

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA STROJNÍ

ÚSTAV MECHANIKY, BIOMECHANIKY A MECHATRONIKY

Odbor Mechaniky a mechatroniky



Diplomová práce

System pro plánování složitějších parkovacích manévrů

Praha, 2018

Michal Sekerka

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Sekerka** Jméno: **Michal** Osobní číslo: **381274**
Fakulta/ústav: **Fakulta strojní**
Zadávající katedra/ústav: **Ústav mechaniky, biomechaniky a mechatroniky**
Studijní program: **Strojní inženýrství**
Studijní obor: **Mechatronika**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Systém pro plánování složitějších parkovacích manévřů

Název diplomové práce anglicky:

Driver assistance system for planning more complex parking maneuvers

Pokyny pro vypracování:

- 1) Seznamte se možnostmi existujících parkovacích asistentů.
- 2) Sestavte simulační model vozidla.
- 3) Připravte nástroj pro zjištění, zda je vybrané parkovací místo dosažitelné.
- 4) Sestavte systém pro plánování trajektorie i v případě, že je parkovací místo dosažitelné pouze složitějším opakovaným manévrováním.
- 5) Ověřte navržené řešení na několika trajektoriích.

Seznam doporučené literatury:

- 1) Lavalley, S.M., Plannig Algorithms, Illinois, University of Illinois, 1999.
- 2) Vrbský, L., Experimentální model vozidla s asistentem pro couvání s přívěsem, bakalářská práce, Praha, ČVUT v Praze, 2016.
- 3) Lv, Z., Zhao, L., Liu, Z., A Path-Planning Algorithm for Automatic Parallel Parking, Proceedings of 2013 Third International Conference on Instrumentation, Measurement, Computer, Communication and Control, IMCCC 2013, 2013, pp. 474-478.
- 4) Wang, L., Guo, L., He, Y., Path Planning Algorithm for Automatic Parallel Parking from Arbitrary Initial Angle, 10th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, 2017, pp. 55-58.

Jméno a pracoviště vedoucí(ho) diplomové práce:

Ing. Petr Beneš, Ph.D., odbor mechaniky a mechatroniky FS

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

Datum zadání diplomové práce: **25.04.2018**

Termín odevzdání diplomové práce: **17.08.2018**

Platnost zadání diplomové práce: _____

Ing. Petr Beneš, Ph.D.
podpis vedoucí(ho) práce

prof. Ing. Milan Růžička, CSc.
podpis vedoucí(ho) ústavu/katedry

prof. Ing. Michael Valášek, DrSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

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

Datum převzetí zadání

Podpis studenta

Čestné prehlásenie

Prehlasujem, že som diplomovú prácu vypracoval samostatne, na základe vlastných poznatkov, pokynov vedúceho diplomovej práce a podkladov, uvedených v zozname na konci práce.

V Prahe dňa

.....

Michal Sekerka

Podakovanie

Chcel by som poďakovať všetkým, ktorí mi akýmkoľvek spôsobom pomohli pri spracovaní tejto diplomovej práce.

Moje poďakovanie patrí najmä vedúcemu práce, Ing. Petrovi Benešovi, Ph.D., za jeho odborné vedenie, metodickú pomoc a cenné rady, ktoré mi poskytol pri vypracovávaní diplomovej práce.

Anotačný list

- Meno autora:** Michal Sekerka
- Názov diplomovej práce:** Systém pro plánování složitějších parkovacích manévrů
- Anglický názov:** Driver assistance system for planning more complex parking maneuvers
- Akademický rok:** 2017/2018
- Obor štúdia:** Mechatronika
- Ústav/odbor:** Ústav mechaniky, biomechaniky a mechatroniky
Odbor Mechaniky a mechatroniky
- Vedúci diplomovej práce:** Ing. Petr Beneš, Ph.D.
- Bibliografické údaje:**
- | | |
|-----------------|--------|
| Počet strán: | 72 |
| Počet obrázkov: | 49 |
| Počet príloh: | 1 x CD |
- Kľúčové slova:** Parkovací asistent, plánovanie trajektórie, kolízia, optimalizácia, genetický algoritmus
- Keywords:** Parking Assistant, Trajectory Planning, Collision, Optimization, Genetic Algorithm
- Abstrakt:** Diplomová práca sa zaoberá plánovaním trajektórie s cieľom navrhnuť vlastný systém hľadania trajektórie parkovacieho manévru pre autonómne vozidlo. Najskôr je vytvorený kinematický a simulačný model vozidla, následne riešenie úlohy analyticky a numericky pomocou lokálnej a globálnej optimalizačnej metódy. Dôležitou súčasťou je vyhodnocovanie kolízie. Na záver je navrhnutý systém podrobený simulačnému experimentu pre rôzne zadané úlohy.
- Abstract:** The diploma thesis deals with the planning a trajectory in order to design its own system of searching for the trajectory of the parking maneuver for an autonomous vehicle. At first, a kinematic and simulation model of the vehicle is created. Consequently, the task is solved analytically and numerically using the local and global optimization method. An important part is collision assessment. Finally, the system is designed for a simulation experiment for various tasks.

Obsah

Úvod	10
1 Súčasný stav skúmanej problematiky	11
1.1 Parkovacie systémy	11
1.1.1 Parkovacie senzory	11
1.1.2 Reverzné a 360° kamery	12
1.1.3 Parkovací asistent	12
1.1.4 Autonómne parkovanie	13
1.1.5 Autonómia v budúcnosti	13
1.2 Plánovanie trajektórie	14
2 Model vozidla	15
2.1 Kinematický model	15
2.2 Simulačný model	17
2.3 Grafický model	19
3 Trajektória – analytický výpočet	21
3.1 Základný návrh CSC trajektórie	21
3.1.1 Algoritmus výpočtu vnútorných tangenciálnych čiar medzi dvomi kružnicami	22
3.1.2 Výpočet dĺžky trajektórie	25
3.1.3 Simulácia kinematického modelu	25
3.1.4 Zhodnotenie základného návrhu CSC trajektórie	27
3.2 Plánovanie trajektórie pre prípad s nenulovými natočeniami vozidla	27
3.2.1 Algoritmus	28
3.2.2 Simulácia kinematického modelu	35
3.2.3 Zhodnotenie plánovania trajektórie pre prípad s nenulovými natočeniami vozidla	37
4 Kolízia – prostredie s prekážkami	38
4.1 Grafické vyhodnotenie kolízie	38
4.2 Algoritmus hľadania trajektórie s vyhodnotením kolízie	39
4.2.1 Simulácia kinematického modelu	40
4.3 Zhodnotenie kolízie – prostredie s prekážkami	41
5 Trajektória – numerický výpočet	42
5.1 Lokálna optimalizačná metóda	42
5.1.1 Formulácia úlohy	42
5.1.2 Popis trajektórie	43
5.1.3 Cieľová funkcia	44

5.1.4	Optimalizačná metóda – FminCon	45
5.1.5	Simulácia kinematického modelu	46
5.1.6	Zhodnotenie lokálnej optimalizačnej metódy.....	49
5.2	Globálna optimalizačná metóda	50
5.2.1	Optimalizačné parametre	50
5.2.2	Cieľová funkcia	50
5.2.3	Optimalizačná metóda – genetický algoritmus.....	51
5.2.4	Simulácia kinematického modelu	55
5.2.5	Zhodnotenie globálnej optimalizačnej metódy	56
6	Simulačný experiment.....	57
6.1	Pozdĺžne parkovanie	57
6.1.1	Úloha č.1: Vozidlo stojacie za parkovacím miestom	57
6.1.2	Úloha č.2: Vozidlo stojacie pred parkovacím miestom	59
6.1.3	Úloha č.3: Menšie parkovacie miesto	61
6.2	Kolmé parkovanie	63
6.2.1	Úloha č.4: Vozidlo stojacie za parkovacím miestom	63
6.2.2	Úloha č.5: Vozidlo stojacie pred parkovacím miestom	65
6.2.3	Úloha č.6: Vozidlo parkujúce popredu.....	66
6.3	Zhodnotenie simulačného experimentu	67
	Záver.....	68
	Zoznam použitých zdrojov	70
	Prílohy	72

Zoznam obrázkov

Obr. 2.1 Kinematický model vozidla	15
Obr. 2.2 Vektor rýchlosti	16
Obr. 2.3 Natočenie o uhol psi	16
Obr. 2.4 Simulačný model	17
Obr. 2.5 Kinematický model v prostredí MATLAB Simulink	18
Obr. 2.6 Subsystem: Uhol natočenia kolies	18
Obr. 2.7 Skoková zmena uhla natočenia kolies.....	19
Obr. 2.8 Grafický model vozidla v programe MATLAB.....	19
Obr. 2.9 Základné rozmery vozidla Škoda Yeti [10]	20
Obr. 3.1 CSC trajektória v priestore bez prekážok	21
Obr. 3.2 Vnútorne tangenciálne čiary medzi dvoma kružnicami	22
Obr. 3.3 Zobrazenie výsledku algoritmu v prostredí programu MATLAB	24
Obr. 3.4 Zobrazenie výslednej trajektórie CSC v prostredí programu MATLAB	24
Obr. 3.5 Výpočet uhlu alfa	25
Obr. 3.6 Obálka vozidla opisujúceho trajektóriu CSC	26
Obr. 3.7 Hľadanie trajektórie pre všeobecný prípad	27
Obr. 3.8 Základná schéma algoritmu pre nájdenie možných trajektórií	28
Obr. 3.9 Orientácia polomeru pri pohybe vľavo a vpravo	29
Obr. 3.10 Určenie smeru pohybu.....	34
Obr. 3.11 Obálka vozidla opisujúceho trajektóriu pre rôzne prípady	36
Obr. 4.1 Uvažované typy prostredia	38
Obr. 4.2 Príklad mapy bez kolízie pri veľkosti elementu 1 cm	39
Obr. 4.3 Príklad mapy s kolíziou pri veľkosti elementu 1 cm.....	39
Obr. 4.4 Pseudokód cyklu vyhodnotenia kolízie	40
Obr. 4.5 Obálka vozidla opisujúceho trajektóriu bez kolízie s prostredím	41
Obr. 5.1 Zadanie optimalizačnej úlohy pre pozdĺžne parkovanie	42
Obr. 5.2 Zadanie optimalizačnej úlohy pre kolmé parkovanie	43
Obr. 5.3 Popis oblúka trajektórie	44
Obr. 5.4 Zápis funkcie FminCon v prostredí programu MATLAB	45
Obr. 5.5 Ukážka rôznych smerov pohybu	47
Obr. 5.6 Funkcia signum.....	47
Obr. 5.7 Mapa prostredia a obálka vozidla opisujúceho trajektóriu, lokálna metóda	49
Obr. 5.8 Diagram použitého Genetického Algoritmu	52

Obr. 5.9 GA - Turnajový výber rodičov.....	54
Obr. 5.10 GA – Viacbodové kríženie	54
Obr. 5.11 Mapa prostredia a obálka vozidla opisujúceho trajektóriu, lokálna metóda	56
Obr. 6.1 Úloha č.1: Vozidlo stojacie za parkovacím miestom.....	57
Obr. 6.2 Riešenie úlohy č.1: Obálka vozidla opisujúceho trajektórie	58
Obr. 6.3 Úloha č.2: Vozidlo stojacie pred parkovacím miestom.....	59
Obr. 6.4 Riešenie úlohy č.2: Obálka vozidla opisujúceho trajektórie	60
Obr. 6.5 Úloha č.3 Menšie parkovacie miesto	61
Obr. 6.6 Riešenie úlohy č.2: Obálka vozidla opisujúceho trajektóriu	61
Obr. 6.7 Detail koncovej fázy manévru riešiacej úlohu č. 3	62
Obr. 6.8 Úloha č.4: Vozidlo stojacie za parkovacím miestom.....	63
Obr. 6.9 Riešenie úlohy č.4: Obálka vozidla opisujúceho trajektórie	64
Obr. 6.10 Úloha č.5: Vozidlo stojacie za parkovacím miestom.....	65
Obr. 6.11 Riešenie úlohy č.5: Obálka vozidla opisujúceho trajektóriu	65
Obr. 6.12 Úloha č.6: Vozidlo parkujúce popredu.....	66
Obr. 6.13 Riešenie úlohy č.6: Obálka vozidla opisujúceho trajektóriu	67

Zoznam tabuliek

Tab. 2.1 Základné rozmery vozidla Škoda Yeti.....	20
Tab. 3.1 Možné prípady polohy dvoch kružníc v priestore	31
Tab. 4.1 Možné hodnoty elementu rastru	38
Tab. 4.2 Počiatočné polohy a parametre prostredia	40
Tab. 5.1 Počiatočné polohy a parametre prostredia	48
Tab. 5.2 Výsledné optimalizačné parametre lokálnej metódy	48
Tab. 5.3 Výsledné optimalizačné parametre globálnej metódy	55
Tab. 6.1 Riešenie úlohy č.1: Výsledné parametre trajektórií.....	58
Tab. 6.2 Riešenie úlohy č.2: Výsledné parametre trajektórií.....	60
Tab. 6.3 Riešenie úlohy č.3: Výsledné parametre trajektórie	62
Tab. 6.4 Riešenie úlohy č.4: Výsledné parametre trajektórií.....	64
Tab. 6.5 Riešenie úlohy č.5: Výsledné parametre trajektórie	66
Tab. 6.6 Riešenie úlohy č.6: Výsledné parametre trajektórie	67

Úvod

V súčasnosti je veľký záujem o služby zjednodušujúce život a vývoj autonómnych zariadení sa neustále posúva vpred. Bežný človek sa veľmi často do kontaktu s týmito technológiami nedostáva, výnimkou je automobilový priemysel. Výrobcovia automobilov ponúkajú ako doplnkové služby vo vozidle rôzne asistenčné služby. Jednou z nich je parkovací asistent. Keďže po cestách chodí čím ďalej tým viac vozidiel, zvyšuje sa aj náročnosť pri nájdení vhodného parkovacieho miesta s dostatkom priestoru. Množstvo vodičov čelí ťažkostiam pri parkovacích manévroch, a preto čoraz viac automobiliek zaraďuje do svojej ponuky pomocnú ruku v podobe samostatne parkujúceho vozidla.

Dnešná doba, bohatá na pokročilé technológie, prináša vyspelé možnosti autonómneho parkovania. Všetky sú však pred verejnou starostlivo chránené, ale vďaka tejto diplomovej práci môžeme nazrieť do problematiky navrhovania trajektórií parkovacích manévrov. Aj napriek existujúcim riešeniam je práca aktuálna pre potrebu návrhu systému, ktorý je súčasťou prebiehajúceho výskumu, kde bude prakticky využitý.

Jednotlivé kapitoly popisujú navrhované metódy hľadania trajektórie vozidla. Vyvinutý systém je podrobený testu funkčnosti v simulačnom experimente, ktorého priebeh a výsledky opisuje posledná kapitola. Pre návrh, ladenie aj testovanie je použité výpočtové prostredie programu MATLAB a Simulink od spoločnosti The MathWorks, Inc.

Cieľ práce

Hlavným cieľom tejto práce je navrhnuť vlastný systém hľadania trajektórie parkovacieho manévru pre autonómne vozidlo v rámci prebiehajúceho výskumného projektu na Ústave mechaniky, biomechaniky a mechatroniky na Fakulte strojní ČVUT v Praze.

Čiastkové ciele vedúce k naplneniu hlavného cieľa:

1. Zoznámiť sa s možnosťami existujúcich parkovacích asistentov.
2. Zostaviť simulačný model vozidla.
3. Pripraviť nástroj pre zistenie, či je vybrané parkovacie miesto dosiahnuteľné.
4. Zostaviť systém pre plánovanie trajektórie i v prípade, že je parkovacie miesto dosiahnuteľné iba zložitejším opakovaným manévrom.
5. Overiť navrhnuté riešenie na niekoľkých trajektóriách.

1 Súčasný stav skúmanej problematiky

O problematike skúmanej v tejto práci sa môžeme dočítať v množstve internetových publikácií, na stránkach predajcov automobilov, v knihách a v odborných článkoch. Problémom je, že väčšinou ide len o informatívny charakter publikácií a detailné informácie ostávajú zamlčané. Z tohto dôvodu sa všetky dostupné zdroje dajú využiť len ako zdroj inšpirácie pre vlastný výskum, ako to bolo aj pri tvorbe tejto práce.

1.1 Parkovacie systémy

Na trhu je viacero systémov od rôznych výrobcov. Ich stručný vývoj a niektoré ich vlastnosti si v nasledujúcom texte priblížime.

1.1.1 Parkovacie senzory

Môže to znieť ako novodobý vynález, ale senzory sa začali objavovať už v sedemdesiatych rokoch dvadsiateho storočia. Ich podoba však bola iná, ako ju poznáme dnes. Pôvodne boli určené na použitie v usmerňovacích zariadeniach pre nevidiacich. Až začiatkom nového tisícročia sa začala táto technológia výraznejšie rozvíjať. Prvým bežným vozidlom, kde sa objavila, bola Toyota Prius v roku 2003 [1].

Parkovacie senzory využívajú technológiu ultrazvuku a sú inštalované do nárazníkov automobilu. Vnímajú okolité prostredie a dokážu merať vzdialenosť medzi vozidlom a prekážkami okolo neho.

Vodič automobilu je upozorňovaný pípajúcim zvukom, ktorý sa zrýchľuje, čím je vozidlo bližšie k objektu. V dnešnej dobe je zvuk sprevádzaný digitálnym obrazom na palubnej doske, ktorý umožňuje presnejšie vyhodnocovanie vzdialeností [1].

Senzory sú najzákladnejším a najbežnejším parkovacím systémom.

Volkswagen Park Pilot

Funkcia Volkswagen Park Pilot (nazývaná aj Park Distance Control) používa viacero vizuálnych a zvukových upozornení, aby informovala vodiča, keď sa vozidlo priblíži do tesnej blízkosti iného objektu. Je to veľmi užitočná funkcia pre vodičov pokúšajúcich sa posúdiť vzdialenosť medzi sebou a inými objektami počas procesu parkovania [2].

1.1.2 Reverzné a 360° kamery

Reverzné parkovacie kamery sa prvýkrát objavili v koncepte automobilu Buick Centurion v roku 1956. Technológia sa však k bežným automobily nedosala až do roku 2000, kedy Nissan predstavil svoj luxusný Infiniti Q45 [1].

Kamery boli neskôr kombinované s parkovacími senzormi pre jasnejší a presnejší popis toho, čo sa deje okolo vozidla. Dnes už dokonca displej ukazuje, kade vozidlo pôjde pri aktuálnom natočení kolies, a tak pomáha pri parkovaní.

Najmodernejšie modely majú viacero kamier umiestených okolo vozidla, čím vytvárajú 360° obraz okolia z vtáčej perspektívy. Niektoré automobily dokonca umožňujú prepínanie na jednotlivé kamery pre detailné sledovanie vybraného miesta [1].

Existujú aj iné druhy senzorov (radar, Lidar a.i.), ktoré nie sú tak veľmi rozšírené a stále sa pracuje na zdokonaľovaní ich aplikácie.

1.1.3 Parkovací asistent

Parkovací asistent je nadstavbový systém, integrovaný do automobilu pre zjednodušenie pri parkovacích manévroch. Primárnou funkciou je prevziať riadenie vozidla a bezpečne zaparkovať na vybrané parkovacie miesto. Táto funkcia je rozšírená o možnosť vyhodnocovania vhodnosti parkovacieho miesta pomocou bočných ultrazvukových senzorov.

Prvá verzia parkovacieho asistenta sa objavila už v roku 2003. Spočiatku bola táto funkcia dosť obmedzená. Snažila sa odhaliť objekty, ktoré neboli automobily alebo neboli dostatočne veľké. Napríklad psy, chodci alebo detské kočíky bolo v okolí ťažké zistiť. V priebehu rokov sa systém stal múdrejším pomocou kamier, parkovacích senzorov a vylepšení softvéru. Automobily sa postupne stali schopnými rozpoznať aj dopravné značky [1].

Dnes parkovací asistent dokáže informovať vodiča, keď nájde vhodné parkovacie miesto. Základné verzie systému, ako napríklad Ford's Active Park Assist alebo Volkswagen Park Assist, dokážu samy točiť volantom, ale vodičovi ostáva kontrola nad pedálmi. Odkedy je možné presnejšie odhaľovanie prekážok, vozidlo môže samovoľne zastaviť, aby sa vyhlo bezprostrednej kolízii.

Volkswagen Park Assist

Volkswagen Park Assist je často označovaný ako Automatický asistent parkovania (z angl. Automatic Parking Steering Assistant). Táto funkcia je pokročilejšia ako funkcia Volkswagen Park Pilot. Funkcia Park Assist umožňuje vozidlu prevziať riadiace funkcie počas kolmého a pozdĺžneho parkovania. Aby to bolo možné vykonať, systém využíva kamery a senzory namontované po obvode vozidla k určeniu, či parkovacie miesto je dostatočne veľké, a následne k navigácii vozidla na požadované miesto. Vodič má po celý čas pod kontrolou plynový a brzdný pedál. Pokiaľ by si vodič prial manéver ukončiť, stačí, ak chytí do rúk volant a systém sa automaticky deaktivuje. [2]

Valeo Park4U®

Tento systém sa začal používať v roku 2007 vo vozidle Volkswagen Touran (Volkswagen Park Assist). Funkcia Park4U® môže taktiež pomôcť vodičovi vyjsť z parkovacieho miesta. Systém si zmeria priestor pred a za vozidlom a zvolí najlepšiu stratégiu pre opustenie priestoru. Zatiaľ čo vodič riadi rýchlosť vozidla, systém prevezme riadenie tak, ako to bolo pri zaparkovaní. Systém zisťuje vhodný okamih na opustenie miesta a automaticky sa deaktivuje, aby umožnil vodičovi zaradiť sa do premávky [3].

1.1.4 Autonómne parkovanie

Automobily, ako napríklad BMW i3 (vyrobené v roku 2015), sú schopné plne autonómne zaparkovať. Tento úkon zahŕňa aj ovládanie riadiacej páky. Funkcia parkovací asistent na konci manévru dokáže zatiahnuť ručnú brzdú (vyžaduje elektrickú ručnú brzdú), aby zabezpečila vozidlo pred pohnutím [1].

Valeo Park4U® Remote

Nová generácia automatického parkovacieho systému Park4U® Remote (premiéra v roku 2011) je plne automatické riešenie, ktoré dokonca umožňuje vodičovi opustiť vozidlo pred začatím parkovacieho manévru, pričom má kontrolu nad vozidlom cez mobilný telefón. [3]

1.1.5 Autonómia v budúcnosti

Aj napriek samostatne parkujúcim automobily v mnohých prípadoch je nutná prítomnosť vodiča. Niektorí výrobcovia, ako napríklad Ford a Volvo, oznámili svoje plány mať plne autonómne vozidlo na ceste už v roku 2021. Tesla už uviedla vylepšenú verziu autopilota, ktorý dokáže riadiť aj parkovať vozidlo úplne bez zásahu vodiča. Vodiči budú môcť privolať svoje vozidlo, ktoré sa

bude riadiť samo, na miesto, kde ich má vyzdvihnúť. Základnú verziu tejto funkcie ponúka Tesla a BMW. Vozidlo dokáže pri pomalých rýchlostiach ísť vpred a vzad, čo je praktické pri parkovaní do úzkej garáže [1].

1.2 Plánovanie trajektórie

Väčšina zdrojov zaoberajúcich sa problematikou plánovania trajektórie zanedbáva dynamické účinky a používa len kinematický model vozidla. Pre výpočty a simulácie je možné použiť zjednodušený jednostopový model vozidla, pri ktorom sa zanedbáva rozdielna rýchlosť a uhol natočenia kolies presnej nápravy [4]. Pre použitie pre reálny systém je následne jednoduché z takéhoto modelu dopočítať rýchlosti a natočenia jednotlivých kolies.

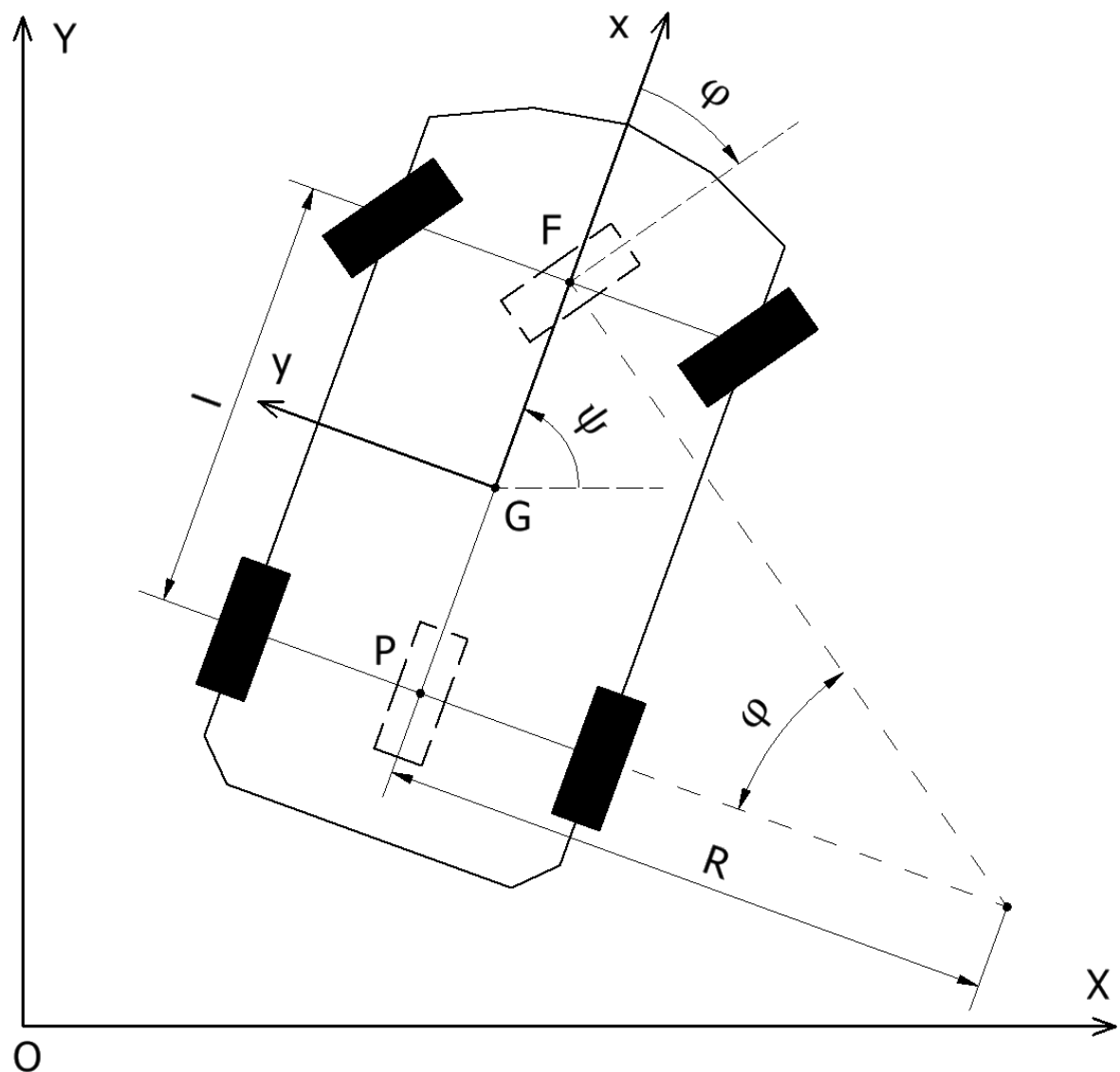
K hľadaniu trajektórie je možné pristupovať analyticky [4] [5], kedy sa exaktne odvodí rovnice popisujúce trajektóriu. Pre zjednodušenie výpočtov je možné uvažovať trajektóriu pozostávajúcu iba z kružníc a čiar, ktorá, ako je dokázané [6], je najkratšou dosiahnuteľnou možnosťou. V praxi to zodpovedá manévru, kedy vozidlo zastaví pred každou zmenou uhlu natočenia kolies. O niečo zložitejšou možnosťou je popis trajektórie pomocou kontinuálne zakrivených čiar (z angl. Continuous-Curvature Path) [7]. Tento popis odpovedá pohybu vozidla, kedy plynule točí volantom bez predchádzajúceho zastavenia. Je možné použiť aj zložitejšie geometrické popisy trajektórie, ako napríklad Bézierove krivky a iné.

Pre zložitejšie manévry a robustnejšie systémy sa používajú zložitejšie algoritmy. Zväčša ide o globálne metódy hľadajúce extrém zvolenej funkcie. Pre rôzne platformy nachádzajú uplatnenie rôzne druhy evolučných algoritmov (genetické algoritmy [8], Monte-Carlo alebo optimalizácia mravčou kolóniou). Prípadne je možné využiť sofistikovanejšie a náročnejšie techniky fuzzy neurónových sietí a umelej inteligencie [9].

2 Model vozidla

2.1 Kinematický model

Pre zjednodušenie výpočtu uhlu natočenia kolies budeme uvažovať dvojkolesový model automobilu. Tento jednostopový model vznikne pridaním virtuálneho kolesa do prostriedku prednej a zadnej nápravy. Pre samotný výpočet nie je dôležité poznať uhly každého kolesa a je možné ich spätne dopočítať zo známeho rozchodu kolies.



Obr. 2.1 Kinematický model vozidla

Ďalej budeme uvažovať pomalý pohyb vozidla pri parkovacom manévri, kedy nedochádza ku preklzu kolies. Kinematický model zobrazený na obrázku vyššie (Obr. 2.1) je popísaný súradnicami $X(t)$ a $Y(t)$ v globálnom systéme súradníc a uhlom $\psi(t)$ vyjadrujúcim natočenie lokálneho

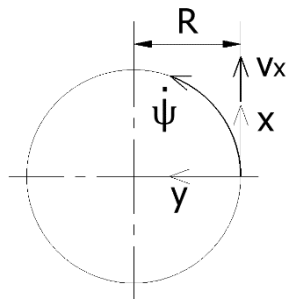
systému súradníc voči globálnemu. Model sa vzťahuje k jeho geometrickému stredú, ktorý reprezentuje bod G , nachádzajúci sa v strede spojnice náprav. Ovládacími prvkami sú rýchlosť pohybu v , ktorú budeme uvažovať konštantnú, a natočenie predných kolies $\varphi(t)$. Z obrázku vyššie (Obr. 2.1) môžeme jednoduchým spôsobom určiť rádus oblúka v závislosti na uhle $\varphi(t)$:

$$R(\varphi) = \frac{l}{\tan \varphi} \quad (2.1)$$

Vzhľadom k malým posuvom budeme pohyb uvažovať iba v tangenciálnom smere.

$$v^{xy} = \begin{bmatrix} v_x \\ 0 \end{bmatrix} \quad (2.2)$$

Potom uhlovú rýchlosť $\dot{\psi}$ dostaneme v tvare:

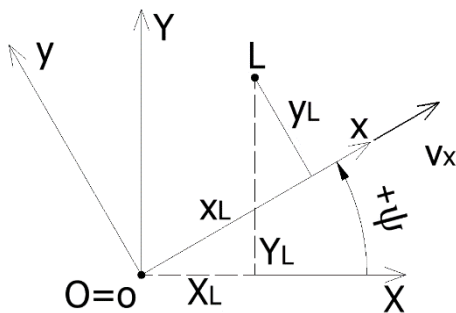


$$v_x = R \cdot \omega = R \cdot \dot{\psi} \quad (2.3)$$

$$\rightarrow \dot{\psi} = \frac{v_x}{l} \tan \varphi \quad (2.4)$$

Obr. 2.2 Vektor rýchlosti

Z teórie kinematiky pre rotačný pohyb platí:



$$X_L = x_L \cdot \cos \psi - y_L \cdot \sin \psi \quad (2.5)$$

$$Y_L = x_L \cdot \sin \psi + y_L \cdot \cos \psi \quad (2.6)$$

$$\begin{bmatrix} X_L \\ Y_L \end{bmatrix} = \begin{bmatrix} \cos \psi & -\sin \psi \\ \sin \psi & \cos \psi \end{bmatrix} \cdot \begin{bmatrix} x_L \\ y_L \end{bmatrix} \quad (2.7)$$

Obr. 2.3 Natočenie o uhol psi

Odvedenú transformačnú maticu použijeme na výpočet zložiek rýchlosti modelu:

$$\begin{bmatrix} v_X \\ v_Y \end{bmatrix} = \begin{bmatrix} \cos \psi & -\sin \psi \\ \sin \psi & \cos \psi \end{bmatrix} \cdot \begin{bmatrix} v_x \\ 0 \end{bmatrix} \quad (2.8)$$

Pre odvedenie rovníc rýchlosti pre bod G uvažujeme obecný rovinný pohyb. Výsledná rýchlosť je rovná súčtu unášanej a relatívnej rýchlosti vozidla.

$$\mathbf{v}_G^{xy} = \mathbf{v}_u^{xy} + \mathbf{v}_{rel}^{xy} = \mathbf{v}_o^{xy} + \boldsymbol{\omega}^{xy} \times \mathbf{r}_G^{xy} = \begin{bmatrix} v_x \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ l \\ 2 \end{bmatrix} \dot{\psi} = \begin{bmatrix} v_x \\ \frac{l}{2} \dot{\psi} \end{bmatrix} \quad (2.9)$$

Transformáciu do globálneho systému súradníc zapíšeme v maticovom tvare:

$$\begin{bmatrix} v_{GX} \\ v_{GY} \end{bmatrix} = \begin{bmatrix} \cos \psi & -\sin \psi \\ \sin \psi & \cos \psi \end{bmatrix} \cdot \begin{bmatrix} v_x \\ \frac{l}{2} \dot{\psi} \end{bmatrix} \quad (2.10)$$

Konečný tvar rovníc kinematického modelu pre stred vozidla dostaneme v nasledovnom tvare:

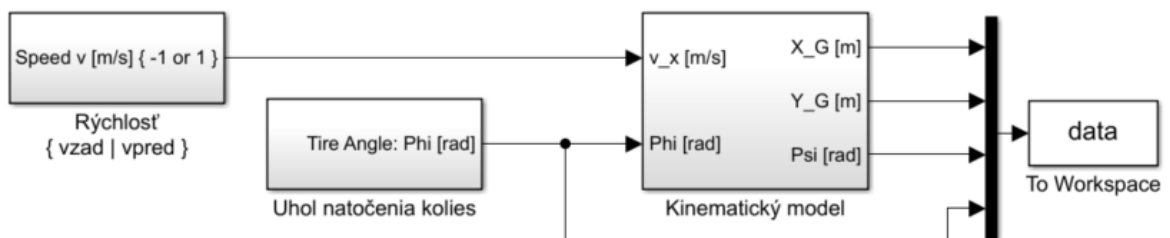
$$\dot{X} = v_x \cdot \cos \psi - \frac{l}{2} \dot{\psi} \cdot \sin \psi \quad (2.11)$$

$$\dot{Y} = v_x \cdot \sin \psi + \frac{l}{2} \dot{\psi} \cdot \cos \psi \quad (2.12)$$

$$\dot{\psi} = \frac{v_x}{l} \tan \varphi \quad (2.13)$$

2.2 Simulačný model

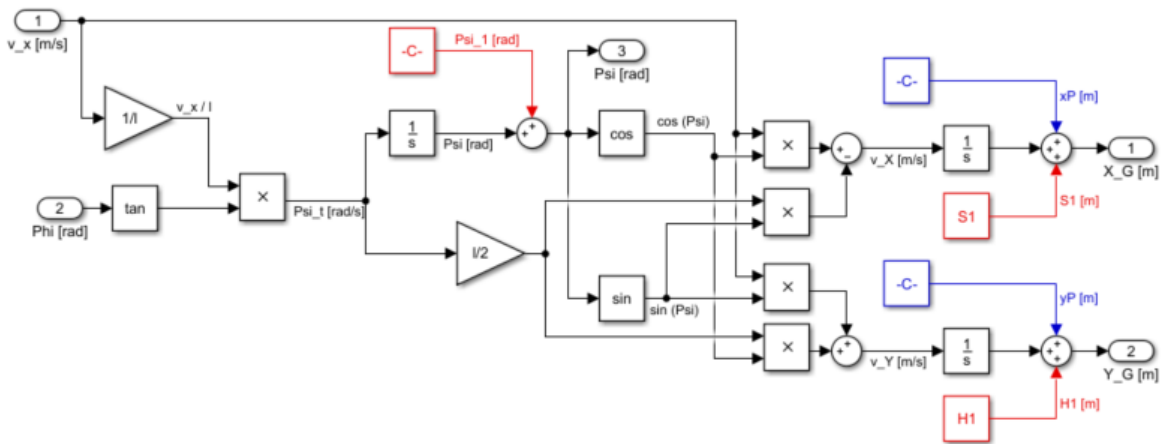
Na výpočet rovníc kinematického modelu bolo použité prostredie MATLAB Simulink. Celý simulačný model (Obr. 2.4) je zložený z troch subsystémov, z čoho prvé dva tvoria vstupy pre tretí a hlavný subsystém Kinematický model. Úlohou subsystémov Rýchlosť a Uhol natočenia kolies je na základe trajektórie riadiť vstupy rýchlosť $v_x(t)$ [m/s] a uhol $\varphi(t)$ [rad] v závislosti na čase. Výstupy sa uložia do premennej „data“ a budú slúžiť na nasledovné vykreslenie priebehu pohybu v animácii.



Obr. 2.4 Simulačný model

Hlavná časť modelu slúži na výpočet rovníc kinematického modelu vozidla. Je to systém s dvomi vstupmi a tromi výstupmi. Výstupmi je poloha stredy vozidla v globálnom systéme súradníc

vyjadrená súradnicami X_G a Y_G a natočením voči tomuto systému ψ . Jeho obsah je znázornený na obrázku nižšie (Obr. 2.5).

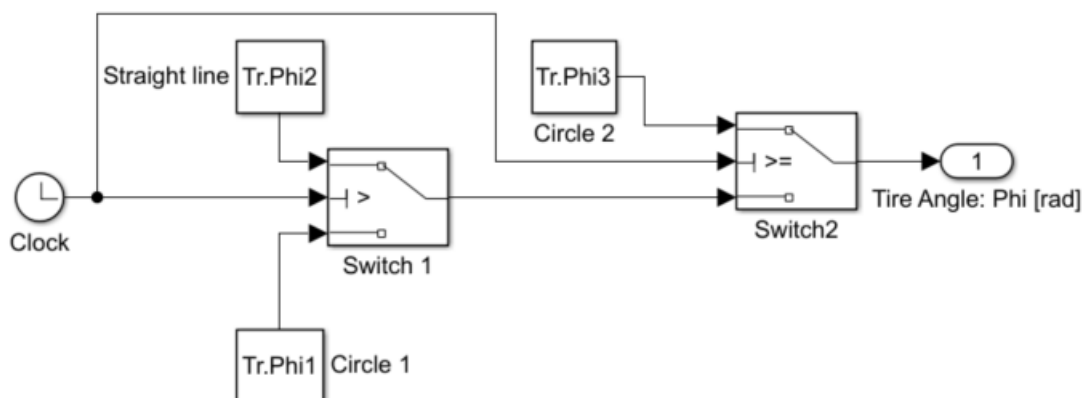


Obr. 2.5 Kinematický model v prostredí MATLAB Simulink

Čierna časť reprezentuje rovnice (2.11),(2.12)a(2.13). Bloky zvýraznené červenou farbou posúvajú model do počiatkovej polohy $[S_1, H_1]$ a natáčajú na počiatkový uhol ψ_1 . Kinematický model sa vťahuje k stredy vozidla, a preto modré bloky posúvajú bod opisujúci trajektóriu do bodu P , ktorý reprezentuje stred zadnej nápravy vozidla. Môžeme ich zapísať v maticovom tvare.

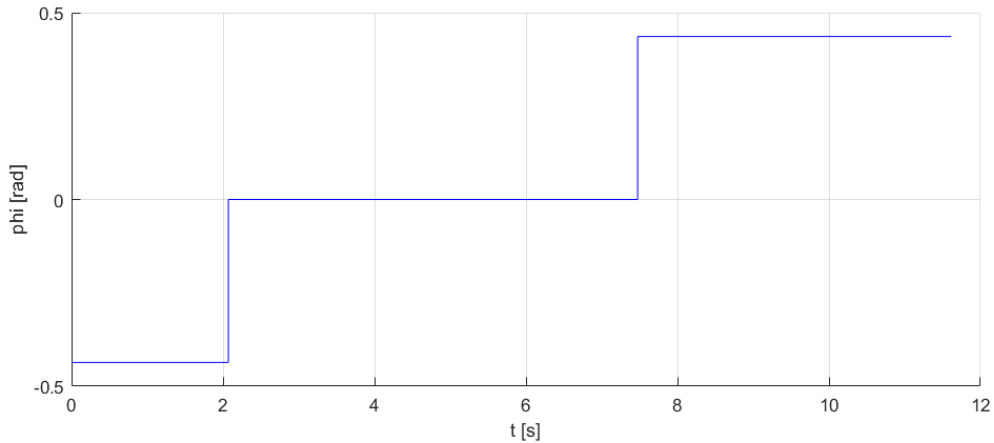
$$\begin{bmatrix} X_P \\ Y_P \end{bmatrix}^{XY} = \begin{bmatrix} X_G \\ Y_G \end{bmatrix}^{XY} + \begin{bmatrix} \cos \psi & -\sin \psi \\ \sin \psi & \cos \psi \end{bmatrix} \cdot \begin{bmatrix} x_P \\ y_P \end{bmatrix}^{xy} \quad (2.14)$$

Subsystem Uhol natočenia kolies (Obr. 2.6) skokovo mení uhol natočenia kolies pomocou dvoch prepínačov (Switch) na základe vypočítaného času jednotlivých častí trajektórie. Ich výpočet je podrobne rozobratý v odseku 3.1.3.



Obr. 2.6 Subsystem: Uhol natočenia kolies

Uvažujeme skokovú zmenu uhla natočenia kolies. V realite by to zodpovedalo zastaveniu a následnému natočeniu kolies do požadovanej polohy. Na obrázku nižšie (Obr. 2.7) je znázornený priebeh uhla φ pre trajektóriu z odstavca 3.1.3 (Obr. 3.6).

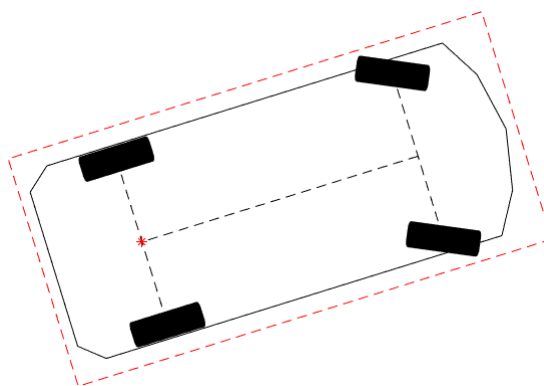


Obr. 2.7 Skoková zmena uhla natočenia kolies

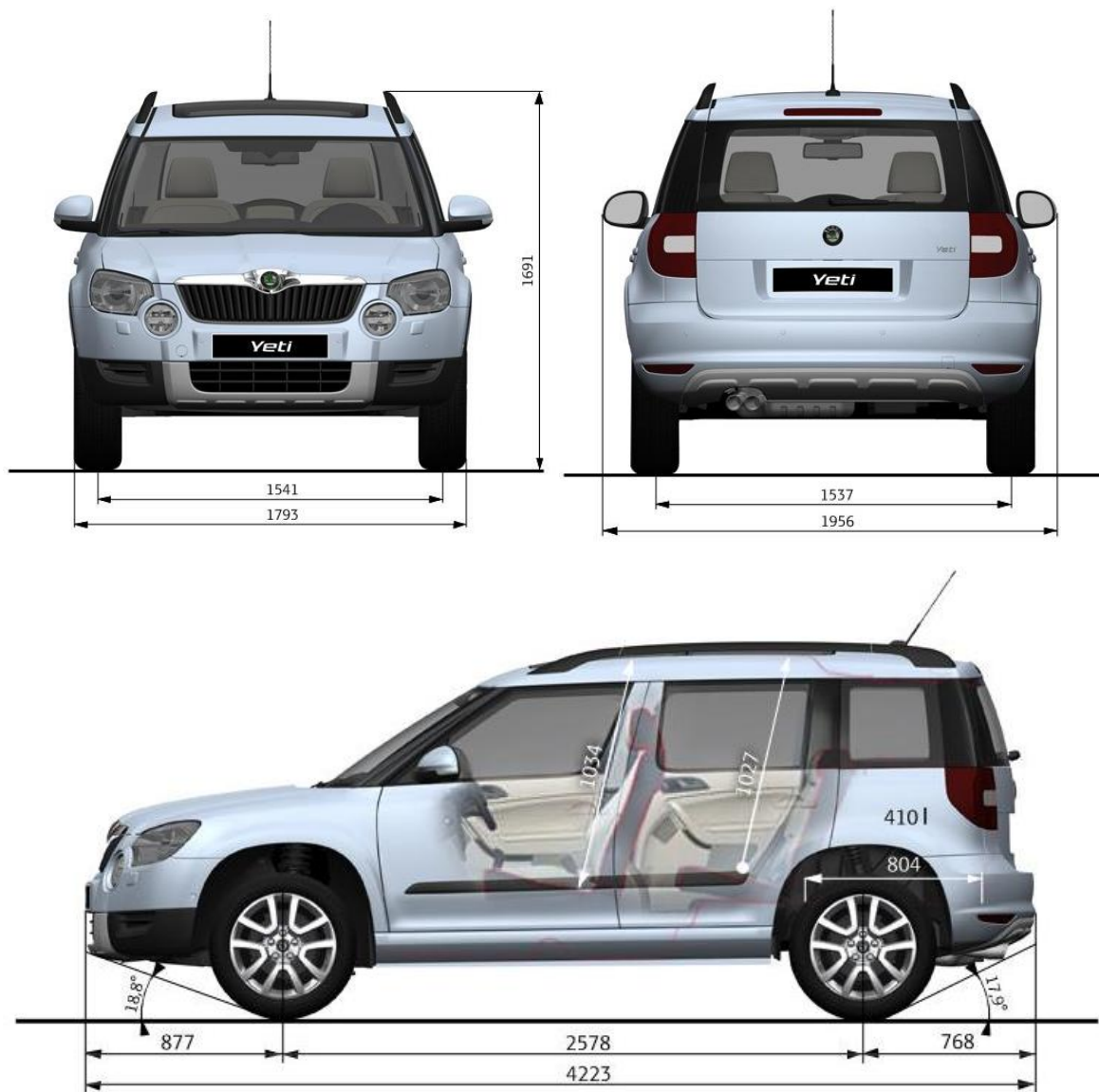
Obdobne aj pre subsystém Rýchlosť.

2.3 Grafický model

Pre účel realistického vykreslenia sa podkladom stalo vozidlo Škoda Yeti (Obr. 2.9). Podľa katalógových rozmerov (Tab. 2.1) bol vytvorený zjednodušený 2D model pôdorysu v programe MATLAB (Obr. 2.8). Účelom tohto modelu je graficky znázorniť priebeh vozidla po trajektórii a zároveň jasnejšie interpretovať dosiahnuté výsledky. Z dôvodu jednoduchosti tvarov je iba ilustračný. Pre neskoršie vyhodnocovanie kolízie uvažujeme tolerovaný priestor okolo vozidla znázornený čiarkovaným obdĺžnikom.



Obr. 2.8 Grafický model vozidla v programe MATLAB



Obr. 2.9 Základné rozmery vozidla Škoda Yeti [10]

Parameter	Jednotka	Hodnota	Popis
L	[m]	4.223	Celková dĺžka
W	[m]	1.793	Celková šírka
t_F/t_P	[m]	1.541 / 1.537	Rozchod nápravy (predná / zadná)
L_F/L_P	[m]	0.877 / 0.768	Vzdialenosť nápravy (predná / zadná)
l	[m]	2.578	Rázvor náprav
D_{out}	[m]	10.3	Vonkajší priemer zatočenia

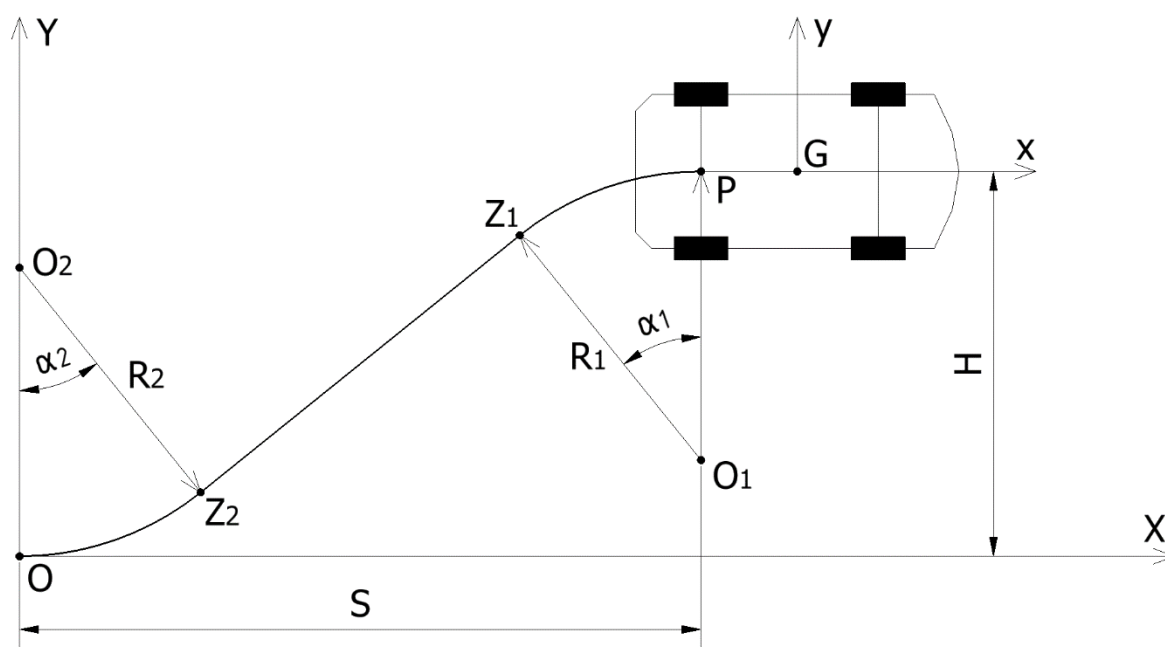
Tab. 2.1 Základné rozmery vozidla Škoda Yeti

3 Trajektória – analytický výpočet

V literatúre [6] je dokázané, že najkratšia uskutočniteľná trajektória je zložená z oblúkov a priamok. Pre túto prácu sme vybrali zloženie oblúk – priamka- oblúk, ďalej označenej ako CSC (z anglického Circle-Straight line-Circle).

3.1 Základný návrh CSC trajektórie

Na začiatok budeme uvažovať jednoduchú trajektóriu typu CSC v priestore bez prekážok. Účelom tohto návrhu je otestovať funkčnosť kinematického modelu a grafického vykreslenia priebehu manévru. Úloha je znázornená na obrázku nižšie (Obr. 3.1).



Obr. 3.1 CSC trajektória v priestore bez prekážok

Kde O_1 a O_2 sú stredy oblúkov počiatocnej a koncovej polohy, pričom R_1 a R_2 sú ich rádiusy. Body Z_1 a Z sú tangenciálne body medzi kružnicami a priamkou. Uhly α_1 a α_2 vymedzujú opísaný oblúk na jednotlivých kružniciach. Bod O označuje stred globálneho systému súradníc a zároveň je koncovým bodom trajektórie. Počiatocnú polohu popisujú súradnice S v horizontálnom a H vo vertikálnom smere. Stredy jednotlivých kružníc jednoducho vyjadríme z obrázka vyššie (Obr. 3.1) ako $O_1 = [S, H - R_1]$ a $O_2 = [0, R_2]$. Pre ďalší výpočet budeme vychádzať z analytickej geometrie, pomocou ktorej tvar kružníc vyjadríme ako

$$(x - a)^2 + (y - b)^2 = r^2, \quad (3.1)$$

kde a, b sú súradnice stredu kružnice a r je polomer.

Priamku spájajúcu oblúky vyjadríme v smernicovom tvare rovnice priamky ako:

$$y = mx + q, \quad (3.2)$$

kde m je smernica priamky a q je priesečník priamky s osou y .

Použitím rovníc (3.1) a (3.2) môžeme našu trajektóriu popísať rovnicami:

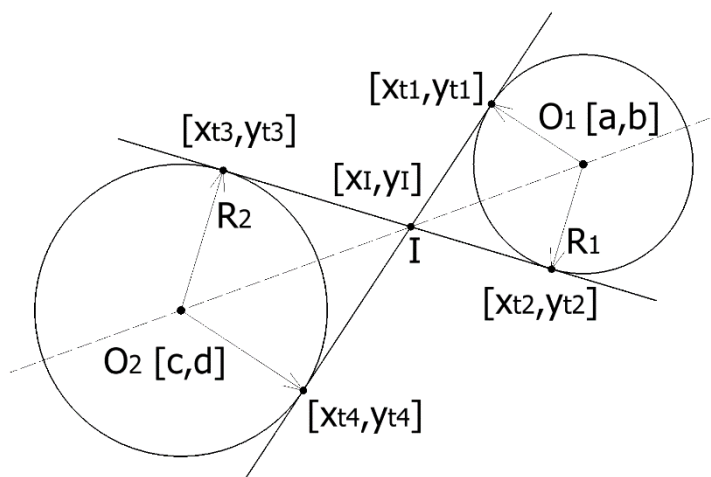
$$p \rightarrow z_1 : (x - S)^2 + (y - (H - R_1))^2 = R_1^2 \quad (3.3)$$

$$z_1 \rightarrow z_2 : y = mx + q \quad (3.4)$$

$$z_2 \rightarrow O : x^2 + (y - R_2)^2 = R_2^2 \quad (3.5)$$

Na výpočet priesečníkov Z_1, Z a parametrov priamky m, q použijeme algoritmus na výpočet vnútorných tangenciálnych čiar medzi dvoma kružnicami. Jeho postup opíšeme v nasledujúcich krokoch.

3.1.1 Algoritmus výpočtu vnútorných tangenciálnych čiar medzi dvoma kružnicami



Obr. 3.2 Vnútorné tangenciálne čiar medzi dvoma kružnicami

Krok č.1 : Výpočet priesečníku I medzi tangenciálnymi priamkami

$$x_I = \frac{cR_1 + aR_2}{R_1 + R_2} \quad y_I = \frac{dR_1 + bR_2}{R_1 + R_2} \quad (3.6)$$

Krok č.2 : Výpočet priesečníku medzi priamkou a kružnicou C_1

Keďže už poznáme súradnice priesečníku priamok I , môžeme jednoducho vypočítať súradnice priesečníkov priamok a kružníc, ktoré si označíme ako x_t a y_t , z rovníc:

$$x_{t1,2} = \frac{R_1^2(x_I - a) \pm R_1(y_I - b)\sqrt{(x_I - a)^2 + (y_I - b)^2 - R_1^2}}{(x_I - a)^2 + (y_I - b)^2} + a \quad (3.7)$$

$$y_{t1,2} = \frac{R_1^2(y_I - b) \mp R_1(x_I - a)\sqrt{(x_I - a)^2 + (y_I - b)^2 - R_1^2}}{(x_I - a)^2 + (y_I - b)^2} + b \quad (3.8)$$

Je dôležité pre každý bod použiť opačné znamienka pred odmocninou, inak vypočítané súradnice nebudú reprezentovať správny bod. Správnosť bodu skontrolujeme vypočítaním hodnoty k pomocou nasledujúceho vzorca. Ak $k = 1$, súradnice odpovedajú hľadanému bodu, inak treba vymeniť hodnoty $y_{t1} \leftrightarrow y_{t2}$.

$$\frac{(b - y_{t1})(y_I - y_{t1})}{(x_{t1} - a)(x_{t1} - x_I)} = k \qquad \frac{(b - y_{t2})(y_I - y_{t2})}{(x_{t1} - a)(x_{t1} - x_I)} = k \quad (3.9)$$

Krok č.3 : Výpočet priesečníku medzi priamkou a kružnicou C_2 .

Obdobne ako v predchádzajúcom kroku s použitím hodnôt kružnice C_2 .

Krok č.4 : Zostavenie rovníc pre vnútorné tangenciálne čiary.

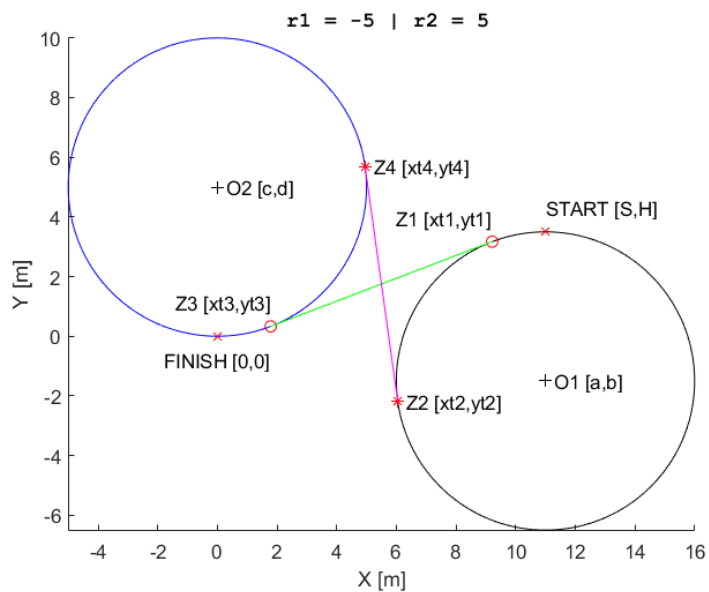
Jednotlivé priamky môžeme zapísať pomocou už známych súradníc rovnicami :

$$\frac{y - y_{t1}}{y_I - y_{t1}} = \frac{x - x_{t1}}{x_I - x_{t1}} \qquad \frac{y - y_{t3}}{y_I - y_{t3}} = \frac{x - x_{t3}}{x_I - x_{t3}} \quad (3.10)$$

Po jednoduchej úprave do tvaru (3.2) získame rovnice na výpočet smernice m a priesečníku s osou y q .

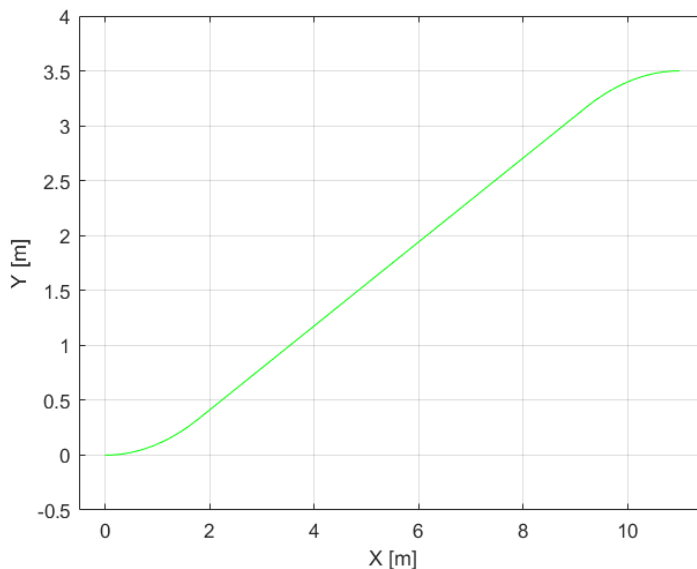
$$m = \frac{y_I - y_t}{x_I - x_t} \qquad q = y_t - m \cdot x_t \quad (3.11)$$

Použitím tohto algoritmu našu vybranú trajektóriu vykreslíme v prostredí MATLAB:



Obr. 3.3 Zobrazenie výsledku algoritmu v prostredí programu MATLAB

Pre túto trajektóriu bol zvolený štartovací bod ako počiatočná poloha vozidla, určený súradnicami $S = 11\text{m}$ a $H = 3.5\text{m}$. Polomery pre úvodné zjednodušenie budeme uvažovať rovnakej veľkosti a zaokrúhlime ich na 5m. Na obrázku vyššie (Obr. 3.3) je znázornený výsledok algoritmu, ktorý spočítal oboje tangenciálne čiary. Pre náš prípad je vyhovujúca trajektória prechádzajúca bodmi Z_1 a Z_2 . Preto výslednú trajektóriu vykreslíme nasledovne:



Obr. 3.4 Zobrazenie výslednej trajektórie CSC v prostredí programu MATLAB

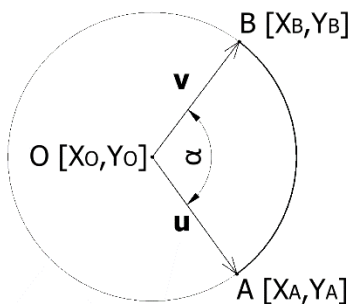
3.1.2 Výpočet dĺžky trajektórie

Celkovú dĺžku trajektórie vypočítame ako súčet jej jednotlivých častí, v našom prípade dva oblúky a úsečka.

Dĺžka oblúka sa určí ako súčin polomeru oblúka a uhla, ktorý vozidlo opíše. Vyjadríme to rovnicou:

$$S_c = r \cdot \alpha, \quad (3.12)$$

kde uhol α dostaneme z jednoduchšej úvahy o uhle dvoch vektorov nasledovným spôsobom.



$$\mathbf{u} = A - O = [x_A - x_O, y_A - y_O] \quad (3.13)$$

$$\mathbf{v} = B - O = [x_B - x_O, y_B - y_O] \quad (3.14)$$

$$\alpha = \arccos \frac{u_x v_x + u_y v_y}{|\mathbf{u}| \cdot |\mathbf{v}|} \quad (3.15)$$

Obr. 3.5 Výpočet uhla alfa

Dĺžku úsečky spočítame ako vzdialenosť dvoch bodov.

$$S_s = \sqrt{(x_{t1} - x_{t2})^2 + (y_{t1} - y_{t2})^2} \quad (3.16)$$

Z toho celkovú dĺžku trajektórie získame ich súčtom.

$$S = S_{C1} + S_s + S_{C2} \quad (3.17)$$

3.1.3 Simulácia kinematického modelu

Vstupy pre simuláciu

Z vyššie vypočítaných výsledkov určíme vstupy potrebné pre simuláciu kinematického modelu, a to uhol natočenia kolies vozidla $\varphi(t)$. Druhý vstup, rýchlosť, budeme uvažovať konštantnej veľkosti.

Z dôvodu časovej závislosti vstupov simulácie si vypočítame časy trvania jednotlivých úsekov manévru, pričom budeme vychádzať z vypočítaných dĺžok a konštantnej rýchlosti.

$$t_i = \frac{S_i}{v}, \quad (3.18)$$

kde i je index pre jednotlivé časti CSC trajektórie.

Celkový čas manévru získame ich súčtom:

$$t_{total} = t_{C1} + t_S + t_{C2} \quad (3.19)$$

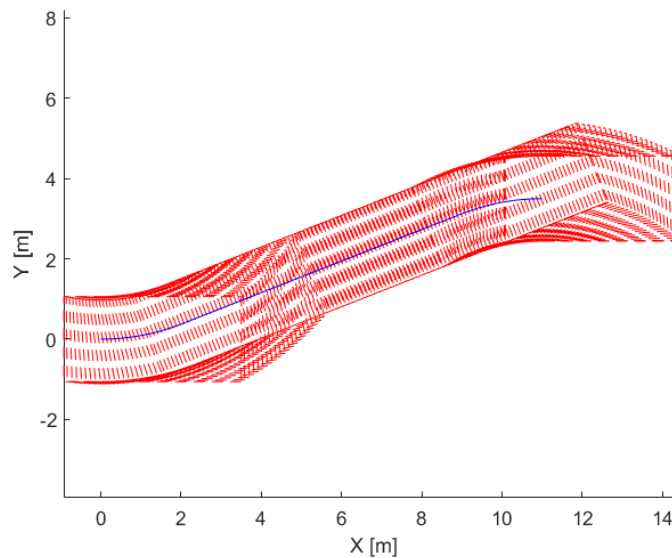
Pre jednotlivé časové úseky určíme potrebné natočenie kolies z rovnice (2.1)

$$\varphi = \arctan\left(\frac{l}{R}\right), \quad (3.20)$$

kde l je rázvor náprav, tzn. vzdialenosť medzi prednou a zadnou nápravou.

Výsledok simulácie kinematického modelu

Na obrázku nižšie (Obr. 3.6) je zobrazený výsledok simulácie kinematického modelu vozidla, pohybujúceho sa po vypočítanej trajektórii CSC, splňujúcej počiatočné podmienky. Červenou farbou je vykreslená obálka obdĺžnika, znázorňujúca priestor vozidla. Pre vyššiu prehľadnosť obrázka nie je zobrazený pôdorys vozidla. Modrou farbou je ďalej vykreslená trajektória určená z algoritmu. Danú trajektóriu opisuje stred zadnej nápravy vozidla.



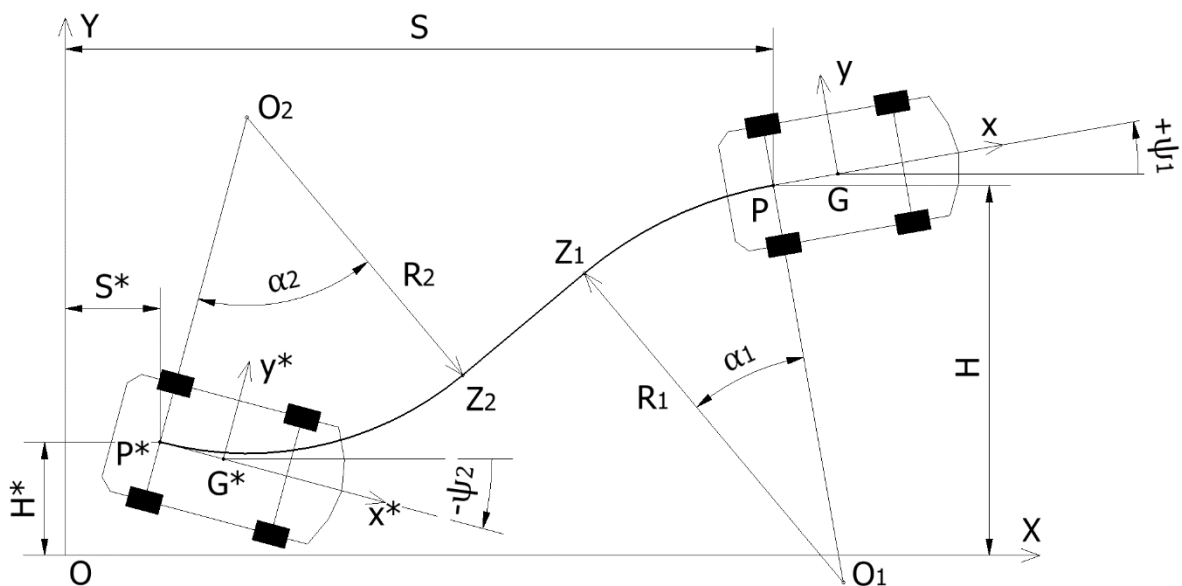
Obr. 3.6 Obálka vozidla opisujúceho trajektóriu CSC

3.1.4 Zhodnotenie základného návrhu CSC trajektórie

Úlohou prvého návrhu trajektórie bolo overiť funkčnosť kinematického modelu a grafického vykreslenia priebehu manévru. Ukázalo sa, že náš kinematický model funguje spoľahlivo pre tento jednoduchý typ trajektórie. Grafické vykreslenie taktiež splňuje požiadavky a preukázalo sa ako funkčné.

3.2 Plánovanie trajektórie pre prípad s nenulovými natočeniami vozidla

V mnohých prípadoch počiatočné natočenie vozidla nie je rovnobežné s parkovacím miestom. Z tohto dôvodu je potrebné navrhnúť algoritmus hľadania trajektórie parkovacieho manévru pre ľubovoľný počiatočný uhol natočenia vozidla. Príklad úlohy s takýmito počiatočnými podmienkami je na obrázku nižšie (Obr. 3.7). Pri hľadaní trajektórie vozidla budeme aj naďalej považovať za ideálnu trajektóriu, zloženú z dvoch oblúkov a priamky (CSC), alebo v niektorých prípadoch len z dvoch oblúkov (CC). Pre možnosť následného použitia algoritmu pre hľadanie zložitejších manévrov kladieme dôraz na robustnosť, a teda všetky hodnoty uvažujeme ako všeobecné. Pre ukážku algoritmu a overenia jeho funkčnosti budeme aj naďalej uvažovať priestor bez prekážok.

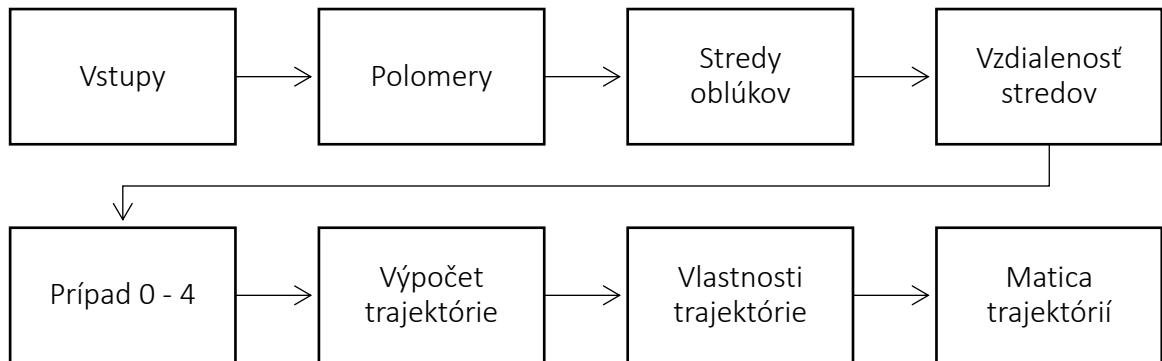


Obr. 3.7 Hľadanie trajektórie pre všeobecný prípad

Počiatočná poloha vozidla je definovaná polohou bodu P v globálnom systéme súradníc a počiatočným natočením vozidla. Pre lepšiu orientáciu v algoritme ju označíme ako $START = (S, H, \psi_1)$ a obdobne aj koncovú polohu označíme ako $FINISH = (S^*, H^*, \psi_2)$. Body Z_1 a Z_2 sú priesečníky medzi jednotlivými časťami trajektórie. Pre každý bod trajektórie platí predpis $P' =$

$(X(t), Y(t), \psi(t))$, kde t je čas. Uhly oblúkov α_1 a α_2 dostaneme z rovnice (3.15). Určenie hodnôt polomerov R_1 a R_2 a polôh stredov oblúkov O_1 a O_2 si ukážeme neskôr pri opisovaní algoritmu.

3.2.1 Algoritmus



Obr. 3.8 Základná schéma algoritmu pre nájdenie možných trajektórií

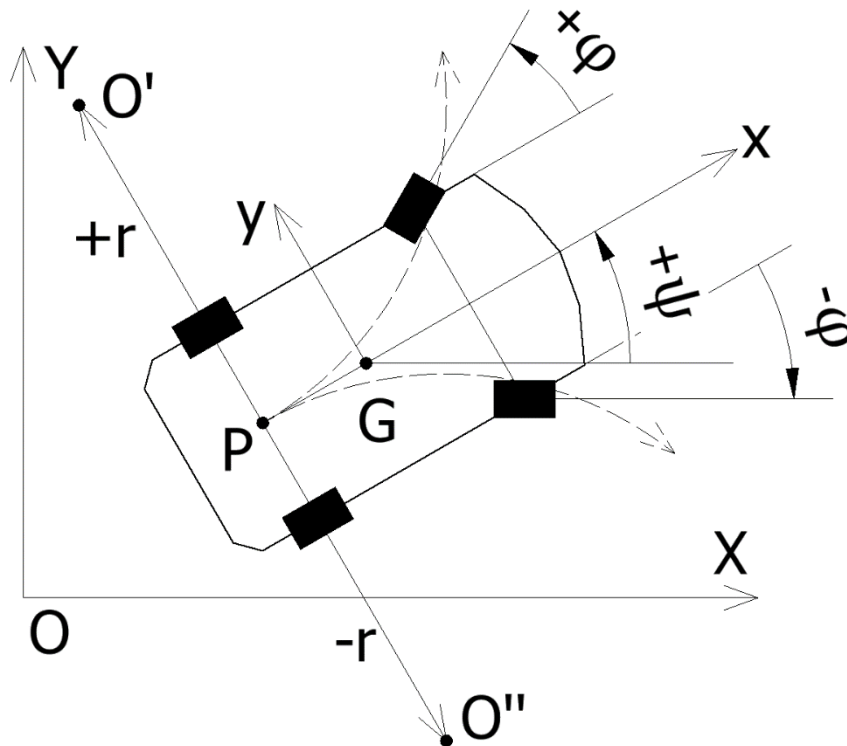
Na obrázku vyššie (Obr. 3.8) je znázornená zjednodušená schéma algoritmu. Jednotlivé časti si vysvetlíme v nasledujúcom texte.

Vstupy

Vstupmi pre algoritmus sú počiatočná pozícia vozidla $START = (S, H, \psi_1)$ a koncová požadovaná pozícia poloha $FINISH = (S^*, H^*, \psi_2)$, ktoré sú dané polohou stredy zadnej nápravy v globálnom systéme súradníc a natočením vozidla voči vodorovnej ose globálneho systému súradníc. Ďalšími dôležitými vstupmi sú hodnota maximálneho natočenia kolies φ_{max} a rázvor náprav l , ktoré získame z vlastností vozidla a budeme ich potrebovať na určenie minimálneho možného polomeru oblúkov.

Polomery oblúkov

Polomery jednotlivých oblúkov vypočítame z rovnice (2.1). Budú nás zaujímať všetky hodnoty, a preto za φ dosadíme hodnoty z intervalu $\langle -\varphi_{max}; +\varphi_{max} \rangle$. Z dôvodu výpočtu na počítači je treba úlohu počítať ako diskretnú, a preto sa vhodne zvolí krok $\Delta\varphi$. Algoritmus bude prebiehať v dvoch cykloch, kedy sa pre každú hodnotu r_2 dosadí postupne každá hodnota r_1 .



Obr. 3.9 Orientácia polomeru pri pohybe vľavo a vpravo

Pre rozlíšenie smeru pohybu a následného výpočtu výstupov musíme zaviesť orientovaný polomer, ktorého smer určuje iba jeho znamienko. Ako možno vidieť na obrázku vyššie (Obr. 3.9), orientácia polomeru je závislá na znamienku uhla natočenia kolies. Z vlastností vozidla je zjavné, že kladný uhol φ , a teda aj kladný polomer r znamená zatáčanie vozidla doľava a záporný doprava.

Nasledujúce výpočty analytickej geometrie nie sú definované pre záporné hodnoty polomeru kružnice, a preto zavedieme hodnotu R podľa rovnice (3.21), ktorá bude niesť informáciu len o veľkosti polomeru. Určenie orientácie bude dôležité pre výpočet polohy stredu oblúka, pri výpočte výstupného uhla natočenia kolies a pri určovaní smeru pohybu. Ale v procese výpočtu geometrických vlastností nebude na orientácii záležať.

$$R_i = |r_i| \quad (3.21)$$

Súradnice stredov oblúkov

Pre obecný bod $P' = (X_p, Y_p, \psi_p)$ jednoducho z obrázka vyššie (Obr. 3.9) určíme polohu stredu oblúka:

$$O = [X_p - r \cdot \sin \psi, Y_p + r \cdot \cos \psi]. \quad (3.22)$$

Pre našu zadanú úlohu pomocou rovnice (3.22) jednoducho určíme polohy stredov oblúkov:

$$O_1 = [a, b] = [S - r_1 \cdot \sin \psi_1, H + r_1 \cdot \cos \psi_1] \quad (3.23)$$

$$O_2 = [c, d] = [S^* - r_2 \cdot \sin \psi_2, H^* + r_2 \cdot \cos \psi_2]. \quad (3.24)$$

Vzdialenosť stredov oblúkov

Vstupom pre nasledujúci bod algoritmu bude vzdialenosť stredov oblúkov. Použijeme vzorec na výpočet vzdialenosti dvoch bodov v priestore (3.16) a dostaneme rovnicu:

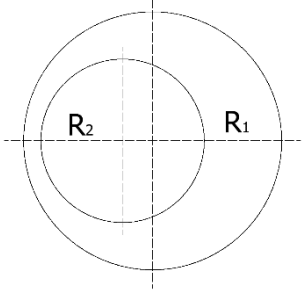
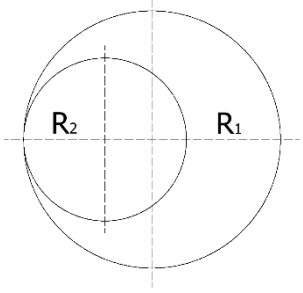
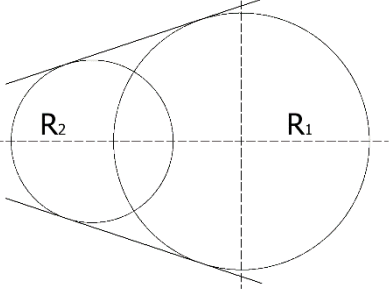
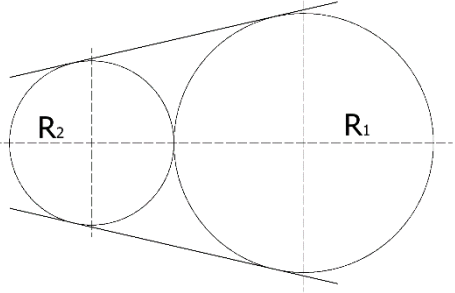
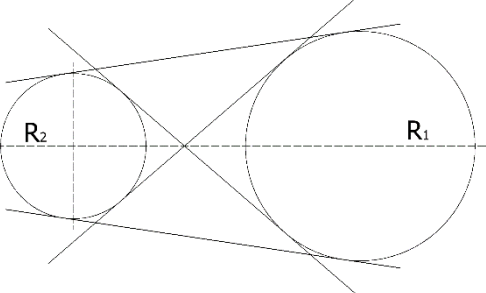
$$D = \sqrt{(a - c)^2 + (b - d)^2} \quad (3.25)$$

Možné prípady polohy dvoch kružníc

Pri dvoch kružniciach v priestore môže nastať päť rôznych prípadov ich vzájomnej polohy, ktoré nám povedia, koľko tangenciálnych čiar medzi nimi vznikne. Na základe toho budeme postupovať pri geometrickom výpočte trajektórie pre každú kombináciu polomerov v priebehu cyklu algoritmu.

Tabuľka nižšie (Tab. 3.1) znázorňuje všetky možné prípady pre nami zvolené druhy trajektórií. Tabuľka obsahuje grafické zobrazenie schémy pre každý prípad, podmienku, za ktorej daný prípad nastáva, a počet možných trajektórií typu CSC a CC. Schémy sú len ilustračné, a teda sú len špeciálnym prípadom jednotlivých prípadov polôh, ktoré platia pre ľubovoľnú polohu stredov splňujúcu danú podmienku.

Špecifický je nultý prípad, kedy neexistuje žiadna trajektória typu CSC ani CC. Preto ak tento prípad nastane, algoritmus ukončí daný cyklus a prejde na nasledujúcu kombináciu polomerov.

Prípado č.	Schéma	Podmienka	Počet trajektórií	
			CSC	CC
0		$D < R_1 - R_2 $	0	0
1		$D = R_1 - R_2 $	0	1
2		$ R_1 - R_2 < D < R_1 + R_2$	2	0
3		$D = R_1 + R_2$	2	1
4		$D > R_1 + R_2$	4	0

Tab. 3.1 Možné prípady polohy dvoch kružníc v priestore

Výpočet trajektórie

V závislosti na prípadoch z predchádzajúcej kapitoly budeme rozlišovať medzi dvoma typmi trajektórie, a teda ich výpočty budú mierne odlišné.

Algoritmus výpočtu vnútorných tangenciálnych čiar medzi dvomi kružnicami sme podrobne opísali v kapitole 3.1.1. Tento algoritmus použijeme v prípade číslo štyri.

V prípadoch číslo dva, tri a štyri použijeme algoritmus výpočtu vonkajších tangenciálnych čiar medzi dvomi kružnicami. Bude veľmi podobný ako pre vnútorné tangenciálne čiary. Jediný rozdiel nastane v prvom kroku, kedy sa pri výpočte priesečníku I medzi tangenciálnymi priamkami zmenia oboje znamienka. Dostaneme rovnice v tvare:

$$x_I = \frac{cR_1 - aR_2}{R_1 - R_2} \qquad y_I = \frac{dR_1 - bR_2}{R_1 - R_2} \qquad (3.26)$$

Zvyšok algoritmu bude totožný.

Tento algoritmus však platí iba v prípade rozdielnych polomerov kružníc, a teda za podmienky $R_1 \neq R_2$. V prípade, že podmienka nie je splnená a polomery sú totožné, neexistuje priesečník I , pretože tangenciálne čiary sú rovnobežky. V tomto prípade preskakujeme prvý krok a ďalej postupuje nasledujúcim spôsobom:

Krok č.2 : Výpočet priesečníku medzi priamkou a kružnicou C_1

$$x_{t1,2} = a \mp R_1 \cdot \sin \gamma \qquad (3.27)$$

$$y_{t1,2} = b \pm R_1 \cdot \cos \gamma \qquad (3.28)$$

Kde uhol γ je sklon smernice priamky prechádzajúcej stredmi kružníc a dostaneme ho zo vzťahu:

$$\gamma = \arctan \frac{b-d}{a-c} \qquad (3.29)$$

Zvyšok algoritmu bude analogický k pôvodnému algoritmu.

V prípadoch číslo jedna a tri je vhodné použiť druhý typ trajektórie bez priamkovej časti, a teda už spomínanú trajektóriu typu CC. Stačí nám vypočítať jeden priesečník medzi kružnicami:

$$x_t = \frac{1}{2} \left[\frac{(c-a)(R_1^2 - R_2^2)}{(a-c)^2 + (b-d)^2} + a + c \right] \qquad y_t = \frac{1}{2} \left[\frac{(d-b)(R_1^2 - R_2^2)}{(a-c)^2 + (b-d)^2} + b + d \right] \qquad (3.30)$$

Z dôvodu diskkrétnej úlohy sa môže stať, že tento prípad nenastane. Aby nastal, tak do množiny hodnôt polomerov prvej kružnice pridáme hodnoty splňujúce podmienky pre prípady číslo jeden a tri. Vychádzame z úvahy rovnosti vzdialenosti stredov kružníc a súčtu (pre prípad tri), prípadne rozdielu (pre prípad jeden) polomerov.

$$D^2 = (a - c)^2 + (b - d)^2 = (R_1^2 \pm R_2^2)^2 \quad (3.31)$$

Dosadíme z rovníc (3.23) a (3.24) a po úpravách dostaneme rovnicu na výpočet polomeru prvej kružnice R_1 v závislosti na polomere druhej kružnice R_2 .

$$r_{01} = \frac{r_2^2 - (S - S^* + r_2 \cdot \sin \psi_2)^2 - (H - H^* - r_2 \cdot \cos \psi_2)^2}{2[-\sin \psi_1 \cdot (S - S^* + r_2 \cdot \sin \psi_2) + \cos \psi_1 \cdot (H - H^* - r_2 \cdot \cos \psi_2) \mp r_2]} \quad (3.32)$$

Za predpokladu, že polomery splňujú podmienku $|r_{01}| \geq |r_{min}|$, sú pridané do množiny polomerov na začiatku algoritmu.

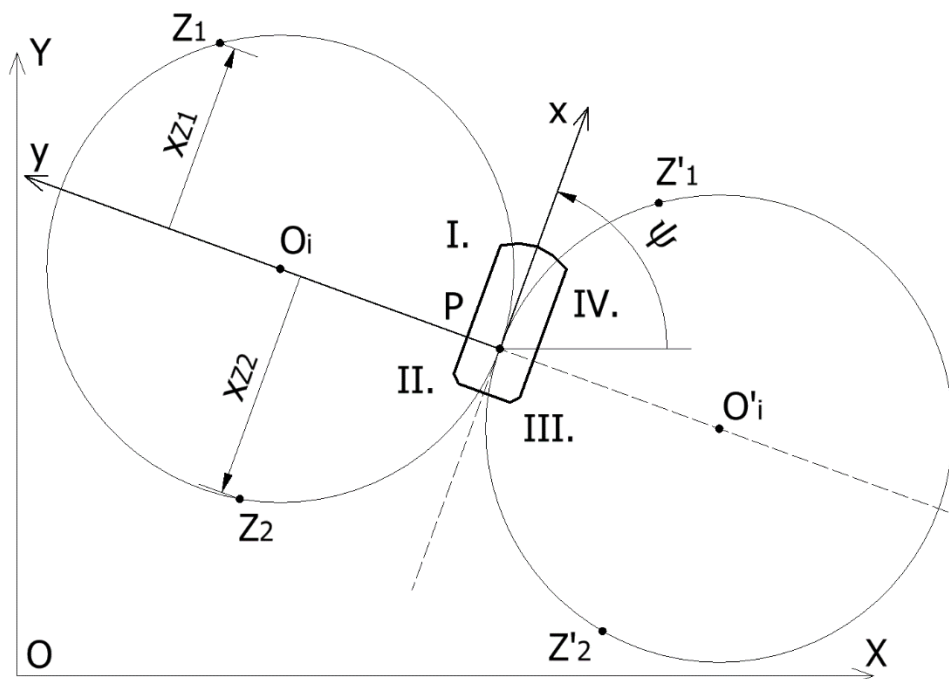
Vlastnosti trajektórie

Dĺžka trajektórie

Dĺžku budeme počítať rovnakým spôsobom ako v odseku 3.1.2. Dĺžku oblúka dostaneme zo vzorca (3.12) a dĺžku úsečky zo vzorca (3.16). Pre trajektóriu CC bude dĺžka úsečky nulová.

Smer pohybu

Veľkosť rýchlosti pre nás nie je dôležitá, a tak ju budeme uvažovať konštantnú $|v_x| = 1 \text{ ms}^{-1}$. Dôležitý však bude smer pohybu vozidla. Pre pohyb vpred budeme uvažovať rýchlosť $v_x = 1 \text{ ms}^{-1}$ a pre smer vzad rýchlosť $v_x = -1 \text{ ms}^{-1}$. Smer (znamienko) rýchlosti určíme z jednoduchšej úvahy. Do bodu $p [X_p, Y_p]$ umiestnime systém súradníc a natočíme ho o uhol ψ_p tak, aby sa zhodol so súradnicovým systémom vozidla. Z kinematiky súčasných pohybov vypočítame polohu cieľového bodu Z . Pokiaľ sa cieľový bod Z nachádza v prvom alebo štvrtom kvadrante ($x_z > 0$), vozidlo sa bude pohybovať vpred, ale ak sa nachádza v druhom alebo treťom kvadrante ($x_z < 0$), bude sa pohybovať vzad. Pre ilustráciu je všeobecný príklad zobrazený na obrázku nižšie (Obr. 3.10).



Obr. 3.10 Určenie smeru pohybu

Pre určenie polohy bodu Z vychádzame z maticového zápisu súčasných pohybov:

$$\begin{bmatrix} X_Z \\ Y_Z \end{bmatrix}^{xy} = \begin{bmatrix} X_P \\ Y_P \end{bmatrix}^{xy} + \begin{bmatrix} \cos \psi & -\sin \psi \\ \sin \psi & \cos \psi \end{bmatrix} \cdot \begin{bmatrix} x_Z \\ y_Z \end{bmatrix}^{xy} \quad (3.33)$$

a po úprave dostaneme tvar:

$$\begin{bmatrix} x_Z \\ y_Z \end{bmatrix}^{xy} = \begin{bmatrix} \cos \psi & \sin \psi \\ -\sin \psi & \cos \psi \end{bmatrix} \cdot \left(\begin{bmatrix} X_Z \\ Y_Z \end{bmatrix}^{xy} - \begin{bmatrix} X_P \\ Y_P \end{bmatrix}^{xy} \right) \quad (3.34)$$

Uhol natočenia ψ v bode Z

Pre určenie smeru pohybu po úsečke budeme potrebovať počiatkový uhol natočenia, ktorý sa zhoduje s koncovým uhlom natočenia pohybu po oblúku. Vypočítame ho z úvahy, že pre ľubovoľný bod kružnice platí rovnica (3.22). Pre vyšetrovateľný bod Z bude mať tvar:

$$O_i = [X_Z - r_i \cdot \sin \psi_Z, Y_Z + r_i \cdot \cos \psi_Z], \quad (3.35)$$

kde $i = 1, 2$ je index pre kružnicu, na ktorej sa bod Z nachádza. Rovnica (3.22) platí aj pre počiatkový bod oblúka $START = (S, H, \psi_1)$. Z rovnosti $O_1(Z) = O_1(START)$ dostaneme sústavu dvoch rovníc pre jednu neznámu:

$$S - r_1 \cdot \sin \psi_1 = X_Z - r_1 \cdot \sin \psi_Z \quad (3.36)$$

$$H + r_1 \cdot \cos \psi_1 = Y_Z + r_1 \cdot \cos \psi_Z \quad (3.37)$$

Vyjadříme z jednotlivých rovnic hľadaný uhol ψ_Z :

$$\psi_Z = \arcsin\left(\frac{S - r_1 \cdot \sin \psi_1 - X_Z}{-r_1}\right) \quad (3.38)$$

$$\psi_Z = \arccos\left(\frac{H + r_1 \cdot \cos \psi_1 - Y_Z}{r_1}\right) \quad (3.39)$$

Túto sústavu nelineárnych rovnic vyriešime jednoducho v prostredí programu MATLAB pomocou funkcie *atan2*:

$$\gg \psi_Z = \text{atan2}\left(\frac{S - r_1 \cdot \sin \psi_1 - X_Z}{-r_1}, \frac{H + r_1 \cdot \cos \psi_1 - Y_Z}{r_1}\right) \quad (3.40)$$

Pozn.: $\text{atan2}(y, x)$ je štvorkvadrantový inverzný tangens vracajúci hodnotu z intervalu $\langle -\pi, \pi \rangle$, ktorá je zhodná s hodnotou uhlu medzi osou x a polohovým vektorom bodu (x, y) . [11]

Matica trajektórií

Všetky vlastnosti a dôležité parametre trajektórie zapíšeme do stĺpcového vektoru a ten následne do celkovej matice trajektórií T . Dostávame maticu nesúcu informácie o všetkých možných trajektóriách typu CSC a CC zo zvoleného počiatočného bodu $START = (S, H, \psi_1)$ do zvoleného koncového bodu $FINISH = (S^*, H^*, \psi_2)$ pre diskrétno natočenie kolies v intervale $\langle -\varphi_{max}; +\varphi_{max} \rangle$ a zvoleným krokom $\Delta\varphi$.

3.2.2 Simulácia kinematického modelu

Vstupy pre simuláciu

Nachádzame sa v priestore bez prekážok, a preto ako vstup pre simuláciu vezmeme najkratšiu trajektóriu. Z matice trajektórií T získame požadovaný vektor s vlastnosťami trajektórie v prostredí programu MATLAB jednoduchým príkazom:

$$\gg [M, I] = \text{min}(T(1, :)), \quad (3.41)$$

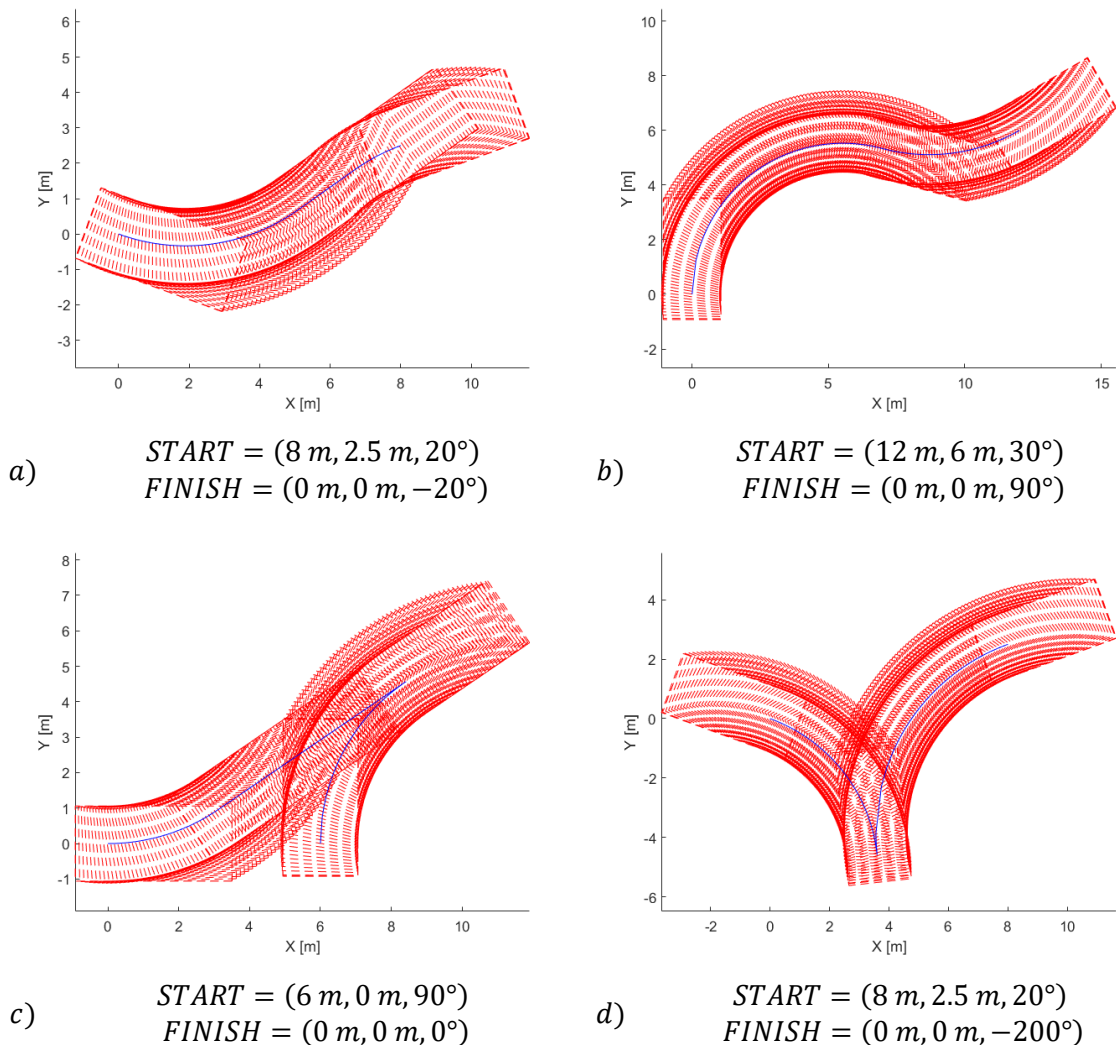
kde M je minimálna hodnota, I je index stĺpca obsahujúceho hľadanú minimálnu hodnotu.

Pozn.: Zápis $\text{min}(T(1, :))$ znamená, že hľadáme minimum v prvom riadku matice vo všetkých stĺpcoch.

Pre simuláciu budeme potrebovať celkový čas a uhol natočenia kolies φ pre jednotlivé úseky. Na ich určenie použijeme vybraný stĺpec matice trajektórií $T(:, I)$, obsahujúci potrebné hodnoty. Postup ich určenia bol opísaný v odseku 3.1.3.

Výsledok simulácie kinematického modelu

Na obrázku nižšie (Obr. 3.11) sú zobrazené výsledky simulácie kinematického modelu vozidla, pohybujúceho sa po vypočítanej trajektórii, splňujúcej počiatočné podmienky. Variant *a*) zobrazuje výsledok úlohy zo začiatku kapitoly (Obr. 3.7) a pri variante *d*) je koncová poloha otočená o 180° . Varianty *a*), *c*) sú trajektórie typu CSC a varianty *b*), *d*) sú trajektórie typu CC. Pri variantoch *c*), *d*) možno vidieť zmenu smeru počas manévru.



Obr. 3.11 Obálka vozidla opisujúceho trajektóriu pre rôzne prípady

Červenou farbou je vykreslená obálka obdĺžnika, znázorňujúca priestor vozidla. Pre vyššiu prehľadnosť obrázka nie je zobrazený pôdorys vozidla. Modrou farbou je ďalej vykreslená trajektória určená z algoritmu. Danú trajektóriu opisuje stred zadnej nápravy vozidla.

3.2.3 Zhodnotenie plánovania trajektórie pre prípad s nenulovými natočeniami vozidla

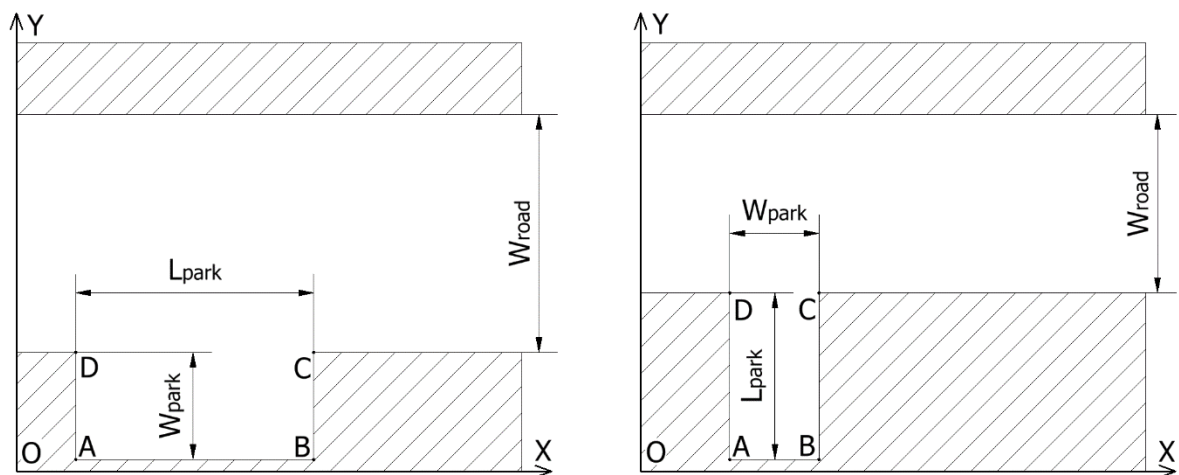
Úlohou druhého návrhu trajektórie bolo vytvoriť robustný algoritmus pre úlohu s nenulovými natočeniami v počiatočnom a koncovom bode. Funkčnosť a robustnosť algoritmu bola overená na mnohých kombináciách počiatočných podmienok v priestore bez prekážok. Vybrané výsledky boli prezentované v odseku 3.2.2.

4 Kolízia – prostredie s prekážkami

Predpokladajme, že okolie vozidla je vyhodnocované všetkými druhmi snímačov. Počiatočný stav vozidla, parkovacie miesto a ďalšie užitočné informácie sa určia z údajov získaných zo senzorov.

4.1 Grafické vyhodnotenie kolízie

V tejto kapitole sa budeme venovať dvom prípadom parkovania. Ilustračne sú znázornené na obrázku nižšie (Obr. 4.1). Pri parkovaní pozdĺžne, zobrazené vľavo, bude uhol koncovej polohy $\psi_{FINISH} = 0^\circ$ a pri parkovaní kolmo, zobrazené vpravo, bude uhol koncovej polohy $\psi_{FINISH} = 90^\circ$.



Obr. 4.1 Uvažované typy prostredia

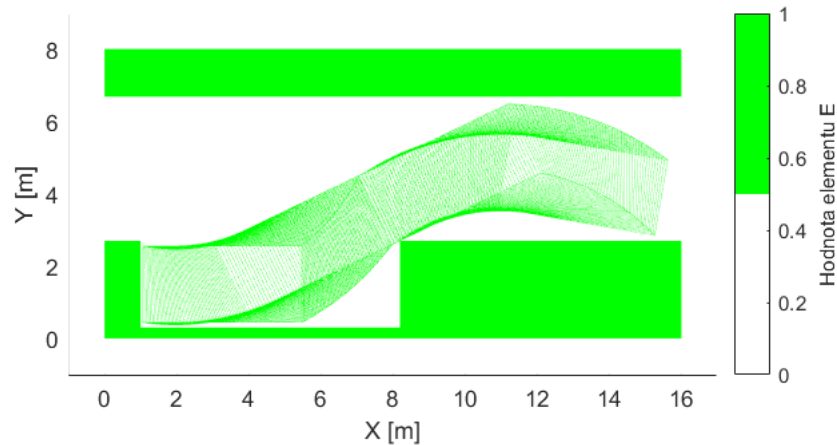
Pre vyhodnotenie kolízie v prostredí programu MATLAB bol vytvorený raster celého priestoru so zvoleným rozsahom elementu siete Δe . Každému elementu bola primárne nastavená nulová hodnota ($\forall E_{ij} = 0$). Pomocou metódy ukázanej v kapitole 3 odseku 3.2.2 dokážeme vybranú trajektóriu obaliť obálkou. Každému elementu, ktorým obálka prechádza, nastavíme pevnú hodnotu 1 ($E_{ij} = 1$). Predpokladáme, že poznáme informácie o prostredí, a preto môžeme každému elementu predstavujúcemu súčasť prostredia alebo inej prekážky priradiť hodnotu +1 ($E_{ij} = E_{ij} + 1$). Takto získame mapu, na ktorej môžu elementy dosahovať troch rôznych hodnôt:

$E_{ij} = 0$	Elementom vozidlo neprechádza ani nie je súčasťou prostredia
$E_{ij} = 1$	Elementom vozidlo prechádza alebo je súčasťou prostredia
$E_{ij} = 2$	Elementom vozidlo prechádza a je súčasťou prostredia – dochádza ku kolízii

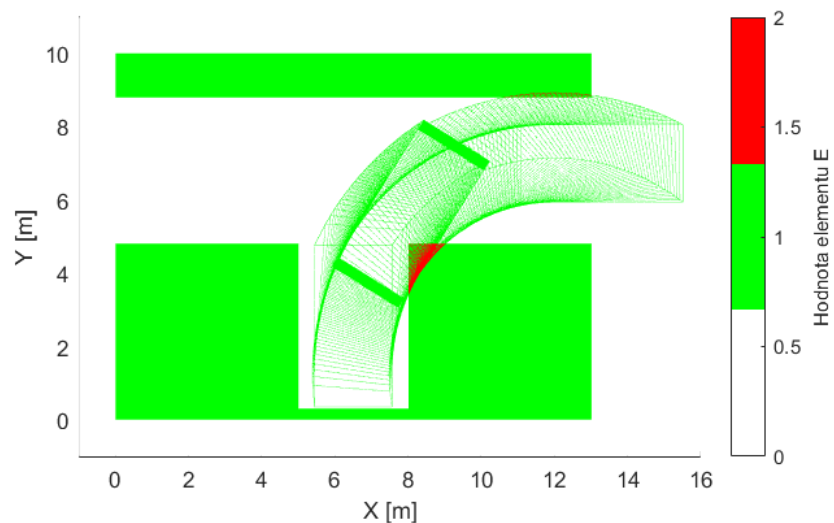
Tab. 4.1 Možné hodnoty elementu rastu

Kde i je index siete v smere osi Y globálneho systému súradníc a j je index siete v smere osi X .

Z tabuľky vyššie (Tab. 4.1) možno vidieť, že určenie kolízie je jednoduché. Pokiaľ je najvyššia hodnota elementu rovná 1 ($\max(E) = 1$), ku kolízii s prostredím nedochádza (Obr. 4.2). Avšak pokiaľ maximálna hodnota elementu je rovná 2 ($\max(E) = 2$), ku kolízii s prostredím dochádza (Obr. 4.3).



Obr. 4.2 Príklad mapy bez kolízie pri veľkosti elementu 1 cm



Obr. 4.3 Príklad mapy s kolíziou pri veľkosti elementu 1 cm

4.2 Algoritmus hľadania trajektórie s vyhodnotením kolízie

Nadviažeme na odsek 3.2.2, kedy sme ako jediné kritérium výberu trajektórie uvažovali jej dĺžku. Pokiaľ do algoritmu pridáme vyhodnotenie kolízie, vznikne nám druhé kritérium. V tomto prípade

budeme postupovať v cykle, pokiaľ nebudú splnené obe kritériá. Cyklus je pomocou pseudokódu v programe MATLAB znázornený na obrázku nižšie (Obr. 4.4).

```
while i < N
[M,I] = min(T(1,:));% Určenie najkratšej trajektórie
    if E == 2 % Nastala kolízia s prostredím
        T(:,I) = []; % Zvolená trajektória sa zmaže
    else
        break % Trajektória vyhovuje a ukonči sa cyklus
```

Obr. 4.4 Pseudokód cyklu vyhodnotenia kolízie

Kde číslo cyklu $i = 1, 2, \dots, N$, pričom N je počet nájdených trajektórií (počet stĺpcov matice trajektórií).

Môže nastať situácia, kedy nevyhovuje ani jedna trajektória z matice trajektórií T . V takom prípade dané parkovacie miesto nie je nami zvoleným postupom dosiahnuteľné pre aktuálne počiatkové podmienky.

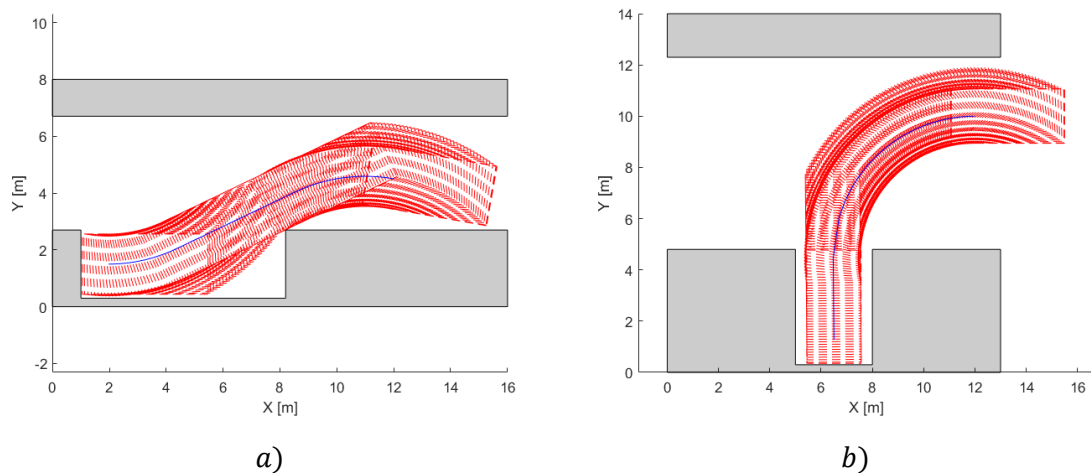
4.2.1 Simulácia kinematického modelu

Na obrázku nižšie (Obr. 4.5) sú zobrazené výsledky simulácie kinematického modelu vozidla, pohybujúcom sa po vypočítanej trajektórii, splňujúcej počiatkové podmienky a bez kolízie s prostredím. Odpovedajú zadaným typom prostredia zobrazenom na začiatku tejto kapitoly (Obr. 4.1). Počiatkové polohy a parametre prostredia sú uvedené v tabuľke nižšie (Tab. 4.2).

Parameter	Jednotka	Hodnota		Popis
		a)	b)	
$S; H$	[m]	12; 4.5	12; 10	Počiatková poloha $X; Y$
ψ_1	[°]	-10	0	Počiatkové natočenie vozidla
$S^*; H^*$	[m]	1.98; 1.5	6.5; 1.28	Koncová poloha $X; Y$
ψ_2	[°]	0	90	Koncové natočenie vozidla
L_{park}	[m]	7.2	4.5	Dĺžka parkovacieho miesta
W_{park}	[m]	2.4	3	Šírka parkovacieho miesta
W_{road}	[m]	4	7.5	Šírka cesty

Tab. 4.2 Počiatkové polohy a parametre prostredia

Červenou farbou je vykreslená obálka obdĺžnika, znázorňujúca priestor vozidla. Pre vyššiu prehľadnosť obrázka nie je zobrazený pôdorys vozidla. Modrou farbou je ďalej vykreslená trajektória určená z algoritmu. Danú trajektóriu opisuje stred zadnej nápravy vozidla. Sivou farbou sú znázornené prekážky tvorené typom prostredia.



Obr. 4.5 Obálka vozidla opisujúceho trajektóriu bez kolízie s prostredím

4.3 Zhodnotenie kolízie – prostredie s prekážkami

V tejto kapitole sme sa venovali vyhodnocovaniu kolízie. Podarilo sa nám vytvoriť funkčnú nadstavbu algoritmu na hľadanie trajektórií. Úspešne sme overili jeho funkčnosť na rôznych prípadoch. Dva z nich boli prezentované v odseku 4.2.1.

Celý postup bol do tejto chvíle založený na analytickom riešení hľadania trajektórie, ktorým sa pre jednoduché manévry podarilo úspešne nájsť trajektóriu. Avšak jeho použiteľnosť je obmedzená a pri zložitejších manévroch je nepoužiteľný. Nájdenie riešenia je veľmi citlivé na zvolené počiatkové podmienky s len veľmi malou prispôsobivosťou. Preto sa v nasledujúcej kapitole pozrieme na iný spôsob riešenia, ktorý by mohol pomôcť zlepšiť robustnosť a prispôsobivosť celého navrhovaného systému.

5 Trajektória – numerický výpočet

Niet pochýb, že hľadanie trajektórie môžeme považovať za optimalizačnú úlohu (napr. hľadanie najkratšej vzdialenosti) s určitými obmedzeniami (napr. dané prostredie a jazda bez kolízie).

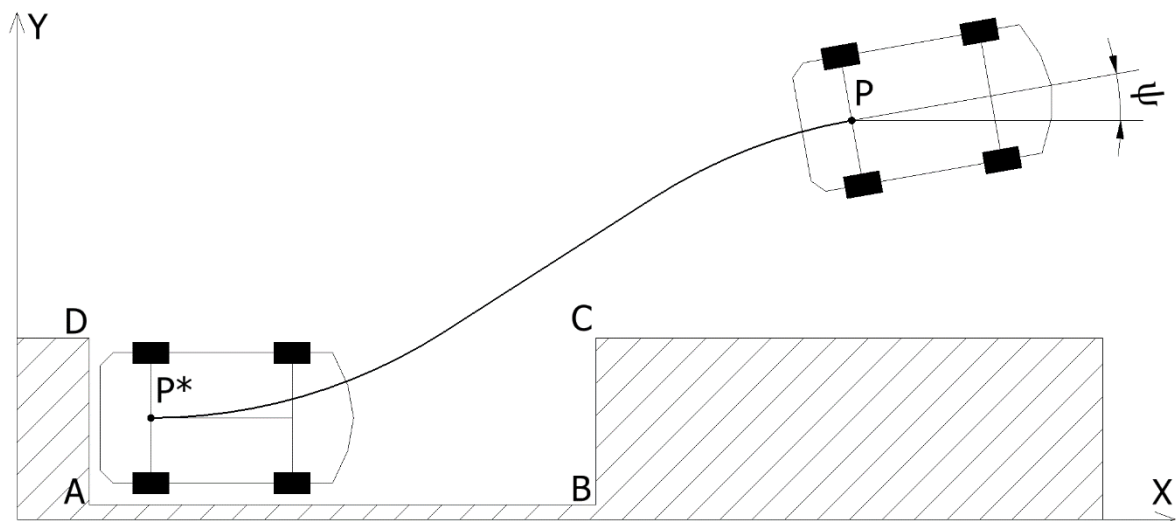
V tejto kapitole použijeme lokálnu a globálnu optimalizačnú metódu na hľadanie trajektórie vozidla.

5.1 Lokálna optimalizačná metóda

V tomto odseku si ukážeme riešenie už známych úloh z predchádzajúcich kapitol pomocou lokálnej optimalizačnej metódy. Všetky predpoklady a vlastnosti vozidla ostávajú nezmenené.

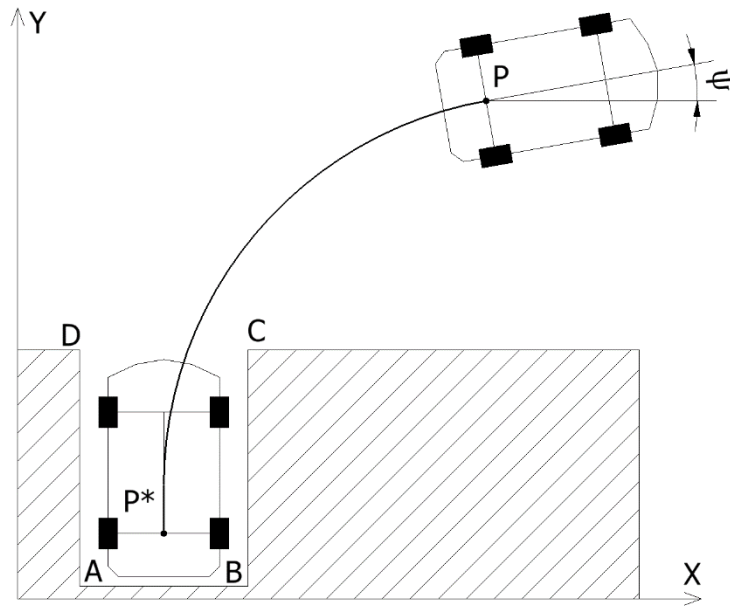
5.1.1 Formulácia úlohy

Úlohou je pomocou lokálnej optimalizačnej metódy nájsť trajektóriu vozidla (model popísaný v kapitole 2) z počiatočnej polohy do koncovej. Počiatočná poloha vozidla je definovaná polohou bodu p v globálnom systéme súradníc a počiatočným natočením vozidla označeným ako $START = (S, H, \psi_1)$ a obdobne aj koncová poloha označená ako $FINISH = (S^*, H^*, \psi_2)$.



Obr. 5.1 Zadanie optimalizačnej úlohy pre pozdĺžne parkovanie

Obrázok vyššie (Obr. 5.1) popisuje všeobecný prípad úlohy pozdĺžneho parkovania a obrázok nižšie (Obr. 5.2) popisuje všeobecný prípad úlohy kolmého parkovania. V oboch úlohách budeme vyhodnocovať kolíziu s prostredím popísanú v kapitole 4.



Obr. 5.2 Zadanie optimalizačnej úlohy pre kolmé parkovanie

Na rozdiel od typu trajektórie popisovanej v kapitole 3 budeme v týchto prípadoch používať trajektóriu zloženú iba z oblúkov pre jednoduchší popis.

5.1.2 Popis trajektórie

Na začiatok budeme uvažovať trajektóriu zloženú z troch oblúkov (typ CCC) pre lepšiu možnosť porovnania s predchádzajúcou analytickou metódou. Táto trajektória je podobná trajektórii typu CSC, keďže v prípade veľkého polomeru stredného oblúka je možné tento oblúk považovať za priamku. Pre skrátenie výpočtového času budeme uvažovať hornú hranicu polomeru oblúkov. Kompletný popis všeobecného prípadu jedného oblúka vysvetľuje nasledujúci odsek.

Popis oblúka trajektórie

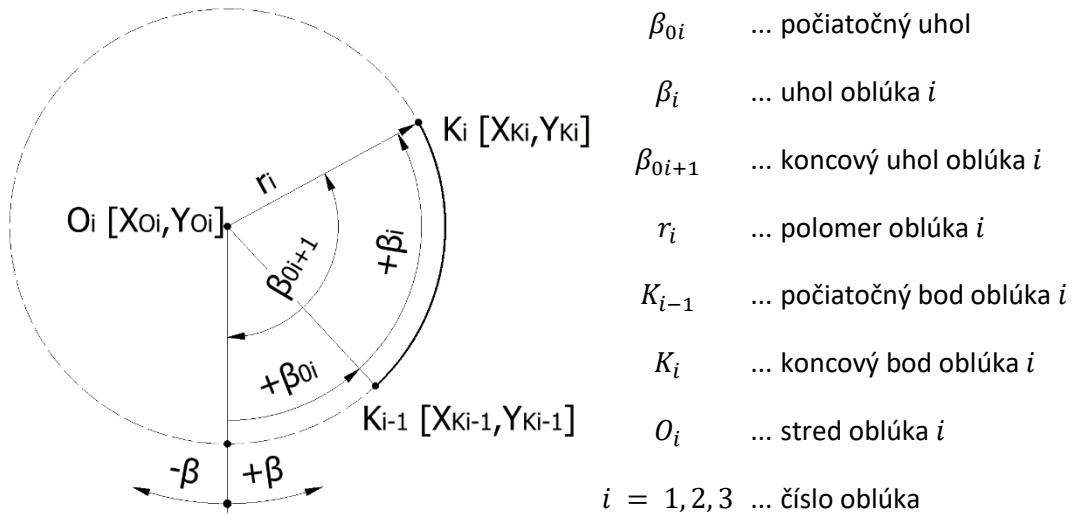
Pre popis oblúka použijeme polárnu sústavu súradníc (r, β) . Pohyb vozidla odpovedá všeobecnému pohybu po kružnici, a preto polárne súradnice popis zjednodušia. Následne ich prevedieme do globálneho (karteziánskeho) systému súradníc (X, Y) pomocou nasledujúcich rovníc:

$$X = X_{0i} + r_i \cdot \sin \beta \quad (5.1)$$

$$Y = Y_{0i} - r_i \cdot \cos \beta \quad (5.2)$$

Kde (X_{0i}, Y_{0i}) je poloha stredu oblúka určená podľa rovnice (3.22) a zároveň počiatok sústavy polárnych súradníc oblúka i . Vzdialenosť od počiatku sústavy súradníc r je zároveň polomerom

oblúka i . Uhol medzi spojnicou bodu a počiatku a osi Y , β je zároveň aktuálnym natočením vozidla ψ . Kladný smer uhla β uvažujeme proti smeru hodinových ručičiek. Grafické znázornenie aj s popisom je na obrázku nižšie (Obr. 5.3). Pre získanie karteziánskych súradníc koncového bodu oblúka K_i dosadíme do rovníc (5.1) a (5.2) hodnotu koncového uhla $\beta_{0i+1} = \beta_{0i} + \beta_i$.



Obr. 5.3 Popis oblúka trajektórie

Pre prvý oblúk bude platiť, že $K_{i-1} = K_0 = [X_p, Y_p] = [S, H]$ a $\beta_{01} = \psi_1$. Koncový bod a uhol oblúka i sú zároveň počiatkovým bodom a uhlom oblúka $(i + 1)$. Podobne ako pre prvý aj pre posledný chceme, aby platilo, že $K_i = K_3 = [X_{p^*}, Y_{p^*}] = [S^*, H^*]$ a $\beta_{0i+1} = \beta_{FINISH} = \psi_2$.

5.1.3 Cieľová funkcia

Cieľom optimalizačnej úlohy bude minimalizovať funkciu, ktorú nazveme cieľovou funkciou. V našom prípade, pri hľadaní trajektórie, je cieľom, aby dosiahnutý koncový bod bol totožný s požadovaným koncovým bodom určeným v zadaní úlohy. Preto cieľovú funkciu definujeme ako súčet štvorcov rozdielov jednotlivých parametrov koncového bodu.

$$CF = (X_{K3} - X_{p^*})^2 + (Y_{K3} - Y_{p^*})^2 + w_\beta (\beta_{FINISH} - \psi_2)^2, \quad (5.3)$$

kde w_β je váha (konštanta) vyjadrujúca, aký veľký význam má natočenie vozidla v koncovej polohe. Pri nižších hodnotách sa zvyšuje odchýlka od presného požadovaného uhla natočenia. V reálnej situácii nikdy nenastane absolútna zhoda, preto malá odchýlka nepredstavuje problém.

5.1.4 Optimalizačná metóda – FminCon

Výpočet v prostredí programu MATLAB bol realizovaný pomocou funkcie *fmincon* [12]. Je to nelineárny nástroj pre hľadanie lokálneho minima obmedzenej nelineárnej funkcie s viacerými premennými. Hľadá minimum funkcie:

$$\min_{par} f(par) \text{ pre ktorú platí } \begin{cases} c(par) \leq 0 \\ ceq(par) = 0 \\ A \cdot par \leq b \\ Aeq \cdot par = beq \\ lb \leq par \leq ub \end{cases} \quad (5.4)$$

Kde *par* je označenie vektora optimalizačných parametrov, *b*, *beq*, *lb* a *ub* sú vektory, *A* a *Aeq* sú matice, *c(par)* a *ceq(par)* sú funkcie vracajúce vektory a *f(par)* je funkcia vracajúca skalár. Funkcie *f(par)*, *c(par)* a *ceq(par)* môžu byť nelineárne funkcie parametrov.

Optimalizačnými parametrami pri hľadaní trajektórie CCC budú polomery *r* a uhly oblúkov β . Pri lokálnej optimalizačnej metóde budeme potrebovať počiatočný odhad parametrov *par*₀.

$$par_0 = [r_{10}, \beta_{10}, r_{20}, \beta_{20}, r_{30}, \beta_{30}] \quad (5.5)$$

V tejto chvíli máme všetko potrebné pre spustenie optimalizácie. Jej zápis v prostredí programu MATLAB je na obrázku nižšie (Obr. 5.4).

```
% fmincon
options = optimset('display','iter');

A = []; b = [];
Aeq = []; beq = [];
lb = []; ub = [];

Optim_Return =
fmincon(@Cost_Fun, par0, A, b, Aeq, beq, lb, ub, @podmienka, options);
```

Obr. 5.4 Zápis funkcie FminCon v prostredí programu MATLAB

V možnostiach (*options*) zmeníme hodnotu zobrazovania priebehu výpočtu. Ostatné hodnoty ponecháme prednastavené (default). Napríklad algoritmus výpočtu (*interior – point*), maximálny počet iterácií (*Maxiter* = 1000), veľkosť kroku zlepšovania cieľovej funkcie (*TolX* = 10^{-6}), maximálny počet volania cieľovej funkcie (*MaxFunEvals* = 3000). Funkcia *podmienka* v sebe obsahuje funkcie *c(par)* a *ceq(par)* a viac si o nej povieme v odseku nižšie.

Podmienka pre *FminCon*

Aby sme zohľadnili vlastnosti vozidla a neprekročili maximálny možný uhol natočenia kolies a tým minimálny možný rádius oblúku, pridáme do funkcie *fmincon* podmienku. Budeme uvažovať polomer v kladnom aj zápornom smere, ako sme opísali v odseku 3.2.1. Preto nebudeme môcť túto podmienku zapísať pomocou lineárnej funkcie. Ďalej budeme kvôli zrýchleniu času výpočtu uvažovať maximálny polomer. Pri veľkom polomere sa z oblúka stáva približne priamka a z toho dôvodu je možné veľmi vysoké polomery zanedbať. Pri jazde po kružnici budeme uvažovať kratšiu možnú trajektóriu, a preto posledná podmienka obmedzí uhol oblúka β na interval $(-\pi, \pi)$.

Na obrázku vyššie (Obr. 5.4) je znázornené, že *A*, *b*, *Aeq*, *beq*, *lb* a *ub* sú prázdne množiny. Znamená to, že sme nepoužili ani jednu lineárnu podmienku, ale z dôvodu syntaxe kódu sa museli všetky uviesť. Podmienky potrebné pre našu úlohu sme zapísali do funkcie *podmienka* ako kvadratické nerovnice.

$$c_1 = R_{min}^2 - r_i^2 \leq 0 \quad (5.6)$$

$$c_2 = r_i^2 - R_{max}^2 \leq 0 \quad (5.7)$$

$$c_3 = \beta_i^2 - \pi^2 \leq 0 \quad (5.8)$$

Všetky nelineárne podmienky sme zapísali ako nerovnice a z toho dôvodu musíme definovať *ceq(par)* taktiež ako prázdnu množinu.

5.1.5 Simulácia kinematického modelu

Vstupy pre simuláciu

Podobne ako pri predchádzajúcich úlohách budeme pre simuláciu potrebovať čas, uhol natočenia kolies a smer rýchlosti pre každý úsek trajektórie.

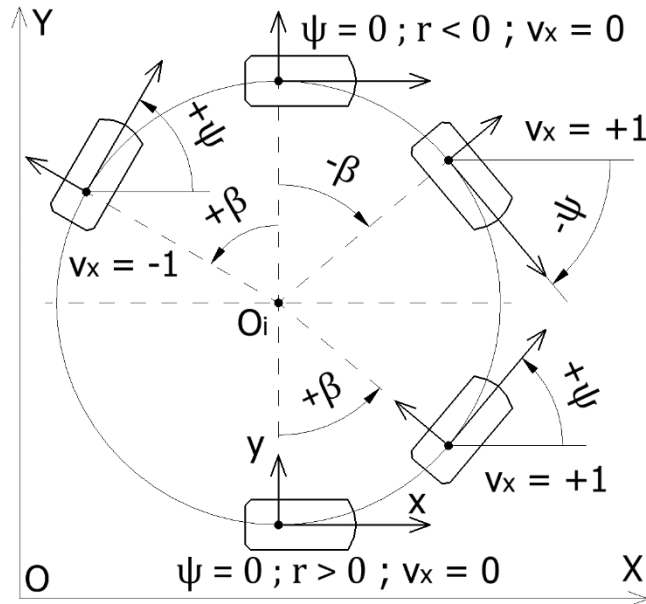
Výpočet dráhy sa nám zjednoduší, pretože nemusíme dopočítavať uhol oblúka ako v minulých úlohách. Pri použití rovnice (3.12) vznikne malá zmena z dôvodu možnej zápornej hodnoty *r*. Dráhu vypočítame ako absolútnu hodnotu.

$$S_{ci} = |r_i \cdot \beta_i| \quad (5.9)$$

Potom čas pre každý oblúk dostaneme z rovnice (3.18).

Uhol natočenia kolies φ určíme z rovnice (3.20).

Veľkosť rýchlosti budeme aj naďalej považovať za konštantu $|v_x| = 1 \text{ ms}^{-1}$. Určenie smeru sa v tomto prípade výrazne zjednoduší, pretože okamžitá hodnota uhlu β je rovná okamžitej hodnote uhlu natočenia vozidla ψ . Na obrázku nižšie (Obr. 5.5) sú znázornené dva prípady pohybu vozidla.

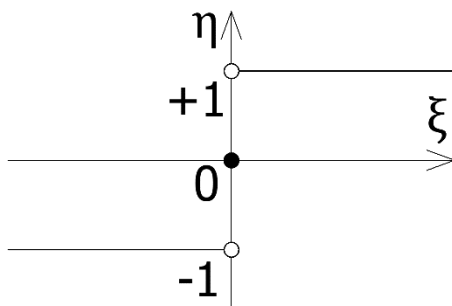


Obr. 5.5 Ukážka rôznych smerov pohybu

Všeobecne určíme smer rýchlosti zo vzorca:

$$v_{xi} = \text{sign}(r_i) \cdot \text{sign}(\beta_i) \cdot |v_x| \quad (5.10)$$

Kde *sign* je funkcia signum, zobrazenie z množiny reálnych čísel do množiny reálnych čísel, ktoré z daného čísla (r a β) extrahuje iba jeho znamienko podľa predpisu:



$$\text{sign}(\xi) = \begin{cases} -1 & \text{ak } \xi < 0 \\ 0 & \text{ak } \xi = 0 \\ 1 & \text{ak } \xi > 0 \end{cases}$$

Obr. 5.6 Funkcia signum

Teraz máme všetky potrebné vstupy pre simuláciu.

Výsledok simulácie kinematického modelu

Na obrázku nižšie (Obr. 5.7) sú zobrazené výsledky simulácie kinematického modelu vozidla, pohybujúceho sa po vypočítanej trajektórii pomocou lokálnej optimalizačnej metódy, splňujúcej počiatkové podmienky, a bez kolízie s prostredím. Zodpovedajú zadaným typom úloh zobrazeným na začiatku tejto kapitoly (Obr. 5.1 a Obr. 5.2). Počiatkové polohy a parametre prostredia sú uvedené v tabuľke nižšie (Tab. 5.1).

Parameter	Jednotka	Hodnota		Popis
		a)	b)	
$S; H$	[m]	12.5; 5.4	14; 11.4	Počiatková poloha $X; Y$
ψ_1	[°]	-10	0	Počiatkové natočenie vozidla
$S^*; H^*$	[m]	3.73; 2.4	6.5; 2.47	Koncová poloha $X; Y$
ψ_2	[°]	0	90	Koncové natočenie vozidla
L_{park}	[m]	7.3	4.5	Dĺžka parkovacieho miesta
W_{park}	[m]	2.4	3	Šírka parkovacieho miesta
W_{road}	[m]	4	7.5	Šírka cesty

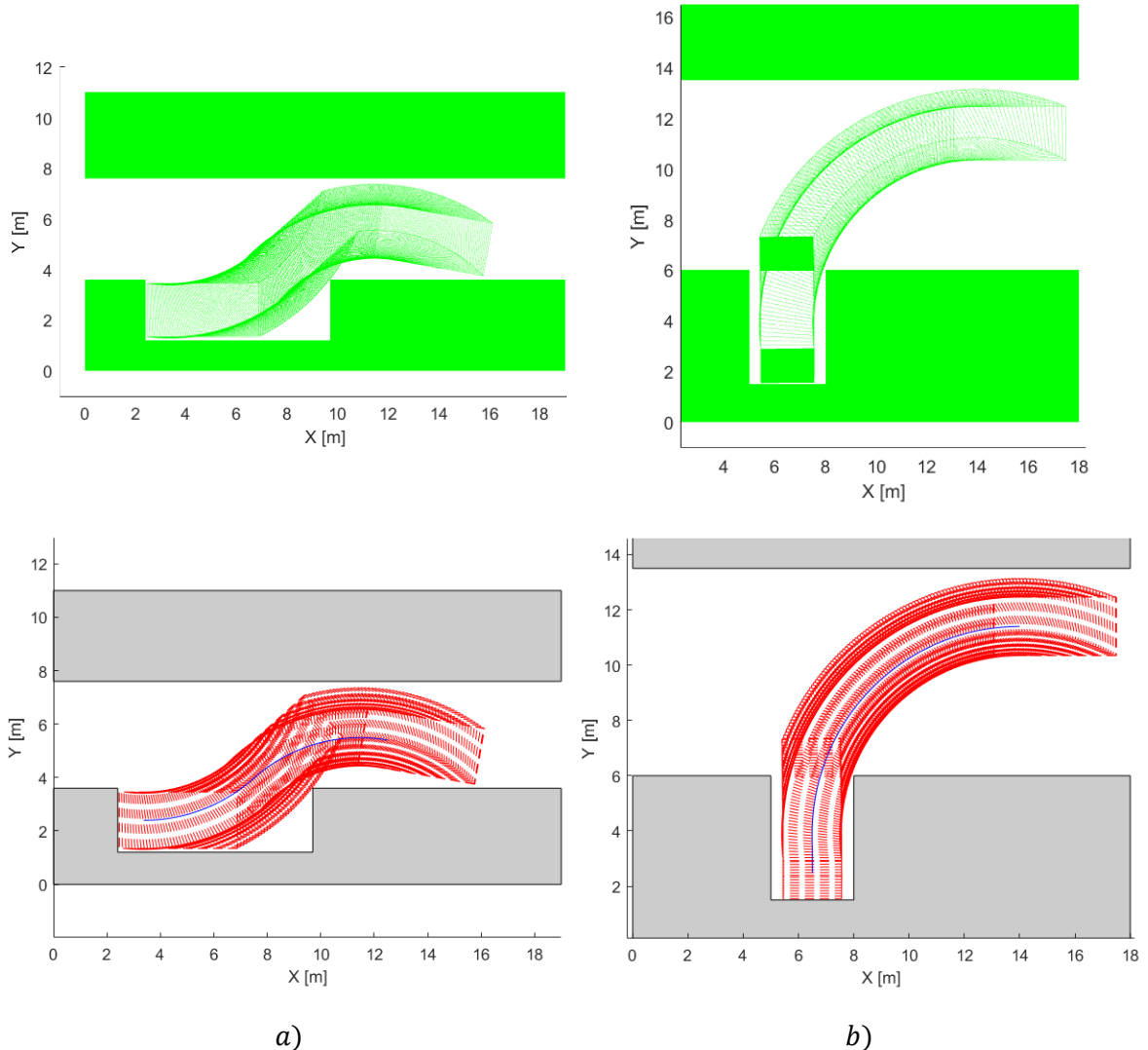
Tab. 5.1 Počiatkové polohy a parametre prostredia

Výsledné optimalizačné parametre sú uvedené v tabuľke nižšie (Tab. 5.2).

Parameter	Jednotka	Hodnota		Popis
		a)	b)	
r_1	[m]	-5.95	-7.51	Polomer oblúka 1
β_1	[rad]	0.90	1.58	Uhol oblúka 1
r_2	[m]	140.21	156.29	Polomer oblúka 2
β_2	[rad]	-5.7e-6	-3.6e-3	Uhol oblúka 2
r_3	[m]	6.19	120.63	Polomer oblúka 3
β_3	[rad]	-0.73	-6.5e-3	Uhol oblúka 3

Tab. 5.2 Výsledné optimalizačné parametre Lokálnej metódy

Červenou farbou je vykreslená obálka obdĺžnika, znázorňujúca priestor vozidla. Pre vyššiu prehľadnosť obrázka nie je zobrazený pôdorys vozidla. Modrou farbou je vykreslená trajektória určená z algoritmu. Danú trajektóriu opisuje stred zadnej nápravy vozidla. Sivou farbou sú znázornené prekážky tvorené typom prostredia.



Obr. 5.7 Mapa prostredia a obálka vozidla opisujúceho trajektóriu, lokálna metóda

5.1.6 Zhodnotenie lokálnej optimalizačnej metódy

Úlohou bolo nájsť trajektóriu vozidla, z počiatočnej pozície do koncovej pozície bez kolízie s prostredím, pomocou lokálnej optimalizačnej metódy. Podarilo sa nájsť riešenie pre obe zvolené úlohy. Úspešne sme overili funkčnosť systému aj na iných prípadoch.

Ukázalo sa, že zvolená numerická metóda je veľmi podobná ako predchádzajúca analytická. Systém aj naďalej zostáva veľmi citlivý na zvolené počiatočné podmienky a počiatočný odhad parametrov.

5.2 Globálna optimalizačná metóda

Pri hľadaní trajektórie pomocou globálnej optimalizačnej metódy budeme uvažovať rovnaké zadanie úlohy ako v predchádzajúcom odseku. Zadanie optimalizačnej úlohy pre pozdĺžne parkovanie je znázornené na obrázku (Obr. 5.1) a pre kolmé parkovanie na obrázku (Obr. 5.2), ktoré sú popísané v odseku 5.1.1.

Popis trajektórie taktiež zostáva totožný ako pri lokálnej metóde a je popísaný v odseku 5.1.2. Jediná zmena nastáva v počte oblúkov, ktorý je možné ľubovoľne zmeniť.

5.2.1 Optimalizačné parametre

Pre zjednodušenie určovania podmienok pre optimalizačné parametre boli zavedené nové parametre p pre polomery oblúkov a p_β pre uhly oblúkov v intervale $\langle 0; 1 \rangle$. Z dôvodu nespojitosti intervalu pre polomer ($r \in \langle -r_{max}; -r_{min} \rangle \cup \langle r_{min}; r_{max} \rangle$) budeme polomer počítať zo spojitého intervalu natočenia kolies ($\varphi \in \langle -\varphi_{max}; \varphi_{max} \rangle$). Pre uhol oblúku platí rovnaká podmienka ako pri lokálnej metóde ($\beta \in \langle -\pi; \pi \rangle$). Pre hodnoty polomerov a uhlov oblúkov dostávame parametrické rovnice v tvare:

$$r = \frac{l}{\tan(\varphi_{max} \cdot \cos(p \cdot \pi))} \quad (5.11)$$

$$\beta = \pi(2p_\beta - 1) \quad (5.12)$$

V rovnici (5.11) člen $\varphi_{max} \cdot \cos(p \cdot \pi)$ prostredníctvom harmonickej funkcie kosínus interpretuje spojité interval uhlu natočenia kolies, keďže funkcia kosínus na intervale $\langle 0; \pi \rangle$ nadobúda hodnoty z intervalu $\langle -1; 1 \rangle$.

5.2.2 Cieľová funkcia

Cieľom optimalizačnej úlohy bude opäť minimalizovať cieľovú funkciu. V našom prípade, pri hľadaní trajektórie, je cieľom, aby dosiahnutý koncový bod bol totožný s požadovaným koncovým bodom určeným v zadaní úlohy. Pri globálnej metóde ju rozšírime o dĺžku trajektórie a hodnotu *penalty*, ktorá bude reprezentovať prítomnosť kolízie. Cieľovú funkciu definujeme ako:

$$CF = w_X(X_{K3} - X_{P*})^2 + w_Y(Y_{K3} - Y_{P*})^2 + w_\beta(\beta_{FINISH} - \psi_2)^2 + w_S S + penalty, \quad (5.13)$$

kde w sú váhy (konštanty) vyjadrujúce význam príslušnej hodnoty.

Penalizácia

K penalizácii cieľovej funkcie dochádza v prípade kolízie. Vyhodnotenie kolízie bolo opísané v kapitole 4. Hodnotu premennej veličiny *penalty* určíme zo vzťahu:

$$penalty = \begin{cases} 0 & ak \max(E_k) = 1 \\ \sum_{k=1}^e (E_k = 2) & ak \max(E_k) = 2 \end{cases} \quad (5.14)$$

Kde k je index elementu, e počet elementov siete a E_k je hodnota elementu k .

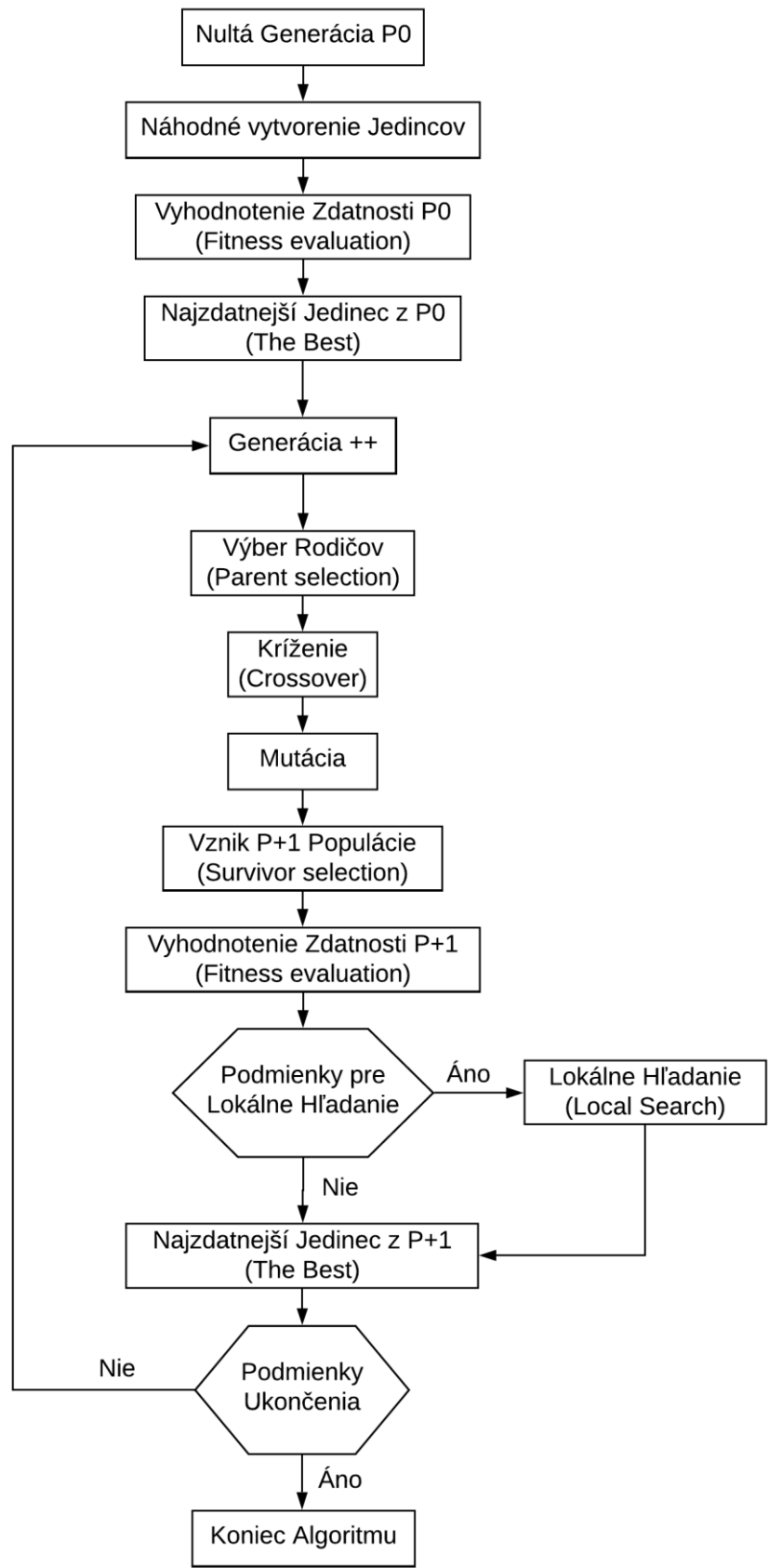
Z rovnice (5.14) vyplýva, že ak ku kolízii nedochádza, hodnota *penalty* je nulová, ale ak ku kolízii dochádza, hodnota *penalty* je rovná súčtu všetkých elementov s hodnotou 2, čiže všetkých elementov, v ktorých ku kolízii došlo.

Pre urýchlenie výpočtu vyhodnocovania cieľovej funkcie a eliminovanie nezmyselných trajektórií zavedieme kružnicu s vhodne zvoleným stredom a polomerom tak, aby prostredie potrebné na manévrovanie ležalo vo vnútri tejto kružnice. Každá trajektória s aspoň jedným bodom mimo tejto kružnice bude výrazne penalizovaná.

5.2.3 Optimalizačná metóda – genetický algoritmus

Genetické algoritmy (GA) sa objavili v roku 1975 [13] a odvtedy boli použité pri mnohých úspešných riešeniach optimalizačných úloh. Genetické algoritmy sú stochastické vyhľadávacie techniky analogické k prirodzenej evolúcii, založenej na princípe prežitia najzdatnejšieho jedinca [14]. Potenciálne riešenie úlohy je zapísané v génoch jedinca tvoriacich populáciu. Každý jedinec je ohodnotený veľkosťou cieľovej funkcie, ktorú nazveme zdatnosť (*fitness*). Mechanizmus výberu, založený na hodnote zdatnosti, aplikujeme na populáciu a jednotlivci sa snažia o prežitie. Čím je jedinec zdatnejší, tým má väčšiu šancu stať sa rodičom a jeho genetická informácia sa v podobe potomkov prenesie do novej generácie. Potomstvo vzniká rôznymi druhmi kríženia a mutácií. Tento proces sa opakuje a populácia sa vyvíja generácia po generácii. Po niekoľkých generáciách začne populácia konvergovať k riešeniu a najlepší jedinec má šancu stať sa požadovaným riešením [8].

Základný algoritmus použitý na výpočet je zobrazený na obrázku nižšie (Obr. 5.8). Jednotlivé kroky budú postupne vysvetlené v nasledujúcom texte.



Obr. 5.8 Diagram použitého genetického algoritmu

Nultá generácia

Jedinec nesúci genetickú informáciu je riadkový vektor čísel. Prvé číslo vyjadruje vek jedinca, druhé jeho zdatnosť, tretie veľkosť penalizácie a zvyšok sú chromozómy reprezentujúce optimalizačné parametre trajektórie. Zápis vyzerá nasledovne:

$$\text{Gén} : [\text{age} , \text{fitness} , \text{penalty} , p_1 , p_{\beta 1} , \dots , p_n , p_{\beta n}], \quad (5.15)$$

kde n je počet oblúkov trajektórie.

Vytvorenie nulte generácie je čisto náhodný proces, pri ktorom sa náhodne vygenerujú všetky optimalizačné parametre jedinca ako náhodné číslo v intervale $(0,1)$. Počet jedincov v generácii je možné ľubovoľne zmeniť na začiatku algoritmu.

Vyhodnotenie zdatnosti

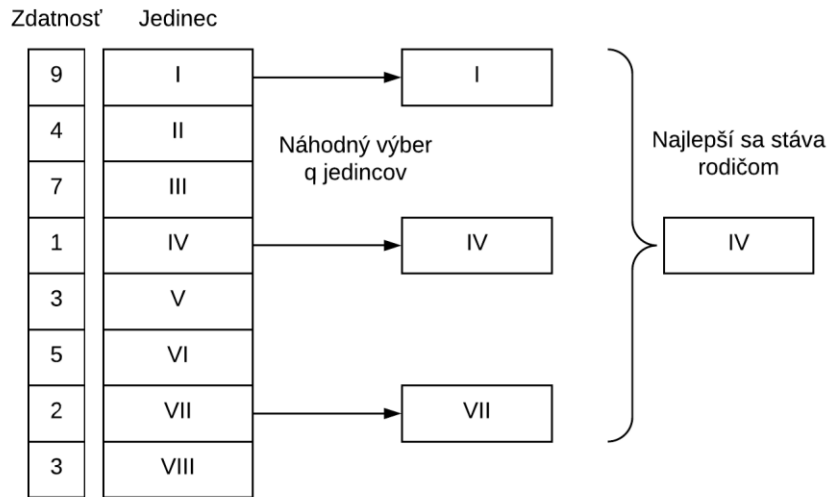
Pre každého jedinca sa spočíta hodnota cieľovej funkcie podľa rovnice (5.13) a zapíše sa do druhého stĺpca vektoru reprezentujúceho tohto jedinca. Do tretieho stĺpca sa zapíše hodnota penalizácie.

Najzdatnejší jedinec

Hľadáme minimum cieľovej funkcie, a preto za najzdatnejšieho jedinca zvolíme jedinca s najmenšou hodnotou zdatnosti. Gén tohto jedinca sa zapíše do premennej *The Best* a predstavuje doposiaľ najlepšie riešenie.

Výber rodičov

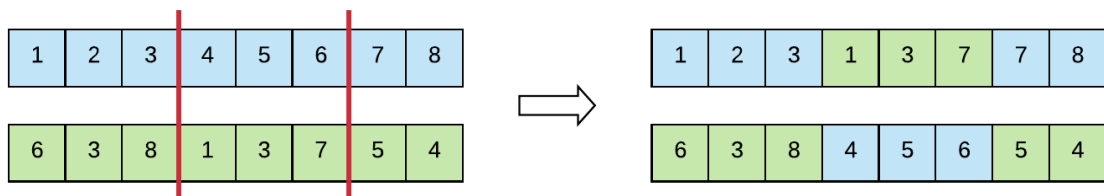
Výber rodičov je proces, pri ktorom náhodne vyberieme dvoch jedincov z populácie za rodičov. Tento proces je možné vykonať rôznymi spôsobmi. Pre našu úlohu bol použitý turnajový typ výberu rodičov (Obr. 5.9). Z populácie sa náhodne vyberie q jedincov, pričom pravdepodobnosť výberu je pre všetkých rovnaká. Jedinec s najlepšou (najnižšou) zdatnosťou sa vyberie za rodiča, ale nezmažeme ho, tak je možné, že sa stane rodičom opakovane.



Obr. 5.9 GA - Turnajový výber rodičov

Kríženie

Jedinci, ktorí boli vybraní za rodičov, sa skrížia, čím si vymenia jednotlivé chromozómy a vzniknú noví potomkovia. Táto operácia sa snaží zjednodušiť reprodukciu a biologický proces kríženia (Obr. 5.10). Tento proces je možné vykonať rôznymi spôsobmi. Pre našu úlohu bol použitý viacbodový typ kríženia. Náhodne sa vyberú dve čísla z intervalu $\langle 1; 2n - 1 \rangle$, kde $2n$ je počet chromozómov. Jedinci si vymenia časť genetického reťazca medzi vybranými číslami.



Obr. 5.10 GA – viacbodové kríženie

Mutácia

U každého nového jedinca sa môže vyskytnúť zmutovaný chromozóm. Podľa zvolenej hladiny pravdepodobnosti sa každý chromozóm môže zmeniť na čisto náhodné číslo z intervalu pre optimalizačné parametre. Tento proces má v algoritme za úlohu udržiavať rôznorodosť jedincov a zabráňovať tak degradácii populácie, kedy sa všetci jedinci stanú totožnými.

Vznik novej populácie

Výber jedincov do novej populácie je možné vykonať rôznymi spôsobmi. Pre našu úlohu bol použitý typ výberu, kde jedna štvrtina novej generácie je tvorená najzdatnejšími jedincami spomedzi pôvodných jedincov a potomstva a zvyšné tri štvrtiny generácie sú náhodne vybraní

jedinci zo zvyšku. Celkový počet jedincov pritom nesmie presiahnuť veľkosť populácie zvolenej na začiatku algoritmu.

Lokálne hľadanie

Lokálne hľadanie prispieva k zlepšeniu riešenia. Ide o použitie lokálnej optimalizačnej metódy FminCon, ktorej princíp bol popísaný v odseku 5.1.4. Pre zrýchlenie výpočtu sa lokálne hľadanie spustí len za určitých podmienok. Prvou podmienkou je dosiahnutie určitého počtu generácií, kedy už je populácia zbavená nezmyselných riešení. Druhou podmienkou je, že najlepší jedinec je iný ako v predchádzajúcej generácii. Za predpokladu, že najlepší jedinec ostal nezmenený aj pre novú generáciu, nie je potrebné znovu spúšťať lokálne hľadanie.

Podmienky ukončenia algoritmu

Pri hľadaní trajektórie je dôležité, aby nedochádzalo ku kolízii, a preto najdôležitejšia podmienka ukončenia je nulová penalizácia. Za predpokladu splnenia tejto podmienky a nezmenenia najlepšieho jedinca po určitý počet generácií môžeme algoritmus ukončiť.

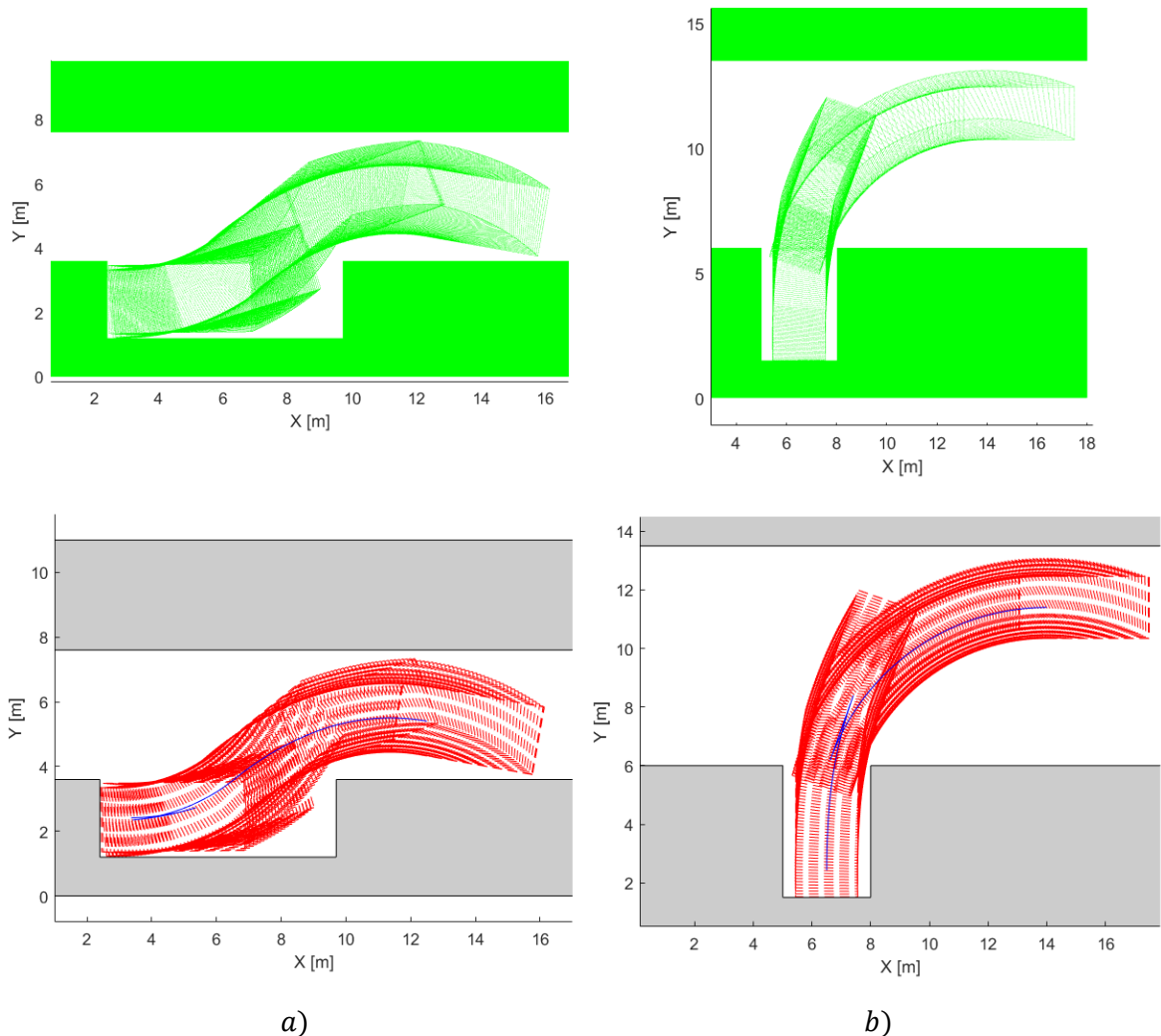
5.2.4 Simulácia kinematického modelu

Globálna optimalizačná metóda bola overená na množstve úloh. Pre názornosť sú uvedené výsledky zadaných úloh pre lokálnu optimalizačnú metódu zobrazené na začiatku tejto kapitoly (Obr. 5.1 a Obr. 5.2) a tabuľka (Tab. 5.1) s počiatočnými polohami a parametrami prostredia je uvedená v odseku 5.1.5. Výsledné optimalizačné parametre sú uvedené v tabuľke nižšie.

Trajektória	a)		b)	
Parameter	r	β	r	β
Jednotka	[m]	[rad]	[m]	[rad]
1	-6.50	0.53	-7.82	1.23
2	-30.02	-4.5e-3	-390.38	-6.0e-3
3	-9.26	0.33	-14.68	0.27
4	5.85	-0.59	-27.94	7.5e-2
5	9.48	0.21	777.22	0
6	7.14	-0.28	-2495.44	-1.8e-5

Tab. 5.3 Výsledné optimalizačné parametre globálnej metódy

Červenou farbou je vykreslená obálka obdĺžnika, znázorňujúca priestor vozidla. Pre vyššiu prehľadnosť obrázka nie je zobrazený pôdorys vozidla. Modrou farbou je ďalej vykreslená trajektória určená z algoritmu. Danú trajektóriu opisuje stred zadnej nápravy vozidla. Sivou farbou sú znázornené prekážky tvorené typom prostredia.



Obr. 5.11 Mapa prostredia a obálka vozidla opisujúceho trajektóriu, lokálna metóda

5.2.5 Zhodnotenie globálnej optimalizačnej metódy

Úlohou bolo nájsť trajektóriu vozidla, z počiatočnej pozície do koncovej pozície bez kolízie s prostredím, pomocou globálnej optimalizačnej metódy. Podarilo sa nájsť riešenie pre obe zvolené úlohy. Úspešne sme overili funkčnosť systému aj na iných prípadoch.

Ukázalo sa, že zvolená numerická metóda je robustnejšia ako predchádzajúce algoritmy. Systém je menej citlivý na zvolené počiatočné podmienky a nevyžaduje žiadny počiatočný odhad parametrov. Nevýhodou globálnych optimalizačných úloh je ich stochastický a heuristický charakter. Z toho vyplýva nejednoznačnosť nájdenia optimálneho riešenia pri opakovaní výpočtu.

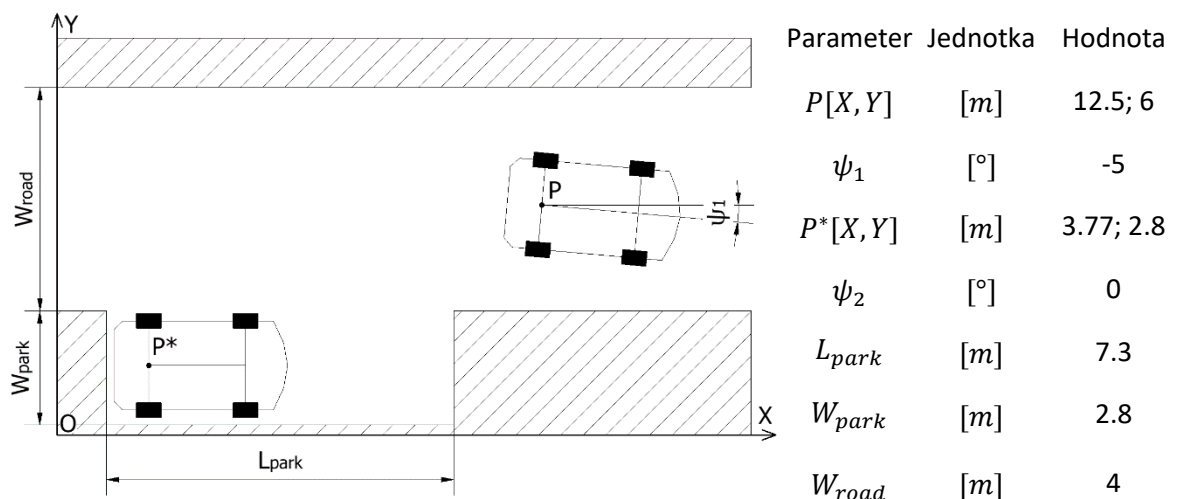
6 Simulačný experiment

V tejto kapitole uvedieme niekoľko prípadov parkovacích manévrov a porovnáme jednotlivé metódy pri ich vyhodnocovaní.

6.1 Pozdĺžne parkovanie

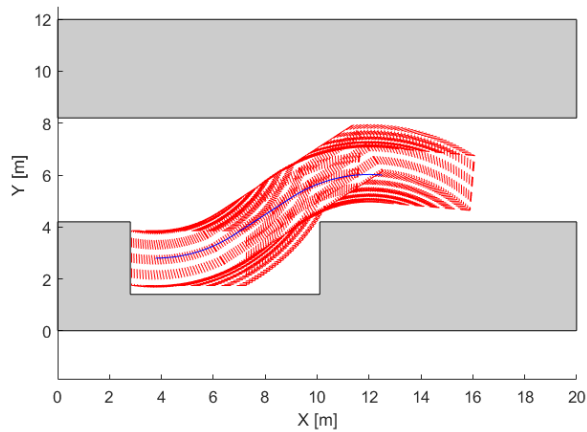
6.1.1 Úloha č.1: Vozidlo stojacie za parkovacím miestom

Táto úloha je najjednoduchšia spomedzi prípadov pozdĺžneho parkovania, pretože vozidlo sa nachádza za parkovacím miestom, a tak môže hneď začať parkovací manéver. Pri tejto úlohe uvažujeme veľký priestor na parkovanie a koncová pozícia je na konci parkovacieho miesta. Na obrázku nižšie (Obr. 6.1) je znázornené zadanie úlohy s počiatočnou a koncovou polohou vozidla a rozmermi prostredia.

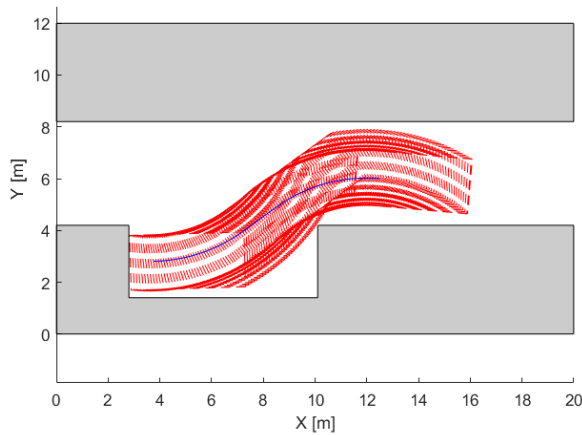


Obr. 6.1 Úloha č.1: Vozidlo stojacie za parkovacím miestom

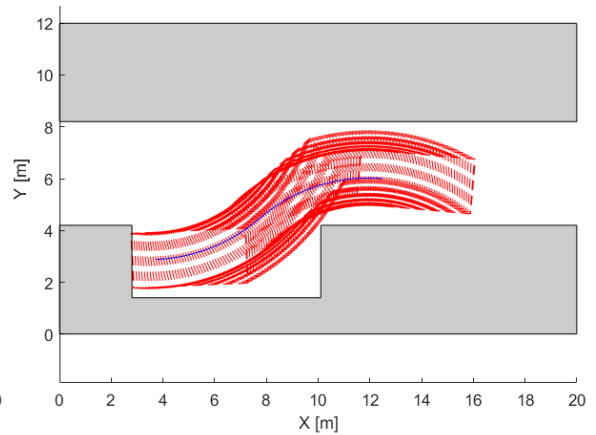
Úloha bola postupne riešená analyticky (odsek 3.2), globálnou optimalizačnou metódou (odsek 5.2) a jej riešenie bolo použité ako odhad pre lokálnu optimalizačnú metódu (odsek 5.1).



a) Analytické riešenie CSC trajektórie



b) Globálna optimalizačná metóda



c) Lokálna optimalizačná metóda

Obr. 6.2 Riešenie úlohy č.1: Obálka vozidla opisujúceho trajektórie

Výsledné hodnoty trajektórií popisuje tabuľka nižšie:

Parameter	Jednotka	Hodnota			Popis
		a)	b)	c)	
r_1	[m]	-5.54	-5.97	-6.44	Polomer oblúka 1
β_1	[rad]		0.72	0.81	Uhol oblúka 1
r_2	[m]	5.54	-6614.13	-15.9	Polomer oblúka 2
β_2	[rad]		2.3e-4	-4.2e-3	Uhol oblúka 2
r_3	[m]		6.01	6.57	Polomer oblúka 3
β_3	[rad]		-0.61	-0.67	Uhol oblúka 3
S_0	[m]	9.50	9.52	9.65	Dĺžka trajektórie

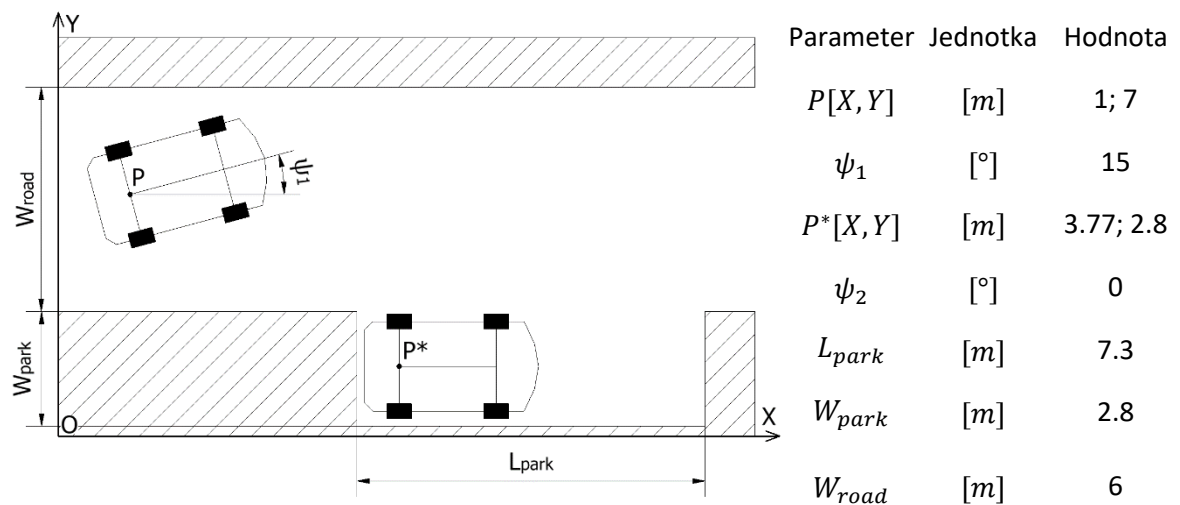
Tab. 6.1 Riešenie úlohy č.1: Výsledné parametre trajektórií

Zhodnotenie výsledkov

Tento jednoduchý manéver sme dokázali vypočítať aj analyticky, aj numericky. Dosiahnuté výsledky sú porovnateľné. Zásadný rozdiel je vo výpočtovom čase, kedy analytické riešenie dokázalo nájsť riešenie približne za 0.53 sekundy, lokálna optimalizačná metóda za približne 0.76 sekundy (s dobrým odhadom parametrov) a globálnej optimalizačnej metóde výpočet trval približne 7.28 sekundy, pričom medzi jedincov bol umiestnený odhad parametrov použitý pre lokálnu metódu.

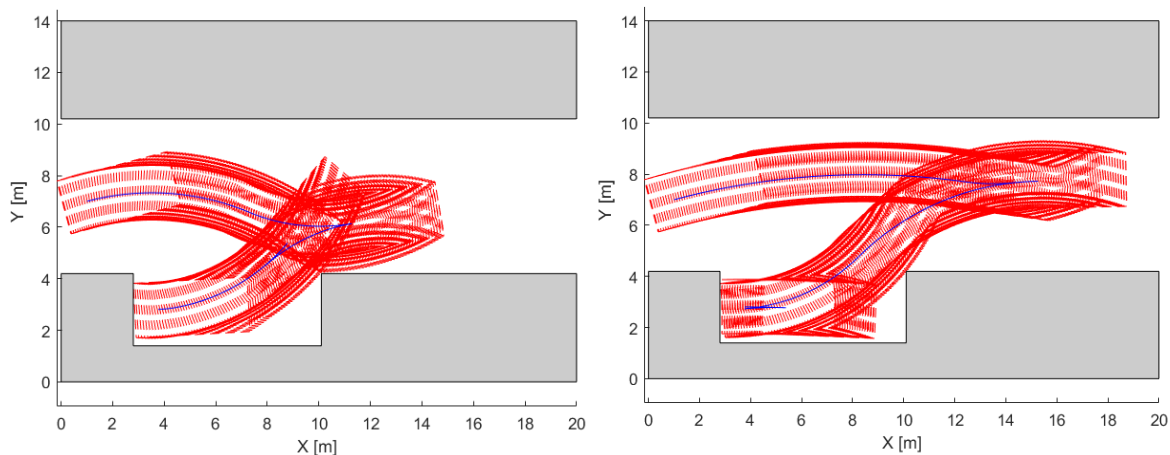
6.1.2 Úloha č.2: Vozidlo stojacie pred parkovacím miestom

Táto úloha je zložitejšia, pretože sa vozidlo musí dostať do ideálnej východiskovej polohy pred začatím parkovacieho manévru z prvej úlohy. Pri tejto úlohe uvažujeme rovnako veľký priestor na parkovanie ako v predchádzajúcej úlohe a koncová pozícia je taktiež na konci parkovacieho miesta. Na obrázku nižšie (Obr. 6.3) je znázornené zadanie úlohy s počiatočnou a koncovou polohou vozidla a rozmermi prostredia.



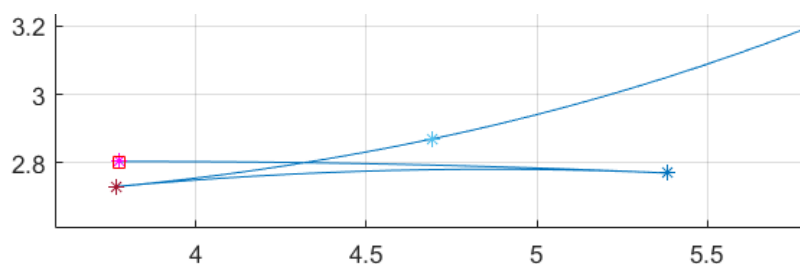
Obr. 6.3 Úloha č.2: Vozidlo stojacie pred parkovacím miestom

Analytická metóda nebola schopná zadanú úlohu vyriešiť pomocou trajektórie typu CSC. Nájsť vhodný odhad parametrov pre lokálnu optimalizačnú metódu je veľmi náročné a keďže je zakomponovaná v globálnej metóde, samostatne nebola použitá. Trajektórie sa skladajú z ôsmich oblúkov.



a) Bez odhadu parametrov

b) S odhadom parametrov



a) Detail koncovej fázy manévru b)

Obr. 6.4 Riešenie úlohy č.2: Obálka vozidla opisujúceho trajektórie

Výsledné hodnoty trajektórií popisuje tabuľka nižšie:

Trajektória	c) Bez odhadu parametrov		d) S odhadom parametrov	
Parameter	r	β	r	β
Jednotka	[m]	[rad]	[m]	[rad]
1	-9.78	-0.66	-28.70	-0.40
2	6.93	0.26	8.06	0.24
3	7.36	0.30	-11.15	-0.08
4	-6.56	0.13	-10.24	0.76
5	-6.87	0.42	7.12	-0.57
6	8.31	0.14	8.25	-0.11
7	7.43	-0.67	-11.94	-0.14
8	5.57	-0.14	-35.94	0.05
Dĺžka trajektórie	21.22		30.42	

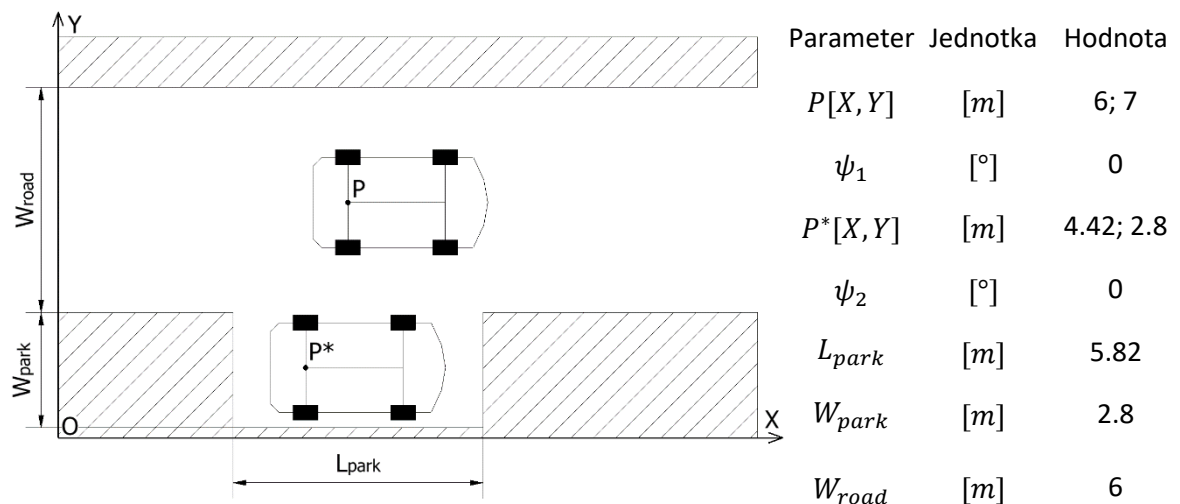
Tab. 6.2 Riešenie úlohy č.2: Výsledné parametre trajektórií

Zhodnotenie výsledkov

Globálna optimalizačná metóda našla riešenie bez odhadu parametrov približne za 18 sekúnd a následne s odhadom pridaným do počítačovej populácie výpočet zabil približne 8 sekúnd.

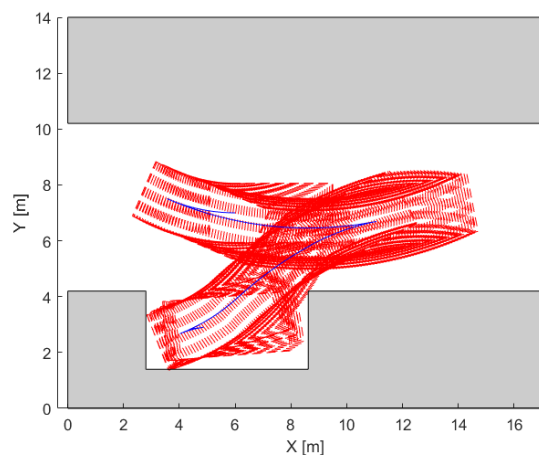
6.1.3 Úloha č.3: Menšie parkovacie miesto

Táto úloha je najzložitejšia, pretože parkovacie miesto je výrazne skrátene a požadovaný koncový bod sa už naďalej nenachádza na konci, ale v strede parkovacieho miesta tak, aby po dokončení zostalo miesto z oboch strán. Uvažovaný priestor je o 1.6 metra dlhší ako vozidlo. Na obrázku nižšie (Obr. 6.5) je znázornené zadanie úlohy s počiatočnou a koncovou polohou vozidla a rozmermi prostredia.

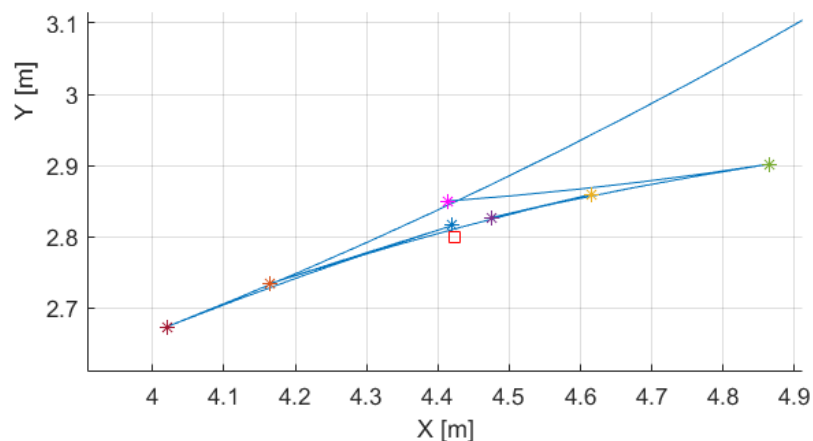


Obr. 6.5 Úloha č.3 Menšie parkovacie miesto

Pre riešenie bola použitá globálna optimalizačná metóda obsahujúca lokálnu optimalizačnú metódu na zlepšenie dosiahnutých riešení. Trajektória je popísaná dvanástimi oblúkmi.



Obr. 6.6 Riešenie úlohy č.2: Obálka vozidla opisujúceho trajektóriu



Obr. 6.7 Detail koncovej fázy manévru riešiacej úlohu č. 3

Výsledné hodnoty trajektórií popisuje tabuľka nižšie:

Parameter	r	β	Parameter	r	β
Jednotka	[m]	[rad]	Jednotka	[m]	[rad]
1	6.14	-0.41	7	-5.97	-0.07
2	12.74	0.60	8	-42.08	6.4e-3
3	-8.20	0.28	9	-5.79	-0.08
4	5.69	0.01	10	26.31	-5.4e-3
5	-13.82	0.25	11	-5.60	-0.7
6	6.79	-0.36	12	5.67	-0.8
Dĺžka trajektórie			20.65		
Tab. 6.3 Riešenie úlohy č.3: Výsledné parametre trajektórie					

Zhodnotenie výsledkov

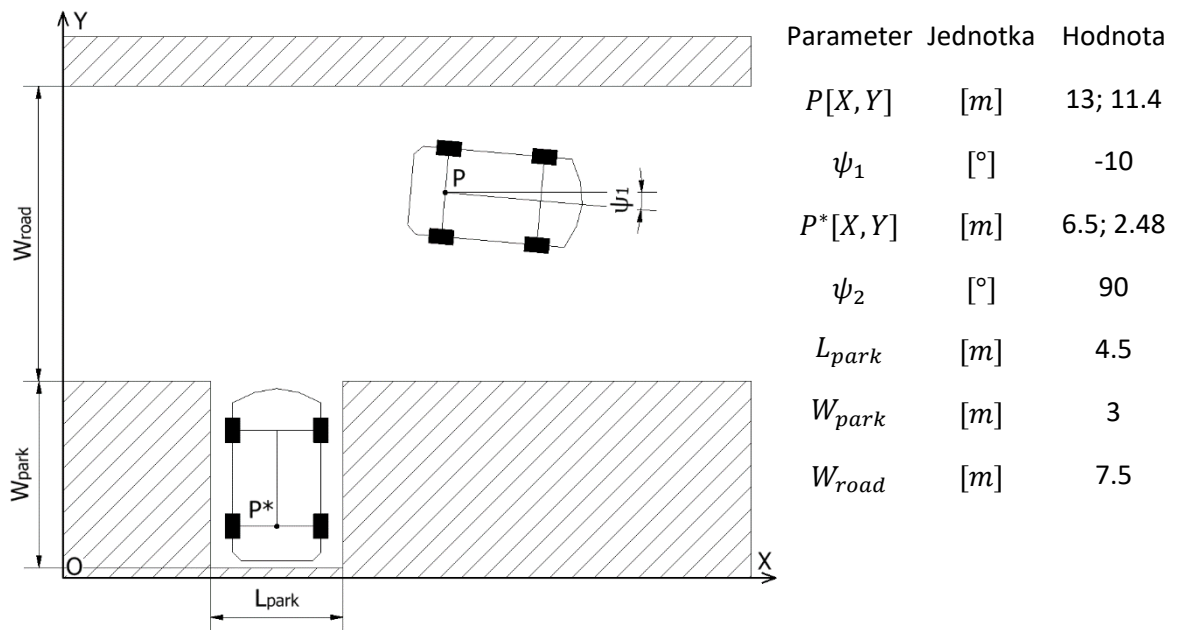
Po použití odhadov parametrov získaných opakovaným riešením úlohy sa úlohu podarilo riešiť s presnejšími výsledkami ako bez odhadov. Výpočet riešenia bez použitia odhadov trval približne 40 sekúnd, zatiaľ čo pri použití odhadov sa skrátil na približne 5 sekúnd.

Výsledná trajektória sa skladá až z dvanástich oblúkov, pretože riešenia pri menšom počte nedosahovali požadovanej presnosti.

6.2 Kolmé parkovanie

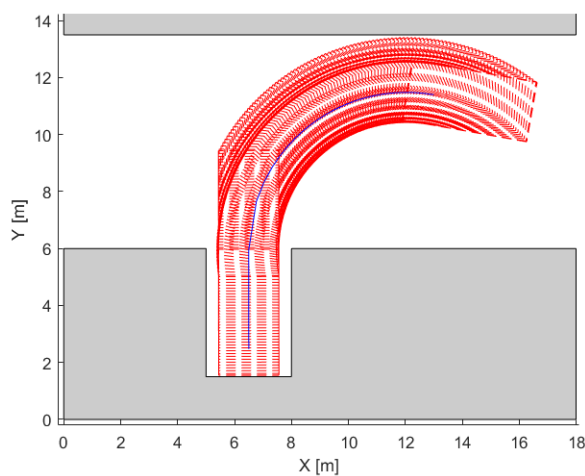
6.2.1 Úloha č.4: Vozidlo stojacie za parkovacím miestom

Táto úloha je najjednoduchšia spomedzi prípadov kolmého parkovania, pretože vozidlo sa nachádza za parkovacím miestom, a tak môže hneď začať parkovací manéver. Pri tejto úlohe uvažujeme veľký priestor na parkovanie. Na obrázku nižšie (Obr. 6.8) je znázornené zadanie úlohy s počiatočnou a koncovou polohou vozidla a rozmermi prostredia.

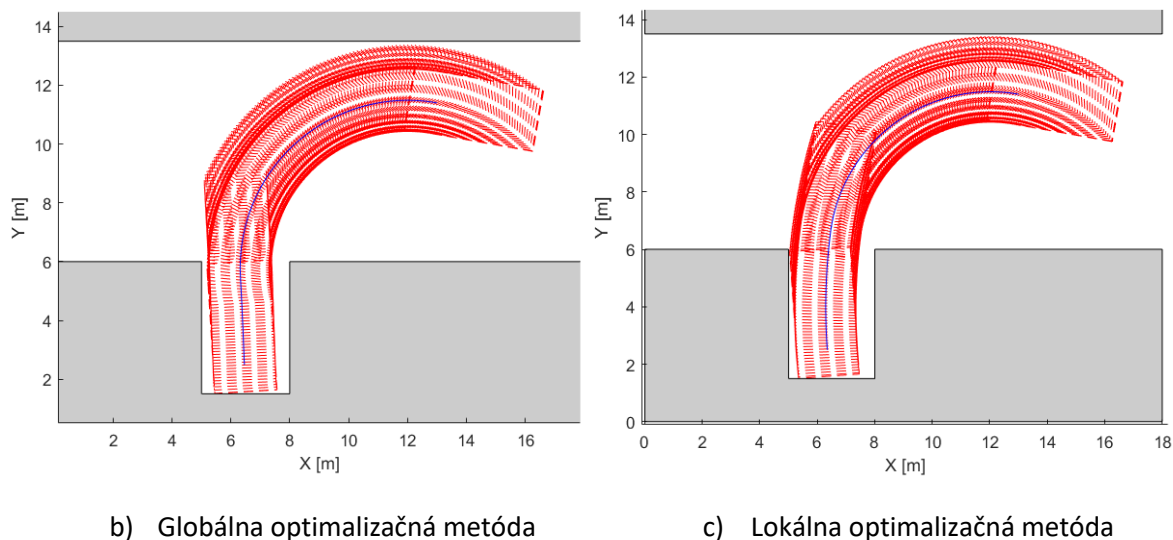


Obr. 6.8 Úloha č.4: Vozidlo stojacie za parkovacím miestom

Úloha, rovnako ako úloha č.1, bola postupne riešená analyticky, globálnou optimalizačnou metódou a jej riešenie bolo použité ako odhad pre lokálnu optimalizačnú metódu.



a) Analytické riešenie CSC trajektórie



Obr. 6.9 Riešenie úlohy č.4: Obálka vozidla opisujúceho trajektórie

Výsledné hodnoty trajektórií popisuje tabuľka nižšie:

Parameter	Jednotka	Hodnota			Popis
		a)	b)	c)	
r_1	[m]	-5.54	-5.70	-5.61	Polomer oblúka 1
β_1	[rad]		1.80	1.60	Uhol oblúka 1
r_2	[m]	0	147.41	165.69	Polomer oblúka 2
β_2	[rad]		-0.02	9.5e-4	Uhol oblúka 2
r_3	[m]		-9	-20.24	Polomer oblúka 3
β_3	[rad]		0.03	0.21	Uhol oblúka 3
S_0	[m]	13.14	13.25	13.49	Dĺžka trajektórie

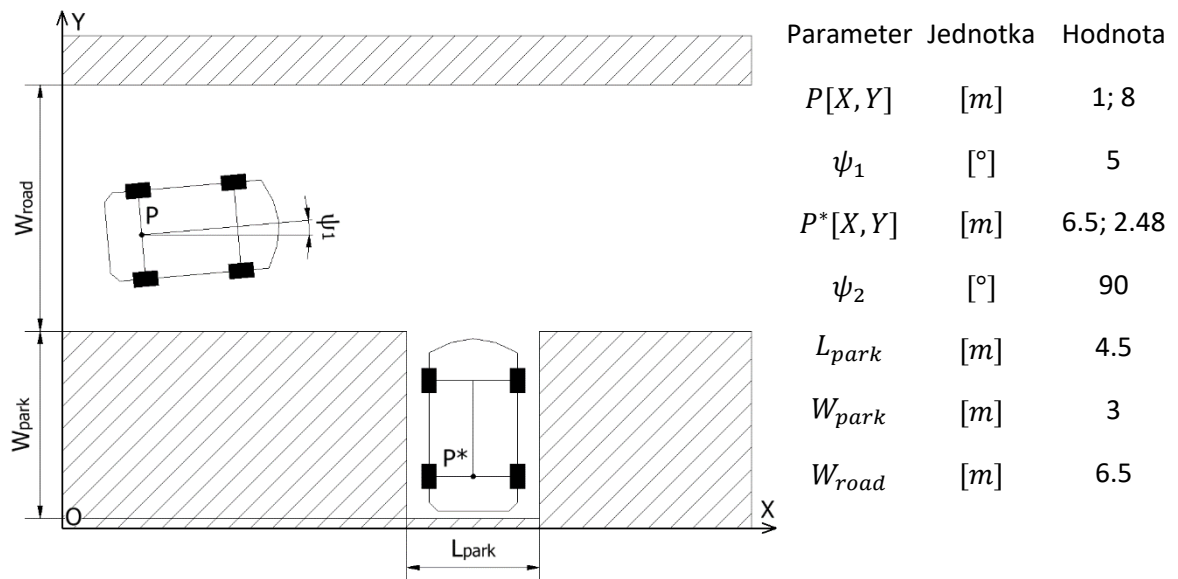
Tab. 6.4 Riešenie úlohy č.4: Výsledné parametre trajektórií

Zhodnotenie výsledkov

Za predpokladu dostatočne veľkého priestoru na pohyb je tento manéver jednoduchý a trajektóriu dostaneme pomocou analytického aj numerického riešenia. Výsledné trajektórie sa líšia len v malých detailoch. Tak ako pri prvej úlohe aj tu nastáva rozdiel pri dobe výpočtu, ale tento raz nie je až taký výrazný. Analytické riešenie zabralo približne 3.2 sekundy, lokálna optimalizačná metóda približne 1 sekundu (s dobrým odhadom parametrov), globálnej optimalizačnej metóde výpočet trval približne 14 sekúnd. V prípade, kedy medzi jedincov bol umiestnený odhad parametrov použitý pre lokálnu metódu, sa doba výpočtu skrátla na približne 3,7 sekundy.

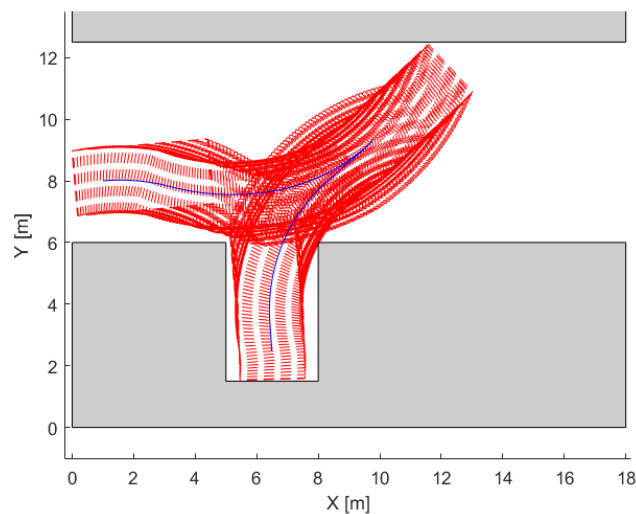
6.2.2 Úloha č.5: Vozidlo stojacie pred parkovacím miestom

Rovnako ako pri úlohe č.2 je táto úloha zložitejšia, pretože sa vozidlo musí dostať do ideálnej východiskovej polohy pred začatím parkovacieho manévru. Pri tejto úlohe uvažujeme o meter menší priestor na manévrovanie, ale parkovacie miesto ostáva ako v predchádzajúcej úlohe. Na obrázku nižšie (Obr. 6.10) je znázornené zadanie úlohy s počiatočnou a koncovou polohou vozidla a rozmermi prostredia.



Obr. 6.10 Úloha č.5: Vozidlo stojacie za parkovacím miestom

Analytická metóda nebola schopná zadanú úlohu vyriešiť pomocou trajektórie typu CSC. Nájst' vhodný odhad parametrov pre lokálnu optimalizačnú metódu je veľmi náročné a keďže je zakomponovaná v globálnej, samostatne nebola použitá. Trajektórie sa skladajú z ôsmich oblúkov.



Obr. 6.11 Riešenie úlohy č.5: Obálka vozidla opisujúceho trajektóriu

Výsledné hodnoty trajektórií popisuje tabuľka nižšie:

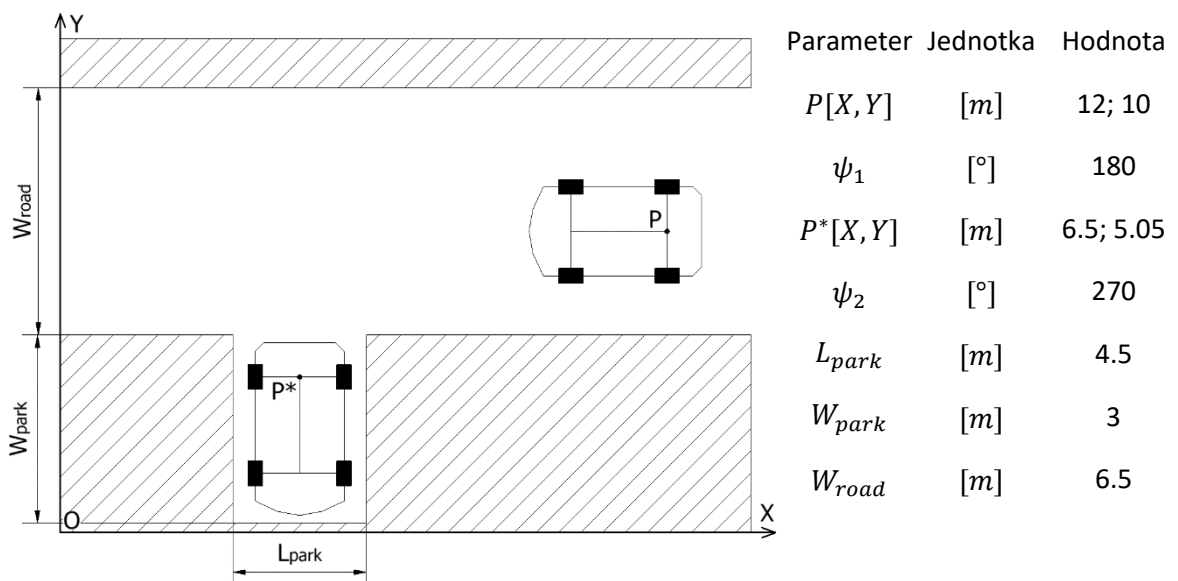
Parameter	r	β	Parameter	r	β
Jednotka	[m]	[rad]	Jednotka	[m]	[rad]
1	-10.12	-0.22	5	10.68	0.03
2	8.96	0.68	6	-7.01	0.18
3	-5.58	0.28	7	-7.53	0.03
4	-5.71	0.68	8	5.66	-0.19
Dĺžka trajektórie			16.65		
Tab. 6.5 Riešenie úlohy č.5: Výsledné parametre trajektórie					

Zhodnotenie výsledkov

Globálna optimalizačná metóda našla riešenie bez odhadu parametrov približne za 14 sekúnd a následne s odhadom pridaným do počítačovej populácie výpočet zabil približne 6 sekúnd.

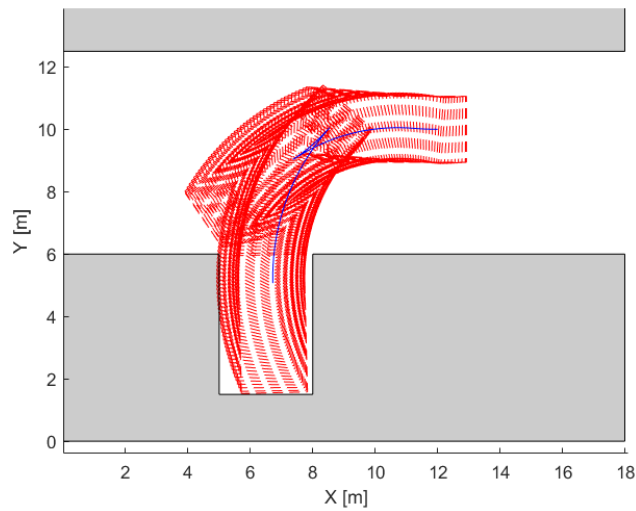
6.2.3 Úloha č.6: Vozidlo parkujúce popredu

Táto úloha je špeciálna tým, že ako jediná zahŕňa parkovanie popredu a nie cúvanie. Samotný manéver nie je zložitý, ale rovnako ako pri predchádzajúcej úlohe uvažujeme menší manévrovací priestor. Na obrázku nižšie (Obr. 6.12) je znázornené zadanie úlohy s počiatočnou a koncovou polohou vozidla a rozmermi prostredia.



Obr. 6.12 Úloha č.6: Vozidlo parkujúce popredu

Pre riešenie bola použitá globálna optimalizačná metóda obsahujúca lokálnu optimalizačnú metódu na zlepšenie dosiahnutých riešení. Trajektória je popísaná šiestimi oblúkmi.



Obr. 6.13 Riešenie úlohy č.6: Obálka vozidla opisujúceho trajektóriu

Výsledné hodnoty trajektórií popisuje tabuľka nižšie:

Parameter	r	β	Parameter	r	β
Jednotka	[m]	[rad]	Jednotka	[m]	[rad]
1	-9.4	-0.08	4	-5.55	8.4e-3
2	6.05	0.65	5	-6.04	0.25
3	6.03	0.02	6	7.22	0.75
Dĺžka trajektórie			11.81		
Tab. 6.6 Riešenie úlohy č.6: Výsledné parametre trajektórie					

Zhodnotenie výsledkov

Globálna optimalizačná metóda našla riešenie bez odhadu parametrov približne za 20 sekúnd a následne s odhadom pridaným do počiatočnej populácie výpočet zabral približne 6 sekúnd.

6.3 Zhodnotenie simulačného experimentu

Simulačný experiment ukázal schopnosť systému úspešne riešiť zadané úlohy. Analytické riešenie sa ukázalo ako vhodné pre jednoduché manévry, kedy výpočtový čas bol veľmi krátky. Pri zložitejších počiatočných pozíciách vozidla sa však ukázalo ako nepoužiteľné. Konečný návrh systému hľadajúci trajektóriu pomocou globálnej optimalizačnej metódy úspešne zvládol vyriešiť aj zložitejšie úlohy a trajektórie pozostávajúce z viacerých manévrov.

Záver

Hlavným cieľom diplomovej práce bolo navrhnuť vlastný systém hľadania trajektórie parkovacieho manévru pre autonómne vozidlo. Navrhnutý systém dokáže plánovať aj zložitejšie parkovacie manévry a jeho funkčnosť bola overená pri riešení niekoľkých úloh.

Na ceste k naplneniu hlavného cieľa bolo nutné splniť jednotlivé čiastkové ciele popísané v úvode práce. Na začiatok bolo dôležité zoznámiť sa s existujúcimi systémami a inšpirovať sa dostupnými zdrojmi, aby bolo možné zvoliť vhodné postupy. Stručný prehľad popisuje prvá kapitola Súčasný stav skúmanej problematiky, čím bol splnený prvý čiastkový cieľ.

Na základe získaných poznatkov bol vytvorený kinematický model vozidla a v prostredí programu Simulink simulačný model. Odvodenie potrebných rovníc a opis jednotlivých častí modelu popisuje druhá kapitola Model vozidla. Tým bol splnený druhý čiastkový cieľ.

Pre overenie správnosti kinematického a simulačného modelu a pre prvotný náhľad do problematiky plánovania trajektórie vozidla bola vytvorená analytická metóda popisujúca trajektóriu zloženú z oblúkov a priamky. Základný jednoduchý návrh trajektórie ako aj návrh pre prípad s nenulovými počiatočnými natočeniami vozidla popisuje tretia kapitola.

Jednou z kľúčových vlastností systému je schopnosť vyhodnotiť kolíziu vozidla počas pohybu po nájdenej trajektórii. Algoritmus vyhodnocovania kolízie, a teda aj nástroj na vyhodnotenie dosiahnuteľnosti parkovacieho miesta popisuje štvrtá kapitola Kolízia – prostredie s prekážkami. Tým bol splnený tretí čiastkový cieľ.

Požiadavka schopnosti systému nájsť trajektóriu i v prípade, že je parkovacie miesto dosiahnuteľné iba zložitejším opakovaným manévrouvaním, vyžadovalo zložitejšiu metódu. Lokálna optimalizačná metóda sa ukázala nedostatočná pri použití samostatne a preto sa stala súčasťou komplexnejšieho postupu globálnej optimalizačnej metódy. Potrebné informácie o oboch metódach, ale aj základný postup riešenia pomocou genetického algoritmu popisuje piata kapitola Trajektória – numerický výpočet. Tým bol splnený štvrtý čiastkový cieľ.

Na záver bol navrhnutý systém podrobený testu v podobe simulačného experimentu. Účelom bolo overiť funkčnosť, ale aj porovnať jednotlivé opisované metódy riešenia. Postup a zhodnotenia jednotlivých úloh popisuje posledná, šiesta kapitola Simulačný experiment. Tým bol splnený posledný, piaty čiastkový cieľ.

Hlavným prínosom je možnosť využiť navrhnutý systém v modeli automobilu na katedre Mechaniky, biomechaniky a mechatroniky na Fakulte strojní ČVUT v Prahe. Ďalším prínosom je možnosť nadviazať na dosiahnuté výsledky pri potrebe ďalšieho výskumu v tejto oblasti. Hlavný program systému je napísaný prehľadne a je možné jednoducho zmeniť rozmery vozidla ale aj jeho celkový typ, napríklad pre použitie pre manipulátory alebo iné pohyblivé zariadenia. Taktiež je pomerne jednoducho možné zmeniť typ trajektórie z popisu pomocou oblúkov na zložitejšie tvary.

Odporúčanie pre prípad ďalšieho výskumu v tejto oblasti je použitie zložitejšieho popisu trajektórie pre dosiahnutie plynulejších manévrov, použitie výkonnejšieho hardvéru pre skrátenie výpočtového času hľadania trajektórie a vyskúšanie nových heuristických postupov pre genetické algoritmy pre zrýchlenie a zlepšenie konvergovania riešenia k ideálnemu riešeniu.

Zoznam použitých zdrojov

- [1] Andreev, N., „A brief history of car parking technology,“ 13 06 2018. [Online]. Available: <https://www.confused.com/on-the-road/gadgets-tech/parking-technology-brief-history>. [Cit. 24 07 2018].
- [2] „What are the differences between VW Park Pilot and Park Assist?,“ 01 04 2018. [Online]. Available: <https://www.schworervolkswagen.com/blog/volkswagen-park-pilot-vs-volkswagen-park-assist/>. [Cit. 24 07 2018].
- [3] „Park4U® Automated Parking.,“ [Online]. Available: <https://www.valeo.com/en/park4u-automated-parking/>. [Cit. 24 07 2018].
- [4] Lv, Z., Zhao, L., Liu, Z., „A Path-Planning Algorithm for Automatic Parallel Parking,“ *2013 Third International Conference on Instrumentation, Measurement, Computer, Communication and Control*, pp. 474 - 478, 2013.
- [5] Wang, L., Guo, L., He, Y., „Path Planning Algorithm for Automatic Parallel Parking from Arbitrary Initial Angle,“ *2017 10th International Symposium on Computational Intelligence and Design (ISCID)*, pp. 55 - 58, 2017.
- [6] Reeds, J. A., Shepp, L. A., „Optimal paths for a car that goes both forwards and backwards,“ *Pacific J. Math.* 145, zv. 2, pp. 367-393, 1990.
- [7] Scheuer, A., Fraichard, Th., „Continuous-curvature path planning for car-like vehicles,“ *Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems.*, pp. 997 - 1003, 1997.
- [8] Hu, Y., Yang, S. X., „A knowledge based genetic algorithm for path planning of a mobile robot,“ *Proceedings of IEEE International Conference on Robotics and Automation, 2004.*, pp. 4350 - 4355, 2004.
- [9] Wang, M., Liu, J., „Fuzzy logic based robot path planning in unknown environment,“ *International Conference on Machine Learning and Cybernetics*, pp. 813 - 818, 2005.

- [10] „Škoda Yeti,“ Škoda Auto, 01 09 2009. [Online]. Available: <http://skoda-auto-web.blog.cz/0909/skoda-yeti>. [Cit. 24 07 2018].
- [11] „MATLAB Help: atan2,“ The MathWorks, Inc., [Online]. Available: <https://uk.mathworks.com/help/matlab/ref/atan2.html>. [Cit. 24 07 2018].
- [12] „MATLAB Help: fmincon,“ The MathWorks, Inc., [Online]. Available: <https://uk.mathworks.com/help/optim/ug/fmincon.html>. [Cit. 24 07 2018].
- [13] Holland, J., Adaptation in natural and artificial systems, Ann Arbor University of Michigan Press, 1975.
- [14] Michalewicz, Z., Genetic algorithms + Data structures = Evolution, Third, Revised and Extended Edition, Springer, 1996.

Prílohy

- Príloha 1: CD médium
- diplomová práca v elektronickej forme (.pdf)
 - zdrojové kódy systému a kinematický model v Simulinku (pričínok Programy)