



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

## ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A BIOMECHANIKY

INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS

## NÁVRH ALGORITMU AUTOMATICKÉHO PARKOVÁNÍ PRO EXPERIMENTÁLNÍ VOZIDLO SE ČTYŘMI ŘÍZENÝMI KOLY

ALGORITHM FOR AUTOMATIC PARKING FOR FOUR WHEEL STEERING VEHICLE

### BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

### AUTOR PRÁCE

AUTHOR

Michaela Vrbková

### VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Robert Grepl, Ph.D.

BRNO 2016



# Zadání bakalářské práce

Ústav:	Ústav mechaniky těles, mechatroniky a biomechaniky
Studentka:	<b>Michaela Vrbková</b>
Studijní program:	Strojírenství
Studijní obor:	Základy strojního inženýrství
Vedoucí práce:	<b>doc. Ing. Robert Grepl, Ph.D.</b>
Akademický rok:	2015/16

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

## **Návrh algoritmu automatického parkování pro experimentální vozidlo se čtyřmi řízenými koly**

### **Stručná charakteristika problematiky úkolu:**

Tato práce navazuje na několik diplomových a bakalářských prací zabývajících se experimentálním vozidlem Car4 obhájených na ÚMTMB FSI VUT v Brně. Náplní práce je vytvoření demonstrátoru parkovacího asistenta včetně implementace do reálné řídicí jednotky a otestování funkčnosti na reálném vozidle.

### **Cíle bakalářské práce:**

- 1) Seznámit se s předchozími pracemi na projektu Car4. Seznámit se s kinematickým modelem vozidla se čtyřmi řízenými koly.
- 2) Na základě dostupných zdrojů vybrat a upravit algoritmus pro řízení natočení kol a pojezdu pro zaparkování do podélného parkovacího místa. Ukázat funkčnost algoritmu na simulaci naprogramované v prostředí MATLAB.
- 3) Navrhnout a realizovat umístění dostupných parkovacích senzorů a řídicí jednotky na vozidle Car4. Implementovat dostupný algoritmus zpracování dat ze senzorů pomocí automatického generování kódu z prostředí Matlab/Simulink.
- 4) Aplikovat navržený algoritmus plánování trajektorie na reálnou úlohu automatického parkování experimentálního vozidla Car4. Prostor pro parkování je nalezen pomocí ultrazvukových senzorů, následně je vypočítána trajektorie a vozidlo zaparkuje.

### **Seznam literatury:**

Valášek, M.: Mechatronika, Vydavatelství ČVUT 1995.

Grepl, R.: Modelování mechatronických systémů v Matlab/SimMechanics, BEN, 2007.

Grepl, R.: Kinematika a dynamika mechatronických systémů, Nakladatelství CERM, ISBN: 978-8-214-3530-8.

Noskievič, P.: Modelování a identifikace systémů, ČVUT 1999.

Horáček, P.: Systémy a modely, ČVUT 1999.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2015/16

V Brně, dne

L. S.

---

prof. Ing. Jindřich Petruška, CSc.  
ředitel ústavu

---

doc. Ing. Jaroslav Katolický, Ph.D.  
děkan fakulty

## **Abstrakt**

Bakalářská práce je zaměřena na tvorbu parkovacího asistenta určeného pro podélné parkování vozidla s přední i zadní natáčecí nápravou (dále jen 4WS). V bakalářské práci je provedena rešerže v oblasti 4WS kinematiky a dostupných senzorů pro parkování. Součástí bakalářské práce je nově vytvořený software pro zaznamenání trajektorií 4WS automobilu včetně popisu jeho funkčnosti. Výsledkem bakalářské práce je algoritmus pro parkovacího asistenta, který byl teoreticky simulován v programu Matlab a poté otestován na skutečném vozítku "Car4" v mechatronické laboratoři FSI Vysokého učení technického v Brně.

## **Summary**

The bachelor thesis is focused on creating a designated parking assistant for the parallel parking of a vehicle with front and rear steering axels (hereinafter 4WS). There is a literature search in the field of 4WS kinematics and sensors available for parking in the bachelor thesis. Part of the bachelor thesis is newly created software to record the trajectories of a 4WS car, including a description of its functionality. The result of the bachelor thesis is an algorithm for parking assistance, which was theoretically simulated in the computer program Matlab and then tested on a real vehicle, "Car4" in the Mechatronic Laboratory FSI, Brno University of Technology.

## **Klíčová slova**

Parkovací algoritmus, parkovací senzory, Car4.

## **Keywords**

Parking algorithm, parking sensors, Car4.

VRBKOVÁ, M. *Návrh algoritmu automatického parkování pro experimentální vozidlo se čtyřmi řízenými koly*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2016. 32 s. Vedoucí bakalářské práce doc. Ing. Robert Grepl, Ph.D.

Prohlašuji, že jsem bakalářskou práci na téma: "Návrh algoritmu automatického parkování pro experimentální vozidlo se čtyřmi řízenými koly" vypracovala samostatně s použitím odborné literatury a zdrojů uvedených na seznamu, který tvoří přílohu této práce.

Michaela Vrbková



Děkuji doc. Ing. Robertu Greplovi Ph.D. za pomoc při vedení bakalářské práce. Mé poděkování patří též lidem z mechatronické laboratoře a především Martinu Brablčovi za neocenitelné rady.

Michaela Vrbková



# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Rešerše</b>	<b>5</b>
2.1	Kinematika 4WS	5
2.1.1	Ackerman	5
2.1.2	Módy řízení	6
2.1.3	Vývoj 4WS automobilů	7
2.1.4	Současné 4WS automobily	7
2.2	Senzory	8
2.2.1	IR senzory	8
2.2.2	Ostatní senzory používané při parkování	10
2.2.3	Dolní propust	12
2.3	Proměnlivá rychlost během parkování	12
2.4	Car4	13
<b>3</b>	<b>Cíle řešení</b>	<b>14</b>
<b>4</b>	<b>Řešení</b>	<b>15</b>
4.1	Software pro nahrávání trajektorií	15
4.1.1	GUI	16
4.1.2	Zobrazení	16
4.1.3	Kolize objektů	18
4.1.4	Kinematický model	18
4.1.5	Výpočet natočení kol	19
4.1.6	Použití joysticku k ovládní GUI	20
4.2	Výběr vhodných trajektorií	20
4.2.1	Tabulka	20
4.2.2	Nahrávání trajektorií v mechatronické laboratoři	21
4.2.3	Úprava nasbíraných dat	22
4.3	Algoritmus parkování a jeho simulace	22
4.3.1	Popis algoritmu	22
4.3.2	Simulace IR senzoru	23
4.3.3	Hledání hran	23
4.4	Reálný experiment	23
4.4.1	Proměnlivá rychlost	23
4.4.2	P regulátor	24
4.4.3	Měření IR senzoru	25
4.4.4	Záznam z experimentu	25
<b>5</b>	<b>Závěr</b>	<b>26</b>
<b>6</b>	<b>Literatura</b>	<b>28</b>
<b>7</b>	<b>Seznam použitých zkratk</b>	<b>31</b>



# 1. Úvod

Parkovací asistenti začínají být nedílnou součástí výbavy moderních aut, protože jak podélné, tak kolmé parkování může být pro řidiče náročné. Roli hrají faktory, jako je velikost parkovacího místa, schopnosti řidiče a rychlost, s jakou je třeba zaparkovat.

Lidé jsou vždy zpočátku nedůvěřiví vůči novým inovacím, proto je zde velká opatrnost z řad řidičů, kteří mají možnost využít parkovacího asistenta, a pokud se nejedná o technické nadšence nebo o lidi, které parkování přivádí k šílenství, pravděpodobně si své auto raději zaparkují sami. Nicméně si řidiči při použití parkovacích asistentů vedou lépe, než když se auto pokouší parkovat sami [1]. I přes použití asistentů je pořád "u kormidla" člověk a zodpovídá za veškeré chování automobilu. Parkovací asistenti změří velikost parkovacího místa, ale pořád je na řidiči, na jakou stranu bude auto parkovat, a jestli se bude jednat o kolmé nebo podélné parkování. Stejně jako parkovací asistent navrhne vhodnou trajektorii a od toho se odvíjející pohyb volantu, ale stále je to řidič, kdo ovládá pedály a řazení. To se v blízké budoucnosti změní systémem, který bude ovládán pouze aplikací v mobilu, aby tím minimalizoval místo, které je třeba nechat okolo auta pro výstup pasažérů. Diskutabilní je, zda se tímhle stylem parkování neztíží podmínky vyjíždění okolních aut z místa, které nebudou mít tento typ parkovacího asistenta. V současnosti je v prodeji BMW 7, které je možné zaparkovat pomocí dálkového ovládání, bohužel to dosud není zkombinováno s parkovacím asistentem a umožňuje pouze jízdu rovně o velikost 1.5 násobku délky auta [2].

Všichni parkovací asistenti fungují na stejném principu, kdy nejprve změří parkovací místo pomocí senzorů, rozhodnou se, zda je možné zaparkovat, navrhnu vhodnou dráhu (strategii) a nakonec přejdou k samotnému parkování.



Obrázek 1.1: Příklad uživatelského rozhraní parkovacího asistenta. [3]

Při parkovacím manévru se využívá různorodých přístupů, ať už se jedná o navrhování vhodné trajektorie nebo o její sledování [4] [5]. Bohužel návrháři těchto systémů využívaných v průmyslu nejsou příliš sdílní, proto tahle práce vychází z volně dostupných zdrojů a odborných článků, většinou dostupných na IEEE [6], kde jsou uvedeny sumarizace výzkumů z technických oborů v podobě několikastránkových článků.

Rešeršní část obsahuje srovnání senzorů používaných jak pro parkovací asistenty, tak pro manuální parkování. Dále jsou v rešerši popsány výhody 4WS kinematiky.

Část práce se zabývá vytvořením softwaru pro sběr vhodných trajektorií, které byly později použité pro parkovací algoritmus.

Parkovací algoritmus byl nejprve otestován v prostředí Matlab při simulaci; později byla funkčnost otestována na Car4, což je experimentální 4WSWD vozítko, vytvořené v mechatronické laboratoři na základě několika bakalářských a diplomových prací, které dále slouží pro edukativní účely.

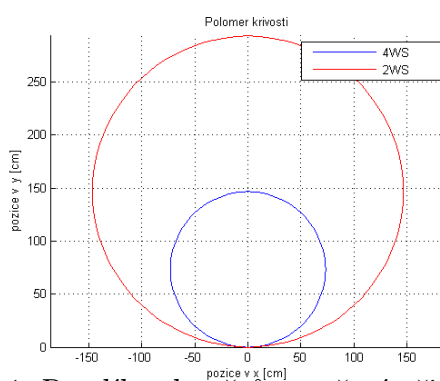
## 2. Rešerše

### 2.1. Kinematika 4WS

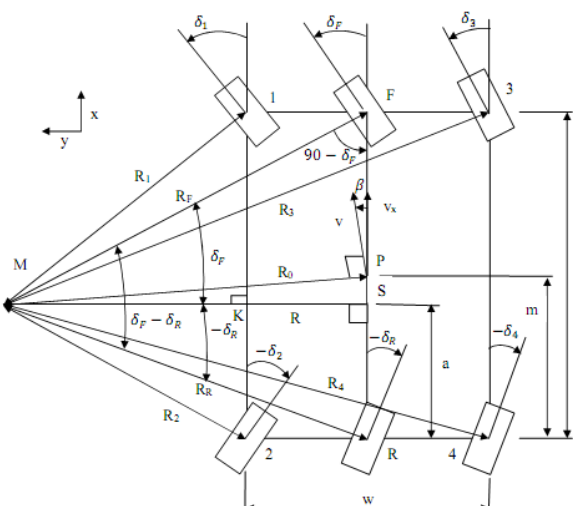
#### 2.1.1. Ackerman

Při zatáčení auta doleva se vnitřní (levé) kolo natočí o větší úhel než vnější (pravé) kolo. Toto dovoluje vnitřnímu kolu sledovat menší kružnici, než tu kterou sleduje vnější kolo, a předchází tak oděru pneumatik a skluzu pneumatik při zatáčení. Ackermannovou podmínkou je mít všechna čtyři kola otáčející se kolem společného středu otáčení během průjezdu zatáčkou. V této práci byl použit holonomní kinematický model se čtyřmi hnanými a čtyřmi řízenými koly. Pomocí splnění Ackermannovy podmínky bylo možné dopočítat natočení jednotlivých kol.

Na následujícím obrázku je patrná výhoda 4WS kinematiky podle Ackermanna.



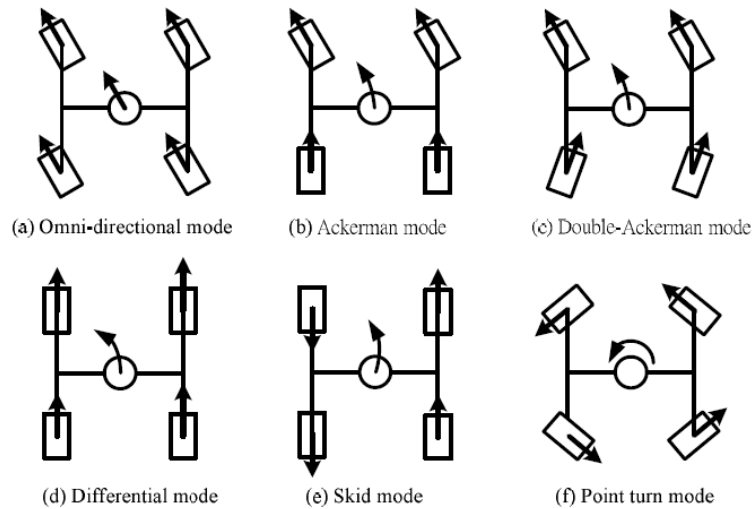
Obrázek 2.1: Rozdíl poloměrů zatočení při 2WS a 4WS



Obrázek 2.2: Příklad 4WS [7]

## 2.1.2. Módy řízení

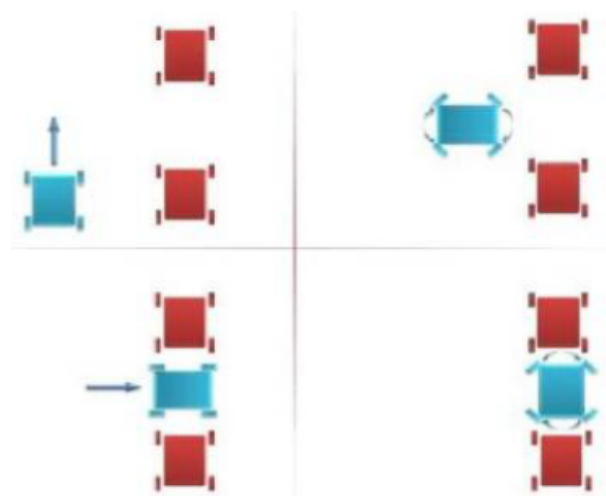
Existuje několik způsobů, jak je možné kontrolovat pohyb auta se čtyřmi řízenými a čtyřmi hnanými koly (4WS4WD). Rozdíl mezi nimi tvoří možné pozice středu otáčení a směr otáčení kol. Zajímavá je kombinace "omni-directional" (všesměrový pohyb) a "point turn" (otáčení na místě). Takové auto může sledovat libovolnou trasu s menší spotřebou energie a vysokou pohyblivostí.



Obrázek 2.3: Různé způsoby řízení 4WS vozidla [8]

- Využití „point turn“ módu

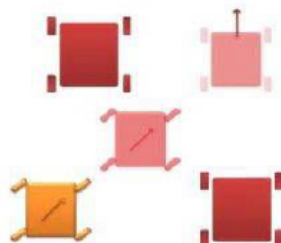
Manévr pro podélné parkování vyžaduje třicestný pohyb auta a velké vstupní natočení. Navíc aby bylo možné úspěšně zaparkovat a nevznikly tím žádné škody, je potřeba nejméně 1.75 násobku délky auta dostupné pro zaparkování 2WS auta. Kombinací systému 4WS Ackerman s módem, který se umí otočit na místě, by se výrazně zjednodušilo parkování, protože by pro zaparkování bylo potřeba jen místo stejně velké, jako je velikost parkujícího auta. Bohužel, tento typ řízení ještě není realizovaný u průmyslově vyráběných aut.



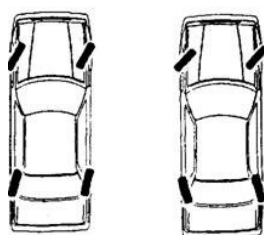
Obrázek 2.4: Paralelní parkování s využitím "point turn" módu [9]

- Využití "omni-directional" módu (krabího pohybu) a "double-Ackermann" módu

Jedná se o dva nejpoužívanější módy řízení u 4WS aut. Mód „omnidirectional“ je používán pro vysoké rychlosti pro zlepšení jízdní stability. Na obr. 2.5 je vidět, jak může zjednodušit změnu pruhu při vysoké rychlosti, zatímco při nízkých rychlostech se využívá double Ackermann, a to pro lepší schopnost manévrování.



Obrázek 2.5: Změna jízdního pruhu [9]



Obrázek 2.6: vlevo: omni-directional, vpravo: double-Ackerman [10]

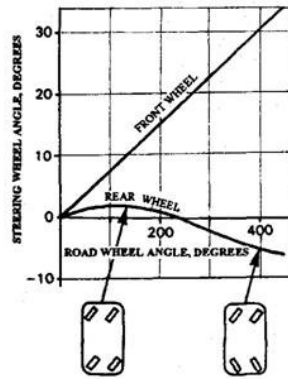
### 2.1.3. Vývoj 4WS automobilů

V současnosti je většina aut uzpůsobena pro 2WS Ackermana. Nicméně už od roku 1987 se začala průmyslově vyrábět auta s 4WS systémy. Jedním z prvních problémů při použití 4WS Ackermana bylo, že při vysokých rychlostech ztrácelo vozidlo stabilitu. Problém se stabilitou byl vyřešen pomocí velikosti natočení volantu. Pokud se natočil volant o malý úhel, zadní kola se natočila (o  $1.5^\circ$ ) stejným směrem jako přední, ale pokud se úhel natočení předních kol stále zvětšoval, zadní kola se začala vracet do rovnovážné polohy a následně se vytočila opačným směrem (o  $5.3^\circ$ ), aby se zmenšil poloměr otáčení (obr. 2.7). Tohoto řízení bylo dosaženo čistě mechanickým způsobem, kombinací excentrické hřídele spolu s planetovou převodovkou [11], [12].

### 2.1.4. Současné 4WS automobily

Komplikovanou mechanickou převodovku nahradily elektricky poháněné aktuátory řízené počítačově, což vedlo k natočení zadních kol na natočení volantu a rychlosti automobilu [23]. Maximální natočení zadních kol se liší podle současné rychlosti automobilu. Přejechod mezi módem, kde se zadní kola vytáčí opačným směrem než přední kola, a módem, kde se přední i zadní kola vytáčí stejným směrem, je u různých značek aut individuální (tab. 2.1), nicméně se tato hodnota vždy pohybuje okolo 60 km/h.

## 2.2. SENZORY



Obrázek 2.7: Změna natočení předních a zadních kol v závislosti na zvětšujícím se natočení volantu [10]

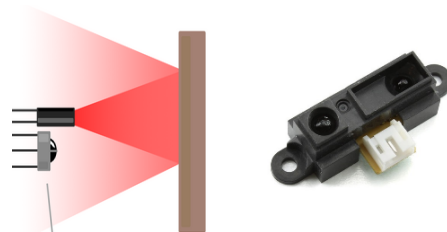
Tabulka 2.1: Maximální velikost natočení zadních kol, při vysoké rychlosti(vlevo) a při nízké rychlosti(vpravo).

Porsche 911 GT [24]	1.5°	2.8°
Renault Laguna GT [23]	2°	3.5°
Audi Q7 [25]	„lehce“	5°

## 2.2. Senzory

### 2.2.1. IR senzory

Infračervené senzory se skládají z led diody vyzařující infračervené světlo, detektoru odraženého světla a obvodu zpracovávajícího signál z detektoru.

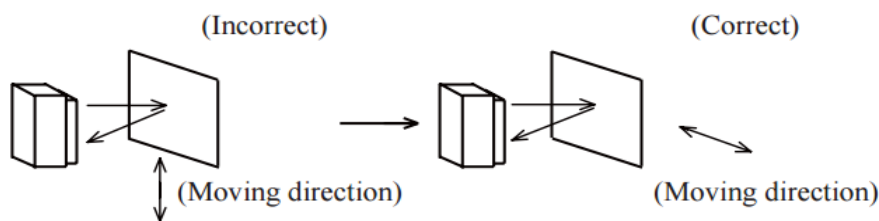


Obrázek 2.8: Schéma fungování IR senzoru [13] a fotografie použitého senzoru [14]

- **Pohybující se senzor**

Pokud se senzor a měřený objekt vůči sobě pohybují, jak je tomu v případě měření překážek jedoucím autem, záleží na orientaci IR senzoru. Od výrobců je doporučeno, aby spojnice mezi led diodou a detektorem byla kolmá na směr pohybu. Tento přístup vede k snížení odchylek měřené vzdálenosti. [14]

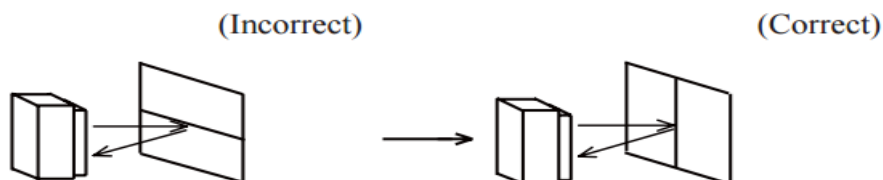




Obrázek 2.9: Příklad špatné orientace (vlevo) a správné orientace (vpravo) při pohybu senzoru [14]

- **Odrazivost objektů a světelné podmínky**

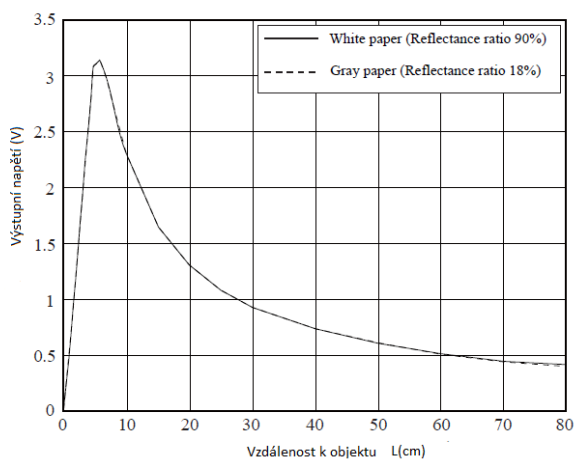
Vzhledem k tomu, že se měření provádí na základě detekce odraženého IR světla, záleží jak na odrazivosti („barvě“) předmětu, tak i na světelných podmínkách při měření. Pokud se na měřených objektech nachází velmi výrazný barevný přechod, je vhodné, aby spojnice mezi led diodou a detektorem byla rovnoběžná s hranicí barevného přechodu. [14]



Obrázek 2.10: Špatně orientovaný senzor (vlevo) a správně orientovaný senzor (vpravo) [14]

Mezi výhody infračervených senzorů patří, že mají rychlejší odezvu než ultrazvukové, jsou levnější a s jejich analogovým výstupem se snáze pracuje. Jsou proto hojně využívány jako senzory přiblížení v robotice [26].

Jejich nevýhodou je nelinearita závislosti vzdálenosti na výstupním napětí a také závislost na odrazivosti okolí. Proto by zjišťování vzhledu okolního prostředí mohlo být nepřesné, nicméně pro otestování parkovacího algoritmu, kdy známe tvar překážek i jejich barvu, je jejich přesnost dostatečná.



Obrázek 2.11: Příklad výstupní charakteristiky při měření vzdálenosti [14]

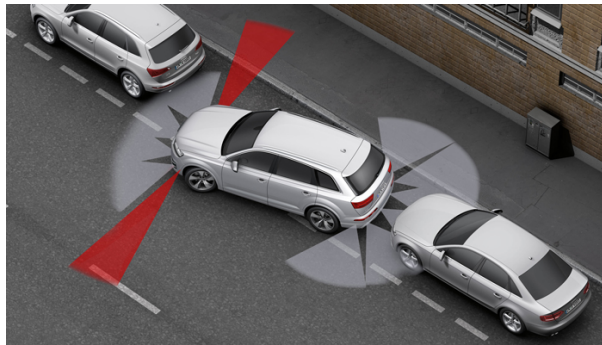
## 2.2. SENZORY

K měření byl použit pouze jeden senzor ve středu auta, protože uvažujeme ideální podmínky, jako je zaparkování podélně vedle auta před parkovacím místem. A také že auta, mezi které se parkuje, jsou zaparkována schopnými řidiči, tudíž není třeba více senzorů pro zjišťování tvaru překážek.

### 2.2.2. Ostatní senzory používané při parkování

- **Ultrazvukové senzory**

Tyto senzory vytváří zvukové vlny s frekvencí vyšší než 20 kHz a měří dobu, za jakou se vrátí jejich echo od překážky, a na základě toho je následně vypočítána vzdálenost. Běžná velikost rozsahu parkovacích senzorů je 0.3 m – 1.5 m. [15]



Obrázek 2.12: Příklad rozmístění ultrazvukových senzorů, které se používá i u parkovacích asistentů. [16]

Mezi výhody patří, že jejich výstup není závislý na světelných podmínkách ani na barvě objektů. Odrazivost vln je závislá na typu materiálu, ale pokud se jedná o parkování mezi auta, případně stěny, není tato vlastnost překážkou.

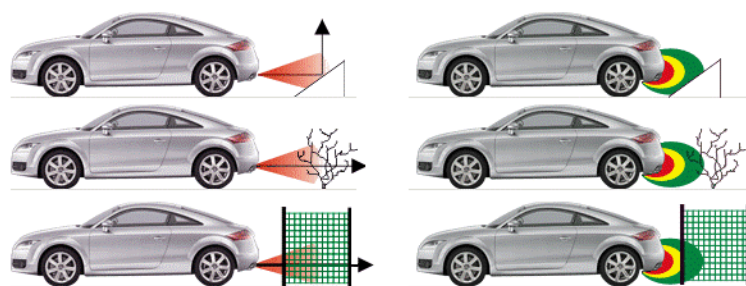
Nevýhodou je v první řadě instalace senzorů, která vyžaduje vrtání do nárazníku. Dalším problémem je, že nemusí zaznamenat veškeré překážky, například pokud se jedná o tenké sloupky, keře nebo cokoliv, co nemá dostatečně velkou plochu pro odrazení vln (obr. 2.13). Pokud mají senzory fungovat správně, je nutné, aby na nich nebylo žádné znečištění. [17] Jedná se o senzory používané u parkovacích asistentů, protože slouží dobře k změření tvaru překážek a mají na to dostatečný rozsah.

- **Elektromagnetické**

Je vytvořeno magnetické pole kolem zadního (předního) nárazníku. Pokud dojde k narušení pole, systém začne vydávat varovné signály, které záleží na vzdálenosti k objektu. Běžný rozsah parkovacích senzorů je asi 0.1-0.7 m.

Výhodou je, že tyto senzory nemusí být na povrchu auta, tudíž nejsou vidět a jejich instalace za nárazník není problém. Současně s tím nejsou závislé na znečištění auta ani na světelných podmínkách. Jejich výhodou je, že reagují na veškeré narušení magnetického pole, nezáleží na tvaru překážek, jejich objemu nebo natočení.

Jejich nevýhodou může být, že mohou reagovat na déšť a jejich instalace není možná, pokud má auto kovový nárazník. Nepoužívají se jako senzory u parkovacích asistentů, protože překážku jen zaznamenají, ale žádné další informace nezjistíme. [17]



Obrázek 2.13: Srovnání ultrazvukových senzorů (vlevo) a elektromagnetických (vpravo) při detekci atypických překážek. [17]

- **Couvací kamera**

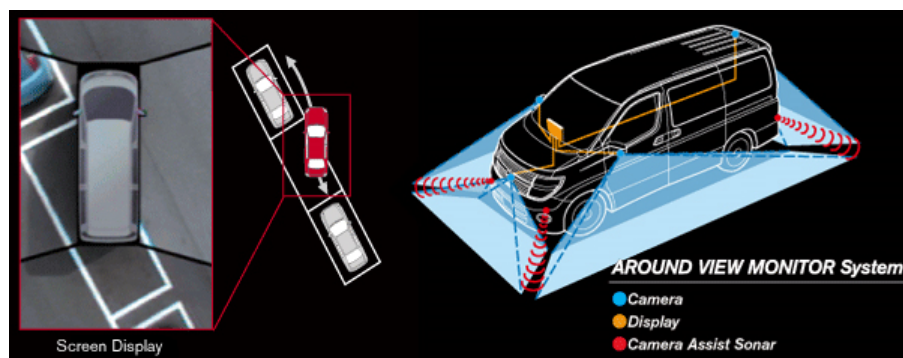
Mezi hlavní parametry patří pozorovací úhel, který bývá od  $90^{\circ}$ - $180^{\circ}$  (pro srovnání lidské oko má pozorovací úhel  $130^{\circ}$ ). Dále se většinou jedná o bezdrátové kamery, které používají analogový signál, což může interferovat s ostatními elektronickými zařízeními v autě, a tím pádem zhoršovat kvalitu obrazu. Tomu se dá předejít použitím digitálního přenosu. Další dostupnou výhodou bývá noční vidění, díky infračerveným senzorům na vzdálenost 7,5 m. [27]



Obrázek 2.14: Příklad výstupu z couvací kamery, kde jsou na monitoru zobrazeny čáry s odhadem vzdálenosti a současně může být například naznačena trasa vozidla v dané chvíli. [18]

- **kamera s úhlem 360**

Jedná se o zobrazení výstupu ze čtyř kamer umístěných dokola na autě (obr. 2.15). Kompozice poskytuje tzv. ptačí pohled. V případě Nissanu se dá tato funkce používat současně se zpětnou kamerou, aby tak maximálně usnadnili orientaci v prostoru při parkování.

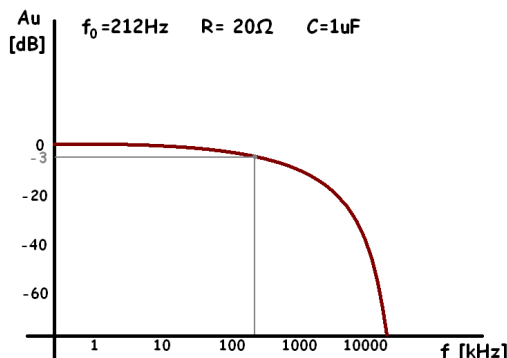


Obrázek 2.15: Rozmístění senzorů na autě a současně příklad toho, jak vypadá výstup na monitoru při podélném parkování [19]

## 2.3. PROMĚNLIVÁ RYCHLOST BĚHEM PARKOVÁNÍ

### 2.2.3. Dolní propust

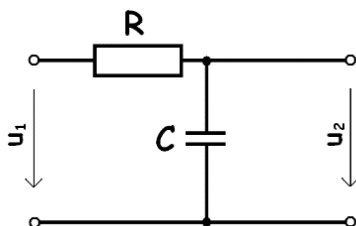
Dolní propust je lineární filtr, který propouští signály pouze nižších frekvencí, než je hranice filtru. Na vstupu je zapojen sériově kondenzátor a po té paralelně rezistor. Tohle zapojení zajistí zvyšující se impedanci členu s rostoucí frekvencí vstupního signálu. Nastavení filtru se provádí pomocí mezní frekvence  $f_0$ , což je hodnota frekvence, kdy zisk poklesne o 3 dB (zhruba 30 %). [20]



Obrázek 2.16: Příklad průběhu zisku v závislosti na frekvenci [20]

Výpočet mezní frekvence:

$$f_0 = \frac{1}{\pi R C} \quad (2.1)$$



Obrázek 2.17: Schéma základního filtru dolní propusti [20]

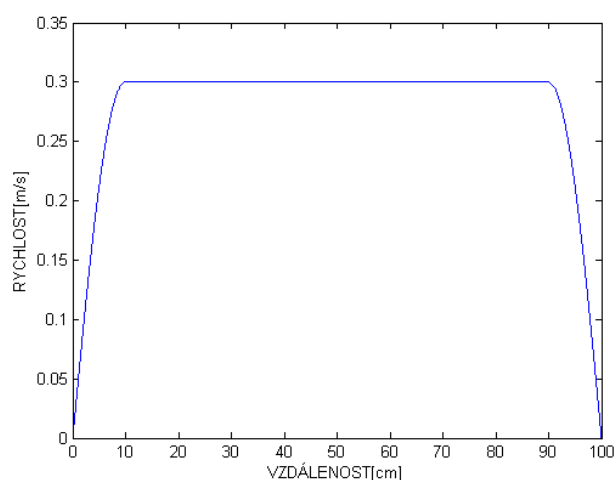
Používá se při zpracování signálu v elektronice pro odstranění nežádoucího šumu vyšších frekvencí. Další využití dolní propusti je například v audio technice. [20]

## 2.3. Proměnlivá rychlost během parkování

Rychlost se mění v závislosti na již ujeté vzdálenosti. V programu je zadáno, jaká vzdálenost je vhodná pro plynulé rozjetí/zastavení autíčka na požadovanou maximální rychlost a také velikost maximální rychlosti. V úsecích, kde se mění rychlost, je křivka, ze které se čte aktuální požadovaná rychlost, vytvořena pomocí sinu [4],

$$f(x) = \begin{cases} 0 < s_d < l_w, & v_{max} \sin\left(\frac{s_d}{2l_w} \pi\right) \\ l_w < s_d < l - l_w, & v_{max} \\ l - l_w < s_d < l, & v_{max} \sin\left(\frac{s_d - (l - 2l_w)}{2l_w} \pi\right) \end{cases}$$

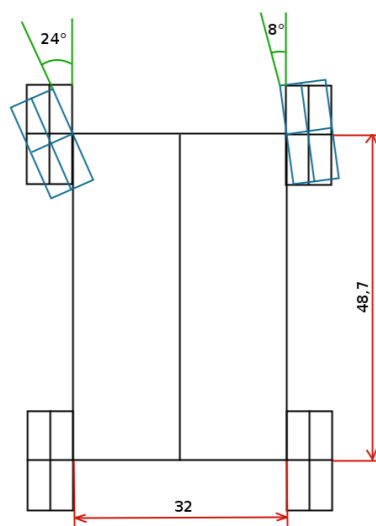
kde  $s_d$  je ujetá vzdálenost,  $l_w$  je vzdálenost potřebná pro zrychlení/zpomalení autíčka  $v_{max}$  je maximální požadovaná rychlost a  $l$  je celková délka trasy.



Obrázek 2.18: Změna rychlosti podle ujeté vzdálenosti

## 2.4. Car4

Car 4 je experimentální vozítko, vytvořené v mechatronické laboratoři na základě několika bakalářských a diplomových prací. Jedná se o vozítko se čtyřmi řízenými a čtyřmi hnanými koly, na kterém se od doby jeho vzniku každoročně uskutečňují závěrečné práce studentů. Každé kolo je individuálně poháněno DC motorem s enkodérem, a to kvůli odometrii a řízení rychlosti. Natáčení kol je řízeno pomocí RC servomotorů. Celé vozítko může být dálkově ovládáno přes rozhraní UART. [22]



Obrázek 2.19: Rozměry Car4

### 3. Cíle řešení

Existují různé přístupy k vytvoření vhodné trajektorie pro dané podmínky. Těmito podmínkami je myšlena velikost parkovacího místa, boční vzdálenost mezi autem a parkovacím místem a také velikost auta, které chceme zaparkovat.

Na základě rešerše a dostupných senzorů byly upraveny cíle, kterých má být v této práci dosaženo. Bylo rozhodnuto, že parkovací strategie bude založená na zkušenostech a schopnostech řidičů. Aby to bylo možné využít, bylo potřeba vytvořit software pro simulaci parkování, kde bude zobrazeno 4WS vozidlo s parkovacím místem a kde se bude zaznamenávat, jak řidiči podélně parkují. K zjištění rozměrů parkovacího místa bude použit infračervený senzor, protože práce s analogovým výstupem z tohoto senzoru, je více uživatelsky přívětivá a také protože se jedná o simulaci, kde jsou rozměry vozidla mnohonásobně menší než rozměry skutečných automobilů, tudíž i rozměry překážek, které je nutné změřit. V neposlední řadě jsou tyto senzory levnější a často používané v robotice na měření vzdáleností.

#### Cíle byly finálně definovány takto:

##### **Analyzovat způsob parkování různých fyzických osob.**

- Vytvořit softwarový тренаžér pro simulaci parkování s parametry experimentálního vozidla Car4 .
- Umožnit nezávislým osobám v softwarovém тренаžéru parkovat experimentální vozidlo Car4 a tak získat data o způsobu jejich parkování.
- Pomocí software na záznam trajektorií analyzovat efektivitu parkování různých fyzických osob a využít je při návrhu parkovacího algoritmu.

##### **Použít infračervené senzory pro zjištění rozměrů parkovacího místa.**

- Vhodně umístit infračervený senzor na Car4 a zajistit jeho funkčnost.
- Zpracovat výstup z infračerveného senzoru.
- Na základě získaných dat z infračerveného senzoru určit reálnou velikost a pozici parkovacího místa pro další softwarové zpracování.

##### **Navrhnout parkovací algoritmus pro řízení natáčení kol a pojezdu pro zaparkování do podélného parkovacího místa.**

##### **Ověřit funkčnost parkovacího algoritmu simulací na softwarovém тренаžéru naprogramovaném v prostředí MATLAB.**

##### **Provedení testu automatického parkování experimentálního vozidla Car4 v mechanické laboratoři FSI Vysokého učení technického v Brně.**

- Prostor pro parkování bude nalezen pomocí infračerveného senzoru.
- Softwarově se vybere vhodná trajektorie pro zaparkování.
- Vozidlo Car4 automaticky zaparkuje.

## 4. Řešení

Existuje mnoho možných způsobů jak navrhnout vhodnou trajektorii k parkování. V této práci byl vytvořen software, který zaznamenával trajektorie při parkování reálných osob. Poté byly zkušenosti se způsoby parkování použity při návrhu parkovacího algoritmu.

Tvorba tohoto softwaru je popsána v části 4.1.

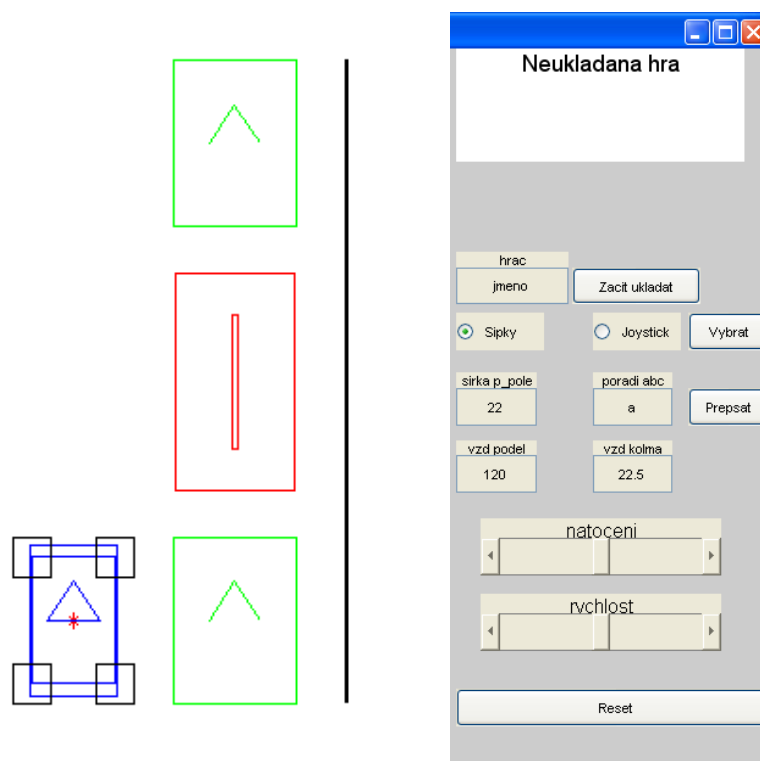
V části 4.2 je vysvětleno, jakým způsobem vozidlo sleduje zvolenou trajektorii a jaké jsou limitní podmínky pro zaparkování.

V části 4.3 je popsán parkovací algoritmus a jeho realizace v MATLABu.

Část 4.4 se věnuje úpravám algoritmu, který je nutný pro reálné parkování.

### 4.1. Software pro nahrávání trajektorií

Byl vytvořen software v Matlabu (obr. 4.1), ve kterém se mohou nasimulovat různorodé podmínky pro parkování, a který sloužil ke sbírání trajektorií. Autíčko se kterým se v programu parkovalo mělo rozměry Car4, které se nachází v mechatronické laboratoři. Tímto způsobem vznikl soubor, který obsahoval trajektorie pro různorodou škálu možných situací.



Obrázek 4.1: GUI

## 4.1. SOFTWARE PRO NAHRÁVÁNÍ TRAJEKTORIÍ

### 4.1.1. GUI

Jak název napovídá, jednalo se o uživatelské rozhraní, které mělo usnadnit ovládání celého programu. Skládalo se z obrazovky, kde byl zobrazen pohyb autíčka, a také z ovládací části, kde se měnily klíčové parametry. Klíčovými parametry je myšlena podélná vzdálenost mezi auty nebo například počáteční pozice, odkud auto začínalo parkovací manévr. K ovládání GUI mohl být použit joystick nebo šipky. Současně bylo v pravém horním rohu pole, které sloužilo jako zpětná vazba pro řidiče, aby věděl, jestli zvládnul zaparkovat s dostatečnou přesností. Také to sloužilo jako upozornění, pokud došlo ke kolizi v cvičném módu, kdy se automaticky nepřerušil celý běh programu. K uložení trajektorie došlo, pokud byly splněny zadané podmínky. Vzhledem k tomu, že nahrané trajektorie měly sloužit pro skutečné autíčko, byly tyto podmínky nastaveny celkem přísně.

Tolerance X směru  $\pm 3[\text{cm}]$

Tolerance Y směru  $\pm 13[\text{cm}]^*$

Tolerance natočení  $\pm 0.5 [^\circ]$

\* Tolerance v Y směru závisela na podélné velikosti parkovacího místa, tolerance  $\pm 13$  byla použita pro podélnou vzdálenost 120 cm.

### 4.1.2. Zobrazení

Byly použity dva souřadnicové systémy, které byly vzájemně posunuty a pootočený, což je vysvětleno v [21], a to lokální systém, ve kterém se natáčela kola, a globální, ve kterém probíhal samotný pohyb celého modelu. Pro oba byla základ matice rotace a posunutí.

Rovnice natočení v lokálním systému:

$$\begin{bmatrix} x_1^i & x_2^i & x_3^i & x_4^i \\ y_1^i & y_2^i & y_3^i & y_4^i \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} \cos \delta & -\sin \delta & 0 & X_{s1} \\ \sin \delta & \cos \delta & 0 & Y_{s2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Rovnice natočení v globálním systému:

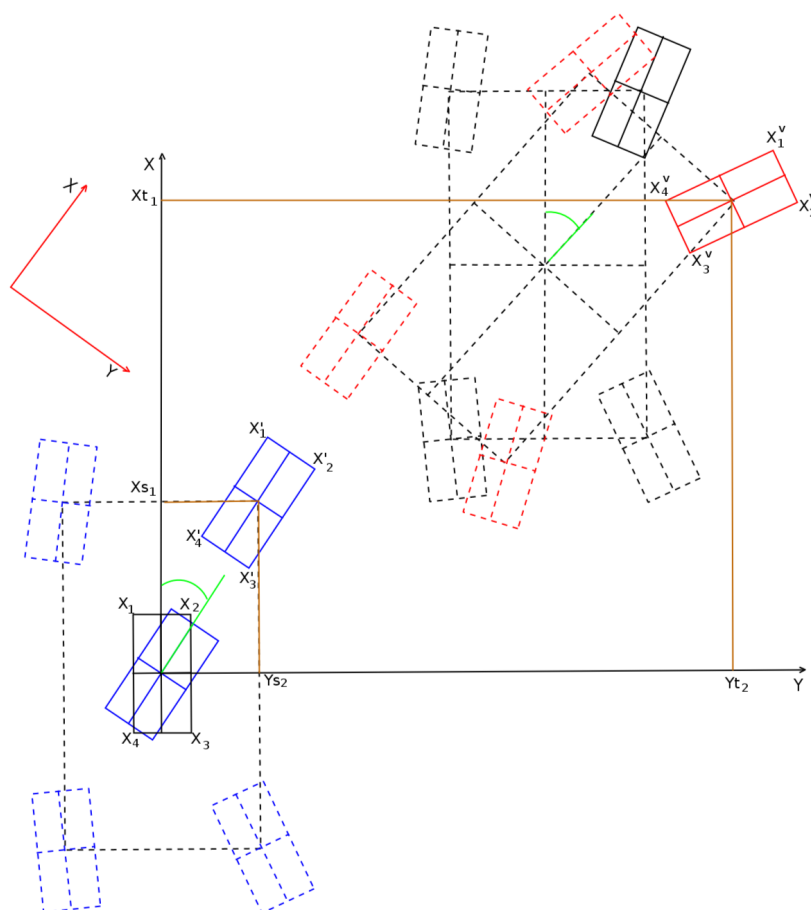
$$\begin{bmatrix} x_1^v & x_2^v & x_3^v & x_4^v \\ y_1^v & y_2^v & y_3^v & y_4^v \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} \cos \delta & -\sin \delta & 0 & X_{t1} \\ \sin \delta & \cos \delta & 0 & Y_{t2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x_1^i & x_2^i & x_3^i & x_4^i \\ y_1^i & y_2^i & y_3^i & y_4^i \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Lokální systém obstarával natáčení kol okolo místa uchycení na autíčku. Úhel, o jaký se měla jednotlivá kola natočit, byl vypočítán v části 4.1.5, tak aby splňovala Ackermannovu podmínku. V globálním systému pak došlo k natočení celého autíčka o úhel, jehož velikost se upravovala podle vztahu:

$$\Delta\varphi = \frac{v}{Y_l T_s} \quad (4.1)$$

kde  $\Delta\varphi$  je změna úhlu celkového natočení autíčka,  $v$  je rychlost,  $Y_l$  je vzdálenost středu otáčení od středu autíčka a  $T_s$  je doba běhu jednoho cyklu programu.





Obrázek 4.2: Zobrazení dvou vzájemně pootočených a posunutých souřadných systémů

Poté bylo autíčko posunuto na výslednou pozici , a to o vzdálenost, která mohla být dopočítána pomocí vzorce pohybu po kružnici:

$$\Delta x = vT_s \cos \varphi \quad (4.2)$$

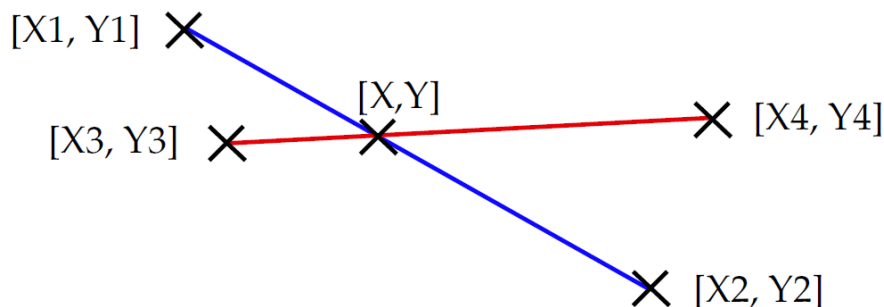
$$\Delta y = vT_s \sin \varphi \quad (4.3)$$

kde  $\Delta x$  a  $\Delta y$  jsou změny vzdálenosti v  $x$  a  $y$  směru, a  $\varphi$  je celkové natočení autíčka. V globálním systému bylo zobrazeno parkovací prostředí, čili okolní auta včetně zobrazeného parkovacího místa. Rozměry okolních aut byly nastaveny tak, aby se jednalo o identická auta s autem, se kterým se parkovalo. Jako překážka sloužila i stěna, jež byla umístěna podél aut, mezi které se parkovalo, čímž byla omezena možnost najetí na obrubník nebo odření auta. K odření by mohlo dojít, kdyby byl za simulovaným parkovacím místem ponechán velký manévrovací prostor, a ten není při reálném parkování vždy dostupný.

## 4.1. SOFTWARE PRO NAHRÁVÁNÍ TRAJEKTORIÍ

### 4.1.3. Kolize objektů

Kvůli způsobu sbírání dat bylo třeba, aby auto nemohlo projíždět překážkami. Proto všechny modely použité v softwaru byly rozděleny na úsečky.



Obrázek 4.3: Znázornění možné kolize v programu

Nejprve se vypočítaly koeficienty **a**, **b** funkce:

$$y = ax + b \quad (4.4)$$

která vyjadřuje rovnici přímky procházející body **1** a **2**. Po té se stejným postupem získaly koeficienty **c**, **d** druhé přímky procházející body **3** a **4**.

$$a = \frac{Y_2}{X_2 + Y_1 - X_1} \quad (4.5)$$

$$b = Y_1 - \frac{Y_2}{X_2 + Y_1 - X_1} X_1 \quad (4.6)$$

Pak se určil průsečík těchto přímek pomocí následujících rovnic.

$$X = \frac{d - b}{a - c} \quad (4.7)$$

$$Y = c \frac{d - b}{a - c} + d \quad (4.8)$$

Poté se ověřilo, jestli tento průsečík leží na daných úsečkách.

$$X \in \langle X_1, X_2 \rangle \cup \langle X_3, X_4 \rangle \\ Y \in \langle Y_2, Y_1 \rangle \cup \langle Y_3, Y_4 \rangle$$

Tento přístup byl použit pro všechny úsečky, které tvořily model auta, a pro modely všech okolních překážek. Pokud byla zaznamenána kolize, auto nemohlo pokračovat v jízdě a bylo nutné program restartovat.

### 4.1.4. Kinematický model

Byla zvažována varianta se středem křivosti pohybujícím se v rovině, což by vedlo k zajímavějším jízdním vlastnostem, nicméně by takové ovládání nemuselo být nutně přínosnější pro parkování a bylo by velmi obtížné (joystickem by se ovládala pozice středu

otáčení a šipkami směr jízdy). Takže byl použit kinematický model 4WS vozidla, ve kterém se střed křivosti pohybuje na ose rovnoběžné s nápravami umístěné uprostřed auta. Střed křivosti se pohybuje po ose pouze vně. I přesto, že model se středem křivosti ve středu auta by značně parkování zjednodušil, jak je uvedeno v 2.1, nebylo by možné tyto trajektorie použít, protože u skutečného vozidla jsme limitováni konstrukčním řešením, které tuto pozici středu otáčení nedovoluje.

Pokud auto dostane pokyn změnit směr, vypočítá střed otáčení a podle toho natočí všechna ostatní kola. Podle Ackermanovy podmínky a podle pohybu středu křivosti jde určit, že přední a zadní kolo na levé straně se natáčí o stejně velký úhel (jedou po pomyslné kružnici o stejném poloměru) a totéž platí pro pravou stranu auta.

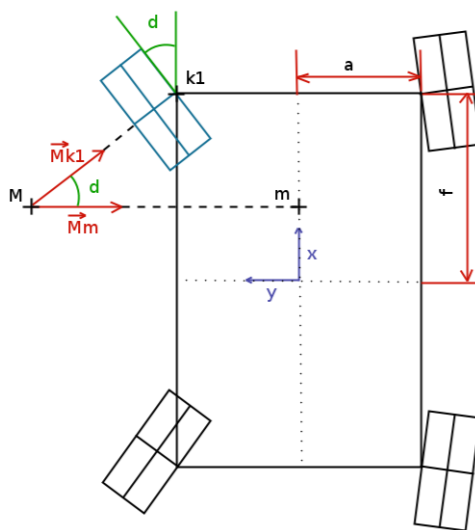
#### 4.1.5. Výpočet natočení kol

Výpočet byl proveden pomocí vektorů. První vektor  $\vec{M}_{k1} = (f - X_L)$ , měl počátek ve středu otáčení  $M = [X_L, Y_L]$  a konec ve středu prvního kola  $k_1 = [f, a]$ . Druhý vektor  $M_m = (0; -Y_L)$  směřoval od středu otáčení stále kolmo k x-ové ose (k bodu  $m = [X_L 0]$ ), pak se vypočítal úhel, který svírají tyto dva vektory.

$$\cos \delta = \frac{\vec{M}_m \vec{M}_{k1}}{|\vec{M}_m| |\vec{M}_{k1}|} \quad (4.9)$$

$$\delta = \cos^{-1} \left( \frac{(-Y_l)(a - Y_l)}{Y_l \sqrt{(f - X_l)^2 + (a - Y_l)^2}} \right) \quad (4.10)$$

Tento postup byl aplikován na všechna kola, a tak se zjistilo natočení jednotlivých kol pro libovolnou polohu středu otáčení.



Obrázek 4.4: Zobrazení použitých vektorů při výpočtu

## 4.2. VÝBĚR VHODNÝCH TRAJEKTORIÍ

### 4.1.6. Použití joysticku k ovládání GUI

Nejprve se použil příkaz "vrjoystick", aby program rozpoznal, že jde o připojený joystick k počítači.

```
joy = vrjoystick(1, 'forcedeedback');
```

Po té se pomocí příkazu "read(joy)" určilo, že má číst hodnoty z tohoto zařízení.

```
[axes, buttons, povs] = read(joy);
```

Dále už zbývalo jen určit, ve kterém směru osy se bude ovlivňovat daná proměnná x a ve kterém proměnná y. Tyto proměnné nabývaly hodnot od -100 do 100, což určovala velikost natočení joysticku.

```
y = axis(joy, 2);  
x = axis(joy, 1);
```

Poté se s tímto výstupem dalo pracovat jako s jakoukoliv jinou číselnou proměnnou.

```
v = (-y) * 70 ;  
del15= (x) * 0.320462;
```

## 4.2. Výběr vhodných trajektorií

### 4.2.1. Tabulka

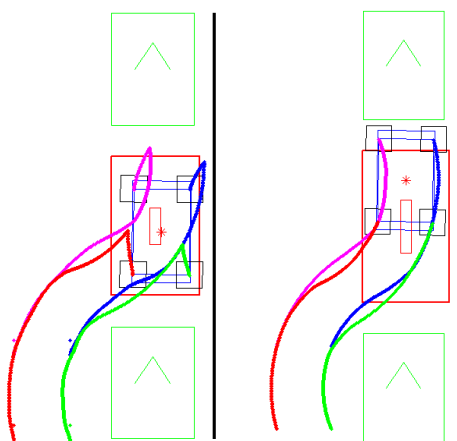
Výběr se provádí podle velikosti parkovacího místa a podle vzdálenosti, která je mezi bokem zaparkovaného auta a bokem parkujícího auta. Čím větší je velikost parkovacího místa, tím snáze se parkuje. Při malých parkovacích místech může dojít k problémům spojených s těsným kopírováním okolních překážek stejně jako k problémům spojených s vícenásobným popojížděním do požadované pozice. Reálné parkování není tak dokonalé jako v simulaci, takže je výhodnější používat rozdělení podle několika různých délek parkovacích míst, a tím snížit vliv možných nepřesností na výsledek parkování. Druhým parametrem je proměnlivá boční vzdálenost, protože parkování nikdy nezačíná ze stejné výchozí pozice, proto bylo potřeba tuto proměnlivou vzdálenost vykompenzovat. Jako nejlepší řešení se ukázal soubor trajektorií nahraných pro jednu velikost parkovacího místa s různými bočními vzdálenostmi. Současně s nimi se mění i výchozí pozice v podélném směru, protože je parkování z dané vzdálenosti jednodušší.

Tabulka 4.1: Výběr vhodné trajektorie podle boční vzdálenosti vozidla k parkovacímu místu a podle délky parkovacího místa

boční vzdálenost	délka p.m. >115	délka p.m.>125
$x > 15 \ \& \ x < 17,5$	trajektorie-115-1	trajektorie-125-1
$x > 17,5 \ \& \ x < 20$	trajektorie-115-2	trajektorie-125-2
$x > 20 \ \& \ x < 22,5$	trajektorie-115-3	trajektorie-125-3
$x > 22,5 \ \& \ x < 25$	trajektorie-115-4	trajektorie-125-4
$x > 25 \ \& \ x < 27,5$	trajektorie-115-5	trajektorie-125-5
$x > 27,5 \ \& \ x < 30$	trajektorie-115-6	trajektorie-125-6
$x > 32,5 \ \& \ x < 35$	trajektorie-115-7	trajektorie-125-7
$x > 35 \ \& \ x < 37,5$	trajektorie-115-8	trajektorie-125-8
$x > 37,5 \ \& \ x < 40$	trajektorie-115-9	trajektorie-125-9

#### 4.2.2. Nahrávání trajektorií v mechatronické laboratoři

Doba, po kterou mohl kdokoliv přijít a zkusit zaparkovat v simulaci s autem, které mělo rozměry Car4, trvala asi 14 dní. Za toto období se nahrálo zhruba 50 trajektorií od 11 různých lidí. Parkování do podélného parkovacího místa není jednoduché, obzvláště když se jedná o 4WS auto, které jste nikdy předtím neřídili. Takže většina nahraných trajektorií byla zbytečně komplikovaná. I když se podařilo zaparkovat, vyžadovalo to víc změn směru jízdy, než bylo nezbytně nutné. Dalším důvodem, proč tyto trajektorie nakonec nebyly použity, bylo i to, že se mi nepodařilo rozumněji vyřešit problém s proměnlivou boční vzdáleností, než souborem několika trajektorií s různou boční vzdáleností. Přičemž všechny trajektorie nahrané v mechatronické laboratoři byly nahrány pro jednu boční vzdálenost, protože v té době takhle varianta řešení vypadala realizovatelně.



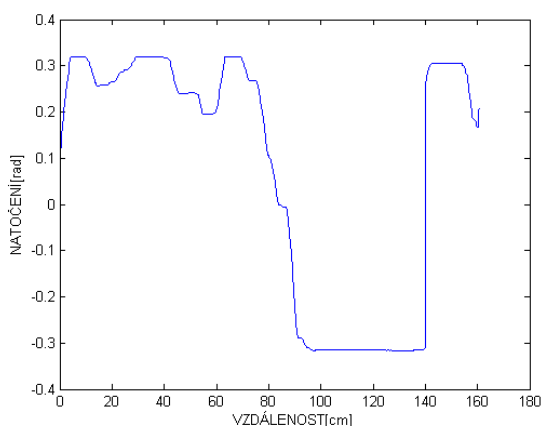
Obrázek 4.5: Příklady použitých trajektorií

### 4.3. ALGORITMUS PARKOVÁNÍ A JEHO SIMULACE

#### 4.2.3. Úprava nasbíraných dat

Perioda s jakou program zaznamenával měřené údaje, byla  $t=0.02s$ . Pro každou trajektorii tímto způsobem vznikl soubor dat, kde pro všechny měřené veličiny bylo zaznamenáno 600-1500 hodnot. Kolik dat bylo naměřeno, záviselo na rychlosti parkování. Získaná data bylo třeba upravit, aby je bylo možné použít pro Car4. Zvolila jsem přístup, kde se kola natáčí v závislosti na již ujeté vzdálenosti. Ujetá vzdálenost se vypočítala podle jednoduchého vzorce pro vzdálenost dvou bodů:

$$\Delta s = \sqrt{|X_{n+1} - X_n|^2 + |Y_{n+1} - Y_n|^2} \quad (4.11)$$



Obrázek 4.6: Změna natočení podle ujeté vzdálenosti

### 4.3. Algoritmus parkování a jeho simulace

Tento algoritmus byl vytvořen na principu stavového automatu, aby bylo možné parkování zobrazit, byl použit předchozí software pro nahrávání trajektorií, jako nástroj pro zobrazení parkování.

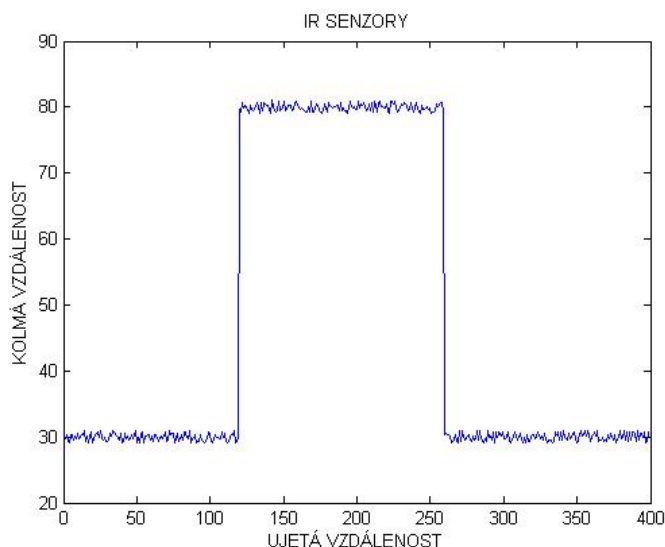
#### 4.3.1. Popis algoritmu

Jedná se o velmi jednoduchý stavový automat se čtyřmi stavy. V první fázi parkování se měří velikost parkovacího místa. Pokud je takové místo nalezeno a vyhovuje minimálním požadavkům pro úspěšné zaparkování, přejde se k fázi dvě. V tuto chvíli se změni směr jízdy a na základě trajektorie vybrané ve fázi jedna se určí, o kolik se má změnit poloha autíčka vůči současné poloze, protože kvůli různé vzdálenosti od boku auta se mění i vzdálenost autíčka od počátku parkovacího místa v podélném směru. Po té se přejde ke třetí fázi, kde začíná samotné parkování.

Tato simulace vznikla z softwaru pro nahrání trajektorií, kde se odsimulovaly první dvě fáze a ve třetí fázi byl nahrazen výstup z joysticku řízením natočení kol v závislosti na již ujeté vzdálenosti. Autíčko parkovalo konstantní rychlostí a znamínko rychlosti ve třetí fázi bylo zjištěno díky interpolaci znamínka rychlosti v závislosti na uražené vzdálenosti z uložených dat. Tímto způsobem autíčko v simulaci parkovalo maximálně pár milimetrů vzdálené od požadované pozice, která byla nahrána v SW pro sběr trajektorií v části 4.1.

### 4.3.2. Simulace IR senzoru

Senzor byl umístěn ve středu auta a simulovaný výstup ze senzoru byl vytvořen pomocí vektoru, kde čísla nabývala tři nominálních hodnot pro vytvoření překážek a jejich dokonale konstantní hodnoty byly rozkmitány pomocí příkazu „rand“, aby se co nejvíce přiblížily reálnému výstupu ze senzoru.



Obrázek 4.7: Hodnoty použité jako výstup z IR senzoru v simulaci

### 4.3.3. Hledání hran

Srovnávají se průměrné hodnoty dvou skupin pěti bodů jdoucích po sobě. Pokud se stane, že rozdíl těchto skupin je větší než 8 cm, program si uloží hodnotu doposud ujeté vzdálenosti jako první hranu. Zároveň také začne ignorovat další možné nalezené hrany po následujících 7 měřeních, aby nedošlo k zaznamenání několika hran pouze během měření hrany jedné. Po té se obdobným způsobem zjistí pozice druhé hrany.

## 4.4. Reálný experiment

Algoritmus navržený a odsimulovaný v předchozí části, sloužil jako předloha, která se upravila, tak aby bylo možné algoritmus použít pro řízení Car4. Předchozí algoritmus měl konstantní rychlost, ale při reálném parkování se lepších výsledků dosáhne při pozvolném rozjíždění a zpomalování. Rychlost vozidla byla řízena přes P-regulátor, který ujetou vzdálenost reguloval změnou střídy na DC motorech. Zároveň se upravovala část programu zpracovávající výstup z infračerveného senzoru, protože měl pozvolnější reakci při přechodu přes hranu překážky(okraj parkovacího místa).

### 4.4.1. Proměnlivá rychlost

Pokud je vzdálenost, kterou má autíčko ujet, kratší než součet vzdálenosti pro zrychlení a pro zpomalení, je maximální rychlost snížena a pro dosažení nové maximální rychlosti

#### 4.4. REÁLNÝ EXPERIMENT

jsou použity trochu upravené křivky, z první a třetí fáze křivky rychlosti. Tato rychlost byla popsána v části 2.3 pro ideální vzdálenosti.

Velikost úseku, o který byla ochuzena první i poslední fáze je  $l_n$ , které se vypočítá následovně:

$$l_n = \frac{2L_w - l}{2} \quad (4.12)$$

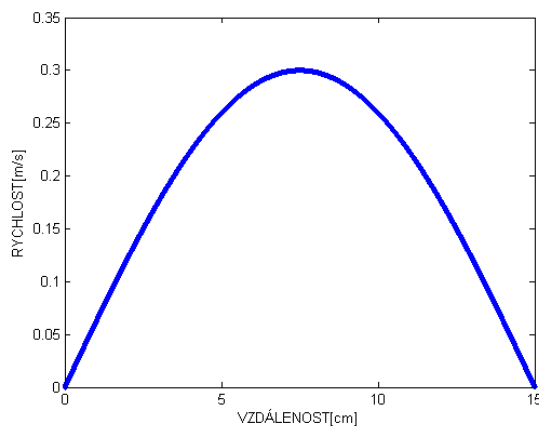
Kde  $l$  je celková velikost trasy,  $l_w$  je vzdálenost vhodná pro rozjetí/zpomalení. Současně byla také snížena  $V_{max}$ , tak aby to odpovídalo poměru vzdáleností:

$$v_{max2} = v_{max1} \frac{l}{l_w + l_w} \quad (4.13)$$

Výpočet rychlosti probíhal podle níže uvedených vzorců:

$$v = \begin{cases} 0 < s_d < l_w - l_n, & v_{max2} \sin\left(\frac{s_d}{2(l_w - l_n)}\pi\right) \\ l_w - l_n < s_d < l, & v_{max2} \sin\left(\frac{s_d - (l - 2(l_w - l_n))}{2(l_w - l_n)}\pi\right) \end{cases}$$

Kde  $s_d$  je ujetá vzdálenost.



Obrázek 4.8: Upravená rychlost pro malé vzdálenosti

#### 4.4.2. P regulátor

Jednalo se o jednoduchý P-regulátor, který sloužil k řízení ujeté vzdálenosti. Protože byla naplánována požadovaná rychlost v závislosti na ujeté vzdálenosti, bylo potřeba ji předělat na ujetou vzdálenost v určitém čase. Čas mezi jednotlivými body se vypočítal jako:

$$\Delta t = \frac{\Delta s}{v} \quad (4.14)$$

Kde  $\Delta s$  je vzdálenost ujetá mezi jednotlivými body,  $v$  je rychlost v dané chvíli a  $\Delta t$  je doba, za jakou se daná vzdálenost ujela.

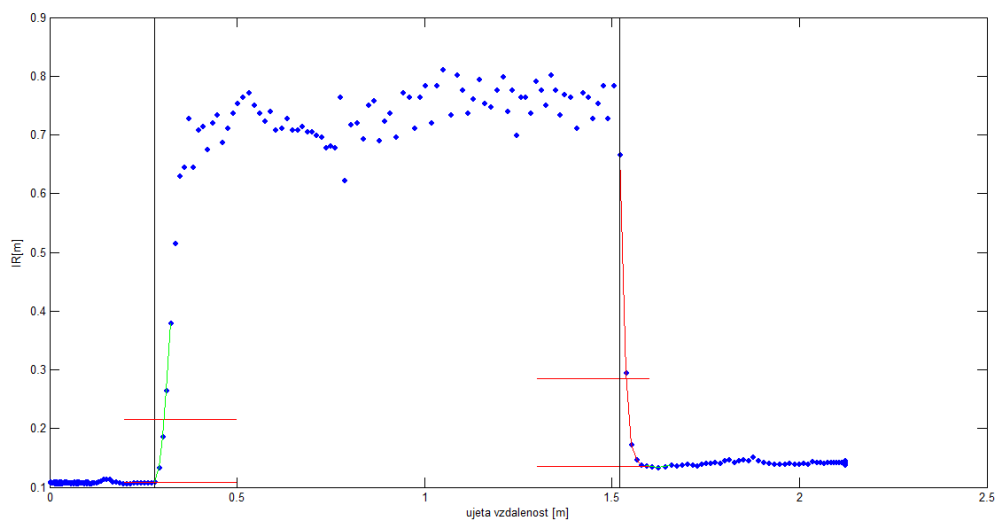
Tímto jednoduchým vzorcem se dřívější křivka přepočítala.



### 4.4.3. Měření IR senzoru

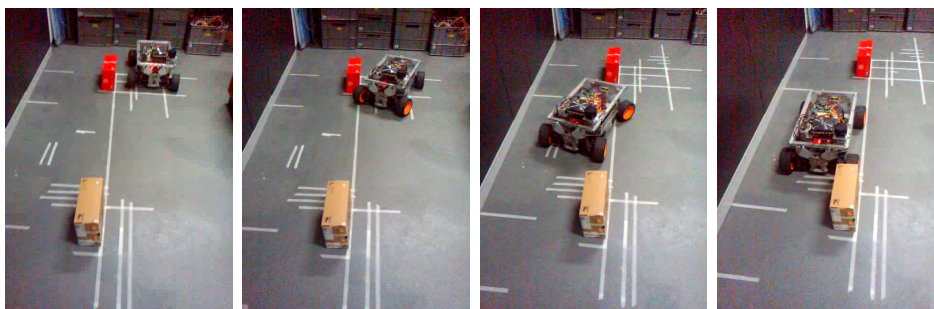
Hledání hran je popsáno v části 4.3.3, ale pro skutečné parkování s Car4 musel být program upraven, aby odpovídal skutečnosti co nejlépe. Takže potom, co jsou nalezeny obě hrany, program počká 10 hodnot (0.5 s) a zpětně hranu vyhledá z druhé strany, protože při měření mezery mezi auty je vzdálenost, kterou IR senzory zaznamenávají, velká (0.7 - 0.9 m) a kvůli tomu dochází k výraznějšímu šumu, který ovlivňuje přesnost odchycení hrany v rozmezí 2 cm. Tím, že se zpětně vyhledává druhá hrana, se detekce hran stává přesnější.

Program roztřídí hodnoty vzdálenosti z IR senzorů do tří skupin, na ty které byly naměřeny odrazem IR světla od prvního auta, od druhého auta a při měření šířky parkovacího místa, od stěny. Tyto soubory hodnot byly ochuzeny o části vyskytující se v blízkosti hran, protože by tím docházelo k zbytečným nepřesnostem, a potom byly zprůměrovány. Měla by být splněna podmínka, že vozítko jede rovně, ale i tak může být rozdíl v měřených hodnotách z IR u jednotlivých aut pár centimetrů.



Obrázek 4.9: Příklad výstupu z IR senzorů v závislosti na ujeté vzdálenosti během první fáze parkování

### 4.4.4. Záznam z experimentu



Obrázek 4.10: Záznam z parkovacího manévru, kde ty nejdůležitější části, ve kterých by mohlo dojít ke střetu s překážkou, byly vymezené pomocí krabic

## 5. Závěr

Cílem této práce bylo vytvoření parkovacího algoritmu pro podélné parkování. V řešeršní části byly shrnuty poznatky o kinematice 4WS vozidla, především různé módy řízení, popis vývoje a současného stavu využívání 4WS v automobilovém průmyslu. Dále zde je uveden přehled parkovacích senzorů, které se vyskytují u automobilů, ať už v kombinaci s parkovacím asistentem, nebo bez něj.

Samotné řešení je rozděleno do více částí. V první části se nachází popis softwaru pro nahrání trajektorií. Obsahuje matematické přístupy použité pro zobrazení, kolizi objektů, kinematický model 4WS.

V další části dochází k objasnění výběru vhodné křivky pro zaparkování na základě tabulky trajektorií. Nahrávání trajektorií nebylo úplně snadné, protože 4WS automobil reaguje rozdílně oproti 2WS, na které je drtivá většina řidičů zvyklá. A poté, co se změnila tolerance pro uložení trajektorie, především tolerance výsledného natočení auta z  $\pm 3^\circ$  na  $\pm 0.5^\circ$ , by z tohoto parkovacího programu byla velmi neoblíbená počítačová hra. K této změně došlo až po prvních pokusech skutečného parkování, kde se ukázalo, jak nehezky by ty  $3^\circ$  vypadaly, umocněné nepřesnostmi vznikajícími při skutečném parkování. Současně je zde popsán způsob sledování vybrané trajektorie pomocí čtení vhodného natočení podle ujeté vzdálenosti automobilu.

Ve třetí části je vysvětleno, jakým způsobem pracuje parkovací algoritmus, a to především, že se jedná o jednoduchý stavový automat se 4 stavy. V prvním stavu dochází k zpracování simulovaného výstupu z infračervených senzorů a následnému nalezení parkovacího místa. V druhém stavu si automobil najede do požadované výchozí pozice. Ve třetím stavu probíhá samotný parkovací manévr, podle zvolené trajektorie. V posledním stavu se automobil zastaví a ukončí parkování. Tento algoritmus byl odsimulován v prostředí Matlabu.

Čtvrtá část se věnuje upravenému algoritmu pro vyzkoušení na Car4, jedná se o úpravu rychlosti pro plynulé rozjetí a zastavení Car4. Proběhla úprava zpracování výstupu z IR senzorů. Dále se do algoritmu zapracoval P-regulátor, který reguloval ujetou vzdálenost Car4.

Cíl analyzovat způsob parkování různých fyzických osob byl splněn, nicméně výsledné trajektorie byly použity pouze jako hrubý podklad pro řešení parkovacího asistenta. Efektivita parkování testovaných osob nebyla vysoká, zjistilo se, že je nutné oslovit řidiče s většími zkušenostmi s řízením 4WS vozů. Software pro simulaci parkování se naopak nad očekávání osvědčil při vizuální kontrole navrženého algoritmu pro parkovacího asistenta.

Použití infračerveného senzoru pro zjištění rozměrů parkovacího místa se ukázalo jako vhodné řešení, nicméně bylo nutno se vypořádat s technickými problémy. Při vzdálenosti na hranici měřitelnosti IR senzorů byl v datech velký šum, což snižovalo přesnost odchycení hrany. Současně se při parkování musela stále používat stejná barva překážek, jinak by rozdíl naměřené vzdálenosti mohl tvořit až několik centimetrů. Kola měla sama o sobě vůli, což mohlo způsobovat nepřesnosti v ujeté vzdálenosti  $\pm 0,75$  cm.

Ověření funkčnosti navrženého algoritmu automatického parkování proběhlo během provedení testu automatického parkování experimentálního vozidla Car4 v mechanotro-

nické laboratoři FSI Vysokého učení technického v Brně. Byl automaticky nalezen prostor pro parkování pomocí infračerveného senzoru, softwarově byla vybrána vhodná trajektorie pro zaparkování a vozidlo Car4 následně automaticky zaparkovalo.

Minimální rozměry parkovacího místa, kde Car4 ještě dokáže zaparkovat, jsou 115 cm na délku a 60 cm na šířku. Tohle platí za předpokladu, že nemá přední a zadní nárazník, které jsou odmontovatelné a jejich instalací získá o 40 cm více délky, čímž se zvětší i minimální parkovací místo. Tyto výsledky se výrazně liší od simulovaných parkovacích míst, kde bylo možné zaparkovat na 90-95 cm, a to protože se jednalo o počítačovou simulaci, kde vznikaly jen velmi malé nepřesnosti při mnohonásobném popojíždění automobilu.

## 6. Literatura

- [1] BIGELON P.: Drivers are better off letting cars park themselves. In: *autoblog*[online]. 2015[cit. 2016-04-20] Dostupné z:<http://www.autoblog.com/2015/09/22/drivers-are-better-off-letting-cars-park-themselves/>
- [2] BOERIU H.: The new key fob for BMW 7 Series activates Remote Control Parking. In: *bmwblog*[online]. 2015 [cit. 2016-05-10]. Dostupné z:<http://www.bmwblog.com/2015/06/10/the-new-key-fob-for-bmw-7-series-activates-remote-control-parking/#comments>
- [3] BOERIU H.: Parking Assistant and Driving Assistant Plus including Traffic Jam Assistant. In: *bmwblog*[online]. 2013 [cit. 2016-05-10]. Dostupné z:<http://www.bmwblog.com/2013/09/19/parking-assistant-driving-assistant-plus-including-traffic-jam-assistant/>
- [4] CHENG, K., Y. ZHANG, H. CHEN: Planning and Control for a Fully-automatic Parallel Parking Assist System in Narrow Parking Spaces. In: *IEEE*[online]. 2013[cit. 2016-05-10]. Dostupné z:<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6629669>
- [5] JIANG, K., L. Seneviratne: A sensor guided autonomous parking system for nonholonomic mobile robots. In: *IEEE*[online]. 1999[cit. 2016-05-10]. Dostupné z:<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=769997>
- [6] IEEE Xplore:Digital Library. [online].Dostupné z:<http://ieeexplore.ieee.org/Xplore/home.jsp>
- [7] JASANSKÝ M.: *Návrh dynamických modelů pro řízení trakce experimentálního vozidla*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2010. 123 s. Vedoucí diplomové práce Ing. Robert Grepl, Ph.D.
- [8] LIN, Chih-Jui, et al.: Design and Implementation of 4WS4WD Mobile Robot and Its Control Applications. In: *IEEE*[online]. 2013[cit. 2016-05-10]. Dostupné z:<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6614666>
- [9] BHISHIKAR, Saket, et al.: Design and Simulation of 4 Wheel Steering System. In: *IJEIT*[online]. 2014[cit. 2016-05-10]. ISSN: 2277-3754. Dostupné z:[http://www.ijeit.com/Vol%203/Issue%2012/IJEIT1412201406\\_65.pdf](http://www.ijeit.com/Vol%203/Issue%2012/IJEIT1412201406_65.pdf)  
<http://what-when-how.com/automobile/four-wheel-steering-4ws-automobile/>
- [10] Four-wheel Steering (4WS) (Automobile). In: *what-when-how.com*[online]. [cit. 2016-05-10]. Dostupné z:<http://what-when-how.com/automobile/four-wheel-steering-4ws-automobile/>
- [11] Steering Angle Sensing Four-Wheel Steering System(4WS). In: *world.honda.com*[online]. [cit. 2016-05-10]. Dostupné z:<http://world.honda.com/history/challenge/19874ws/>

- [12] SEVERSON A.: Four-wheel steering demystified. In:<http://autoweek.com/>[online]. 2015[cit. 2016-04-20] Dostupné z:<http://autoweek.com/article/car-life/four-wheel-steering-demystified>
- [13] What is an IR Sensor?. *Carnegie Mellon's Robotics Academy*[online]. [cit. 2016-05-05] Dostupné z:[http://education.rec.ri.cmu.edu/content/electronics/boe/ir\\_sensor/1.html](http://education.rec.ri.cmu.edu/content/electronics/boe/ir_sensor/1.html)
- [14] Sharp-GP2Y0A21YK0F. *sharpsma.com*[online]. 2006[cit. 2016-05-10] Dostupné z:[http://www.sharpsma.com/webfm\\_send/1489](http://www.sharpsma.com/webfm_send/1489)
- [15] MUNENORI, Hikita: An introduction to ultrasonic sensors for vehicle parking. In:[newelectronics.co.uk](http://www.newelectronics.co.uk)[online]. 2010[cit. 2016-05-10] Dostupné z:<http://www.newelectronics.co.uk/electronics-technology/an-introduction-to-ultrasonic-sensors-for-vehicle-parking/24966/>
- [16] Auto park assist. *audi-kozhikode.com*[online]. [cit. 2016-05-10] Dostupné z:[http://www.audi-kozhikode.com/in\\_partner/p\\_30216/en/models/q7/audi\\_q7/equipments/Audi-Parking-system-plus-with-rear-view-camera.html](http://www.audi-kozhikode.com/in_partner/p_30216/en/models/q7/audi_q7/equipments/Audi-Parking-system-plus-with-rear-view-camera.html)
- [17] PD1 Parking Dynamics Features & Advantages. *parkingdynamics.co.uk*[online].2008 [cit. 2016-05-10] Dostupné z:<http://www.parkingdynamics.co.uk/Features#advantages>
- [18] Beckwith, Jimi: Drivers embrace Ford reversing camera. In:[carbuyer.co.uk](http://www.carbuyer.co.uk)[online]. 2013[cit. 2016-05-10] Dostupné z:<http://www.carbuyer.co.uk/news/86674/drivers-embrace-ford-reversing-camera>
- [19] Around View Monitor. *nissan-global.com*[online].[cit. 2016-05-10] Dostupné z:<http://www.nissan-global.com/EN/TECHNOLOGY/OVERVIEW/avm.html>
- [20] Dolní propust. *cs.wikipedia.org*[online]2014.[cit. 2016-05-10] Dostupné z:[https://cs.wikipedia.org/wiki/Doln%C3%AD\\_propust](https://cs.wikipedia.org/wiki/Doln%C3%AD_propust)
- [21] GREPL,Robert.*Kinematika a dynamika mechatronických systémů*. 1.vyd. Brno:CERM, 2007. ISBN 978-80-214-3530-8.
- [22] GREPL, Robert, et al.: Development of 4WS/4WD Experimental Vehicle: platform for research and education in mechatronics. In:*IEEE*[online]. 2011[cit. 2016-05-10]. Dostupné z:<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5971241>
- [23] HANLON, Mike: Electronic four-wheel steering from Renault. In:[gizmag.com](http://www.gizmag.com)[online]. 2008[cit. 2016-05-10] Dostupné z:<http://www.gizmag.com/electronic-four-wheel-steering-from-renault/9133/>
- [24] AUSTIN, Michael: The Return of Four-Wheel Steering. In:[popularmechanics.com](http://www.popularmechanics.com)[online]. 2013[cit. 2016-05-10] Dostupné z:<http://www.popularmechanics.com/cars/a9862/the-return-of-four-wheel-steering-16311550/>

## LITERATURA

- [25] HALVORSON, Bengt: 2017 Audi Q7: Driving The Only New SUV With All-Wheel Steering. In: *motorauthority.com*[online]. 2015[cit. 2016-05-10] Dostupné z:[http://www.motorauthority.com/news/1101476\\_2017-audi-q7-driving-the-only-new-suv-with-all-wheel-steering](http://www.motorauthority.com/news/1101476_2017-audi-q7-driving-the-only-new-suv-with-all-wheel-steering)
- [26] BENET, G., F. Blanes, J.E. Simó, P.Pérez: Using infrared sensors for distance measurement in mobile robots. In: *ELSEVIER*[online]. 2002[cit. 2016-05-10]. Dostupné z:<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.109.5864&rep=rep1&type=pdf>
- [27] Rear-View Cameras Reviews. *toptenreviews.com*[online].2016 [cit. 2016-05-10] Dostupné z:<http://rear-view-cameras-review.toptenreviews.com/>

## 7. Seznam použitých zkratek

4WS	Four wheel steering (vozidlo s dvěma natáčecími nápravami)
2WS	Two wheel steering (vozidlo s natáčecí přední nápravou)
4WD	Four wheel drive (vozidlo s čtyřmi poháněnými koly)
IR senzory	Infrared senzory (Infračervené senzory)
GUI	Graphical user interface (Grafické uživatelské rozhraní)
UART	Universal Asynchronous Receiver/Transmitter (Univerzální asynchronní sériové rozhraní)

## 8. Seznam příloh

Přílohy na CD

- GUI
- Parkovací Algoritmus
- Simulace Parkovacího manévru
- Soubor trajektorií