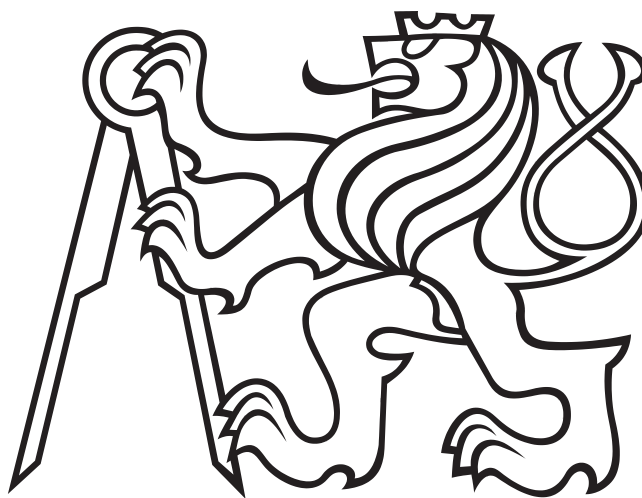


České vysoké učení technické v Praze

Fakulta strojní

Ústav přístrojové a řídicí techniky



Diplomová práce

**Identifikace kritických scénářů pro
testování vozidel s vyšší úrovní
automatizace**

Bc. Denisa Kaisrlíková

2024



ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Kaisrlíková** Jméno: **Denisa** Osobní číslo: **493566**
Fakulta/ústav: **Fakulta strojní**
Zadávající katedra/ústav: **Ústav přístrojové a řídicí techniky**
Studijní program: **Automatizační a přístrojová technika**
Specializace: **Automatizace a průmyslová informatika**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Návrh scénářů pro testování vozidel s vyšší úrovní automatizace

Název diplomové práce anglicky:

Design of Scenarios for Testing Vehicles with Higher Levels of Automation

Pokyny pro vypracování:

1. Proveďte rešerši metrik pro určení kritičnosti scénářů při testování vozidel
2. Proveďte rešerši legislativy v souvislosti s automatizovanými systémy (předpis OSN č.157 – ALKS a EU 2022/1426)
3. Proveďte rešerši vhodných nástrojů pro implementaci testování
4. Navrhněte scénáře k otestování.
5. Implementujte navržené scénáře vybranými nástroji a vyhodnoťte je.
6. Sestavte vyhodnocovací metriku dle dostupných nástrojů
7. Porovnejte získané výsledky s dostupnými výsledky a proveďte diskusi

Seznam doporučené literatury:

1. Proposal for the 01 series of amendments to UN Regulation No. 157 (Automated Lane Keeping Systems), 2023. 2nd ed. EU: United Nations,
2. COMMISSION IMPLEMENTING REGULATION (EU) 2022/1426 [online], 2022. In: EUR-LEX, PART 1 [cit. 2024-04-04]. Dostupné z: https://eur-lex.europa.eu/eli/reg_impl/2022/1426/oj
3. Konstantinos, M., at.al., 2022. Driver models for the definition of safety requirements of automated vehicles in international regulations. Application to motorway driving conditions [online]. In: . Brusel: EUR-LEX, s. 2-14 [cit. 2024-04-04]. Dostupné z: doi:<https://doi.org/10.1016/j.aap.2022.106743>

Jméno a pracoviště vedoucí(ho) diplomové práce:

Ing. Michal Kuchař U12110.3

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Ing. Martin Vítoušek odbor automatického řízení FS

Datum zadání diplomové práce: **26.04.2024** Termín odevzdání diplomové práce: **31.05.2024**

Platnost zadání diplomové práce: _____

Ing. Michal Kuchař
podpis vedoucí(ho) práce

prof. Ing. Tomáš Vyhřídál, Ph.D.
podpis vedoucí(ho) ústavu/katedry

doc. Ing. Miroslav Španiel, CSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomantka bere na vědomí, že je povinna vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

Datum převzetí zadání

Podpis studentky

Prohlášení

Prohlašuji, že jsem předloženou práci vypracovala samostatně a že jsem uvedla veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací. Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mé práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou a veškeré jejich dokumentace (dále souhrnně jen "Dílo"), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, avšak pouze k nevýdělečným účelům. Toto oprávnění je časově, teritoriálně i množstevně neomezené.

Datum:

.....

Podpis

Poděkování

Ráda bych poděkovala vedoucímu práce za věcné připomínky při plnění cílů této diplomové práce. Dále velmi děkuji firmě TÜV SÜD za poskytnutí veškerých materiálů, konzultací a podpory při její implementaci. Bez této podpory by práce nemohla vzniknout. Nakonec bych chtěla poděkovat celé své rodině za podporu ve studiu, díky které mohla vzniknout i tato práce.

Abstrakt

V dnešní době s rostoucím množstvím vozidel roste i provoz na silnicích a tím se zvyšuje i riziko nehody. Vývojem pomocných asistentů lze toto riziko snížit. Zároveň je ale nutné provést jejich otestování. Práce je zaměřena na testování vyšší úrovně automatizace řízení vozidel. Jedná se o identifikaci, parametrizaci a následnou implementaci vybraných nebezpečných dopravních situací. Konkrétně: Zařazení vozidla před jedoucím vozidlo do jízdního pruhu a zpomalení vozidla v jízdním pruhu před dalším vozidlem. Ty jsou následně simulovány pro různé kombinace volitelných parametrů a faktorů. Výsledná data jsou pomocí metrik kritičnosti ohodnocena a analyzována. Díky těmto výsledkům bude možné odhalit potencionálně nebezpečné scénáře, na které je potřeba se zaměřit při testování daného systému a jeho uvedení do provozu. Práce je implementována v Matlab/Simulink.

Klíčová slova : Kritické scénáře, automobilita, testování, analýza, metriky kritičnosti, Cut-in

Abstract

In today's world, with the increasing number of vehicles, road traffic is also growing, which in turn raises the risk of accidents. The development of assistive systems can reduce this risk. However, it is also necessary to test these systems. This work is focused on testing higher levels of vehicle automation. It involves identifying, parameterizing, and subsequently implementing selected hazardous traffic situations. Specifically, this includes the insertion of a vehicle into the lane in front of another moving vehicle and the deceleration of a vehicle in the lane in front of another vehicle. These scenarios are then simulated for various combinations of optional parameters and factors. The resulting data are evaluated and analyzed using criticality metrics. Thanks to these results, it will be possible to identify potentially dangerous scenarios that need to be addressed during the testing of the system and its deployment. The work is implemented in Matlab/Simulink.

Key words : Critical scenarios, automobility, testing, analysis, criticality metrics, Cut-in

Obsah

1	Úvod	12
2	SAE International	13
3	Systém ALKS	14
3.1	Základní vlastnosti	14
3.1.1	Minimální vzdálenost mezi vozidly	14
3.1.2	Převod řízení a přechodová fáze	15
3.1.3	Rizika a manévry	15
3.1.4	Omezení funkce	15
	Dosah detekce	15
	Poruchy systému	16
3.1.5	Bezpečnost a predikce	16
3.1.6	Snímání objektů a rozpoznávání událostí	16
4	Testování systému	17
4.1	První výkonnostní model	17
4.1.1	Identifikace modelu	17
4.1.2	Implementace	18
4.1.3	Vyhodnocení	18
4.2	Druhý výkonnostní model	19
4.2.1	Identifikace modelu	19
4.2.2	Implementace	19
4.2.3	Vyhodnocení	20
	Příčná kontrola	20
	Podélná kontrola	20
5	Kritické scénáře	23
5.1	Scénář zpomalení	23
5.2	Scénář Cut-IN	24
6	Vybrané metriky kritičnosti scénáře	26
6.1	Time to colision TTC	26
6.2	Number of colision CR	27
6.3	Minimum safe distance violation MSDV	28
6.4	Minimum safe distance factor MSDF	30
7	Pomocný skript pro porovnání výsledků	32
7.1	Implementace	32
	Funkce resize	32
	Funkce detect_circles	32
	Funkce percentage	32
	Funkce disp_mask	33

8 Implementace scénářů a metrik	35
Automated driving system Toolbox	35
Vehicle dynamics Toolbox	35
8.1 Implementace pomocí Matlab	35
8.2 Implementace pomocí Simulinku	37
8.2.1 Hlavní skript	37
Vstupy	37
Main	37
Výstupy	38
8.2.2 Simulink model	38
Inicializace	38
Simulace	38
9 Vyhodnocení dat	40
9.1 Scénář zpomalení	40
9.1.1 Model řidiče 1	40
9.1.2 Model řidiče 2	42
9.2 Scénář Cut-IN	45
9.2.1 Model řidiče 1	45
9.2.2 Model řidiče 2	49
10 Vytvoření nové vyhodnocovací metody	53
11 Závěr	58
Obsah příloženého CD	62

Seznam obrázků

Obrázek 1	SAE úrovně automatizace [1]	13
Obrázek 2	Diagram rozhodovacího modelu řidiče 1 [2]	18
Obrázek 3	Diagram rozhodovacího modelu řidiče 2 [2]	19
Obrázek 4	Fuzzy metriky PFS a CFS [3]	22
Obrázek 5	Výchozí pozice aktéru scénáře Deceleration [2]	23
Obrázek 6	Definice akce scénáře Deceleration [2]	23
Obrázek 7	Počáteční podmínky scénáře Cut-in [2]	24
Obrázek 8	Průběh scénáře Cut-in [2]	24
Obrázek 9	Implementace metriky TTC	26
Obrázek 10	Výsledek metriky TTC pro scénář Zpomalení z pohledu řidiče 2	27
Obrázek 11	Implementace metriky MSDV	29
Obrázek 12	Výsledek metriky MSDV pro scénář Zpomalení z pohledu řidiče 2	29
Obrázek 13	Implementace metriky MSDF	30
Obrázek 14	Výsledek metriky MSDF pro scénář zpomalení z pohledu řidiče 2	31
Obrázek 15	Schéma struktury pomocného skriptu pro porovnání výsledků	32
Obrázek 16	Grafy výsledků s definovanou maskou	33
Obrázek 17	Výpis dat získaných při analýze grafů do konzole	34
Obrázek 18	Schéma implementace scénáře pomocí kódu v Matlabu	36
Obrázek 19	Schéma celého skriptu s využitím Simulinku	37
Obrázek 20	Schéma Simulink modelu	38
Obrázek 21	Výsledky simulací dle specifikace [2]	40
Obrázek 22	Výsledky implementace scénáře v Matlabu	40
Obrázek 23	Výsledky implementace scénáře v Simulinku	41
Obrázek 24	Výsledky simulací dle specifikace scénáře Zpomalení [2]	42
Obrázek 25	Výsledky implementace scénáře Zpomalení v Matlabu	43
Obrázek 26	Výsledky implementace scénáře Zpomalení v Simulinku	43
Obrázek 27	Výsledky simulací dle specifikace scénáře Cut-in [2]	45
Obrázek 28	Výsledky implementace scénáře Cut-in v Matlabu	45
Obrázek 29	Výsledky implementace scénáře Cut-in v Simulinku	46
Obrázek 30	Výsledky simulací dle specifikace scénáře Cut-in [2]	49
Obrázek 31	Výsledky implementace scénáře Cut-in v Matlabu	50
Obrázek 32	Výsledky implementace scénáře Cut-in v Simulinku	50
Obrázek 33	Rozhodovací algoritmus metriky	54
Obrázek 34	Výsledek nové metriky pro scénář Cut-in model řidiče 2 a rychlosti 130km/h a 100km/h	55
Obrázek 35	Identifikace hranic oblastí grafu	56
Obrázek 36	Identifikace průnikových oblastí grafu mírně rizikové a rizikové	56

Seznam tabulek

Tabulka 1	Tabulka faktorů použitých v modelu řidiče 1 [2]	18
Tabulka 2	Tabulka faktorů použitých v modelu řidiče 2 [2]	22
Tabulka 3	Procentuální zastoupení jednotlivých barev grafu	41
Tabulka 4	Procentuální zastoupení jednotlivých barev grafu při absenci pozadí . .	42
Tabulka 5	Procentuální zastoupení jednotlivých barev grafu	44
Tabulka 6	Procentuální zastoupení jednotlivých barev grafu při absenci pozadí . .	44
Tabulka 7	Procentuální zastoupení jednotlivých barev grafu	48
Tabulka 8	Procentuální zastoupení jednotlivých barev grafu při absenci pozadí . .	48
Tabulka 9	Procentuální zastoupení jednotlivých barev grafu	52
Tabulka 10	Procentuální zastoupení jednotlivých barev grafu při absenci pozadí . .	52

Seznam zdrojových kódů

1	Implementace metriky CS v Matlabu	28
---	---	----

Seznam zkratek a symbolů

d_{min}	minimální vzdálenost mezi vozidly [m]
v_{ALKS}	rychlost vozidla řízeného systémem ALKS [m/s]
t_{odstup}	minimální časový odstup mezi vozidlem se systémem a vozidlem před ním [s]
$dist_{lat}$	okamžitá boční vzdálenost mezi dvěma vozidly [m]
$dist_{lon}$	okamžitá podélná vzdálenost mezi dvěma vozidly [m]
$length_{ego}$	délka vozidla ALKS [m]
l_{cut-in}	délka Other vozidla [m]
$u_{cut-in,lat}$	okamžitá boční rychlost Other vozidla [m/s]
$u_{ego,lon}$	okamžitá podélná rychlost vozidla ALKS [m/s]
$u_{cut-in,lon}$	okamžitá podélná rychlost Other vozidla [m/s]
a_{ego}	okamžité zrychlení ALKS vozidla [m/s^2]
d_1	bezpečná vzdálenost při úplném zastavení dvou vozidel [m]
d_{safe}	hranice bezpečné vzdálenosti vypočítaná metrikou [m]
d_{unsafe}	hranice nebezpečné vzdálenosti vypočítaná metrikou [m]
$a_{max,ego}$	upravené okamžité zrychlení ALKS vozidla [m/s^2]
$u_{ego,lon,N}$	očekávaná podélná rychlost vozidla ALKS po reakční době [m/s]
d_{new}	očekávaná podélná změna vzdálenosti mezi vozidlem ALKS a Other vozidlem [m]
T	reakční doba vozidla ALKS [s]
$b_{ego,comf}$	komfortní zpomalení vozidla ALKS [m/s^2]
$b_{ego,max}$	maximální zpomalení vozidla ALKS [m/s^2]
$b_{cut-in,max}$	maximální zpomalení Other vozidla [m/s^2]
PFS	předvídatelná fuzzy náhradní bezpečnostní metrika
CFS	kritická fuzzy náhradní bezpečnostní metrika
$a_{max,ego}$	maximální zrychlení Ego vozidla [m/s^2]
$rel_{velocity}$	je relativní vzdálenost mezi vozidly [m]
$rel_{distance}$	je relativní rychlost mezi vozidly [m/s]
$m_{collision}$	je počet simulací, ve kterých nastala kolize [—]

$n_{simulation}$	je celkový počet simulací [—]
v_r	je rychlost zadního vozidla [m/s]
v_f	je rychlost předního vozidla [m/s]
r	je časová odezva [s]
a_{maxA}	je maximální akcelerace zadního vozidla [m/s^2]
a_{minDec}	je maximální zpomalení zadního vozidla [m/s^2]
a_{maxDec}	je maximální zpomalení předního vozidla [m/s^2]

1 Úvod

V dnešní době si mnoho lidí nedokáže představit žít bez osobního vozidla. Denně vozidlo využívá mnoho z nás k dopravě do práce, na nákup, či dopravení dětí do různých zařízení. O víkendu jej mnoho z nás využívá pro návštěvu rodiny či za účelem cestování. Pro mnoho lidí je jízda v autě takzvaně "denním chlebem" a tráví mnoho času na silnicích. Je tedy zřejmé, že každý, kdo vlastní osobní automobil přijde do kontaktu s dopravními komunikacemi. S rostoucím množstvím přihlášených vozidel roste i provoz. S tím se zvyšuje i riziko nehody a ze silnic se mohou stávat nebezpečná místa.

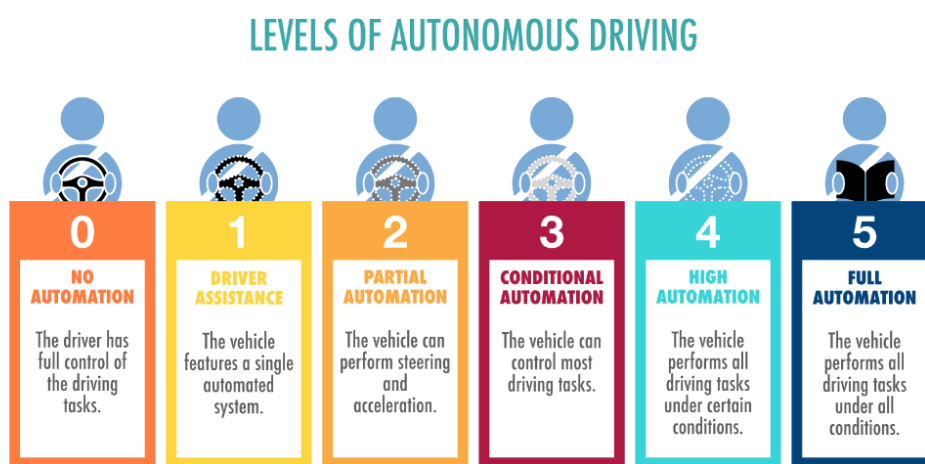
Vzhledem k tomu, že i já patřím do skupiny aktivních řidičů a hodně cestuji, vnímám toto riziko z vlastní zkušenosti. Ráda se podílím na testování a vývoji jakýchkoliv systémů, které pomohou zmírnit dopady tohoto rostoucího trendu a pomoci zvýšit komfort a bezpečnost všech účastníků dopravního provozu.

Ve své práci jsem se zaměřila na testování vyšší úrovně automatizace řízení vozidel. Jedná se o identifikaci, parametrizaci a následnou implementaci vybraných nebezpečných dopravních situací. Jmenovitě jsou to dvě dopravní situace: Zařazení před jedoucí vozidlo a zpomalení. Ty jsou následně simulovány pro různé kombinace volitelných parametrů a faktorů. Výsledná data jsou pomocí metrik kritičnosti ohodnoceny a analyzovány. Díky těmto výsledkům budou zřejmé potenciálně nebezpečné scénáře, na které je potřeba se zaměřit při testování daného systému pro automatizaci řízení vozidla.

Práce vznikla ve spolupráci s firmou TÜV SÜD, která mi poskytla veškeré odborné materiály, konzultace a vedla celý můj projekt.

2 SAE International

Asociace zabývající se technickými standardy a inovacemi v automobilovém průmyslu se nazývá SAE¹ International. Byla založena ve Spojených státech v roce 1905 Henrym Fordem a Andrewem Rickerem. Poskytuje výrobcům a inženýrům z celého světa různé technické normy a publikace, které mohou sloužit jako zdroje dat. Zaměřením této práce je automatizace řízení na vyšší úrovni. SAE rozděluje automatizaci řízení do několika úrovní. Označují se zkratkou: SAE Level "úroveň". Na obrázku: 1 jsou tyto úrovně definované. Systém, kterým se práce zabývá je řazen do úrovně SAE Level 3 s názvem ALKS²[4].



Obr. 1: SAE úrovně automatizace [1]

¹SAE - Společnost automobilových inženýrů

²ALKS - Automatický systém udržování v jízdním pruhu

3 Systém ALKS

Jízdní funkce ALKS neboli Automatický systém udržování v jízdním pruhu. Systém vznikl rozšířením a spojením dvou používaných jízdních asistentů ACC (adaptivního tempomatu) a LCA (asistenta pro centrování jízdy v jízdním pruhu). Principem ALKS je ovládat podélný i příčný pohyb vozidla tak, aby se nacházelo ideálně uprostřed jízdního pruhu a zároveň nedošlo ke kolizi s ostatními účastníky provozu. Sekvence těchto kroků je definována jako DDT³. Vozidlo je tedy primárně řízeno systémem bez intervence řidiče. Lze jej použít pro všechna vozidla třídy M⁴ i N⁵ [2].

3.1 Základní vlastnosti

Systém by měl být tvořen několika základními funkcemi a logickými sekvencemi, které jsou uvedeny níže. Jedná se o obecnou definici systému, platnou pro jeho bezpečný a z pohledu legislativy možný provoz. Lze se ovšem setkat i se státy, které tento systém neumožňují používat na jejich území dle této normy. Jedná se například o Čínu či Kanadu [2].

3.1.1 Minimální vzdálenost mezi vozidly

Systém musí být schopen zjistit vzdálenost k dalšímu vozidlu a na základě této informace přizpůsobit rychlost vozidla, aby nedošlo ke srážce. Vzdálenost musí být větší nebo rovna minimální požadované vzdálenosti mezi jedoucimi vozidly v daném jízdním pruhu. V případě nemožnosti dodržení mezery při působení vnějších podmínek, jako například zařazení nového vozidla do jízdního pruhu. V dalším ověřovacím cyklu musí být mezera upravena. Nelze-li bezpečného rozestupu dosáhnout lehkým zpomalením, musí dojít k aktivaci jednoho z bezpečnostních manévrů. Minimální vzdálenost je dána vzorcem:

$$d_{min} = v_{ALKS} \cdot t_{odstup} \quad (1)$$

kde:

- d_{min} = minimální vzdálenost mezi vozidly [m]
- v_{ALKS} = rychlost vozidla řízeného systémem ALKS [m/s]
- t_{odstup} = minimální časový odstup v sekundách mezi vozidlem řízeným systémem a vozidlem před ním [s]

[2]

³DDT - Dynamic Driving Task

⁴Kategorie M - motorová vozidla, která mají nejméně čtyři kola a používají se pro dopravu osob

⁵Kategorie N – motorová vozidla, která mají nejméně čtyři kola a používají se pro dopravu nákladů

3.1.2 Převod řízení a přechodová fáze

Požadavek na převod řízení z automatického pilota na manuální mód je definován souborem pravidel pro předání dynamické jízdy (DDT) ze systému na řidiče. Doba této akce se nazývá přechodová fáze. Situace vyžadující přechod řízení ze systému na řidiče mohou být plánované (např.: sjezd z dálnice) či neplánované (např.: zóna stavebních úprav komunikace). Po jejich identifikaci musí dojít k požadavku na přechod řízení, pokud možno s dostatečným časovým předstihem. Jakákoliv intervence řidiče musí mít vyšší prioritu a systém musí být přerušen/deaktivován. Několik možných situací známých z praxe jsou nahrány do paměti systému. Během celého přenosu řízení musí být systém zcela aktivní. Nedojde-li k jeho naplnění do 10s, systém musí aktivovat MRM⁶. Přechod tedy lze ukončit pouze řádnou deaktivací systému nebo pomocí MRM [2].

3.1.3 Rizika a manévry

Bezprostřední riziko srážky je definováno jako situace, která vede ke srážce vozidla s jiným účastníkem silničního provozu nebo překážkou, které se nelze vyhnout brzděním nižším než $5m/s^2$. Manévr s minimálním rizikem (MRM) je postup minimalizace rizik v provozu, který systém automaticky provádí po požadavku na přechod. Pokud-li například nastane situace, že řidič na předání řízení nereaguje, nastane-li vážná porucha ALKS systému či porucha vozidla. Tento manévr je omezen nejvyšším možným zpomalením $4m/s^2$ a vozidlo musí setrvat v daném jízdním pruhu. V případě nečitelnosti značení musí pokračovat po vhodně identifikované trajektorii. Manévr lze ukončit řádnou deaktivací. Druhým možným způsobem ukončení je zastavením vozidla, spuštěním varovných světel a vyčkáním na deaktivaci řidičem.

Emergency maneuver⁷ neboli nouzový manévr prováděný systémem v situaci, kdy je vozidlo vystaveno bezprostřednímu riziku kolize a jehož účelem je zabránit srážce nebo ji zmírnit. K jeho aktivaci musí dojít v případě, že je nutné aktivovat zpomalení přesahující $5m/s^2$. Manévr musí zabránit kolizi a být proveden tak, aby nedošlo k ohrožení bezpečnosti. Vozidlo nesmí vyjet mimo jízdní pruh, ve kterém se nachází. Manévr je praktikován prudkým zpomalením/brzděním. Vozidlo může být zpomaleno či zastaveno. Po dosažení bezpečného stavu vozidla, musí být systém opět aktivní a pokračovat v řízení. Je-li nutné nouzově zastavit a stát, systém musí aktivovat výstražná světla do 5s od aktivace manévru a udržovat je zapnuté, po celou dobu trvání manévru [2].

3.1.4 Omezení funkce

Dosah detekce je dán konkrétním senzorem a musí být specifikován. Je definován jako vzdálenost, při které systém bezpečně rozpozná cíl. Musí zohledňovat aktuální podmínky

⁶MRM - Manévr s minimálním rizikem

⁷EM - Nouzový manévr

prostředí. Do nich patří: změna poloměru zatáček nebo nepříznivé počasí [2].

Poruchy systému systém deaktivují či jej uvedou do omezeného režimu. Možnými příčinami je například: selhání ALKS, vážná porucha ALKS či vážná porucha vozidla. Z důvodu bezpečnosti musí systém neustále ověřovat selhání systému a rozsah detekce senzorů [2].

3.1.5 Bezpečnost a predikce

Systém nesmí ohrožit či vytvářet nebezpečné situace pro cestující či ostatní účastníky provozu. Musí být schopen předvídat nebezpečné situace a adekvátně reagovat. To znamená, že musí být schopen detekovat riziko srážky, zejména s jiným účastníkem silničního provozu. Jedná se především o vozidla jedoucí před řízeným vozidlem nebo vedle něj. Dá-li se srážce vyhnout, musí provést takové bezpečné kroky, aby došlo k její eliminaci. V případě nevyhnutelnosti kolize musí být schopen snížit důsledky. DDT musí splňovat dopravní předpisy dané země. Neustále musí být kontrolovány prvky vozidla, potřebná k řízení řidičem. Jsou to například: viditelnost přes čelní sklo (zamlžení), zapnutí světel, ovládání stěračů apod. Nesmí být ovlivnitelný elektrickým či magnetickým polem. Dále nesmí být systém zneužitelný řidičem. Například: aktivace systému současně s používáním entertainmentu. Systém musí být schopen udržet vozidlo ve stabilní poloze mezi jízdami pruhy [2].

3.1.6 Snímání objektů a rozpoznávání událostí

Vozidlo ALKS musí být vybaveno takovým snímacím systémem, aby určilo jízdní prostředí jako je geometrie vozovky před sebou, značení jízdnic pruhů apod. Dosah předních senzorů musí být minimálně 46m, boční rozsah musí být přes celý vedlejší jízdní pruh vlevo i vpravo. Výrobce musí definovat strategii rozpoznávání okolního prostředí a systém se dle něj musí umět přizpůsobit [2].

4 Testování systému

Testování systému je popsáno v předpisu UN R157 [2]. Zde jsou formovány veškeré zkoušky, kterými musí být daný systém otestován. Některé z nich je povinen provést výrobce a předložit jejich výsledky schvalovací instituci. Jedná se o testy konkrétního systému v konkrétním typu vozidla. Ostatní testy jsou obecné pro všechny typy vozidel a kontrolují obecně funkčnost systému.

Zde jsou předepsány hlavní řídicí funkce a strategie systému:

- udržení vozidla v jízdním pruhu,
- zabránění střetu s jiným účastníkem silničního provozu nebo předmětem blokujícím jízdní pruh,
- jízda za vozidlem jedoucím vpředu,
- vjetí jiného vozidla do jízdního pruhu,
- statická překážka poté, co vozidlo jedoucí vpředu změnilo jízdní pruh,
- zkouška vyhodnocení zorného pole.

Každá zkouška má definovaný závěr a minimální požadavky na její provedení. Tato práce se zabývá kritickými scénáři, při kterých systém nemusí být schopen zabránit kolizi. Jmenovitě kritický scénář Deceleration⁸ a Cut-in⁹. Jejich implementace a parametrizace je definována pomocí dvou výkonnostních modelů. Jedná se o algoritmus vnímání, vyhodnocení a následné reakce na vzniklou krizovou situaci [2].

4.1 První výkonnostní model

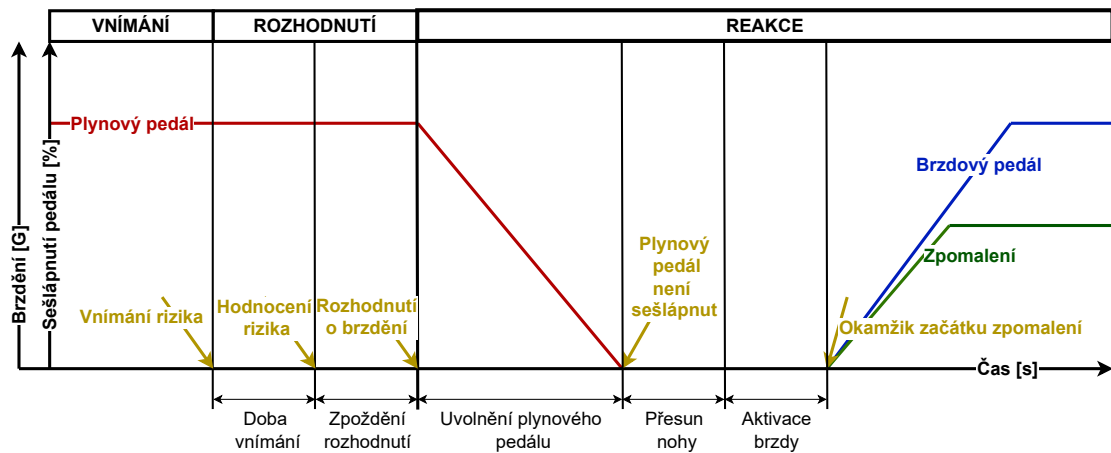
4.1.1 Identifikace modelu

První výkonnostní model je sestaven dle simulace zkušeného a pozorného řidiče. Je popsán pomocí schématu na obrázku: 2. Model je složen ze 3 hlavních částí: vnímání, rozhodnutí a reakce. Tyto hlavní stavy jsou zachyceny na časové ose systematicky za sebou. Na druhé ose je popsán stav plynového pedálu (zobrazen čarami červené barvy) a brzdového pedálu (zobrazen čarami modré barvy). Fáze vnímání zahrnuje detekci potencionálního nebezpečí a člověk se začíná připravovat na fázi rozhodování. V další fázi - rozhodnutí, dochází k identifikaci vzniklé situace. Fáze končí rozhodnutím se pro určitý sled kroků, kterými řidič zabrání nehodě. Tyto předchozí fáze se odehrávají v podvědomí řidiče. Poslední fáze je reakce. Jedná se o provedení kroků, kterými chce řidič zabránit kolizi. Jedná se o aktivní provedení jednotlivých kroků ze účelem zabránění kolize. Model je založen na základě ideálního řidiče. Reálně ovšem může reakci řidiče ovlivnit jeho aktuální fyzický či psychický

⁸Deceleration - Zpomalení vozidla

⁹Cut-in - vnoření vozidla do jízdního pruhu před jedoucím vozidlo

stav. Faktory ovlivňující rozhodovací schopnost řidiče jsou například: únava, nemoc či stres. V tabulce 1 jsou číselně uvedeny faktory použité v tomto modelu. Vozidlo se systémem je dále označováno jako *Ego vozidlo*¹⁰ [2].



Obr. 2: Diagram rozhodovacího modelu řidiče 1 [2]

Tab. 1: Tabulka faktorů použitých v modelu řidiče 1 [2]

Rizikové faktory		Hodnoty
Bod vnímání rizika	Změna jízdního pruhu	Odklon středu vozidla o více než 0,375m od středu jízdního pruhu
	Zpomalení	Rozdíl zpomalení <i>Ego vozidla</i> a jeho vzdálenost od následujícího vozidla
Čas hodnocení rizika		0.4 s
Doba trvání		0.75 s

4.1.2 Implementace

Vybraný scénář je po jeho identifikaci implementován dle daného použitého modelu řidiče. Implementace a simulace je provedena v programu Matlab/Simulink s využitím toolboxu Driving scenario designer. Z proběhnutých simulací jsou uložena data potřebná k vyhodnocení scénáře. V tomto modelu se jedná zejména o vzájemnou polohu vozidel.

4.1.3 Vyhodnocení

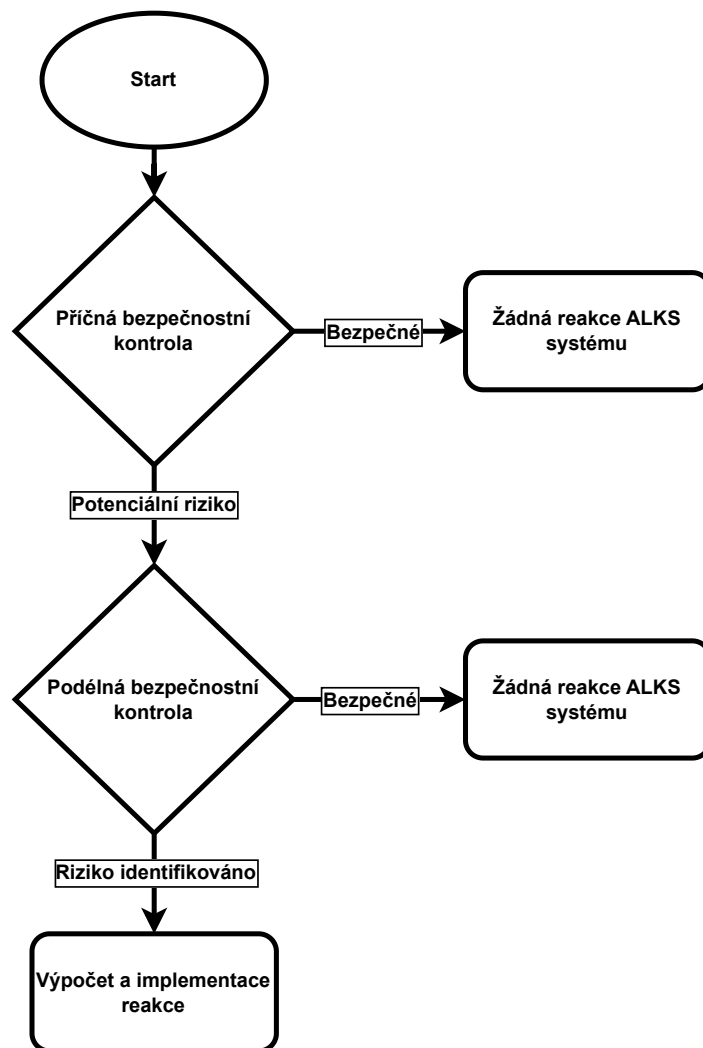
Data získaná z proběhlých simulací jsou vyhodnocena pomocí metrik kritičnosti. U tohoto modelu probíhá základní vyhodnocení jednoduchou podmínkou, zda došlo ke srážce či nikoliv. U složitějších scénářů může být podrobněji specifikováno místo srážky aktérů. Vyhodnocuje se opět vzájemná poloha obou aktérů během scénáře a také jejich vzájemná rychlost. Získané výsledky jsou shrnuty do přehledných grafů.

¹⁰Ego vozidlo - jiné označení vozidla řízeném řídicím systémem ve scénáři

4.2 Druhý výkonnostní model

4.2.1 Identifikace modelu

Druhý výkonnostní model je založen na modelu řidiče, který předvídá riziko kolize a je schopen dle toho přiměřeně brzdit. Na obrázku: 3 je diagram vyhodnocovacího algoritmu. Tento model je opět složen ze 3 hlavních částí: příčná bezpečnostní kontrola, podélná bezpečnostní kontrola a reakce. Tento diagram se opakuje v pomyslné smyčce. Dochází zde ke dvěma kontrolám okolí vozidla. Pokud jsou obě bezpečnostní kontroly vyhodnoceny kladně, systém se dostane do stavu kritické situace a provede bezpečnostní manévr [2].



Obr. 3: Diagram rozhodovacího modelu řidiče 2 [2]

4.2.2 Implementace

Implementace scénáře probíhá stejně jako v případě prvního modelu řidiče. Ze simulace jsou ukládána všechna důležitá data, která jsou následně využita k hodnocení scénáře pomocí metrik kritičnosti. Kromě polohy vozidel jsou v metrikách dále zahrnuty i jejich rychlosti, zrychlení nebo velikosti vozidel.

4.2.3 Vyhodnocení

Simulacemi získaná data jsou vyhodnocena ve dvou zmíněných směrech - příčně a podélně. Dle výsledků těchto hodnot je případně aplikován brzdný manévr.

Příčná kontrola

$$\frac{dist_{lat}}{u_{cut-in,lat}} < \frac{dist_{lon} + lenght_{ego} + l_{cut-in}}{u_{ego,lon} - u_{cut-in,lon}} + 0.1 \quad (2)$$

kde:

- $dist_{lat}$ = příčná vzdálenost mezi vozidly [m]
- $u_{cut-in,lat}$ = příčná rychlost *Other vozidla* [m/s]
- $dist_{lon}$ = podélná vzdálenost vozidel [m]
- $lenght_{ego}$ = délka *Ego vozidla* [m]
- l_{cut-in} = délka *Other vozidla* [m]
- $u_{ego,lon}$ = podélná rychlost *Ego vozidla* [m/s]
- $u_{cut-in,lon}$ = podélná rychlost *Other vozidla* [m/s]

Potencionální riziko kolize nastane, jsou-li splněny následující podmínky:

- a) Zadní část *Other vozidla*¹¹ se musí nacházet před *Ego vozidlem* ve smyslu podélného směru.
- b) *Other vozidlo* se pohybuje směrem k *ALKS vozidlu*¹².
- c) Podélná složka rychlosti *ALKS vozidla* je větší než podélná rychlost *Other vozidla*.
- d) Splnění rovnice 2.

[2]

Podélná kontrola spočívá v posouzení získaných dat pomocí dvou fuzzy kritických metrik PFS¹³ a CFS¹⁴. Potenciální riziko nebezpečí nastává ve chvíli, kdy je hodnota jedné z metrik větší než 0 [2].

Proactive Fuzzy Surrogate Safety Metric je definována následujícími rovnicemi:

¹¹Other vozidlo - „Další“ vozidlo, provádějící manévr ovlivňující vozidlo ALKS

¹²ALKS vozidlo - vozidlo řízené systémem ALKS

¹³PFS - Proactive Fuzzy Surrogate Safety Metric

¹⁴CFS - Critical Fuzzy Surrogate Safety Metric

$$PFS(dist_{lon}) = \begin{cases} 1 & \text{if } 0 < dist_{lon} - d_1 < d_{unsafe} \\ 0 & \text{if } dist_{lon} - d_1 > d_{safe} \\ \frac{dist_{lon} - d_{safe} - d_1}{d_{unsafe} - d_{safe}} & \text{if } d_{unsafe} < dist_{lon} - d_1 < d_{safe} \end{cases} \quad (3)$$

$$d_{safe} = u_{ego,lon}T + \frac{u_{ego,lon}^2}{2b_{ego,comf}} - \frac{u_{cut-in,lon}^2}{2b_{cut-in,max}} + d_1 \quad (4)$$

$$d_{unsafe} = u_{ego,lon} + \frac{u_{ego,lon}^2}{2b_{ego,max}} - \frac{u_{cut-in,lon}^2}{2b_{cut-in,max}} \quad (5)$$

[5]

kde:

- d_1 = bezpečná vzdálenost při úplném zastavení dvou vozidel [m]
- d_{safe} = hranice bezpečné vzdálenosti vypočítaná metrikou [m]
- d_{unsafe} = hranice nebezpečné vzdálenosti vypočítaná metrikou [m]

Critical Fuzzy Surrogate Safety Metric je definována následujícími rovnicemi [5]:

$$CFS(dist_{lon}) = \begin{cases} 1 & \text{if } 0 < dist_{lon} < d_{unsafe} \\ 0 & \text{if } dist_{lon} \geq d_{safe} \\ \frac{dist_{lon} - d_{safe}}{d_{unsafe} - d_{safe}} & \text{if } d_{unsafe} \leq dist_{lon} < d_{safe} \end{cases} \quad (6)$$

$$d_{safe} = \begin{cases} \frac{(u_{ego,lon} - u_{cut-in,lon})^2}{2a_{max,ego}} & \text{if } u_{ego,lon,N} \leq u_{cut-in,lon} \\ d_{new} + \frac{(u_{ego,lon,N} - u_{cut-in,lon})^2}{2b_{ego,comf}} & \text{if } u_{ego,lon,N} > u_{cut-in,lon} \end{cases} \quad (7)$$

$$d_{unsafe} = \begin{cases} \frac{(u_{ego,lon} - u_{cut-in,lon})^2}{2a_{max,ego}} & \text{if } u_{ego,lon,N} \leq u_{cut-in,lon} \\ d_{new} + \frac{(u_{ego,lon,N} - u_{cut-in,lon})^2}{2b_{ego,max}} & \text{if } u_{ego,lon,N} > u_{cut-in,lon} \end{cases} \quad (8)$$

$$a_{max,ego} = \max(a_{ego}, -b_{ego,comf}) \quad (9)$$

$$u_{ego,lon,N} = u_{ego,lon} + a_{max,ego} \quad (10)$$

$$d_{new} = \left(\frac{u_{ego,lon} + u_{ego,lon,N}}{2} - u_{cut-in,lon} \right) \quad (11)$$

kde:

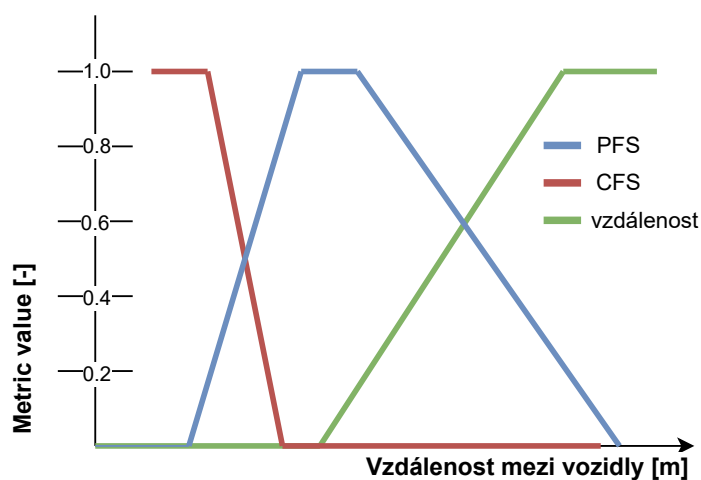
- $a_{max,ego}$ = maximální zrychlení *Ego vozidla* [m/s^2]
- $u_{ego,lon,N}$ = očekávaná podélná rychlost vozidla ALKS po reakční době [m/s]
- d_{new} = očekávaná podélná změna vzdálenosti mezi vozidlem ALKS a *Other vozidlem* [m]

Tabulka 2 obsahuje hodnoty parametrů pro model řidiče 2.

Tab. 2: Tabulka faktorů použitých v modelu řidiče 2 [2]

Rizikové faktory	Hodnoty
Doba vnímání rizika	Čas, kdy hodnota PFS nebo CFS není rovna 0
Reakční doba ALKS vozidla	$t = 0.75s$
Bezpečná vzdálenost vozidel po ukončení akce	$d_1 = 2m$
Komfortní zpomalení ALKS vozidla	$b_{egoComf} = 4m \cdot s^{-2}$
Maximální zpomalení ALKS vozidla	$b_{egoMax} = 6m \cdot s^{-2}$
Maximální zpomalení <i>Other vozidla</i>	$b_{cutInMax} = 7m \cdot s^{-2}$

Průběh hodnot metrik PFS a CFS lze vykreslit také graficky pomocí fuzzy množin. Jejich reprezentace pomocí fuzzy množin je na obrázku: 4. Zatímco CFS metrika je zaměřena na menší vzdálenosti mezi vozidly, PFS metrika se uplatňuje i při větších vzdálenostech. Konkrétní hodnoty hranic jsou dány kombinací parametrů scénáře.



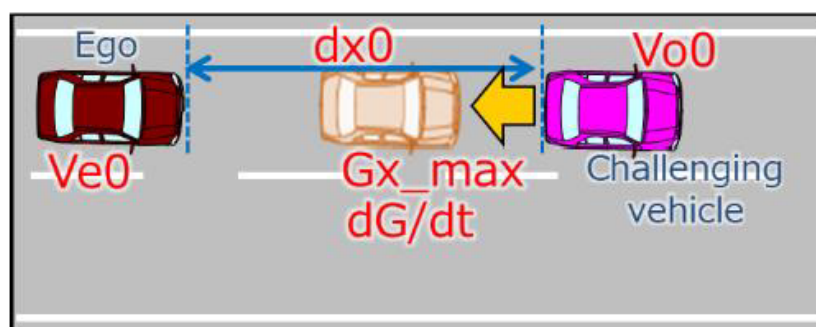
Obr. 4: Fuzzy metriky PFS a CFS [3]

5 Kritické scénáře

V předpise se vyskytují 3 základní scénáře, které testují funkčnost systému ALKS. V této práci jsou implementovány dva z nich. Třetí ze základních scénářů je spojením prvních dvou. Scénář zpomalení byl vybrán kvůli jeho jednoduchému pohybu obou aktérů. Lze na něm pochopit princip implementace a seznámení se s toolboxem¹⁵. Druhým scénářem byla zvolena situace Cut-in¹⁶, která je velmi rozmanitá.

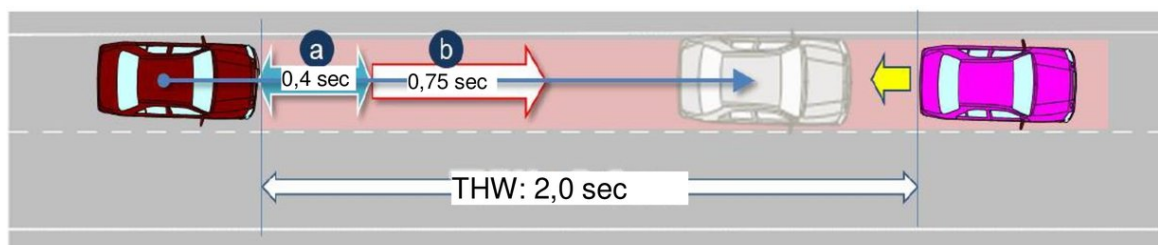
5.1 Scénář zpomalení

Tato situace je tvořena dvěma aktéry - *Ego vozidlem* a *Other vozidlem*. Dráha scénáře je složena ze dvou jízdních pruhů přímého směru. Počáteční pozice obou vozidel je definována podélným rozestupem vozidel a počátečními rychlostmi obou vozidel. Na obrázku: 5 je definován výchozí stav scénáře.



Obr. 5: Výchozí pozice aktéru scénáře Deceleration [2]

Obě vozidla se pohybují po přímé trajektorii ideálně uprostřed jízdního pruhu, jedoucí za sebou. Hlavní akcí scénáře je náhlé zpomalení *Other vozidla*. Následuje časová prodleva, způsobená reakcí na vzniklou situaci dle příslušného modelu řidiče. Poté následuje reakce *ALKS vozidla* na danou situaci a ukončení scénáře v koncových polohách vozidel.



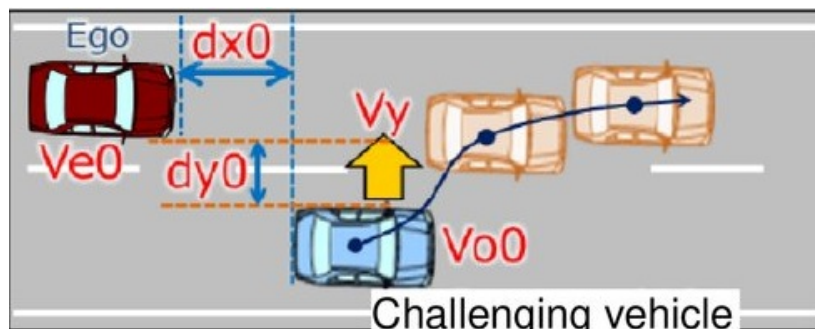
Obr. 6: Definice akce scénáře Deceleration [2]

¹⁵toolbox - rozšíření implementovaných funkcí programu Matlab/Simulink

¹⁶Cut-in - scénář, ve kterém se před jedoucím vozidlo v jízdním pruhu zařadí nové vozidlo

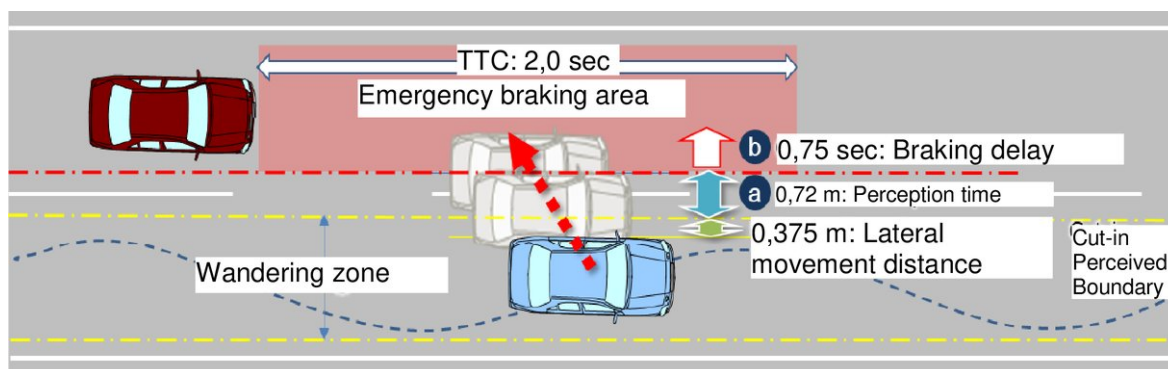
5.2 Scénář Cut-IN

Scénář obsahuje obdobně dva aktéry - vozidla označované *Ego* a *Other*. Dráha scénáře se skládá ze dvou jízdních pruhů přímého směru. Počátečními podmínkami je počáteční vzdálenost obou vozidel ve směru osy x a y , rychlost *Other* vozidla ve směru x a y , rychlost *Ego* vozidla. Situace je znázorněna na obrázku: 7.



Obr. 7: Počáteční podmínky scénáře Cut-in [2]

U tohoto scénáře je dále specifikována šířka jízdního pruhu $3.5m$, šířka modelu vozidla $1.9m$ a poloha vozidel v jízdních pruzích - jízda uprostřed jízdního pruhu. Jak už vyplývá z názvu scénáře, jedná se o situaci, kdy se *Other vozidlo* zařadí do jízdního pruhu před *Ego vozidlo* - zařazení vozidla. Přičemž jsou testovány různé kombinace rychlostí při různé vzájemné vzdálenosti ve směru jízdy.



Obr. 8: Průběh scénáře Cut-in [2]

V tomto scénáři se *Other vozidlo* pohybuje ve směru směru jízdy ale i příčně - dochází ke změně jízdního pruhu. Pohyb aktérů je zachycen na obrázku: 8. V jízdním pruhu je specifikována *Wandering zone*¹⁷, která vymezuje polohu vozidla mezi hranicemi jízdního pruhu, v situaci, kdy vozidlo nemění směr jízdy. Je dána šířkou vozidla a prostorem určeným k příčnému manévrování pro udržení vozidla uvnitř jízdního pruhu. Jakmile se vozidlo dostává na hranici *Wandering zone*, vstoupí do oblasti, ve které je možné očekávat cut-in

¹⁷Wandering zone - zóna v jízdním pruhu, ve které může vozidlo manévrovat aniž by bylo systémem detekována možnost změny pruhu *Other* vozidla

manévr. Vzdálenost a na obrázku: 8 odpovídá době rozpoznání manévru. Je to vzdálenost, kterou manévrující vozidlo urazí při identifikaci situace systémem *Ego vozidla*. Po ujetí této vzdálenosti, by mělo *Other vozidlo* být v zóně Emergency braking *Ego vozidla*. Její velikost je definována obdobně jako při scénáři zpomalení pomocí metriky TTC ve směru jízdy. V příčném směru je dána šířkou vozidla a šířkou určenou k manévrování v rámci jízdního pruhu. Poté je systémem implementována brzdná reakce. Čas zpoždění reakce je stanoven na 0.75 s. Scénář končí dokončením cut-in manévru *Other vozidlem*, pokud nedošlo během simulace ke kolizi aktérů.

6 Vybrané metriky kritičnosti scénáře

K vyhodnocení získaných dat se využívá mnoho metod. Běžně se lze setkat s vizuální kontrolou testera. Jeho úloha je v mnoha aplikacích nenahraditelná. Ovšem, existuje spousta funkcí - metrik, které mohou pomoci s celkovou identifikací krizových situací a zefektivnit jejich vyhodnocení. Následuje rozbor několika z nich, které jsou následně použity v této práci.

6.1 Time to colision TTC

Jedná se o metriku vyhodnocující čas, za který by došlo ke srážce při momentální vzdálenosti aktérů a jejich rychlostí. Pokud je metrika implementována na scénář zpomalení, je *Other vozidlo* považováno za statické a jeho rychlost je tedy nulová. Pokud jsou oba aktéři dynamičtí a není-li definováno jinak, jsou v metrice použity relativní hodnoty vzdálenosti a rychlosti mezi vozidly [6].

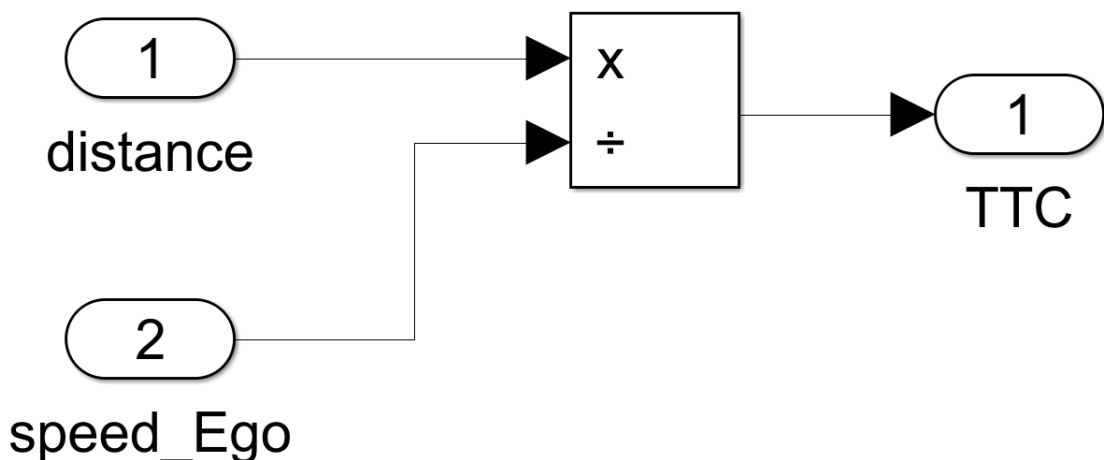
$$TTC_{dec} = dist_{lon}/u_{ego,lon} \quad (12)$$

$$TTC_{cutIN} = rel_{distance}/rel_{velocity} \quad (13)$$

kde:

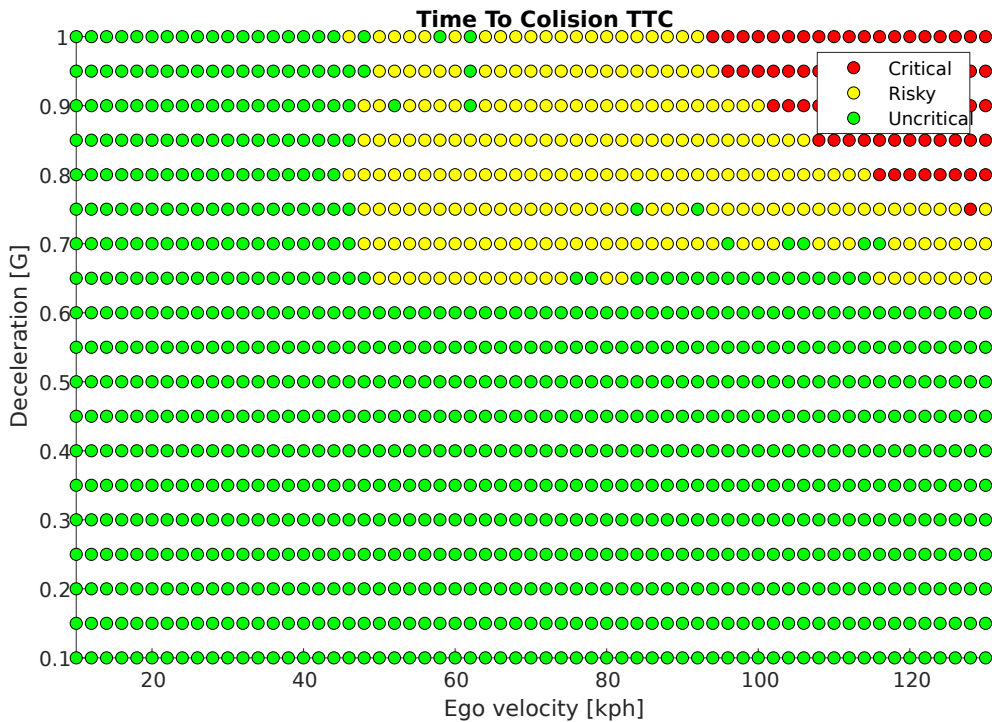
- $rel_{distance}$ = relativní vzdálenost mezi vozidly [m]
- $rel_{velocity}$ = relativní rychlost mezi vozidly [m/s]

Na obrázku: 9 je uvedena bloková implementace metriky v Simulinku.



Obr. 9: Implementace metriky TTC

Byla zvolena grafická implementace metriky s třemi stupni rizikovosti scénáře. Pokud je hodnota menší než 0,5s, scénář je označen červenou barvou jako kritický. Rizikové scénáře označené žlutou barvou jsou mezi hodnotami 0,5 – 1,5s. Situace, kdy lze jednoduše zabránit kolizi mají hodnotu vyšší než 1,5s a jsou označeny zelenou barvou. Výsledek metriky pro scénář zpomalení z pohledu řidiče 2 lze vidět na obrázku: 10.



Obr. 10: Výsledek metriky TTC pro scénář Zpomalení z pohledu řidiče 2

6.2 Number of collision CR

Metrika počtu kolizí udává procentuální zastoupení scénářů, ve kterých došlo ke kolizi ze všech provedených simulací. Kolizní scénáře jsou takové, ve kterých dojde k nehodě aktérů či ke kolizi s okolními objekty. Posuzuje schopnost daného systému zabránit kolizi. Jedná se o nástroj používaný v celkové analýze [6].

$$CR = (m_{collision} / n_{simulation}) \cdot 100 \quad (14)$$

kde:

- $m_{collision}$ = počet simulací, ve kterých nastala kolize [–]
- $n_{simulation}$ = celkový počet simulací [–]

Metrika byla implementována pomocí kódu v Matlabu, který lze vidět ve Zdrojovém kódu: 1. Jejím výstupem je vytisknutí výsledků do konzole ve tvaru: "Percentage of crash scenarios: **hodnota %**".

```

1 count_of_simulation = n;
2 crash_simulation = 0;
3 for i = 1:(n-1)
4     if distance_plot(i) < 0
5         crash_simulation = crash_simulation+1;
6     else
7         continue
8     end
9 end
10
11 CR = (crash_simulation / count_of_simulation) * 100;
12 disp([" Percentage of crash scenarios:", num2str(CR), "%"])

```

Zdrojový kód 1: Implementace metriky CS v Matlabu

6.3 Minimum safe distance violation MSDV

Vyhodnocení minimální bezpečné vzdálenosti je dáno několika parametry jako jsou jejich rychlosti, zrychlení a vzdálenosti v příčném i podélném směru. Okolo *Ego vozidla* je definována bezpečná zóna od ostatních objektů provozu. Tato oblast by měla splňovat pravidla modelu RSS¹⁸. Pokud během manévru *Ego vozidla* dojde ke vstupu objektu do této bezpečnostní zóny, metrika je vyhodnocena logickou hodnotou 1. Podélná velikost obálky je dána následující rovnicí [6]:

$$d_{longMin} = v_r r + \frac{1}{2} a_{maxA} r^2 + \frac{(v_r + r a_{maxA})^2}{2 a_{minDec}} - \frac{v_f^2}{2 a_{maxDec}} \quad (15)$$

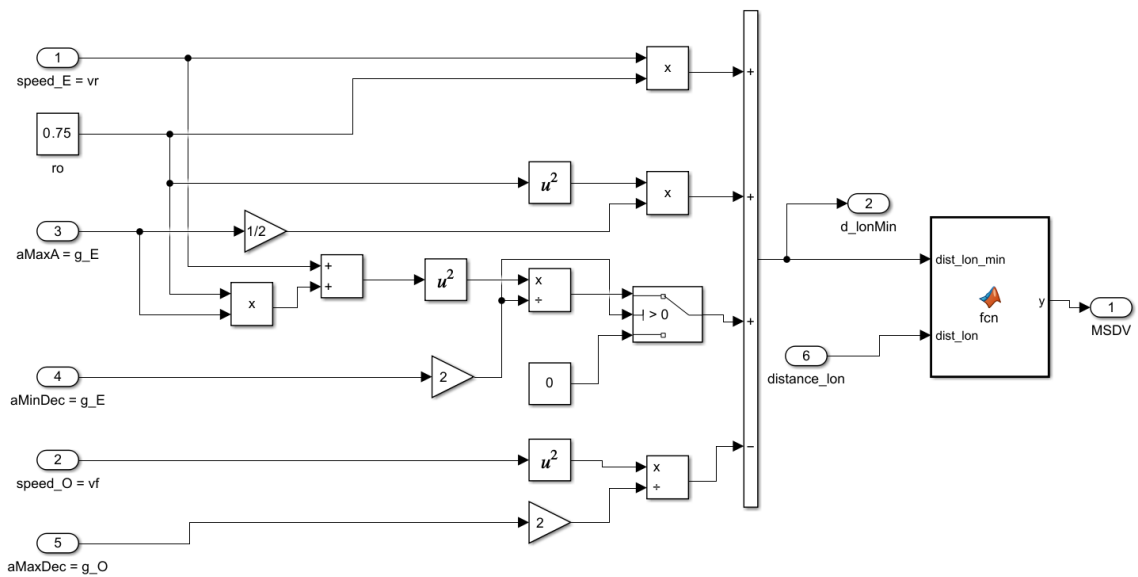
kde:

- v_r = rychlost zadního vozidla [m/s]
- r = časová odezva [s]
- v_f = rychlost předního vozidla [m/s]
- a_{maxA} = maximální akcelerace zadního vozidla [m/s^2]
- a_{minDec} = maximální zpomalení zadního vozidla [m/s^2]
- a_{maxDec} = maximální zpomalení předního vozidla [m/s^2]

$$MSDV = \begin{cases} 1 & \text{if } dist_{lon} < d_{longMin} \\ 0 & \text{else} \end{cases} \quad (16)$$

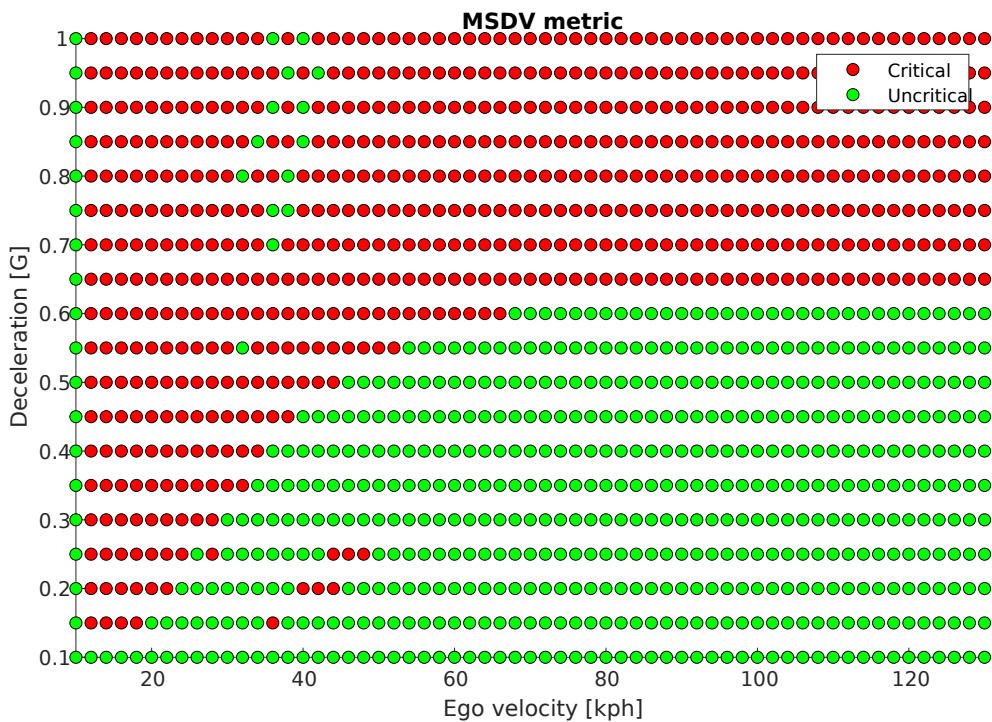
¹⁸RSS - Responsibility-sensitive safety

Na obrázku: 11 je uvedena bloková implementace metriky v Simulinku.



Obr. 11: Implementace metriky MSDV

Výsledky metriky jsou reprezentovány pomocí grafu, který obsahuje dva stupně rizikovosti scénáře. Jelikož je metrika binární, zelenou barvou jsou označeny kombinace parametrů, kdy nedojde k porušení obálky. Naopak červené body označují kombinaci parametrů, při nichž dojde k narušení bezpečnostní obálky - výstupem je logická 1. Výsledek lze vidět na obrázku: 12, pro scénář zpomalení s použitím modelu řidiče 2.



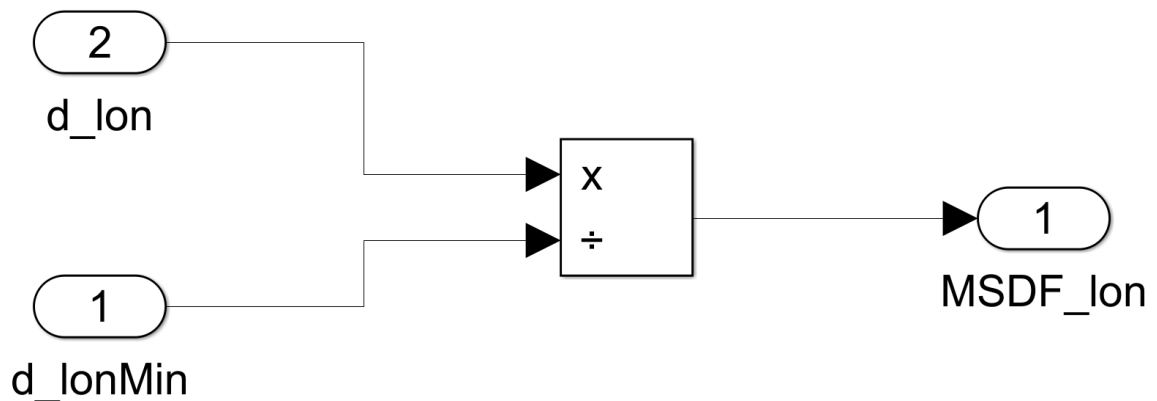
Obr. 12: Výsledek metriky MSDV pro scénář Zpomalení z pohledu řidiče 2

6.4 Minimum safe distance factor MSDF

Faktor minimální bezpečné vzdálenosti je dán naměřenou podélnou i příčnou vzdáleností k vypočítaným bezpečným vzdálenostem [6].

$$MSDF_{long} = \frac{dist_{lon}}{d_{longMin}} \quad (17)$$

Na obrázku: 13 je uvedena bloková implementace metriky v Simulinku.

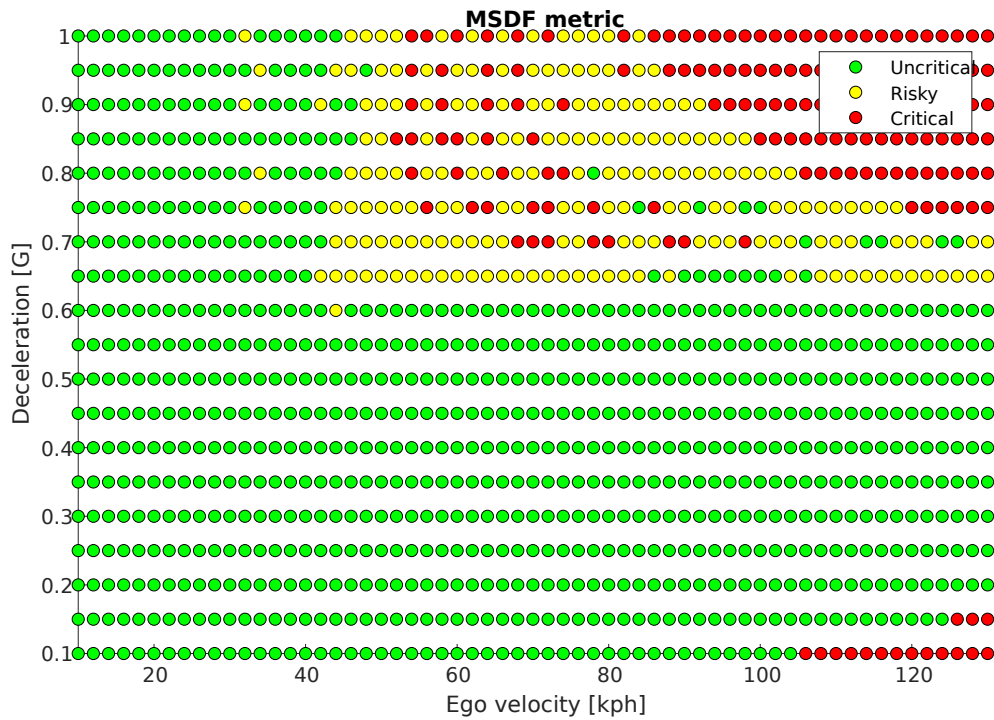


Obr. 13: Implementace metriky MSDF

Výsledky metriky jsou opět interpretovány pomocí grafu. Ten je rozdělen dle podmínek do tří skupin rizikovosti scénáře. Rozdělení je následující:

- nekritický scénář je označen zelenou barvou, pokud je výsledná hodnota větší než 1
- rizikový scénář je označen žlutou barvou, pokud je výsledná hodnota větší než 0,5 a zároveň menší než 1
- kritický scénář je označen červenou barvou, pokud je výsledná hodnota nižší než 0,5

Na obrázku: 14 je vidět graf pro scénář zpomalení z pohledu modelu řidiče 2.



Obr. 14: Výsledek metriky MSDF pro scénář zpomalení z pohledu řidiče 2

V grafech se místy objevují outliers body. Jsou způsobeny chybou simulace. Jelikož data Other vozidla jsou vzorkována dle příslušné periody. V místech, kde je hodnota na hranici rozhodujících kriterií může dojít k malému překročení hranice a díky tomu je bod špatně vyhodnocen.

7 Pomocný skript pro porovnání výsledků

Pro lepší analýzu výsledků byl vytvořen skript, který pomáhá s vyhodnocením podobnosti grafů na základě barevnosti jednotlivých pixelů. Výstupem jsou identifikované body grafů dle jejich barvy pomocí nastavené masky. Masky slouží pro vizuální kontrolu grafů testerem. Ten je díky tomu schopen lépe zanalyzovat výsledky a vzájemně je porovnat. Druhým výstupem je procentuální zastoupení jednotlivých barevných pixelů v grafu vzhledem k celému grafu a vzhledem k sumě všech barevných pixelů při absenci pozadí. Nástroj je dále používán v sekci 9 - Vyhodnocení dat.

7.1 Implementace

Pro napsání skriptu byl zvolen jazyk Python, který disponuje mnoha funkcemi a knihovnami v oblasti zpracování obrazu. Implementace funkce je vysvětlena pomocí schématu na obrázku: 15.



Obr. 15: Schéma struktury pomocného skriptu pro porovnání výsledků

Nástroj využívá dvou dostupných Python knihoven. OpenCV je open-source knihovna obsahující funkce pro práci s obrazem a počítačové vidění [7]. NumPy je také open-source knihovna, která poskytuje funkce pro práci s matematickými poli a obsahuje mnoho matematických funkcí [8]. Dále se skript skládá ze 4 hlavních funkcí: `resize`, `detect_circles`, `percentage`, `disp_mask`.

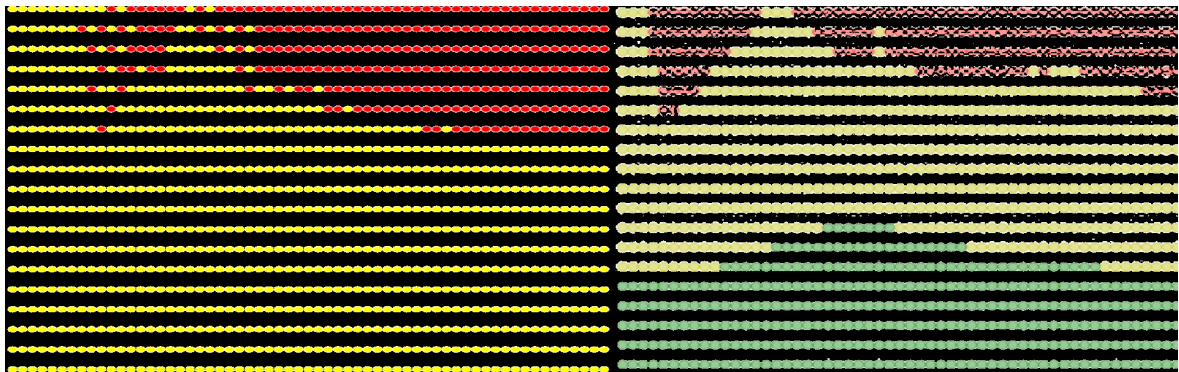
Funkce `resize` jejím výstupem je obrázek o předepsané velikosti a rozměru. Slouží pro sjednocení velikosti obrázků. Vstupem je požadovaný obrázek pro porovnání [7].

Funkce `detect_circles` umožňuje detekci kruhů. Vstupem funkce je obrázek s definovanou velikostí. V prvním kroku je převeden do barevného modelu HSV. Jedná se o definici barev pomocí tří parametrů: odstín (hue), sytost (saturation) a hodnota jasu (value). Takto definované barevné spektrum nejvíce odpovídá lidskému vnímání barev a tím lze lépe definovat hledaný objekt dle jeho barevnosti [9]. Následně jsou definovány spodní a horní hranice jednotlivých hledaných barev. Následně je vytvořena binární maska obrázku, která obsahuje informaci o každém pixelu, zda vyhovuje zadanému kritériu hledané barvy. Tím dojde k filtraci požadovaných oblastí a identifikaci obrázku. Výstupem jsou jednotlivé bitové masky a suma přes všechny masky [7].

Funkce `percentage` vyhodnocuje shodu jednotlivých barev v obrázků. Každý nenulový člen v masce nese informaci, zda daný pixel splňuje kritérium dané barvy. Tím jsou tyto

pixely filtrovány a přiřazeny do jednotlivých skupin barev. Následně je spočten jejich počet a jejich procentuální zastoupení v celém obrázku. Následně je ještě vypočítáno procentuální zastoupení jednotlivých skupin barev ku sumě všech definovaných barvám.

Funkce `disp_mask` zobrazuje finální masky. Vstupem funkce jsou kompletní masky složené ze všech masek jednotlivých barev obou porovnávaných obrázků. Masky jsou následně vyobrazeny do jednoho okna vedle sebe pro vizuální porovnání. Na obrázku: 16 je uveden příklad výstupu maskovaných obrázků určených k vizuálnímu srovnání testerem [7].



Obr. 16: Grafy výsledků s definovanou maskou

Posledním krokem tohoto nástroje je vytisknutí důležitých dat získaných během analýzy do terminálu. Jsou to: počty jednotlivých barevných pixelů, procentuální zastoupení jednotlivých pixelů a procentuální zastoupení jednotlivých skupin pixelů vzhledem ke všem hledaným skupinám pixelů. Příklad výpisu do konzole je uveden na obrázku: 17.

```
PS C:\Users\Denisa\Desktop\DP_new\practice\comparator> & C:/Users/Denisa/AppData/Local/Programs/Python/Py
DP_new/practice/comparator/compare.py
COUNT OF PIXELS EACH COLOR OUTPUT
Number of green pixels OUTPUT 35902
Number of red pixels OUTPUT 7533
Number of yellow pixels OUTPUT 126855
Number of black pixels OUTPUT 2299
COUNT OF PIXELS EACH COLOR REQ
Number of green pixels REQ 86982
Number of red pixels REQ 16261
Number of yellow pixels REQ 134945
Number of black pixels REQ 56
PERCENTAGE OF EACH COLOR
Share of red pixels OUTPUT 1.256% and REQ 2.71%
Share of yellow pixels OUTPUT 21.143% and REQ 22.491%
Share of green pixels OUTPUT 5.984% and REQ 14.497%
Share of black pixels OUTPUT 0.383% and REQ 0.009%
Share of orange pixels OUTPUT 4.008% and REQ 0.012%
Share of background pixels OUTPUT 65.848% and REQ 59.76%
PERCENTAGE OF EACH COLOR COMPARE WITHOUT BACKGROUND
Share of green pixels without background 18.258% and REQ 36.499%
Share of red pixels without background 3.831% and REQ 6.823%
Share of yellow pixels without background 64.512% and REQ 56.624%
Share of black pixels without background 1.169% and REQ 0.023%
Share of orange pixels without background 12.231% and REQ 0.03%
PS C:\Users\Denisa\Desktop\DP_new\practice\comparator> \cite{opencv-python}\cite{opencv-python}
```

Obr. 17: Výpis dat získaných při analýze grafů do konzole

8 Implementace scénářů a metrik

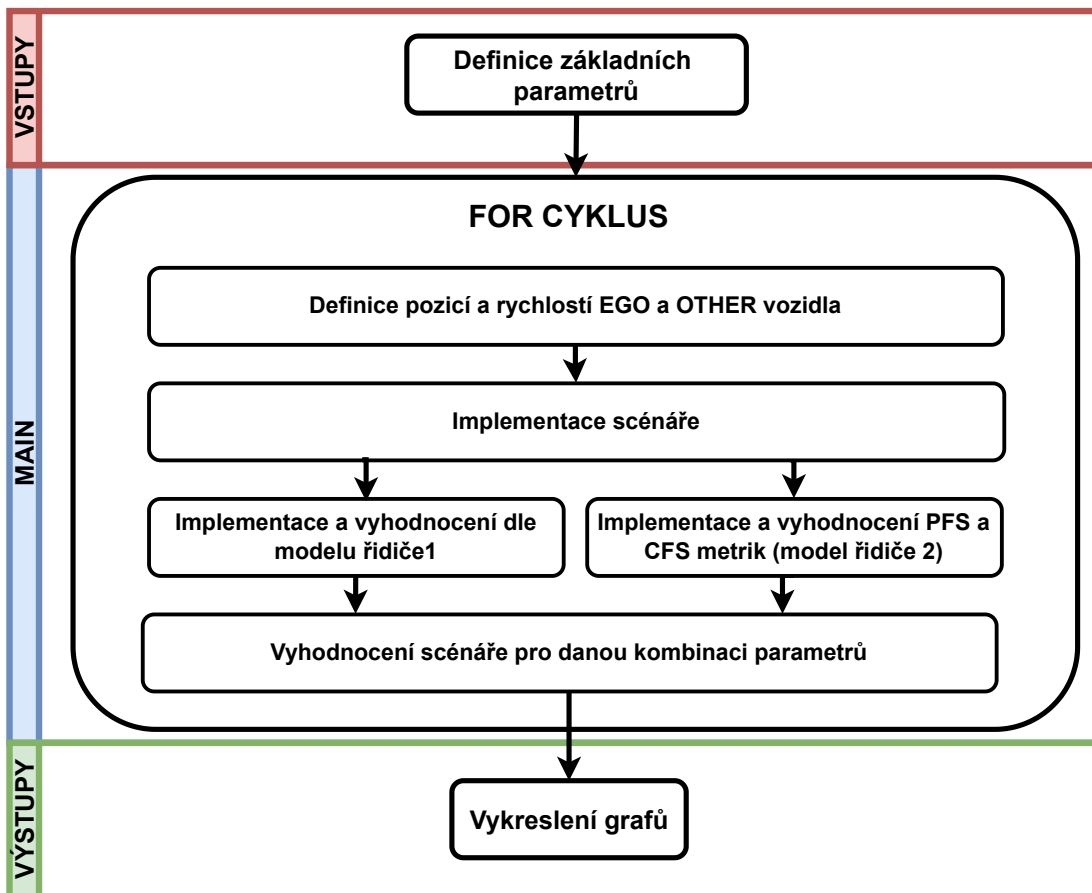
Existuje mnoho softwarů, kterými lze simulovat různé dopravní situace. Z hlediska dostupnosti je možné využít open-source softwaru CARLA nebo Apollo [10]. Jsou to nástroje navržené primárně pro simulace. Obsahují velmi propracované grafické prostředí a vizuální simulace. Nejsou tedy příliš vhodné pro implementaci konkrétních nastavení. Z toho důvodu byla zvolena implementace v programu Matlab/Simulink. Byla provedena rešerše dostupných Matlab toolboxů z nichž byly využity následující: Automated driving system toolbox a Vehicle dynamics toolbox.

Automated driving system Toolbox Tento balíček funkcí obsahuje nástroje, přínosné při vývoji a testování řídicích algoritmů a systému automatického řízení. Disponuje mnoha pokročilými funkcemi jako je například: implementace scénářů, detekce a sledování objektů, rozpoznávání jízdních pruhů apod. Využité funkce v implementaci jsou: Scenario reader a World To Vehicle. Skript také pracuje s aplikací Driving Scenario Designer. Pro grafické zpracování scénáře lze využít bloků 3D simulace vozidel [11].

Vehicle dynamics Toolbox Tento balíček obsahuje možnosti simulace dynamiky vozidla. Obsahuje mnoho funkcí, díky kterým lze simulovat různá dynamická nastavení nejen osobních vozidel. Umožňuje také definovat prostředí a jeho parametry, čímž lze analyzovat chování systémů v různých prostředích. Bylo využito funkce: Vehicle 3DOF Dual Track [12].

8.1 Implementace pomocí Matlab

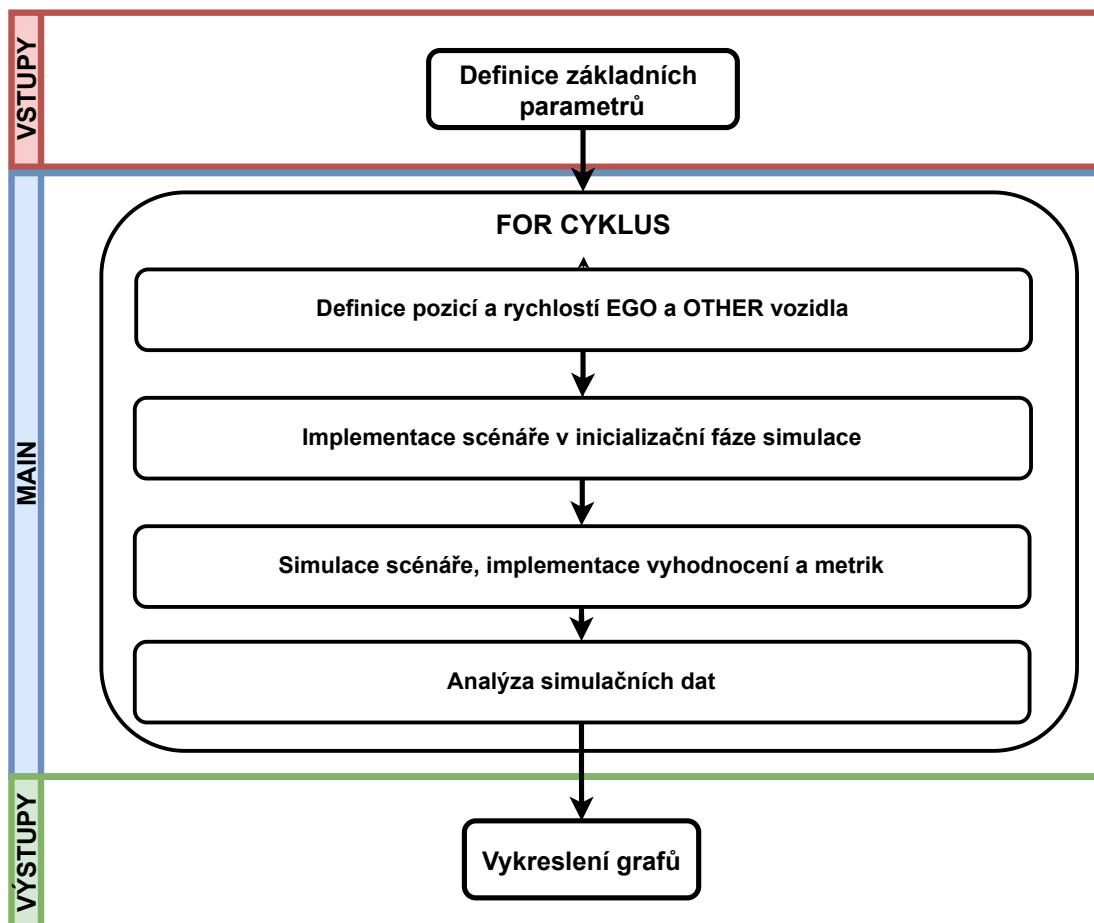
Na začátku byly scénáře implementovány pomocí kódu v Matlabu. V tomto kroku byla využita funkce Driving Scenario Designer. Pro jednotlivé modely řidiče je potřeba specifická implementace scénáře, jiné vyhodnocovací algoritmy a odlišné parametry. Simulace každého modelu je složena ze dvou souborů - Driver_model a scen_deceleration. První zmíněný je strukturovaný dle schématu 18 a je skrz něj volán příslušný scénář - scen_deceleration. Pro interpretaci algoritmu kódu bylo zvoleno názorné schéma na obrázku: 18. V první části označené jako vstupy jsou definovány všechny další potřebné parametry pro daný scénář. Následně je vytvořena for smyčka, která vypočítává konkrétní vstupy scénáře pro danou kombinaci parametrů. Pokračuje vyvoláním scénáře, jeho simulací a uložením žádaných hodnot. Na konci smyčky jsou data vyhodnocena dle daného modelu řidiče a je provedena inkrementace. Po proběhnutí for smyček pro všechny kombinace parametrů dle daného datasetu, následuje poslední část skriptu - výstupy. Výstupem je graf/y, které jsou následně v sekci 9 - Vyhodnocení dat porovnány s grafy výsledků dle specifikace.



Obr. 18: Schéma implementace scénáře pomocí kódu v Matlabu

8.2 Implementace pomocí Simulinku

Implementace scénáře v Simulinku je zabalena do jednoho skriptu, který volá několik dílčích funkcí a samotnou simulaci v Simulinku. Tyto jednotlivé části celého modelu jsou popsány v schématu na obrázku: 19.



Obr. 19: Schéma celého skriptu s využitím Simulinku

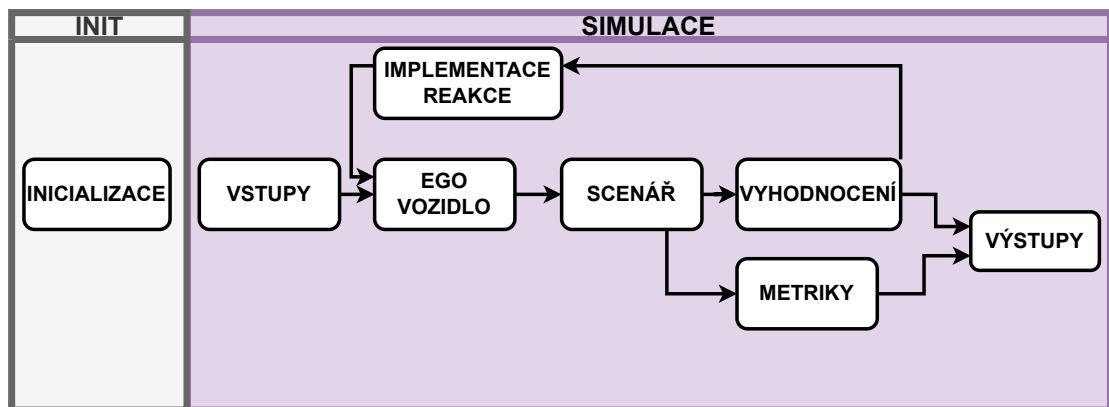
8.2.1 Hlavní skript

Vstupy Skript začíná definováním vstupních parametrů a proměných. Mezi ně patří: struktura *Ego vozidla*, rychlosti obou aktérů, jejich pozice a další pomocné proměnné. Poté následuje kombinace For cyklů. Zde je možné nastavit požadované rozsahy hodnot, pro které je potřeba provést simulaci. Tato část je ve schématu 19 označena červenou barvou. Následovat bude modrá část - main.

Main Hlavní část začíná spuštěním Simulink modelu scénáře a obou vozidel. V inicializační fázi jsou stanoveny body trajektorie obou vozidel a rychlost *Other vozidla* v těchto bodech dle vstupních hodnot. Následně jsou tyto body vloženy do generátoru scénáře, který dle těchto bodů aproximuje vhodnou trajektorii. Nyní je scénář implementován a

lze jej použít pro simulaci. Následně je spuštěna simulace realizovaná Simulink modelem. Jeho podrobné schéma je vyobrazeno na obrázku: 20 a podrobněji bude rozebráno v dalších podsekcích. Během simulace jsou sbírána data o aktérech scénáře a jsou částečně vyhodnocována. Po ukončení simulace se kód vrací do Matlab skriptu, kde dále pokračuje For cyklus s analýzou daného případu dle nasbíraných dat. Analýza je uložena do příslušných proměnných.

Výstupy Po doběhnutí všech For cyklů, je analýza graficky zpracována do jednotlivých grafů. V této části je dostupné vyhodnocení scénářů dle daného modelu řidiče a jednotlivé grafy použitých metrik kritičnosti. Tato část je vyznačena na schématu 19 zelenou barvou.



Obr. 20: Schéma Simulink modelu

8.2.2 Simulink model

Simulink model je složen ze dvou hlavních částí - inicializace a simulace. Jeho schéma je vyobrazeno na obrázku: 20.

Inicializace Tato část je spuštěna před každou simulací. Jedná se o skript, kde jsou definovány již zmíněné body trajektorie. Následně je dle těchto bodů vygenerován samotný scénář, který bude odsimulován.

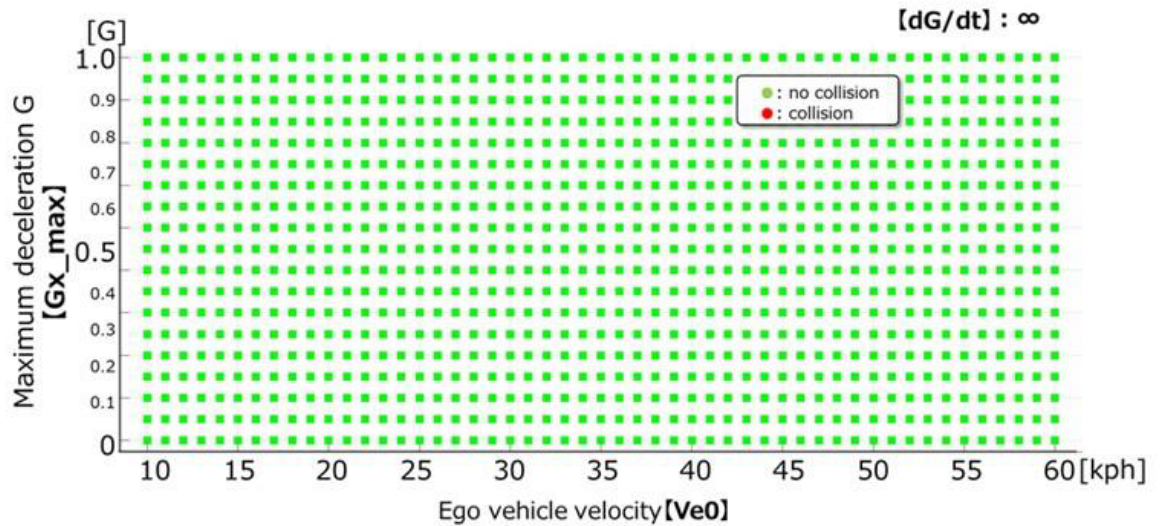
Simulace Simulační část modelu je tvořena několika základními bloky. Schéma bude následně popisováno zleva doprava. Zcela vlevo jsou definovány základní vstupy, které jsou vyžadovány následujícími bloky. Jedná se o počáteční polohy vozidel a jejich rychlosti. Dále na schématu následuje blok *Ego vozidlo*. Zde je pomocí bloku *Vehicle Body 3DOF Dual Track* vygenerováno *Ego vozidlo*, obsahující předepsanou kinematiku a dynamiku. Z jeho výstupu je vytvořena struktura *Ego vozidla* obsahující potřebná data pro jeho zařazení do scénáře. Nad blokem *Ego vozidla* se nachází blok implementace reakce. Tento blok se liší na základě použitého modelu řidiče. Zde dochází k implementaci reakce vozidla na vzniklou situaci dle dat ze simulace scénáře. Dále vpravo následuje blok scénáře, který obsahuje

chování ostatních aktérů scénáře a trajektorii *Ego vozidla*. Poté jsou data ze scénáře využita k jeho vyhodnocení a zároveň jsou použita jako vstupy metrik kritičnosti. Z těchto dvou bloků nad sebou napravo (vyhodnocení a metriky) jsou potřebná data ukládána do proměnných a jsou použity jako výstupy simulace.

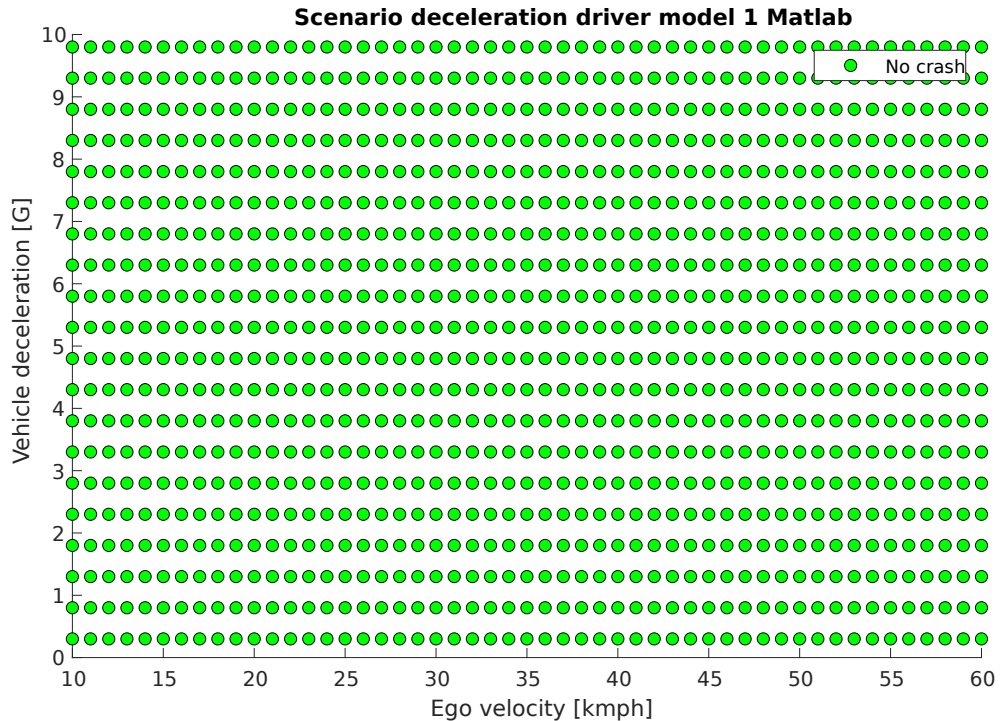
9 Vyhodnocení dat

9.1 Scénář zpomalení

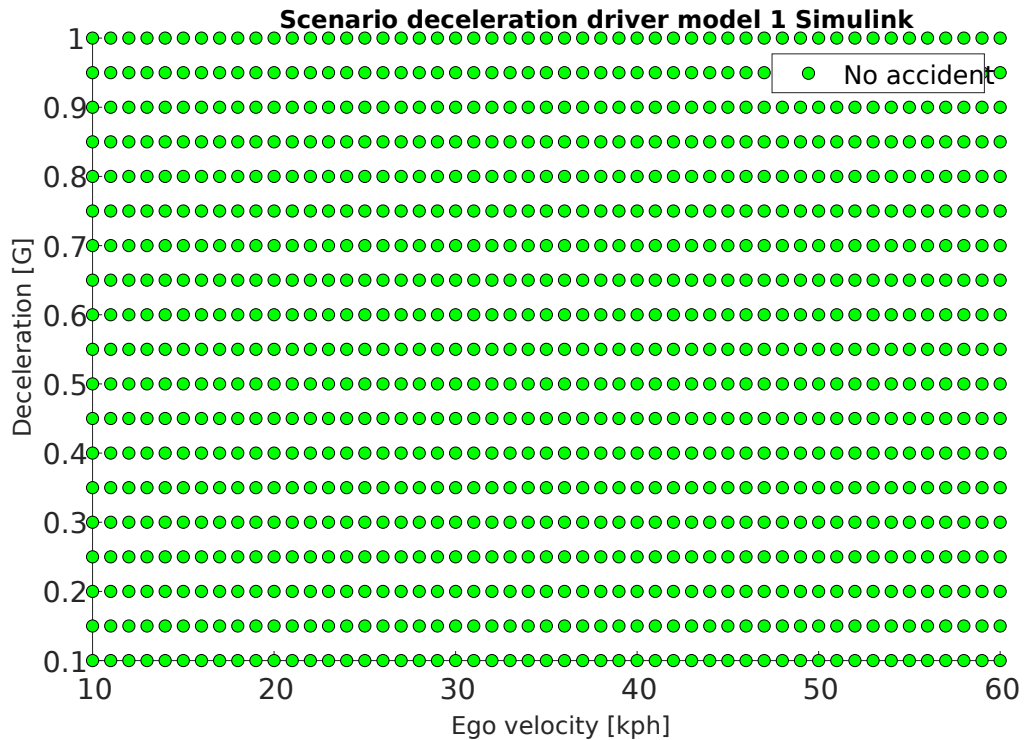
9.1.1 Model řidiče 1



Obr. 21: Výsledky simulací dle specifikace [2]



Obr. 22: Výsledky implementace scénáře v Matlabu



Obr. 23: Výsledky implementace scénáře v Simulinku

Na obrázcích: 21, 22 a 23 jsou vykresleny grafy výsledků vyhodnocení scénáře zpomalení pro model řidiče 1. Na vodorovné ose je uvedena rychlost *Ego vozidla* v rozmezí 10 – 60 *km/h* a na svislé ose je uvedeno maximální brzdění v hodnotách tíhové zrychlení G. Přičemž 1G odpovídá $9.8m \cdot s^{-2}$. Výsledky dle specifikace a implementace v Matlabu i Simulinku jsou stejné. Shodnost je dána především povahou scénáře, který má jednoduchý pohyb a trajektorii. Dále i vyhodnocovací logika je jednoduchá - došlo-li k nárazu či nikoliv. Zelené body zobrazují situace, kdy nedošlo ke kolizi. Naopak červené body reprezentují srážku vozidel.

Dle porovnávacího skriptu bylo dosaženo následující barevné shodnosti grafů. Porovnání specifikace, Matlab implementace a Simulink implementace je uvedena v tabulce 3:

	Matlab implementace [%]	Simulink implemetace [%]	Specifikace [%]
Červená	0,012	0,014	1,141
Žlutá	0,011	0,007	1,09
Zelená	52,557	50,584	30,594
Pozadí	42,647	46,122	62,44

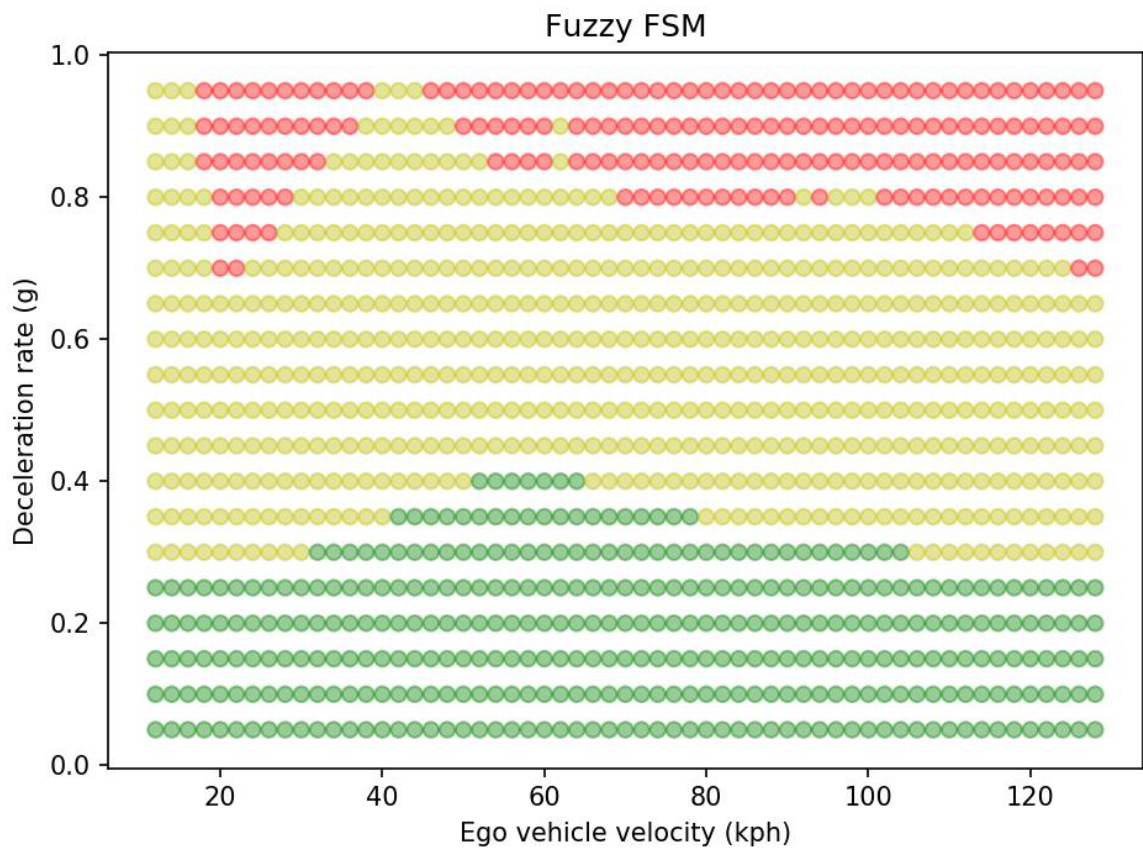
Tab. 3: Procentuální zastoupení jednotlivých barev grafu

Následuje porovnání jednotlivých zastoupení barev při potlačení pozadí. Lze porovnat v tabulce 4:

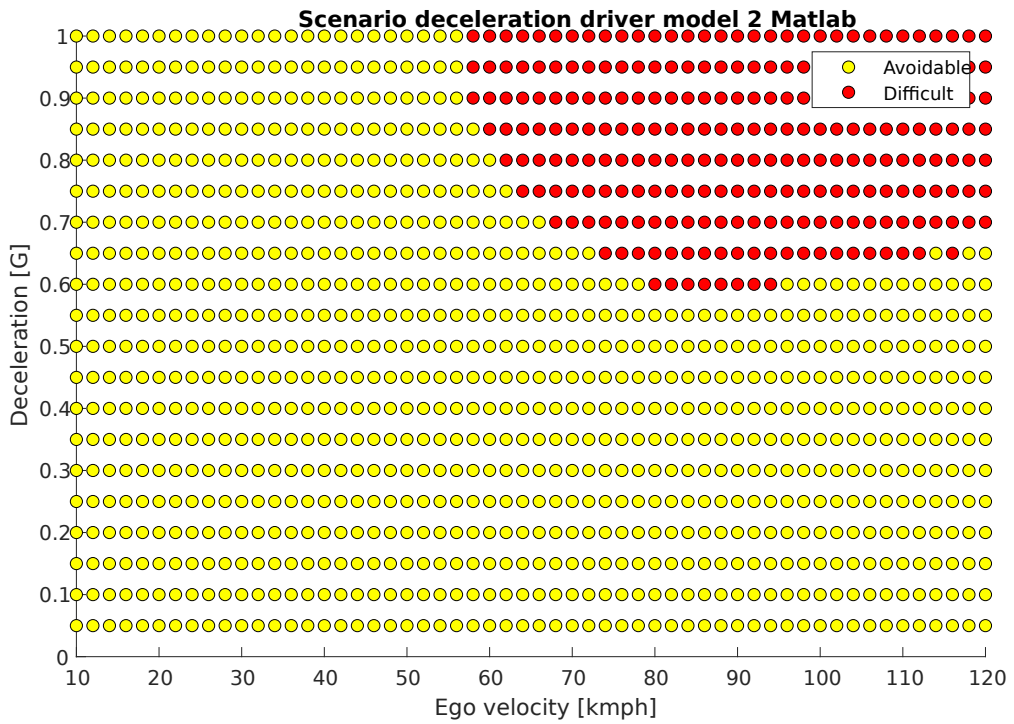
	Matlab implementace [%]	Simulink implemetace [%]	Specifikace [%]
Zelená	99,958	99,958	93,204
Červená	0,022	0,028	3,476
Žlutá	0,02	0,015	3,321

Tab. 4: Procentuální zastoupení jednotlivých barev grafu při absenci pozadí

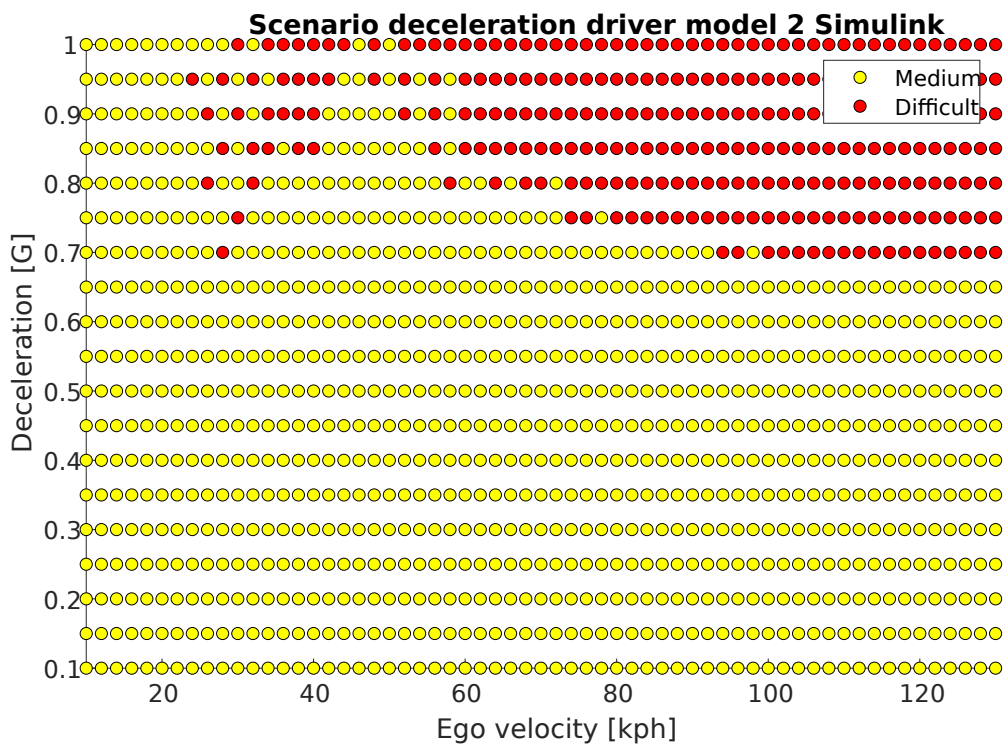
9.1.2 Model řidiče 2



Obr. 24: Výsledky simulací dle specifikace scénáře Zpomalení [2]



Obr. 25: Výsledky implementace scénáře Zpomalení v Matlabu



Obr. 26: Výsledky implementace scénáře Zpomalení v Simulinku

Na obrázku: 24 je zobrazen graf výsledků dle specifikace pro scénář zpomalení dle modelu řidiče 2. Na grafech 25 a 26 jsou zobrazeny výsledky implementace v Matlabu a Simulinku.

V grafu jsou barevně odlišeny 3 náročnosti scénáře. Hranice obtížnosti jsou definovány dle specifikace následovně:

- zelená - jednoduchá reakce: $PFS = 0$,
- žlutá - středně těžká reakce: $PFS > 0$ a $CFS < 0.5$,
- červená - obtížná reakce: $CFS \geq 0.5$.

Na vodorovné ose je uvedena rychlost *Ego vozidla* v km/h a na svislé ose je uvedeno opět tíhové zrychlení G . Jak lze vidět, výsledek implementace Matlab/Simulink a specifikace se liší. Zelené oblasti nelze dosáhnout pro dané počáteční podmínky. Například: při použití stejných počátečních podmínek, tedy metrika TTC s časovým údajem $2s$. Rychlost vozidel je zvolena $25km/h$, počáteční vzdálenost mezi vozidly je $14m$. Bezpečná hranice metriky PFS, jejíž hodnota je uvedena jako d_{safe} a je rovna v tomto případě $9.79m$. Výsledný rozdíl jsou tedy $2m$. Ve vyhodnocení se ještě odečítá konstanta s hodnotou $2m$. Sníží-li se vzdálenost mezi aktéry o $2, 5m$, dojde k aktivaci PFS metriky. Jelikož dle metriky je velikost zpomalení *Ego vozidla* dáno hodnotou PFS metriky, nelze dosáhnout nulové hodnoty metriky pro dané parametry scénáře (zelená oblast grafu). Další chybou implementace je simulace vozidla v toolboxu Driving scenario designer. Data jsou zaznamenávána s periodou $0, 1s$. Za tuto dobu při rychlosti $25km/h$ vozidlo ujede téměř $0.7m$. Akceleraci vozidla nelze zcela přesně nastavit. Dále také nelze nastavit podrobněji dynamiku vozidla. Z tohoto důvodu byla zvolena implementace v Simulinku pro dosažení reálnějších dat zejména v kritických oblastí grafu.

V následujících tabulkách jsou uvedeny procentuální shody výsledného grafu dle specifikace v porovnání s výsledky dle implementací v Matlabu a Simulinku. V tabulce 5 jsou porovnány procentuální zastoupení jednotlivých barev. V tabulce 6 jsou porovnány zastoupení jednotlivých barevných skupin při absenci pozadí.

	Matlab implementace [%]	Simulink implemetace [%]	Specifikace [%]
Červená	6,338	6,464	5,287
Žlutá	23,49	20,869	25,878
Zelená	0,0	0,0	13,45
Pozadí	71,128	73,649	52,67

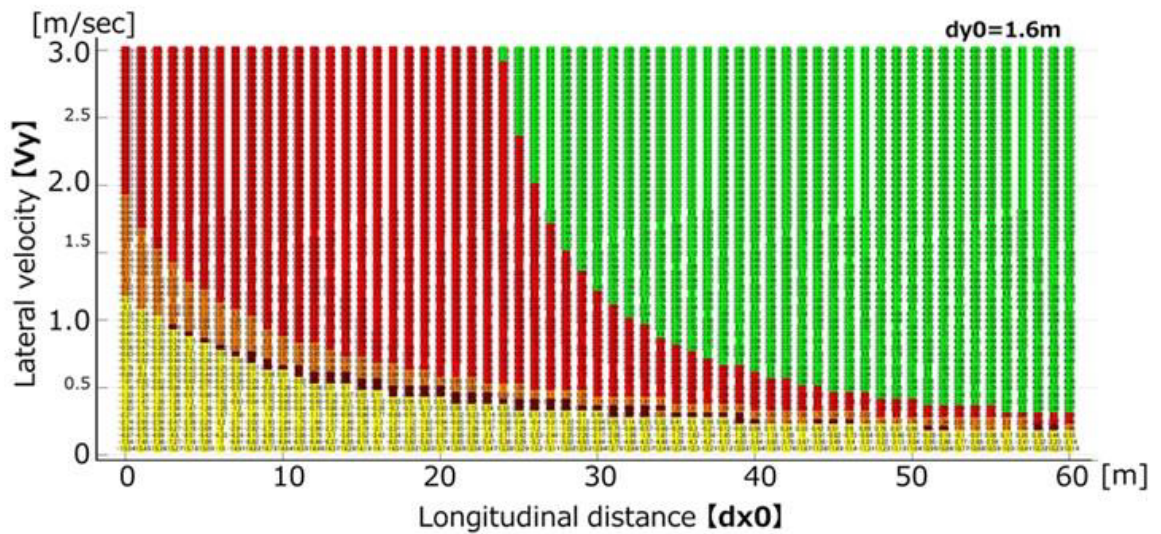
Tab. 5: Procentuální zastoupení jednotlivých barev grafu

	Matlab implementace [%]	Simulink implemetace [%]	Specifikace [%]
Zelená	0,0	0,0	30,146
Červená	21,25	23,649	11,851
Žlutá	78,75	76,351	58,004

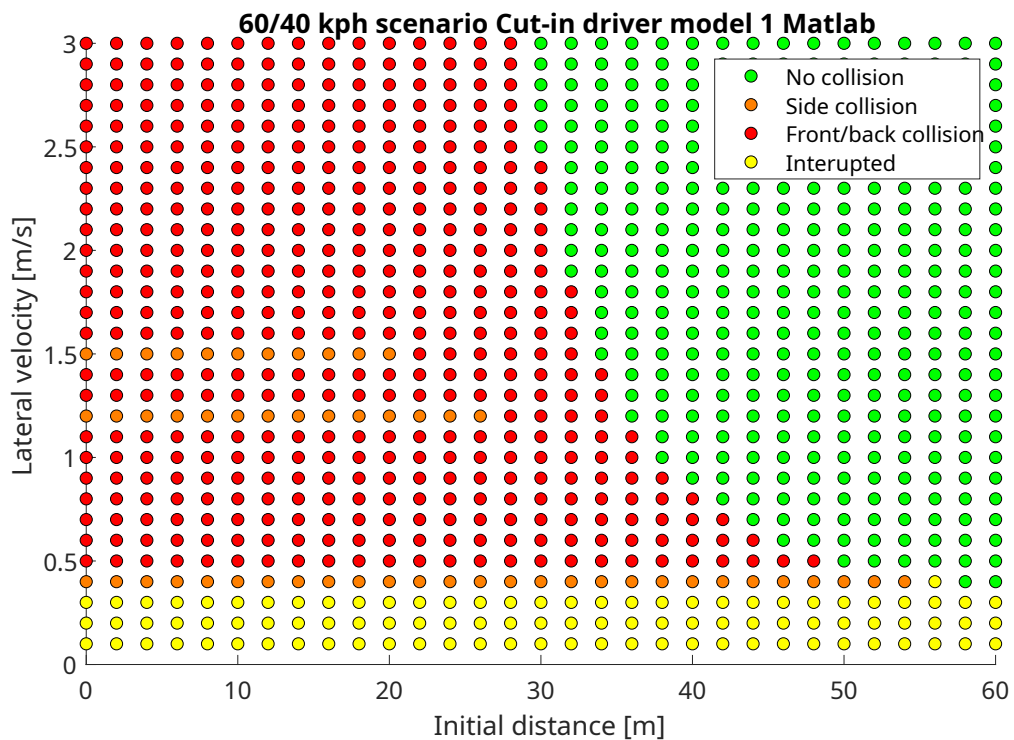
Tab. 6: Procentuální zastoupení jednotlivých barev grafu při absenci pozadí

9.2 Scénář Cut-IN

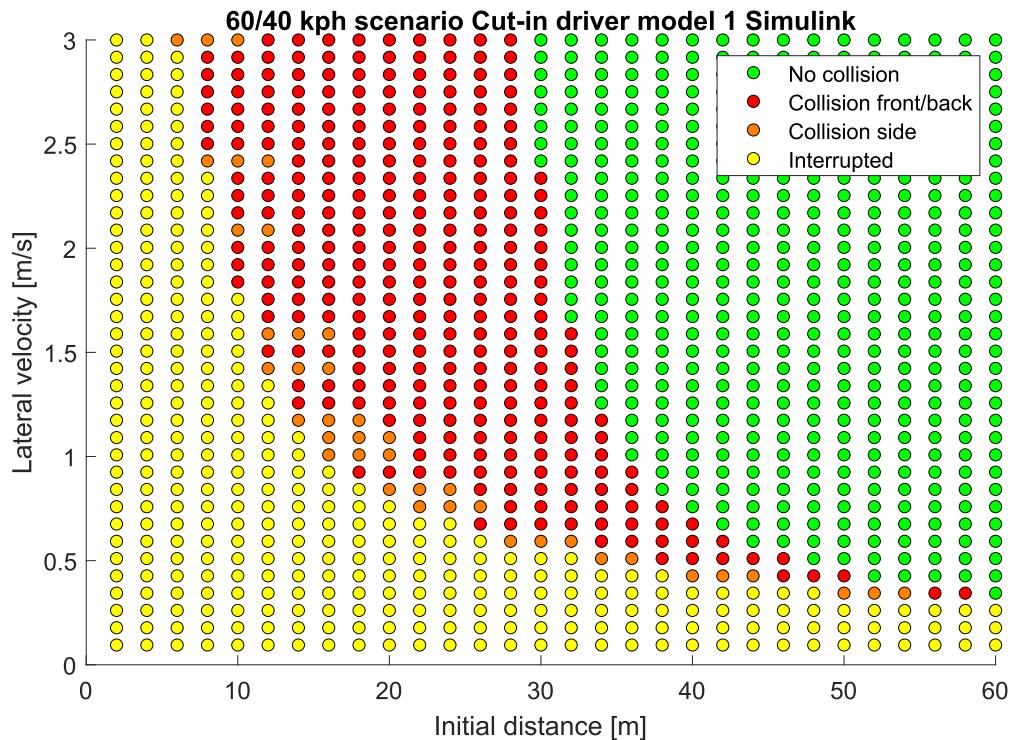
9.2.1 Model řidiče 1



Obr. 27: Výsledky simulací dle specifikace scénáře Cut-in [2]



Obr. 28: Výsledky implementace scénáře Cut-in v Matlabu



Obr. 29: Výsledky implementace scénáře Cut-in v Simulinku

Na obrázcích: 29 a 28 jsou výsledné grafy scénáře Cut-in při použití modelu řidiče 1. Grafy jsou výsledky při volbě podélné rychlosti *Ego vozidla* 60km/h a podélné rychlosti *Other vozidla* 20km/h . Dle grafu ve specifikaci na obrázku: 27 lze rozlišit pět různých situací:

- zelená - scénář bez kolize,
- červená - scénář, kde došlo k přední/zadní kolizi,
- oranžová - scénář, kde došlo k boční kolizi,
- žlutá - scénář, kde došlo k předjetí *Other vozidla Ego vozidlem*, přičemž *Other vozidlo* se v danou chvíli zcela nezařadilo do pruhu,
- hnědá - dle specifikace není definováno.

Tyto body jsou určeny nastavením vyhodnocovacích kritérií. Pro vyhodnocení dat ze scénářů bylo zvoleno následující řešení. Z dat polohy obou aktérů je určena jejich vzájemná poloha. Pokud se polohy rovnají či je poloha *Ego vozidla* vzdálenější v podélném směru, dojde k vyhodnocení srážky červené barvy. Pokud jsou polohy vozidel v příčném směru rovny (či v dané toleranci) a zároveň poloha vozidel v podélném směru je v rozmezí velikosti vozidel, scénář je vyhodnocen oranžovou barvou. Speciálním případem je situace reprezentována žlutou barvou. Pokud se *Ego vozidlo* dostane během manévru *Other vozidla* zcela před něj a zároveň *Other vozidlo* nepřekročilo definovanou vzdálenost, je scénář označen jako "přerušeni". Při vyhodnocení tohoto scénáře bylo opět nalezeno několik

faktorů, které je nutné zohlednit vzhledem ke zjednodušení modelu. K hlavnímu zkrácení dochází opět v periodě vzorkování. Data jsou sbírána s periodou 0,1 s. Pokud vozidlo jede nejvyšší testovanou rychlostí - 60 km/h, za tuto dobu pro představu urazí zaokrouhleně 1,667 m. Záleží na konkrétní implementaci daného systému, jak často je schopen data vyhodnocovat. S nižší periodou vzorkování je systém schopen rychleji zareagovat na blížící se nebezpečí a určit přesněji dobu kolize a zároveň její typ. Dále lze vyhodnocení zpřesnit implementací senzorů, např.: ultrazvuků. Není ovšem definováno, že systém ALKS musí používat pro vyhodnocení výhradně ultrazvukové senzory. Momentálně je na trhu nabízeno mnoho přístrojů, kde princip zpracování dat se velice liší. Jsou to například: zmíněné ultrazvukové senzory, LiDARy, radary či kamery. V moderním automobilovém průmyslu se vyhodnocovací logiky běžně zakládají na datech získaných pomocí sestavy "Sensor fusion". Jedná se o sestavu mnoha senzorů, které mohou být stejného či jiného typu. Spojením jejich výstupů je získáno velké množství přesnějších dat, která jsou následně vyhodnocována systémem. Záleží tedy na výrobci, jakými přístroji bude získávat data z okolí. Ty se mohou lišit v technických parametrech v závislosti na výrobci. Pokud by se i s přihlédnutím k této nepřesnosti implementovali senzory v Matlabu, opět i zde lze nalézt několik omezení. Limitace jsou například v umístění senzorů, nastavení jejich zorného úhlu nebo v nastavení jejich dosahu. Dále nelze zcela přesně simulovat okolní podmínky, tj. déšť, mlha či sníh. Ty mohou významně ovlivnit fungování systému. Z důvodu robustnosti a flexibility vyhodnocovacího algoritmu, bylo zvoleno vyhodnocení dat vzájemné polohy obou aktérů. Tedy jejich polohu v jednotném koordinovaném systému vodorovné osy X a horizontální osy Y.

Implementace v Matlabu je díky výše zmíněným vneseným chybám velice nepřesná. Pokud by pro testera byla stěžejní pouze orientační informace, zda při dané kombinaci parametrů dojde ke kolizi či nikoliv. Nebo také přibližná poloha oblasti, ve které dochází ke změně výsledku simulace z bezkolizní situace ke kolizní situaci. Pro tyto orientační analýzy, by byl tento model dostačující. Ovšem k přihlédnutí časové náročnosti obou implementací, kde ověření jedné simulace pomocí Matlab kódu a simulace pomocí Simulinku je téměř shodná, je příhodné zvolit použití Simulink modelu. Nejen, že více odpovídá reálným podmínkám, ale lze z něj získat podrobnější informace o průběhu scénáře. Tyto informace mohou být stěžejní, při další podrobnější analýze. V praxi se s formou ukládání těchto dat a jejich následnou podrobnější analýzou lze setkat velmi často. Dle specifikace byl tento model řidiče testován pro celkem 11 kombinací vzájemných podélných rychlostí vozidel. Pro interpretaci výsledků byla zvolena jedna zmíněná kombinace parametrů. Ostatní výsledky lze ovšem nalézt na přiloženém datovém nosiči. Některé datasey mají uložena data pro rozsah počáteční vzdálenosti 0 – 30 m z důvodu úspory dat. Ve vynechané oblasti se nevyskytují podstatné informace.

Následně byly oba grafy zanalyzovány pomocí vytvořeného porovnávače grafů. Jejich

barevná shodnost je vyjádřena v procentech v následujících tabulkách 7 a 8:

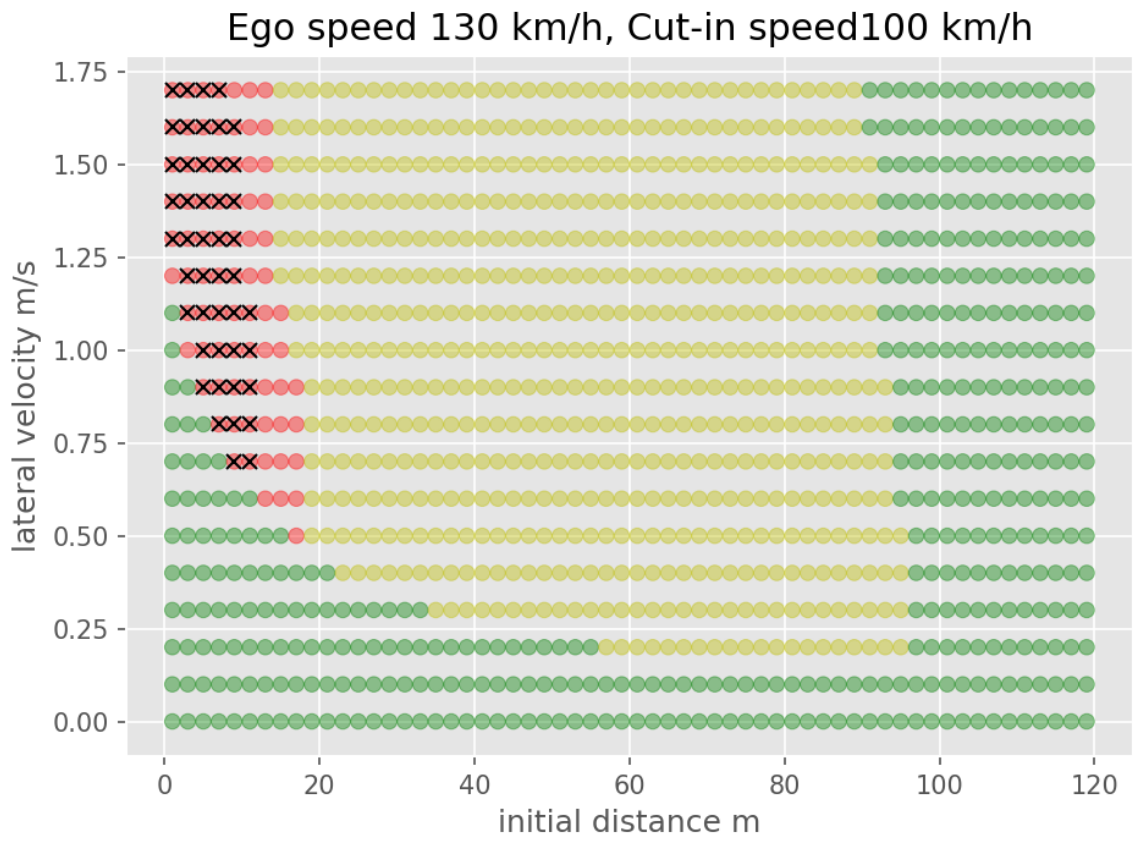
	Matlab implementace [%]	Simulink implemetace [%]	Specifikace [%]
Červená	8,49	6,51	16,918
Žlutá	2,079	7,794	11,899
Zelená	7,24	10,311	33,739
Černá	0,0	0,0	0,0
Oranžová	1,121	0,967	2,532
Pozadí	79,771	73,348	19,854

Tab. 7: Procentuální zastoupení jednotlivých barev grafu

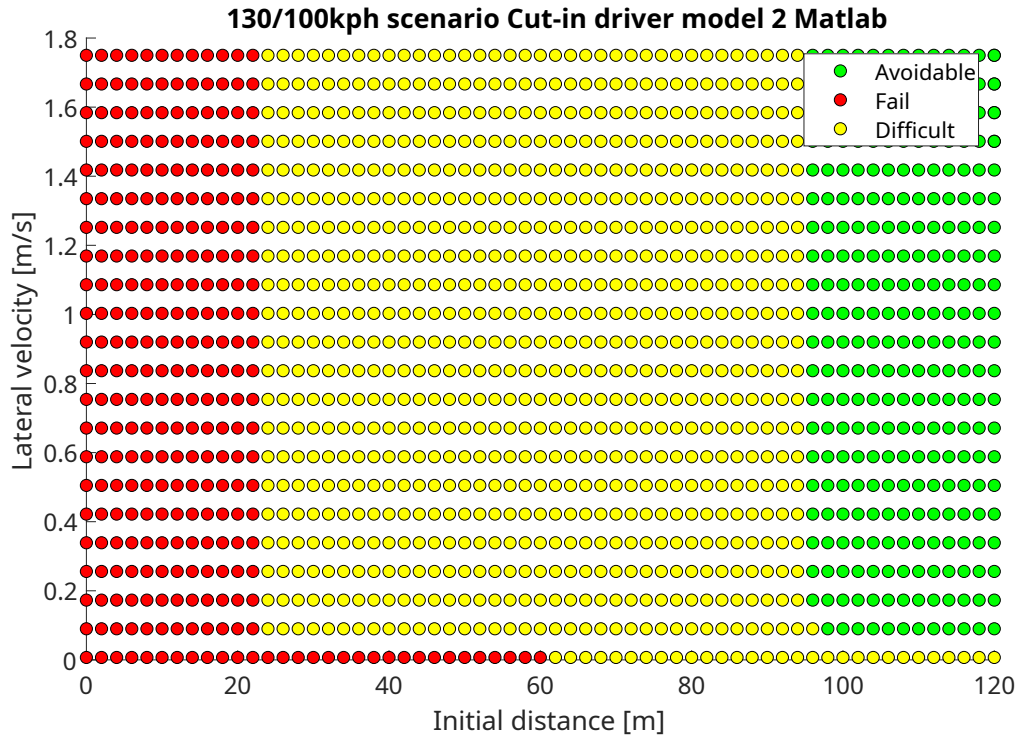
	Matlab implementace [%]	Simulink implemetace [%]	Specifikace [%]
Zelená	38,245	40,305	51,836
Červená	44,851	25,449	25,992
Žlutá	10,981	30,467	18,282
Černá	0,0	0,0	0,0
Oranžová	5,923	3,779	3,89

Tab. 8: Procentuální zastoupení jednotlivých barev grafu při absenci pozadí

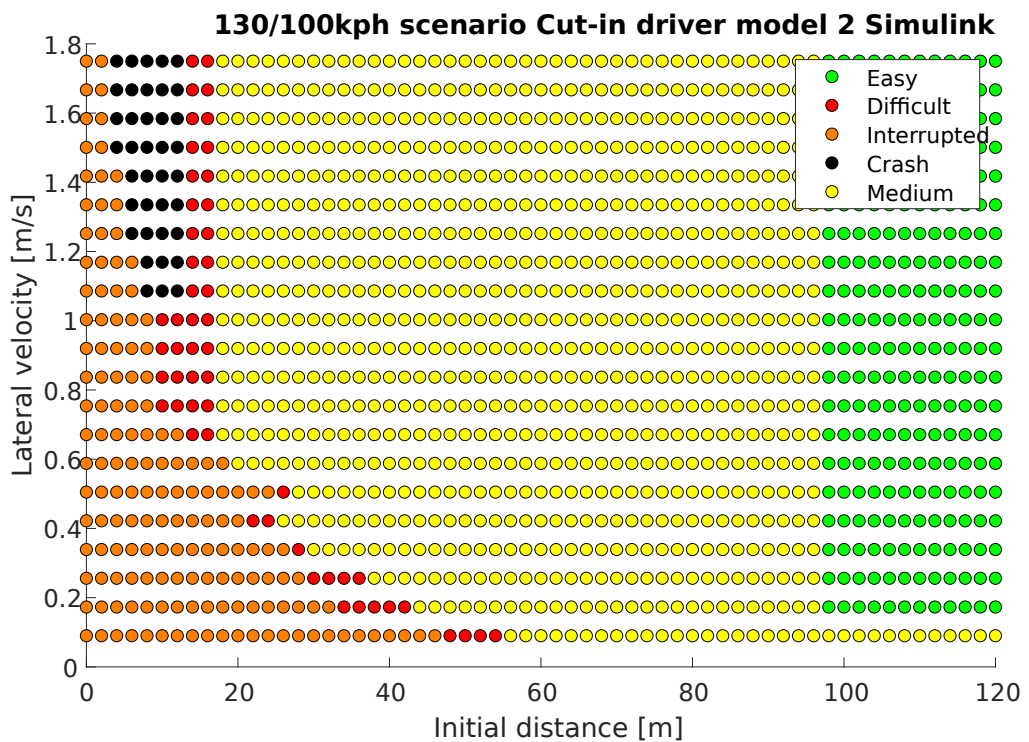
9.2.2 Model řidiče 2



Obr. 30: Výsledky simulací dle specifikace scénáře Cut-in [2]



Obr. 31: Výsledky implementace scénáře Cut-in v Matlabu



Obr. 32: Výsledky implementace scénáře Cut-in v Simulinku

Dle specifikace jsou uvedeny 4 kombinace podélných rychlostí obou vozidel při testování modelu řidiče 2. Na obrázku: 30 jsou uvedeny výsledky pro počáteční vzdálenost vozidel

od 120/60m do 0m a příčnou rychlost *Other vozidla* od 1, 75km/h do 0km/h dle specifikace. Na obrázcích: 31 a 32 jsou uvedeny výsledky Matlab/Simulink implementace. Barevné rozlišení jednotlivých skupin výsledků simulací je následující:

- zelená - jednoduchá reakce: $PFS \leq 0.85$,
- žlutá - středně těžká reakce: $PFS > 0.85$ a $CFS < 0.9$,
- červená - obtížná reakce: $CFS \geq 0.9$,
- černá - scénář, při kterém došlo ke střetu vozidel.

Pro interpretaci a zhodnocení modelu byla opět vybrána pouze jedna varianta kombinace parametrů. Konkrétně byla zvolena kombinace podélné rychlosti *Ego vozidla* 130km/h a podélné rychlosti *Other vozidla* 100km/h. Jedním z důvodů výběru těchto konkrétních parametrů je výskyt všech definovaných oblastí a také výskyt těchto rychlostí v praxi. Tento asistent je velice vhodný pro použití na dálnicích, kde je v České republice nejvyšší povolená rychlost právě 130km/h. Druhá rychlost ze zvolené kombinace je na těchto typech silnicí pro zařazující se vozidlo velmi běžná. Pro mnoho typů vozidel je tato rychlost maximální povolená rychlost vzhledem k jejich konstrukci a využití. Mohou to být například: autobus, přívěsný vozík či nákladní automobil. Implementace pomocí Matlab scriptu je opět poměrně nepřesná a velmi orientační. Hraniční parametry jednotlivých skupin výsledků se pohybují v podobném rozmezí. Opět lze tento model použít pro obecný a přibližný vzhled do dané problematiky. Pro konkrétní testování a validaci systému je ovšem tento model velice nepřesný a není vhodné jej použít. Model dle Simulinku již více odpovídá reálným podmínkám a poskytuje opět detailnější informace. Ve specifikaci je zanedbána oblast, která byla v modelu řidiče 1 nazývána jako "přerušení". Pro tento speciální výsledek simulace bylo použito stejného vyhodnocovacího algoritmu jako v implementaci pro model řidiče 1. Z grafu na obrázku: 32 je patrné, že tato situace se vyskytuje i při použití modelu řidiče 2 a je potřeba ji do výsledné vyhodnocovací logiky zahrnout. V této implementaci pro ni byla zvolena oranžová barva. Z důvodu zachování konzistentnosti barev pro následné porovnání se specifikací pomocí porovnávače grafů.

V následujících tabulkách 9 a 10 lze srovnat procentuální barevnostní shodu jednotlivých grafů:

	Matlab implementace [%]	Simulink implemetace [%]	Specifikace [%]
Červená	4,303	1,256	2,71
Žlutá	13,533	21,143	22,491
Zelená	4,369	5,984	14,497
Černá	0,0	0,383	0,009
Oranžová	0,0	4,008	0,012
Pozadí	77,377	65,848	59,76

Tab. 9: Procentuální zastoupení jednotlivých barev grafu

	Matlab implementace [%]	Simulink implemetace [%]	Specifikace [%]
Zelená	19,677	18,258	36,499
Červená	19,377	3,831	6,823
Žlutá	60,946	64,512	56,624
Černá	0,0	1,169	0,023
Oranžová	0,0	12,231	0,03

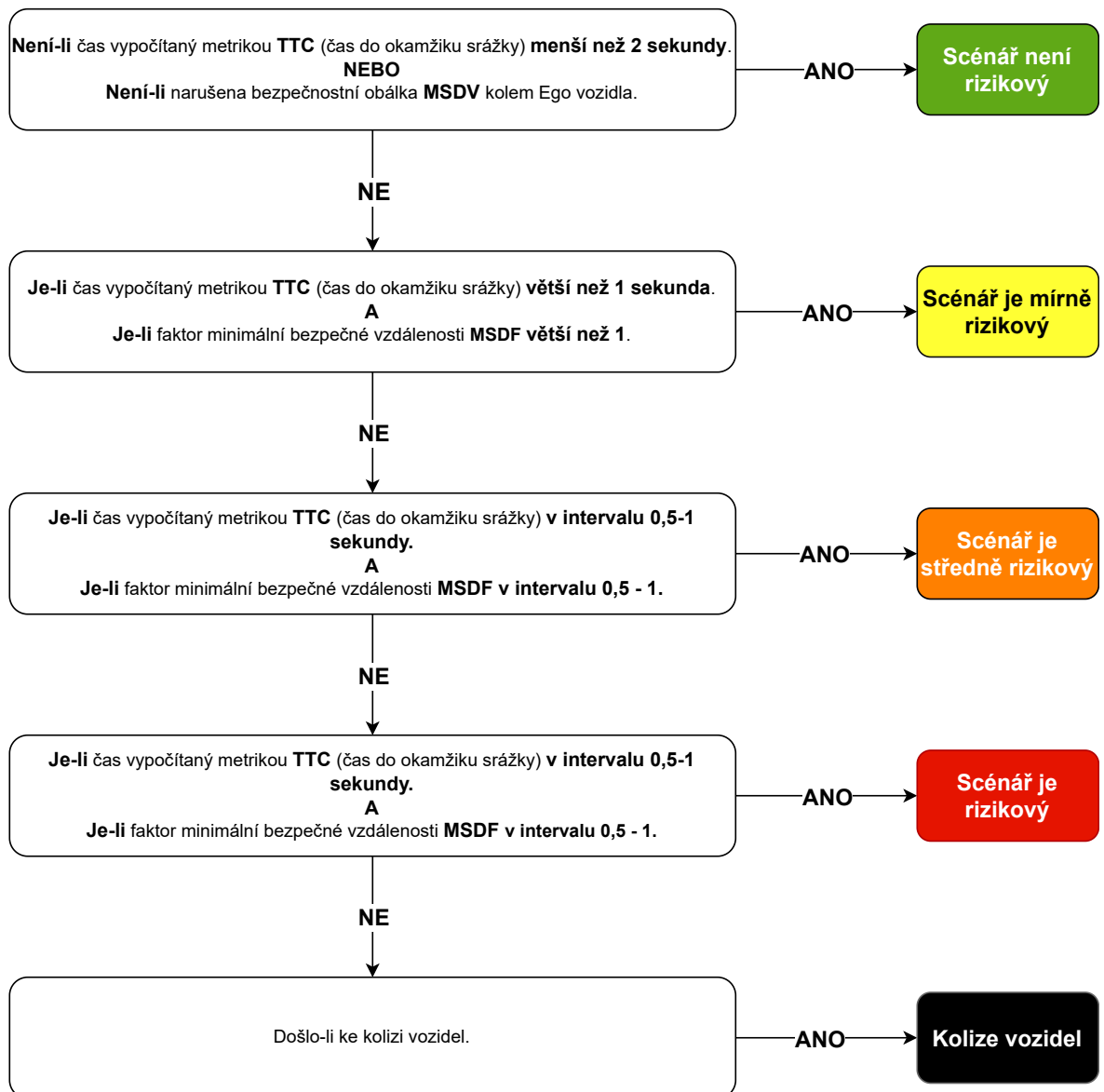
Tab. 10: Procentuální zastoupení jednotlivých barev grafu při absenci pozadí

Za účelem zlepšení čitelnosti grafů byla nastavena černá hranice jednotlivých bodů všech grafů implementovaných v Matlab/Simulink.

10 Vytvoření nové vyhodnocovací metody

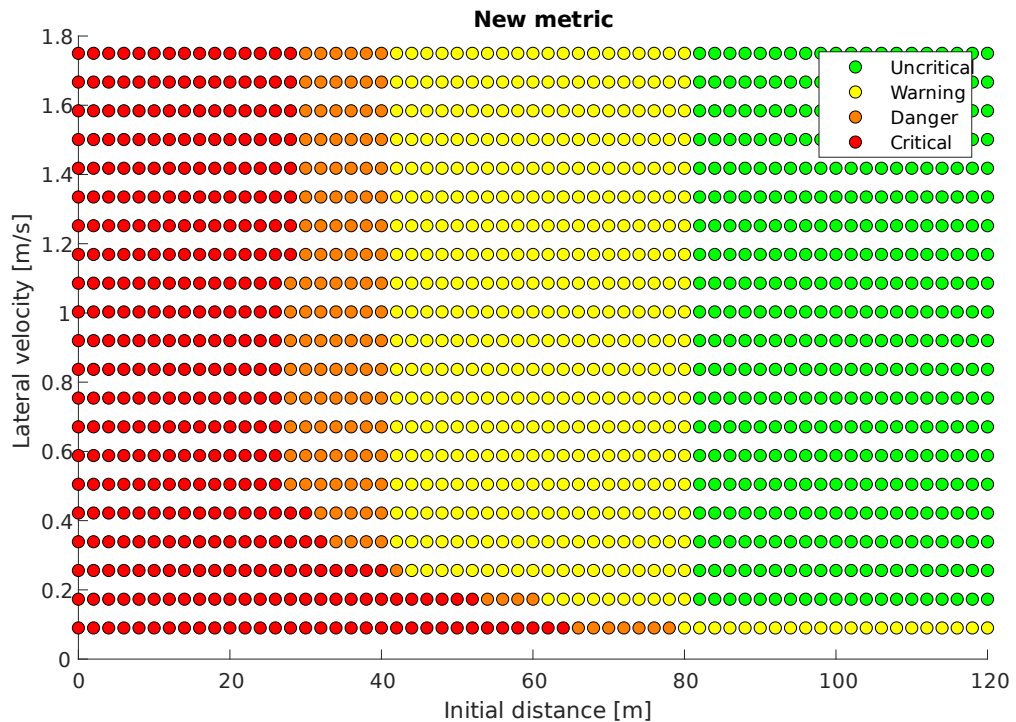
Z rešerše metrik kritičnosti a jejich implementace byly získány potřebné informace o dané problematice vyhodnocování scénářů. Zadaným úkolem bylo vytvořit novou metodu pro vyhodnocení scénářů, která by se dala uplatnit v praxi. Bylo využito výstupů použitých metrik kritičnosti k identifikaci hraničních parametrů. Hraniční parametry jsou hodnoty veličin, které se pohybují na hranici jednotlivých stupňů kritičnosti scénářů. Nejdříve byla vytvořena nová vyhodnocovací metrika s využitím již používaných metrik kritičnosti. Konkrétně se jedná o: TTC, MSDV a MSDF. Následně jejich spojením bylo vytvořeno škálování kritičnosti daných parametrů do 5 kategorií. Rozhodovací algoritmus metriky je na obrázku: 33. Schéma obsahuje dva sloupce. V levém sloupci je vždy uvedena podmínka. Pokud je splněna, scénář je přiřazen do jedné z pěti skupin. Bylo použito opět barevné odlišení jednotlivých skupin dle stupně kritičnosti:

- zelená barva - scénář je bezrizikový,
- žlutá barva - scénář je mírně rizikový,
- oranžová barva - scénář je středně rizikový,
- červená barva - scénář je rizikový,
- černá barva - kolize vozidel.



Obr. 33: Rozhodovací algoritmus metriky

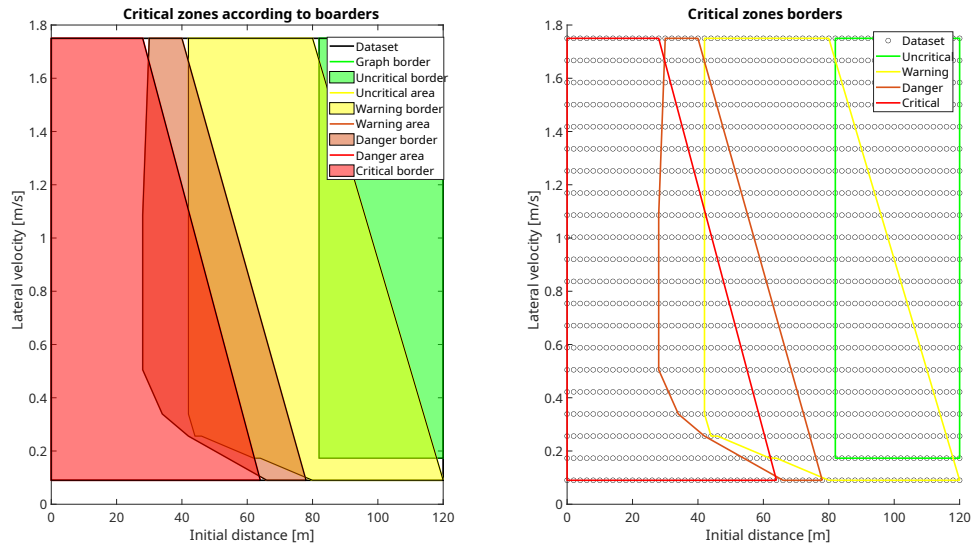
Grafickým výstupem metriky je graf s příslušnými parametry na jednotlivých osách pro daný scénář. Příklad výsledku metriky pro konkrétní scénář je vidět na obrázku: 34. Jedná se o použití metriky pro scénář Cut-in model řidiče 2 s parametry rychlostí vozidel 130/100km/h.



Obr. 34: Výsledek nové metriky pro scénář Cut-in model řidiče 2 a rychlosti 130km/h a 100km/h

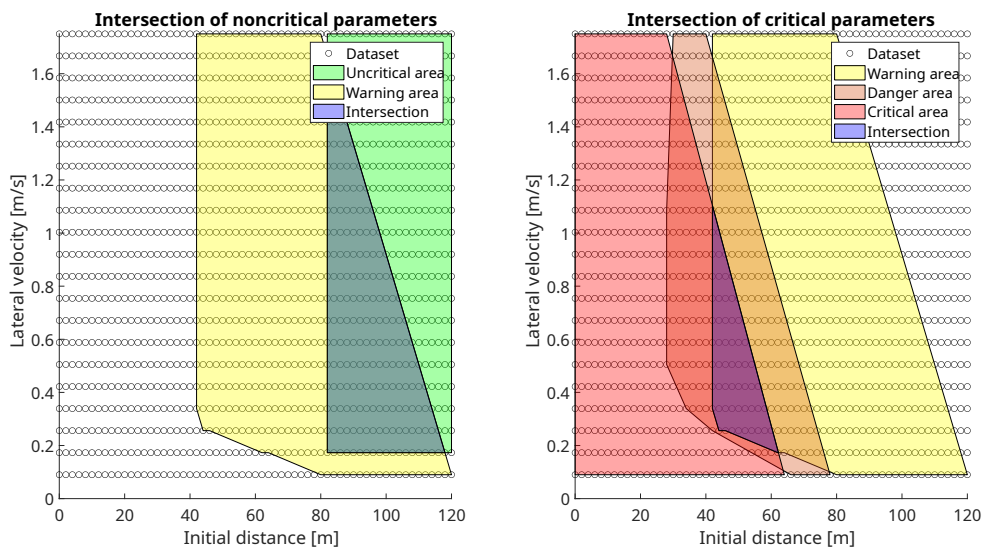
Dalším krokem vyhodnocovací metody je detekce hran jednotlivých barevných oblastí grafu. Pro jejich identifikaci byla použita funkce "boundary()", jejímž výstupem jsou hraniční body dané skupiny [13]. Pro zjednodušení vyhodnocení je dalším grafickým výstupem obrázek s dvěma subploty¹⁹. Tyto výsledky si lze prohlédnout na obrázku: 35. Na levém subplotu jsou zobrazeny hranice jednotlivých skupin včetně barevného rozlišení celé oblasti. Lze si prohlédnout překrývání daných oblastí, které jsou cílovým objektem této identifikace. Napravo jsou vykresleny již jen hranice daných oblastí. Osy grafů jsou stále definovány konkrétními parametry daného scénáře.

¹⁹Subplot - podgraf v jednom grafu



Obr. 35: Identifikace hranic oblastí grafu

Posledním grafickým výstupem je identifikace oblastí průniku. K identifikaci jednotlivých oblastí průniku byla použita funkce "intersect()" [14]. Výsledkem jsou opět dva subploty. Na levém je zobrazen průnik mírně rizikové a nerizikové oblasti. Vpravo je průnik rizikových oblastí. Osy grafů jsou opět dány konkrétními parametry daného scénáře. Tyto grafy jsou zobrazeny na obrázku: 36.



Obr. 36: Identifikace průnikových oblastí grafu mírně rizikové a rizikové

Cílem této nově implementované metody bylo vytvořit nový pohled na identifikaci kritičnosti scénáře a automaticky najít kritickou oblast, na kterou je nutné se při testování systému zaměřit. Tyto oblasti jsou graficky vykresleny na posledním grafickém výstupu metriky.

Následně jsou ještě do konzole vypsány doporučené intervaly kritických parametrů, na které by se tester měl zaměřit.

11 Závěr

Cílem této práce bylo identifikovat a implementovat kritické scénáře, provést jejich simulace pro různé kombinace parametrů a následně zanalyzovat jejich výsledky. Byla vytvořena rešerše testovaného systému, identifikace kritických scénářů, rešerše bezpečnostních předpisů a kritických metrik. Následně se práce zaměřila na implementaci testovacího modelu. Byl proveden rozbor dostupných softwarů vhodných pro implementaci. Hlavním důvodem zvolením pro implementaci program Matlab/Simulink byla jeho robustnost a možnosti nastavení parametrů tak, aby se model co nejvíce přiblížil reálným situacím a reálným vozidlům.

Scénáře a vozidla byly nejprve implementovány pomocí kódu v Matlabu s využitím některých toolboxů. Při vyhodnocování těchto výsledků bylo zjištěno, že jsou zatíženy chybami vnesenými jednoduchostí modelu. Například absencí dynamiky vozidla. Z tohoto důvodu byl navržen jiný způsob implementace, který by výsledek zpřesnil. Pro tyto účely byl zvolen a použit program Simulinku. Následně byl vytvořen model celého testu, který zahrnuje: základní model vozidla, požadovaný reakční systém, scénář a vyhodnocovací metriky. Nakonec byly tyto výsledky porovnány s dostupnou specifikací a určena jejich shodnost, případně byla provedena identifikace odlišných oblastí v daných implementacích.

Při vypracování práce bylo zjištěno, že by bylo vhodné implementovat unifikovaný nástroj pro porovnání výsledků jednotlivých modelů. Díky znalostem získaných při studiu z oboru strojového vidění a zpracování obrazu bylo využito programovacího jazyka Python a jeho knihovny OpenCv. Tato knihovna obsahuje široké množství funkcí pro strojové zpracování obrazu. Výstupem tohoto nástroje - porovnávače jsou grafy s maskou jednotlivých oblastí a procentuální/početní zastoupení jednotlivých skupin barev výsledků.

Posledním bodem zadání této práce bylo vytvoření nové metodiky pro identifikaci kritických parametrů daných scénářů. Bylo využito již použitých metrik a dle jejich výsledků vytvořen nový rozhodovací algoritmus. Jeho výstupem je graf, obsahující oblasti s rozdílnou mírou kritičnosti. Dále byly tyto oblasti automaticky detekovány, zejména jejich hranice a průnik s ostatními oblastmi kritičnosti. Průnikem vybraných oblastí byly detekovány rozsahy parametrů, které jsou rizikové. Díky těmto grafickým výstupům je tester lépe schopen identifikovat potenciálně rizikové oblasti. Metoda také navrhuje rozsahy potenciálně kritických parametrů, na které by se tester měl zaměřit při podrobnější analýze. Nástroj pro identifikaci hranic oblastí lze použít pro grafické výstupy všech metrik kritičnosti implementovaných v tomto modelu.

Konceptuální model navržený a implementovaný v této práci je možné použít v praxi pro různé modely vozidel, systémů a scénářů. Dále je možné tento model rozšířit o další vyhodnocovací periferie, dynamiku vozidel či implementaci prostředí jako může být například povrch vozovky či povětrnostní podmínky. Věřím, že tento model pomůže

při vyhodnocování nových bezpečnostních systémů a zároveň napomůže k vytvoření bezpečnějšího provozu na našich silnicích.

Reference

- [1] ACKODrive. Autonomous cars: Levels of autonomous driving explained. <https://ackodrive.com/car-guide/autonomous-cars-and-levels-of-autonomous-driving/>, 2022. (cit. 18.4. 2024).
- [2] United Nations Economic Commission for Europe. Proposal for the 01 series of amendments to un regulation no. 157 (automated lane keeping systems). <https://unece.org/sites/default/files/2022-05/ECE-TRANS-WP.29-2022-59r1e.pdf>, 2022. (cit. 18.4. 2024).
- [3] Basil Papadopoulos Konstantinos Mattas, George Botzoris. Safety aware fuzzy longitudinal controller for automated vehicles. <https://www.sciencedirect.com/science/article/pii/S2095756421000635>, 2020. (cit. 3.5. 2024).
- [4] SAEinternational. Sae levels of driving automation™ refined for clarity and international audience. <https://www.sae.org/blog/sae-j3016-update>, 2021. (cit. 18.4. 2024).
- [5] Konstantinos Mattas a kolektiv. Driver models for the definition of safety requirements of automated vehicles in international regulations. application to motorway driving conditions. <https://www.sciencedirect.com/science/article/pii/S0001457522001798>, 2022. (cit. 18.4. 2024).
- [6] Lukas Westhofen a kolektiv. Criticality metrics for automated driving: A review and suitability analysis of the state of the art. <https://doi.org/10.1007/s11831-022-09788-7>, 2023. (cit. 3.5. 2024).
- [7] Heinisuo Olli-Pekka and team OpenCV. OpenCV Python packages. <https://github.com/opencv/opencv-python>, 9 2016. citováno 3.5. 2024.
- [8] Charles R. Harris and team. Array programming with NumPy. *Nature*, 585:357–362, 2020. (cit. 3.5. 2024).
- [9] Wikipedie. Hsv. <https://cs.wikipedia.org/wiki/HSV>, 2024. (cit. 18.4. 2024).
- [10] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017. (cit. 3.5. 2024).
- [11] The MathWorks Inc. Automated driving toolbox. <https://www.mathworks.com/help/driving/>, 2022. (cit. 6.5. 2024).
- [12] The MathWorks Inc. Vehicle dynamics blockset. <https://www.mathworks.com/help/vdynblks/>, 2022. (cit. 6.5. 2024).

- [13] The MathWorks Inc. `boundary`. <https://www.mathworks.com/help/matlab/ref/boundary.html>, 2024. (cit. 6.5. 2024).
- [14] The MathWorks Inc. `intersect`. <https://www.mathworks.com/help/matlab/ref/double.intersect.html>, 2024. (cit. 6.5. 2024).

Obsah příloženého CD

- **Složka porovnavac**
 - Zdrojový kód porovnávače
 - Vstupní data porovnávače
- **Složka deceleration**
 - Složka script
 - Zdrojové kódy Matlab implementace obou modelů řidiče
 - Datasets z jednotlivých simulacích
 - Složka simulink
 - Zdrojové kódy implementace obou modelů řidiče
 - Simulink model
 - Datasets z jednotlivých simulacích
- **Složka cut-in**
 - Složka script
 - Zdrojové kódy Matlab implementace obou modelů řidiče
 - Datasets z jednotlivých simulacích
 - Složka simulink
 - Zdrojové kódy implementace obou modelů řidiče
 - Simulink model
 - Datasets z jednotlivých simulacích