro potřeby řízení dopravy) v řádu desítek sekund a dlouhodobý sběr dat (pro statistické účely bez rozlišení kategorií vozidel) v řádu několika hodin. Mezi sledované veličiny může patřit: kategorie vozidla, intenzita, rychlost, aj. Službou PVD lze sbírat i data, která nejsou běžné dopravní sčítače schopny zaznamenat (např. stav světlometů, mlhových světel apod.), v takovém případě je však nezbytné, aby projíždějící vozidla vybavená C-ITS jednotkami měla tyto jednotky připojeny k vozidlovým systémům.

Nad rámec krátkodobého a dlouhodobého sběru dat jednotka RSU odesílá do C-ITS back office také základní data (ID OBU jednotky, kategorii vozidla, čas, polohu, rychlost a směr) z každého detekovaného vozidla v dosahu, vždy však pouze informace z první CAM zprávy, kterou od daného vozidla obdrží. Díky datům z jednotlivých vozidel je možné stanovit dojezdové doby mezi předdefinovanými profily a následně je řidičům prezentovat formou IVI zpráv nebo prostřednictvím standardních kanálů (např. portálů dopravních informací). Na základě vnitřních algoritmů v BO je dále možné s agregovanými daty pracovat a dále je vyhodnocovat celkově nebo zvlášť v rámci jednotlivých virtuálních zón (průměrná rychlost, intenzita, min./max. rychlost, délka kolon, aj.). Přesnost těchto výpočtů je nicméně svázána se stupněm penetrace vozidel vybavených C-ITS jednotkou. Získaná data mohou být následně využívána v nadřazených řídicích centrech. Na základě dat z CAM zpráv jednotlivých vozidel lze vyhodnotit různé události na komunikaci (např. kolona) a následně generovat zprávy DENM/IVI varující ostatní řidiče před těmito mimořádnostmi.

Parametry vysílány z RSU do BO [23]

- ID RSU jednotky
- Časová stopa konce intervalu sběru dat
- Typ sběru dat: dlouhodobý či krátkodobý
- Nastavení délky intervalu
- Pro každou detekční zónu:
 - ID detekční zóny
 - Definice detekční zóny
 - Pro každou kategorii vozidel:
 - ID kategorie vozidel
 - Průměrná rychlost
 - Min. a max. rychlost
 - Intenzita
 - Další parametry získány z CAN sběrnice (je-li připojena)

Tyto parametry jsou na RSU nastavitelné z BO.

Parametry odesílány z RVU do BO k výpočtu dojezdových dob [23]

- ID RVU jednotky
- Kategorie vozidla
- Čas
- Poloha
- Rychlost
- Směr

2.4.5 Závěr analýzy detekce pomocí V2X technologie

V2X technologie umožňuje detekovat velké množství dat a dalších parametrů, pomocí kterých je možné určit stav dopravního proudu. Technologie má nicméně příkoří v počtu dat kvůli nízké penetraci v provozu. Prognóza ale naznačuje, že by penetrace měla v čase postupně narůstat. Z hlediska algoritmu VLŘD by tato technologie měla v budoucnu nahradit detekci pomocí technologie Bluetooth/Wifi ve fázi, kdy detekce pomocí Bt/Wifi již nebude možná z důvodu anonymizace MAC adres.

Ze zmíněných důvodů nelze tuto technologii aktuálně brát jako plnohodnotnou detekční metodu, je ale vhodné algoritmus VLŘD připravit na vstup i této technologie právě z hlediska prognózy.

2.5 Závěr analýzy datových zdrojů pro algoritmus VLŘD

Primárním zdrojem dopravních dat pro adaptivní algoritmus VLŘD, vzhledem k provedené analýze, byla zvolena metoda detekce indukční smyčkou neboli ASD. Ředitelství silnic a dálnic tuto technologii využívá na většině dálnic a komunikacích 1. třídy. ASD mají také vysokou spolehlivost detekce a jsou schopny i klasifikace projíždějících vozidel. Tyto vlastnosti jsou pro navrhnutý algoritmus VLŘD zásadní, protože rozhodování algoritmus vychází ze znalosti aktuálního stavu dopravního proudu. A z provedené analýzy vychází, že nejlépe je tyto informace schopno poskytnout ASD.

Nedostatkem této technologie je její cena a komplikovanější instalace. Také může dojít k výpadku technologie. V tomto případě je nutné zajistit náhradní zdroj dopravních dat, aby algoritmus mohl být schopen dále operovat a vytvářet virtuální portály. K tomu by mohly posloužit ostatní datové zdroje, jež byly předmětem této analýzy. Detekce pomocí FCD, Bluetooth/Wifi a V2X dávají relativně podobné výstupy. U FCD a Bluetooth/Wifi se jedná hlavně o dojezdové doby, zatímco V2X technologie může nabídnout i intenzitu a rychlost, nicméně její datová základna je aktuálně velmi nízká. Dojezdové doby mohou být cenným zdrojem při zpřesňování stavu dopravního proudu a při výpadku primární technologie mohou posloužit jako náhradní zdroj dat. Časem se ale předpokládá, že detekce pomocí V2X technologie by mohla oba tyto zdroje předčít díky velkému množství parametrů, které je možné touto technologií detekovat. Nicméně je velmi závislá na penetraci vozidlovými jednotkami, která je aktuálně nízká. Výhoda tří výše zmíněných technologií také spočívá v jejich ceně a komplexitě instalace. Co se týče FCD technologie komplexita instalace je téměř nulová, nejvíce práce pro úspěšnou detekci je nutné vykonat na straně serveru, který dopravní data shromažďuje. U Bluetooth/Wifi už může být instalace komplexnější, nicméně tím, že se jedná o neintruzivní způsob detekce, je instalace těchto detektorů oproti instalaci detektorů indukčních smyček značně jednodušší a levnější. V2X technologie využívá k detekci vozidlové jednotky. Komplexita instalace, ani cena nejsou v tomto případě příliš vysoké v kontextu detektoru indukčních smyček.

Výsledkem analýzy tedy je využití indukčních smyček jako primárního zdroje dopravních dat pro adaptivní algoritmus VLŘD. Jako doplňující, či záložní je doporučeno použít technologie detekce pomocí FCD, Bluetooth/Wifi a V2X technologie.

Kapitola 3

Možnosti predikce dopravní situace

Předmětem kapitoly je představení SW nástroje, který byl využit v rámci návrhu adaptivního algoritmu VLŘD k ovládání dopravního modelu a jeho možnosti v oblasti predikce.

Tím, že adaptivní algoritmus je navržen primárně pro dálnice, nejsou jeho nároky na simulační SW příliš vysoké (Není potřebná logika světelně řízených křižovatek, není potřeba simulovat chodce atd.), protože simulovaná situace je silně specifická. Důležité pro výběr vhodného simulačního SW byly tyto aspekty:

- Jednoduchý import silniční sítě do SW, nejlépe z dostupného zdroje jako jsou např. Open Street Maps
- Přístup do knihoven s funkcemi pro ovládání simulačního SW pomocí programovacího prostředí
- Možnost implementace logiky proměnného dopravního značení, které by bylo pomocí adaptivního algoritmu možné ovládat
- Cena

V rámci výběru byly zvažovány tyto podpůrné SW:

- AIMSUN
- MATSIM
- SUMO
- VISSIM

Podpůrné SW AIMSUN a VISSIM jsou obsáhlé nástroje, které mají širokou škálu možností a velmi pravděpodobně by bylo možné adaptivní algoritmus vytvořit přímo v nich a poté lehce ověřit jeho funkčnost. Také tyto nástroje splňují požadavky kladené na začátku této kapitoly, krom jednoho. Tím je cena. Oba tyto nástroje jsou zpoplatněny a společnost AIMSUN například nabízí služby na míru zákazníkovi. Nicméně cena je jeden z hlavních důvodů proč nebyly tyto dvě služby vybrány pro ověření návrhu.

Dalším z důvodů byla nedostupnost zdrojových kódů, čímž se z celé služby stává tzv. „blackbox“, u kterého není přesně vidět, jak požadované výstupy spočítá. U služeb MATSIM a SUMO je tento problém vyřešen tím, že jsou klasifikovány jako Open Source, což znamená, že veškeré zdrojové kódy obou služeb jsou veřejně dostupné na internetu včetně zdokumentovaného postupu, jak služby sestavit a všech knihoven potřebných k ovládání SW pomocí programového řešení [24] [25]. Samozřejmě pocitu blackboxu se

nelze zbavit ani v těchto případech, protože služby jsou složité a jejich komplexní pochopení by mohlo jednoho člověka stát spousty času, nicméně otevřenost vývojového postupu umožňuje aspoň z povrchu nahlédnout pod pokličku celého SW a je možný další rozvoj těchto služeb.

Nakonec byl pro potřeby adaptivního algoritmu VLŘD vybrán simulační nástroj SUMO, který je podrobně popsán v následující podkapitole.

3.1 SUMO

SUMO (Simulation of Urban MObility) je otevřený nástroj pro dopravní simulace vyvíjený centrem DLR v oddělení Institutu pro dopravní systémy v Německu. Jedná se o všestranný nástroj, který najde uplatnění jak v oboru výzkumu, kdy je možné pomocí něho najít nejkratší možné cesty, testování signálních plánů či simulace efektivity hromadné veřejné dopravy [26].

SUMO vznikl roku 2001 a nyní je z něho soubor několika programů. Umožňuje na základě dopravních měření a O/D matic využívat různé routovací algoritmy a simulování křižovatek. Obsahuje také rozhraní TraCI, jež slouží k online simulaci. Toto rozhraní bylo využito právě v rámci ověření návrhu algoritmu.

Díky podpoře vývoje v rámci instituce DLR není u nástroje SUMO pravděpodobné, že by u něj byla ukončena podpora, tak jako tomu bylo u jiných dostupných SW. SUMO bylo vyvíjeno za účelem poskytnout veřejnosti plnohodnotný simulační SW, který by se svými funkcemi vyrovnal komerčním produktům jako jsou AIMSUN a VISSIM které byly zmíněny dříve [26]. Navíc přístupnost programových knihoven pro ovládání simulace nabízejí v tomto ohledu zajímavou možnost, jak si nástroj přizpůsobit ke svým potřebám.

3.1.1 Funkční koncept

Pro úspěšné spuštění simulace v prostředí SUMO je potřeba několik elementů. Toto je seznam nejdůležitějších z nich:

- Data o dopravní síti (např. silniční komunikace, cesty pro pěší)
- Dodatečné informace o dopravní infrastruktuře (např. SSZ)
- Dopravní požadavky

Dohromady tyto elementy vytváří simulační scénář. Protože dopravní simulační modely jsou typicky užívány pro stochastické chování, měl by být takovýto scénář simulován několikrát a na základě všech simulací vytvořen statistický závěr.

Ke kvantitativnímu vyhodnocení poskytuje SUMO široký soubor výstupních souborů, jako jsou:

- Vozidlové trajektorie (pozice a rychlost)
- Dopravní data získaná z modelovaných detektorů

- Agregovaná dopravní data z prvků sítě (hrany, pruhy)
- Agregovaná data napříč celou cestou vozidla, nebo osoby
- Protokoly změn SSZ
- Agregovaná dopravní data pro celou simulaci
- Emise, spotřeba energie a mnoho dalšího

Tyto výstupy mohou být zobrazeny pomocí nástrojů SUMO nebo importovány do jiných aplikací. Pro usnadnění importu do jiných aplikací SUMO obsahuje nástroje pro převod výstupních souborů do jiných formátů pro import pomocí jazyku python nebo prostředí matlab [27].

3.1.2 Data o dopravní síti

Podklady dopravní sítě jsou v prostředí SUMO určeny definičním souborem s příponou „*net.xml*“. Tento soubor obsahuje definici spojovacích bodů (Junctions), definici hran (Edges) a definici jednotlivých pruhů na hranách (Lanes). Na obrázku 22 je vyobrazen příklad definice hrany s pruhy. Na obrázku 23 je vyobrazen příklad definice spojovacího bodu.

```
<edge id="<ID>" from="<FROM_NODE_ID>" to="<TO_NODE_ID>" priority="<PRIORITY>">
  <lane id="<ID>_0" index="0" speed="<SPEED>" length="<LENGTH>" shape="0.00,495.05 248.50,495.05"/>
  <lane id="<ID>_1" index="1" speed="<SPEED>" length="<LENGTH>" shape="0.00,498.35,2.00 248.50,498.35,3.00"/>
</edge>
```

Obrázek 22: Příklad definice hrany s pruhy [28]

Parametr s názvem „id“ je označení hrany, které by mělo být napříč ostatními hranami unikátní. Parametr „from“ má značit hranu, ze které definovaná hrana vychází, Parametr „to“ naopak značí hranu, do které má hrana vést. Parametr s názvem „priority“ uvádí důležitost hrany.

Dále jsou v rámci hrany definované jednotlivé pruhy. Parametr s názvem „id“ je unikátní identifikátor pruhu, v zavedené praxi se skládá z identifikátoru hrany a parametru „index“. „Index“ značí číslo pruhu, musí být v rámci jedné hrany unikátní, jinými slovy hrana nemůže obsahovat dva pruhy se stejnými indexy. Parametr „speed“ značí maximální povolenou rychlost v pruhu. „Length“ značí délku pruhu a „shape“ definuje geometrii pruhu pomocí polyline popisující středovou čáru. Parametr „shape“ nesmí být prázdný a musí obsahovat alespoň dva body [28].

```
<junction id="<ID>" type="<JUNCTION_TYPE>" x="<X-POSITION>" y="<Y-POSITION>"
  inclanes="<INCOMING_LANES>" intlanes="<INTERNAL_LANES>"
  shape="<SHAPE>"
  ... requests ...
</junction>
```

Obrázek 23: Příklad definice spojovacího bodu [28]

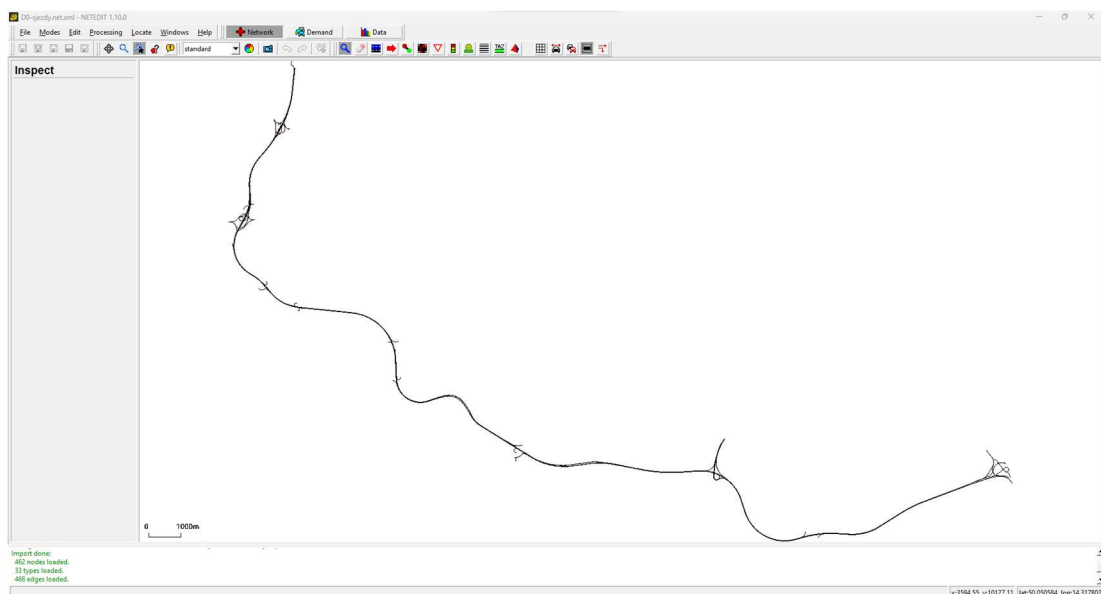
Spojovací bod reprezentuje oblast křížících se dvou různých proudů. Parametr „id“ je unikátní identifikátor spojovacího bodu. „Type“ definuje typ spojovacího bodu, např. zda je prioritní, nebo zda je to slepý bod, který dále nikam nevede. „x“ a „y“ definuje pozici spojovacího bodu v síti simulace pomocí simulačního souřadnicového systému.

„incLanes“ definuje pruhy vstupující do bodu, „intLanes“ definuje pruhy vystupující ze spojovacího bodu. „Shape“ je polygon popisující hranice spojovacího bodu. „Requests“ jsou definice předností [28].

3.1.2.1 Vytvoření definičního souboru *.net.xml*

Vytvoření definičního souboru dopravního podkladu *.net.xml* je možné několika způsoby. První možností je, že uživatel sám soubor *.net.xml* napíše a definuje si jednotlivé hrany a křižovatky. Výše uvedené popisy atributů definičního souboru by měly stačit k tomu, aby uživatel byl schopen si vytvořit tento soubor sám, jen s pomocí textového editoru. Nicméně tento způsob vytvoření není příliš efektivní a může vést k chybám.

Trochu efektivnějším způsobem, jak vytvořit dopravní podklad pro simulaci může být využití SW, který je součástí služby SUMO s názvem NetEdit. Tento editační SW umožňuje pomocí vestaveného GUI vytváření dopravních podkladů přímo v souřadnicovém systému simulátoru. Podobu GUI lze vidět na obrázku níže.



Obrázek 24: GUI editovacího SW definičních dopravních podkladů NetEdit

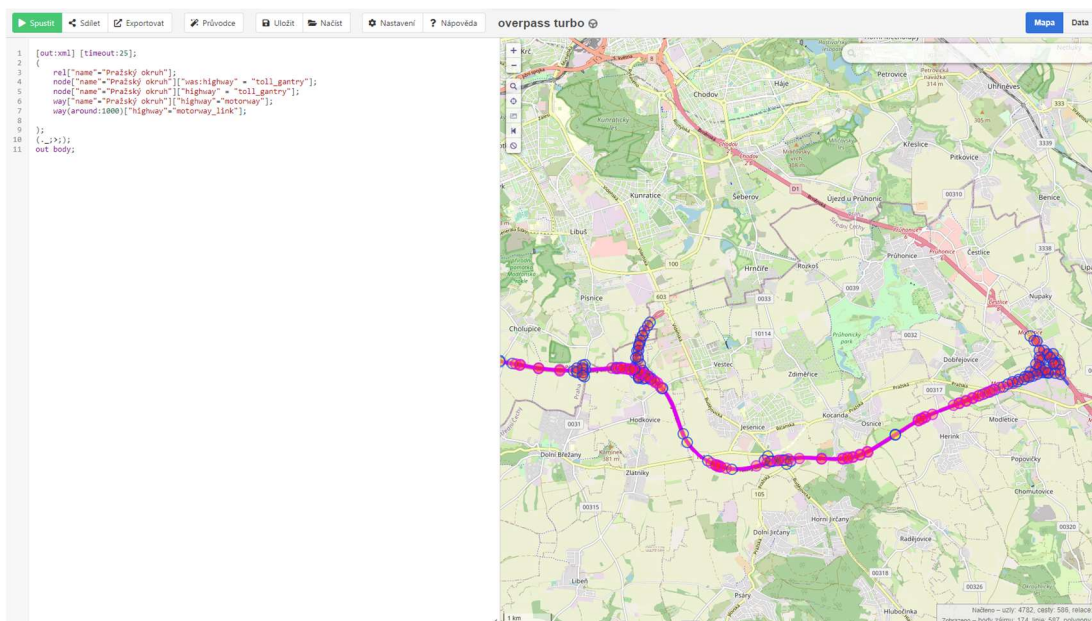
Program NetEdit umožňuje vytváření jednotlivých hran a křižovatek snadným způsobem typu „klikni a táhni“, je tedy možné v něm vytvořit libovolný dopravní podklad jednodušeji než sepisovat definiční prvky ručně. Nicméně pro větší dopravní sítě, které by měly být kopií skutečných dopravních sítí, není toto také optimální způsob. Překreslování skutečných dopravních sítí by v tomto programu mohlo zabrat vysoké množství času a mohlo by být také neefektivní, protože v něm při překreslování mohou vzniknout chyby. Pro vytváření definičních vrstev dopravních podkladů skutečných dopravních sítí je mnohem efektivnější využít funkci importu z mapových podkladů.

3.1.2.2 Import definičních dopravních podkladů z mapové služby *OpenStreetMap*

Jak již bylo řečeno, překreslovat reálné dopravní sítě pomocí grafického nástroje NetEdit nemusí být z hlediska časového zatížení a správnosti efektivní. Simulační nástroj SUMO umožňuje převod definičního souboru *.xml* vyexportovaného ze služby OverPassAPI

OpenStreetMaps do definičního souboru dopravních podkladů služby SUMO *.net.xml*. OverPassAPI OpenStreetMaps je rozhraní nad databází Služby OpenStreetMaps, které díky vstupním parametrům může zobrazit pouze vyhledávané části mapových podkladů a vyexportovat je do preferovaného výstupního formátu.

Na internetu je velké množství služeb, které ulehčují práci s OverPassAPI OpenStreetMaps. Jeden z nejznámějších je nástroj s názvem OverPass-Turbo dostupný na doméně [29].



Obrázek 25: GUI nástroje OverPass-Turbo [29]

Na obrázku 25 je zobrazeno GUI nástroje Turbo. V levé části je okno, do kterého je možné zadat parametry pro vyhledávání v databázi OpenStreetMaps. V pravé části je zvýrazněn výstup zadaného vyhledávání nad mapovými podklady. Ve vyobrazeném příkladu byl zadán na dotaz zobrazení pouze prvků, které:

- Mají v názvu hrany „Pražský okruh“
- Mají spojovací body které mají v názvu „Pražský okruh“ a zároveň obsahují parametr s názvem „was:highway“ s hodnotou „toll_gantry“
- Mají spojovací body které mají v názvu „Pražský okruh“ a zároveň obsahují parametr s názvem „highway“ s hodnotou „toll_gantry“
- Mají hrany s názvem „Pražský okruh“ a zároveň obsahují parametr „highway“ s hodnotou „motorway“
- Na body se do okruhu 1000 metrů napojují hrany, které obsahují parametr „highway“ jehož hodnota je „motorway_link“

Kombinací těchto podmínek vrátí OverPass API hrany a jejich spojnice pouze komunikace D0 s jejich nájezdnými hranami do okruhu 1000 m. Parametr na začátku dotazu *[out:xml]* definuje výstupní soubor, do kterého se má výstup vypsát.

Nástroj je unikátní právě v tom, že uživateli přehledně zobrazí, které prvky si v dotazu na API vybral. Část výstupního souboru XML, který byl vrácen na dotaz je vyobrazen na obrázku 26.

```
<?xml version="1.0" encoding="UTF-8"?>
<osm version="0.6" generator="Overpass API 0.7.57.1 74a55df1">
<note>The data included in this document is from www.openstreetmap.org. The data is made available under ODbL.</note>
<meta osm_base="2022-03-18T08:53:12Z"/>

<node id="247778" lat="50.1203020" lon="14.5894398">
  <tag k="maxspeed" v="40"/>
  <tag k="traffic_sign" v="maxspeed"/>
</node>
<node id="250663" lat="50.0922677" lon="14.2958684">
  <tag k="traffic_sign" v="CZ:I71a"/>
  <tag k="traffic_sign:direction" v="forward"/>
</node>
<node id="250666" lat="50.0850020" lon="14.2948014"/>
<node id="250717" lat="50.0676209" lon="14.2816074"/>
<node id="252057" lat="50.0592323" lon="14.2765873"/>
<node id="252059" lat="50.0494314" lon="14.2783745"/>
<node id="252060" lat="50.0415053" lon="14.2702873"/>
<node id="252061" lat="50.0381057" lon="14.2718195"/>
<node id="252062" lat="50.0331064" lon="14.2805081"/>
<node id="252063" lat="50.0272698" lon="14.2896910"/>
<node id="252064" lat="50.0259842" lon="14.2932811"/>
<node id="252066" lat="50.0228399" lon="14.3276786"/>
<node id="252067" lat="50.0224620" lon="14.3290681"/>
<node id="703036" lat="50.0211517" lon="14.3325305"/>
<node id="703039" lat="50.0190831" lon="14.3360428"/>
<node id="703041" lat="50.0181918" lon="14.3371492"/>
<node id="703042" lat="50.0160909" lon="14.3393600"/>
<node id="703077" lat="50.0251587" lon="14.2982782">
  <tag k="highway" v="motorway_junction"/>
  <tag k="name" v="Ořech"/>
  <tag k="name:de" v="Worschech"/>
  <tag k="ref" v="19"/>
</node>
<node id="778500" lat="50.0571199" lon="14.2767913">
  <tag k="highway" v="motorway_junction"/>
  <tag k="name" v="Třebonice"/>
  <tag k="name:de" v="Trebonitz"/>
  <tag k="ref" v="23 B"/>
</node>
```

Obrázek 26: Ukázka výstupního souboru *.xml* ze služby OverPass API

V tuto chvíli je potřeba převést vygenerovaný *.xml* soubor do podoby *.net.xml* který slouží jako dopravní podklad pro simulaci.

Tento krok je možné provést pomocí funkce *netconvert*, která slouží v prostředí SUMO k převodu *.xml* souborů do formátu *.net.xml*. Je třeba zmínit, že simulační nástroj SUMO není primárně stavěn jako nástroje VISSIM a AIMSUN, kde většina interakce s nástrojem je primárně zprostředkována pomocí GUI. Nástroj SUMO je spíše orientován jako konzolová aplikace, která je využívána pomocí příkazů s argumenty. Proto i příkaz *netconvert* je nutné provést z konzolového prostředí, kde se pomocí argumentů specifikuje, jaký soubor má být převeden, jaký název má mít výstupní soubor, případně další nastavení. Příklad takového příkazu s argumenty je možné vidět na obrázku 27.

```
netconvert --osm-files berlin.osm.xml -o berlin.net.xml
```

Obrázek 27: Příklad spuštění funkce *netconvert* pro konverzi *.xml* souboru do souboru *.net.xml* [28]

Obrázek 27 vyobrazuje převod souboru *berlin.osm.xml*, který je vyexportován ze služby OpenStreetMaps do souboru *berlin.net.xml*, který je již simulačním nástrojem SUMO čitelný a lze nad ním provádět simulační operace.

3.1.3 Dodatečné informace o dopravní infrastruktuře

Další možností, jak ovlivnit chod simulace, případně její výstupy jsou tzv. dodatečné soubory (Additional files), které přidávají různé prvky do simulace jako jsou např. detektory, proměnné dopravní značení, stanice městské hromadné dopravy atd.

Definice těchto prvků je jako většina definičních souborů v prostředí sumo realizována souborem ve formátu *.xml* se specifickou strukturou. Příklad této struktury je uveden na obrázku 28.

```
<additional>
  <inductionLoop id="myLoop1" lane="foo_0" pos="42" freq="900" file="out.xml"/>
  <inductionLoop id="myLoop2" lane="foo_2" pos="42" freq="900" file="out.xml"/>

  <busStop id="station1" lane="foo_0" startPos="5" endPos="20"/>

  <vType id="bus" maxSpeed="20" length="12"/>
</additional>
```

Obrázek 28: Příklad definičního souboru dodatečných informací [28]

Z příkladu je zjevné, že struktura se podobá struktuře, která byla vyobrazena v minulé kapitole. Opět má uživatel možnost vytvořit dodatečné prvky v simulaci sám pomocí textového editoru. Nicméně při větším množství dodatečných prvků může tento postup ztrácet na efektivitě. Existuje tedy možnost vytvořit dodatečné prvky pomocí grafického nástroje NetEdit, jenž byl zmíněn v minulé podkapitole. Tento nástroj poskytuje vytváření dodatečných prvků pomocí vkládání bodů přímo v GUI aplikace. Po vložení prvků NetEdit vygeneruje soubor ve formátu *.xml*, který obsahuje definici zadaných prvků. Při spuštění simulace se poté v argumentech uvede cesta k vygenerovanému souboru, aby simulace probíhala i nad těmito prvky.

Tím, že dodatečné prvky jsou definovány v textovém souboru ve formátu *xml* vzniká možnost vygenerovat tento soubor pomocí programovací metody. Tato metoda je výhodná, pokud se dodatečných prvků nachází na síti mnoho a přestává být efektivní vytvářet dodatečné prvky pomocí nástroje NetEdit. Tato metoda byla použita při generování detektorů a proměnných značení v ověřovací části této práce.

3.1.4 Dopravní požadavky

Pokud by uživatel měl vygenerovány podklady dopravní sítě s dodatečnými prvky a spustil by nad těmito podklady simulaci v prostředí SUMO, zjistil by, že mu po jeho síti nejezdí žádná vozidla. To je způsobeno tím, že uživatel musí definovat poslední důležitý

soubor, kterým definuje vozidla, jež do simulace budou vstupovat, jejich množství a jejich trasy. Soubor těchto definic se nazývá dopravní požadavky.

V rámci této podkapitoly je vhodné zavést následující pojmy:

TRIP je pohyb vozidla z jednoho místa do druhého definovaný počáteční hranou (edge), cílovou hranou a časem, kdy má vozidlo vyrazit.

ROUTE je rozšířená verze typu *TRIP* obsahující nejen počáteční a koncovou hranu, nýbrž i všechny hrany, přes které vozidlo projede. Nástroj SUMO vyžaduje *ROUTES* jako vstup pro pohyb vozidel. Existuje několik možností, jak v nástroji SUMO *ROUTES* vygenerovat.

- Užití definic *TRIP*
Jak již bylo popsáno výše, každý *TRIP* sestává z alespoň počáteční, koncové hrany a z času, kdy má vozidlo vyrazit. To je užitečné, pokud uživatel chce ručně vytvořit dopravní požadavky nebo když chce uživatel importovat vlastní data pomocí svého vytvořeného skriptu. Pro převod definic *TRIP* do *ROUTE* je k dispozici nástroj *duarouter*.
- Užití definic intenzit
Tento postup je víceméně stejný jako u použití definic pomocí *TRIP*, nicméně je možné sjednotit více vozidel, které vstoupí do simulace pomocí počáteční a koncové hrany.
- Užití randomizace
Toto je rychlá cesta k získání alespoň nějakých dopravních požadavků, pokud uživatel nemá přístup k dopravním datům. Pomocí této metody jsou dopravní požadavky vygenerovány náhodně, nicméně výsledky simulace jsou velmi nerealistické
- Užití O/D matic
O/D matice (Origin-Destination-Matrices) neboli matice intenzit přepravních vztahů jsou často dostupné od dopravních autorit. Je nutné je převést pomocí nástroje *od2trips* do *TRIPS*
- Užití definic intenzit a odbočovacích poměrů
Uživatel má také možnost vynechat definici koncové hrany pro intenzity a místo ní použít odbočovací poměry na křižovatkách. Převod je možný pomocí nástroje *jtrrouter*.
- Užití dat z detektorů
Nástroj SUMO umožňuje generování *TRIPS* na základě dat z detektorů indukčních smyček. K převodu slouží funkce *dfrouter*.
- Ruční definice
I v definici dopravních požadavků je možné soubor *.xml* vytvořit ručně pomocí textového editoru.
- Užití statistik o populaci
Nástroj *activitygen* umožňuje převod statistik o populaci do dopravních požadavků.

- Užití dat z jiných zdrojů

Z výše uvedeného výčtu je patrné, že nástroj SUMO disponuje velkým množstvím nástrojů pro vytvoření definic dopravních požadavků. Díky tomu může uživatel, pokud má k dispozici dopravní data, relativně jednoduše vytvořit dopravní požadavky odpovídající jeho naměřeným hodnotám. Tato skutečnost činí z nástroje SUMO flexibilní nástroj pro simulaci dopravy [27].

Na obrázcích 29 a 30 jsou vyobrazeny vzorové definice dopravních požadavků.

```
<routes>
  <vType id="type1" accel="0.8" decel="4.5" sigma="0.5" length="5" maxSpeed="70"/>

  <route id="route0" color="1,1,0" edges="beg middle end rend"/>

  <vehicle id="0" type="type1" route="route0" depart="0" color="1,0,0"/>
  <vehicle id="1" type="type1" route="route0" depart="0" color="0,1,0"/>

</routes>
```

Obrázek 29: Vzorová definice vozidel a jejich trasy [28]

V rámci objektu *ROUTE* je nutné definovat parametry „id“ a „edges“. „id“ je stejně jako v předchozích definicích unikátní identifikátor. „edges“ je seznam hran, přes které definovaná cesta vede. Objekt „vType“ je definice typu vozidla. Ta se skládá z parametrů „id“ značící unikátní identifikátor typu vozidla, „accel“ značící schopnost zrychlení typu vozidla vyjádřenou v m/s^2 , „decel“ značící brzdící schopnost typu vozidla vyjádřenou v m/s^2 , „sigma“ značící parametr *car-following* modelu, „length“ vyjadřující délku vozidla v metrech a „maxSpeed“ značící maximální rychlost, kterou se vozidlo může pohybovat v m/s. Nastavitelných parametrů pro ty vozidla je mnohem více a je možné je dohledat v dokumentaci SUMO [28]. Dále už jsou v objektu *ROUTES* definována jednotlivá vozidla, kterým je díky parametru „type“ přidělen definovaný typ vozidel a parametru „route“ jež přiřadí vozidlo na definovanou trasu.

```
<flow id="type1" color="1,1,0" begin="0" end="7200" period="900" type="BUS">
  <route edges="beg middle end rend"/>
  <stop busStop="station1" duration="30"/>
</flow>

<route id="routel" edges="beg middle end rend"/>
<flow id="type2" color="1,1,0" begin="0" end="7200" period="900" type="BUS" route="routel">
  <stop busStop="station1" duration="30"/>
</flow>

<flow id="type3" color="1,1,0" begin="0" end="7200" period="900" type="BUS" from="beg" to="end">
  <stop busStop="station1" duration="30"/>
</flow>
```

Obrázek 30: Vzorová definice intenzit a jejich tras [28]

Dopravní požadavky je možné také do simulace definovat pomocí tzv. „flow“ znamenající v překladu „intenzita“. Na tento typ dopravního požadavku lze nahlížet jako na periodický proces vkládání vozidel do simulace, který je ohraničen parametry „begin“, „end“ a „period“. „begin“ značí časový krok v simulaci ve kterém se mají začít vkládat vozidla. „end“ značí časový krok v simulaci, kdy má být vkládání vozidel ukončeno. „period“ značí po jaké periodě se mají vozidla vkládat. Dále je možné pomocí parametru

„type“ definovat typ vozidel, jenž bude v rámci vložení vkládán a také pomocí parametru „route“ definovat cestu po které se mají vložená vozidla pohybovat. Další parametry jsou popsány v dokumentaci SUMO [28].

3.1.5 Spuštění simulace

Po definování všech potřebných souborů je možné přistoupit ke spuštění simulace nad dodanými soubory. Před spuštěním simulace je nutné definovat jakou dopravní síť, jaké dodatečné a dopravní požadavky má nástroj k simulaci použít, případně v jakém časovém intervalu má simulace probíhat. Tyto proměnné jsou definovány v konfiguračním souboru nástroje SUMO s koncovkou *.sumocfg*. Příklad takového souboru je možné vidět na obrázku 31.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <!-- generated on 2022-02-04 15:06:42 by Eclipse SUMO GUI Version 1.10.0
4 -->
5
6 <configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://sumo.dlr.de/xsd/sumoConfiguration.xsd">
7
8   <input>
9     <net-file value="\D0-sjezdy.net.xml"/>
10    <route-files value="\System\assets\Simulation\Input.rou.xml"/>
11    <additional-files value="\System\assets\detectors.xml"/>
12  </input>
13  <time>
14    <begin value="0"/>
15    <end value="1200"/>
16    <step-length value="1"/>
17  </time>
18 </configuration>
```

Obrázek 31: Vzor konfiguračního souboru nástroje SUMO

Je patrné, že i když má konfigurační soubor jinou koncovku, strukturou se stále jedná o formát *xml*.

V sekci *input* jsou uvedeny všechny soubory nezbytné pro provedení simulace, které byly podrobně popsány v předchozích podkapitolách. *Net-file* je hodnota odkazující se na cestu k souboru obsahující definici dopravní sítě nad kterou bude simulace prováděna. *Route-files* je hodnotou odkazující se na cestu k souboru definující dopravní požadavky. *Additional-files* je hodnotou odkazující se na cestu k souboru definující dodatečné informace. Tato jediná hodnota je speciální v tom směru, že souborů obsahující definici dodatečných prvků může být více. Pokud tomu tak je, jsou jednotlivé odkazy v této hodnotě odděleny čárkou.

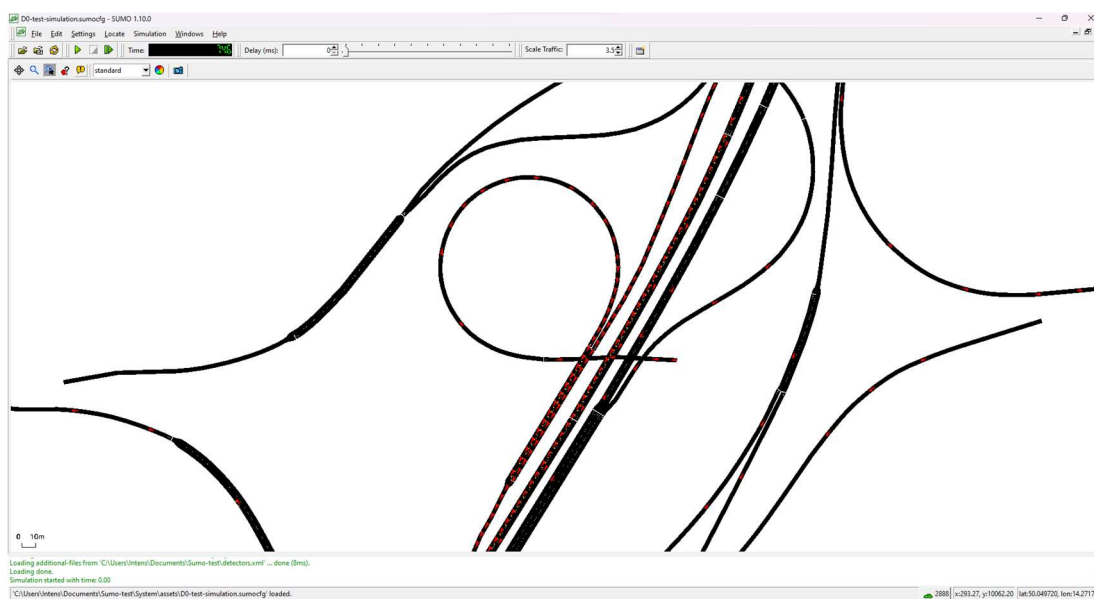
Sekce *time* definuje časový interval simulace, tedy počátek simulace a její konec. Jednotky času jsou sekundy. Je také možné definovat délku jednoho kroku simulace. Tato možnost může být vhodná při simulaci dlouhých časových intervalů, kdy krok velikosti jedné sekundy může být již zbytečně malý, nicméně přesnost simulace se může snížit. Je také nutno zmínit, že čas je zde reprezentován absolutní formou.

Simulaci je možné spustit v několika režimech. Tyto režimy jsou:

- Pomocí SUMO-gui
- V příkazovém řádku
- Pomocí knihovny TraCI

3.1.5.1 Spuštění simulace pomocí SUMO-gui

Uživatelsky nejpřívětivějším je spuštění simulace v programu SUMO-gui. Tento nástroj slouží jako grafické rozhraní a přímo zobrazuje na dopravní síti simulovaná vozidla. Navíc si k tomu uživatel může zobrazit další výstupy v podobě grafů atd. Příklad takového zobrazení je možné vidět na obrázku 32.

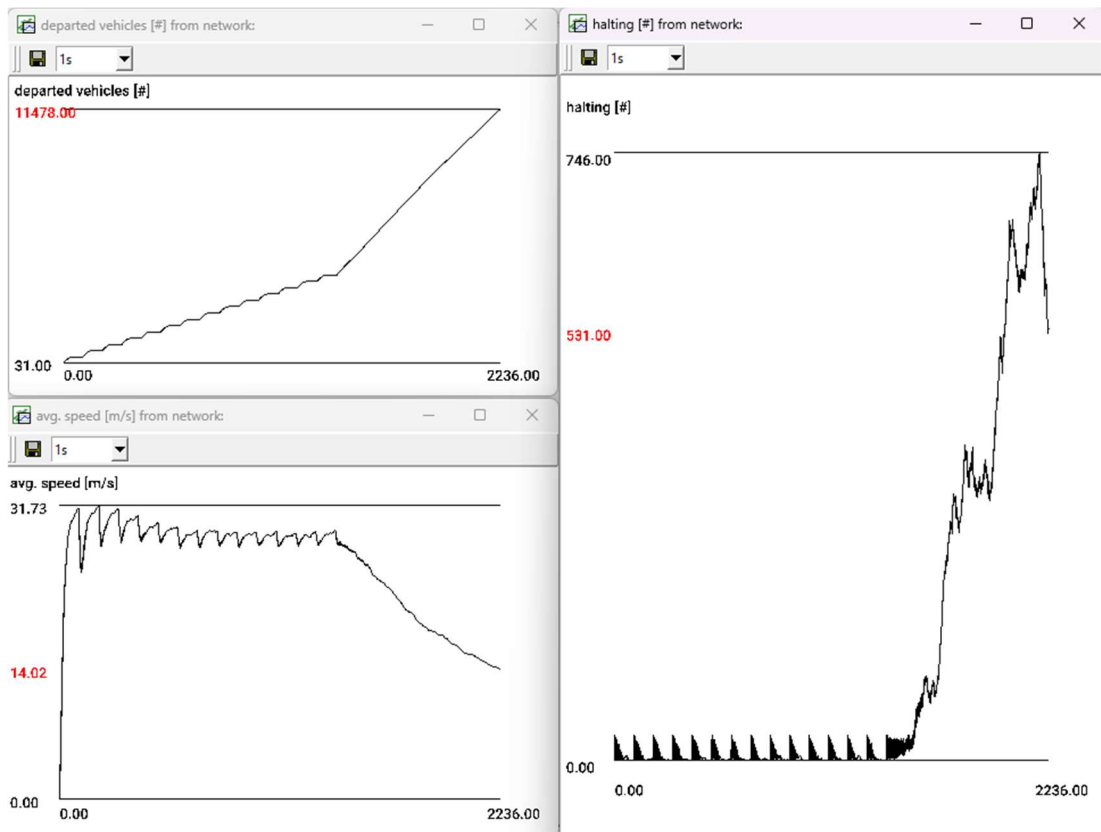


Obrázek 32: Příklad vyobrazení průběhu simulace pomocí SUMO-gui

Na obrázku 32 je vyobrazena situace z úseku D0, kde na jednom z nájezdů vzniká kongesce. Vozidla jsou zde značena trojúhelníky červené barvy. Toto zobrazení je upravitelné v definici dopravních požadavků, kde je možné definovat jak tvar, tak barvu.

SUMO-gui umožňuje také zobrazení tzv. realtime výstupů, které jsou realizovány formou grafů. Uživatel si tak při simulaci může zobrazit některé parametry simulace a přímo v jejím chodu sledovat jejich vývoj. Příklad takového výstupu je zobrazen na obrázku 32, kde byly zobrazeny parametry průměrná rychlost (avg. Speed), počet vypravených vozidel (departed vehicles) a počet zastavení (# halt). V uvedeném příkladu je možné vidět, jak v průběhu simulace byly dopravní požadavky chvíli po půlce měřené periody 8x znásobeny. To vyústilo v razantní zvýšení počtu vypravených vozidel, zastavení a snížení průměrné rychlosti.

Tyto výstupy je možné posléze uložit do formy grafických souborů, jako jsou soubory s příponou *.png*, atd.



Obrázek 33: Možný výstup nástroje SUMO-gui

3.1.5.2 Spuštění simulace pomocí příkazového prostředí

Uživatelsky méně přívětivějším způsobem spuštění simulace je možnost spuštění v příkazovém prostředí. Simulace je spuštěna pomocí příkazu s argumenty, které odkazují buď na cesty k jednotlivým potřebným definičním souborům nebo odkazující pouze na konfigurační soubor simulace *.sumocfg*. V rámci argumentů je možné definovat výstupní veličiny a cesty, kam se mají definované výstupy uložit.

Tato metoda je vhodná pro uživatele, který nemá velký zájem ve sledování průběhu simulace a zajímá jej pouze výsledek.

Příklad spuštění simulace pomocí příkazového řádku je vyobrazen na obrázku 34. Průběh simulace je naznačen na obrázku 35.

```
sumo.exe -c test.sumocfg
```

Obrázek 34: Vzorový příkaz pro spuštění simulace pomocí příkazového prostředí [28]


```

Loading configuration ... done.
Loading net-file from 'C:\Users\Intens\Documents\Sumo-test\D0-sjezdy.net.xml' ... done (84ms).
Loading additional-files from 'C:\Users\Intens\Documents\Sumo-test\detectors.xml' ... done (2ms).
Loading done.
Simulation version 1.10.0 started with time: 0.00
Step #1000.00 (16ms ~ = 62.50*RT, ~87250.00UPS, vehicles TOT 2850 ACT 1396 BUF 0)
Simulation ended at time: 1000.00
Reason: The final simulation step has been reached.
Performance:
Duration: 13.02s
Real time factor: 76.8344
UPS: 75879.062620
Vehicles:
Inserted: 2850
Running: 1396
Waiting: 0
Statistics (avg):
RouteLength: 8948.32
Speed: 29.46
Duration: 302.71
WaitingTime: 0.17
TimeLoss: 26.44
DepartDelay: 14.64

```

Obrázek 35: Konzolový výstup aplikace

3.1.5.3 Spuštění simulace pomocí knihovny TraCI a libsumo

Tato metoda spuštění je vhodná pro ovládání simulace za běhu a pro vyčítání různých hodnot. Ovládání simulace za běhu je možná pomocí funkcí dostupných v rámci knihovny TraCI. Tento přístup je vhodný pro uživatele, který chce simulaci v jejím průběhu nějakým způsobem měnit. Z hlediska návrhu algoritmu pro VLŘD se jeví jako nejlepší způsob interakce algoritmu s modelem právě tento přístup. Implementace algoritmu VLŘD je totiž plánována jako programová a pokud bude program ovládat simulaci, potřebuje k tomu patřičné funkce. Tyto funkce jsou právě obsaženy v knihovně TraCI.

Samotné spuštění simulace a její krokování je ovšem realizováno pomocí funkcí *start* a *step* obsažených v knihovně libsumo. Libsumo je knihovna obsahující funkce samotného simulačního nástroje. Zjednodušeně se dá říci, že obsahuje všechny funkce, které jsou dostupné v nástrojích SUMO-gui a NetEdit. Je tedy možné pomocí této knihovny kontrolovat běh simulace, ale zároveň je možné s její pomocí vytvářet jednotlivé prvky simulace jako jsou například hrany, detektory, atd.

Pro adaptivní algoritmus VLŘD je tedy podstatná kombinace těchto dvou knihoven, kdy libsumo sloužící jako nástroj pro kontrolu běhu simulace, hledání nejkratších cest, atd. a TraCI sloužící k vyčítání důležitých dopravních parametrů a ovlivňování simulace za běhu, dávají dohromady kompletní nástroj pro simulační modul algoritmu VLŘD.

Na obrázku 36 je vyobrazen pseudokód, který spustí simulaci v prostředí Python pomocí metod dostupných v knihovně libsumo. Na obrázku 37 je vyobrazen pseudokód vyčítání polohy vozidla v každém kroku simulace pomocí metody *subscribe*, jež je součástí knihovny TraCI.

```
import libsumo
libsumo.start(["sumo", "-c", "test.sumocfg"])
libsumo.simulationStep()
```

Obrázek 36: Pseudokód spuštění simulace a jejího posunu o jeden krok pomocí knihovny libsumo

```
while traci.simulation.getMinExpectedNumber() > 0:
    for veh_id in traci.simulation.getDepartedIDList():
        traci.vehicle.subscribe(veh_id, [traci.constants.VAR_POSITION])
    positions = traci.vehicle.getAllSubscriptionResults()
    traci.simulationStep()
```

Obrázek 37: Pseudokód vyčítání polohy vozidla pomocí metody *subscribe* knihovny TraCI

3.2 Možnosti predikce dopravní situace pomocí nástroje SUMO

V předchozích podkapitolách byly shrnuty kroky a potřebné soubory k úspěšné inicializaci prostředí SUMO do funkčního stavu. Tato podkapitola se zaměří na možnosti predikce dopravní situace pomocí inicializovaného prostředí. Dopravní situací je v tomto případě myšlen stav dopravního proudu v čase budoucím od spuštění simulace.

V kapitole 3.1.2.2 byla detailně popsána možnost importu dopravní sítě pro simulaci z mapových podkladů volně dostupné služby OpenStreetMaps. OpenStreetMaps (OSM) je, jak již bylo zmíněno, volně dostupná, bezplatná služba poskytující mapové podklady spravované komunitou uživatelů. Uživatelé se zároveň starají o to, aby poskytované mapové podklady obsahovaly aktuální informace [30]. Tyto skutečnosti naznačují, že se jedná o spolehlivý zdroj mapových podkladů, který by mohl s relativně vysokou přesností reflektovat skutečný stav. Jelikož je pro predikci dopravní situace pomocí dopravního modelu důležité mít co nejpřesnější kopii dopravní sítě, pro kterou má být predikce vytvořena, je tedy možnost importu dopravní sítě ze služby OSM do nástroje SUMO velmi důležitá funkce. Díky této funkci je možné relativně snadně a rychle získat digitální kopii oblasti, nad kterou je cílena simulace dopravní situace.

Dalším důležitým krokem pro predikci dopravní situace, je dosažení věrné projekce reálného prostředí. Po dosažení tohoto stavu je možné, díky běhu ve virtuálním prostředí, posunout čas rychleji kupředu. Tím je možné si nasimulovat, jak by mohla simulace vypadat v budoucnu, za aktuálně naměřených podmínek. Aby bylo možné do simulace dostat stav aktuálně naměřených podmínek, je důležité, aby k tomu zvolený simulační nástroj měl adekvátní funkce, pomocí kterých by byl tento cíl dosažitelný nebo alespoň nástroje, pomocí kterých by takové funkce bylo možné naimplementovat. Nástroj SUMO takové funkce i nástroje obsahuje. Jak bylo zmíněno v kapitole 3.1.1, nástroj SUMO potřebuje ke svému běhu 3 definiční soubory. Soubor s definicí silniční sítě, definice dodatečných prvků a dopravní požadavky.

Aby vůbec bylo možné nějakým způsobem predikovat dopravní situaci v definovaném místě, je nutné na tomto místě provést sadu pozorování. Bez těchto pozorování by nebylo možné určit, jak dopravní situace v definovaném místě vypadá. Pravidelným pozorováním je možné určit, jak se doprava v definovaném místě chová a zda se v tomto chování neobjevují jisté opakující se vzorce. Vzorce chování dopravního proudu se již v minulosti mnoho vědců zabývalo a definovali je. Takto definované jsou dnes obsaženy v různých modelovacích nástrojích, přičemž jedním z nich je právě nástroj sledovaný v této práci, SUMO [27] [31]. Tím, že vzorce chování jsou v modelech obsaženy, je už “jen“ potřebné vložit do těchto vzorců hodnoty stavů dopravy z různých bodů. Toho je možné dosáhnout strategickým umístěním detektorů do míst, kde se hodnoty dopravního proudu z nějakého důvodu mění. Například před a za nájezdy na dálniční komunikace nebo v blízkosti dopravních křižovatek. Pokud jsou posléze tyto naměřené hodnoty vloženy do předdefinovaných vzorců je výsledkem kopie naměřené dopravní situace a jelikož je díky vzorcům známo jak se bude dopravní tok chovat dále, nabízí se i určitá možnost predikce. Kvůli stochastické povaze dopravy, nebude nikdy kopie, ani predikce dokonale přesná. Nicméně provedením více pokusů je možné statisticky usoudit jakým směrem se bude doprava dále odvíjet.

Samotné vzorce ale pro přesnou simulaci reálné dopravní situace nestačí. Na jednotlivé koeficienty vzorců modelů lze nahlížet jako na nastavovací parametry modelu. Tyto parametry jsou pro každou simulovanou oblast jiné a jejich definování pro určitou oblast nemusí být jednoduchá disciplína. Definice parametrů by ale měla mít základ na reálně naměřených datech nebo na větším počtu sledování cílené oblasti. Nicméně provedením více pokusů je možné statisticky usoudit jakým směrem se bude doprava dále odvíjet.

Jak bylo zmíněno, k vytvoření vhodné kopie jsou tedy potřebné aktuální hodnoty dopravního proudu a definované koeficienty modelů. Hodnoty dopravního proudu je možné aktuálně změřit pomocí různých detektorů, některé z nich byly zmíněny v kapitole 2. Nástroj SUMO umožňuje tyto naměřené hodnoty převést do dopravních požadavků, kterými následně “naplňuje“ definovanou dopravní síť. Dodatečnými informacemi je možné tyto detektory do dopravní sítě zavést a sledovat jejich hodnoty, tedy je možné touto cestou kontrolovat, že dopravní proud v simulaci dosahuje obdobných hodnot na virtuálních detektorech jako na detektorech skutečných. Dalším z faktorů, který může ovlivnit aktuální dopravní situaci a její vývoj do budoucna je počasí. I tento faktor je v aktuálních podmínkách měřen a jeho hodnoty je možné pomocí parametrů jednotlivých hran převést do modelového prostředí. Nicméně existují i faktory, u nichž předpovídat jejich vznik je komplexní problém a které svou podstatou dopravní proud velmi ovlivní. Mezi tyto faktory patří hlavně dopravní nehody. Vznik dopravní nehody je silně stochastický jev a jeho predikce může být velmi obtížná. Nicméně pravděpodobnost vzniku se vlivem jiných dopravních parametrů může snižovat a zvyšovat, což dává dopravním modelům jistou možnost se o predikci alespoň pokusit. I tento faktor je v modelech nástroje SUMO implementován a je možné sledovat, zda se v rámci simulace tento jev někde nevyskytl [27].

Dle výše uvedeného existují možnosti predikce dopravní situace pomocí nástroje SUMO a lze i tvrdit, že nástroj SUMO byl k těmto predikcím vytvořen.

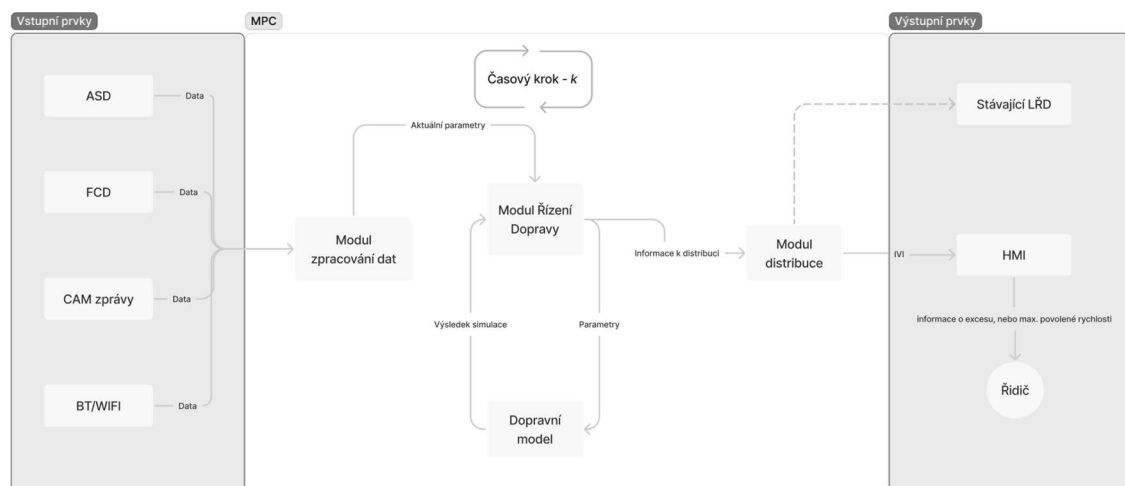
Kapitola 4

Návrh adaptivního algoritmu pro systém VLŘD

Tato kapitola obsahuje vlastní návrh adaptivního algoritmu pro systém VLŘD. V návrhu jsou zahrnuty poznatky z předešlých analýz. Algoritmus vychází z návrhu algoritmů, jež jsou založeny na teorii řízení (control-theory). Součástí algoritmu tedy je dopravní model, který umožňuje predikci dopravní situace na řízené komunikaci, jakožto řízení po celé její délce. Jelikož se jedná o návrh pro systém VLŘD, nejsou distribuované zprávy závislé na polohách jednotlivých řezů, na kterých by se výstupy algoritmu zobrazovaly. To dává oproti LŘD značnou výhodu, protože systém může distribuovat své výstupy do libovolného místa na komunikaci. Samozřejmě, tato schopnost je limitována penetrací vozidlových jednotek, které dokážou výstup VLŘD zachytit a zobrazit jej uživateli. Nicméně hlavní využití systému nastane teoreticky v době nástupu autonomních vozidel, kterým by bylo možné tímto systémem předávat informace o rychlostních limitech. V tomto stavu by se naskytly nové možnosti v oblasti řízení dopravního proudu, kdy by bylo zajištěno dodržování rychlostních omezení vygenerovaných algoritmem VLŘD.

4.1 Základní schéma navrženého algoritmu

V rámci návrhu bylo vytvořeno základní schéma navrženého algoritmu, které by nastínilo koncept algoritmu, jeho jednotlivé prvky a jejich vazby s naznačením typu dat, jež jsou vazbami předávány. Základní schéma navrženého algoritmu je vyobrazeno na obrázku 38. Obrázek může být vzhledem k nemožnosti změny fontů některých vazeb nečitelný, jednotlivé části jsou proto vyobrazeny ve větším měřítku v dalších kapitolách.



Obrázek 38: Základní schéma navrženého algoritmu pro systém VLŘD

Schéma bylo dle identit prvků rozděleno na 3 části. Část s názvem “Vstupní prvky“ obsahuje souhrn prvků, kterými jsou prvky vstupující do algoritmu. Část s názvem “MPC“, je řídicí část algoritmu obsahující potřebné prvky k takovému řízení. Její hlavní úlohou je zpracování a vyhodnocení dat vstupních prvků a následné generování výstupů. “Výstupní prvky“ je část obsahující prvky systému, které nějakým způsobem pracují s výstupy algoritmu.

Na schématu je naznačen časový krok – k . Toto značení symbolizuje, že výměny parametrů mezi jednotlivými prvky jsou prováděny v každém časovém kroku k , který je libovolně definován.

Centrálním prvkem celého algoritmu, potažmo i systému VLŘD je modul řízení dopravy. Tento prvek vyhodnocuje výstupy dopravního modelu, poveluje dopravní model a generuje parametry dopravnímu modelu, nad kterými jsou spouštěny jednotlivé simulace. Také, pokud jsou splněny předdefinované podmínky, poskytuje optimální hodnotu výstupu k distribuci.

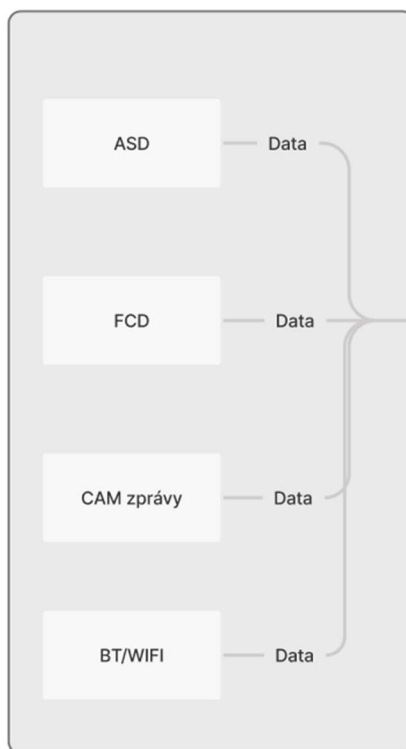
4.2 Popis jednotlivých částí algoritmu

V následujících podkapitolách jsou podrobně rozebrány jednotlivé prvky navrženého algoritmu dle schématu. Popsány jsou v pořadí z levé strany schématu k pravé.

4.2.1 Vstupní prvky

Část vstupních prvků obsahuje, jak již bylo naznačeno, prvky figurující v navrženém algoritmu zdroj vstupních parametrů do algoritmu. V první části této práce bylo uvedeno, že vstupní prvky algoritmu LŘD jsou ve většině případů totožné. Jedná se o dopravní detektory, jež poskytují dopravní data reflektující aktuální dopravní situaci. Oproti aktuálně využívaným algoritmům, jejichž hlavním vstupním prvkem jsou detektory indukčních smyček, je cílem návrhu tohoto algoritmu rozšířit jeho vstupní prvky o další detekční technologie které jsou aktuálně dostupné. Navíc je cílem navrhnout algoritmus

tak, aby relativně jednoduchou konfigurací byl rozšiřitelný o další budoucí možné datové zdroje. Důvod takto navrženého algoritmu je pokus o dosažení co nejpřesnější metody řízení, kdy je vycházeno z úvahy, že čím větší počet vstupních prvků vstupuje do algoritmu, tím lépe je algoritmus schopen v dopravním modelu dosáhnout situace odpovídající reálnému stavu a tím lepší by měla být jeho schopnost řídit řízenou komunikaci a predikovat její stav. Nicméně i přes větší počet vstupních prvků do algoritmu, nemusí všechny prvky přesně reflektovat aktuální dopravní stav. Některé vstupní prvky, jako například Bluetooth/Wifi detektory, zachytí pouze část vozidel dopravního proudu, jak bylo popsáno v kapitole 2.3, je proto nutné v rámci konfigurace jednotlivých vstupních prvků definovat prioritizaci. Tato prioritizace by měla přesně určit, které prvky budou sloužit v rámci algoritmu jako výchozí pro určení stavu dopravního proudu a které prvky slouží “pouze“ jako upřesňující. Veškerá konfigurace vstupních prvků, včetně prioritizace je definována v části MPC v prvku s názvem “Modul zpracování dat“, jež je podrobně rozebrán v další části práce. Výřez schématu, zobrazující popisovanou část navrženého algoritmu je zobrazen na obrázku níže.



Obrázek 39: Schéma části vstupních prvků

Další podkapitoly pojednávají o jednotlivých typech detekce, jež byly analyzovány v přechodných částech této práce. Nicméně návrh algoritmu je přizpůsoben variabilitě možných datových vstupů. Díky tomu může být počet vstupních prvků teoreticky neomezen. Prakticky je samozřejmě limitován jednotlivými technologiemi detekce.

4.2.1.1 ASD (Automatické Sčítače Dopravy)

Jedná se o detektory typu indukčních smyček. Tento vstupní prvek by měl být v rámci prioritizace zařazen na první místo. Dle provedené analýzy v kapitole 2.1 této práce se

totiž jedná o prvek který je schopen nejpřesněji reflektovat aktuální stav dopravního proudu, včetně možnosti klasifikace jednotlivých vozidel. Výstupem těchto prvků jsou data o intenzitách v měřených bodech. Na rozdíl od aktuálně využívaného algoritmu LŘD, budou výstupy z tohoto prvku převáděny na jednotková vozidla pouze za účelem statistického vyhodnocení. Hlavní vyhodnocení bude ale založeno počtech jednotlivých vozidel, z nichž budou vygenerovány intenzity vozidel každé kategorie, jež budou vstupovat do modelované sítě dopravního modelu. Dalším důvodem, proč by měl tento prvek být prioritní při určování dopravního proudu je jeho pokrytí na dálničních sítích.

4.2.1.2 FCD (Floating Car Data)

Dalším vstupním prvkem, s nižší prioritou vyhodnocení, než prvek ASD jsou detektory typu FCD. Podrobný popis tohoto typu detekce je dostupný v kapitole 2.2. Tyto detektory jsou schopny poskytovat dojezdové doby komunikace, na které se zrovna nachází. Je nutné v případě těchto detektorů filtrovat pouze detekce, které se stahují k řízené komunikaci. Výstupem těchto detektorů jsou dojezdové doby. Dojezdové doby mohou sloužit v rámci algoritmu ke kalibraci a validaci dopravního modelu.

4.2.1.3 Bluetooth/Wifi

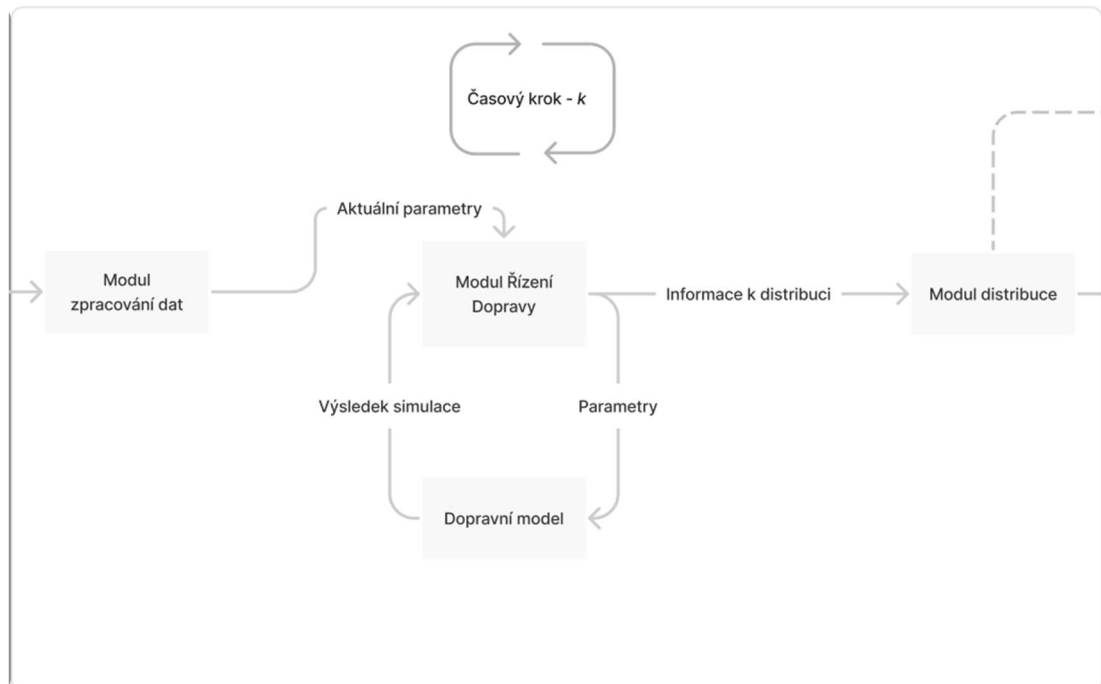
Tento vstupní prvek je výhodný hlavně vzhledem k jeho implementaci a nižší ceně. Princip technologie je podrobně popsán v kapitole 2.3. Výstupem této technologie jsou, stejně jako u FCD, dojezdové doby, ale na rozdíl od FCD dojezdové doby v přesně definovaných úsecích. Nicméně tato technologie nedokáže zachytit velkou část dopravního proudu a ještě nachází úskalí ve zpoždění dodaných výstupů. Z těchto důvodů je nutné tuto technologii v rámci prioritizace umístit na nižší příčky, stále se ale jedná o technologii, jejíž výstupy je možné v rámci algoritmu využít pro validaci a kalibraci dopravního modelu.

4.2.1.4 CAM zprávy

Detekce pomocí CAM zpráv není aktuálně vhodným vstupním prvkem, kvůli nízké penetraci vozidlových jednotek, nicméně předpokládá se, že by tato technologie v budoucnu mohla nahradit detekci technologií Bluetooth/Wifi, případně i detekci pomocí FCD. Výstupy této technologie jsou v podobě intenzit, rychlosti a dojezdových časů. Podrobně je tato technologie detekce popsána v kapitole 2.4. Kvůli aktuální nízké penetraci je ale nutné tuto technologii v prioritizaci vstupů do algoritmu prozatím umístit na poslední příčku.

4.2.2 MPC (Model Predictive Control)

Tato část je hlavní částí navrženého algoritmu. Její součástí jsou všechny prvky, potřebné pro úspěšné řízení komunikace pomocí systému VLŘD. Žádný prvek není v této části redundantní a každý má přesně definovanou svou úlohu, jakožto vstupní a výstupní parametry. Přesně definované jsou i vazby mezi jednotlivými prvky. Schéma vyobrazující část MPC je na obrázku 40.



Obrázek 40: schéma části MPC

Dále jsou popsány všechny prvky části MPC v pořadí z levé části schématu, do pravé.

4.2.2.1 Modul zpracování dat

Úlohou tohoto modulu je interpretace dat ze vstupních prvků do modulu řízení dopravy, který nad nimi spouští dopravní simulaci. Způsob interpretace by měl být v rámci modulu konfigurovatelný, nicméně jeho výstupem by měly být aktuální hodnoty intenzit jednotlivých klasifikačních souborů a hodnoty dojezdových dob v definovaném časovém okamžiku včetně jejich geografických informací.

Aby bylo možné v rámci tohoto modulu jednoduše přidávat nové vstupní prvky, je nutné navrhnout konfiguraci jednotlivých vstupních prvků. Tato konfigurace by měla být rozdělena na dvě části.

První částí by měla být konfigurace skupin detektorů. Ta by měla být realizována formou databáze obsahující ke každé skupině informace, které jsou pro skupinu napříč detektory totožné. Konfigurační databáze skupin by tedy minimálně měla obsahovat pro každou skupinu typ poskytovaných dat (intenzity, rychlosti, TT), časový interval poskytovaných dat, definici způsobu dodávání dat (zda budou data odesílána automaticky do modulu nebo si je bude modul sám stahovat). V případě stahování dat je nutné uvést koncový bod,

na kterém budou data dostupná), prioritu skupiny (číselná priorita detekční skupiny v intervalu <1:X>, kde X je počet vstupních prvků a číselné ohodnocení 1 znamenající nejvyšší prioritu a číselné ohodnocení X znamenající nejnižší prioritu), název skupiny a její unikátní identifikátor.

Druhou částí pak bude konfigurace jednotlivých detektorů opět realizována formou databáze obsahující informace o poloze detektoru, unikátní identifikátor detektoru (nejlépe shodný s identifikátorem přiřazeným poskytovatelem), identifikátor skupiny a pokud má detektor vlastní koncový přístup, tak jeho odkaz.

Vzhledem k identitě tohoto prvku se nabízí praktická realizace prvku ve dvou variantách. První variantou je stav, kdy jsou vstupními prvky dopravní data dodávána přímo do modulu zpracování dat. V tomto případě je vhodné zvolit realizaci pomocí tzv. REST API, kdy modul zpracování dat "poslouchá" pomocí tohoto rozhraní na zavolání od jednotlivých vstupních prvků, a které mu v rámci tohoto zavolání odešlou dopravní data s určitými parametry. Těmito parametry by měly být např. časové razítko, interval za jaký byla data naměřena, případně jestli se jedná o tzv. „vehicle by vehicle“ data, identifikátor a poloha detektoru. Další částí tohoto modulu by měla být služba, jež je schopna si z definovaného úložiště stáhnout datovou sadu v případě, že detektor sám data neodesílá do modulu zpracování dat.

Po shromáždění jednotlivých datových zdrojů je dalším úkolem modulu zpracování dat jejich interpretace pro modul řízení dopravy. Tato interpretace by měla spočívat v časové agregaci jednotlivých datových zdrojů a jejich rozdělení dle geografické polohy. Takto připravené datové vzorky posléze odešle do modulu řízení dopravy. Interpretovaná data by tedy měla mít podobu jednoho řádku pro agregovaný časový interval, obsahující v případě intenzit intenzity pro každou klasifikační kategorii a geografickou informaci. V případě dojezdových dob, by měl být výstupem jeden řádek, pro každý relevantní geografický úsek, obsahující čas dojezdové doby za agregovaný časový interval a geograficky ohraničený souřadnicemi začátku úseku a konce úseku.

4.2.2.2 Modul řízení dopravy

Tento prvek je centrálním prvkem navrženého algoritmu. Měl by na základě výstupů dopravního modelu definovat hodnoty rychlostních omezení, které by měly být v rámci algoritmu hlavním výstupem.

Prvním krokem pro určení tohoto výstupu je získání aktuálních dopravních parametrů, jež jsou poskytnuty modulem zpracování dat. Jak bylo zmíněno, dopravní data by měla být dodána v agregované podobě, kdy by pro každý detekční bod či úsek měla být dodána informace o hodnotě intenzit pro každou klasifikační kategorii, v případě dojezdových dob její hodnota času. Z takto poskytnutých dopravních dat vytvoří modul řízení dopravy vstupní podklady pro dopravní model, na jejichž základě spustí simulaci. Následně sleduje průběh simulace a vyhodnotí její výstup. O výstupu dopravní simulace bude podrobně pojednáno v další podkapitole, nicméně mělo by se ve stručnosti jednat o výstup obsahující relevantní informace o možnosti vzniku excesu, tedy například hodnoty časových zdržení na jednotlivých hranách, případně průměrná rychlost. Vzhledem ke

stochastickému chování dopravního proudu by modul řízení dopravy měl dopravních simulací spustit více najednou a statisticky porovnat jednotlivé výstupy. Navržený postup statistického porovnání jednotlivých výstupů dopravního modelu je pomocí aritmetického průměru jednotlivých hodnot, v souvislosti s kvantitativním počtem a určení pravděpodobnosti vzniku. Pro upřesnění navrhovaného způsobu vyhodnocení je možné uvést příklad, kdy výsledkem pěti spuštěných simulací byl v čase t ve dvou případech výsledek hodnot časových zdržení na hraně α např. deset a patnáct sekund a ve čtyřech případech výsledek hodnot časových zdržení na hraně β např. pět, deset, patnáct a dvanáct sekund. Modul řízení dopravy by měl v tomto případě z obou výstupů spočítat aritmetický průměr dle vzorce:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

Což je v prvním případě hodnota rovna 12,5 s a v druhém případě hodnota rovna 10,5 s. Ve vyhodnocení by měl figurovat i parametr n , tedy počet hodnot. Výsledkem by tedy v uvedeném příkladu měl být závěr vzniku časového zdržení v čase t od okamžiku měření na hraně α s hodnotou 12,5 s, s pravděpodobností 40 % a na hraně β s hodnotou 10,5 s, s pravděpodobností rovnou 80 %. Výsledné pravděpodobnosti jsou vyjádřeny statistickou definicí pravděpodobnosti, kdy se jedná o poměr úspěšných pokusů ku celkovému počtu pokusů.

Součástí tohoto modulu by měla být databáze, do níž by byly v každém kroku ukládány vstupní dopravní parametry dopravní simulace a výstupní hodnoty obsahující informace o vzniku dopravního excesu v simulaci času t (tedy časová hodnota od spuštění simulace) a jejich přidružená pravděpodobnost. Tato databáze umožňuje algoritmu “učení“, kdy by po delší době provozu byl schopen už pouze na základě dopravních vstupů bez provedení dopravní simulace určit, jak by mohla dopravní situace na sledované komunikaci v čase t od naměření hodnot vypadat (samozřejmě je nutné brát v potaz zpoždění při přenosu informací, stejně jako dobu vyhodnocení dopravního modelu). Po delší době provozu algoritmu by pak tedy bylo možné spouštět dopravní model pouze tehdy, pokud jsou vstupní dopravní parametry silně se odchylojící od již v minulosti zpracovaných vstupních dopravních parametrů.

Pokud jsou výstupem dopravního modulu informace o vzniku dopravního excesu, pokusí se modul řízení dopravy této situaci předejít snížením maximální povolené rychlosti v místě vzniku excesu a v definované délce úseku před ním, toto snížení promítnout do sítě dopravního modelu a znovu spustit dopravní simulaci a porovnat výsledky. Pokud dosáhl stavu, kdy se mu podařilo snížit čekací doby na minimum, případně maximalizovat průměrné rychlosti na úsecích k hodnotě vydaného rychlostního omezení, předá vygenerované rychlostní omezení, spolu s geografickými informacemi modulu distribuce, který je distribuuje účastníkům silničního provozu.

Databáze obsahující dopravní vstupní parametry a výstupy dopravního modelu závislé na těchto vstupech by měla být rozšířena také o informaci, jakým rychlostním omezením byl takový stav optimalizován. Opět jde o druh optimalizace, kdy by algoritmus po delší době

provozu nemusel vyhodnocovat všechny stavy, ale pouze jen ty, které ještě v minulosti neanalyzoval.

Součástí tohoto modulu by také měl být konfigurační soubor obsahující konfigurační parametry ohledně časové hodnoty t v sekundách, jež definuje časový interval, který má být simulován v rámci dopravního modelu. Součástí by také měla být definice rychlostních omezení, které může algoritmus generovat. V poslední řadě by měla obsahovat definici časového intervalu, po kterém má analýzu a generování rychlostních omezení provádět.

4.2.2.3 Dopravní model

Tento prvek je část algoritmu obsahující povelovatelný dopravní model, případně softwarový nástroj, jehož součástí jsou dopravní modely. Tento prvek nemusí být přímo součástí navrhovaného algoritmu, může se jednat o samostatný subjekt, který funguje nezávisle na algoritmu.

Jeho hlavním úkolem, z hlediska navrženého algoritmu, je simulování dopravního proudu na základě poskytnutých dopravních dat v časovém intervalu t od spuštění simulace. Cílem tohoto prvku je, aby dokázal co nejlépe reflektovat reálný stav a aby ho dokázal reflektovat i do budoucna. Díky tomu, že se jedná o programový nástroj, spuštěný na výpočetní technice, měl by být schopen v relativně krátkém časovém intervalu zobrazit stav simulované dopravní komunikace v budoucnu. Tento prvek je tedy do algoritmu zakomponován z hlediska predikce.

Dopravní model jako takový jsou matematické rovnice popisující pohyb vozidla, případně ve vyšších měřítkách celý shluk vozidel. Rozdělují se na mikroskopické, mesoskopické a makroskopické, kdy mikroskopické modely popisují pohyb jednotlivých vozidel, zatímco makroskopické popisují pohyb celého shluku vozidel. Pro přesnější výsledky modelované situace z hlediska navrženého algoritmu by byly nejvhodnější modely mikroskopické, jejichž úskalí je ovšem náročnější výpočetní náročnost. Nicméně v oblasti tohoto algoritmu je nutné simulovat pouze dálniční komunikace a jejich navazující vjezdy a sjezdy, což by mohlo v rámci výpočetní náročnosti mikroskopických modelů jejich výpočetní náročnost výrazně snížit.

Jak již bylo řečeno, tento prvek může být libovolně zvolen v rámci realizace navrženého algoritmu, pouze musí splňovat definované požadavky, které jsou pro algoritmus nezbytné. Těmito požadavky jsou:

- Jednoduché vložení dopravní sítě z externích zdrojů
- Možnost ovládání modelu programovými prostředky
- Možnost úpravy rychlostních omezení při běhu simulace
- Schopnost do jisté míry přesvědčivě reflektovat chování dopravního proudu
- Možnost spuštění několik simulací v jeden okamžik

V rámci vyhodnocení efektivity navrženého algoritmu byl jako tento prvek zvolen simulační SW SUMO, jež všechny tyto požadavky obsahuje, nicméně pokud by někdy v budoucnu došlo k reálné implementaci tohoto algoritmu, je možné tento prvek

navrhnout zcela samostatně kombinací jednotlivých dopravních modelů a vyhnout se užití softwarů 3. stran.

Výstupními parametry tohoto prvku by měly být geografické informace o komunikaci, na které má vzniknout dopravní exces a pravděpodobnost jeho vzniku.

Konfigurace tohoto prvku je předpokládána jako nejkompexnější ze všech prvků navrženého algoritmu. Její součástí by měly být definice dopravního proudu, definice simulované dopravní sítě a konfigurace jednotlivých parametrů modelu. Tuto konfiguraci by ale měl, krom definice dopravní sítě, obsluhovat modul řízení dopravy.

4.2.2.4 Modul distribuce

Prvek navrženého algoritmu modul distribuce má za úkol na základě parametrů převzatých z modulu řízení dopravy, distribuovat tyto parametry účastníkům silničního provozu.

Logika tohoto prvku by měla být velmi jednoduchá. Pouze pokud dostane vstupní parametry, tak je distribuuje pomocí geografických informací obsažených ve vstupním parametru. Pokud vstupní parametry neobdrží, neměl by nic distribuovat. Primitivní logikou tohoto prvku by měla být zajištěna nižší poruchovost tohoto prvku.

Součástí prvku by, jako u ostatních prvků tohoto algoritmu, měla být konfigurace, která definuje cílový distribuční kanál. Distribučních kanálů může být v rámci konfigurace víc. Každý distribuční kanál by měl být definován v konfiguraci včetně formátu výstupní zprávy a distribuční modul by měl v rámci své funkce vstupní parametry postupně na všechny takto definované kanály odeslat.

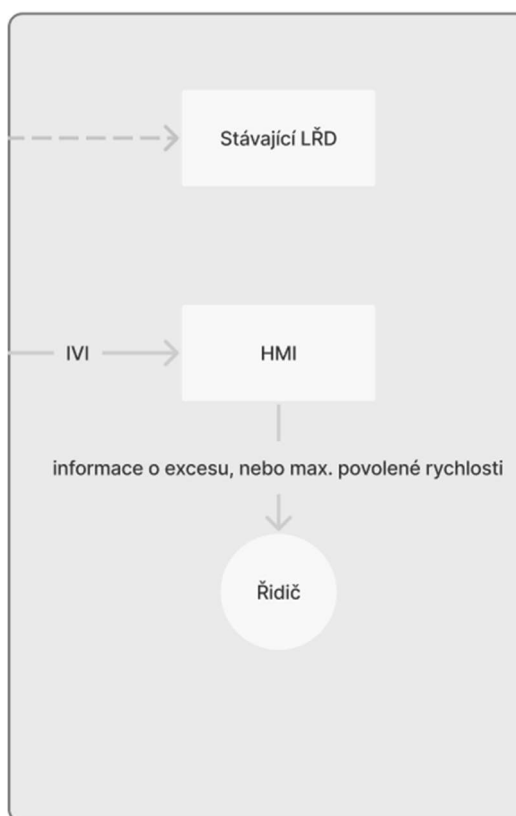
V hlavním schématu je naznačena vazba i na stávající systém LŘD, nicméně tato vazba je pouze orientační a má naznačovat možnost variability výstupních kanálů, kdy výstupních kanálů může být n počet. V případě napojení na stávající systém LŘD by však bylo nutné rozšířit modul distribuce o službu, která by rychlostní omezení vygenerované modulem řízení dopravy přiřazovala jednotlivým řezům LŘD.

Jelikož se jedná o virtuální LŘD, jeví se jako aktuálně nejvhodnější distribuční médium zprávy z oboru C-ITS IVI zprávy, tedy In Vehicle Information. IVI zprávy jsou virtuální portály, které se šíří pomocí V2X komunikace. Z hlediska VLŘD se jedná o optimální způsob distribuce, protože je možné pomocí této zprávy přenést informaci o rychlostním omezení vygenerované algoritmem VLŘD.

4.2.3 Výstupní prvky

Poslední částí schématu navrženého algoritmu je nastínění výstupních prvků navrženého algoritmu. Vzhledem k tomu, že se jedná o virtuální LŘD je hlavním navrženým výstupním prvkem HMI, jež je schopno interpretovat IVI zprávy, které jsou navrženy jako hlavní distribuční médium. Dalším hlavním výstupním prvkem, jež reaguje na výstupy navrženého algoritmu je účastník silničního provozu obsluhující řízené vozidlo, zkráceně řidič. Z hlediska legislativního ale není řidič povinen na výstupy algoritmu VLŘD reagovat, proto jsou rychlostní omezení generována algoritmem klasifikována jako doporučená.

Jak již bylo zmíněno, výstupních prvků v rámci algoritmu může být n počet. Jednotlivé výstupní prvky je možné definovat v konfiguraci modulu distribuce, jež je součástí části MPC a má za úkol distribuovat výstupy algoritmu VLŘD do jednotlivých výstupních prvků, definovaných právě v konfiguraci modulu distribuce. Je tedy možné algoritmus rozšířit o další výstupní prvky v průběhu jeho provozu. Naznačení výstupních prvků navrženého algoritmu je vyobrazeno na obrázku 41.



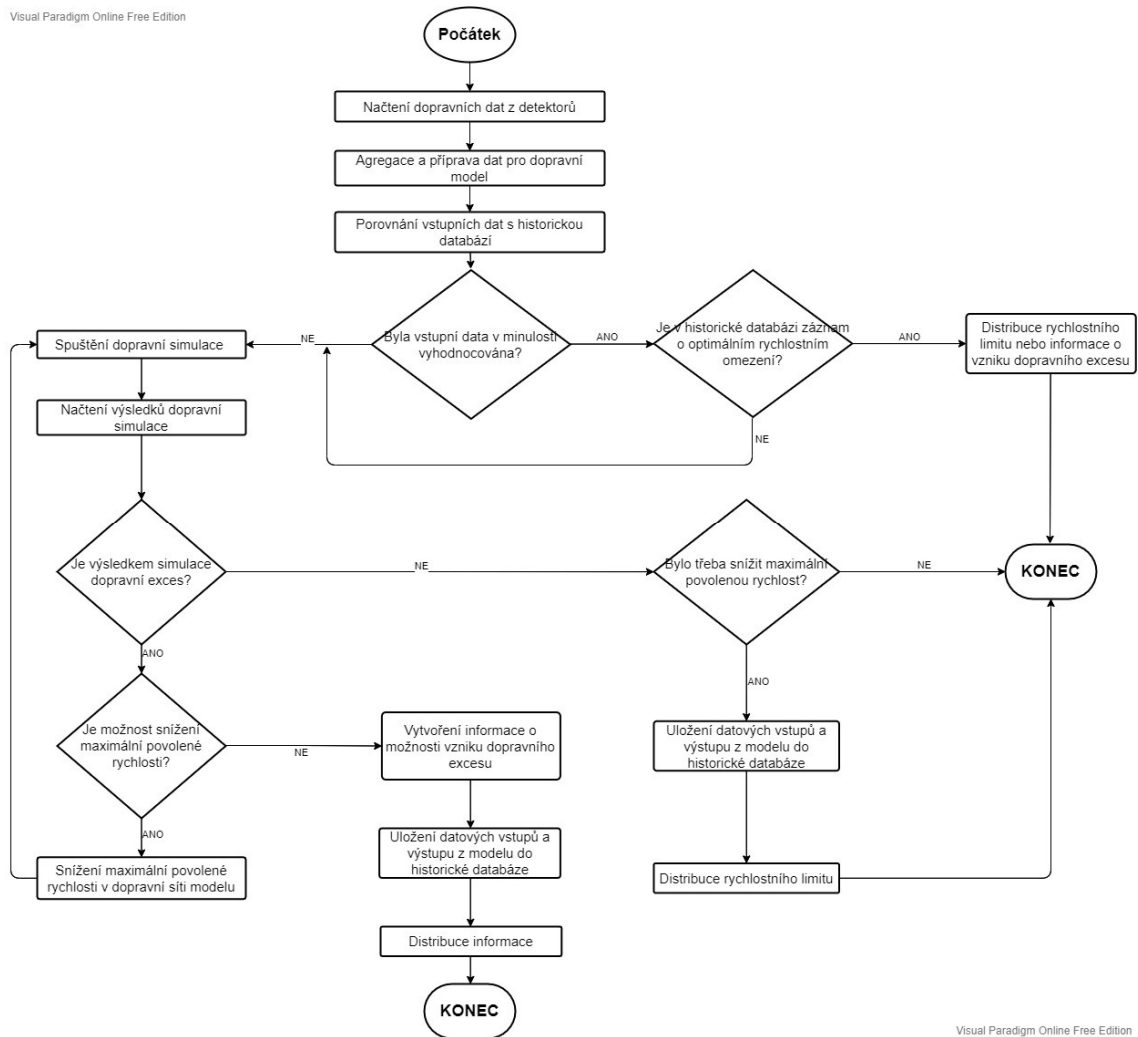
Obrázek 41: Schéma výstupních prvků navrženého algoritmu

Ve schématu výstupních prvků navrženého algoritmu je vyobrazena vazba výstupního prvku HMI na řidiče. Tato vazba by měla být primárním výstupem algoritmu. Ve schématu je také naznačena vazba na stávající systém LŘD. Tato vazba je ovšem vyobrazena pouze jako orientační a je primárně myšlena jako naznačení variability výstupních prvků. Samotná implementace do stávajícího systému LŘD je možná,

nicméně její implementace by vyžadovala více úsilí než pouhou definici v konfiguraci modulu distribuce. To z toho důvodu, že aktuální systém LŘD je vázán na fyzické řezy, na nichž jsou následně zobrazovány rychlostní omezení. V rámci algoritmu VLŘD by v případě implementace do stávajícího systému LŘD bylo nutné implementovat další službu, jež by interpretovala výstupy navrženého algoritmu do geografických poloh jednotlivých řezů.

4.3 Popis funkce algoritmu

Předmětem této podkapitoly je vlastní popis funkce algoritmu jako celku. Podrobně je v této části popsáno, jak algoritmus postupuje v jednom časovém kroku k , jaké má v rámci kroku k dispozici rozhodnutí a jak na ně reaguje. Pro lepší přehlednost byl v rámci této podkapitoly vytvořen jednoduchý vývojový diagram, zobrazující jednotlivé události v rámci jednoho časového kroku k . Vývojový diagram je vyobrazen na obrázku 42.



Obrázek 42: Vývojový diagram navrženého algoritmu v jednom časovém kroku k

Na počátku každého časového kroku k je prvním procesem načtení aktuálních dopravních dat z dopravních detektorů. Tento proces je pod gescí modulu zpracování dat. Následujícím procesem v gesci stejného modulu je agregace a interpretace dopravních dat, aby nad nimi mohla být spuštěna simulace.

Další proces je již v gesci modulu MPC a jeho účelem je porovnání poskytnutých dopravních parametrů s databází historicky vyhodnocených dopravních parametrů. Vyhodnocení má za primární úlohu ušetření výpočetních zdrojů, díky rozhodnutí nespouštět simulaci, pokud již byly podobné dopravní parametry v minulosti modelem vyhodnocovány. Takto naznačené porovnání s jeho jednotlivými důsledky je vyobrazeno ve vývojovém diagramu v rozhodovacím prvku s názvem “Byla vstupní data v minulosti vyhodnocována?”. Pokud algoritmus dojde k závěru, že vstupní dopravní data byla již v minulosti vyhodnocena, postoupí ve vývojovém diagramu k rozhodovacímu prvku s názvem “Je v historické databázi záznam o optimálním rychlostním omezení?”. Tento prvek slouží primárně jako kontrolní a má za účel zkontrolovat zda k poskytnutým datům, pokud se v databázi nachází, existuje přidružená hodnota optimálního rychlostního omezení (logika vyhodnocování je komplexnější a zahrnuje porovnávání trendů jednotlivých dopravních veličin. Podrobnější popis je možné nalézt v podkapitole 4.3.4). Pokud neexistuje, algoritmus se bude chovat jako by takový záznam v databázi vůbec neexistoval a bude pokračovat směrem vyhodnocení jako v případě neznámé vstupní hodnoty. Pokud ale přidružená hodnota optimálního rychlostního omezení existuje, algoritmus tuto hodnotu použije k distribuci koncovým uživatelům prostřednictvím modulu distribuce a jeho úloha v tomto časovém kroku končí.

Pokud ale algoritmus dojde k závěru, že vstupní dopravní parametry ještě v minulosti nevyhodnocoval, postoupí ve vývojovém algoritmu do jeho levé části. V této části se algoritmus pokusí spustit dopravní simulaci nad poskytnutými dopravními parametry. Výsledek simulace může z hlediska algoritmu dosáhnout dvou různých stavů. Prvním je, že výsledek z výstupních hodnot simulace bude naznačovat vysokou pravděpodobnost vzniku dopravního excesu při poskytnutých dopravních parametrech. Druhým stavem je, že výsledek z výstupních hodnot simulace nebude naznačovat vysokou pravděpodobnost vzniku dopravního excesu při poskytnutých dopravních parametrech. Takto popsany rozhodovací prvek je ve schématu výše zobrazen pod názvem “Je výsledkem simulace dopravní exces?”.

V případě druhého stavu postupuje algoritmus ve vývojovém diagramu do prostřední části schématu, kde ověřuje, zda bylo před spuštěním simulace nutné snížit maximální povolené rychlosti na simulované dopravní síti. Pokud dojde k závěru, že snížení maximálních povolených rychlostí nebylo provedeno, tak v tomto případě není nutnost ze strany algoritmu generovat rychlostní omezení a v takovémto stavu se v tomto časovém kroku ukončí.

V případě prvního stavu, tedy že výsledkem simulace byl dopravní exces, se algoritmus dostává ve vývojovém prvku k rozhodovacímu prvku s názvem “Je možnost snížení maximální povolené rychlosti?”. Systém LŘD obecně by měl možnost snižovat rychlost pouze do omezené rychlostní hodnoty a s určitými pravidly (pravidla jsou více rozebrána

v podkapitole 4.3.3). Z logického hlediska by neměl mít možnost například snížit rychlostní omezení až na nulovou hodnotu, takové rychlostní omezení by nebylo efektivní. Zároveň by tato hodnota neměla být ani příliš blízká nule. Většina algoritmů LŘD je navržena tak, aby minimální hodnota rychlostního omezení, kterou mohou vygenerovat byla v intervalu 60-80 km/h [7] [9]. Minimální hodnota rychlostního omezení, kterou může vygenerovat algoritmus VLŘD je konfigurovatelná v rámci konfigurace modulu řízení dopravy části MPC, jak bylo popsáno v kapitole 4.2.2.2. Rozhodovací prvek “Je možnost snížení maximální povolené rychlosti?” by tedy měl kontrolovat aktuální hodnotu maximální povolené rychlosti na modelované dopravní síti oproti hodnotě minimálního rychlostního omezení z konfigurace modulu řízení dopravy. Pokud je porovnáním zjištěné, že hodnoty jsou stejné, algoritmus VLŘD nemá v tomto případě možnost, jak danou dopravní situaci svými výstupy ovlivnit, a protože výsledkem dopravní situace byl dopravní exces, poskytne tuto informaci účastníkům silničního provozu, předtím ale tuto informaci zapíše do historické databáze, aby v budoucích vyhodnocení tuto situaci s poskytnutými dopravními parametry předpokládal. Předání informace o vysoké pravděpodobnosti vzniku dopravního excesu je realizováno textovou informací, jež se zobrazí řidiči na HMI ve virtuálním portálu v případě distribuce IVI zprávami.

Pokud algoritmus dojde k závěru, že má možnost snížit maximální povolenou rychlost na řízené silniční síti, učiní tak a v dalším kroku se vrací zpět na proces spuštění simulace, jak je naznačeno v levé části diagramu. Procesy a rozhodovací podmínky nastíněny v levé části algoritmu by měly opakovaným vykonáváním dojít k určení optimálního rychlostního omezení. Pokud v rozhodovacím procesu došel algoritmus do stavu, kdy nebyl výsledkem simulace dopravní exces a zároveň bylo třeba v předešlých procesech snížit maximální povolenou rychlost, tak tuto hodnotu maximální povolené rychlosti, nastavené v modelu uloží do historické databáze navázanou na vstupní hodnoty. Po tomto procesu, pomocí modulu distribuce, distribuuje vygenerovanou hodnotu účastníkům silničního provozu a v tomto časovém kroku jeho činnost končí.

Takto jsou ve stručnosti popsány procesy v jednom časovém kroku k . Časový krok je možné opakovat po zvoleném časovém intervalu a měl by se opakovat v rámci provozu algoritmu neustále.

4.3.1 Možné poruchové stavy

V rámci navrženého algoritmu je již při návrhu možné definovat některé poruchové stavy, které mohou při provozu nastat. Obsahem této podkapitoly tedy je výčet těchto poruchových stavů a navržený postup, jak by v těchto stavech měl algoritmus reagovat. Jelikož se nejedná o klíčový systém, bez jehož funkčnosti by mohlo dojít k velkým finančním či věcným ztrátám, je většina poruchových stavů řešena způsobem nečinnosti algoritmu. Nicméně při této nečinnosti by měla být zajištěna informovanost provozovatele systému, aby se poruše věnoval a aktivně ji řešil.

Samotný algoritmus by měl mít implementovaný způsob detekce nefunkčnosti jeho částí. To lze implementovat například způsobem tzv. "timeoutů" kdy, pokud řídicí část algoritmu nedostane při povolování libovolného prvku algoritmu do definovaného časového intervalu odpověď, je takovýto prvek považován za nedostupný.

4.3.1.1 Nedostupnost dopravního modelu

V rámci provozu navrženého algoritmu může nastat situace, kdy prvek dopravního modelu nebude z určitého důvodu dostupný. Důvodů může být více, například síťová nedostupnost, nebo disfunkce samotného modelu, nicméně všechny důvody vyústí v nemožnost určení maximální povolené rychlosti.

V tomto případě je nutné, aby prvně o této skutečnosti byl informován provozovatel systému. Toto informování by mělo být periodické a nemělo by ustát do té doby, dokud nebude zajištěna opětovná dostupnost modelu. Po celou dobu nedostupnosti modelu by měl při nabytí nových dopravních parametrů pouze vyhodnocovat tyto parametry oproti historické databázi a v případě, že má záznamy o optimální hodnotě maximální povolené rychlosti, tak tyto informace distribuovat. V případě, že tyto informace dostupné nemá, nesmí žádné informace distribuovat a jeho činnost by měla být omezena.

Je žádoucí, aby nedostupnost tohoto typu kontroloval řídicí modul dopravy.

4.3.1.2 Nedostupnost vstupních datových zdrojů

Tato nedostupnost může být celková nebo částečná. Celkovou nedostupností je myšleno, že více než 50 % definovaných datových zdrojů v modulu zpracování dopravních dat je nedostupných. V tomto případě by algoritmus měl o této skutečnosti periodicky informovat provozovatele systému a neměl by distribuovat jakékoliv výstupy.

Částečnou nedostupností je myšlen stav, kdy je nedostupných definovaných datových zdrojů méně než 50 %. V tomto případě by se algoritmus měl pokusit nahradit chybějící datové zdroje pro dopravní model z historické databáze. Pokud v historické databázi požadované hodnoty nejsou, opět by se algoritmus měl dostat do stavu, kdy nedistribuuje žádné výstupy. Stejně jako v případě celkové nedostupnosti i v tomto případě je ale prvním krokem informovat o tomto stavu provozovatele systému.

Je žádoucí, aby nedostupnost tohoto typu kontroloval modul zpracování dopravních dat

4.3.1.3 Nedostupnost distribučních kanálů

V rámci provozu algoritmu může dojít k nedostupnosti některých distribučních kanálů definovaných v modulu distribuce. V tomto případě je nutné, aby byl opět periodicky informován o této skutečnosti provozovatel systému. Jelikož v tomto případě není algoritmus schopen jakkoliv distribuovat své výstupy, neměl by se o to tedy ani pokoušet. Nicméně stále je žádoucí, aby pokračoval ve vyhodnocování jednotlivých vstupních parametrů, jež mu jsou poskytovány.

Je žádoucí, aby nedostupnost tohoto typu kontroloval modul distribuce.

4.3.2 Provozní režimy algoritmu

Jako v případě již navržených algoritmů i tento ve svém návrhu zahrnuje jednotlivé provozní režimy, ve kterých je algoritmus možné provozovat. Na rozdíl od aktuálně využívaného algoritmu v ČR jsou pro tento algoritmus navrženy pouze dva provozní režimy. Automatický a poloautomatický. Důvodem pro návrh pouze dvou provozních režimů, navíc těch, ze kterých je víceméně vyloučen lidský faktor, je předpokládané zvýšení efektivity řízení.

4.3.2.1 Automatický režim

Automatický režim je inspirován automatickým režimem algoritmu využívaného v ČR. V tomto režimu je provoz algoritmu plně automatizován a není v jeho průběhu nutný jakýkoliv zásah operátora. Algoritmus v každém časovém kroku k vyhodnocuje jednotlivé datové vstupy a na jejich základě vytváří odpovídající rychlostní omezení. Operátor je schopen je kontrolovat na centrálním HMI, kde jsou vyobrazeny všechny výstupy algoritmu nad mapovými podklady. Nemá ale možnost v tomto režimu jakkoliv do výstupů zasahovat. Tento režim je z hlediska návrhu algoritmu považován za hlavní provozní režim, který by měl být aktivován po většinu času jeho provozu.

4.3.2.2 Poloautomatický režim

Tento režim je obdoba automatického režimu s tím rozdílem, že jednotlivé výstupy algoritmu jsou předávány operátorovi ke schválení. V režimu schválení je operátor schopen před schválením v HMI virtuální portály upravit. Měl by být schopen upravit rychlostní omezení, případně textovou informaci, pokud je součástí virtuálního portálu. Neměl by ale mít možnost upravit geografickou zónu, ve které se má portál zobrazit. Pokud operátor odmítne portál navržený algoritmem, portál se odstraní. Tento režim by měl sloužit výhradně ke kalibraci navrženého algoritmu, případně ke korekci, pokud by některé výstupy algoritmu operátor vyhodnotil jako nevalidní. Nicméně nemělo by se jednat o režim, který by byl z hlediska provozu algoritmu spuštěn po většinu času jeho provozu.

4.3.3 Harmonizace rychlostních omezení

Bod, který ještě v rámci návrhu adaptivního algoritmu VLŘD nezazněl, je harmonizace rychlostních omezení. V rámci modulu řízení dopravy je nutné mít implementovanou logiku distribuce rychlostních omezení a zároveň logiku generování těchto omezení. Tyto logiky by měly předcházet problémům, jež mohou vyústit při špatném postupu distribuce či generování rychlostních omezení.

Pokud je výstupem modulu dopravního modelu dopravní exces a modul řízení dopravy dojde k závěru, že je nutné vytvořit rychlostní omezení pro zvýšení kapacity dané komunikace, musí postupovat dle určitých pravidel. Pravidla musí zabezpečit, že například algoritmus nebude generovat rychlostní omezení, jejichž rozdíl s aktuálně platným rychlostním omezením nebude větší než 20 km/h [2]. Jinými slovy, je nežádoucí, aby algoritmus nastavoval rychlostní omezení hodnoty 60 km/h na komunikaci, jejíž aktuální rychlostní omezení je 130 km/h.

Pokud se algoritmus rozhodne pro rychlostní omezení 60 km/h na komunikaci kde je aktuálně nastavena hodnota rychlostního omezení 130 km/h, je nutné, aby předchozí silniční hrany byly postupně připraveny na tuto změnu. To je v případě aktuální LŘD řešeno postupným zvyšováním rychlostního omezení na hranách před cílovou hranou. Pokud tedy například algoritmus rozhodne, že nastaví rychlostní omezení 60 km/h, další 3 řezy před ním ukazují rychlostní omezení o 20 km/h. Tedy řez $x-1$ zobrazuje rychlostní omezení 80, řez $x-2$ rychlostní omezení 100 a řez $x-3$ rychlostní omezení 120. Tímto je zajištěna harmonizace dopravy před místem, kde by měl algoritmus nařídít vyšší rozdíl rychlostí [2].

V rámci návrhu algoritmu VLŘD je doporučeno tuto část algoritmu implementovat v souladu s návrhem aktuálního algoritmu LŘD v ČR, protože takto navrhnutý je vyhovující.

4.3.4 Prognóza trendu

Pro dosažení uspokojivých výsledků při predikci pomocí modulu dopravního modelu je vhodné, aby dopravnímu modelu byla předána před spuštěním simulace informace o trendu naměřených veličin. Tento trend by následně měl být zohledněn při simulaci a je i možné v budoucím rozvoji zahrnout tento trend do rozhodovacího procesu, zda je nebo není nutné spouštět dopravní simulaci.

Hodnota trendu může u různých naměřených veličin do jisté míry naznačit, jakým směrem se bude do budoucna vyvíjet. U každé naměřené dopravní veličiny to může znamenat něco jiného. Například vzrůstající trend intenzity může naznačovat, že na sledovanou komunikaci najíždí více vozidel a tím by se za určitou dobu mohla naplnit její kapacita. Naopak klesající trend intenzity může předznamenat volný dopravní proud. Naopak u rychlosti klesající trend může znamenat právě se tvořící kongesci. Rostoucí trend rychlosti může značit právě se rozjíždějící kongesci.

Kombinací těchto hodnot by bylo možné odvodit, zda má doprava klesající tendenci či vzrůstající. Ve vzrůstajících případech je možné pak spustit modul dopravního modelu, aby ověřil, zda se kapacita komunikace při takovém rostoucím trendu nezaplní. Tuto hodnotu by bylo možné přidat do historické databáze a pak na jejím základě porovnávat i hodnoty těchto koeficientů v minulosti.

Je také nutné prognózu trendu vzít v úvahu při generování vstupních parametrů dopravního modelu, případně modelu tuto hodnotu (pokud s ní umí pracovat) přímo

poskytnout. V případě generování vstupních parametrů do modelu je nutné vstupní parametry, jimiž mohou být například intenzity, vynásobit touto naměřenou hodnotou, aby se vstupy dopravního modulu zvyšovaly, případně snižovaly dle toho, jak se aktuálně dopravní proud vyvíjí.

V rámci ověření návrhu algoritmu byl implementován model lineární regrese, který spočte koeficient vzrůstu či poklesu proložením naměřených dat přímkou. Tento model není ideální, protože nemusí úplně přesně reagovat na změny ve vstupních datech, nicméně jeho obdoba je implementována i v rámci algoritmu stávajícího systému LŘD [2].

Kapitola 5

Ověření návrhu algoritmu ve zvoleném simulačním SW

Obsahem této kapitoly je popis způsobu ověření navrženého algoritmu ve zvoleném simulačním SW a následná prezentace výsledků. Ve 3. kapitole této práce byl jako simulační software zvolen nástroj SUMO, jenž vyhovuje požadavkům ověřovacího procesu.

5.1 Ověřovací proces

Vzhledem k vysoké náročnosti určení přesných definic modelových koeficientů, aby se modely chovaly obdobně jako sledovaná realita, byl v rámci ověřovacího procesu zvolen přístup, jenž by umožnil algoritmus ověřit i bez definic modelových koeficientů.

Přístup spočívá v ověření návrhu algoritmu mezi dvěma simulacemi s totožnými modelovými koeficienty. Jeden z modelů bude představovat v ověřovacím procesu realitu, druhý bude sloužit jako nástroj predikce dopravních hodnot právě na síti modelu, který je považován za realitu. Tímto přístupem je možné zajistit ideální stav, na kterém je možné ověřit, zda algoritmus provádí správně požadovanou činnost. Taky je díky tomuto přístupu jednodušší sběr výsledných dat a jejich následné vykreslování.

Cílem je nasimulovat dva dopravní stavy v definovaném úseku. Prvním stavem je stav, kdy je doprava stabilní, druhým naopak stav, kdy je doprava těsně před stavem kolapsu. V obou těchto případech sledovat chování měřených veličin. Následným krokem je v obou stavech spustit navržený algoritmus a opět sledovat chování naměřených veličin.

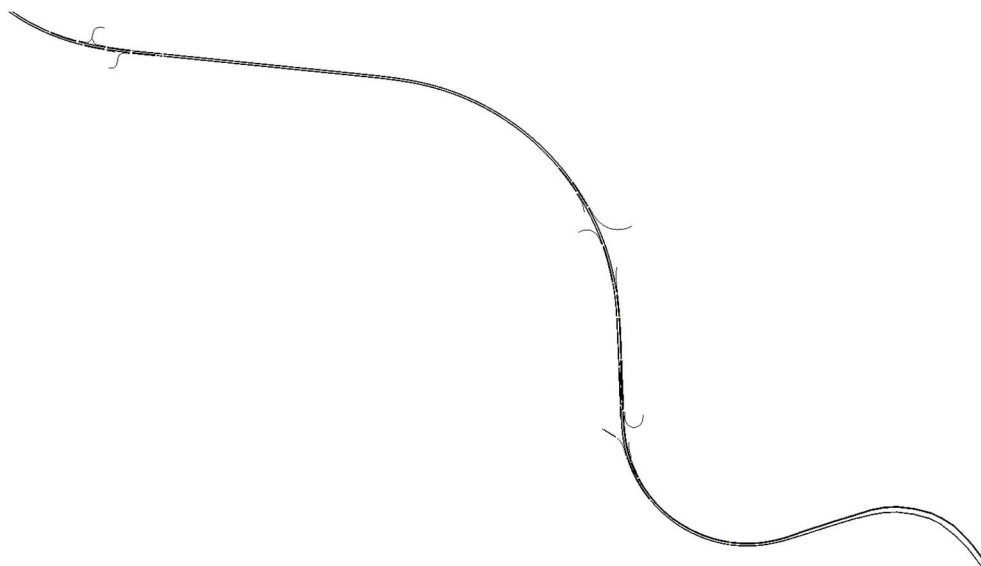
Výsledné vykreslení hodnot naměřených dopravních veličin by mělo prokazatelně vypovědět účinnost navrženého algoritmu.

Takto navržený proces není nejvhodnější z hlediska komplexního ověření navrženého algoritmu kvůli předpokladu, že dopravní model používaný k predikci plně odpovídá měřené realitě. Ve skutečném světě tomu tak není a na kalibraci modelu, aby co nejlépe odpovídal měřené realitě, je možné nahlížet téměř jako na samostatný vědní obor, proto byl v rámci této práce zvolen postup nastíněný výše, který pro základní ověření funkčnosti navrženého algoritmu postačí. Pokud budou výsledky ověření pozitivní, je možné se v dalším kroku zabývat definicí koeficientů modelu. Tyto možnosti jsou shrnuty v kapitole 6.

5.2 Implementace ověřovacího procesu

V předchozí podkapitole byl nastíněn ověřovací proces, pomocí kterého je ověřen navržený algoritmus. Tato podkapitola stručně shrne, jak byl tento ověřovací proces implementován.

Celý ověřovací proces, i implementace navrženého algoritmu, byla provedena v prostředí Python. V prvním kroku byla vytvořena dopravní síť, na které bylo ověření provedeno. Jako testovací dopravní síť byla zvolena část Pražského okruhu D0, konkrétně se jednalo o úsek mezi Lochkovem a Ořechem. Podobu ověřovacího úseku je možné vidět na obrázku 43.

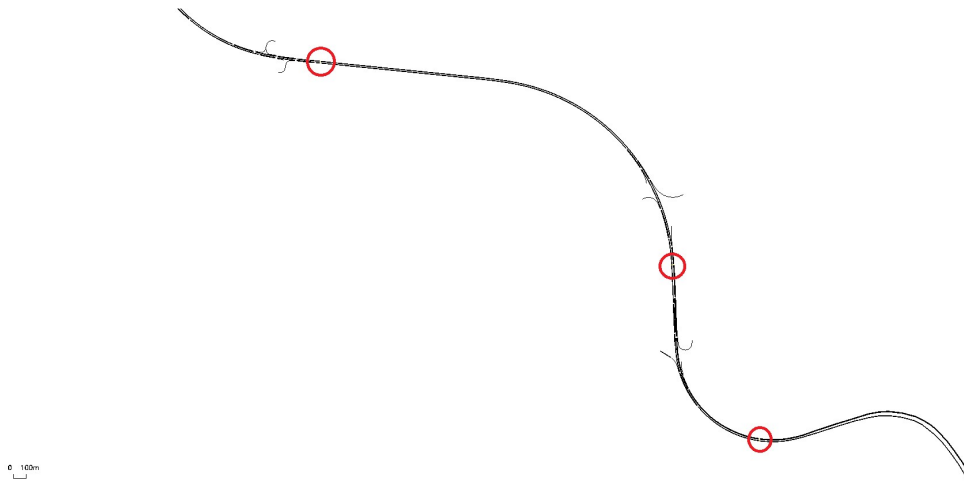


Obrázek 43: Ověřovací úsek dopravní sítě v SUMO-GUI

Postup importování dopravní sítě byl totožný s tím, jak je popsán v kapitole 3.1.2.2. Zdrojová dopravní síť je ze zdroje [30].

Dopravní požadavky byly do modelu, jenž má v rámci ověření představovat realitu, vygenerovány na základě dopravních dat z indukčních smyčkových detektorů ze dne 16.5. 2012. Novější data nebyla v rámci vypracování této práce k dispozici, nicméně pro ověřovací proces je datová sada dostačující. Datové vstupy byly v podobě agregovaných minutových hodnot intenzit, rychlostí a obsazenosti, kategoricky rozděleny z obou pruhů.

Pozice jednotlivých detektorů, ze kterých byly využity datové sady v rámci ověřovacího procesu jsou zachyceny na obrázku 44.



Obrázek 44: Vyznačené polohy jednotlivých detektorů v ověřovací síti

Dalším krokem byl převod data smyčkových detektorů do vstupů simulované dopravní sítě. Postupů převodu bylo zvažováno několik, protože postup přímo souvisí s vazbou na parametry v obrázku 43 z modulu řízení dopravy do dopravního modelu. Touto vazbou je především myšleno předání předpřipravených vstupních souborů vygenerovaných na základě vstupních dat pro dopravní model tak, aby byl posléze dopravní model pouze spuštěn a byly z něho následně vyčteny výsledky.

Jednotlivé hodnoty naměřených intenzit by měly korespondovat s počty jednotlivých vozidel vpuštěných do dopravního modelu. Nicméně ve skutečném světě není možné z minimálně ze smyčkových detektorů změřit cílovou destinaci řidiče. Přitom cíl vozidla je z hlediska modelování jeden z podstatných faktorů kdy, pokud je více řidiči preferován jeden cíl, může to znamenat nasycení hran, které k preferovanému cíli vedou a tím se zvyšuje pravděpodobnost kongesce. Pokud model nebere tyto preference v potaz, může v následných simulacích docházet ke špatným závěrům.

Tento problém je možné částečně vyřešit O/D maticemi, jež obsahují informace o vstupní a výstupní hraně každého vozidla. Nicméně O/D matice z tohoto úseku nebyly v době psaní diplomové práce k dispozici, proto byl zvolen jiný přístup.

Sít' byla rozdělena na množiny počátečních hran a na množiny hran vstupních. Počáteční hrany jsou hrany nemající žádného předchůdce nacházejícího se na dálnici. Vstupními hranami byly definovány hrany, jež se nenacházejí na dálnici, ale také nemají žádného předchůdce. Vstupní hrany jinými slovy symbolizují nájezdy na dálnici a počáteční hrany symbolizují vstupní body dopravního proudu který se již na dálnici nacházel. Při úvaze nad dálniční sítí lze dospět závěru, že vozidla, která se na ní pohybují, mají relativně malou množinu možností kudy se na dálniční sít' dostat, stejně jako množina prvků výstupních. V případě obrázku 43 je velikost množiny možných vstupů do dálnice rovna 4 v jednom směru, přičemž velká většina vozidel se již v dopravním proudu bude nacházet, tedy její vstupní hranou bude počátek sítě, jenž je v případě obrázku 43 hrana v levém horním rohu.

Pro další popis postupu generování jednotlivých vstupních souborů je zapotřebí definovat tyto proměnné:

- Q_v : Intenzita vstupních hran
- Q_p : Intenzita počátečních hran

Postup zvoleného generování tedy spočíval v přiřazení hodnoty intenzit množině počátečních hran a následném porovnávání hodnot Q_p a hodnot Q_v (reprezentována detektorem umístěným za ní). Pokud platí:

$$Q_v < Q_p$$

Tak hodnota rozdílu Q_p a Q_v je ze simulace nejbližším výstupem před vstupní hranou odvedena ven ze simulace.

Pokud platí že:

$$Q_v > Q_p$$

Tak je naopak vstupní hraně přiřazena intenzita, která je rovna rozdílu Q_v a Q_p . Tímto postupem jsou nakonec v jedné iteraci vygenerovány vozidlové vstupy pro všechny vstupní a počáteční hrany v jednom měřeném intervalu pro oba směry. Pro alespoň malé nastolení stochasticity ověřovaného silničního modelu byla každému vozidlu vybrána náhodná výstupní hrana.

Tento postup má své nevýhody. Z důvodu absence O/D matic, ze kterých by bylo možné spočítat odbočovací poměry na jednotlivých výstupních hranách není model zdaleka dokonalý. Nicméně v rámci ověření byl postup generování vstupních souborů jak pro model, jenž má zastupovat realitu, tak pro model, jenž je využíván v rámci predikce, stejný. Tím se zajistí opět podmínka, že model využívaný pro predikci je schopen relativně přesně popsat sledovanou realitu. Ve skutečném světě není možné takovýto postup generování použít, protože by predikční model docházel ke špatným závěrům. V rámci skutečnosti by bylo vhodné využít pro generování vstupních hodnot modelu na základě intenzit O/D matic, ze kterých by byla jednotlivá vozidla z modelu směřována pryč na základě skutečných pozorování (samozřejmě za předpokladu, že O/D matice je vytvořena z reprezentativního vzorku měření a není chybná).

Dalším krokem byla implementace navrženého algoritmu. Implementace byla provedena přesně podle návrhu, jež byl popsán v kapitole 4. Nicméně některé části algoritmu byly v rámci ověření zjednodušeny. Harmonizace rychlostních omezení byla v implementaci návrhu algoritmu zahrnuta, ale jen velmi zjednodušeně. Optimální implementace harmonizace rychlostních signálů je komplexní problém, jehož řešení přesahuje meze této práce. Byla tedy v rámci ověření zvolena varianta, kdy bude pro ověření funkčnosti algoritmus spínat rychlostní omezení pouze na 100 a 80 km/h. V ostrém provozu je vhodné při implementaci harmonizace využít postup, jenž je popsán v provozním řádu aktuálně funkčního systému LŘD [2].

Pro určení koeficientu prognózy trendu, ke kterému je přihlíženo při generování vstupních hodnot byla v rámci ověření zvolena metoda jednoduché lineární regrese, která

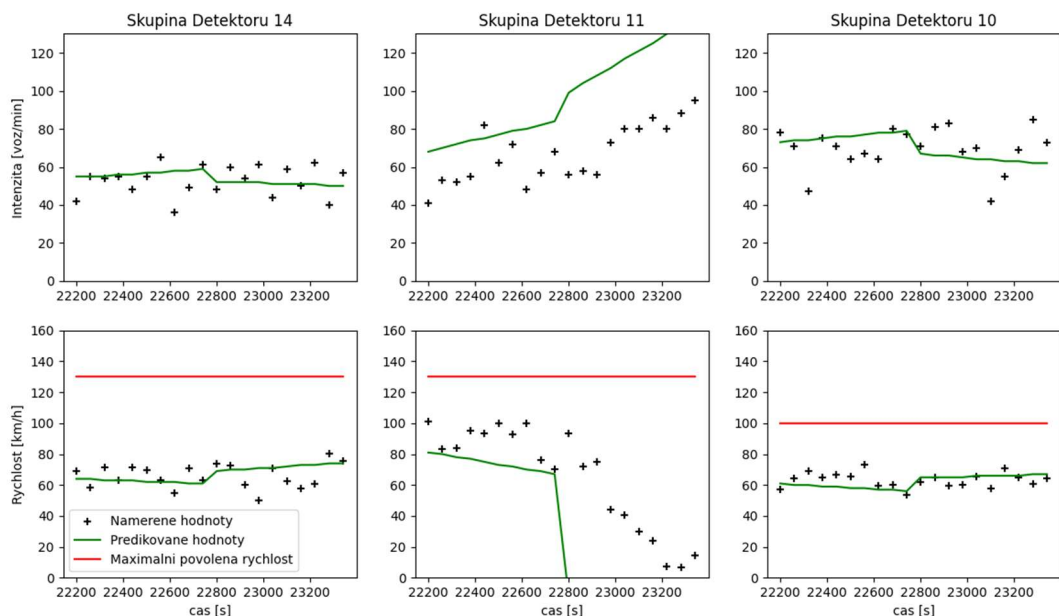
z naměřených dat určí stoupající či klesající trend. Tento trend bude určován vždy ze stejně dlouhého časového úseku jako je délka časového úseku, jenž má být algoritmem predikován. Obecná rovnice lineární regrese je znázorněna níže.

$$y = \beta_0 + \beta_1 * x_1$$

Hodnotou x je v rámci implementace vždy volena hodnota času, která stále lineárně narůstá. Hodnotou y je v implementaci volena intenzita. Koeficient β_1 pak značí její nárůst, či pokles. Pomocí tohoto koeficientu jsou následně určeny hodnoty intenzit v predikovaném okamžiku. Predikované hodnoty intenzit jsou následně předány dopravnímu modelu, který nad nimi provede simulaci.

Z Obrázku 46 je patrné, že spočtené koeficient lineární regrese relativně uspokojivě vystihují budoucí možný trend intenzity u jednotlivých skupin detektorů. Nejlépe je to vidět u skupiny detektoru 11, kde je razantní pokles rychlosti, jejíž pokles je relativně výstižně vyjádřen příslušnou přímkou značící spočtené lineární koeficienty.

Vystupy detektoru pri velke zatezi a zobrazeni vypocteného trendu



Obrázek 45: Spočtené hodnoty koeficientů při vysoké zátěži

Určení excesu bylo v rámci ověření detekováno v modelu kombinací obsazenosti jednotlivých hran simulace a čekací doby. Pokud se tyto hodnoty dostaly na konci simulace nad definovanou mez, dopravní model usoudí, že tyto hrany jsou v kongesci a vrátí je zpět modulu řízení dopravy, aby nad nimi mohl vykonat řídicí signály.

Výstupy dopravního modelu jsou následně do simulace reality vloženy formou povinných rychlostních omezení. Tím jsou zajištěné ideální podmínky pro ověření algoritmu. Zdrojové kódy obsahující implementaci ověřovacího procesu jsou dodány v příloze této práce s příponou „zdrojove-kody“.

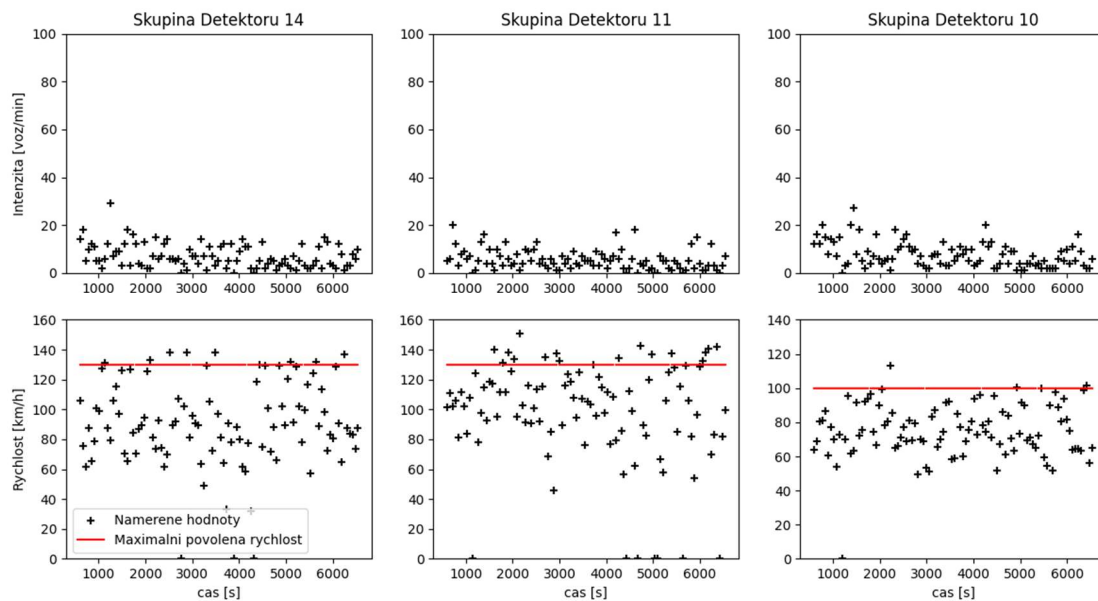
5.3 Ověření navrženého algoritmu

Jak bylo popsáno v podkapitole 5.3 v rámci ověření navrženého algoritmu budou simulovány dvě situace. První, kdy se dopravní proud nachází ve volném stavu a druhý, kdy se dopravní proud nachází ve stavu saturovaném. V obou situacích bude jedna simulace spuštěna bez implementace navrženého algoritmu a druhá ji naopak bude mít spuštěnou. V situaci, jež byla znázorněna v předchozí podkapitole, budou jednotlivé řezy charakterizovány detektorem, jehož naměřené hodnoty budou zobrazeny na následujících figurách. V nich budou vyobrazeny i maximální povolené rychlosti, které budou implementací navrženého algoritmu ovládány.

5.3.1 Situace ve volném dopravním proudu

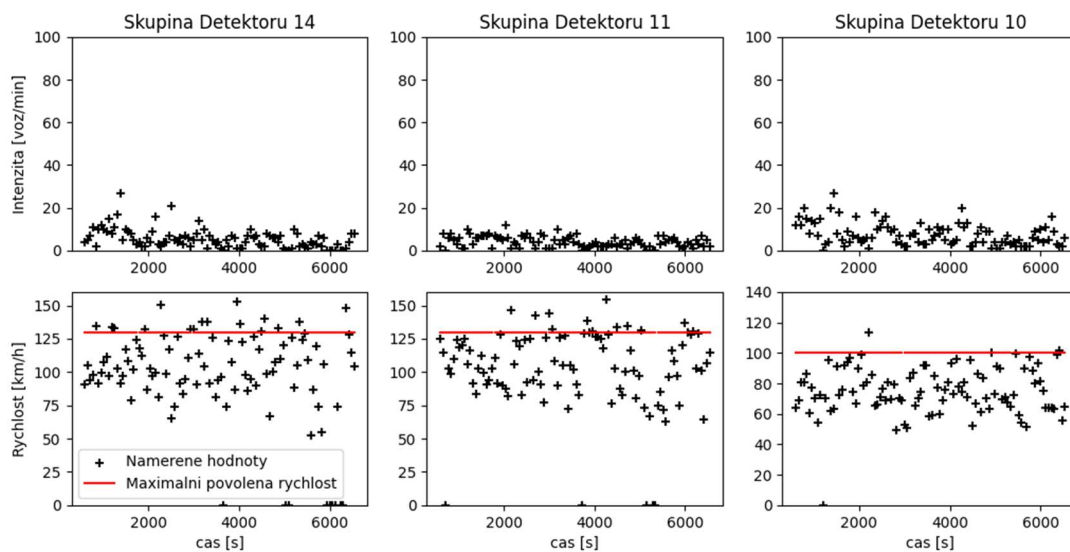
Pro simulování této situace byl zvolen časový horizont testovacích dat od půlnoci do dvou hodin ráno. V této době není pravděpodobné, že by hodnoty intenzit byly vysoké a lze tedy v této době požadované situace docílit. Předpoklady v rámci tohoto ověření jsou, že implementace navrženého algoritmu VLŘD nebude v tomto simulovaném stavu nijak zasahovat do řízené komunikace. Výstupy simulace při vypnutém řídicím algoritmu jsou vyobrazeny na obrázku 46 a výstupy při zapnutém řídicím algoritmu jsou vyobrazeny na obrázku 47.

Porovnání detektorů při nízké zátěži a vypnutým řídicím algoritmem



Obrázek 46: Hodnoty sledovaných detektorů při vypnuté implementaci ověřovaného algoritmu

Porovnání detektorů při nízké zátěži se spuštěným řídicím algoritmem

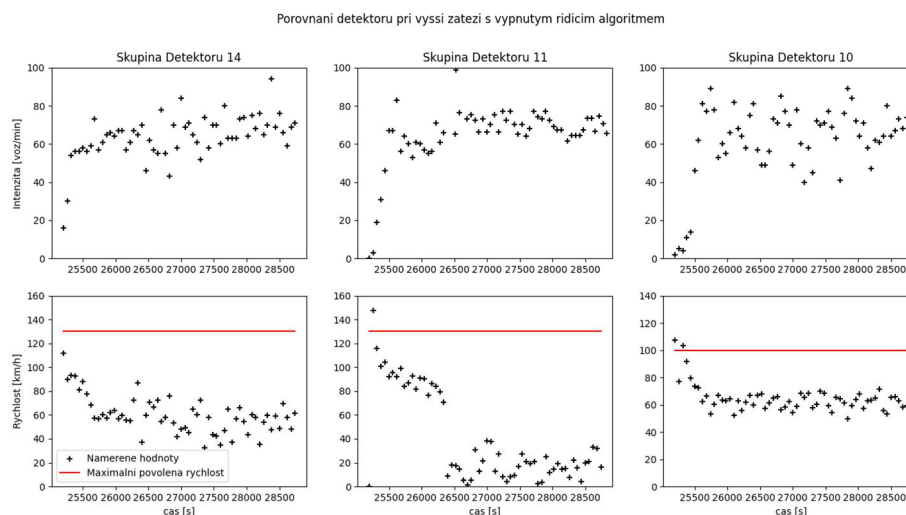


Obrázek 47: Hodnoty sledovaných detektorů při zapnuté implementaci ověřovaného algoritmu

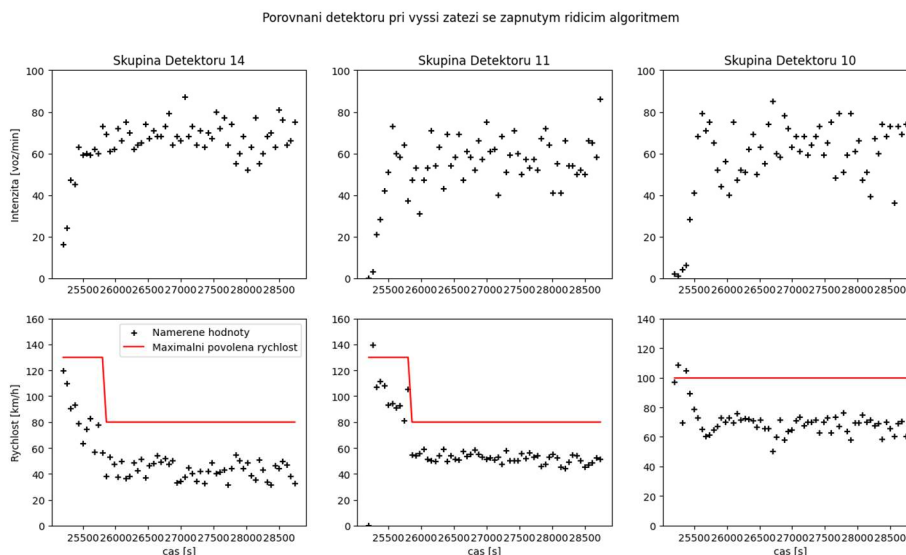
Z uvedených grafů je zřejmé, že algoritmus dospěl na základě výsledků z modulu dopravního modelu, že není nutné v tomto případě měnit maximální rychlostní omezení, protože dopravní proud se nachází ve volném stavu.

5.3.2 Situace při vyšší dopravní zátěži

Pro simulování tohoto stavu byl zvolen časový horizont od 7 hodiny do 8 hodiny ranní. V tomto časovém horizontu je pravděpodobnost vzniku excessu, který bude třeba řídit, vyšší. Proto by v této situaci měla implementace navrženého algoritmu snížit maximální rychlostní stupeň, aby se pokusila dopravnímu excessu předejít. Výsledky simulovaných situací jsou uvedeny na obrázku 48, jež představuje situaci, kdy je řídicí algoritmus vyřazen z provozu. Na obrázku 49 je vyobrazena situace, kdy je řídicí algoritmus v provozu v ranní špičce.



Obrázek 48: Znázorněná situace s vypnutím řídicím algoritmem



Obrázek 49: Znázorněná situace s řídicím algoritmem v provozu

Z vyobrazených grafů je vidět, že hned po prvním predikčním intervalu vydal algoritmus rychlostní omezení 80 km/h na hrany, jež byly součástí sledovaných hran. Při spuštění simulace s vypnutým řídicím algoritmem lze vidět na obrázku 48, že bez sníženého rychlostního omezení se v místě detektoru skupiny 11 razantně snížila rychlost, která se rozptýlila v intervalu rychlostí 0–40 km/h. Tato skutečnost ukazuje na vznik kongesce. V obrázku 49, kdy byl řídicí algoritmus zapnutý, lze sledovat, že vlivem nastaveného rychlostního omezení se dopravní proud jistým způsobem harmonizoval okolo hodnoty rychlosti 50 km/h. V této situaci by ovšem bylo vhodnější, kdyby algoritmus v místě skupiny detektorů 11 aktivoval rychlostní omezení na 60 km/h, nicméně jiná rychlostní omezení nebyla z důvodu časové náročnosti v rámci ověřovacího procesu implementována. Výsledek tohoto měření ale potvrzuje předpoklad, kdy algoritmus měl na základě dopravního modelu určit možnost vzniku kongesce a v místě jejího potencionálního vzniku vytvořit rychlostní omezení. Jak však již bylo zmíněno, ověřovací proces byl simulován v ideálních podmínkách, kdy dopravní model využívaný k predikci přesně odpovídal měřené realitě. Za těchto podmínek lze konstatovat, že návrh algoritmu je správný.

5.4 Zhodnocení výsledků ověřovacího procesu

Dle vyobrazení z podkapitol 5.3.1 a 5.3.2 lze usoudit, že v ideálních podmínkách je navržený algoritmus VLŘD vyhovující. V situaci, kdy bylo předpokládáno, že navržený algoritmus zareaguje pomocí výstupů rychlostních omezení, byl z hlediska ověřovacího postupu předpoklad naplněn. Jak již ale bylo zmíněno, ověřovací proces počítal ve velké většině s ideálními podmínkami. Užítý dopravní model pro predikci dopravní situace byl přesnou kopií reálného světa. Ve skutečném světě je dosažení takového stavu velmi komplexním problémem. V ověřovacím procesu byl predikční interval nastaven na 5 minut. Simulace tohoto intervalu trvala prostředí SUMO cca 2-3 minuty na průměrně dimenzované výpočetní technice. Doba výpočtu není z hlediska řízení komunikace příliš optimální. Při delší době výpočtu může začít docházet k prodlevám mezi změnami jednotlivých řídicích systémů, což může způsobit jistou míru nejistoty. Nicméně jak bylo popsáno v kapitole 4, tento nedostatek je možné vyřešit zavedením kontrolní hodnoty času či dostatečným dimenzováním výpočetního HW. Výsledná hodnota doby výpočtu ale také značí, že při zvětšující se dopravní síti, jež má být algoritmem řízena, se nároky na výpočetní techniku zvýší. Samotná optimalizace řídicího algoritmu by ale také mohla snížit dobu výpočtu, jelikož implementace navrženého algoritmu rozhodně není optimální vzhledem k omezeným programovacím znalostem.

Obecně ale lze tvrdit, že způsob řízení dálniční komunikace pomocí algoritmu založeného na teorii řízení, jež predikuje dopravní situaci na základě dopravního modelu, je způsob, který by mohl do budoucna fungovat. Z hlediska virtuálního LŘD je tento způsob řízení velmi vhodný díky možnosti generovat rychlostní omezení kdekoliv na sledované dopravní síti.

Kapitola 6

Další směry vývoje v dané oblasti

Směrů vývoje v dané oblasti je několik. Mezi první lze zařadit ověření návrhu algoritmu v situaci, která více odpovídá realitě. Tedy využít dynamicky měnící se data jako vstupní data a analyzovat chování algoritmu. Na tento směr navazuje další možnost vývoje což je modul dopravního modelu. Dalším vývojem by určitě mohla být detailnější analýza matematických dopravních modelů, jež by nahradily nástroj SUMO, který je vhodný na jednoduchou implementaci a prvotní ověření navrženého algoritmu, nicméně na provozní podmínky již tolik vhodný být nemusí. V rámci tohoto směru by byla také zajímavá možnost navrhnutí nového modelu. Tento směr nicméně může být vývojově náročný.

Dalším směrem vývoje může být zaměření se na oblast určení trendů měřených dopravních veličin. Lineární regrese použitá v rámci ověření sice nastiňovala dobré výsledky, nicméně jedná se o jednoduchý model, který nemusí být přesný za všech okolností, které mohou na sledované komunikaci nastat. Nabízí se i možnost kdy na základě porovnání koeficientů trendu v jednotlivých časových okamžicích se bude algoritmus rozhodovat, zda simulaci spustit či nikoliv. Například, pokud by koeficient trendu vycházel klesající v obou přechozích časových intervalech, tak simulace spuštěna nebude. Tím by mohl být ušetřen výpočetní výkon a i efektivita, jelikož simulace by se nemusela provádět v každém časovém kroku.

Dalším možným směrem vývoje je zaměření se na harmonizaci dopravního proudu pomocí algoritmu VLŘD. Díky virtuálnímu prostředí umožňuje systém VLŘD více možností, jak dopravní proud harmonizovat díky možnosti vygenerovat rychlostní omezení v podstatě kamkoliv do řízené sítě.

Směr dalšího vývoje se odvíjí i ve směru validace modulu dopravního modelu. V tomto oboru je možné se více zaměřit na větší využití naměřených hodnot TT, které mohou krom jiného sloužit právě k validaci modulu dopravního modelu.

Nejzajímavější směr dalšího vývoje směřuje již více do budoucnosti. V době autonomních vozidel nabízí algoritmus VLŘD velké možnosti v oblasti řízení dopravního proudu. Za předpokladu že budou autonomní vozidla schopna přijímat výstupy algoritmu, je možné takto relativně jednoduše homogenizovat dopravní proud. Zároveň by bylo možné dopravní proud pomocí systému VLŘD v případě potřeby přeměrovat. V oblasti autonomních vozidel je zaručeno, že budou plně respektovat výstupy algoritmu VLŘD, proto by efektivita systému VLŘD i systému LŘD měla obecně razantně vzrůst.

Závěr

Cílem práce byla rešerše aktuálně využívaných i nevyužívaných algoritmů systémů LŘD, jejich implementace, analýza vhodných datových zdrojů, analýza vhodných simulačních SW pro ověření návrhu adaptivního algoritmu VLŘD a pro využití v rámci implementace navrženého algoritmu, návrh adaptivního algoritmu VLŘD a jeho následné ověření.

V rámci prvního bodu bylo zjištěno, že existují tři dohledané způsoby, jak algoritmy systému LŘD navrhnout. Prvním z nich byly algoritmy založené na pravidlech. Tento typ algoritmů využívá k určení výstupních hodnot předem definované konstanty měřených dopravních parametrů a na základě jejich porovnání s aktuálně naměřenými hodnotami rozhoduje, které rychlostní omezení zvolit. Dalším typem algoritmů jsou algoritmy založené na teorii fuzzy množin. Tyto typy algoritmů fungují podobně jako algoritmy založené na pravidlech, ale hraniční podmínky nejsou vyjadřovány číselnou hodnotou nýbrž fuzzy množinou. Výhodou těchto algoritmů je lepší přehlednost, a dle citované studie i dosahování lepších výsledků při řízení komunikace, než u algoritmů založených na pravidlech [8]. Posledním typem algoritmů, jenž byl v rámci rešerše popsán jsou algoritmy založené na řídicí teorii. Tento typ algoritmů využívá k řízení silničního úseku dopravní model, jenž predikuje budoucí dopravní situaci. Tento přístup k návrhu algoritmu byl zvolen pro tuto práci.

Cílem analýzy využitelných datových zdrojů pro algoritmus VLŘD bylo shrnutí možných datových zdrojů, jež by byly využitelné v rámci návrhu adaptivního algoritmu VLŘD. Vedle známých metod detekce, jako jsou indukční smyčky či FCD sběr dat, byly analyzovány také metody detekce pomocí C-ITS technologie a Bt/Wifi technologie. Jako nejvhodnější datový zdroj pro algoritmus VLŘD byl zvolen, dle provedené analýzy, zdroj z indukčních smyček. Tento datový zdroj dokáže nejspolehlivěji detekovat hodnoty intenzit dopravního proudu. Ostatní analyzované metody detekce jsou schopny detekovat dojezdové doby, jež mohou být v rámci algoritmu VLŘD využity pro ověření modulu dopravního modelu také pro zpřesnění detekce.

Z analýzy vhodných simulačních SW pro užití v rámci návrhu adaptivního algoritmu VLŘD byl jako nejvhodnější zvolen nástroj SUMO. Jeho výhody oproti ostatním simulačním SW, jež byly v rámci práce analyzovány, jsou volně přístupné knihovny funkcí, které umožňují simulační SW ovládat pomocí rozhraní programovacích jazyků a jeho cena, která je díky licenci „Open Source“ nulová. Další výhodou je jeho obsáhlá dokumentace, kde je zdokumentována většina funkcí a postupů jak nástroj využívat [28]. Nevýhodou tohoto nástroje může naopak být méně intuitivní GUI, nicméně to bylo v rámci práce využito pouze pro pozorovací operace, nikoliv pro ovládání simulace.

Samotný návrh algoritmu VLŘD, představený v této práci, vychází z metody návrhu algoritmu založené na teorii řízení, jež byla analyzována v 1. kapitole. Návrh algoritmu byl dle identit prvků rozdělen na tři části. První částí je vstupní část, která zahrnuje všechny vstupní prvky vstupující do navrhovaného algoritmu. Druhou částí byla zvolena část s názvem MPC, jež obsahuje všechny prvky nutné pro řízení definované silniční komunikace. Třetí zvolenou částí jsou výstupní prvky navrhovaného algoritmu. Predikce je v návrhu algoritmu zajištěna modulem dopravního modelu, jehož součástí je dopravní model odpovídající skutečnosti. Na základě výsledků tohoto modelu jsou generována rychlostní omezení, která jsou posléze účastníkům silničního provozu předána formou IVI zprávy. Vstupní prvky, stejně jako výstupní, jsou v rámci návrhu algoritmu VLŘD variabilní. Jinými slovy, algoritmus byl navržen tak, aby bylo možné jeho jednotlivé vstupní prvky relativně jednoduše přidávat či odebrat. Pro případnou změnu vstupních nebo výstupních prvků by mělo stačit pozměnit konfiguraci příslušných modulů.

Ověření navrženého algoritmu bylo provedeno v ideálních podmínkách a spíše než efektivita řízení byla v rámci ověřovacího procesu ověřena schopnost algoritmu správně reagovat na různé situace. Tento předpoklad byl při ověřovacím procesu potvrzen. Nicméně vzhledem k povaze ověřovacího procesu by prvním dalším krokem v rozvoji tohoto návrhu mělo být ověření efektivitivy řízení. Další oblastí možného rozvoje je zaměření na logiku harmonizace dopravy, která ve virtuálním světě přináší nové možnosti.

Z hlediska autora je zpracované téma zajímavou oblastí, kterou je vhodné se zabývat do budoucna. Sice v aktuální době nenalezne systém VLŘD takové uplatnění, nicméně v době budoucí, při úvaze autonomních vozidel je uplatnění mnohem vyšší. Systém VLŘD by se v tomto kontextu mohl stát nedocenitelným nástrojem pro řízení dopravního proudu. Nicméně cesta k tomuto stavu je dlouhá a samotný návrh algoritmu, jež byl popsán v této práci, bude muset být v rámci této cesty dále optimalizován a rozvíjen. Velké možnosti se naskýtají v oblasti harmonizace dopravního toku nebo v analýze vhodnějších zástupců pro modul dopravního modelu. Zda byl takto navrhovaný algoritmus vhodně navržen do reálného provozu, je možné zjistit pouze na základě komplexních testů v modelované verzi skutečného provozu. I tato oblast by mohla být zajímavá pro další rozvoj.

Citovaná literatura

1. Beneš, Jiří. *Inteligentní metody řízení dopravy na líniových komunikacích*. Praha : České vysoké učení technické v Praze, 2013.
2. D1 Líniové řízení dopravy, SOKP. *Část B: Provozní řád*. Praha : Eltodo a.s., 2013.
3. Ředitelství silnic a dálnic České republiky. *Celostátní sčítání dopravy 2010*. [Online] ŘSD ČR, 2010. [Citace: 4. 4 2022.] <http://scitani2010.rsd.cz>.
4. Ředitelství silnic a dálnic České republiky. *Celostátní sčítání dopravy 2016*. [Online] ŘSD ČR, 2016. [Citace: 10. 4 2022.] <http://scitani2016.rsd.cz>.
5. Doležalová, Štěpánka. Centrum dopravního výzkumu. *Jaký vliv měla pandemie COVID-19 na silniční nákladní a veřejnou dopravu?* [Online] 18. 11 2021. [Citace: 3. 3 2022.] <https://www.cdv.cz/tisk/jaky-vliv-mela-pandemie-covid-19-na-silnicni-nakladni-a-verejnou-dopravu/>.
6. Dopravní info. *Líniové řízení provozu*. [Online] [Citace: 12. 3 2022.] <http://portal.dopravniinfo.cz/telematicke-aplikace/liniove-rizeni-provozu>.
7. Grumert, E.F., Tapani, A. a Ma, X. Characteristics of variable speed limit systems. [Online] 31. 5 2018. [Citace: 15. 3 2022.] <https://doi.org/10.1186/s12544-018-0294-8>.
8. Li, Duo a Ranjitkar, Prakash. A fuzzy logic-based variable speed limit controller. [Online] 2015. [Citace: 12. 4 2022.] <https://onlinelibrary.wiley.com/doi/epdf/10.1002/atr.1320>.
9. Allaby, P., Hellings, B. a Bullock, M. *Variable Speed Limits: Safety and Operational Impacts of a Candidate Control Strategy for Freeway Applications*. [Online] IEEE Transactions on Intelligent Transportation Systems, 3 2007. [Citace: 12. 3 2022.] <https://ieeexplore.ieee.org/document/4382934>.
10. National highways. *red x*. [Online] [Citace: 12. 4 2022.] <https://nationalhighways.co.uk/road-safety/red-x/>.
11. gov.uk. *Motorways (253 to 274)*. [Online] [Citace: 25. 3 2022.] <https://www.gov.uk/guidance/the-highway-code/motorways-253-to-273#rule258>.
12. Office of operations. *CHAPTER 3. VARIABLE SPEED LIMIT STATE OF THE PRACTICE*. [Online] U.S. Department of Transportation. [Citace: 26. 3 2022.] <https://ops.fhwa.dot.gov/publications/fhwahop17003/ch3.htm>.
13. Trafikverket. *The road to ITS*. [Online] [Citace: 27. 3 2022.] https://trafikverket.ineko.se/Files/sv-SE/11280/RelatedFiles/2011_145_The_Road_%20to ITS.pdf.

14. Bast. *Traffic management systems on Federal Highways*. [Online] Federal highways research institute. [Citace: 1. 4 2022.] https://www.bast.de/EN/Traffic_Engineering/Subjects/traffic-management-systems.html.
15. Votruba, Zdeněk, Klečáková, Jana a Kalika, Marek. *Systémová analýza*. Praha : České vysoké učení technické, 2008. 9788001040812.
16. Hegyi, A., De Schutter, Bart a Hellendoorn, J. IEEE Explore. *IEEE Transactions on Intelligent Transportation Systems*. [Online] 7. 3 2005. [Citace: 10. 4 2022.] <https://ieeexplore.ieee.org/document/1402434>.
17. Vacenovský, Martin. *Posouzení využití V2X technologií jako vhodného zdroje dopravních dat*. Praha : České vysoké učení technické v Praze, 2020.
18. Příbyl, Ondřej. *Detektory zasahující do vozovky: úvod do detekce*. [Online] [Citace: 26. 3 2022.] <https://zolutarev.fd.cvut.cz/mzd/ctrl.php?act=show,file,23843>.
19. Ječmen, Aleš. *Monitorování dopravního proudu plovoucími vozidly*. Praha : České vysoké učení technické v Praze, 2015.
20. Google maps. [Online] Google Inc. [Citace: 22. 7 2020.] <https://www.google.com/maps/>.
21. Bluetooth SIG, Inc. *Understanding Bluetooth Range*. [Online] [Citace: 12. 4 2022.] <https://www.bluetooth.com/learn-about-bluetooth/key-attributes/range/>.
22. OpenWeb. *How far will your Wifi signal reach?* [Online] [Citace: 12. 4 2022.] <https://openweb.co.za/how-far-will-your-wifi-signal-reach/>.
23. *C-ROADS: Specifikace systému*. Praha : autor neznámý, 2018.
24. Zdrojové kódy prostředí Mat-Sim. [Online] [Citace: 12. 3 2022.] <https://github.com/matsim-org/matsim-lib>.
25. Zdrojové kódy prostředí SUMO. [Online] Eclipse. [Citace: 1. 3 2022.] <https://github.com/eclipse/sumo>.
26. Vaniš, Miroslav. *Ověření mikroskopických modelů dopravy na reálných dopravních datech*. Praha : České vysoké učení technické v Praze, 2015.
27. P. A. Lopez et al. *Microscopic Traffic Simulation using SUMO*. [Online] 2018. [Citace: 10. 11 2021.] <https://ieeexplore.ieee.org/document/8569938>.
28. *SUMO Documentation*. [Online] [Citace: 5. 10 2021.] <https://sumo.dlr.de/docs/>.
29. Overpass turbo. [Online] [Citace: 14. 2 2022.] <https://overpass-turbo.eu>.
30. *OpenStreetMap.org*. [Online] [Citace: 14. 3 2022.] <https://www.openstreetmap.org/about>.
31. *Sumo publications*. [Online] [Citace: 12. 1 2022.] https://sumo.dlr.de/docs/Publications.html#krauss1998_1.

32. ETSI TS 102 637-2. *ntelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications;: Part 2: Specification of Cooperative Awareness Basic Service.* [Online] 2012. [Citace: 2. 7 2020.] https://www.etsi.org/deliver/etsi_ts/102600_102699/10263702/01.02.01_60/ts_10263702v010201p.pdf.

Seznam zkratek

| | |
|------|----------------------------------|
| ASD | Automatické sčítače dopravy |
| BO | Back-Office |
| CAM | Cooperative awareness message |
| FO | Front-End Office |
| GUI | Graphical User Interface |
| LŘD | Liniové řízení dopravy |
| OBU | On Board Unit |
| OSM | Open Street Maps |
| SUMO | Simulation of Urban MObility |
| ŘSD | Ředitelství silnic a dálnic |
| VLŘD | Virtuální liniové řízení dopravy |
| XML | Extensible Markup Language |

PŘÍLOHY