



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

DEPARTMENT OF COMPUTER SYSTEMS

**VÝPOČETNÍ MODEL A ANALÝZA SAMOČINNĚ
ŘÍZENÉHO VOZIDLA**

COMPUTATIONAL MODEL AND ANALYSIS OF SELF-DRIVEN VEHICLE

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. MILAN GARDÁŠ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JOSEF STRNADEL, Ph.D.

BRNO 2019

Zadání diplomové práce



20627

Student: **Gardáš Milan, Bc.**
Program: Informační technologie Obor: Matematické metody v informačních technologiích
Název: **Výpočetní model a analýza samočinně řízeného vozidla**
Computational Model and Analysis of Self-Driven Vehicle
Kategorie: Modelování a simulace

Zadání:

1. Zdokumentujte požadavky, problémy, pojmy a principy související s činností samočinně řízeného vozidla; proveďte detailní rešerši v této oblasti. Navrhněte vhodnou abstrakci samočinně řízeného vozidla, jeho okolí a aspektů klíčových z hlediska jeho řízení a chování.
2. Proveďte detailní rešerši v oblasti prostředků výpočetního modelování systémů a analýzy jejich vlastností a zvolte prostředky vhodné k řešení zadané práce.
3. Pomocí zvolených prostředků vytvořte výpočetní model řízení a chování samočinně řízeného vozidla a výpočetní model jeho okolí s cílem analyzovat vliv způsobu řízení vozidla na chování vozidla v daných podmínkách.
4. Vlastnosti modelu ověřte v několika vhodně zvolených situacích (např. způsoby řízení, typy okolí, podmínky).
5. Diskutujte a zhodnoťte možnosti vytvořeného modelu z hlediska analýzy vlivů zmíněných v bodu 3 a navrhněte možné směry pokračování v projektu.

Literatura:

- Dle pokynů vedoucího.

Při obhajobě semestrální části projektu je požadováno:

- Splnění bodů 1 a 2 zadání, vytvoření základního výpočetního modelu samočinně řízeného vozidla.

Podrobné závazné pokyny pro vypracování práce viz <http://www.fit.vutbr.cz/info/szz/>

Vedoucí práce: **Strnadel Josef, Ing., Ph.D.**
Vedoucí ústavu: Sekanina Lukáš, prof. Ing., Ph.D.
Datum zadání: 1. listopadu 2018
Datum odevzdání: 22. května 2019
Datum schválení: 26. října 2018

Abstrakt

Diplomová práce se zabývá samočinně řízenými vozidly. Nejprve obsahuje popis současného vývoje těchto vozidel, tedy z čeho se skládají, jak fungují, a je nastíněn jejich další možný vývoj. Dále se zde nachází popis možných nástrojů použitelných pro modelování. Práce obsahuje návrh, popis vývoje a testování simulačního modelu ve verifikačním prostředí UPPAAL Stratego. Výsledný model je systémem vzájemně komunikujících časovaných automatů. Analýza vlastností modelu je založená na metodě statistického ověřování modelu. Model umožňuje zkoumat chování autonomního vozidla v situacích, které odpovídají běžnému provozu.

Abstract

This thesis discusses autonomous vehicles. At first it contains describing development of these type of vehicles, how they work and discuss their future development. Further it describe tools which can be used for create model of autonomous vehicle. The thesis includes design, description of the development and testing of the model in the UPPAAL Stratego verification environment. The resulting model is a system of intercommunicating timed automata. The analysis of the model properties is based on the method of statistical verification. The model allows us to investigate behavior of an autonomous vehicle in situations which correspond to regular traffic.

Klíčová slova

autonomní vozidlo, modelování, statistické ověřování modelu, stochastické časované automaty, UPPAAL Stratego

Keywords

autonomous vehicle, modelling, statistical model checking, stochastic timed automata, UPPAAL Stratego

Citace

GARDÁŠ, Milan. *Výpočetní model a analýza samočinně řízeného vozidla*. Brno, 2019. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Josef Strnadel, Ph.D.

Výpočetní model a analýza samočinně řízeného vozidla

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Josefa Strnadela Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Milan Gardáš
22. května 2019

Poděkování

Tímto bych rád poděkoval svému vedoucímu Ing. Josefovi Strnadelovi, Ph.D. za odborné vedení, vstřícnost při konzultacích a cenné rady, které mi pomohly tuto práci vyhotovit.

Obsah

1	Úvod	3
2	Autonomní vozidla	4
2.1	Historie	4
2.2	Popis autonomního vozidla	5
2.3	Senzory autonomních vozidel	6
2.3.1	Lidar	6
2.3.2	Inerciální měřicí jednotka	8
2.3.3	Global Positioning System	9
2.3.4	Radar	10
2.3.5	Kamery	11
2.4	Způsoby řízení autonomních vozidel	11
2.4.1	Normální jízda	12
2.4.2	Předjíždění jiného vozidla	12
2.4.3	Parkování	12
2.5	Současný stav	13
2.6	Bezpečnost vozidel	15
2.7	Možný budoucí vývoj	17
3	Prostředky pro modelování systémů	19
3.1	Teorie	19
3.2	Nástroje	20
3.2.1	UPPAAL	20
3.2.2	MATLAB / Simulink	21
3.2.3	Dymola / Modelica	21
3.2.4	Jade	22
3.2.5	SIMLIB	23
4	Zvolené realizační prostředky	24
4.1	UPPAAL	24
4.1.1	UPPAAL SMC	26
4.1.2	UPPAAL Stratego	27
4.2	Případová studie	28
5	Návrh a popis modelu	30
5.1	Návrh modelu	30
5.2	Použité abstrakce	31
5.3	Popis jednotlivých automatů	31

5.3.1	Řízení simulace	31
5.3.2	Výběr reakce autonomního vozidla	32
5.3.3	Řízení neautonomních vozidel	38
5.4	Popis konstant, proměnných a funkcí	39
6	Vyhodnocení vlastností modelu	41
6.1	Reakce vozidla na jiné vozidlo	41
6.1.1	Udržování bezpečné vzdálenosti	41
6.1.2	Předjíždění jednoho vozidla	43
6.1.3	Předjíždění jednoho vozidla, pokud se nachází vozidlo i v protisměru	45
6.1.4	Pokus o předjetí více než jednoho vozidla	47
6.1.5	Pokus o předjetí více než jednoho vozidla s vozidlem v protisměru .	49
6.2	Experimenty s ověřením konkrétního cíle	51
6.2.1	Předjíždění vozidla, které začne zrychlovat	51
6.2.2	Reakce na chybu senzoru	52
6.2.3	Reakce na dopravní značky ovlivňující rychlost	53
6.3	Výpočetní náročnost	55
7	Závěr	56
	Literatura	57
A	Obsah příloženého CD	60

Kapitola 1

Úvod

Jízda automobilem je v dnešní době jedním z nejrozšířenějších, a také nejoblíbenějším způsobem pro přepravu z bodu A do bodu B. Vlastnictví osobního automobilu se v Evropě a Severní Americe považuje v podstatě za samozřejmost, a není výjimkou, pokud rodina vlastní dvě nebo více osobních automobilů. Tento velký rozmach přepravy sebou ovšem také přináší úskalí v podobě dopravních nehod. Podle údajů Evropské hospodářské komise OSN [13], bylo na území USA od roku 2005 vždy minimálně 1 500 000 nehod, které zapříčinily úmrtí téměř 400 000 lidí, což je pro srovnání více lidí, než kolik lidí má uvedeno trvalé bydliště v Brně.

Výrobci automobilů se samozřejmě snaží neustále vylepšovat automobily tak, aby byly bezpečnější. Je ovšem otázka, zda nejslabším článkem automobilu není člověk, a tedy by jeho nahrazením zásadně nebylo nehod a úmrtí zaviněných automobily. V dnešní době již v několika málo městech v USA probíhá testování autonomních vozidel. Bohužel ani jim se nevyhnuly smrtelné dopravní nehody, protože osoba, která měla dohlížet na správnou funkci automobilu, se nevěnovala věcem, kterým měla. Tyto moderní technologie ovšem mají také potenciál jednou nahradit lidi, kteří se řízením automobilu živí, a dá se předpokládat, že je tato situace moc nenadchne. Výjimkou také nejsou situace¹, kdy jsou tyto testovací vozidla napadána lidmi, kteří na ně hází kamení nebo jim propichují pneumatiky.

Tato diplomová práce se bude zabývat všeobecným popisem samočinně řízených vozidel v kapitole 2. Dále budou představeny možné prostředky pro modelování a simulaci systémů v kapitole 3. Kapitola 4 popisuje nástroj, který byl zvolen k praktické realizaci této práce. V kapitole 5 je popsán návrh a implementace simulačního modelu autonomního vozidla. Nakonec bude v kapitole 6 vytvořený model podrobený testům ve kterých bude zjištěno, zda model splňuje očekávání autora.

¹<https://www.nytimes.com/2018/12/31/us/waymo-self-driving-cars-arizona-attacks.html>

Kapitola 2

Autonomní vozidla

Tato kapitola se zabývá popisem autonomních vozidel. Nejprve se snaží o vysvětlení toho, co je vlastně samočinně řízené vozidlo. Poté stručně popisuje jejich historii, popisuje rozdělení jednotlivých typů automobilů podle míry interakce od asistenčních systémů. Dále objasňuje, jaké důležité senzory vlastně takové vozidlo obsahuje, jak funguje jejich řízení, a jaký je současný stav vývoje těchto vozidel. Nakonec se snaží o podání autorova pohledu na budoucnost tohoto segmentu.

2.1 Historie

Koncept jednoho z prvních samočinně řízených automobilů byl představen firmou General Motors v roce 1939 na světové výstavě v New Yorku. Tento koncept byl ovládán pomocí elektromagnetických polí, které vytvářely hřebíky umístěné v cestě.

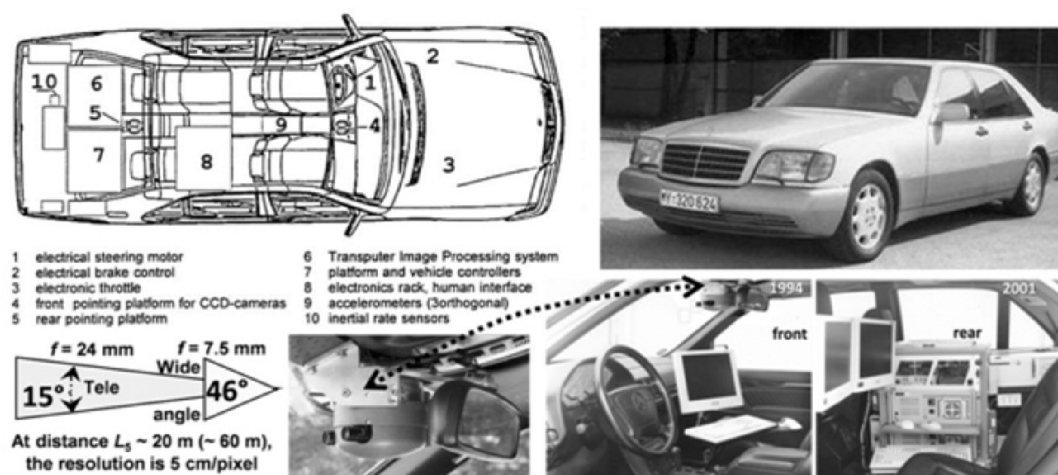
V roce 1958 vytvořil Chevrolet pravděpodobně první autonomní vozidlo v USA. Bylo vybaveno speciálním rádiovým přijímačem spolu se zvukovými a vizuálními zařízeními, která mohla ovládat řízení, zrychlování a brzdění vozidla. Světla okolo vozovky sloužila k určení polohy a rychlosti vozidla.



Obrázek 2.1: Chevrolet z roku 1958. Stejný typ byl použit pro autonomní vozidlo².

V roce 1979 byl na Stanfordově univerzitě vytvořen robot Cart, jenž byl radiově spojen s velkým sálovým počítačem. Díky tomuto spojení dokázal zmapovat a projet kolem objektů, které mu byly postaveny do cesty. Zároveň také dokázal monitorovat svou vlastní polohu.

V roce 1995 Mercedes-Benz třídy S navržený Ernstem Dickmannsem a jeho týmem z Mnichovské univerzity podnikl cestu z Mnichova do Kodaně, jež měřila 1590 kilometrů. Tento automobil využíval počítačové vidění a mikroprocesory pro paralelní výpočty. Při cestě byla dosažena nejvyšší rychlost 175 *km/h*. Umožňoval jízdu v normální dopravě a manévrování mezi ostatními automobily. Největší ujetá vzdálenost bez zásahu člověka byla 158 kilometrů.



Obrázek 2.2: Autonomní vozidlo Mercedes³.

2.2 Popis autonomního vozidla

Samočinně řízené vozidlo se na první pohled od klasického osobního automobilu téměř vůbec neliší. Také se skládá ze součástí, jako jsou motor, převodovka, kola a brzdy. Hlavní rozdíl spočívá v tom, že takové vozidlo ke svému provozu v ideálním případě nepotřebuje řidiče, proto musí být poněkud chytřejší než obyčejný automobil. Dosáhnutí této inteligence se provádí pomocí získávání dat z mnoha různých druhů senzorů, kde ty nejpodstatnější jsou popsány níže. Tyto data jsou vyhodnocována a s jejich pomocí se ovládá natočení kol, zrychlování či brzdění. Autonomní vozidla lze rozdělit do šesti úrovní [20].

Úroveň 0 - Žádná automatizace

V tomto případě se nejedná v podstatě o žádnou automatizaci. Plnou kontrolu nad vozidlem má člověk, který všechny systémy ovládá sám. Vůz jej může maximálně varovat, ale nijak nezasahuje do řízení. Automobil této kategorie může obsahovat například systém rozpoznávání dopravních značek nebo systém pro varování o opuštění jízdního pruhu, kde může vyslat zvukový signál nebo zavibrovat se sedačkou.

²Převzato z <https://www.velocetoday.com/self-drive-cars-and-you-a-history-longer-than-you-think/>

³Převzato z <https://www.engineering.com/DesignerEdge/DesignerEdgeArticles/ArticleID/12665/The-Road-to-Driverless-Cars-1925-2025.aspx>

Úroveň 1 - Podpora řidiče

Automobily spadající do této kategorie mají systémy, které mohou aktivně zasáhnout do řízení. Takových systémů může být i několik, avšak systémy nespolupracují dohromady, tedy nejsou vzájemně kombinovány. Řízení má stále na starosti člověk, který je za auto zodpovědný. Systémy spadající do této kategorie mohou být například systém automatického brzdění, adaptivní tempomat nebo asistent pro udržování v jízdním pruhu.

Úroveň 2 - Částečná automatizace

U vozidel, která patří do této kategorie, dokážou elektronické systémy pomocí senzorů a kamer převzít více aspektů řízení místo řidiče. Například je možné zkombinovat adaptivní tempomat a systém pro udržování vozidla v jízdním pruhu. Do této kategorie také spadá systém pro automatické parkování. Osoba sedící za volantem ovšem musí být kdykoliv připravena zareagovat a převzít řízení. Tyto systémy také fungují také pouze za omezených podmínek.

Úroveň 3 - Podmíněná automatizace

U tohoto typu vozidla již systémy mohou za určitých podmínek plně převzít kontrolu nad vozidlem. Řidič nemusí mít ruce na volantu ani nemusí sledovat cestu. Ovšem stále musí být připraven na upozornění na převzetí řízení. Taková vozidla budou sama umět zrychlovat, zpomalovat, zatáčet a vyhnout se překážce, což je zajištěno použitím velkého množství senzorů. Toto bude možné jen na vybraných úsecích, typicky na moderních dálnicích s dobře vyznačenými jízdními pruhy.

Úroveň 4 - Vysoká automatizace

Automobily z této kategorie budou mít rozšířené fungování systémů z předchozí úrovně. Již bude možné nechat automobil řídit i ve městě, protože se předpokládá komunikace s dopravním značením, díky čemuž bude možné regulovat dopravu. Řidič stále bude moci převzít kontrolu nad vozidlem, ale nebude to téměř vůbec potřeba. Situace, kdy by řidič mohl převzít kontrolu nad řízením, nastane buď při selhání systémů pro řízení, nebo nepříznivé povětrnostní podmínky.

Úroveň 5 - Plná automatizace

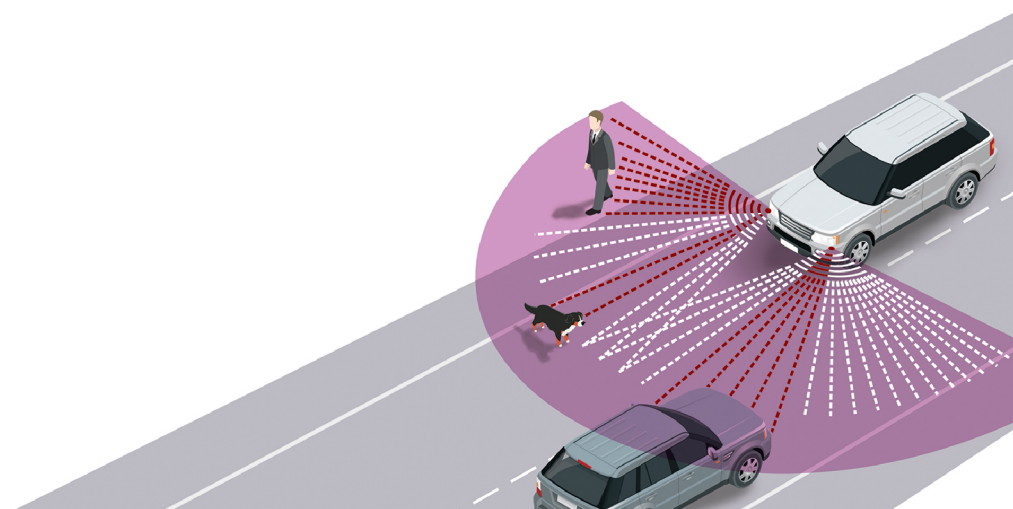
Automobil zvládne veškeré jízdní situace na všech cestách. Jedinou úlohou člověka bude zadání cíle cesty a vozidlo už se postará o zbytek.

2.3 Senzory autonomních vozidel

V této části práce jsou popsány důležité senzory, které jsou v současné době nezbytnou součástí každého autonomního vozidla.

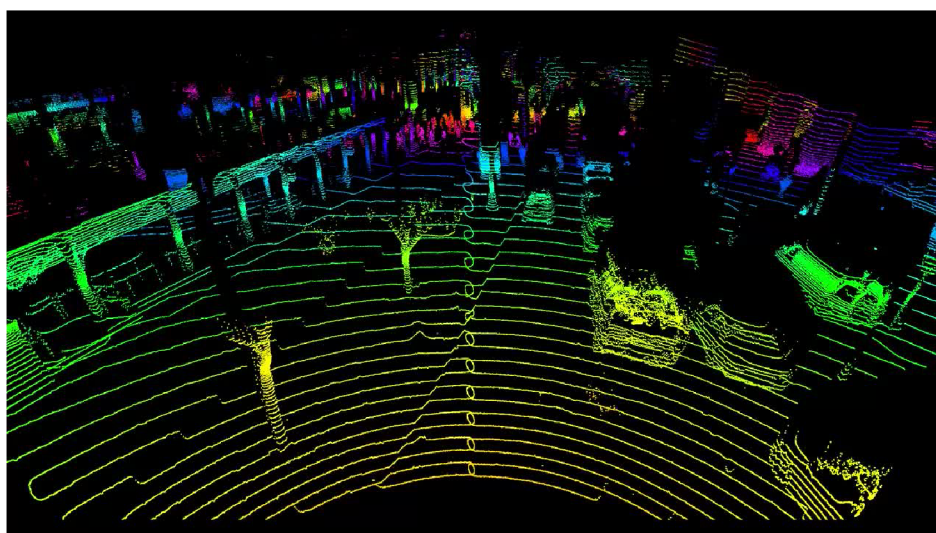
2.3.1 Lidar

Lidar (Light Detection And Ranging) je metoda, která používá ultrafialové, viditelné nebo infračervené světlo pro měření vzdálenosti [22]. Lidar nejprve vyzařuje laserový paprsek, který se odrazí od objektu zpátky k lidar. Princip činnosti je zobrazen na obr. 2.3. Rozdíl času mezi vysláním a přijmutím paprsku slouží k dopočítání vzdálenosti mezi objektem a lidarem. Technologie je velmi podobná sonaru, který používá ultrazvukové vlny, a radaru,



Obrázek 2.3: Obrázek znázorňující princip lidarů. Nejprve jsou vyslány laserové paprsky (bílé čáry). Paprsky, které narazí do nějakého objektu, se od něj odrazí (červené čáry) a vrátí se do lidarů. Z těchto paprsků je určena vzdálenost objektu od zařízení s lidarem⁴.

jenž využívá rádiové vlny. Koncovka -dar ve slově lidar pochází právě ze slova radar. Velkou výhodou je možnost zaměření celé řady materiálů včetně nekovových předmětů, hornin, oblačnosti či dokonce jednotlivých molekul. Mezi další výhody patří možnost mapování ve velkém rozlišení, protože například při leteckém průzkumu lze mapovat terén s rozlišením až 30 centimetrů. V autonomním vozidle plní úlohu detektoru překážek, kde funguje jako rotující snímač, který mapuje prostředí kolem vozidla a vytváří mapu objektů kolem vozidla. Jak vypadá taková mapa je možné vidět na obr. 2.4. Tato mapa poskytuje údaje o tom, kde se v okolí nacházejí potenciální překážky, v jakém postavení je oproti nim vozidlo a jaký je rozdíl vzdáleností.



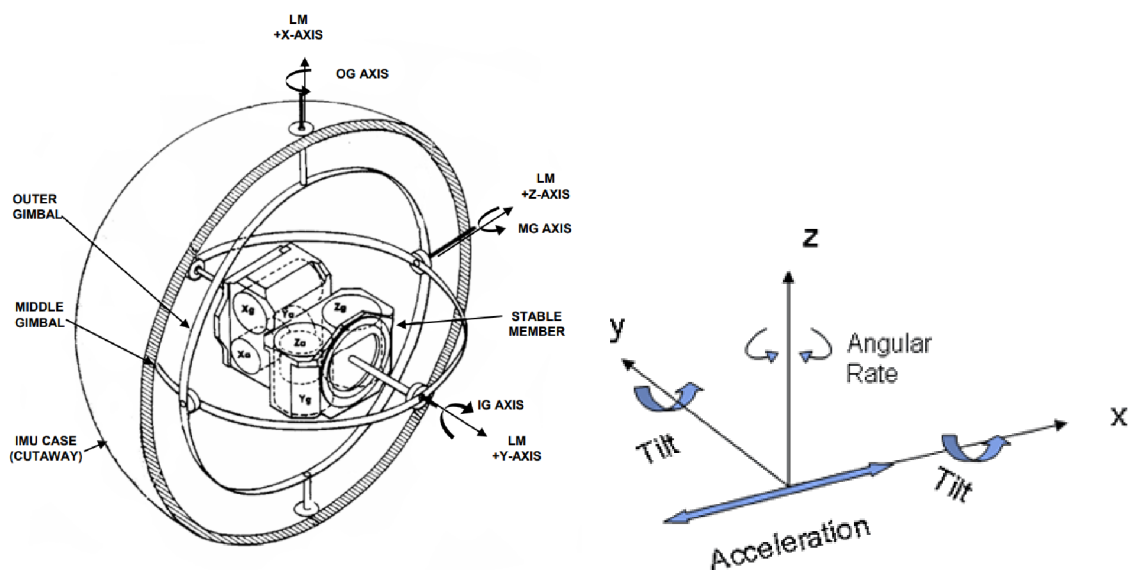
Obrázek 2.4: Obrázek znázorňující data získaná pomocí lidarů⁵.

⁴Převzato z thestack.com/wp-content/uploads/2017/03/EPC_LiDAR_car_for_print_CMYK.jpg

S pomocným softwarem je možné analyzovat vlastnosti překážek. Mezi tyto vlastnosti patří například rychlost či směr pohybu, pokud je překážka detekována jako automobil. Na základě těchto informací je možné varovat řidiče nebo může vozidlo samo zasáhnout do řízení. Vedle lidarů se ve vozidle nachází také již zmíněný radar a sonar, to protože lidar špatně pracuje v nepříznivých povětrnostních podmínkách. Mezi další nevýhody patří vyšší cena a malý maximální dosah, který se pohybuje mezi 500 až 2000 metry. Na větší vzdálenosti je lepší radar, protože u překážky, která je od nás vzdálena dva kilometry nepotřebujeme vědět, o jakou překážku se jedná, ale je potřeba vědět, že se tam nějaká nachází.

2.3.2 Inerciální měřicí jednotka

Inerciální měřicí jednotka (IMU) je elektronické zařízení, které obsahuje akcelerometry, gyroskopy, magnetometr a matematický koprocesor. Akcelerometr je zařízení pro měření pohybové akcelerace v jedné ose. Typické je ovšem použití trojosého akcelerometru, který měří akceleraci ve třech osách. Gyroskop je zařízení, které měří otáčení v rovině a úhlovou rychlost. Tyto údaje dokáže zjistit bez ohledu na zrychlení tělesa a gravitaci. Magnetometr je součástí, jejíž úkol je měřit směr a sílu magnetického pole. V IMU se používá ke kalibraci měření úhlu okolo osy z. Matematický koprocesor se stará o automatickou kalibraci všech členů, čištění signálu a výpočet co nejpřesnějších úhlů otáčení.



Obrázek 2.5: Obrázek nalevo zobrazuje průřez IMU⁶. Obrázek napravo znázorňuje veličiny, které je možné měřit na jednotlivých osách⁷.

Tato jednotka může nahradit GPS, které může být blokováno budovami nebo tunely, protože její výhodou je, že nepotřebuje žádný externí signál. Její nevýhodou ovšem je to, že počáteční pozice vozidla musí být zadána ručně nebo určena pomocí GPS navigace, protože IMU nemá žádnou informaci o počáteční pozici.

⁵Převzato z nytimes.com/2017/05/25/automobiles/wheels/lidar-self-driving-cars.html

⁶Převzato z https://en.wikipedia.org/wiki/Inertial_measurement_unit

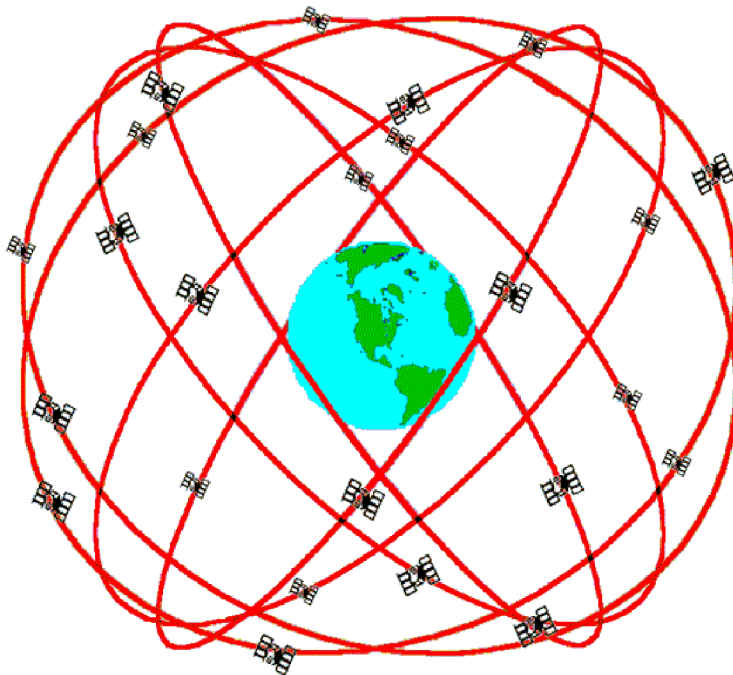
⁷Převzato z <http://home.iitk.ac.in/~vyas/ABS/IMU1.html>

Je možné také spojit několik IMU do jednoho zařízení, které se poté označuje jako inerciální navigační systém (INS) [23]. INS spolu s GPS jsou často integrovány do jednoho sledovacího systému, kde jsou využity jejich silné prvky. INS získává data o pozici a rychlosti, jež jsou průběžně vypočítávány integrací akcelerace a úhlové rychlosti. Tato integrace ovšem způsobuje nepřesnost v dlouhodobém měřítku, protože malé chyby jednotlivých senzorů se akumulují s časem. Výhoda použití INS je v rychlosti aktualizace dat, která je oproti GPS řádově lepší.

2.3.3 Global Positioning System

Global Positioning System (dále jen GPS) je globální družicový systém provozovaný Ministerstvem obrany USA. Pomocí tohoto systému je možné určit polohu přijímače, který se nachází kdekoli na zemi s přesností na metry, a čas s přesností na nanosekundy. GPS je složena ze tří segmentů: kosmického, řídicího a uživatelského.

Kosmický segment je tvořen 24 satelity, které jsou rozmístěny kolem Země na 6 oběžných drahách. Jejich oběžné dráhy je možné si prohlédnout na obr. 2.6. Toto rozmístění zajišťuje, že z každého místa na Zemi jsou v každém okamžiku minimálně 4 satelity v přímé viditelnosti. Každý satelit obsahuje několik velmi přesných atomových hodin, vysílač a přijímač umožňující komunikaci a další řídicí elektroniku.



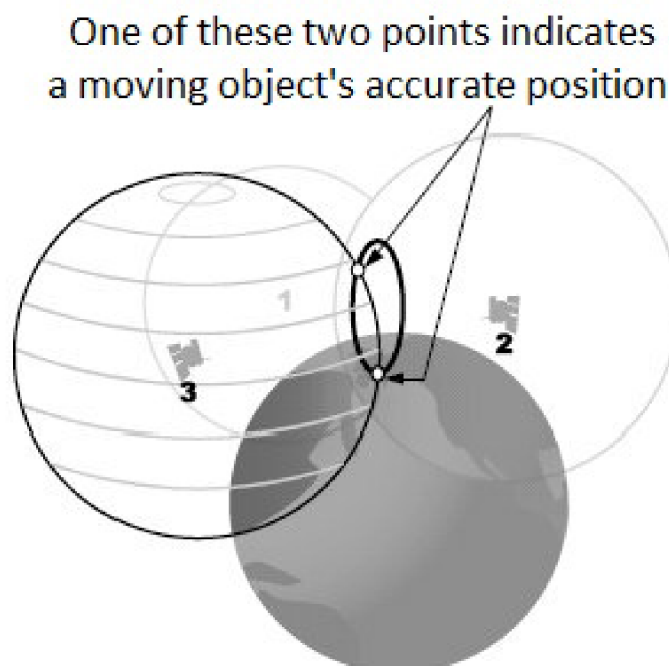
Obrázek 2.6: Oběžné dráhy satelitů okolo Země⁸.

Řídicí segment monitoruje kosmický segment. Je tvořen monitorovacími stanicemi, které jsou umístěny po celém světě, a hlavní řídicí stanicí, jež leží v Colorado Springs. Obyčejné stanice sbírají data ze satelitů, která zasílají řídicí stanici, kde se zpracují a vypočítají informace o oběžných drahách a korekci času. Tyto informace jsou posléze zaslány zpět satelitům, které je vysílají na GPS přijímače.

⁸Převzato z https://www.kip.zcu.cz/kursy/svt/svt_www/3_soubory/3_1.htm

Uživatelský segment tvoří uživatelé s GPS přijímači, kteří přijímají signál z jednotlivých družic, díky čemu je možné vypočítat polohu antény, nadmořskou výšku, případně určit přesný čas a datum.

Pro určení polohy si přijímač nejprve vypočte vzdálenost, která jej dělí od několika okolních družic. Podle pravidel geometrie, pokud je známá vzdálenost k jednomu satelitu, pak přijímač leží někde na povrchu koule s poloměrem rovným dané vzdálenosti, jejíž střed tvoří daná družice. Pokud známe také vzdálenost ke druhému satelitu, tak se prostor, kde se přijímač může nacházet, omezí na kružnici. Konečně pokud známe i vzdálenost k dalšímu satelitu, tak z průniku koulí, jejichž středy tvoří jednotlivé satelity, získáme dva body, kde jeden je naprosto nesmyslný, protože je buď daleko ve vesmíru, nebo naopak v hluboko v zemi, a druhý udává naši polohu na zeměkouli. Tomuto principu se říká trilaterace, která je graficky znázorněna na obr. 2.7.



Obrázek 2.7: Princip trilaterace⁹.

V praxi je ovšem nutné použít informace ze čtyř satelitů, protože s měřením a počítáním vzdálenosti vznikají chyby. Chyba může také vzniknout odchýlením družice od své dráhy, proto jsou vybaveny pomocnými motory, aby byla případná odchylka korigována.

2.3.4 Radar

Je zařízení určené k vyhledávání cílů, určení jejich polohy, případně i jejich identifikaci pomocí elektromagnetického záření [25]. Radar vyrábí opakované rádiové impulsy o vysokém výkonu, které vysílá do prostoru anténou, tyto impulsy se poté odrazí od překážky a vrátí se zpět do antény. Z rozdílu časů mezi vysláním a přijetím impulsu je možné vypočítat vzdálenost objektu od radaru. Aby došlo k odrazu impulsu, musí být vlnová délka elektromagnetické vlny menší než rozměr cíle. Proto radary pracují většinou na frekvencích řádu

⁹Převzato z <https://www.svethardware.cz/jak-funguje-gps/21826-5>

GHz. U autonomních vozidel se používají dvě frekvence konkrétně 24 GHz a 77 GHz [7]. Ve vozidlech se uplatňují tři druhy radarů, které jsou rozděleny podle jejich dosahu, konkrétně se jedná o radar na krátkou, střední a dlouhou vzdálenost. Tyto radary se kromě detekce objektů používají pro pokročilé systémy pro podporu řízení (ADAS).

Radar na krátkou vzdálenost má dosah mezi 0.2 m až 30 m. Typicky se používá v parkovacích senzorech.

Radar na střední vzdálenost pracuje v rozsahu 30–80 m. Využití najde v systému pro nouzové brzdění.

Radar na dlouhou vzdálenost se používá jako senzor pro udržování bezpečné vzdálenosti mezi vozidly, dále také pro detekci silničního značení. Dosah tohoto typu radaru je od 80 m do zhruba 200 m.

2.3.5 Kamery

Kamery jsou v dnešní době podstatnou součástí moderních automobilů. Využívá se buď kamerová síť z několika jednotlivých kamer, které zachycují okolí vozidla ve 360°, nebo přímo sférická kamera. Z obrazu kamer je možné pomocí metod počítačového vidění získat podstatné informace o okolí kolem vozidla. Počítačové vidění je třeba chápat jako soubor dílčích technik pro napodobení vidění spíše než jako komplexní řešení náhrady vidění [6]. Můžeme například zjistit velikost a vzdálenost objektu, který kamera zaznamenala. Dále nám může pomoci klasifikovat zachycené objekty do různých skupin, například rozpoznání psa a kočky, s čímž má lidar problémy.



Obrázek 2.8: Sférická kamera Ladybug5¹⁰.

Kamery mohou být užitečné například při detekci dopravních značek, protože lidar i radar sice dokáže detekovat dopravní značku, ale například značky upravující maximální dovolenou rychlost na 70 km/h nedokáže zjistit žádné informace. Na druhé straně obraz z kamery, která danou značku zaznamená, se vyhodnotí pomocí počítačového vidění, které dokáže přečíst, o jakou značku se jedná, díky čemu se poté upraví rychlost vozidla na požadovanou míru.

2.4 Způsoby řízení autonomních vozidel

Tato část popisuje chování autonomních vozidel v různých situacích.

¹⁰Převzato z <https://www.ptgrey.com/ladybug5-360-degree-usb3-spherical-camera-systems>

2.4.1 Normální jízda

Pro normální jízdu po vozovce je potřeba, aby vozidlo vědělo, co se nachází v jeho okolí. Tento průzkum se provádí dvěma způsoby. První možností je použití 2D a 3D lidarů, které dovedou poskytnout poměrně komplexní obrázek o okolním prostředí [8]. Druhou možností je využití kamer, které jsou ve velké počtu umístěny ve vozidle, jejichž obraz je následně zpracováván pomocí počítačového vidění [10]. V praxi se typicky tyto dva přístupy kombinují, pro větší spolehlivost, protože například z údajů získaných pomocí kamer nemusí být přesně odhadnutelné, v jaké vzdálenosti se daný objekt nachází. Dále pokud jedeme za jiným vozidlem, tak auto musí být schopné zjistit jeho rychlost, aby případně mohlo být schopno upravit tu svoji na takovou úroveň, kdy bude v bezpečné vzdálenosti. Také si musí hlídat, zda se stále nachází ve svém jízdním pruhu a zda nepřekračuje povolenou rychlost, k čemuž využívá kamery a zpracování jejich obrazu pomocí počítačového vidění.

2.4.2 Předjíždění jiného vozidla

Pro předjíždění jiného vozidla je možné využít fuzzy logiku [12]. Pro jízdu v jednom pruhu nebo předjíždění do levého vedlejšího jízdního pruhu stačí pouze čtyři pravidla fuzzy logiky. Konkrétně se jedná o tyto pravidla:

- IF `lateral_error` right THEN `steering` left
- IF `lateral_error` left THEN `steering` right
- IF `angular_error` right THEN `steering` left
- IF `angular_error` left THEN `steering` right

kde lingvistické proměnné `lateral_error` a `angular_error` mají lingvistické hodnoty `left` a `right`, kde každá má svou funkci příslušnosti, jejíž tvar udává, jak moc chceme, aby proměnné ovlivňovaly řízení. Pro předjíždění je potřebné zjistit hodnotu vzdálenosti, která je nutná pro úspěšné předjetí vozidla. Tato hodnota se dá vypočítat ze vztahu mezi rychlostí vozidla a vzdáleností mezi vozidly. U předjížděcího manévru se snažíme, aby byl minimalizován čas, který autonomní vozidlo pobývá ve druhém jízdním pruhu, a zároveň maximalizována bezpečnost předjížděcího manévru. Například pro rychlost 55 km/h je vzdálenost pro úspěšné předjetí 64 m . Vozidlo také musí brát do úvahy maximální povolenou rychlost, protože pokud bude v zóně s maximální povolenou rychlostí 90 km/h předjet vozidlo, které pojede 87 km/h , pak předjíždění postrádá smysl. Pokud by snad vozidlo chtělo pomalejší vozidlo předjet, je nutné zvážit, zda by mělo mít možnost porušit dopravní předpisy tím, že zrychlí na větší než povolenou rychlost, ale předjížděcí manévr bude trvat menší dobu.

2.4.3 Parkování

Pro automatické parkování je potřebná znalost trojice souřadnic (x, y, ϕ) , kde x a y jsou souřadnice středu zadních kol automobilu. ϕ je úhel karoserie vozidla. Pro výpočet jednotlivých souřadnic je možné použít následující rovnice:

$$x = v \cdot \cos\alpha \cdot \cos\phi \quad (2.1)$$

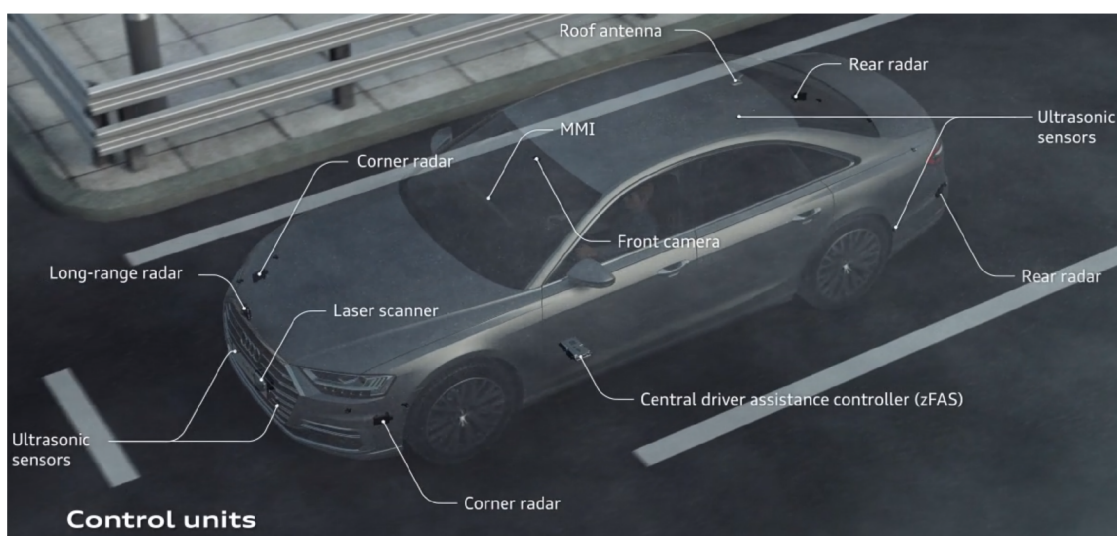
$$y = v \cdot \cos\alpha \cdot \sin\phi \quad (2.2)$$

$$\phi = \frac{v}{L} \sin\alpha \quad (2.3)$$

kde v je rychlost předních kol a L je rozvor kol vozidla. Pro úspěšný parkovací manévr je nutné, aby se pozice automobilu přizpůsobila danému parkovacímu místu. Vozidlo by mělo být rovnoběžně k parkovacímu místu. Samotný manévr se potom skládá několika operací prováděných v nízké rychlosti, které postupně posouvají vozidlo na parkovací místo [15].

2.5 Současný stav

V současné době se vyrábí mnoho automobilů, které mají asistenční prvky z kategorie 2. Ovšem existuje několik vozidel, která spadají do kategorie 3. Jedno z prvních sériově vyráběných vozidel je nejnovější model Audi A8, které obsahuje 41 asistenčních systémů pro rozpoznání a vyhodnocení aktuálního jízdního prostředí. Umístění senzorů ve vozidle je možné vidět na obr. 2.9. Tyto systémy jsou zde propojeny a tvoří inteligentní síť senzorů a algoritmů.



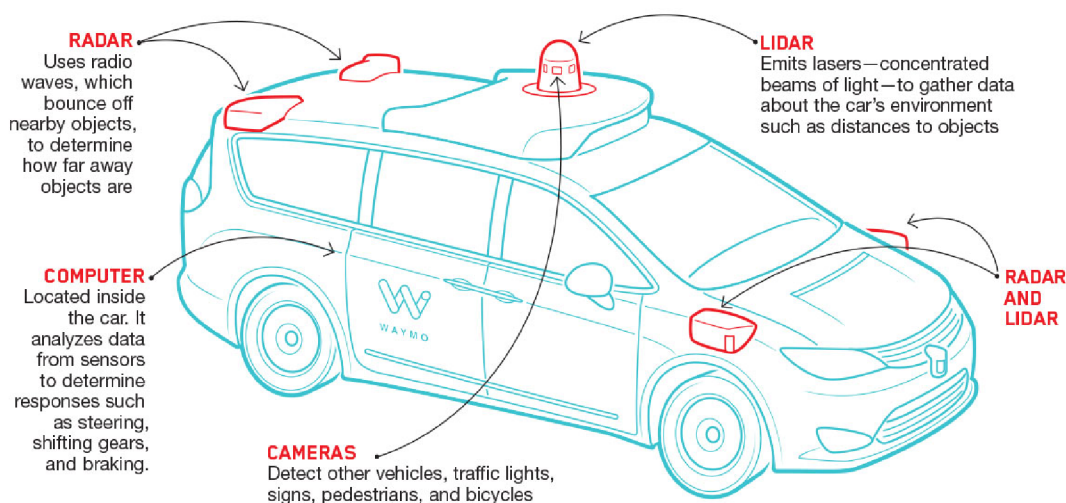
Obrázek 2.9: Senzory Audi A8, které je první sériově vyráběné vozidlo z kategorie 3¹¹.

Vozidlo disponuje funkcí Audi AI traffic jam pilot¹¹, což je první systém umožňující vysoce automatizovanou jízdu. Tento systém funguje tak, že pokud řidič přijede k místu s dopravní zácpou, potom řidič může zmáčknout tlačítko, které aktivuje tento systém, a odložit ruce z volantu a nohu z plynového pedálu, protože řízení v zácpě převezme počítač. K tomu, aby to bylo možné, je zapotřebí splnit ještě, aby vjezd do protisměru byl znemožněn zátarasy a silnice měla dopravní značení. Obě tyto podmínky by měla splňovat každá nová dálnice. Řízení pomocí tohoto systému probíhá až do rychlosti 60 km/h nebo pokud dopravní zácpa skončí. Poté je řidič upozorněn, aby opět převzal ovládání nad vozidlem, pokud je tato výzva řidičem ignorována, tak je vozidlo zastaveno u kraje vozovky.

Technologické společnosti, které nejsou přímo výrobci automobilů, se snaží k výrobě samočinně řízeného vozidla přistupovat jinou cestou. Nejprve zakoupí vozidlo, které splňuje jejich požadavky jako rozměry karoserie, dostatek prostoru v kufru a interiéru vozu, různé senzory nebo asistenční technologie. Do takového vozu nainstalují další senzory, které nejsou v jeho sériové výbavě. Společnosti, které dělají tento druh činnosti, jsou označovány jako

¹¹Převzato z <https://www.audi-technology-portal.de/en/electrics-electronics/driver-assistant-systems/audi-a8-audi-ai-traffic-jam-pilot>

výrobci technologií pro autonomní řízení vozidla. Mezi takové společnost patří Waymo. To používá více než tisíc vozidel značky Chrysler, do kterých jsou zabudovány senzory a systém pro autonomní řízení. Schéma automobilu této společnosti je na obr. 2.10.



Obrázek 2.10: Automobil Chrysler od společnosti Waymo¹².

Právě tato společnost dne 5. 12. 2018 spustila první veřejně dostupnou taxislužbu, v jejímž vozovém parku jsou pouze autonomní vozidla, ve kterých sice sedí řidič, ale ten zasáhne pouze v případě potřeby.

Také v České republice existuje společnost roboauto [21], která se snaží o vývoj autonomního vozidla. Toto vozidlo je k vidění na obr. 2.11. Ve svém automobilu využívá technologie pro detekci objektů pomocí neuronových sítí a umělou inteligenci, která vylepšuje své reakce po každé jízdě. Dále systém pro detekci vodorovného dopravního značení. Pro testování funkčnosti různých podsystémů vyvíjeného vozidla mimo komunikace využívá také počítačovou hru GTA V.

U všech druhů autonomních vozidel je zapotřebí velmi důkladné testování. To většinou probíhá mimo veřejné komunikace, protože každá země, která vůbec umožňuje testování tohoto typu vozidel má určenou přesnou oblast pro testování (výjimkou je například Nový Zéland, který umožňuje testovat na obyčejných silnicích za předpokladu, že bude dodržovaná bezpečnost a nebude narušen provoz po komunikaci¹⁴). Také není neobvyklé, že si daná společnost vybuduje vlastní komunikace. Takové testy jsou nutné, aby se zabránilo následným nehodám v normálním provozu, případně bylo zjištěno, jak se autonomní vozidlo chová v situaci, která je běžnou součástí dopravního provozu. Bohužel ani přes obrovské množství testování se autonomní vozidla nevyhnula dopravním nehodám, kdy některé dokonce skončily úmrtím člověka.

¹²Převzato z <https://scienceworld.scholastic.com/issues/2017-18/090417/hands-free-ride.html#1070L>

¹³Převzato z <https://www.roboauto.cz/>

¹⁴<https://www.transport.govt.nz/assets/Uploads/Our-Work/Images/T-Technology/Testing-Autonomous-Vehicles-in-New-Zealand.pdf>



Obrázek 2.11: Autonomní automobil od české společnosti roboauto¹³.

2.6 Bezpečnost vozidel

S postupným rozmachem autonomních vozidel je potřebné se zabývat otázkami jejich bezpečnosti. V tomto případě, ale nejsou myšleny události typu dopravní nehoda, kde následky budou v podstatě stejné, jako u nehod jen s automobily řízenými čistě lidskými řidiči, ale je nutné, se také zaměřit na zabezpečení systémů samočinně řízených vozidel. Útoky, které je možné vykonat na autonomní vozidla, se dělí do dvou kategorií, konkrétně na pasivní a aktivní útoky [26]. Pasivní útoky se zaměřují na získání informací a jejich vyhodnocení, zároveň je velmi těžká jejich detekce, protože neprovádí žádné změny dat. Aktivní útoky již provádějí změnu dat, čímž může vozidlo vykonat činnosti, které by se původními daty neprováděly.

O aktivní útok na systém detektoru čar a útok se snahou vzdáleného ovládnutí vozidla se v roce 2018 pokusila čínská firma Tencent Security. Tyto útoky prováděla na automobil Tesla S obsahující aktuální software.

Útoky na detektor čar na silnici

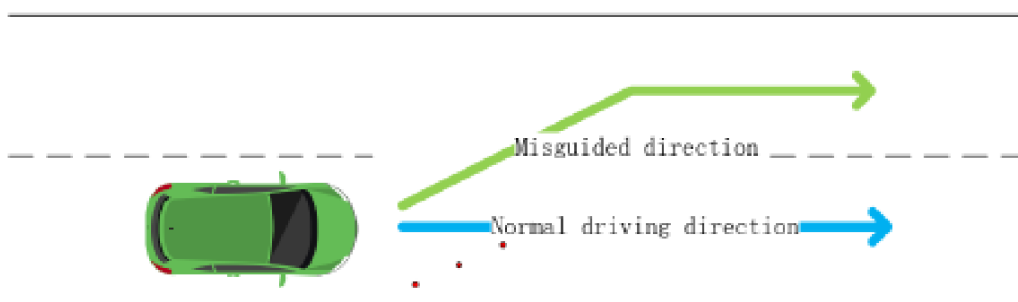
Detekce jízdních pruhů je u autonomního vozidla Tesla implementována pomocí počítačového vidění a následného zpracování pomocí neuronové sítě, která vytváří digitální mapu svého okolí. Na tento systém byly vytvořeny dva útoky. Oba útoky se snaží obelstít neuronovou síť pomocí úpravy okolí, čímž se do kamery dostávají upravené záběry. První útok se snaží zamaskovat značení parkovacího místa tím, že na jednu z čar je použit optimalizační algoritmus, který se jí snaží upravit tak, aby čára byla znehodnocena tak, že automobil přestane dané parkovací místo detekovat. Tato situace se nachází na obr. 2.12. Jako funkční řešení se ukázalo přidat čáře různý tvar skvrn, které vozidlo přestane vyhodnocovat jako čáru, ačkoliv člověk by takovou změnu jistě zaznamenal.

Druhý pokus o útok se týká naopak přidání čáry na vozovku, čímž je možné například donutit autonomní vozidlo ke změně jízdního pruhu. Velkým problémem tohoto útoku je ovšem nutnost fyzické modifikace silnice, po které projíždí vozidlo, takže by se dalo uvažovat o proveditelnosti tohoto útoku, pouze pokud autonomní vozidlo jezdí stále stejnou trasu, kdy ji útočník dokáže zjistit, na což by mohl použít pasivní útok. Pro zmatení detektoru



Obrázek 2.12: Upravená levá čára parkovacího místa což způsobuje to, že vozidlo nedetekuje parkovací místo.

jízdnicích pruhů stačí pouze několik značek vhodně vložených na silnici. Tuto situaci popisuje obr. 2.13.



Obrázek 2.13: Znázornění toho, co se stane, pokud na vozovku přidáme značky, které vozidlo vyhodnotí jako čáru.

Výsledkem přidání těchto značek je přesměrování autopilota vozidla do jiného pruhu, což může být právě záměrem útočníka.

Snaha o získání vzdálené kontroly nad vozidlem

U automobilů Tesla je systém pro řízení ovládan pomocí Electric Power Assisted Steering (EPAS), který je postaven na základě počítačového vidění a systémech se sběrnici používaných v automobilovém průmyslu (Ethernet, CAN, LIN). Systém pro řízení má integrovány dvě rozhraní CAN sběrnice a je spojen s koprocesorem LB, který se stará o monitorování zpráv na CAN a rozhoduje, která část systému řízení má být spuštěna. V samotném systému se nachází služba cantx, který přijímá signály ze systému počítačového vidění a transformuje je na signály k ovládní vozidla. Takto je možné, aby útočník dynamicky vkládal nebezpečný kód, kterým upravuje úhel natočení kol automobilu. Pro demonstraci útoku byl připojen ovladač k mobilnímu zařízení pomocí Bluetooth. Tento mobil transformoval řídicí signály z ovladače na signály pro službu cantx, kdy po kompromitování systému pro

řízení byly tyto signály periodicky zasílány, a na mobilní zařízení byl zasílán úhel natočení kol v reálném čase.

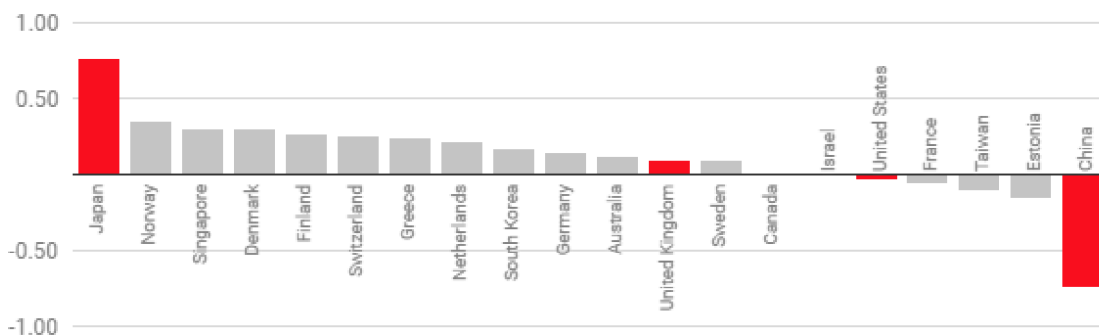
Více informací je možné získat z dokumentu od autorů¹⁵.

2.7 Možný budoucí vývoj

V blízké budoucnosti se dá očekávat větší rozmach autonomních vozidel. Tyto vozidla budou propracovanější, s lepšími technologiemi snad by dokonce mohly předpokládat chování chodců a ostatních řidičů.

Rozvoj autonomní dopravy ovšem přinese také úskalí, protože mnoho profesí, jako jsou taxikáři, řidiči dodávek, pravděpodobně přijde o práci, což může vzbudit nepokoje mezi těmito profesemi.

Je ovšem nutné odpovědět si také na morální otázky, jako je když se vozidlo dostalo do situace, kdy si má vybrat, zda nabourá do jiného vozidla nebo přejede několik lidí. Na takové otázky se snažili odpovědět výzkumníci z MIT pomocí Moral Machine. Po vyhodnocení téměř 40 milionů odpovědí [1], bylo na výsledcích zajímavé, že situace se může lišit od regionu, protože při výběru mezi mladým a starým člověkem by lidé z Evropy a Ameriky volili mladšího, v Asii by to bylo přesně naopak. Další zajímavá situace nastává, když si můžeme vybrat, zda mezi tím, že zabijeme člověka nebo se sami zabijeme tím, že nabouráme do překážky. V tomto průzkumu se ukázalo, že největší procento lidí, kteří by se byli ochotni obětovat je v Japonsku a naopak nejméně v Číně, což znázorňuje graf na obr. 2.14.



Obrázek 2.14: Graf znázorňující ochotu vlastního sebeobětování nebo zabití jiného člověka¹⁶.

Na této studii je možné poukázat, jak těžké bude standardizovat chování autonomního vozidla pro celý svět, proto je tedy možné, že když nastoupíme do autonomního vozidla v Japonsku, tak se budeme muset připravit na to, že nás automobil může zabít, protože při kritické situaci zvolí smrt posádky, čímž zachrání jiného člověka. Na druhou stranu je možné, že když pojedeme v jiné části světa, tak autonomní vozidlo zvolí možnost zachránit svou posádku. Je otevřenou otázkou, kdo bude zodpovědný za úmrtí člověka, kterého zabije autonomní vozidlo, protože současná legislativa s tímto jevem nepočítá.

S dalšími zajímavými výsledky přišla studie [24], kde byla vybrána skupina 100 lidí, kteří byli posazeni do simulátoru vozidla. Tento simulátor umožňoval nastavit tři typy řízení vo-

¹⁵[keenlab.tencent.com/en/whitepapers/Experimental_Security_Research_of_Tesla_Autopilot.pdf](https://www.keenlab.tencent.com/en/whitepapers/Experimental_Security_Research_of_Tesla_Autopilot.pdf)

¹⁶Převzato z <https://www.technologyreview.com/s/612341/a-global-ethics-study-aims-to-help-ai-solve-the-self-driving-trolley-problem/>

zidla, konkrétně normální, kdy se simulátor choval jako běžné vozidlo, agentní, kdy bylo simulováno vozidlo schopné ovládat své řízení a rychlost, chovalo se tedy jako autonomní vozidlo, a nakonec mohl simulátor simulovat antropomorfní vozidlo, což je typ vozidla stejný jako agentní, ale má své vlastní jméno, pohlaví a v předem stanovených místech pouštělo záznam lidského hlasu, který komentoval danou událost. Pro tento experiment byla nachystána situace, kdy byla protijedoucím vozidlem způsobena nehoda, které se nedalo vyhnout, přičemž byla jasně způsobena právě protijedoucím vozidlem. Experimenty bylo zjištěno, že skupina lidí, kteří byli ve vozidle řízeném agentně, více obviňovali své vozidlo za danou nehodu než lidé, kteří řídili sami. Zajímavější je ovšem to, že lidé, kteří byli v antropomorfním vozidle, obviňovali za způsobení nehody své vozidlo prokazatelně méně než skupina lidí, kteří byli uvnitř pouze agentního vozidla. Tento výsledek prokazuje souvislost mezi antropomorfismem a vnímáním odpovědnosti, ale přesná povaha této souvislosti nemůže být testována, protože není možné vytvořit stejné nehody zaviněné výhradně řidiči samotnými.

Osobně si myslím, že největší problém bude, až nastane doba, kdy budou existovat plně autonomní vozidla, protože nelze předpokládat, že si je všichni budou moci pořídit. Toto může přinést velké problémy, jelikož řidič řídící klasické auto si může začít více dovolovat jako například projíždět semafor na červenou, s tím že autonomní vozidlo si s danou situací poradí.

Kapitola 3

Prostředky pro modelování systémů

Tato kapitola se zabývá modelováním. Nejprve popisuje co to vlastně modelování je, čeho se jím snažíme dosáhnout, k čemu je dobrá simulace. Poté nabízí přehled nástrojů, které je možné použít pro modelování a simulaci samočinně řízených vozidel.

3.1 Teorie

Cílem simulace je získat nové znalosti o zkoumaném systému. Abychom mohli provádět simulaci, je nutné vytvořit vhodný model tohoto systému [19].

Systém je soubor elementárních částí, které mají mezi sebou určité vazby. Model je napodobenina systému jiným systémem.

Modelování je proces vytváření modelů systému na základě našich znalostí. Pro modelování je nutné znát důkladně původní systém a dokázat určit, které jeho prvky jsou důležité.

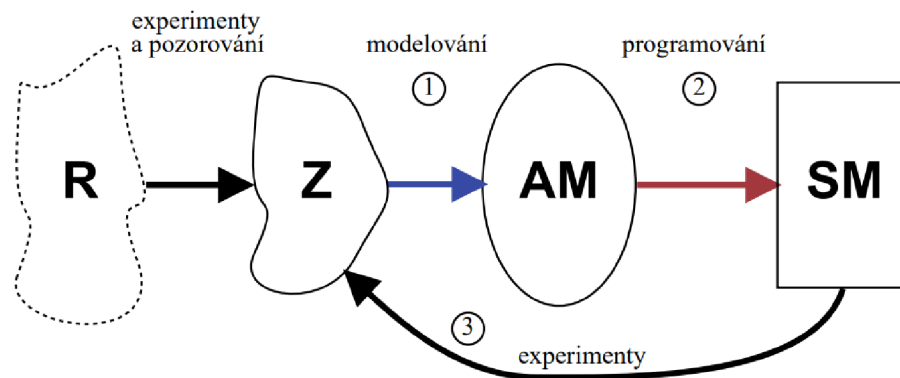
Simulace je experimentování se simulačním modelem. Jejím cílem je získání nových znalostí o systému v závislosti na vstupních veličinách.

Abstraktní model je model, který nezahrnuje všechny naše znalosti o modelovaném systému, ale vybíráme jen vlastnosti, které jsou pro naše účely podstatné. Díky tomu dosáhneme zjednodušení modelu na zvládnutelnou úroveň.

Simulační model musí zahrnovat všechny vlastnosti abstraktního modelu. Je to vlastně abstraktní model zapsaný v programovacím jazyce.

Validita modelu je snaha dokázat, že pracujeme s modelem adekvátním modelovanému systému. Tato vlastnost nelze dokázat úplně přesně, proto se za validitu modelu považuje míra správnosti získaných výsledků.

Typická posloupnost při modelování spočívá v získání znalostí o reálném systému. Z těchto znalostí se vytvoří abstraktní model, který se pomocí programovacího jazyka převede na simulační model. Experimentováním s tímto modelem získáváme další znalosti, které slouží k upřesňování abstraktního modelu. obr. 3.1. zachycuje celý proces získávání znalostí s využitím simulace.



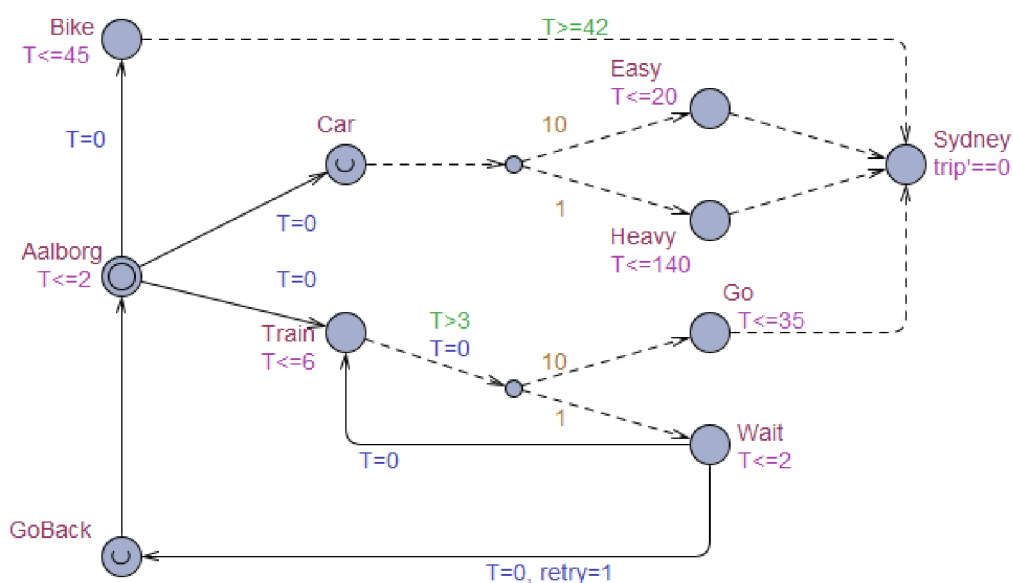
Obrázek 3.1: Schéma celého procesu získávání znalostí s využitím simulace [19].

3.2 Nástroje

Následující podkapitoly popisují jednotlivé nástroje, které je možné využít pro modelování systémů. K modelování systémů je samozřejmě možné použít řadu jiných nástrojů, jenž staví na jiných základech například na formalismu DEVS, Petriho sítích nebo jiných.

3.2.1 UPPAAL

Rodina nástrojů UPPAAL slouží pro modelování, simulaci a verifikaci systémů reálného času. Tyto systémy jsou reprezentovány pomocí jednoho či více časovaných automatů, rozšířených o proměnné typu `integer`, strukturované datové typy a synchronizaci pomocí signálů [5]. Skládají se z grafického uživatelského rozhraní a prostředku pro model-checking. Obsahují tři hlavní části: jazyk pro popis modelu, simulátor a verifikátor. V tomto nástroji se systém modeluje pomocí stochastických časovaných automatů, což jsou jednoduše řečeno konečné automaty s časem. Ukázka automatu je na následujícím obr. 3.2.

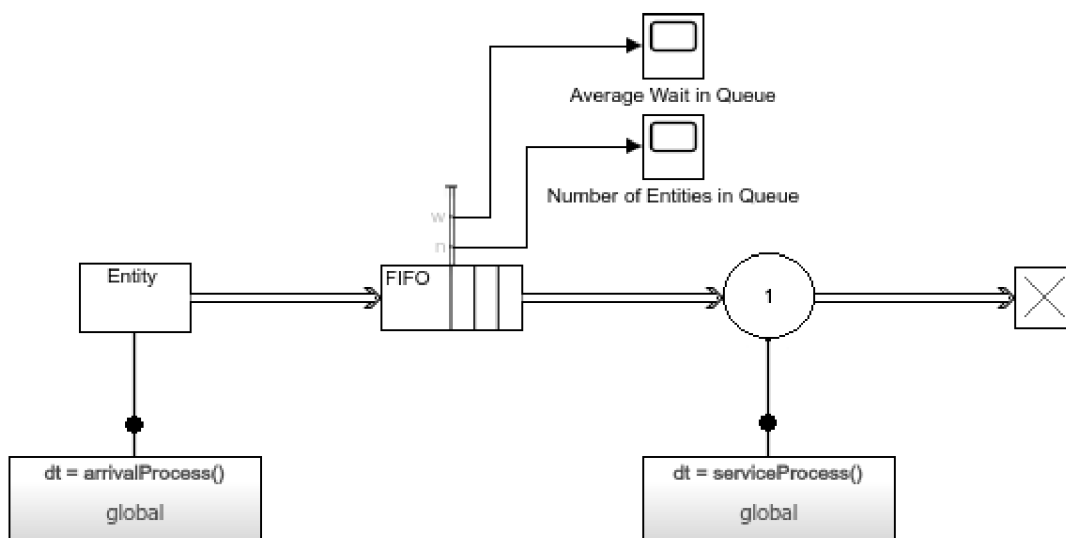


Obrázek 3.2: Ukázka modelu jízdy z města Aalborg do Sydney pomocí různých dopravních prostředků vytvořeného v prostředí UPPAAL Stratego.

3.2.2 MATLAB / Simulink

MATLAB je interaktivní vývojové prostředí a programovací jazyk. Mezi jeho výhody patří maticové operace, umožnění vizualizace funkcí a dat, programování algoritmů ať už v jazyce MATLAB nebo jiných programovacích jazycích např. C++, C#, Java, Python. Je snadno rozšiřitelný o různé funkce pomocí toolboxů.

Simulink je nástroj pro vytváření modelů a jejich následnou simulaci v programovém prostředí MATLABu [16]. Vytváření modelu probíhá vytvářením jednotlivých bloků modelovaného systému. Podporovány jsou lineární i nelineární systémy, spojitě i diskrétní modely, případně jejich kombinace. Vytváření modelu probíhá skládáním blokových diagramů v grafickém uživatelském rozhraní. Obsahem Simulinku je také obsáhlá knihovna s již hotovými bloky, ať již jednoduchých jako je blok reprezentující funkci sinus nebo i složitých např. převodovka automobilu. Tyto bloky lze samozřejmě použít pro vytvoření složitějšího modelu. Ukázka jednoduchého modelu je zobrazena na obr. 3.3.

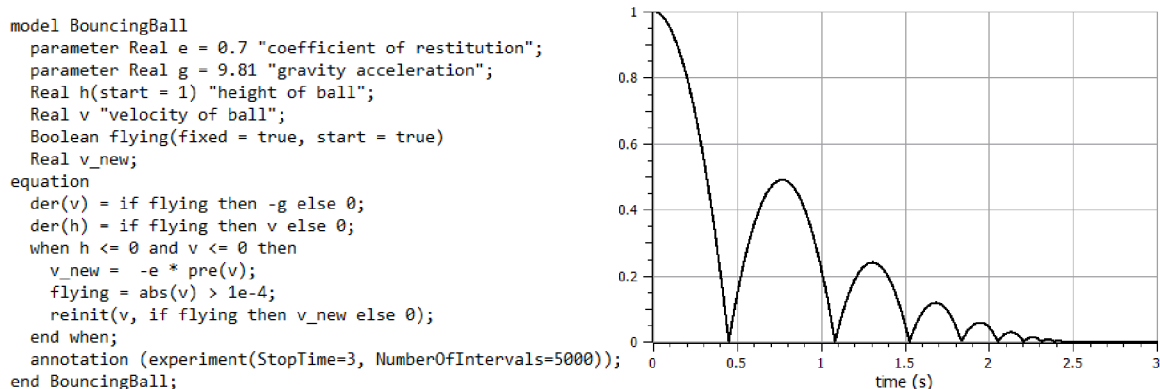


Obrázek 3.3: Ukázka jednoduchého modelu fronty v prostředí Simulink.

3.2.3 Dymola / Modelica

Simulační systém Dymola umožňuje modelovat a simulovat fyzikální systémy. Charakteristickou vlastností Dymoly je to, že se jedná o objektově orientovaný jazyk. Umožňuje skládání modelů. Dymola je založena na použití modelů z jazyka Modelica. Existují knihovny znovupoužitelných komponent modelů pro různé aplikační oblasti. Dymola obsahuje překladač pro rovnice Modeliky pro generování zdrojového kódu v jazyce C. Model je možné vytvářet dvěma způsoby buď je psát přímo ve zdrojovém kódu, nebo použít grafický editor modelů, kde se vytvoří blokové schéma, jež se bude skládat z komponent, které se poté propojí pomocí konektorů. Uvnitř bloků se nachází rovnicové popisy. Rovnice jsou automaticky upraveny tak, aby se systém zjednodušil. Dále kontroluje, zda se v rovnicích nenachází nějaká chyba. Mezi další vlastnosti patří možnost propojování s jinými systémy např. s MATLABem. Umožňuje simulaci reálného času, což může být užitečné, když chceme vytvořeným modelem řídit nějakou jinou součást. Výsledky je možné vizualizovat pomocí grafů, animací (2D nebo 3D) nebo je možné získat výsledky ve formátu pro MATLAB.

Modelica je objektivě orientovaný jazyk pro modelování komplexních fyzikálních systémů [18]. Modely jsou popsány schémata v blokovém diagramu. Schémata obsahují komponenty jako je rezistor nebo hydraulický válec. Tyto komponenty mají konektory, které popisují možnosti interakce. Pomocí spojování těchto konektorů je vytvářen model. Interně je model definován pomocí rovnicového popisu v syntaxi jazyka Modelica. Tento jazyk slouží pro textový popis všech součástí modelu a strukturuje komponenty modelu do knihoven. Jazyk Modelica je určen pro diferenciální, algebraické a diskrétní rovnice. Neumožňuje přímo definovat parciální diferenciální rovnice, ale je možné definovat některé diskrétní parciální diferenciální rovnice. Jednoduchý model, napsaný v jazyce Modelica a simulovaný pomocí programu OpenModelica, je zobrazen na obr. 3.4.

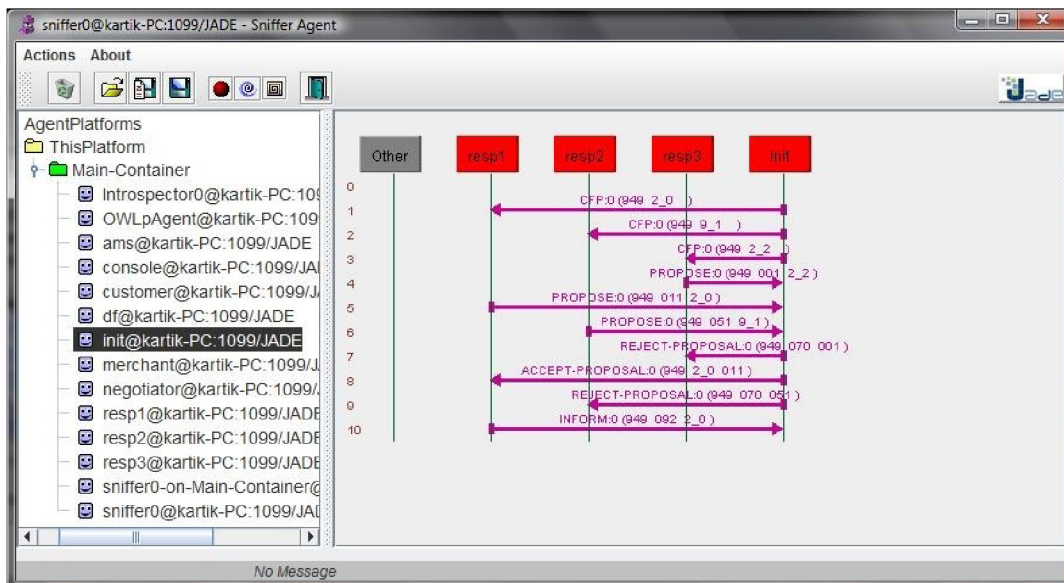


Obrázek 3.4: Na levé straně je zobrazen kód jednoduchého modelu padajícího míče. Graf napravo ukazuje průběh vzdálenosti míče od země v čase.

3.2.4 Jade

Jade (Java Agent DEvelopment Framework) je softwarový framework. Je určený pro vývoj aplikací založených na agentech, kteří jsou v souladu s FIPA specifikacemi pro inteligentní multi-agentní systémy [9]. Agent je chápán jako někdo, kdo jedná v nějakém prostředí v zájmu klienta. Umělý agent je tedy dílo, vytvořené nějakým člověkem, které jedná jako autonomní samostatná jednotka, která je umístěna v nějakém prostředí, kde jedná ve prospěch svého klienta. Agentní systém se skládá ze samotného agenta a prostředí, ve kterém operuje. Ukázka jednoduchého agenta se nachází na obr. 3.5. FIPA je nezisková organizace zabývající se standardizací v oblasti agentních technologií. Samotný Jade se skládá ze tří hlavních částí, konkrétně Runtime environment, což je prostředí pro běh agentů, dále z knihovny tříd a GUI nástroje pro správu a ladění. Každý agent obsahuje agentní identifikátor (AIS), který tvoří jeho jméno, adresa a vyhledávací body. Základem pro vývoj v tomto prostředí je třída Agent, která zapouzdřuje životní cyklus agenta, jeho prostředky pro zasílání a přijímání zpráv [14]. Funkcionalita agenta je tvořena implementací metod, které vyjadřují danou činnost, přičemž existují tři základní druhy chování:

- Jednoduché – provádí nějakou konkrétní akci, mohou být provedeny jednou či vícekrát
- Složené – slouží ke spuštění dalších chování ve stanoveném pořadí
- Specifické pro role v komunikačních protokolech - metody odpovídají jednotlivým stavům protokolu, jako parametr obdrží příslušné zprávy



Obrázek 3.5: Ukázka prostředí a sniffer agenta v Jade¹⁷.

3.2.5 SIMLIB

SIMLIB je simulační knihovna vyvíjena na FIT VUT Brno, jejímž hlavním autorem je dr. Ing. Petr Peringer. Poskytuje základní prostředky pro popis spojitých, diskretních i kombinovaných modelů a prostředky pro řízení simulace. Popis modelů probíhá přímo v jazyce C++, a není proto nutný překladač simulačního jazyka [17]. Díky této koncepci je možné používat ostatní prostředky vytvořené v jazyce C++ jako např. grafické knihovny a uživatelská rozhraní. Za nevýhodu tohoto přístupu lze považovat nemožnost dodatečných syntaktických a sémantických kontrol, které by se mohly provádět při použití překladače simulačního jazyka. Model je chápán jako množina prvků, které jsou spolu navzájem propojeny vazbami. Tyto vazby spolu s chováním prvků definují chování systému jako celku. Objekty modelu provádějí určité akce, jako odezvu na přijímané zprávy, a současně provádějí jiné akce, které jsou nezávislé na přijímaných zprávách. Akce mění stav objektu, jenž je dán obsahem jeho vnitřních datových struktur. Ukázka kódu jednoduchého modelu, který modeluje generátor zákazníku, je na obr. 3.6.

```

Facility Box("Box");
Histogram Table("Table",0,25,20);

class Customer : public Process {
    double Prichod; //attribute of each customer
    void Behavior() { //behavoir specification
        Prichod = Time; //incoming time
        Seize(Box); //start of service
        Wait(10); //time of service
        Release(Box); //end of service
        Table(Time-Prichod); //waiting and service time
    }
};

class Generator : public Event { // model of system's input
    void Behavior() { // behavior specification
        (new Customer)->Activate(); // new customer
        Activate(Time+Exponential(1e3/150)); //
    }
};

int main() { // experiment description
    SetOutput("model2.out");
    Init(0,1000); // initialization time 0..1000
    (new Generator)->Activate(); // customer generator
    Run(); // simulation
    Box.Output(); // print of results
    Table.Output();
    return 0;
}

```

Obrázek 3.6: Ukázka modelu zákazníka a jejich generátoru vytvořeného pomocí knihovny SIMLIB.

¹⁷Převzato z <https://confluence.oceanobservatories.org/display/CIDev/JADE+Framework>

Kapitola 4

Zvolené realizační prostředky

Tato kapitola popisuje rodinu nástrojů UPPAAL, kde nástroj UPPAAL Stratego byl vybrán pro realizaci praktické části této diplomové práce.

4.1 UPPAAL

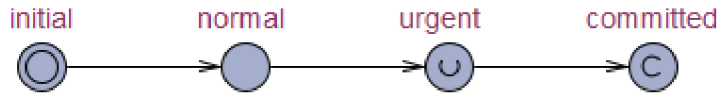
UPPAAL je nástroj určený pro modelování, simulaci a verifikaci systémů reálného času. Jako formalismus používá časované automaty, což je šestice $A = (L, l_0, X, \Sigma, E, Inv)$ [3], kde:

- L je konečná množina stavů
- $l_0 \in L$ je počáteční stav
- X je konečná množina hodin
- Σ je konečná množina akcí
- $E \in L \times \mathcal{B}(X) \times \Sigma \times 2^X \times L$ je konečná množina hran
- $Inv : L \rightarrow \mathcal{B}(C)$ přiřazuje invariant každému stavu

Modelovací jazyk nabízí další funkce, jako jsou proměnné typu `integer` s ohraničeným rozsahem. Dotazovací jazyk, který je určen ke specifikaci vlastností, jež mají být kontrolovány, je podmnožina CTL logiky.

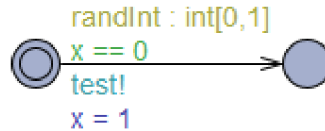
Model se skládá z procesů, které jsou složeny ze stavů. Čas je měřený pomocí hodin časovaných automatů, kde platí, že tento čas je spojitý. Chování modelu určují stavy a hrany reprezentující spojení mezi nimi. Existují čtyři typy stavů. **Initial** je typ stavu, který určuje počáteční stav automatu. V každém automatu musí být právě jeden. **Urgent** je typ stavu, který zastavuje čas, tedy čas nemůže běžet, pokud se automat nachází ve stavu s vlastností urgent. Před provedením přechodu ze stavu s vlastností urgent mohou probíhat jiné přechody, které ale nesmí posouvat čas. **Committed** je typ stavu, který zastavuje čas, tedy čas nemůže běžet. Pokud se automat nachází ve stavu s vlastností committed, tak je nutné, aby další provedení přechodu byl právě ze stavu s touto vlastností. **Normal** je typ stavu, který nepatří ani do jedné z předešlých kategorií. **Initial** a **normal** stavy mají položku `invariant`, která specifikuje podmínku, jež musí platit, když se automat nachází v daném stavu. Grafické ukázky jednotlivých stavů jsou na obr. 4.1.

Hrany obsahují položky `select`, `guard`, `synchronization` a `update`. `Select` nedeterministicky přiřazuje dané proměnné hodnotu z určeného intervalu. `Guard` je podmínka, která



Obrázek 4.1: Různé druhy stavů v systému UPPAAL. Nejprve je zde stav `initial`, který přechází do stavu `normal`, ten umožňuje přechod do stavu s vlastností `urgent`, ze kterého vede přechod do stavu s vlastností `committed`.

vyhodnotí výraz v ní uložený, a pokud je výraz vyhodnocen jako `true`, pak je možné vykonat tento přechod. `Synchronization` je mechanismus, který umožňuje pomocí zasílání signálu typu `chan` provést přechod a předat řízení jinému automatu. Pokud je `t` synchronizační kanál, pak provedením přechodu z jednoho stavu do jiného stavu přes hranu s `t!` způsobí, že se automat dostane do jiného stavu a řízení se předá jinému automatu, který obsahuje hranu s `t?` v položce `synchronization`. Existují tři druhy synchronizačních kanálů, kde obyčejný `chan` umožňuje plynutí času během předání řízení, tak `urgent chan` plynutí času nepovoluje. Poslední typ `broadcast chan` umožňuje předat řízení více automatům. Položka `update` umožňuje provedením přechodu volat funkce a upravovat hodnoty proměnných. Ukázka hrany obsahující všechny položky je zobrazena na obr. 4.2.



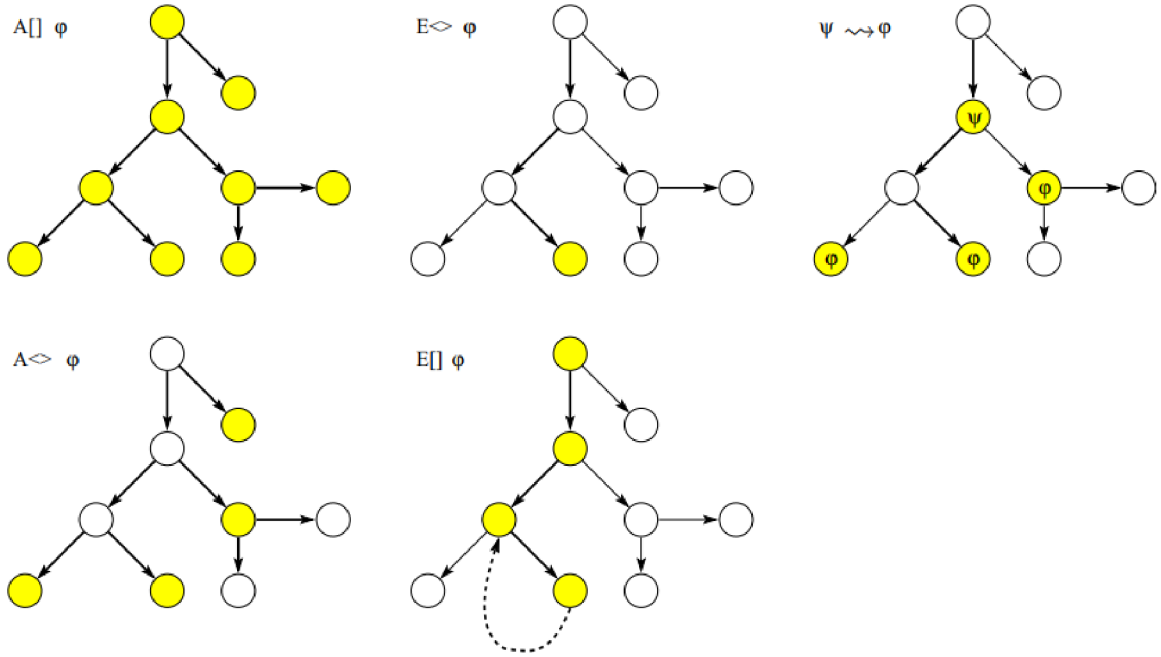
Obrázek 4.2: Hrana s položkami `select` (žlutá barva), `guard` (zelená barva), `synchronization` (tyrkysová barva) a `update` (modrá barva).

Simulátor umožňuje ruční nebo automatické procházení dosažitelných stavů vytvořeného modelu, zároveň umožňuje také sledovat obsah jednotlivých proměnných typu `integer`, `bool` a obsah prvků pole. Dovoluje také zobrazení intervalů, kterých mohou nabývat proměnné reprezentující čas. Bohužel není možné sledování hodnoty proměnných datového typu `double`. Je také umožněno uložit a nahrát posloupnost provedených přechodů daného modelu.

Nástroj dále obsahuje položku verifikace, ve které je možné pomocí podmnožiny formulí CTL logiky verifikovat vytvořený systém. obr. 4.3 ukazuje podporované formule. Pokud je formule nesplněna, pak je možné si do simulátoru nechat nahrát posloupnost přechodů, která vede k nesplnění formule.

Přehled podporovaných formulí:

- $A \langle \rangle p$ - Eventually - vyhodnocena `true` tehdy a jen tehdy, pokud všechny možné posloupnosti přechodů skončí ve stavu, který splňuje p
- $E \langle \rangle p$ - Possibly - vyhodnocena `true` tehdy a jen tehdy, pokud existuje posloupnost příkazů $s_0 \rightarrow s_n$, kde s_0 je počáteční stav a s_n je stav, který splňuje p
- $A [] p$ - Invariantly - vyhodnocena `true` tehdy a jen tehdy, když každý dosažitelný stav splňuje p
- $E [] p$ - Potentially always - vyhodnocena `true` tehdy a jen tehdy, pokud existuje posloupnost přechodů $s_0 \rightarrow s_1 \rightarrow \dots \rightarrow s_i \rightarrow \dots$ kde p je vždy splněna
- $p - - > q$ - Leads To - pokud někdy bude splněna formule p , pak q bude také splněna



Obrázek 4.3: Formule podporované v UPPAAL. Žluté stavy znázorňují, ve kterých stavech je formule φ splněna [2].

4.1.1 UPPAAL SMC

Nástroj UPPAAL SMC rozšiřuje klasický UPPAAL o symbolický model checking (SMC). V časovaných automatech je umožněno, aby hodiny měly různé tempo. Hlavní myšlenka je v monitorování simulací systému, a následného zpracování výsledků pomocí statistiky k tomu, aby bylo rozhodnuto, zda model splňuje vlastnost s nějakou mírou důvěry. Z podstaty věci se jedná o kompromis mezi testováním a klasickým model checkingem. Je známo, že metody založené na simulaci mají výrazně menší požadavky na paměť a čas [5]. Dále přidává podporu pro hybridní hodiny (hybrid clock), které neovlivňují logiku modelu, protože nemohou být použity v **guard** a **invariant** položkách hran, respektive stavů. Mohou být ovšem použity pro obyčejné diferenciální rovnice (ODE). Přidává také možnost dynamického vytváření časovaných automatů za běhu. Přehled možností pro ověřování vlastností modelu přidanych v UPPAAL SMC:

- $Simulate\ n\ [\leq T]\ \{prom_1, \dots, prom_n\}$ - provedení n simulací modelu s časovým intervalem $\langle 0; T \rangle$, kdy jsou zjištěny hodnoty proměnných $prom_1$ až $prom_n$
- $Pr[\leq T](\langle \rangle (vyraz))$ - hledání s jakou pravděpodobností nastane *vyraz* v časovém intervalu $\langle 0, T \rangle$
- $Inf\{vyraz\}$ - hledání nejmenší hodnoty jaké může *vyraz* nabývat
- $Sup\{vyraz\}$ - hledání největší hodnoty jaké může *vyraz* nabývat

Záložka **Options** položka **Statistical parameters** umožňuje upravit výchozí hodnoty použité při počítání pravděpodobnosti, čímž je možné ovlivnit dobu běhu ověření vlastnosti modelu. Asi nejzajímavější položkou v této nabídce je **probability uncertainty** (ϵ), která udává, s jakou nejistotou pravděpodobnosti budou pracovat dotazy na zjištění pravděpodobnosti.

4.1.2 UPPAAL Stratego

UPPAAL Stratego kombinuje různé techniky pro analýzu z jiných nástrojů z rodiny UPPAAL a zaměřuje se na strategie [11]. Strategie je obecně předpis akcí jednoho hráče pro všechny stavy, které mohou vést k dosažení cíle tohoto hráče. V tomto nástroji jsou nápomocné k tomu, aby model vybíral takové přechody, které vedou ke splnění výrazu v definici strategie [4]. Je založen na Timed Game Automaton (TGA), což je sedmice $G = (L, l_0, X, \Sigma_c, \Sigma_u, E, Inv)$ [3], kde:

- Σ_c je konečná množina kontrolovatelných přechodů
- Σ_u je konečná množina nekontrolovatelných přechodů
- $(L, l_0, X, \Sigma_c \cup \Sigma_u, E, Inv)$ je časovaný automat

Je tedy vidět, že TGA se od normálního časovaného automatu liší tím, že rozděluje množinu přechodů na dvě samostatné množiny. Tyto množiny se nazývají množina kontrolovatelných a množina nekontrolovatelných přechodů, kde $\Sigma_c \cap \Sigma_u = \emptyset$. Kontrolovatelné přechody jsou ovládány kontrolérem a nekontrolovatelné ovládá prostředí. Kontrolér se snaží zabránit tomu, aby prostředí způsobovalo nebezpečné situace, a aby zároveň pracoval žádoucím způsobem.



Obrázek 4.4: Grafický rozdíl mezi kontrolovatelným přechodem na levé straně a nekontrolovatelným přechodem na pravé straně.

Strategie je vytvářena ve verifikátoru klíčovým slovem **strategy S**, a používá se příkazem **under S**, kde **S** je proměnná reprezentující strategii. Je možné ji použít na dotazy z jiných nástrojů rodiny UPPAAL. obr. 4.5. uvádí některé možné příkazy, na které je možné použít strategie.

Safety	A[] prop under NS
Liveness	A<> prop under NS
Guarantee objective	strategy NS = control: A<> prop
Guarantee objective	strategy NS = control: A[] prop
Evaluation	Pr[bound] (<> prop) under SS
Expected	value E[bound;int](min: prop) under SS
Simulations	simulate int [bound]{expr1,expr2} under SS
Minimize objective	strategy DS = minE (expr) [bound]: <> prop under NS
Maximize objective	strategy DS = maxE (expr) [bound]: <> prop under NS

Obrázek 4.5: Dotazy, na které je možné použít strategie [4].

Na tomto místě je nutné zmínit, že při použití funkce pro generování náhodných čísel a funkce **rint**, která slouží pro převod čísel datového typu **double** na číslo datového typu **integer**, není možné použít symbolické dotazy $A[]$ a $E<>$ ¹⁸. Dále bylo zjištěno, že pokud jsou v modelu volány tyto funkce, tak není možné vytvářet strategie.

¹⁸<https://stackoverflow.com/questions/55951475/uppaal-smc-avoid-state-space-explosion>

4.2 Případová studie

Následující kapitola popisuje model, který je reprezentovaný automatem na obr. 3.2. Tento model se po instalaci nástroje UPPAAL Stratego nachází v ukázkových souborech ve složce *demo/stratego/traffic.xml*. Časová jednotka tohoto příkladu je jedna minuta. Model zkoumá, jaký typ dopravního prostředku je nejlepší pro osobu se jménem Kim využít při cestě z města Aalborg do města Sydney. K dispozici je přitom kolo, automobil a vlak. Nejprve musíme reprezentovat výběr dopravního prostředku, kde tato činnost trvá maximálně dvě minuty. Jízda na kole je sice nejpomalejší, ale je jisté, že se do cílové destinace vždy dostaneme mezi 42 až 45 minutami. Dále se nabízí jízda automobilem, který je oproti jízdě na kole rychlejší, pouze pokud nebude hustá doprava. Šance na hustou dopravu je 9%, přičemž cesta pak bude trvat až 140 minut, zatímco pokud bude doprava mírná, tak bude cesta trvat 20 minut. Poslední možností je využít služeb vlakové dopravy, která jezdí v pravidelných šestiminutových intervalech. Zde hrozí s pravděpodobností 9% nebezpečí, že vlak bude zrušen, a budeme nuceni si vybrat, zda vyčkat na další vlak nebo se vydat domů, a znovu si promyslet volbu dopravního prostředku. Pokud vlak není zrušen, tak cesta do Sydney trvá 35 minut. Na základě těchto informací můžeme zkoumat například to, zda se osoba za všech případů dostane do Sydney během jedné hodiny. Tuto vlastnost ověřuje dotaz 4.1. Odpověď na tuto otázku je samozřejmě záporná, protože výběr jakéhokoliv prostředku mimo jízdě na kole toto nezaručuje.

$$A \langle \rangle Kim.Sydney \ \&\& \ time \leq 60 \quad (4.1)$$

Protože ne vždy se dostaneme do Sydney během 60 minut, tak chceme zjistit jaká je pravděpodobnost, že se tam dostaneme během 60 minut. Tuto hodnotu je možné zjistit pomocí dotazu 4.2, čímž zjistíme, že se tak stane s pravděpodobností zhruba 95%.

$$Pr[\leq 60](\langle \rangle Kim.Sydney) \quad (4.2)$$

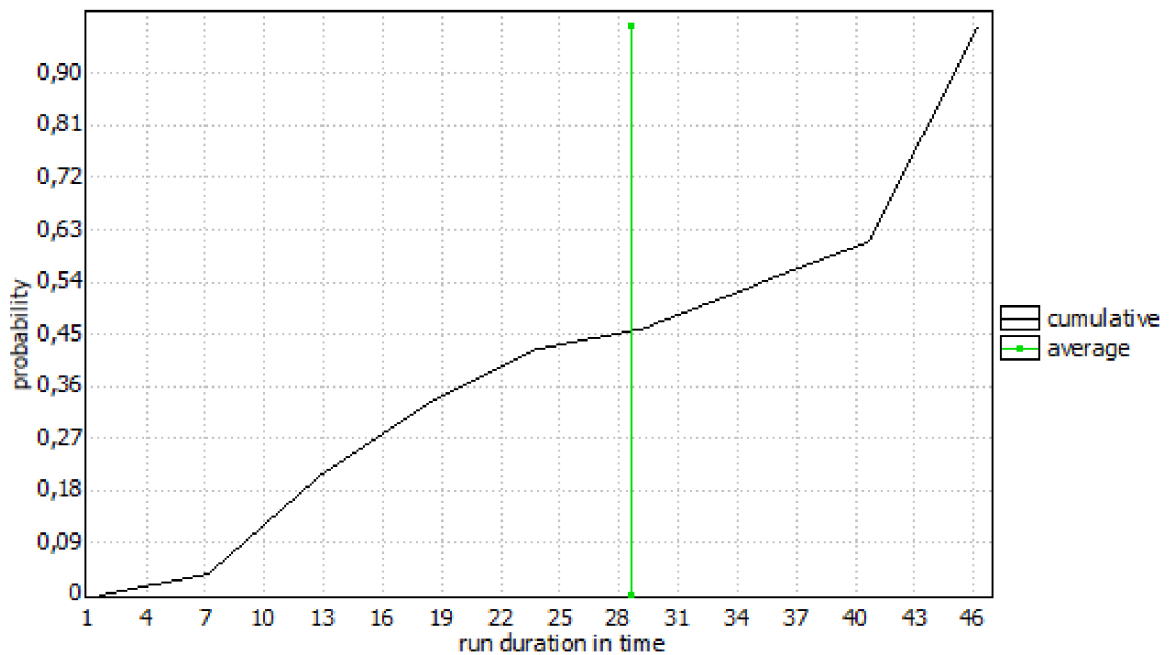
Tuto skutečnost zobrazuje graf na obr. 4.6.

Protože chceme zajistit, aby se Kim vždy do jedné hodiny dostala do Sydney, tak přichází na řadu strategie 4.3, která způsobí, že tato vlastnost bude vždy splněna.

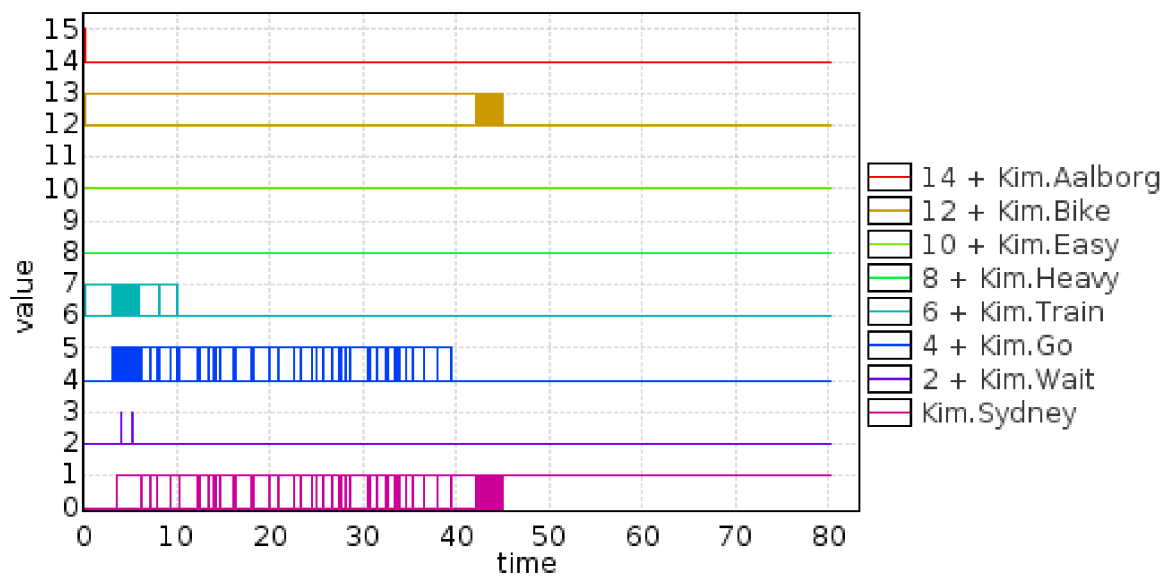
$$strategy \ GoSafe = control : A \langle \rangle Kim.Sydney \ \&\& \ time \leq 60 \quad (4.3)$$

Po úspěšném vypočítání strategie *GoSafe* můžeme ověřit, zda je možné se vždy dostat do Sydney během jedné hodiny, což uděláme pomocí dotazu 4.4. Protože je tato formule splněna, tak můžeme prohlásit, že se Kim při využití této strategie vždy dostane do města Sydney během jedné hodiny. Následující obr. 4.7 zobrazuje využití dopravních prostředků při cestě do Sydney.

$$A \langle \rangle (Kim.Sydney \ \&\& \ time \leq 60) \ under \ GoSafe \quad (4.4)$$



Obrázek 4.6: Graf znázorňující jaká je pravděpodobnost toho, že Kim bude v nějakém čase ve městě Sydney.



Obrázek 4.7: Graf znázorňující délku strávenou v jednotlivých stavech při spuštění simulace stokrát.

Kapitola 5

Návrh a popis modelu

Tato kapitola se zabývá vytvořením modelu samočinně řízeného vozidla. Pro tento účel byl vybrán nástroj UPPAAL Stratego, který je popsán v kapitole 4. Protože nástroj pracuje s časovanými automaty, budou v této kapitole popsány právě jednotlivé automaty, ze kterých se skládá výsledný model.

5.1 Návrh modelu

Při navrhování modelu samočinně řízeného vozidla bylo možné tento model vytvořit dvěma způsoby. První možnost je zabývat se samotným vozidlem, tedy zaměřit se na chování při jízdě, předjíždění jiného vozidla nebo parkování. Druhou možností je zaměřit se ne na samotné vozidlo, ale na více těchto vozidel. U této možnosti by se model zabýval především vzájemnou komunikací vozidel, kde by bylo nutné řešit, které vozidlo má na starosti jakou činnost, jak se tyto činnosti mění na základě počtu ovládaných vozidel. Případně určit, jaké rozmezí počtu vozidel je optimální s ohledem na celkovou velikost kolony vozidel a množství získaných dat od vozidel. Tato práce se zabývá modelováním první možnosti, konkrétně se zabývá chováním při běžné jízdě a při předjíždění.

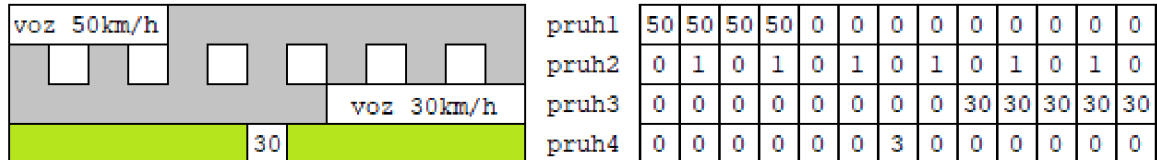
Reálné vozidlo a fyzikální děje mající vliv na pohyb automobilu jsou velmi rozsáhlé a některé i těžce popsatelné v celé své podstatě. Budeme proto předpokládat, že se vozidlo pohybuje po rovné cestě bez nerovností, a zároveň na něj nemá vliv odpor vzduchu. Soustředíme se hlavně na jeho korektní chování v běžném silničním provozu, tedy na jízdu, která vyžaduje reakci na ostatní vozidla. Zvýšení a snížení rychlosti bude skokové o konstantní hodnotu, a zároveň si autonomní vozidlo dokáže udržet hodnotu své rychlosti, tak dlouho dokud není okolnostmi donuceno ji změnit. Mezi takové okolnosti patří například reakce na pomaleji jedoucí vozidlo, jež se nachází v detekovatelné vzdálenosti a v budoucnosti by hrozila srážka to vše za předpokladu, že ho není možné z nějakých okolností předjet. Chování modelu bude ovládat několik časovaných automatů, které budou měnit své stavy podle hodnot proměnných, jež získají své hodnoty pomocí funkcí, které modelují různé využití senzorů autonomních vozidel. Lidar umožňuje vytvoření funkcí pro detekci vozidel, určení jejich rychlosti a vzdálenosti od samočinně řízeného vozidla. Kamery se starají o rozpoznávání svislého a vodorovného dopravního značení.

Je také vytvořeno jednoduché chování vozidel, která nejsou autonomní. Tato vozidla mají možnost změnit svou rychlost každou sekundu, což slouží k věrnějšímu zachycení reality okolního prostředí. Neautonomní vozidla jsou namodelovány tak, aby neporušovaly dopravní předpisy, tedy nemají dovoleno překračovat maximální povolenou rychlost v daném místě

silnice, a pokud je autonomní vozidlo předjíždí, tak nesmí zrychlovat. Časovým automatem je vždy ovládáno vozidlo, jež je nejbližší autonomnímu vozidlu. Pokud se v modelu nachází ještě jiná vozidla, pak jsou řízena funkcí, generuje náhodné číslo, a podle něj upraví hodnotu rychlosti každého takového vozidla, samozřejmě za předpokladu, že je to možné. Rychlost vozidel je omezena v tom, že nesmí překročit maximální povolenou rychlost jak již bylo zmíněno, ale také nesmí klesnout pod hodnotu `maxRychlost - 20`.

5.2 Použité abstrakce

Jako základní jednotka času byla zvolena jedna sekunda, proto pokud bude u některé hodinové proměnné v modelu údaj 1, pak se bude jednat o jednu sekundu. Vozovka je reprezentována čtyřmi poli datového typu `integer`, kde `pruh1` reprezentuje protisměr, `pruh2` představuje vodorovné dopravní značení oddělující `pruh1` a `pruh3`, `pruh3` reprezentuje pruh silnice, ve kterém jedou vozidla řízená lidmi ve stejném pruhu jak autonomní vozidlo a `pruh4` představuje okolí silnice, ve kterém se nacházejí dopravní značky, které způsobují upravení maximální povolené rychlosti vozidel. Názorná ukázka je na obr. 5.1. Každé neautonomní vozidlo zabírá pět prvků pole, kde jsou reprezentovány hodnotou své rychlosti. Vodorovné dopravní značení je uloženo jako posloupnost nul a jedniček, kde nula značí část bez bílé barvy a jednička pravý opak. Za dopravní značku jsou v tomto modelu považovány pouze značky upravující maximální povolenou rychlost v úseku vozovky od místa umístění této značky do další značky případně konce silnice. Pro správnou funkčnost modelu je potřebné inicializovat proměnnou `maxRychlost` na kladnou hodnotu, která určuje maximální povolenou rychlost, dokud není rozpoznána dopravní značka v `pruh4`.



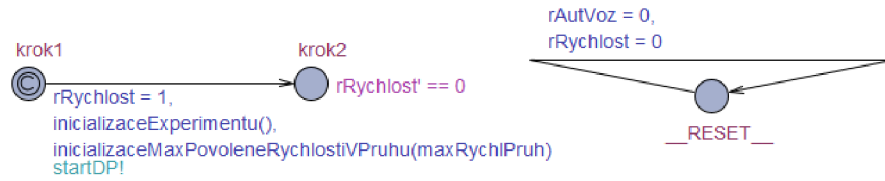
Obrázek 5.1: Obrázek ilustrující význam jednotlivých proměnných `pruh1` – `pruh4`.

5.3 Popis jednotlivých automatů

Tato kapitola obsahuje popis jednotlivých automatů, které jsou zařazeny do podkapitol, podle funkce zastávající v modelu.

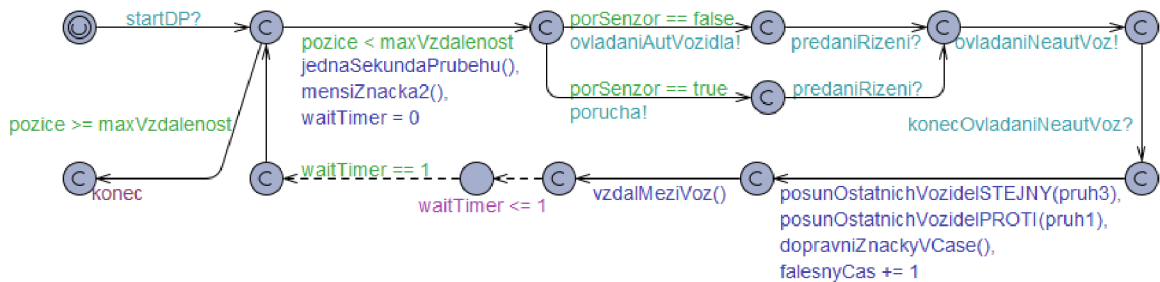
5.3.1 Řízení simulace

Spuštění modelu má za úkol automat zobrazený na obr. 5.2. Ten se stará o inicializaci hybridních hodin `rRychlost`, které jsou od verze 4.1.20.5 nutné pro korektní vykreslování grafů. Funkce `inicializaceExperimentu`, přiřadí do proměnných hodnoty, které byly použity v kapitole 6 k provedení experimentů. Funkce `inicializaceMaxPovoleneRychlostiVPruhu` má na starosti zjištění maximální povolené rychlosti na každém metru vozovky. Jejím parametrem je reference pole, do kterého uloží výsledek. Nakonec vyše signál `startDP`, čímž se řízení předá automatu pro řízení simulace. Stav `__RESET__` označuje proměnné, které neovlivňují strategie.



Obrázek 5.2: Automat, který spustí celý model.

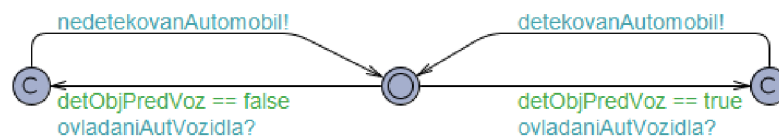
O řízení simulace se stará automat, jehož diagram je na obr. 5.3. Jedná se o mozek celého modelu. Nejprve čeká na signál `startDP`, a po jeho přijetí rozhoduje, zda vozidlo již ujelo vzdálenost větší než je hodnota proměnné `maxVzdalenost`. Pokud je odpověď kladná, pak přejde do stavu `konec` a činnost modelu je ukončena. Pokud je ovšem odpověď záporná, tak se podle hodnoty proměnné `porSenzor` rozhoduje zda nastala chyba senzoru a je na ni potřeba zareagovat nebo signálem `ovladaniAutVozidla` předá řízení automatu, který vybere reakci autonomního vozidla. Poté čeká na signál `predaniRizeni`, po jehož přijetí se posílá signál `ovladaniNeautVozidla`. Po přijetí signálu `konecOvladaniNeautVoz` jsou posunuty neautonomní vozidla o vzdálenost podle své rychlosti, samozřejmě za předpokladu, že se v modelu nějaké nacházejí, a je kontrolováno, zda vozidlo neprošlo kolem dopravní značky, která omezuje rychlost na menší než je aktuální rychlost vozidla. Nakonec je posunut čas o jednu sekundu. Takto se opět dostaneme do stavu, kdy je kontrolována ujetá vzdálenost, čímž je dokončena jedna iterace tohoto cyklu.



Obrázek 5.3: Automat, jenž řídí simulaci. Postupně předává řízení automatům pro řízení autonomního vozidla a automatu pro řízení neautonomních vozidel. Při poruše senzoru se předá řízení automatu pro jeho obsluhu.

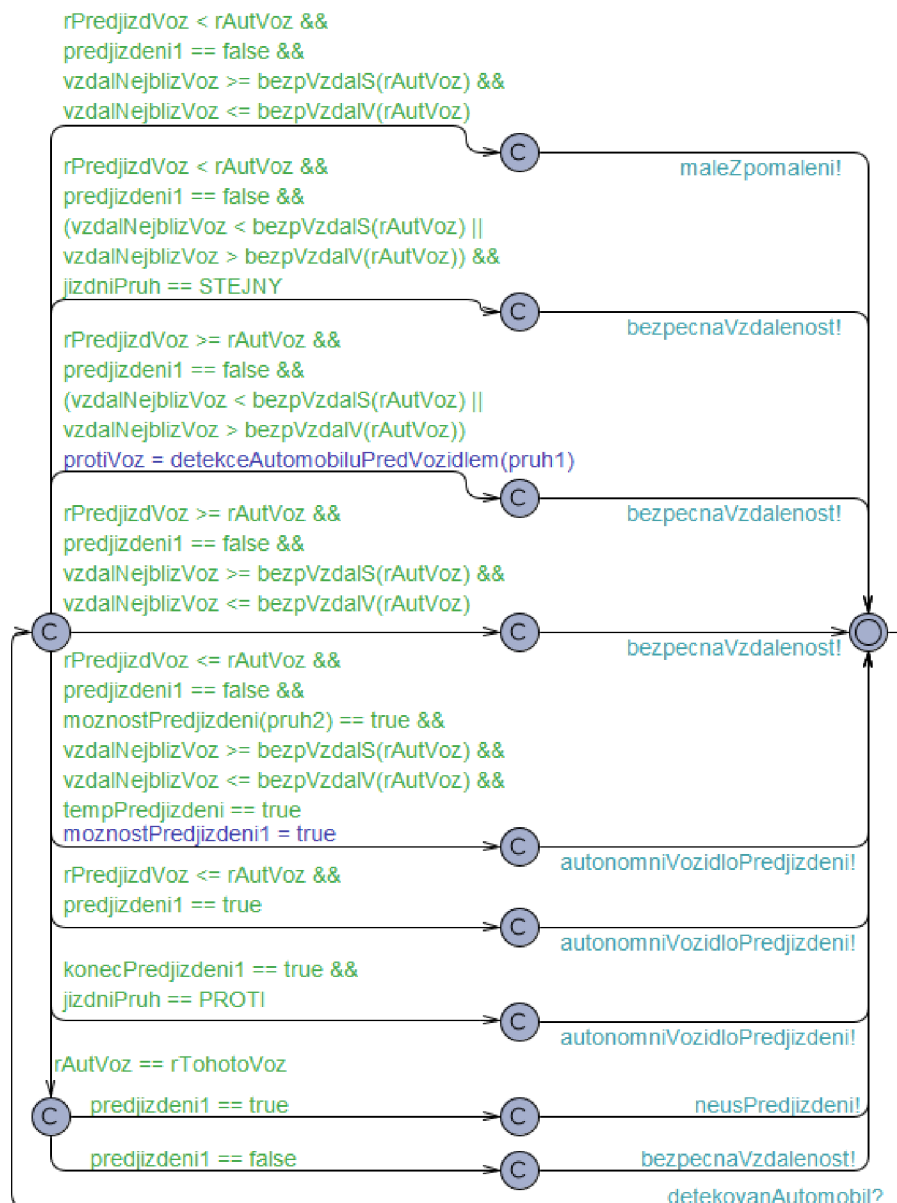
5.3.2 Výběr reakce autonomního vozidla

Tato činnost musela být rozdělena na několik menších automatů, které kontrolují hodnoty, podle kterých je možné vybrat akci autonomního vozidla. Signál `ovladaniAutVozidla` jde do automatu, jenž je na obr. 5.4. Ten se stará o předání řízení automatům, které vybírají činnost podle toho, zda se před autonomním vozidlem nachází, resp. nenachází nějaký objekt.



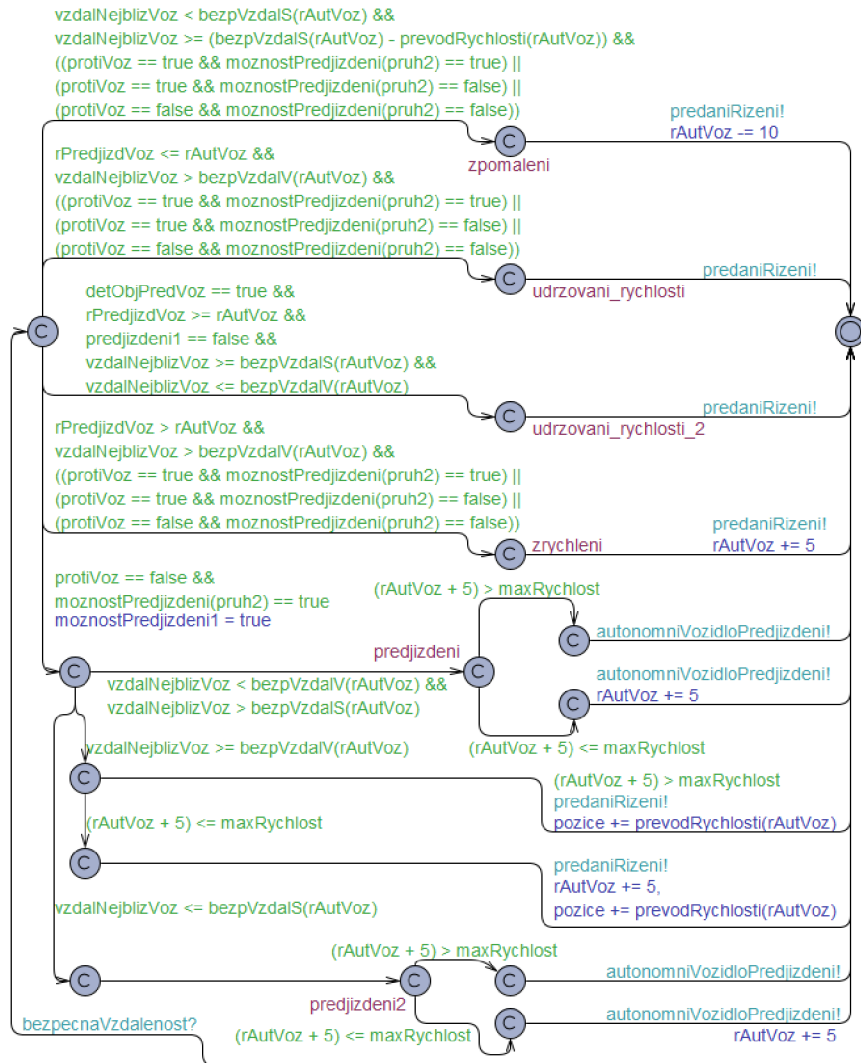
Obrázek 5.4: Automat, který rozhoduje, zda se před vozidlem nachází či nenachází překážka.

Nejprve se zaměříme na možnost, kdy se před autonomním automobilem nachází jedno či více vozidel. Tento automat je zobrazen na obr. 5.5. Čeká na signál `detekovanAutomobil` od předešlého automatu. Poté, pokud autonomní automobil zrovna neprovádí předjížděcí manévr, tak kontroluje poměr rychlostí autonomního vozidla s vozidlem jedoucím před ním a funkcemi `bezpVzda1S` a `bezpVzda1V` zkoumá, zda se nachází v intervalu bezpečné vzdálenosti. Dále zkoumá hodnotu proměnné `predjizdeni1`, která reprezentuje to, zda vozidlo provádí předjížděcí manévr. Podle vyhodnocení těchto hodnot probíhá rozhodnutí, zda si vozidlo bude držet bezpečnou vzdálenost, zpomalí, protože jede rychle a narazilo by do automobilu před ním, zpracovává předjížděcí manévr nebo došlo k neúspěšnému předjíždění. Ke každé reakci přísluší signál `maleZpomaleni`, `bezpecnaVzdalenost!`, `autonomniVozidloPredjizdeni!`, `neusPredjizdeni!`, resp. `neusPredjizdeni`.



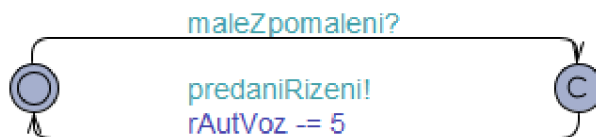
Obrázek 5.5: Automat, který vybírá akci, pokud autonomní vozidlo ve svém dohledu detekuje nějaký objekt.

Pokud předchodzí automat vyšle signál `bezpecnaVzdalenost`, což znamená, že autonomní vozidlo nepředjíždí, tak se spustí automat na obr. 5.6. Tady, pokud se v protisměru nenachází vozidlo, které je ve vzdálenosti, kdy by bylo možné jej detekovat, a podle výsledku funkce `moznostPredjizdeni`, která kontroluje, zda se po celé vzdálenosti, kdy dokáže počítačové vidění zpracovat obraz, nachází přerušovaná nebo plná čára. Pokud je předjíždění možné, pak se podle toho, zda se vozidlo nachází v intervalu bezpečné vzdálenosti, buď pošle signál `autonomniVozidloPredjizdeni`, kterým bude zahájen předjížděcí manévr, nebo je poslán signál `predaniRizeni`, který značí, že vozidlo je moc daleko na to, aby mohlo začít předjíždět, a proto se pokusí zrychlit. Pokud z nějakého důvodu předjíždění není možné, tak se vozidlo podle poměru mezi rychlostí autonomního vozidla a vozidla před ním, a zároveň podle jejich vzájemné vzdálenosti rozhoduje, zda bude udržovat svou rychlost, zpomalí nebo zrychlí. Poté se vyšle signál `predaniRizeni`, čímž se znovu spustí automat pro řízení simulace. Pokud se v protisměru nenachází žádné vozidlo, a zároveň je dostatečně blízko jinému vozidlu, tak je možné začít předjíždět vysláním signálu `autonomniVozidloPredjizdeni`.



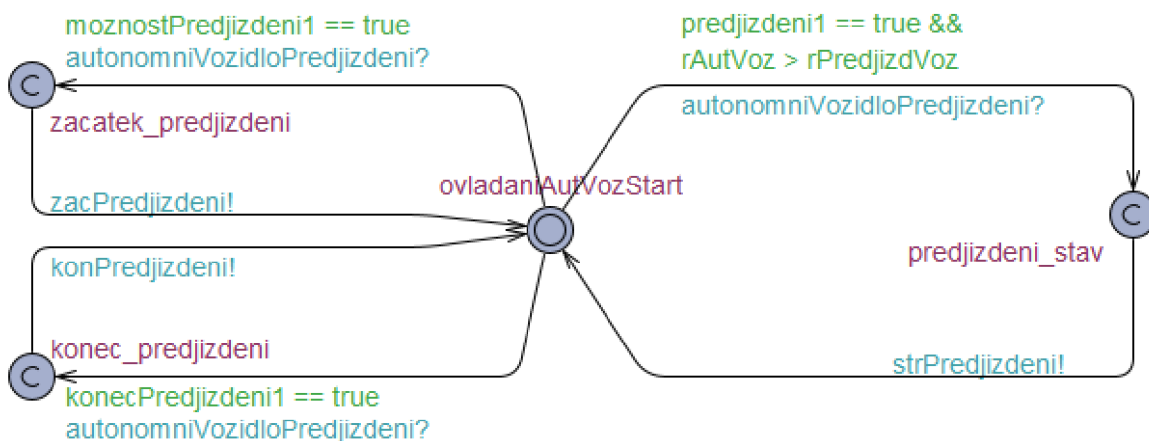
Obrázek 5.6: Automat, který slouží k tomu, aby se vozidlo nacházelo v intervalu bezpečné vzdálenosti od vozidla nacházejícího se před ním, nebo jej začalo předjíždět, pokud není vozidlo v protisměru.

Pokud byl vyslán signál `maleZpomaleni`, pak je vozidlo zpomalen o 5 *km/h*. Poté vysláním signálu `predaniRizeni` je předáno řízení modelu zpět automatu pro řízení simulace. Tuto činnost provádí automat na obr. 5.7.



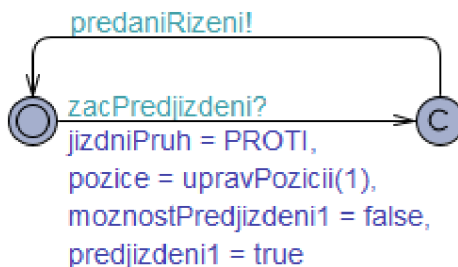
Obrázek 5.7: Automat, kterým se sníží rychlost vozidla o 5 *km/h*.

Chování při snaze předjíždět rozhoduje automat na obr. 5.8. Ten podle hodnot proměnných `moznostPredjizdeni1`, `predjizdeni1` a `konecPredjizdeni1` vybere akci, kterou bude autonomní vozidlo provádět s příchodem signálu `autonomniVozidloPredjizdeni`. Na základě hodnot proměnných je voleno, zda autonomní vozidlo začne předjíždět pomocí signálu `zacPredjizdeni`, provádí předjížděcí manévr, který ještě není možné ukončit signálem `strPredjizdeni`, nebo povede ukončení předjíždění signálem `konPredjizdeni`.



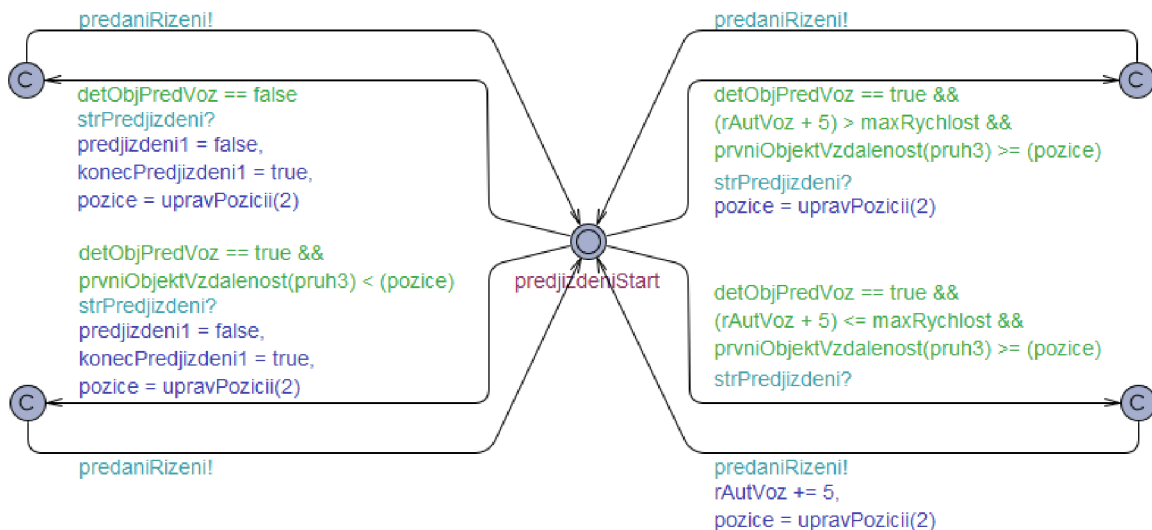
Obrázek 5.8: Automat, který rozhoduje o akci při předjíždění.

Následuje automat, který je na obr. 5.9, jenž se stará o začátek předjížděcího manévru autonomního vozidla. Jeho činnost je taková, že nejprve čeká na signál `zacPredjizdeni`, po jeho přijetí automobil přejede do druhého jízdního pruhu, upraví svou pozici a nastaví proměnné reprezentující předjíždění na odpovídající hodnoty. Poté signálem `predaniRizeni` předá řízení hlavnímu automatu.



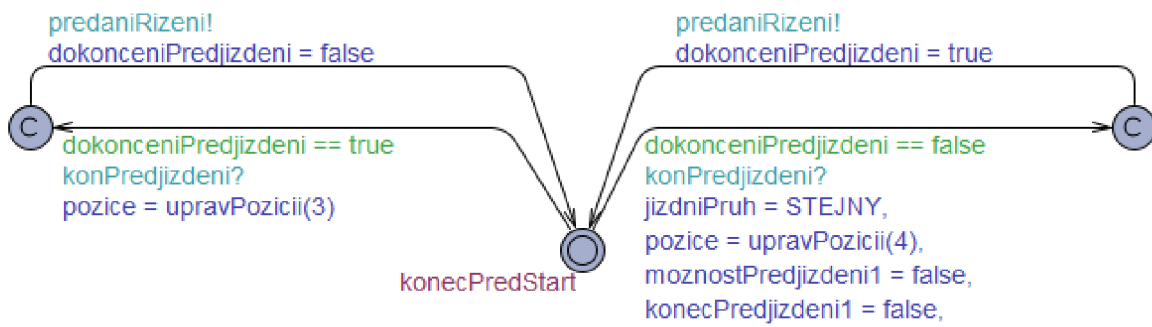
Obrázek 5.9: Automat, kterým vozidlo zahajuje předjížděcí manévr.

Ovládání autonomního vozidla při předjíždění je řízeno pomocí automatu na obr. 5.10, který je spuštěn signálem `strPredjizdeni`. Ten zajišťuje, že vozidlo bude opravdu předjeto pomocí funkce `prvniObjektVzdalenost`. Tato funkce je potřebná, protože může nastat situace, kdy už se nacházíme před předjížděným vozidlem, ale ve vzdálenosti detekce lidarů se nachází ještě jiné vozidlo, které ale neohrožuje bezpečné předjetí, nicméně je detekováno, čímž by neumožňovalo dokončit tento předjížděcí manévry. Protože chceme, aby byl tento manévry co nejrychlejší, tak se snažíme zvětšovat rychlost autonomního vozidla až na úroveň aktuální maximální povolené rychlosti. Právě proto, že se před předjížděným vozidlem může nacházet ještě jiné vozidlo, obsahuje tento automat dva stavy, které umožňují nastavit proměnné pro ukončení předjížděcího manévry.



Obrázek 5.10: Automat, který řídí vozidlo při předjížděcím manévru.

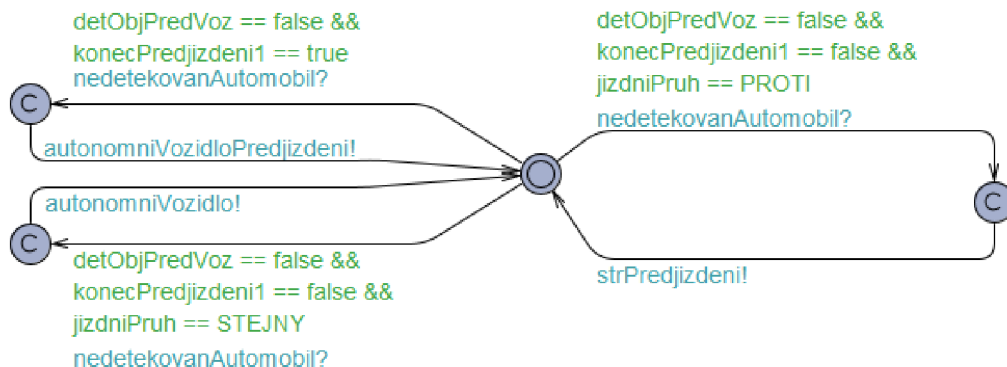
Ukončení předjížděcího manévry má na starost automat zobrazený na obr. 5.11. Jeho činnost je taková, že čeká na signál `konPredjizdeni`, díky čemu se nejprve autonomní vozidlo posune o vzdálenost odpovídající své rychlosti, a potom v čase o vteřinu větším je předjíždění ukončeno přesunem do správného jízdního pruhu, upravením pozice a nastavením proměnných na znamení, že vozidlo ukončilo předjíždění.



Obrázek 5.11: Automat, kterým vozidlo ukončuje úspěšný předjížděcí manévry.

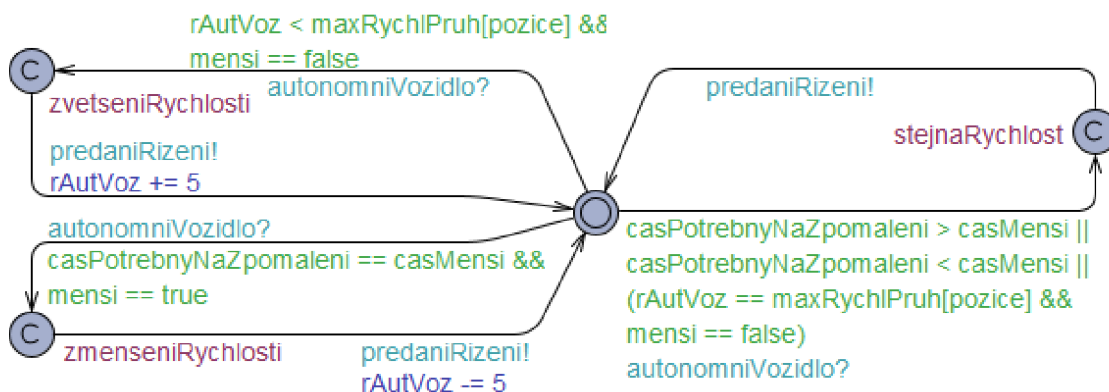
Automat na obr. 5.12. modeluje chování vozidla, pokud se v prostoru před ním nenachází žádný objekt, přičemž ke svému spuštění vyžaduje signál `nedetekovanAutomobil`. Tuto si-

tuaci, ale musíme rozdělit na dva možné případy, které se od sebe liší tím, v jakém jízdním pruhu se vozidlo nachází. Jestliže je vozidlo v pruhu, který odpovídá směru jízdy, pak je vše v pořádku a vozidlo může dále pokračovat v jízdě zasláním signálu `autonomniVozidlo`. Pokud je ovšem vozidlo v jízdním pruhu, ve kterém je směr jízdy opačný, tak víme, že se vozidlo snaží předjíždět a vzhledem k tomu, že již nedetekuje vozidlo před sebou, můžeme se přesunout k dokončení předjížděcího manévru, což je způsobeno zasláním signálu `strPredjizdeni`.



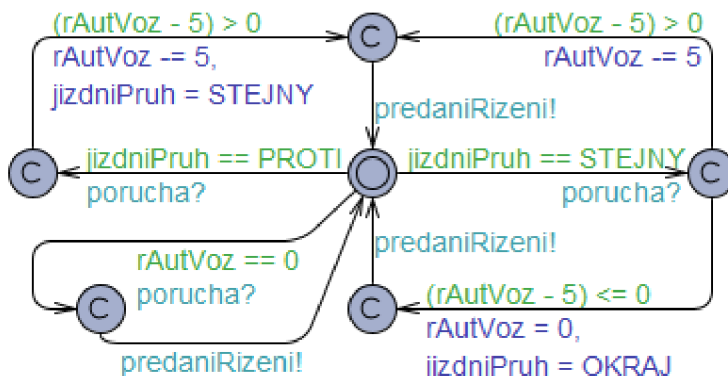
Obrázek 5.12: Automat, jenž volí činnost autonomního automobilu za předpokladu, že se před ním nic nenachází.

Pokud nebylo detekováno žádné vozidlo a byla detekována dopravní značka, tak se signálem `autonomniVozidlo` předá řízení automatu na obr. 5.13. Ten pokud detekuje značku která má stejnou nebo vyšší hodnotu rychlosti, tak ji v podstatě ignoruje a případné změny v rychlosti provede až poté co kolem této značky projede. Pokud je ovšem detekována značka, která má menší hodnotu povolené rychlosti než je aktuální rychlost vozidla tak, vozidlo nastaví proměnnou `mensi` na `true`, která vozidlu neumožní zrychlovat ani při projetí kolem značky, která povolenou rychlost zvýší. Vozidlo si vypočítá dobu, kterou bude potřebovat pro zpomalení na rychlost uvedenou na značce, kterou uloží do proměnné `casPotrebnyNaZpomaleni`. Do proměnné `casMensi` vloží hodnotu času, potřebného pro dojetí ke značce. Pomocí těchto dvou proměnných je potom zajištěno, že vozidlo bude zpomalovat v posledním časovém okamžiku tak, aby neporušilo dopravní předpis. Po provedení některé z těchto možností je řízení vráceno zpět automatu pro řízení simulace pomocí signálu `predaniRizeni`.



Obrázek 5.13: Automat, který se stará o reakci autonomního vozidla na dopravní značky.

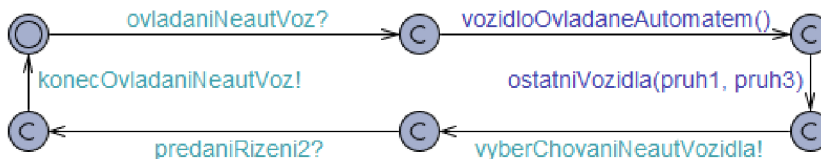
Poslední možností při předání řízení od automatu řízení simulace bylo to, že nastane chyba senzoru. Pokud tato skutečnost nastane, tak se stará o reakce autonomního vozidla automat na obr. 5.14. Ten podle jízdního pruhu, ve kterém se vozidlo nachází, rozhoduje, zda bude pouze zpomalovat vozidlo nebo se změnit i jízdní pruh. Před úplným zastavením vozidla je ještě odstaveno mimo vozovku.



Obrázek 5.14: Automat, který se stará o reakci autonomního vozidla při zjištění chyby u některého ze senzorů.

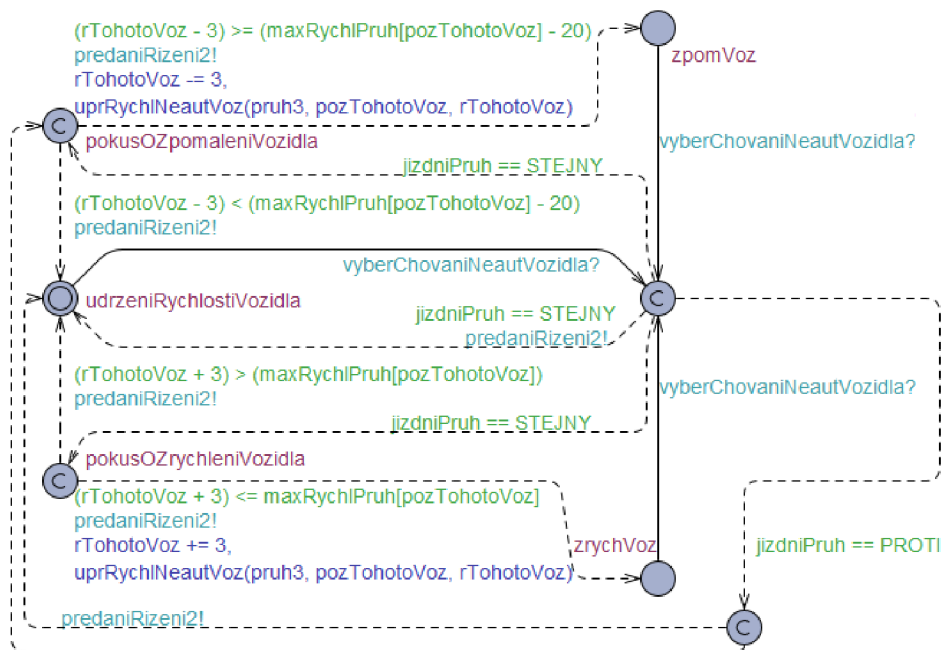
5.3.3 Řízení neautonomních vozidel

Pro větší zachycení reality jsou ovládány také všechna neautonomní vozidla, která se nacházejí v prostředí, konkrétně jsou ovládány jejich rychlosti. Protože zvolený nástroj neumožňuje dynamické vytváření instancí automatů, je potřeba vytvořit hlavní automat pro řízení neautonomních automobilů, který je zobrazen na obr. 5.15. Ten je spuštěn signálem `ovladaniNeautVoz`, poté nalezne vozidlo, které se nachází nejbližší autonomnímu vozidlu, protože právě tento automobil bude řídit jiný automat, který je na obr. 5.16. Ostatní vozidla budou řízena pomocí funkce `ostatniVozidla`. Po provedení této funkce se signálem `vyberChovaniNeautVozidla` předá řízení právě automatu pro ovládání nejbližšího lidsky ovládaného vozidla. Dále se čeká na signál `predaniRizeni2`, aby se mohlo ovládání modelu předat hlavnímu automatu pomocí signálu `konecOvladaniNeautVoz`.



Obrázek 5.15: Automat, který řídí úpravy rychlosti neautonomních vozidel.

Automat, který řídí nejbližší neautonomní vozidlo. To má na výběr mezi zpomalením, zůstáním na stejné rychlosti a zrychlením. Pokud je zvolena možnost zpomalení, pak je kontrolováno, zda by rychlost neklesla pod hodnotu `maxRychlost - 20`, protože poté by vozidlo jelo zbytečně pomalu, a byla by velká pravděpodobnost srážky. Naopak u zrychlení je kontrolováno, zda by vozidlo po zrychlení nepřekročilo maximální povolenou rychlost v daném úseku. Pokud autonomní vozidlo předjíždí, pak je vozidlu, které ovládá tento automat povoleno pouze zpomalovat nebo udržovat svou rychlost.



Obrázek 5.16: Automat, jenž ovládá vozidlo, které je nejbližší autonomnímu vozidlu.

5.4 Popis konstant, proměnných a funkcí

Tato kapitola se zabývá popisem některých konstant a proměnných použitých v modelu. Dále obsahuje popis některých funkcí, které jsou zmíněny v textu a není popsána jejich činnost. Všechny konstanty, proměnné a funkce jsou okomentovány v souboru, který obsahuje tento model.

Konstanty a proměnné

OKRAJ – konstanta označující místo kolem vozovky ve kterém autonomní vozidlo zastaví, pokud došlo k poruše senzoru

PROTI – konstanta označující jízdní pruh ve kterém je směr jízdy opačný než směr jízdy autonomního vozidla

STEJNY – konstanta označující jízdní pruh ve kterém je směr jízdy stejný jako směr jízdy autonomního vozidla

casMensi – doba za kterou autonomní vozidlo přesune ze současné pozice na úroveň dopravní značky s menší hodnotou rychlosti

casPotrebnyNaZpomaleni – hodnota vyjadřující čas potřebný pro zpomalení autonomního vozidla na rychlost určenou dopravní značkou

detObjPredVoz – vyjádření toho, zda je nebo není detekováno vozidlo před autonomním vozidlem

dokonceniPredjizdeni – označení toho, zda autonomní vozidlo dokončilo předjíždění

falesnyCas – proměnná, která vyjadřuje aktuální čas v datovém typu `integer`

jizdniPruh – reprezentace jízdního pruhu ve kterém se nachází autonomní vozidlo

konecPredjizdeni1 – proměnná signalizující konec předjíždění

maxRychlost – proměnná, která uchovává hodnotu aktuální maximální rychlosti

maxRychlPruh – pole obsahující hodnoty maximální povolené rychlosti na každém místě vozovky

maxVzdalenost – konstanta, která slouží k ohraničení délky cesty, pokud chceme provádět experimenty, kdy se chceme dostat do stavu **konec**
mensi – příznak detekce dopravní značky, která určuje menší rychlost než aktuální **rAutVoz**
moznostPredjizdeni1 – informace o tom, zda čára mezi pruhy umožňuje předjíždění
porSenzor – proměnná označující zda došlo k poruše senzoru
pozice – proměnná obsahující aktuální ujetou vzdálenost autonomního vozidla
predjizdeni1 – označuje, zda autonomní vozidlo provádí předjížděcí manévr
protiVoz – proměnná označující zda se v protisměru nachází detekovatelné vozidlo
rAutVoz – proměnná uchováající hodnotu aktuální rychlosti autonomního vozidla
rPredjizdVoz – rychlost vozidla, které autonomní vozidlo chce předjet
rTohotoVoz – rychlost nejbližšího vozidla
tempPredjizdeni – pomocná proměnná pro zjištění zda je možné předjíždět
vzdalNejblizVoz – vzdálenost mezi autonomním vozidlem a nejbližším vozidlem
waitTimer – proměnná, která slouží k posunu času modelu

Funkce

Minimální doporučená vzdálenost je určena jako vzdálenost, kterou vozidlo ujede za dvě vteřiny¹⁹. Pro snadnější modelování byla určena maximální doporučená vzdálenost jako vzdálenost, kterou vozidlo ujede za tři vteřiny. Pokud bude někde v textu zmíněn interval bezpečné vzdálenosti, tak je jím myšlena množina celých čísel, která obsahuje prvky ležící mezi výsledky funkcí **bezpVzdaLS** a **bezpVzdaLV** včetně jejich výsledků.

bezpVzdaLS(rAutVoz) – vypočítání dolního ohraničení doporučené bezpečné vzdálenosti.
bezpVzdaLV(rAutVoz) – vypočítání horního ohraničení doporučené bezpečné vzdálenosti.
detekceAutomobilu(pruh) – zjištění toho, zda se v jízdním pruhu daném parametrem nachází vozidlo, které je možné detekovat
dopravniZnackyVCase – detekce toho, že autonomní vozidlo projede kolem dopravní značky
moznostPredjizdeni(pruh2) – zjištění typu čáry jízdními pruhy **pruh1** a **pruh3**
mensiZnacka2 – detekce značky s menší hodnotou než je rychlost autonomního vozidla
ostatniVozidla(pruh1, pruh3) – nalezení vozidel nacházejících se v jízdních pruzích **pruh1** a **pruh3**
posunOstatnichVozidelPROTI(pruh1) – posun vozidel v protisměru jízdy autonomního vozidla, parametrem je pole reprezentující jízdni pruh
posunOstatnichVozidelSTEJNY(pruh3) – posun vozidel ve směru jízdy autonomního vozidla, parametrem je pole reprezentující jízdni pruh
prevodRychlosti – funkce, která převede rychlost z jednotky *km/h* na *m/s*
prvniObjektVzdalenost – zjištění vzdálenosti mezi autonomním vozidlem a prvním objektem, který je možné detekovat, a zároveň se nachází před autonomním vozidlem.
upravPozicii(param) – upravení proměnné **pozice**, parametr je pro výběr větve funkce
uprRychlNeautVoz(pruh3, pozTohotoVoz, rTohotoVoz) – upravení hodnoty rychlosti vozidla ovládaného automatem na obr. 5.16
vozidloOvladaneAutomatem – nalezení vozidla, které bude řízeno automatem na obr. 5.16
vzdalMeziVoz – vypočítání vzdálenosti mezi autonomním vozidlem a nejbližším vozidlem, které je před ním

¹⁹https://www.cspds.cz/storage/files/vzdalenost_mezi_vozidly.pdf

Kapitola 6

Vyhodnocení vlastností modelu

Tato kapitola se zabývá experimenty, kterými byly zkoumány vlastnosti modelu. Tyto experimenty jsou rozděleny podle činnosti, která je ověřována. Každý dotaz pro simulaci obsahuje proměnnou `rRychlost`, která je určena pouze pro korektní vykreslení grafů a z grafu je odstraněna.

6.1 Reakce vozidla na jiné vozidlo

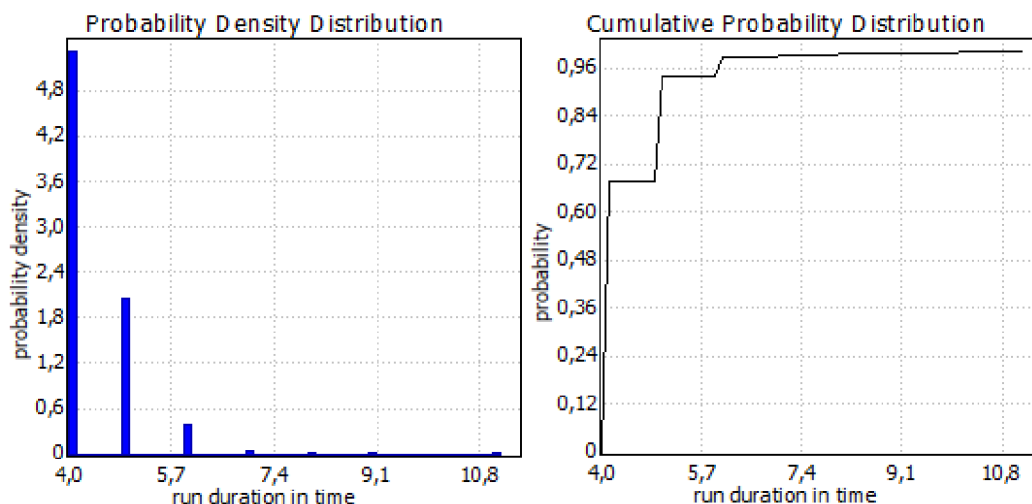
Tyto experimenty se zabývají chováním autonomního vozidla vzhledem k okolí, ve kterém se nachází jedno či více lidsky řízených vozidel. V těchto experimentech jsou neautonomní vozidla řízena náhodně, proto vykonáním daného dotazu nad stejnými vstupními parametry není zaručen výsledek, který bude naprosto stejný. Parametr nejistoty pravděpodobnosti (ε) je roven hodnotě 0.0005 pro všechny experimenty.

6.1.1 Udržování bezpečné vzdálenosti

V tomto experimentu se před samočinně řízeným vozidlem nachází jeden automobil, který je ve vzdálenosti, jež je větší než doporučená bezpečná vzdálenost. Mezi jízdními pruhy je plná čára, proto není možné předjíždět. Za těchto podmínek je úkolem autonomního vozidla přiblížit se k vozidlu řízenému lidskou bytostí na vzdálenost, která bude v intervalu bezpečné vzdálenosti. Je také předpokládáno, že autonomní vozidlo upraví svou rychlost tak, aby se v tomto intervalu udrželo. Pro ověření této vlastnosti byl vytvořen dotaz 6.1, který se ptá, jaká je pravděpodobnost, že v čase $\langle 0, 60 \rangle$ bude vzdálenost mezi autonomním vozidlem a vozidlem v intervalu bezpečné vzdálenosti.

$$\begin{aligned} Pr[<= 60] (<> ((vzdalenostMeziVozidly > bezpVzdaS(rAutVoz)) \\ \&\& (vzdalenostMeziVozidly < bezpVzdaV(rAutVoz)))) \end{aligned} \quad (6.1)$$

Výsledkem je interval pravděpodobnosti $[0.999, 1]$, tedy že v nejméně 99% případů bude vzdálenost mezi vozidly v intervalu bezpečné vzdálenosti. Tato skutečnost je zobrazena na obr. 6.1.

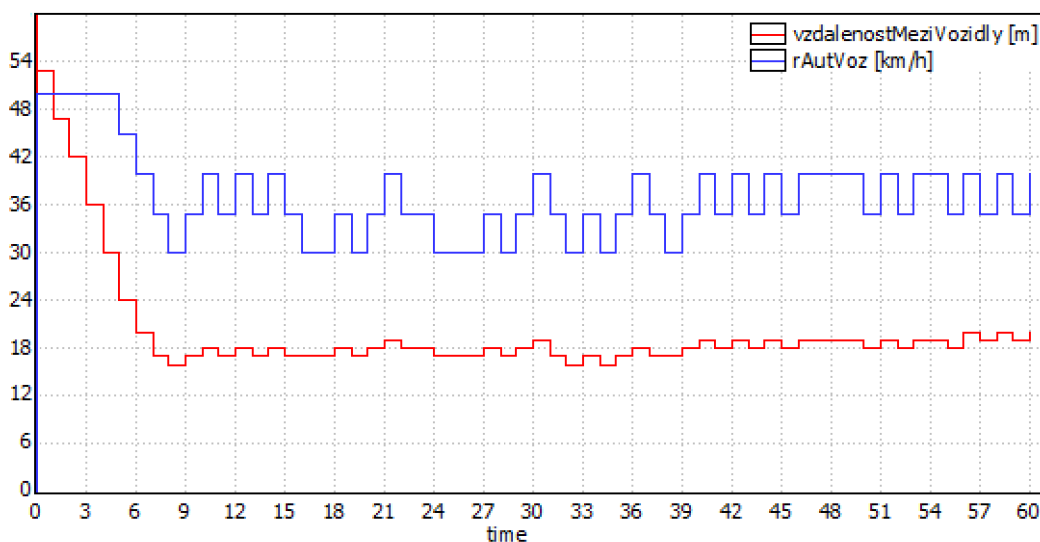


Obrázek 6.1: Z grafu hustoty pravděpodobnosti je možné zjistit, že autonomní vozidlo se dostane do intervalu bezpečné vzdálenosti mezi časy 4 až 11 sekund. Distribuční funkce potvrzuje domněnku, protože v čase 11 sekund se její křivka blíží jedničce.

Nad stejnými daty byla spuštěna simulace dotazem 6.2, pro ověření toho, že model se skutečně udržuje za vozidlem, které se na silnici nachází před ním.

$$\text{simulate } 1 \text{ } [\leq 60] \{ \text{vzdalenostMeziVozidly}, r\text{AutVoz}, r\text{Rychlost} \} \quad (6.2)$$

Výsledek této simulace je zobrazen na obr. 6.2. Z grafu je možné vyčíst, že vozidlo upravilo svou rychlost, tak aby nenarazilo do automobilu před ním, a vzdálenost mezi vozidly se ustálila v intervalu $\langle 16, 20 \rangle$ metrů.

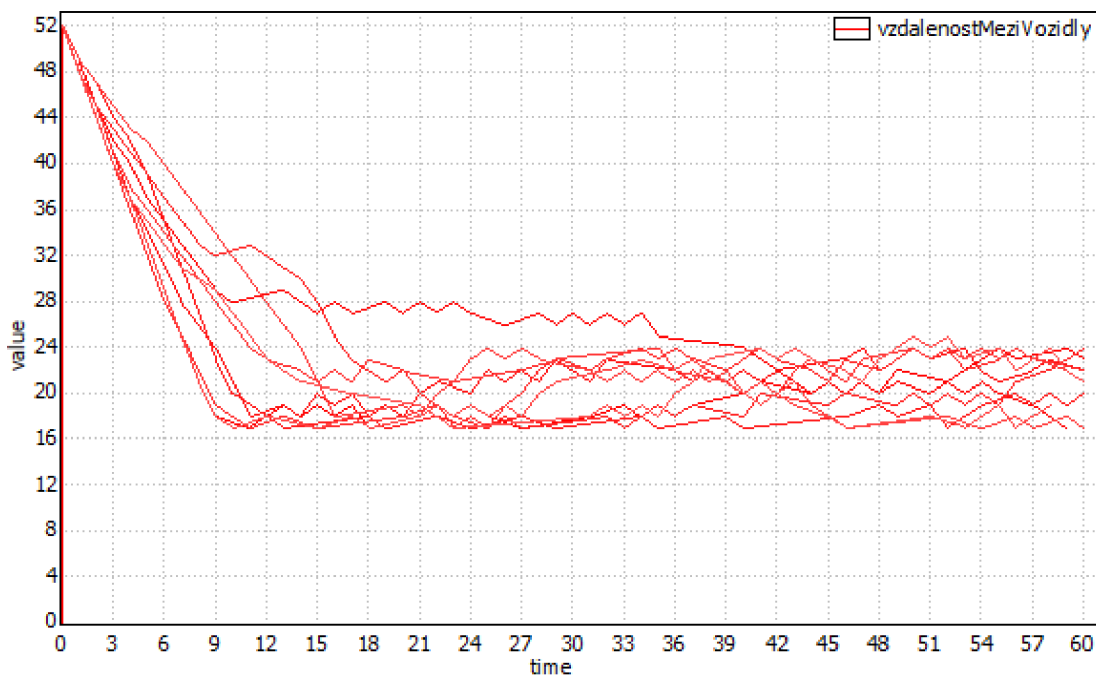


Obrázek 6.2: Graf znázorňující vzdálenost mezi vozidly, v grafu znázorněno červenou čarou, a rychlost autonomního vozidla v čase, v grafu modrou čarou. Z grafu je patrné, že vzdálenost mezi vozidly ze své počáteční hodnoty klesne až k hodnotě 16 metrů, což odpovídá hodnotě dolní hranice bezpečné vzdálenosti, která je pro rychlost 30 km/h rovná 16 m . U této hodnoty tato hodnota osciluje, ale nikdy pod ní neklesne.

Dále byl sledován vývoj vzdálenosti při spuštění více simulací. Toto je možné pomocí dotazu 6.3, který spustí deset simulací zároveň.

$$\text{simulate } 10 \text{ } [\leq 60] \{ \text{vzdalenostMeziVozidly} \} \quad (6.3)$$

Výsledek odpovídá očekávání, kdy se autonomní vozidlo přiblíží k jinému vozidlu a pohybuje se v intervalu bezpečné vzdálenosti. V tomto případě nikdy neklesne pod 16 metrů.



Obrázek 6.3: Graf demonstrující vzdálenost mezi vozidly při spuštění simulace desetkrát. Je možné pozorovat, že vzdálenost mezi vozidly klesne k hodnotě 20 metrů, a v celém zbytku času se pohybuje kolem této hodnoty.

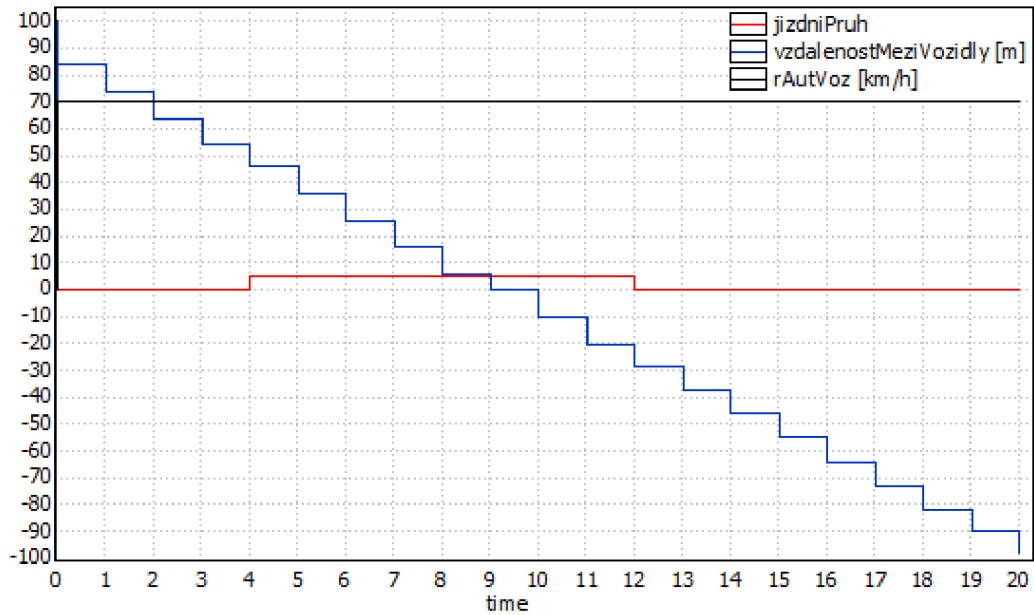
Experiment dopadl podle očekávání. Bylo potvrzeno, že pokud vozidlo nebude moci předjet vozidlo, které se nachází v jeho směru jízdy, pak přizpůsobí svou rychlost tak, aby si od něj udržovalo bezpečnou vzdálenost.

6.1.2 Předjíždění jednoho vozidla

V tomto experimentu má autonomní vozidlo za úkol předjet jedno vozidlo, které jede rychlostí 30 km/h a nachází se ve vzdálenosti 85 m od samočinně řízeného vozidla. Samotné autonomní vozidlo jede rychlostí 70 km/h. Maximální povolená rychlost na silnici je 70 km/h, která nebude změněna žádnou dopravní značkou. Pro demonstraci funkčnosti slouží dotaz 6.6, který zkoumá změnu jízdního pruhu, vývoj vzdálenosti mezi vozidly a rychlost autonomního vozidla.

$$\text{simulate } 1 \text{ } [\leq 20] \{ \text{jizdniPruh}, \text{vzdalenostMeziVozidly}, \text{rAutVoz}, \text{rRychlost} \} \quad (6.4)$$

Výsledek tohoto dotazu je zobrazen v grafu na obr. 6.4. Z něj je možné vyčíst, že vozidlo si bude držet stále stejnou rychlost, díky čemu se bude neustále přibližovat automobilu, který chce předjet. Jakmile se dostane do rozmezí bezpečné vzdálenosti, tak započne předjížděcí manévry, který ukončí po dostání celé své karoserie před předjížděné vozidlo.



Obrázek 6.4: Graf zobrazující předjetí obyčejného vozidla autonomním vozidlem. Nejprve se autonomní vozidlo přiblíží do vzdálenosti, která leží v intervalu bezpečné vzdálenosti. V čase 4 sekundy započne samotný předjížděcí manévr, kdy autonomní vozidlo přejede do druhého pruhu (červená čára). Vzhledem k tomu, že již jede maximální povolenou rychlostí, tak si stále udržuje stejnou rychlost (černá čára). Vzdálenost mezi vozidly je znázorněna modrou čarou. K ukončení předjížděcího manévru dojde v čase 12 sekund.

Zajímají nás pravděpodobnosti počátečního času začátku předjížděcího manévru a jeho konce. Nejprve začátek předjížděcího manévru. Toto rozložení pravděpodobnosti je možné získat pomocí dotazu. 6.5.

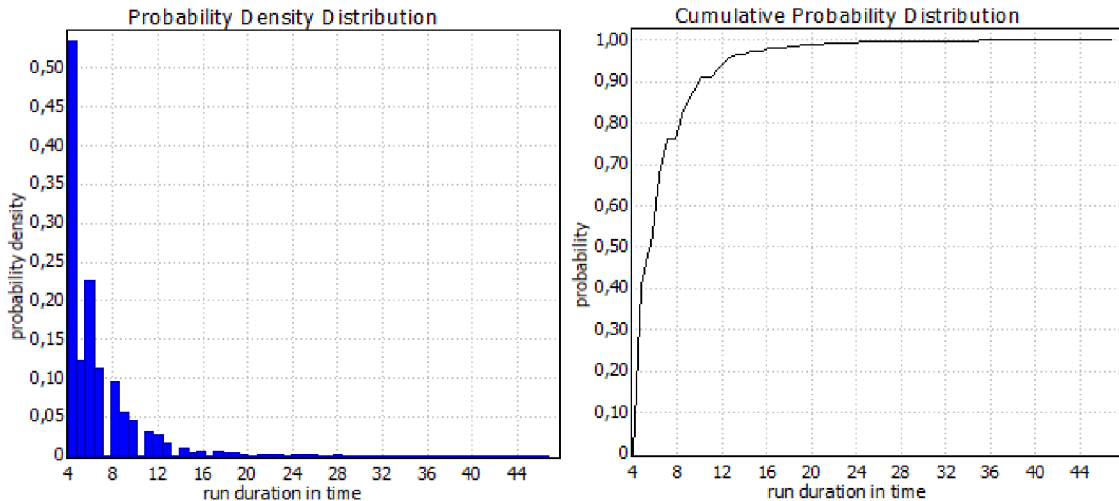
$$Pr[<= 60] (<> (jizdniPruh == PROTI)) \tag{6.5}$$

Interval $[0.9999, 1]$ je výsledkem tohoto dotazu, tedy předjíždění v naprosté většině případů bude zahájeno. Graf hustoty pravděpodobnosti a distribuční funkce je k vidění na obr. 6.5.

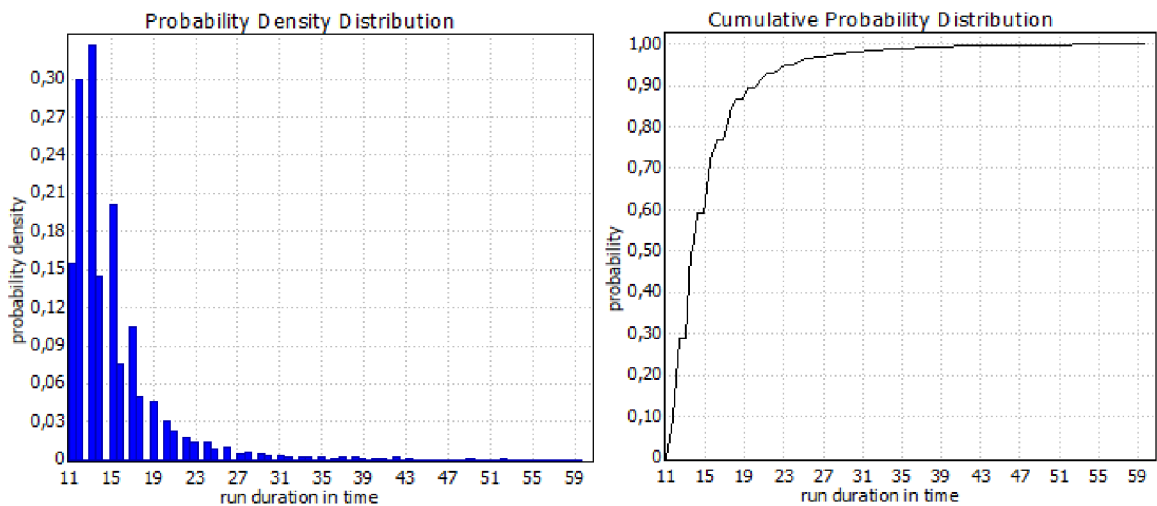
Dále nás zajímá pravděpodobnost, kdy autonomní vozidlo dokončí předjíždění. Dotaz 6.6 umožňuje zjistit hodnoty pravděpodobnosti pro různé časové okamžiky..

$$Pr[<= 60] (<> ((vzdalenostMeziVozidly < 0) \&\& jizdniPruh == STEJNY)) \tag{6.6}$$

Výsledkem je interval $[0.998969, 0.999995]$, tedy předjíždění bude do 60 vteřin dokončeno s velmi velkou pravděpodobností. Následují grafy hustoty pravděpodobnosti a distribuční funkce na obr. 6.6.



Obrázek 6.5: Z grafu hustoty pravděpodobnosti je možné zjistit, že předjížděcí manévř s největší pravděpodobností započne v čase 4 sekundy. Graf distribuční funkce ukazuje, že v čase 20 vteřin už bude s pravděpodobností 98% předjížděcí manévř zahájen.



Obrázek 6.6: Graf zobrazující hustotu pravděpodobnosti, která se nachází na levé straně, a graf distribuční funkce na pravé straně. Z grafu hustoty pravděpodobnosti vyplívá, že největší šance na ukončení předjíždění je v čase 13 sekund. Dále z distribuční funkce je možné vyvodit, že v čase 23 sekund již bude na 96% předjíždění ukončeno.

Experiment potvrdil očekávání, že model autonomního vozidla dokáže předjet jedno vozidlo. Při počáteční rychlosti 70 km/h a předjížděným automobilem, který začíná s rychlostí 30 km/h , započne předjížděcí manévř nejdříve ve 4 vteřině a skončí nejdříve v čase 11 sekund.

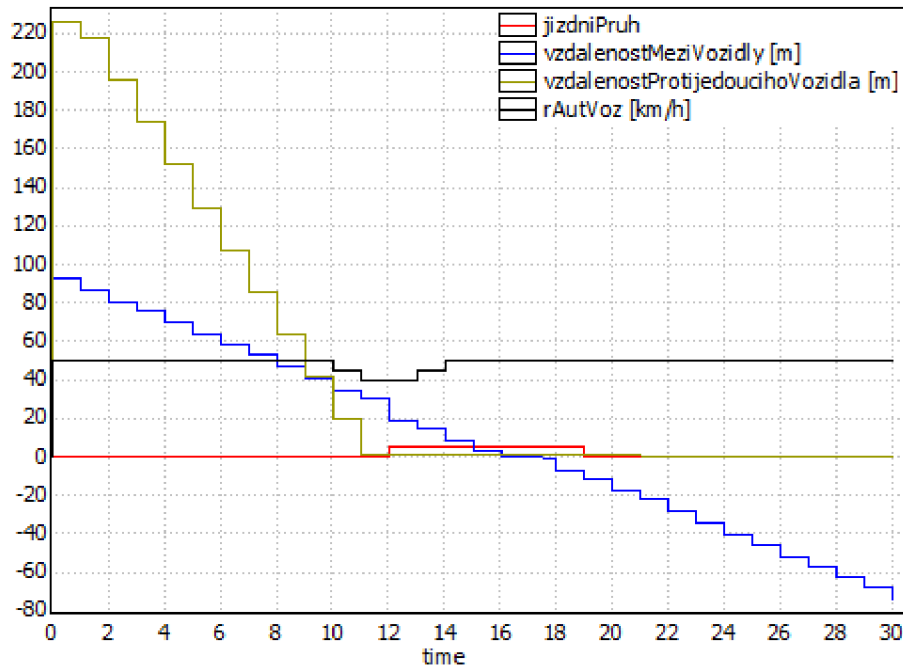
6.1.3 Předjíždění jednoho vozidla, pokud se nachází vozidlo i v protisměru

Tento experiment zkoumá předjíždění vozidla za předpokladu, že se v protisměru nachází vozidlo, které brání okamžitému začátku předjížděcího manévřu. Tento automobil je od

autonomního vozidla vzdálen 220 m a jede rychlostí 30 km/h. Samočinně řízené vozidlo začíná s rychlostí 50 km/h. Vozidlo, jež se snažíme předjet má rychlost 30 km/h a nachází se ve vzdálenosti 50 m. Díky těmto hodnotám autonomní vozidlo nemůže provést předjížděcí manévr hned, ale je nutné počkat než vozidlo jedoucí v protisměru projede kolem samočinně řízeného vozidla, a teprve poté je možné začít předjížděcí manévr. Pro ověření funkčnosti je možné spustit simulaci dotazem 6.7.

$$\text{simulate } 1 \ [<= 30] \ \{jizdniPruh, vzdalenostProtijedoucihoVozidla, vzdalenostMeziVozidly, rAutVoz, rRychlost\} \quad (6.7)$$

Provedením dotazu 6.7 získáme graf nacházející se na obr. 6.7.



Obrázek 6.7: Graf zobrazující předjetí obyčejného vozidla autonomním vozidlem, když se nachází v protisměru detekovatelné vozidlo. Nejprve se autonomní vozidlo přiblíží k vozidlu, které chce předjet, až na spodek intervalu bezpečné vzdálenosti (modrá čára). Zároveň lze pozorovat, že musí mírně brzdit (černá čára). Následně čeká, až kolem něj projede vozidlo v protisměru (zelenohnědá čára), tedy hodnota vzdalenostProtijedoucihoVozidla se dostane na nulu. Poté je možné zahájit předjížděcí manévr a úspěšně předjet vozidlo.

Dále je potřeba ověřit, že vozidlo nikdy nevjede do protisměru, pokud se v něm bude nacházet vozidlo, se kterým by mohlo dojít ke srážce. To s jakou pravděpodobností se může stát tento případ, vyhodnotí dotaz 6.8.

$$\text{Pr}[<= 60] \ (<> \ ((vzdalenostProtijedoucihoVozidla > 0) \ \&\& \ (jizdniPruh == \text{PROTI}))) \quad (6.8)$$

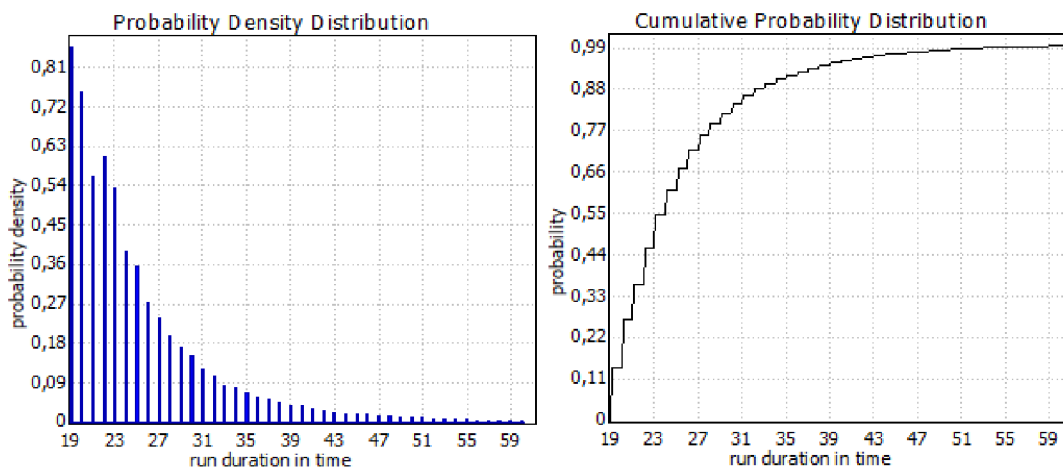
Jeho výsledkem je interval $[0, 0.01]$. Tedy, že existuje asi 1% šance, že tato situace může nastat. Dále je možné využít dotaz 6.9, který zjišťuje, zda když se vozidla nachází ve stejném jízdním pruhu, tak jsou buď od sebe vzdálena na vzdálenost větší než dolní ohraničení

doporučené bezpečné vzdálenosti, nebo je vzdálenost menší než nula, což značí předjetí vozidla.

$$Pr[<= 60] (<> (((vzdalenostMeziVozidly > bezpVzdaS(rAutVoz)) || (vzdalenostMeziVozidly < 0)) \&\& jizdniPruh == STEJNY)) \quad (6.9)$$

Výsledkem dotazu je pravděpodobnost z intervalu $[0.999, 1]$, který potvrzuje očekávání, že mezi autonomním vozidlem a automobilem pohybujícím se před ním bude vzdálenost alespoň spodek intervalu bezpečné vzdálenosti, nebo se bude samočinně řízené vozidlo nacházet před ním.

Využitím dotazu 6.5, z předchozího experimentu, můžeme ověřit s jakou pravděpodobností začne předjížděcí manévru v jistém časovém okamžiku. Výsledný interval tohoto dotazu je $[0.995616, 0.996616]$ což udává, že k zahájení předjížděcího manévru dojde v 99,5 až 99,6% případů. Doba potřebná pro vykonání tohoto dotazu při nezměněném parametru ε se pohybuje kolem 45 minut. Graf hustoty pravděpodobnosti a distribuční funkce je k vidění na obr. 6.8.



Obrázek 6.8: Z grafu hustoty pravděpodobnosti je patrné, že předjíždění s největší pravděpodobností začne v čase 19 sekund a pro každý další časový okamžik klesá pravděpodobnost započetí předjíždění. Distribuční funkce ukazuje, že v čase 60 sekund byl téměř jistě předjížděcí manévru již započat.

Experimentem bylo ověřeno, že pokud se nachází v protisměru vozidlo, které zabraňuje započetí předjížděcího manévru, protože by v případě provádění předjížděcího manévru hrozila srážka, tak model vozidla dokáže počkat do doby, než bude silnice volná. Poté se již autonomní vozidlo bude chovat, jako když pouze předjíždí vozidlo.

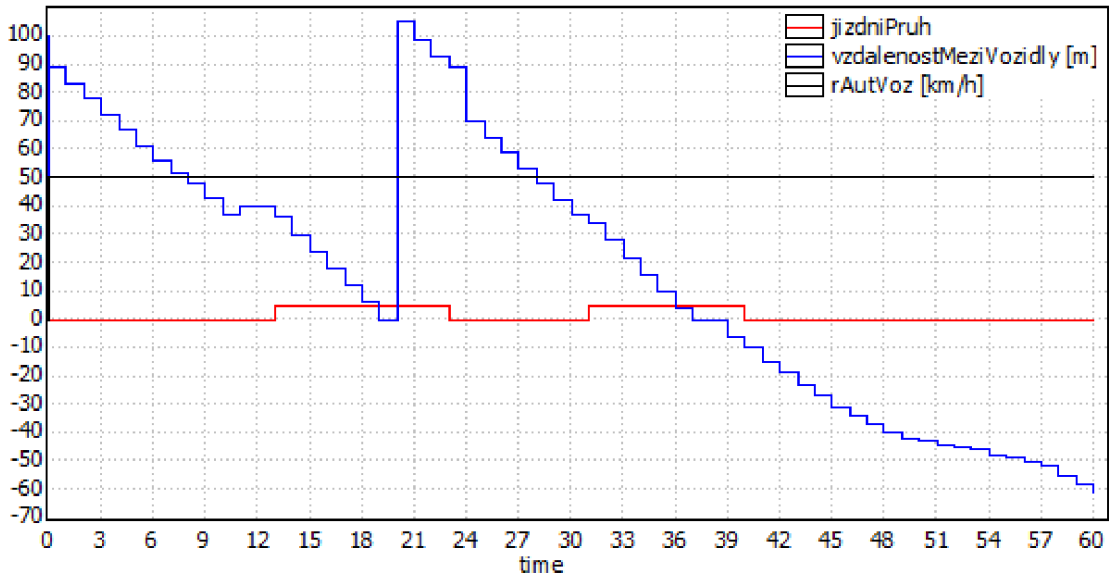
6.1.4 Pokus o předjetí více než jednoho vozidla

Tento experiment se zabývá tím, zda autonomní vozidlo dokáže předjet více než jedno vozidlo. Situace je taková, že samočinně řízené vozidlo začíná s počáteční rychlostí 50 km/h , což je také maximální povolená rychlost. Dále se v systému nacházejí dvě vozidla s lidskými řidiči, obě mají rychlost 30 km/h a nacházejí se 100 m , respektive 220 m .

Využitím dotazu pro simulaci 6.10, je možné získat jeden průběh tohoto experimentu. Jeho výsledek je k vidění na obr. 6.9.

$simulate\ 1\ [<= 60]\ \{jizdniPruh,\ vzdalenostMeziVozidly,\ rAutVoz,\ rRychlost\}$ (6.10)

Nejprve se autonomní vozidlo přiblíží k vozidlu, které chce předjet a započne předjížděcí manévr. Po jeho dokončení se přibližuje k dalšímu vozidlu, které také úspěšně předjede. U tohoto experimentu je nutné zmínit, že ne vždy se podaří předjet obě dvě vozidla, protože automat, respektive funkce, která řídí vozidla, vybírá náhodně, zda vozidlo zpomalí, bude držet stejnou rychlost nebo zrychlí. Tento nedeterminismus může posunout předjíždění druhého vozidla na pozdější čas nebo dokonce zabránit tomuto předjetí.

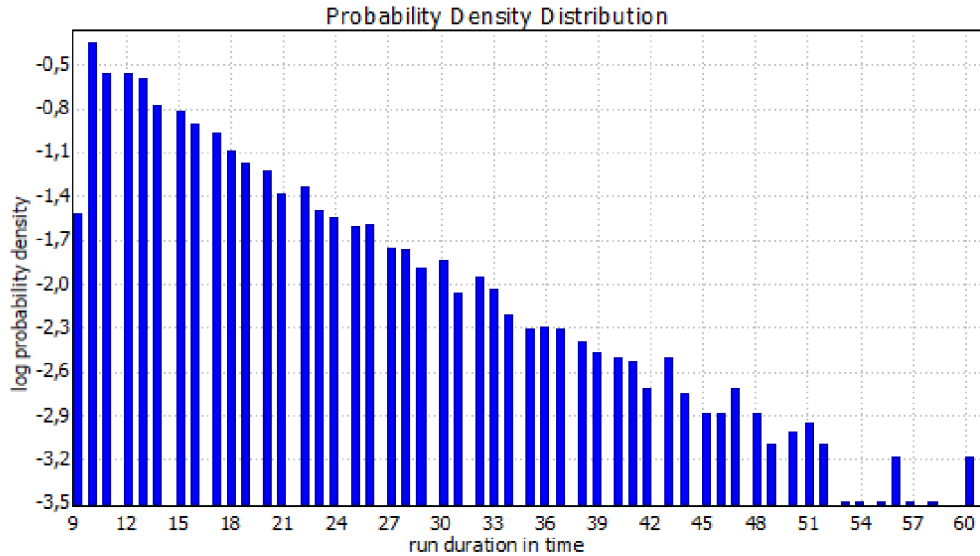


Obrázek 6.9: Graf zobrazující předjetí dvou obyčejných vozidel autonomním vozidlem. Je možné pozorovat postupné přibližování autonomního vozidla k automobilu, což znázorňuje modrá čára. Autonomní vozidlo nemůže zrychlit, protože jede maximální povolenou rychlostí (černá čára). Po dostání se na dost blízkou vzdálenost samočinně řízené vozidlo započalo předjížděcí manévr (červená čára). Po úspěšném dokončení předjížděcího manévru byla stejná posloupnost kroků zopakována i pro předjetí dalšího automobilu, který se nacházel ve stejném směru jízdy jako autonomní vozidlo.

Dále je možné využít dotaz 6.11, který každé vteřině z intervalu $\langle 0, 60 \rangle$ přiřadí pravděpodobnost, že se samočinně řízené vozidlo bude nacházet v protisměru.

$$Pr[<= 60] (\langle \rangle (jizdniPruh == PROTI)) \quad (6.11)$$

Výsledná funkce hustoty pravděpodobnosti je k vidění na obr. 6.10, kde má y-nová osa logaritmické měřítko pro lepší znázornění pravděpodobností. Tento experiment se zabýval předjetím více vozidel za předpokladu, že není žádné vozidlo v protisměru. Výsledek simulace odpovídá očekávání, kdy se vozidlo po předjetí prvního vozidla přibližovalo ke druhému vozidlu, a to poté bylo také úspěšně předjeto.



Obrázek 6.10: Graf hustoty pravděpodobnosti, kde y-nová osa má logaritmické měřítko. Z grafu je možné zjistit, že doba předjetí prvního vozidla nastane s největší pravděpodobností okolo 10 vteřiny. Díky logaritmickému měřítku osy, graf připomíná s jistou mírou představivosti graf lineární funkce. Pravděpodobnosti, které z této křivky vybočují, označují pravděpodobné časy pro předjetí druhého vozidla.

6.1.5 Pokus o předjetí více než jednoho vozidla s vozidlem v protisměru

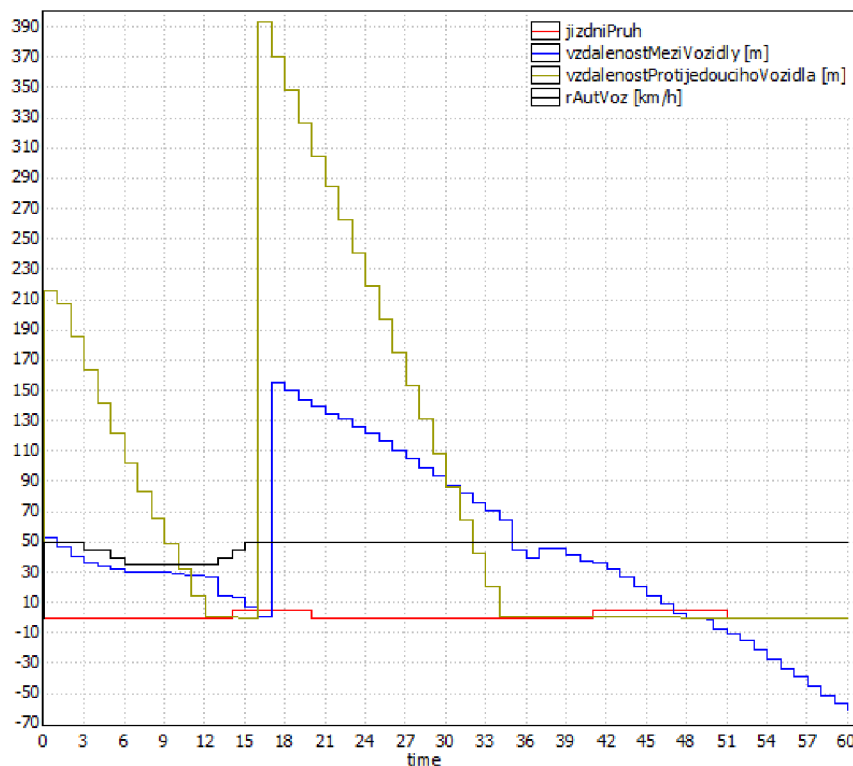
Tento experiment se zaměřuje na pokus o předjetí více vozidel za předpokladu, že se v protisměru nachází vozidlo. Počáteční rychlost autonomního vozidla je 50 km/h . Všechny ostatní vozidla mají počáteční rychlost 30 km/h . První vozidlo v protisměru je vzdáleno 220 m , druhé je ve vzdálenosti 700 m . Vozidla ve směru jízdy samočinně řízeného vozidla se nacházejí 100 m a 220 m před autonomním vozidlem. Cílem tohoto experimentu je zjistit, zda autonomní vozidlo dokáže bez nehody předjet obě vozidla nacházející se v jeho směru jízdy.

Nejprve využijeme simulaci, kterou provedeme pomocí dotazu 6.12. Její výsledek je zobrazen na obr. 6.11.

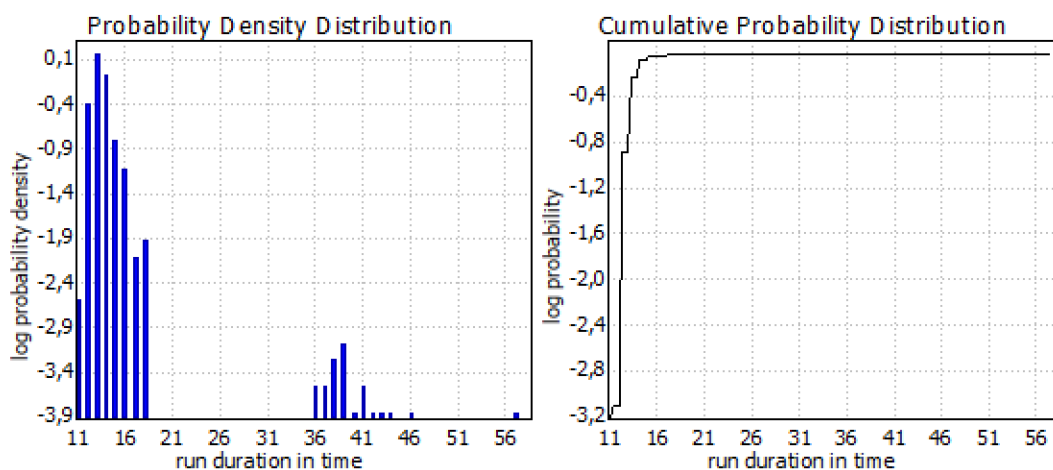
$$\text{simulate } 1 \ [\leq 60] \ \{ \text{jízdníPruh, vzdalenostProtijedoucíhoVozidla,} \\ \text{vzdalenostMeziVozidly, rAutVoz, rRychlost} \} \quad (6.12)$$

Dále nás zajímají pravděpodobnosti v jakých časech započnou předjížděcí manévry. Tyto hodnoty lze získat pomocí dotazu z předchozího experimentu 6.11. Výsledný graf hustoty pravděpodobnosti a distribuční funkce je k vidění na obr. 6.12. Oba dva grafy mají y-novou osu s logaritmickým měřítkem.

Tímto byl ověřen cíl tohoto experimentu, tedy že autonomní vozidlo dokáže počkat, dokud kolem něj neprojde vozidlo nacházející se v protisměru, a teprve potom zahájit předjížděcí manévr. Na tomto experimentu byla tato vlastnost demonstrována dvakrát, při předjíždění dvou vozidel s dvěma vozidly v protisměru.



Obrázek 6.11: Graf zobrazující předjetí dvou obyčejných vozidel autonomním vozidlem. Nejprve se autonomní vozidlo přiblíží k vozidlu, které chce předjet a započne předjížděcí manévř. Po jeho dokončení se s menšími problémy přibližuje ke druhému vozidlu, které také úspěšně předjede. U tohoto experimentu je nutné podotknout, že ne vždy se podaří předjet obě dvě vozidla protože, automat, respektive funkce, která řídí vozidla vybírá náhodně zda vozidlo zpomalí, bude držet stejnou rychlost nebo zrychlí.



Obrázek 6.12: Z grafu hustoty pravděpodobnosti zobrazuje pravděpodobnosti lze vyčíst, že zahájení předjíždění prvního vozidla se uskuteční někdy mezi v časy 11–18 vteřin. Předjíždění druhého vozidla nastane s největší pravděpodobností okolo hodnoty 39 vteřin, ale s jistou malou pravděpodobností může být také druhý předjížděcí manévř zahájen v časovém okamžiku 57 sekund.

6.2 Experimenty s ověřením konkrétního cíle

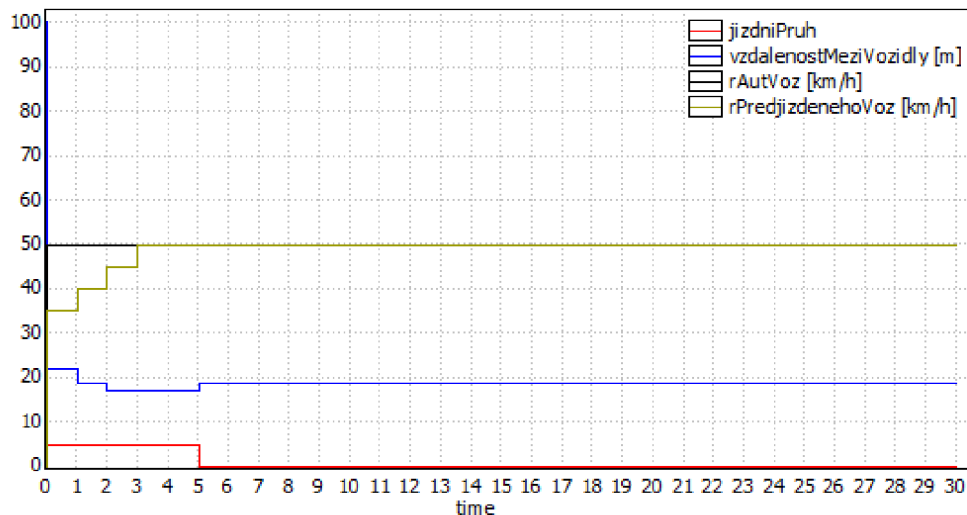
Tato kapitola se zabývá experimenty, pro které je nedeterministické chování neautonomních vozidel nahrazeno tak, aby tyto vozidla vykonaly předem danou činnost. Dále je v některých experimentech pozměno chování některých automatů.

6.2.1 Předjíždění vozidla, které začne zrychlovat

Tento experiment se zabývá chováním autonomního vozidla v situaci, kdy předjíždí jiné vozidlo, které po zjištění, že se jej někdo snaží předjet, tak zrychlí na maximální povolenou rychlost. Cílem autonomního vozidla by mělo být předjetí daného vozidla, ale pokud zjistí, že jede stejnou rychlostí, tak přeruší předjížděcí manévry a vrátí se do jízdního pruhu, který odpovídá směru jízdy. Autonomní vozidlo má počáteční rychlost rovnu 50 km/h a jede v pruhu, který odpovídá směru jízdy. Neautonomní vozidlo začíná s rychlostí 30 km/h se vzdáleností 30 m od samočinně řízeného vozidla. Experiment vyžaduje omezení chování neautonomních vozidel tak, že budou moci pouze zvýšit svou rychlost, což se provede úpravou automatu na obr. 5.16, kde budou smazány hrany, které by umožňovali zpomalení a udržení stejné rychlosti za předpokladu, že je možné zrychlit. Dotaz 6.13 slouží pro ověření této situace pomocí simulace.

$$\text{simulate } 1 \text{ [} \leq 30 \text{]} \{ \text{jizdniPruh, vzdalenostMeziVozidly,} \quad (6.13) \\ \text{rAutVoz, rRychlost, rTohotoVoz} \}$$

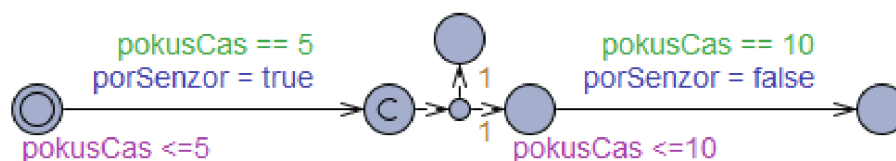
Výsledek dotazu je na obr. 6.13. Je možné pozorovat, že autonomní vozidlo okamžitě zahájilo předjížděcí manévry. Na tuto událost zareagovalo předjížděné vozidlo, a postupně začalo zvyšovat svou rychlost, až se dostalo na rychlost, která je v daném úseku maximální. Samočinně řízené vozidlo zjistilo, že automobil, který chtělo předjet zvýšil svou rychlost na stejnou hodnotu, jakou jede samo, a protože není možné zrychlit, tak je ukončen předjížděcí manévry neúspěšně.



Obrázek 6.13: Z grafu je možné zjistit, že předjížděné vozidlo okamžitě začalo zrychlovat až na rychlost 50 km/h (zelenohnědá čára). Autonomní vozidlo si drží svou rychlost, která je také 50 km/h , protože se vzdálenost mezi vozidly (modrá čára) nemění, tak je předjíždění ukončeno v čase 5 sekund.

6.2.2 Reakce na chybu senzoru

Tento experiment se zabývá tím, co se stane, pokud za jízdy autonomnímu vozidlu přestane fungovat některý ze senzorů. Předpokládá se, že řízení a brzdy jsou v pořádku. Cílem tohoto experimentu je zkontrolovat, zda vozidlo zastaví. Situace je taková, že autonomní vozidlo jede rychlostí 50 km/h po cestě, na které se nenachází žádné jiné vozidlo. Když náhle přestane fungovat některý ze senzorů, čímž se nastaví proměnná `porSenzor` na hodnotu `true`. Očekávanou reakcí vozidla je to, že okamžitě začne brzdit, a při zabrzdění na rychlost 0 km/h se přesune mimo vozovku. Aby to nebylo tak jednoduché, existuje 50% šance, že se jednalo o planý poplach a sensor znovu funguje korektně. Schéma automatu, který řídí generování chyby senzoru, je na obr. 6.14.

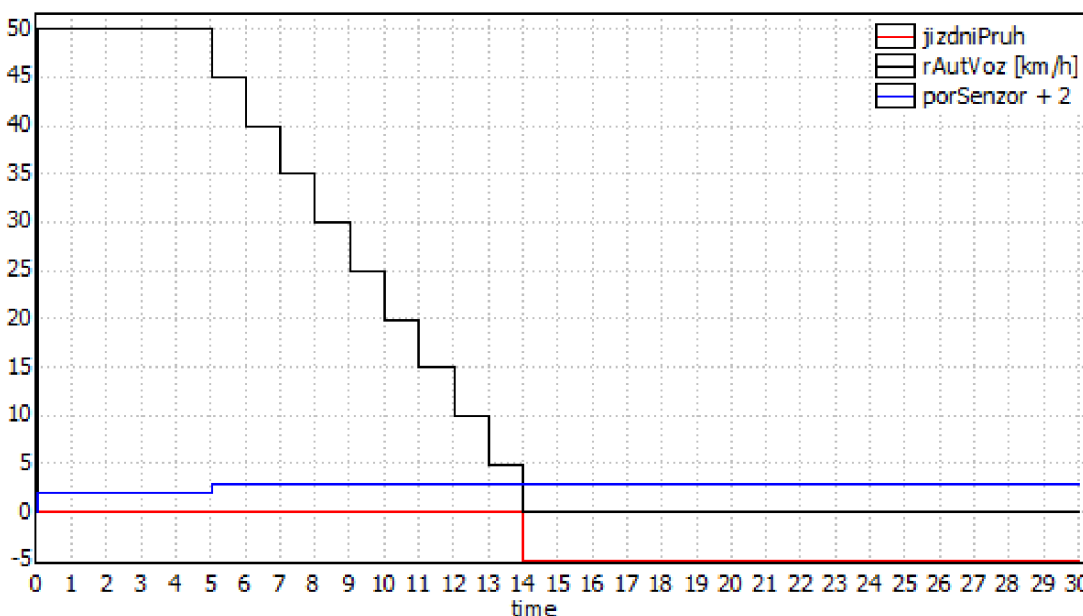


Obrázek 6.14: Automat, jenž má na starost generování chyby senzoru. Po pěti vteřinách vygeneruje chybu. Existuje 50% šance, že sensor nebyl chybný, ale došlo jen ke krátkému výpadku spojení po dobu pěti sekund.

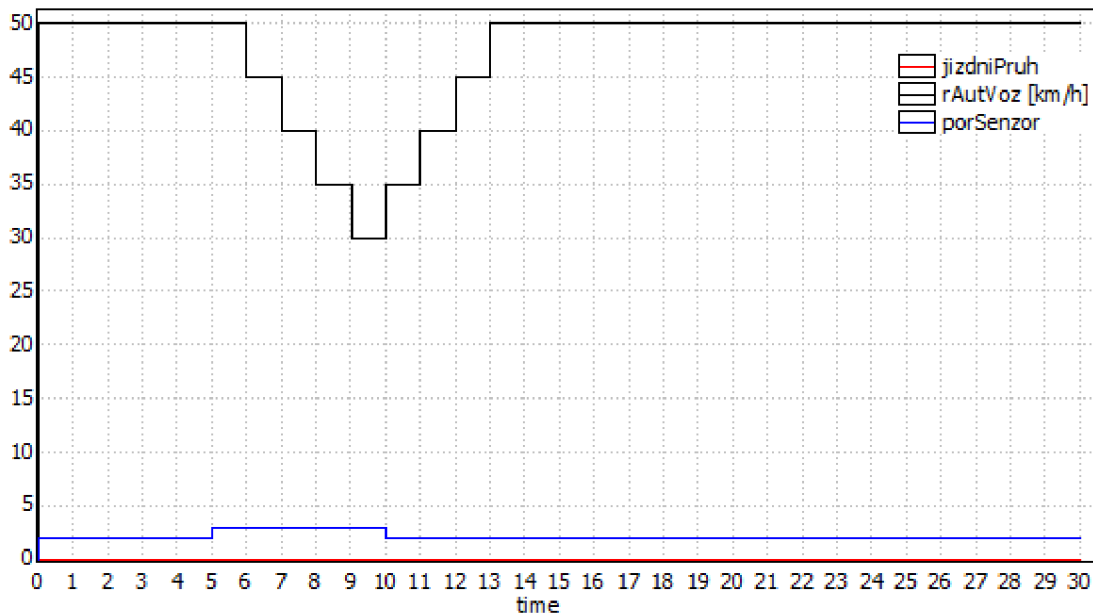
Ověření, že mohou nastat obě dvě situace, poskytuje následující dotaz 6.14, který zkoumá rychlost autonomního vozidla.

$$\text{simulate } 1 \text{ [} \leq 30 \text{] } \{ \text{jizdniPruh, } r\text{AutVoz, } \text{porSenzor} + 2 \} \quad (6.14)$$

Výsledky dotazu jsou zobrazeny na obrázcích 6.15 a 6.16.



Obrázek 6.15: Situace, kdy sensor opravdu obsahuje chybu, která nastane v čase 5 sekund (modrá čára). Autonomní vozidlo (černá čára) zastaví po deseti sekundách a přejeđe do odstavného pruhu (červená čára).



Obrázek 6.16: Situace, kdy došlo k výpadku spojení mezi řídicí jednotkou a senzorem v čase 5 sekund (modrá čára). V čase 10 sekund bylo spojení opět obnoveno. Po obnovení spojení samočinně řízené vozidlo opět akcelerovalo na rychlost 50 km/h (černá čára).

Cíl tohoto experimentu byl splněn, kdy bylo předvedeno, že pokud nastane chyba senzoru, tak autonomní vozidlo zareaguje brzděním až do zastavení. Pokud se ovšem jednalo o chybu spojení se senzorem a toto spojení bylo obnoveno, vozidlo zrychlí a může opět pokračovat v jízdě.

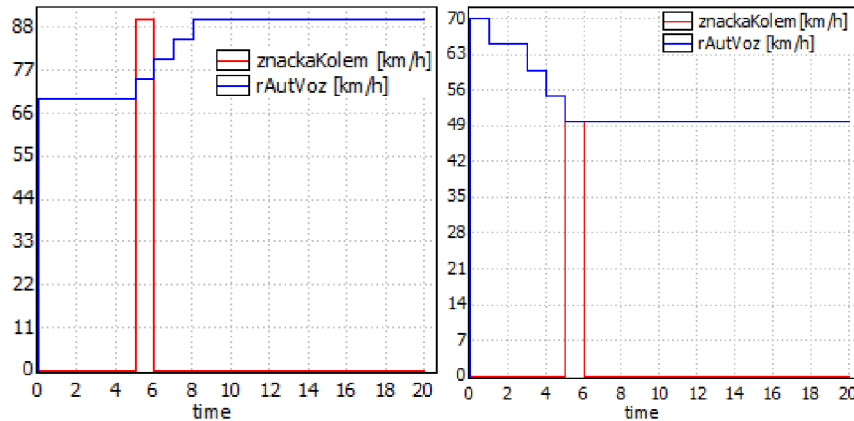
6.2.3 Reakce na dopravní značky ovlivňující rychlost

Následující experiment se zabývá reakcí autonomního vozidla na dopravní značky, které omezují maximální dovolenou rychlost. Autonomní vozidlo začíná na rychlosti 70 km/h a ve vzdálenosti 100 m se nachází dopravní značka, která zvyšuje maximální povolenou rychlost na hodnotu 90 km/h . Očekávaná reakce je taková, že autonomní vozidlo začne zrychlovat na tuto rychlost nejdříve v okamžiku, kdy projede kolem této značky. Pro ověření je použit následující dotaz 6.15. Ten zkoumá hodnotu rychlosti autonomního vozidla v závislosti na hodnotě značky kolem, které projede.

$$simulate 1 [<= 20] \{ značkaKolem, rAutVoz \} \quad (6.15)$$

Nyní zkusíme reakci na značku, která upravuje maximální povolenou rychlost na hodnotu menší než je aktuální rychlost autonomního vozidla. Experiment je opět ověřen pomocí dotazu 6.15. Tentokrát je očekáváno, že vozidlo nejpozději při projetí kolem značky pojede rychlostí, která je právě na dopravní značce. Výsledky simulace pro tyto dvě nastavení jsou k vidění na obr. 6.17.

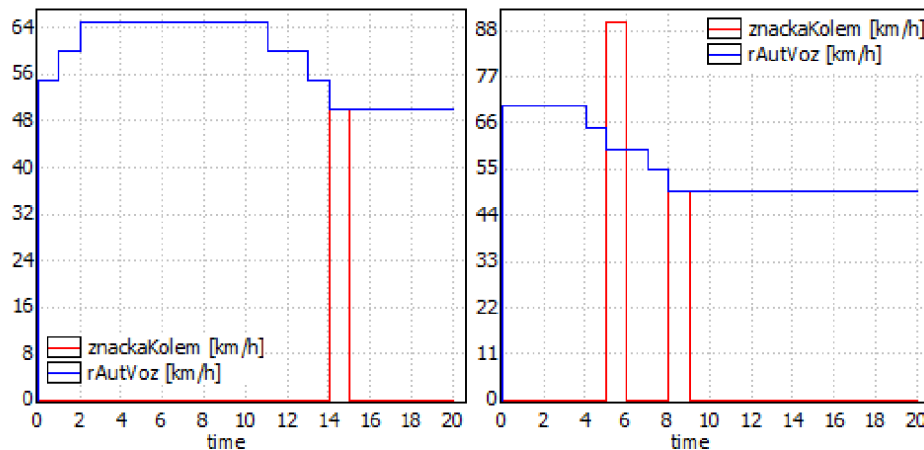
Další reakce na značku bude trochu komplikovanější. Nejprve nastavíme počáteční hodnotu rychlosti autonomního vozidla na 55 km/h a nastavíme, že nejvyšší povolená rychlost bude 90 km/h . Následně umístíme značku s hodnotou 50 km/h do vzdálenosti 250 m . Výsledkem této situace by mělo být zrychlení a následné zpomalení na hodnotu 50 km/h .



Obrázek 6.17: Levý obrázek ukazuje zvýšení rychlosti autonomního vozidla (modrá čára) po projetí kolem značky (červená čára od 0 do 90), což odpovídá očekávanému výsledku pro první vstup. Obrázek napravo ukazuje schopnost autonomního vozidla (modrá čára) zabrzdit tak, že přesně při průjezdu kolem značky pojede rychlostí uvedenou na značce (červená čára od 0 do 50).

Zrychlení by mělo být možné, protože se dopravní značka na začátku nachází mimo dohled modelu autonomního vozidla.

Poslední zkoumání budeme provádět s dvojicí dopravních značek. Samočinně řízené vozidlo bude začínat na rychlosti 70 km/h a značky budou mít hodnoty 90 km/h a 50 km/h a budou umístěny ve vzdálenosti 100 m, respektive 150 m. Předpokládaný výsledek je takový, že vozidlo přijede ke značce s hodnotou 50 km/h rychlostí 50 km/h. Obě dvě reakce je opět možné získat pomocí dotazu 6.15. Jeho výsledky jsou zobrazeny na obr. 6.18.



Obrázek 6.18: Levý obrázek ukazuje zvýšení rychlosti autonomního vozidla (modrá čára) před detekováním značky s hodnotou 50 km/h. Po její detekci si vozidlo udržuje rychlost a začne zpomalovat až v okamžiku, kdy mu zpomalení přesně vyjde, aby při průjezdu kolem ní jel hodnotou 50 km/h. Druhý případ se zaměřuje na dvě značky, které jsou od sebe vzdáleny v krátké vzdálenosti. Protože pro vozidlo je důležitější značka, která zpomaluje, tak autonomní vozidlo nemůže zrychlit ani, když projede kolem značky s hodnotou 90 km/h (červená čára od 0 do 90). Vozidlo mezitím začalo brzdit, aby stihlo upravit svou rychlost, tak, aby kolem druhé značky projelo správnou rychlostí.

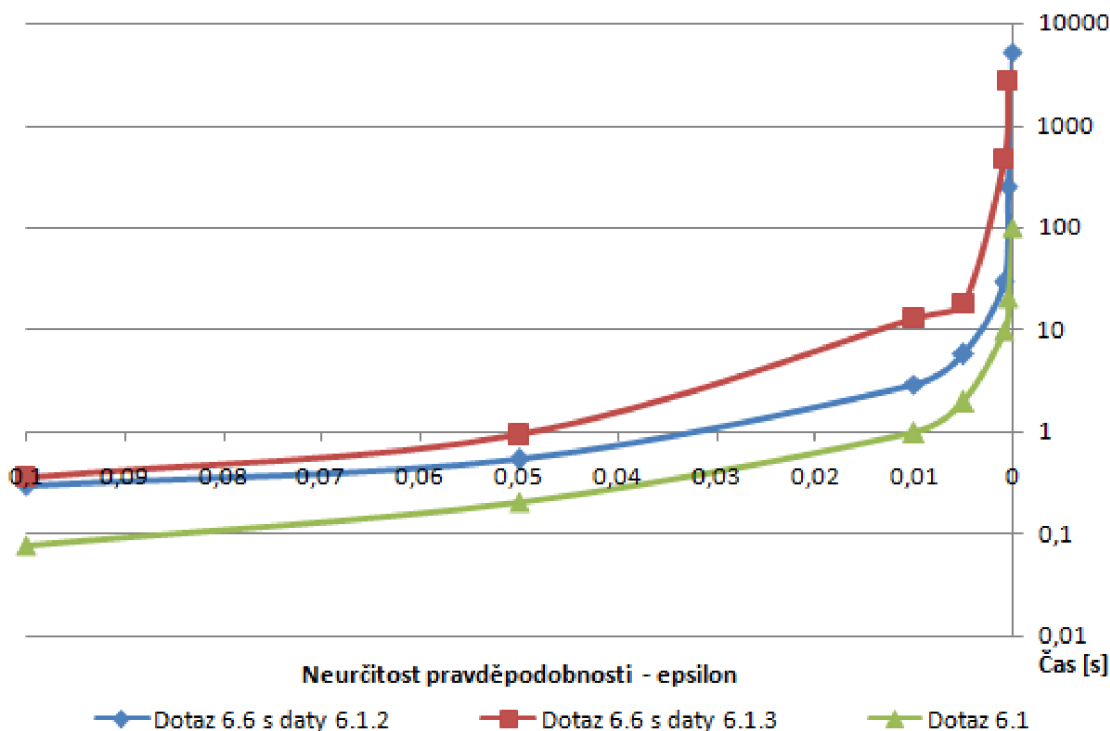
Výsledky pro všechny vstupy dopadly přesně podle očekávání, kdy pokud vozidlo přijíždí ke značce s menší rychlostí, než jakou samo jede, tak zpomalí takovým způsobem, aby jelo požadovanou rychlostí co nejkratší dobu před značkou. Na druhou stranu za předpokladu detekce značky s vyšší hodnotou rychlosti, vozidlo začne zrychlovat až poté, co kolem této značky projede.

6.3 Výpočetní náročnost

Tato podkapitola se zaměřuje na experiment s dobou trvání pravděpodobnostního dotazu. Pomocí změny parametru ε byl vytvořen experiment, který zkoumá dobu trvání pravděpodobnostního dotazu, pokud tento parametr měníme. Výsledky se nacházejí v tabulce 6.1. Hodnoty z tabulky byly vyneseny do grafu, jenž se nachází na obr. 6.19. Z grafu lze získat informace o tom, že doba běhu pravděpodobnostního dotazu narůstá se snižováním hodnoty parametru ε exponenciálně.

epsilon	0.1	0.05	0.01	0.005	0.001	0.0005	0.0001
dotaz 6.1	0,078	0.203	0.969	1.937	9.578	20.64	97.781
dotaz 6.6 s daty z 6.1.2	0.297	0.531	2.86	5.672	28.937	248.125	5243.906
dotaz 6.6 s daty z 6.1.3	0.36	0.922	12.829	17.812	458.922	2728.172	-

Tabulka 6.1: Tabulka, která obsahuje délku trvání pro dva dotazy, kde jeden byl použit dvakrát s jinými vstupními daty, s různou hodnotou parametru statistického parametru ε .



Obrázek 6.19: Závislost doby běhu pravděpodobnostního dotazu na parametru ε .

Kapitola 7

Závěr

V této diplomové práci byla nastíněna problematika samočinně řízených vozidel. Ve druhé kapitole byly vymezeny pojmy týkající se problematiky této práce, vytvořil jsem přehled senzorů, které obsahují samočinně řízená vozidla, popis jejich jízdy a shrnutí současného i možného budoucího stavu tohoto odvětví. Třetí kapitola shrnuje nástroje pro modelování a simulaci, kterými je možné vytvořit model vozidla tohoto typu. Čtvrtá kapitola důkladněji popisuje nástroj UPPAAL Stratego, který byl vybrán pro realizaci praktické části této diplomové práce. Pátá kapitola se zabývá návrhem a vytvářením simulačního modelu, kde popisuje, jaké skutečnosti bylo možné modelovat a popisuje samotný vytvořený model vozidla, které je možné považovat za autonomní, protože se řídí pomocí dat, jež jsou získána pomocí senzorů. Vlastnosti tohoto modelu byly ověřeny pomocí experimentů, které se skládaly ze situací, s nimiž se může samočinně řízené vozidlo setkat v běžném provozu. Provedené experimenty potvrdily, že se model chová tak, jak očekával autor této práce.

Jako možné rozšíření této práce by bylo možné rozšířit funkcionalitu modelu o další vlastnosti, kde by se například mohly vyskytovat kamiony nebo lidský faktor v podobě chodců. Dále by mohla být důkladněji prozkoumána oblast poruch senzorů a případného chování autonomních vozidel, pokud tato okolnost nastane. Jako další zajímavý námět by mohlo být řešení parkování na různých typech parkovišť. Samozřejmě se také nabízí možnost udělat model více odpovídající skutečnosti, tedy například přidat do modelu model brzd, který by více odrážel realitu nebo namodelovat akceleraci podle skutečného automobilu. V neposlední řadě by bylo možné se zabývat jinou oblastí samočinně řízených vozidel, konkrétně jízdě více vozidel v koloně. V tomto případě by se mohla řešit komunikace mezi jednotlivými modely, které vozidlo zodpovídá za určitou činnost, jaký je optimální počet vozidel ve skupině, aby nebyla narušena plynulost provozu.

Literatura

- [1] Awad, E.; Dsouza, S.; Kim, R.; aj.: The Moral Machine experiment. *Nature*, ročník 563, č. 7729, 11 2018: s. 59–64, ISSN 0028-0836, doi:10.1038/s41586-018-0637-6.
URL <https://doi.org/10.1038/s41586-018-0637-6>
- [2] Behrmann, G.; David, A.; Larsen, K. G.: A Tutorial on Uppaal. In *Formal Methods for the Design of Real-Time Systems, Lecture Notes in Computer Science*, ročník 3185, editace M. Bernardo; F. Corradini, Springer Berlin Heidelberg, 2004, ISBN 978-3-540-23068-7, s. 200–236, doi:10.1007/978-3-540-30080-9_7.
URL http://dx.doi.org/10.1007/978-3-540-30080-9_7
- [3] David, A.; Fang, H.; Larsen, K. G.; aj.: Verification and Performance Evaluation of Timed Game Strategies. In *Formal Modeling and Analysis of Timed Systems*, editace A. Legay; M. Bozga, Cham: Springer International Publishing, 2014, ISBN 978-3-319-10512-3, s. 100–114.
- [4] David, A.; Jensen, P. G.; Larsen, K. G.; aj.: Uppaal Stratego. In *Tools and Algorithms for the Construction and Analysis of Systems*, editace C. Baier; C. Tinelli, Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, ISBN 978-3-662-46681-0, s. 206–211.
- [5] David, A.; Larsen, K. G.; Legay, A.; aj.: *Uppaal SMC tutorial*. *International Journal on Software Tools for Technology Transfer*, ročník 17, 8 2015: s. 397–415, doi:10.1007/s10009-014-0361-y.
- [6] Geiger, A.; Lenz, P.; Urtasun, R.: Are we ready for autonomous driving? The KITTI vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, June 2012, ISSN 1063-6919, s. 3354–3361, doi:10.1109/CVPR.2012.6248074.
- [7] Gert Rudolph, U. V.: Three Sensor Types Drive Autonomous Vehicles. [Online; navštíveno 22.11.2018].
URL <https://www.sensorsmag.com/components/three-sensor-types-drive-autonomous-vehicles>
- [8] Gupta, N.; Vijay, R.; Korupolu, P. V. N.; aj.: Architecture of Autonomous Vehicle Simulation and Control Framework. In *Proceedings of the 2015 Conference on Advances In Robotics, AIR '15*, New York, NY, USA: ACM, 2015, ISBN 978-1-4503-3356-6, s. 65:1–65:6, doi:10.1145/2783449.2783515.
URL <http://doi.acm.org/10.1145/2783449.2783515>
- [9] Mazal, Z.: FIPA abstraktní architektura, nástroj JADE. [Online; navštíveno 10.12.2018].

- URL https://wis.fit.vutbr.cz/FIT/st/cfs.php?file=%2Fcourse%2FAGS-IT%2Flectures%2FAGS17_FIPA.pdf
- [10] Menze, M.; Geiger, A.: Object scene flow for autonomous vehicles. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, ISSN 1063-6919, s. 3061–3070, doi:10.1109/CVPR.2015.7298925.
- [11] Mikucionis, M.: UPPAAL Stratego - Introduction. [Online; navštíveno 14.01.2019]. URL <http://people.cs.aau.dk/~maris/stratego/intro.html>
- [12] Naranjo, J. E.; Gonzalez, C.; Garcia, R.; aj.: Lane-change fuzzy control in autonomous vehicles for the overtaking maneuver. *IEEE Transactions on Intelligent Transportation Systems*, ročník 9, č. 3, 2008: str. 438.
- [13] Statistics of Road Traffic Accidents in Europe and North America. [Online; navštíveno 02.01.2019]. URL <https://www.unece.org/fileadmin/DAM/trans/main/wp6/publications/RAS-2017.pdf>
- [14] Palmisano, I.: Jade. [Online; navštíveno 10.12.2018]. URL <http://jade.tilab.com/>
- [15] Paromtchik, I. E.; Laugier, C.: Motion generation and control for parking an autonomous vehicle. In *Proceedings of IEEE International Conference on Robotics and Automation*, ročník 4, April 1996, ISSN 1050-4729, s. 3117–3122 vol.4, doi:10.1109/ROBOT.1996.509186.
- [16] Peasley, E.: An Introduction to Using Simulink. [Online; navštíveno 20.12.2018]. URL http://www2.eng.ox.ac.uk/~labejp/Seminar/Simulink/Simulink_Introduction.pdf
- [17] Peringer, P.: SIMLIB Home Page. [Online; navštíveno 21.12.2018]. URL <https://www.fit.vutbr.cz/~peringer/SIMLIB/>
- [18] Peringer, P.: Simulační nástroje a techniky. [Online; navštíveno 21.12.2018]. URL <http://www.fit.vutbr.cz/study/courses/SNT/public/Prednasky/SNT.pdf>
- [19] Peringer, P.: *Modelování a simulace. Studijní opora*. VUT, 2012.
- [20] Pěnička, R.: *PĚT STUPŇŮ K AUTONOMNÍMU ŘÍZENÍ*. [Online; navštíveno 20.11.2018]. URL <https://www.skoda-storyboard.com/cs/inovace/pet-stupnu-k-autonomnimu-rizeni/>
- [21] Roboauto: Roboauto - First self-driving car in the Czech Republic. <https://www.roboauto.cz/>, [Online; navštíveno 05.12.2018].
- [22] Service, N. O.: LIDAR—Light Detection and Ranging—is a remote sensing method used to examine the surface of the Earth. [Online; navštíveno 14.01.2019]. URL <http://oceanservice.noaa.gov/facts/lidar.html>
- [23] Skog, I.; Schumacher, A.; Handel, P.: A Versatile PC-Based Platform For Inertial Navigation. In *Proceedings of the 7th Nordic Signal Processing Symposium - NORSIG 2006*, June 2006, s. 262–265, doi:10.1109/NORSIG.2006.275238.

- [24] Waytz, A.; Heafner, J.; Epley, N.: The mind in the machine: Anthropomorphism increases trust in an autonomous vehicle. *Journal of Experimental Social Psychology*, ročník 52, 2014: s. 113 – 117, ISSN 0022-1031, doi:<https://doi.org/10.1016/j.jesp.2014.01.005>.
URL <http://www.sciencedirect.com/science/article/pii/S0022103114000067>
- [25] Wikipedie: Radar — Wikipedie: Otevřená encyklopedie. [Online; navštíveno 22.11.2018].
URL <https://cs.wikipedia.org/wiki/Radar>
- [26] Yağdereli, E.; Gemci, C.; Aktaş, A. Z.: A study on cyber-security of autonomous and unmanned vehicles. *The Journal of Defense Modeling and Simulation*, ročník 12, č. 4, 2015: s. 369–381, doi:10.1177/1548512915575803, <https://doi.org/10.1177/1548512915575803>.
URL <https://doi.org/10.1177/1548512915575803>

Příloha A

Obsah příloženého CD

CD disk přiložený k práci obsahuje následující soubory:

- doc/ - uchovává všechny potřebné dokumenty k vytvoření této zprávy
- src/ - obsahuje zdrojový soubor modelu ve formátu .xml
- tool/ - obsahuje UPPAAL Stratego verze 4.1.20-5
- TechnickaZprava - soubor ve formátu .pdf obsahující tuto zprávu
- README - soubor ve formátu .txt s popisem obsahu příloženého CD