VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A BIOMECHANIKY

FACULTY OF MECHANICAL ENGINEERING

INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS

PLÁNOVÁNÍ TRASY PRO PODŘÍZENÉ MOBILNÍ JEDNOTKY SEMIAUTONOMNÍHO KONVOJE

PATH PLANNING FOR SEMIAUTONOMOUS CONVOY SLAVE UNITS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE — MILAN HURBAN

AUTHOR

VEDOUCÍ PRÁCE — doc. Ing. JIŘÍ KREJSA, Ph.D.

SUPERVISOR

BRNO 2014

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav mechaniky těles, mechatroniky a biomechaniky
Akademický rok: 2013/14

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

student(ka): Milan Hurban

který/která studuje v **bakalářském studijním programu**

obor: **Mechatronika (3906R001)**

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:

Plánování trasy pro podřízené mobilní jednotky semiautonomního konvoje

v anglickém jazyce:

Path planning for semiautonomous convoy slave units

Stručná charakteristika problematiky úkolu:

Hlavním cílem práce je návrh a realizace metody plánováni trasy pro mobilní jednotky semiautonomního konvoje. Metoda plánování trasy by měla zajistit sledování trajektorie předchozí mobilní jednotky v rámci konvoje. Součástí práce by měla být i simulace kompletního konvoje s implementovanou metodou plánování podřízených jednotek a manuálním řízením čelní jednotky. Výsledná aplikace by měla být snadno implementovatelná do robustnější aplikace, která bude zajišťovat funkci vysokoúrovňových systémů každé mobilní jednotky semiautonomního konvoje.

Cíle bakalářské práce:

1. prostudujte metody plánování trasy pro mobilní roboty s důrazem na podřízené jednotky robotického konvoje
2. navrhněte metodu pro plánování trasy v rámci semiautonomního konvoje,
3. výsledné řešení implementujte a simulačně ověřte.

Seznam odborné literatury:

S.LaValle: Planning algorithms, Cambridge University Press, 2006
K.C.Ng, M.M.Trivedy: A neuro-fuzzy controller for mobile robot navigation and multirobot convoying, Systems, Man, and Cybernetics, Part B, 1998

Vedoucí bakalářské práce: doc. Ing. Jiří Krejsa, Ph.D.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2013/14.

V Brně, dne 4.11.2013

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA STROJNÍHO INŽENÝRSTVÍ
DĚKANÁT

L.S.

prof. Ing. Jindřich Petruška, CSc.
Ředitel ústavu

prof. RNDr. Miroslav Doupovec, CSc., dr. h. c.
Děkan

Abstrakt

Bakalářská práce se zabývá návrhem metody plánování trasy pro podřízené mobilní jednotky semiautonomního konvoje. Cílem je především na základě dat ze zpracování obrazu kamery zajistit, aby jednotka sestavila a následně projela trajektorii kopírující pohyb předchozí jednotky v rámci konvoje. Součástí práce je simulace kompletního konvoje v programu MATLAB s implementovanou metodou plánování trajektorie podřízených jednotek a manuálním řízením čelní jednotky. Výsledný program je připraven k implementaci do robustní aplikace, kde bude zajišťovat funkci vysokoúrovňových systémů každé mobilní jednotky semiautonomního konvoje.

Klíčová slova

Semiautonomní konvoj, plánování trajektorie

Abstract

This bachelor thesis deals with the design of a trajectory planning method for semiautonomous convoy slave units. The main objective is to ensure that the unit calculates and follows a trajectory that closely matches the leading unit based mainly on image processing data from the camera. The thesis includes a MATLAB simulation of the whole convoy with manual control of the leading vehicle and an implemented path planning algorithm. The resulting code is ready to be integrated into a robust application where it will provide high-level control of each unit of the semiautonomous convoy.

Keywords

Semiautonomous convoy, trajectory planning

Bibliografická citace

HURBAN, M. Plánování trasy pro podřízené mobilní jednotky semiautonomního konvoje. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2014. 39 s. Vedoucí bakalářské práce doc. Ing. Jiří Krejsa, Ph.D.

Poděkování

Děkuji tímto doc. Ing. Jiřímu Krejsovi, Ph.D. za cenné připomínky a rady při vypracování bakalářské práce.

Čestné prohlášení

Prohlašuji, že jsem bakalářskou práci na téma „*Plánování trasy pro podřízené jednotky semiautonomního konvoje*“ vypracoval samostatně pod vedením vedoucího bakalářské práce doc. Ing. Jiřího Krejsy, Ph.D. a s použitím odborné literatury a pramenů uvedených v seznamu, který tvoří přílohu této práce.

V Brně dne 30. 5. 2014 .....

Milan Hurban

Obsah

1 Úvod

Autonomní řízení vozidel je v současné době odvětvím zažívajícím značný rozvoj. Pravděpodobně největší uplatnění by taková vozidla nalezla v systémech automatizovaných dálnic, a to jak pro nákladní, tak osobní dopravu. V případě nákladní dopravy se jako atraktivní jeví zejména schopnost provozovat vozidlo bez potřeby dbát na únavu řidiče, nehledě na plat a jiné náklady spojené se zaměstnáváním lidí. V osobní dopravě by využití spočívalo především ve zvýšení bezpečnosti přepravy. Již dnes má řidič lepšího auta k dispozici množství asistenčních systémů, které kompenzují hlavní omezení lidských řidičů jako jsou reakční doba, omezená znalost stavu vozidla v každém okamžiku, či prostá nezkušenost. Při správném navržení autonomního vozidla by bylo veškeré řízení obstaráváno výpočetní technikou a byl by tak ze systému vyřazen člověk, který je obvykle nejslabším článkem.

Zajímavým konceptem se jeví řazení takových autonomních jednotek do formací, především do konvojů. Inteligentní řízení a komunikace mezi jednotkami by umožnily jízdu v řadě za sebou s minimálními rozestupy. Předáváním informací o zrychlování/zpomalování by se fronty na křižovatkách rozjížděly a zastavovaly v jeden okamžik, čímž by se zlepšila plynulost provozu. Zároveň kompaktnost takových formací vede k ušetření místa na cestách, kterého je dnes čím dál méně. V neposlední řadě se také jízdou těsně za sebou šetří palivo kvůli výhodným aerodynamickým vlastnostem takového uspořádání.

Jedním z hlavních problémů, které musí současné projekty řešit, je získání a zpracování potřebných dat z okolí. Je možné, že v budoucnosti budou i samotné dálnice vybaveny řadou elektronických senzorů a komunikačních prostředků a celý systém bude fungovat jako jeden celek, ale dnešní projekty, chtějí-li být úspěšné, musí být navrhované na současný stav cest. Autonomní vozidlo tak musí obsahovat veškerou senzoriku, komunikační a výpočetní techniku potřebnou ke zjištění svého stavu a naplánování vlastní trajektorie.

Hlavním cílem této práce je navrhnout a realizovat takovou metodu plánování trajektorie, která by odpovídala elektronickému vybavení naší autonomní podřízené jednotky v rámci semiautonomního konvoje. Tento konvoj sestává z vedoucí jednotky ovládané člověkem a z alespoň jedné podřízené jednotky, jejímž jediným cílem je následovat vedoucí jednotku a pokud možno co nejlépe při tom kopírovat její trajektorii. V následujících kapitolách jsou popsány vybrané prostudované, již existující metody, senzorika a výpočetní vybavení naší jednotky, zvolená metoda s ohledem na vhodnost k naší situaci, implementace do reálného vozidla a simulace v programu MATLAB, sloužící k testování různých metod a ověření experimentálních výsledků.

2 Rešeršní studie metod plánování trajektorie

K získání přehledu o již vyvinutých metodách plánování trasy bylo nastudováno několik akademických článků, přičemž důraz byl kladen na ty, jejichž výsledky jsou aplikovatelné na naši situaci, viz popis systému v kapitole 3.

2.1 Základní informace

Navigací robotů se obecně zabývají odvětví robotiky, teorie řízení a umělé inteligence. Robotika se zabývá algoritmickým převáděním vysokoúrovňových požadavků na nízkoúrovňové úkony, jako je například ovládání aktuátorů. Teorie řízení řeší především otázky dynamiky, zpětnovazebních smyček, stability a ovladatelnosti spojitých systémů. Umělá inteligence se snaží o matematický popis řešení problémů, se kterými se roboti mohou při navigaci setkat. [12]

Velká část plánování trasy je zaměřená na roboty, kteří se pohybují v místnostech či jiných uzavřených prostorech. Klasickými problémy jsou v takových případech například:

- Naplánování co nejkratší trasy spojující dva body v prostředí
- Detekce a objíždění překážek
- Plánování s omezeními

Jak je podrobněji rozebráno v kapitole 3, jednotky našeho konvoje jsou čtyřkolová vozidla s automobilovým podvozkem. Taková vozidla se označují jako neholonomní. Při plánování trasy se na ně tak vztahují diferenciální omezení - vozidlo se například nedokáže otočit na místě.

Náš konvoj se ale hlavně oproti typickým případům pohybu v místnostech a známém prostředí vyznačuje tím, že je navržen k pohybu bez potřeby lokalizace a mapování okolí. Vedoucí jednotka konvoje je řízená člověkem a právě na něj tedy spadá zodpovědnost za plánování trasy, detekci překážek a vyhýbání se jim. Autonomní podřízené jednotky se tímto tedy nemusejí zabývat a plánování trasy pro ně spočívá v nalezení a realizaci takové trajektorie, která bude co nejpřesněji odpovídat té, kterou projelo vedoucí vozidlo.

Z absence absolutního souřadného systému vyplývá, že veškeré informace k naplánování a projetí trajektorie musí robot získat z údajů o vzdálenostech mezi

vozidly konvoje a z jejich stavů, především rychlostí a natočení. Naše vozidlo tak bude spoléhat na inerciální snímače ke zjištění vlastního stavu, vizuální snímače a dálkoměry jiných typů ke zjištění polohy ostatních vozidel a komunikaci mezi jednotkami k umožnění fúze dat z mnoha zdrojů. Polohu vedoucí jednotky tak můžeme například určit z údajů kamer hned několika vozů a znalosti jejich umístění v konvoji, z odhadu na základě poslední známé polohy a "předpovědí"z inerciálních snímačů, atp.

Problémy navigace lze řešit v několika úrovních komplexnosti a záleží především na konkrétním systému, jak složitý popis systému budeme považovat za dostatečný. Přestože je většina problémů v řízení nelineárních, linearizace a použití zpětnovazební regulace je pro mnoho aplikací postačujícím řešením, jako například v [5]. Jinde je využito nelineárních metod řízení, například Slide Mode Control (SMC) [8] či spojení neuronových sítí a fuzzy regulace [13].

2.2 Zpětnovazební řízení a sestavování trajektorie

Jako první bylo prozkoumáno řízení pouze lineární zpětnovazební regulací v podobě PID regulátoru či jeho variací. Tento způsob řízení je v aplikacích teorie řízení velice rozšířený, především proto, že je ve srovnání s ostatními způsoby nenáročný jak na vývoj a ladění, tak na samotnou realizaci regulátoru. Je tedy možné například navrhnout řídící jednotku vozidla konvoje tak, aby bylo sledování vedoucího vozidla ovládáno dvěma regulátory - jedním pro úhel natočení, který by se snažil minimalizovat odchylku mezi směřováním sledující jednotky a polohou vedoucí jednotky, a druhým pro rychlost vozidla, pomocí které by se snažil udržovat konstantní vzdálenost mezi vozidly. Takové řízení ale není příliš sofistikované a samo o sobě nedokáže zajistit sledování stopy vedoucího vozidla. Prakticky ve všech prozkoumaných metodách tak PID regulátory slouží až k realizaci menších dílčích úkolů, jako například snaha minimalizovat odchylku od vypočtené trajektorie.

Další z metod zpětnovazební regulace je stavová regulace. Tou lze nahradit funkci PID regulátorů, ale stavový popis soustavy přináší i řadu svých výhod. Jednou z nich je například možnost využití estimátorů a jiných konstrukcí pracujících se stavy vozidla. V robotice je velice používaný Kálmánův filtr, který je jednou z hlavních metod sloužících k fúzi dat z více zdrojů. Využití stavové regulace je v naší problematice podobné, jako využití PID regulátoru. Také se hodí spíše k doplnění jiných metod. Zároveň jsou pro obě z představených zpětnovazebních metod problémem nelinearity systému, který je třeba buď linearizovat, nebo zvolit některý z nelineárních způsobů regulace.

Poměrně neobvyklé řešení je popsáno v článku [5]. Autoři se museli potýkat se značnými nelinearitami hnacího ústrojí kamiónů, pro které byla metoda navrhována. Toto bylo vyřešeno pro podélné řízení použitím dvouvrstevné řídící struktury, kde vnitřní smyčka linearizovala kritické části systému na základě jeho modelu. Ve vnější smyčce pak již bylo možné použít lineární stavové řízení. Pro laterální řízení byla navržena "elektronická tažná tyč", kde byla využita kinematika tažné tyče k získání matematického modelu. Řízení spočívalo v myšlence, že zadní náprava

přívěsu sledujícího vozidla musí jet po trajektorii se stejným poloměrem křivosti, jako zadní náprava přívěsu vedoucího vozidla. Udržování takového stavu bylo uskutečněno pomocí regulátoru s klouzavým režimem (Sliding Mode Control).

Mimo tyto koncepty bylo prozkoumáno i použití akcí. Řízení se v tomto případě skládá z příkazů, které má robot vykonat, například jízda rovně po danou dobu, či zatáčení s daným poloměrem křivosti trajektorie.

V následující sekci je popsáno několik metod výpočtu trajektorie. Zároveň jsou představeny dva reprezentativní návrhy z odborných článků, které byly v rámci rešeršní studie prozkoumány.

2.2.1 Typy trajektorií

Jednou z možných trajektorií, kterou by mohla sledující vozidla sledovat, je trasa sestavená z diskrétních bodů, které odpovídají polohám vedoucího vozidla v čase. Vozidlo se vždy navádí na nejbližší bod a po projetí v určité vzdálenosti s danou tolerancí dochází ke skokové změně cíle navigace. To může být pro zpětnovazební regulaci problémem, zejména jsou-li body snímány s velkými odstupy. Tato metoda není o mnoho sofistikovanější než základní řízení PID regulátory a je proto používána velice zřídka.

Dalším přístupem je sestavení trajektorie pomocí křivek, které jsou vypočteny tak, aby procházely polohami vedoucího vozidla v čase a jejich tečny odpovídaly natočení vozidla v daných bodech. Tak získáme spojité řešení definované matematickými vztahy, což umožňuje také spojité řízení zpětnovazební regulací.

Metoda navržená v [9] využívá křivku složenou z kružnicových oblouků. Systém spoléhá na informace o poloze a natočení vedoucího vozidla ze zpracování obrazu dvou kamer v rovnoběžném stereoskopickém uspořádání. Systém je navržen k vizuální detekci značek, ale umožňuje i použití termovize či radaru. K získání přesnějších hodnot je následně využit rekurzivní filtr, který kompenzuje nepřesnosti vzniklé nízkým rozlišením kamer, vibracemi způsobenými terénními nerovnostmi a nedokonalou rovnoběžností os obou kamer. Naměřené údaje jsou poté zpracovány generátorem jízdních akcí, který vypočítá poloměry zatáčení jednotlivých segmentů a rychlost vozidla. K zajištění udržování bezpečné vzdálenosti mezi vozidly je longitudinální řízení realizováno dvěma módy, jeden pro jízdu stálou rychlostí a jeden pro náhlé zrychlování a zpomalování jízdy. Sledování trasy je možné doplnit i algoritmy k detekci a objíždění překážek. Metoda byla ověřena na skutečných vozidlech a dosahovala uspokojivých výsledků.

Avanzini a kolektiv přistoupili v [1] k využití B-spline křivek, které mají (při zvolení vyššího řádu polynomu) oproti segmentům z kružnicových oblouků spojitou také druhou derivaci. Poloměr zatáčení vozidla se tak při projíždění takové trasy mění spojitě, což lépe odpovídá skutečné situaci. Důraz byl v této metodě kladen na komunikaci mezi vozidly, což umožnilo decentralizovanou realizaci řízení. Porovnáváním dat z více vozidel bylo možné předejít akumulaci chyb, ke které dochází v lokálních přístupech k řízení. Při počítání aktuálního segmentu je vždy manipulováno pouze s několika posledními kontrolními body, což zaručuje stálé výpočetní nároky

a nedochází tak k saturaci komunikačních kanálů. Samotné sledování trajektorie je zajištěno stavovou regulací, které se snaží dosáhnout konvergence odchylek úhlu a vzdálenosti k nule pomocí úhlu řízení vozidla a jeho rychlosti. I tato metoda byla úspěšně otestována na skutečných vozidlech. Stavové řízení je použito i v metodě popsané v článku [4]. Tento se ale zabýval spíše problémy stability systému s ohledem na komunikaci mezi jednotkami a udržování daných formací pro jízdu po předem naplánované trajektorii.

2.3 Neuronová síť

Neuronové sítě, přesněji umělé neuronové sítě, jsou systémy, které nereagují na podněty algoritmicky, ale jako výsledek součinnosti velkého množství vysoce provázaných buněk, které pracují paralelně. Hlavním rozdílem oproti klasickým algoritmickým postupům je tedy fakt, že není třeba konkrétně postihnout všechny možné situace, které mohou nastat. Namísto toho jsou neuronové sítě naučeny pro určité podněty dosáhnout určitých výsledků a všechny ostatní podněty jsou posuzovány podle podobnosti těm, které již síť zná. Nemáme přitom ale kontrolu nad tím, *jak přesně* daného výsledku síť dosáhne. [17]

Velkou výhodou je možnost aplikovat neuronové sítě i na vysoce nelineární systémy, nebo takové systémy, u kterých ani nemáme k dispozici potřebný matematický aparát k popisu. Neuronové sítě jsou také vhodné pro aplikace probíhající v reálném čase díky paralelizaci výpočtů.

V praxi je výhodné použít neuronovou síť v kombinaci s klasickým algoritmickým postupem. K takovému řešení v souvislosti s problematikou navigace robotů přistoupili i autoři práce [13]. V této práci byl navržen regulátor, který skloubil prvky neuronových sítí a fuzzy regulátorů. Použití fuzzy logiky umožnilo obejít proces učení neuronové sítě velkým množstvím numerických dat tak, že je neuronová síť učena pomocí spolehlivých expertních pravidel.

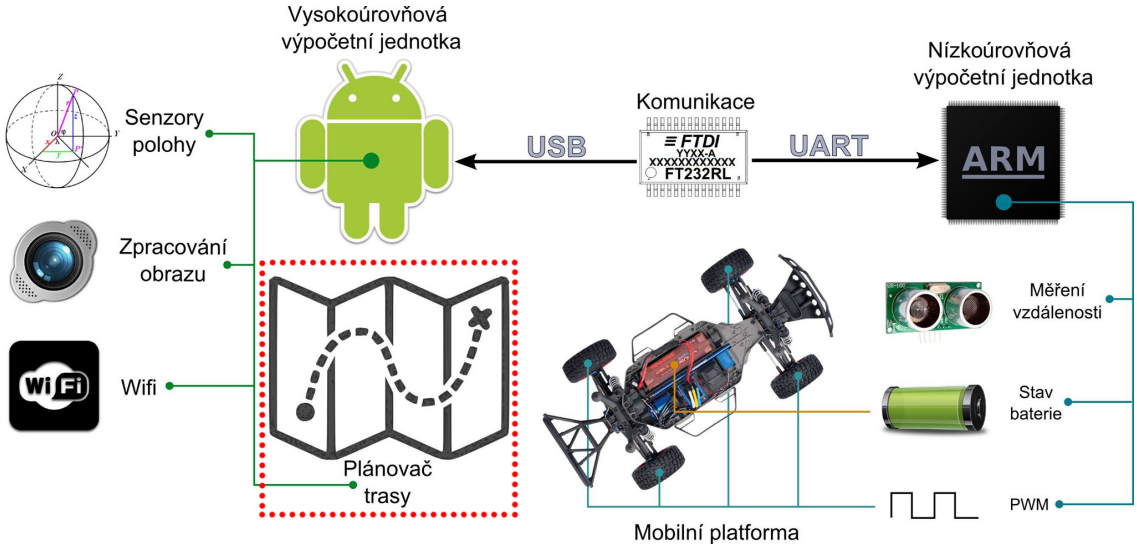
Navržený regulátor se skládá ze tří částí - fuzzy logiky, pravidlové neuronové sítě a další, menší neuronové sítě, která z předchozí odebírá defuzzyfikovaná data a provádí optimalizaci výstupů. Systém má čtyři vstupy ze senzorů - odchylku ve vzdálenosti, odchylku v orientaci, rychlost vedoucího vozidla a rychlost sledující jednotky. Manipulovanými veličinami (výstupy) jsou úhel natočení přední nápravy, celkové natočení vozidla a rychlost. Regulátor se vyznačuje především odolností proti rušení a velmi malými nároky na počet pravidel a data k učení. Pozitivně je hodnocena i schopnost zdokonalovat se v požadovaných úkonech - při správném nastavení parametrů není možné, aby se systém učením sám od sebe zhoršil. Sledování vozidel v konvoji bylo plynulé a reakce vozidel na změny rychlosti velice rychlá.

Přestože bylo použitím tohoto regulátoru dosaženo velice uspokojivých výsledků, z důvodů nedostatečné znalosti těchto konceptů a jejich teoretické náročnosti k tomuto řešení pravděpodobně zatím nepřistoupíme.

3 Popis semiautonomního konvoje a vybavení jednotek

3.1 Popis konvoje

Náš semiautonomní konvoj se skládá z vedoucí jednotky ovládané člověkem a jedné či více podřízených autonomních jednotek. Cílem celého projektu je navrhnout a realizovat takový systém, který by umožnil jízdu autonomních jednotek ve stopě vyjeté vedoucí jednotkou nezmapovaným terénem. Vozidla tedy nebudou spoléhat na lokalizaci a mapování a odpadá tak použití senzorů GPS či navigačních majáků. Bylo rozhodnuto, že hlavním způsobem navádění vozidel bude zpracování obrazu kamery. Tento přístup k navigaci mobilních robotů je velice rozšířený, zejména proto, že kamery jsou v dnešní době již poměrně levné a z jejich obrazu je možné získat vhodným zpracováním mnoho informací. Nevýhodami jsou pak výpočetní nároky na samotné zpracování. Zároveň je kvalita získaných dat ovlivněna okolními světelnými podmínkami, například bez speciálního vybavení jsou kamery prakticky nepoužitelné v noci. K doplnění potřebných informací o stavu vozidel byla přidána řada dalších senzorů.



Obrázek 3.1: Schématické znázornění jednotky konvoje

Jako základ našich jednotek byla použita RC vozidla se stejnosměrnými motory. V rámci bakalářské práce Vojtěcha Kolomazníka byly z autonomních jednotek

odstraněny RC komponenty a řízení bylo nahrazeno deskou plošných spojů s mikrokontrolerem a senzory. Martin Garaj ve své práci navrhl komunikační rozhraní mezi jednotlivými úrovněmi řízení, které slouží k předávání dat ze senzorů jedním směrem a posílání příkazů směrem druhým. Schématický nákres systému našeho konvoje je na obrázku 3.1.

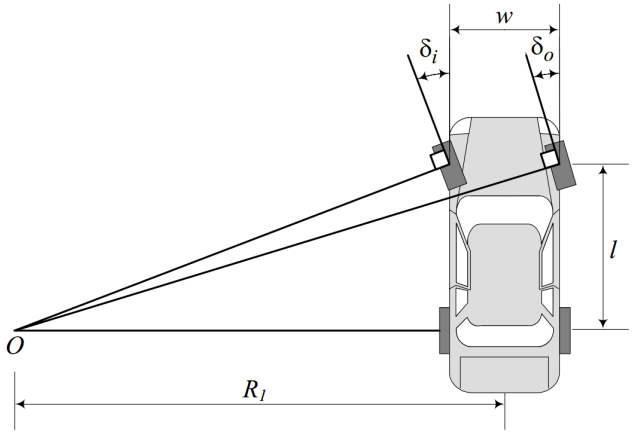
Cílem této bakalářské práce je navrhnout metodu plánování trasy tak, aby bylo zajištěno sledování trajektorie vedoucího vozidla. Společně s rešeršní částí je tak třeba vzít v úvahu aktuální vybavení a navrhnout dostatečně přesný matematický model jednotek.

3.2 Kinematický model Ackermannova řízení

Náš konvoj je složen ze čtyřkolových robotů s náhonem na všechny čtyři kola a podvozkem s pevnou zadní nápravou a pohyblivou přední nápravou. Takový podvozek nalezneme u valné většiny automobilů. Zanedbáme-li nyní dynamické jevy (například řekneme, že vozidlo jede velice pomalu), zjistíme, že aby nedocházelo ke smýkání kol, musí všechna jet po soustředných kružnicích různých poloměrů. Pro podvozek s pevnou zadní nápravou bude střed těchto kružnic ležet na ose procházející zadní nápravou. Úhly natočení předních kol pak musí být různé a musí splňovat tzv. *Ackermannovu podmínku* [6]:

$$\cot \delta_o - \cot \delta_i = \frac{w}{l} \tag{3.1}$$

Znázornění této podmínky nalezneme na obrázku 3.2.

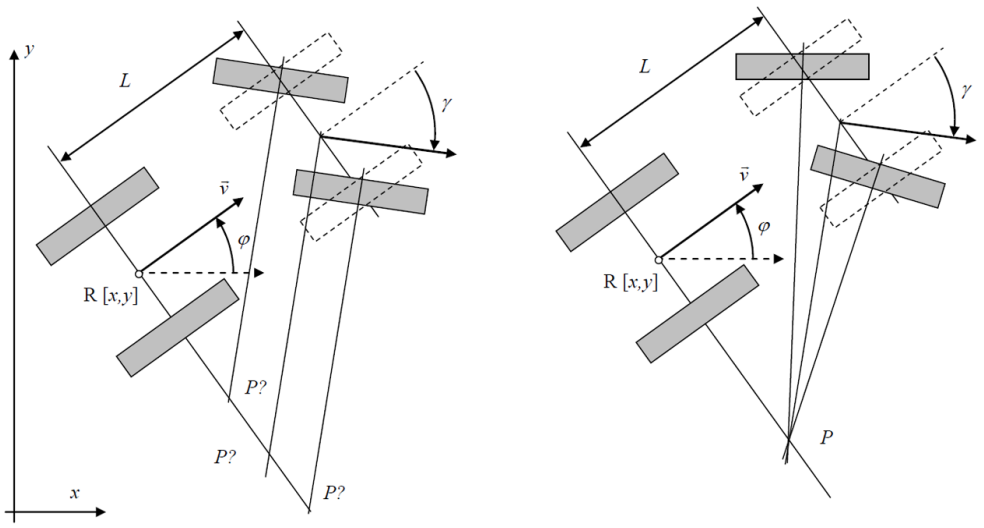


Obrázek 3.2: Znázornění Ackermannovy podmínky [6]

Splnění této podmínky by nebyl problém, kdyby náš podvozek umožňoval nezávislé natáčení obou předních kol. Podobně jako v automobilech je ale zatáčení přední nápravy realizováno čtyřkloubovým mechanismem, který není možné navrhnout tak, aby Ackermannovu podmínku vždy přesně splňoval [6]. Existují však přibližná

řešení a chyba, které se dopouštíme, není příliš veliká. Proto ji při návrhu modelu zanedbáváme.

Zejména pro účely simulace nás zajímá, jak popsat pohyb vozidla s Ackermannovým podvozkem ve dvourozměrném souřadném systému. Pro zjednodušení uvažujeme tříkolkový model, přední kola tak nahradíme jedním, umístěným uprostřed přední nápravy. Zavedeme souřadnicový systém s osami x a y, úhel natočení vozidla měřený od osy x k ose vozidla označíme φ a úhel natočení virtuálního předního kola označíme γ.



(a) (b)

Obrázek 3.3: Automobilový podvozek. Konstantní úhel řízení pro obě kola (a), Ackermannovo řízení (b) [11]

Matematický popis kinematiky vozidla s takovýmto řízením je s použitím souřadného systému a značení veličin podle obrázku 3.3 realizován rovnicemi 3.2 [11] [12].

$$\begin{aligned} \dot{x} &= v \cos \varphi \\ \dot{y} &= v \sin \varphi \\ \dot{\varphi} &= \frac{v}{L} \tan \gamma \end{aligned} \tag{3.2}$$

Jak bylo zmíněno, toto je čistě kinematický model a nepopisuje tedy dynamické jevy, jako je zejména smýkání kol z důvodu setrvačnosti vozu. V praxi existuje mnoho možností, jak upravit Ackermannův podvozek tak, aby si poradil i s tímto. Tato problematika ale přesahuje rozsah této práce. Pro naši situaci je kinematický popis dostačující.

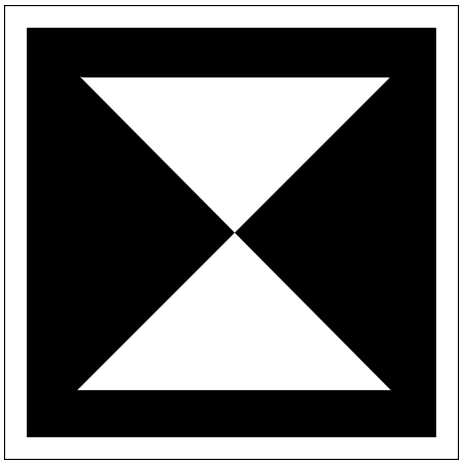
3.3 Senzory

V následujících podsekcích jsou popsány senzory a veličiny, o kterých nám tyto senzory dávají informace. U každého je také zhodnocení použitelnosti v našem konvoji.

3.3.1 Kamera

Pravděpodobně nejdůležitějším vybavením použité podřízené jednotky je kamera. Softwarovým zpracováním obrazu můžeme vhodnou volbou značky a metody získat informace především o odchýlení sledované jednotky od osy kamery, která je shodná s osou vozidla.

V dosavadním postupu vývoje semiautonomního konvoje byl využíván mobilní telefon s operačním systémem Android a jeho integrovaná kamera. Ke zpracování obrazu byla použita knihovna OpenCV, která je velice rozšířená v aplikacích souvisejících s vizuální navigací. Jako značka ke sledování byl vybrán tvar zobrazený na obrázku 3.4. Tato značka se skládá z černých a bílých čtverců a trojúhelníků. Díky vysokému kontrastu a ostrým hranám je snazší ji detekovat, zároveň je možné důkladným nafocením z různých úhlů sestavit databázi, ze které je možné získávat i údaje o natočení a vzdálenosti značky.



Obrázek 3.4: Značka použitá k zjištění polohy a natočení vozidla zpracováním obrazu

Zpracování obrazu je ovšem výpočetně poměrně náročná činnost a je třeba volit kompromis mezi frekvencí snímkování (framerate) a rozlišením, které ovlivňuje přesnost. Na stávajícím hardwaru bylo bohužel k dosažení frekvence přibližně $25 fps$ nutné snížit rozlišení pod $0,2 Mpx$, což výrazně zhoršilo kvalitu získaných dat. Dalším problémem byla špatná detekce značky za měnících se světelných podmínek. Toto bylo vyřešeno použitím adaptivního thresholdingu opět z knihovny OpenCV. I tak je bohužel s tak malým rozlišením problém spolehlivě detekovat značku na vzdálenosti větší, než $0,5 m$, což je velice nedostačující. Je tedy jisté, že použitelných výsledků bude dosaženo až s plánovaným upgradem na systém s kvalitní samostatnou kamerou a dedikovanou grafickou kartou.

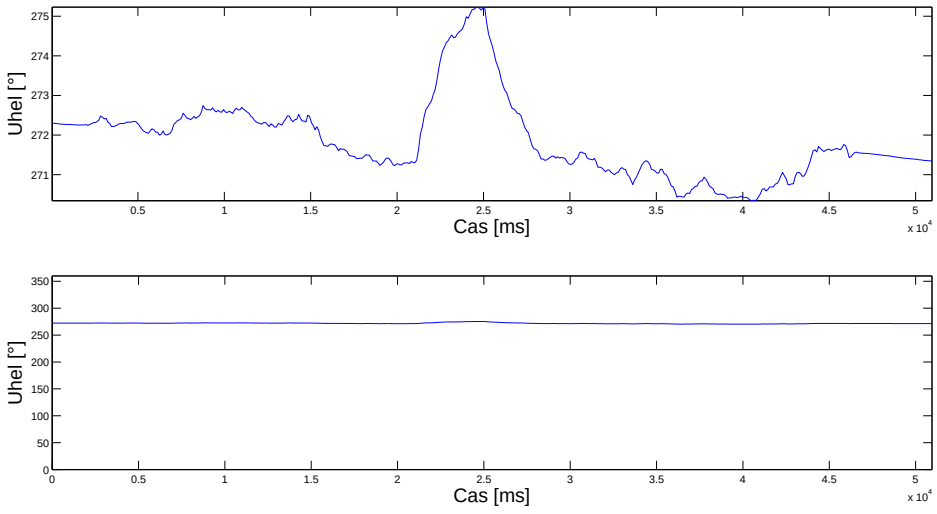
Navíc se ukázalo, že i za dobrých podmínek není detekce značky naprosto spolehlivá a může docházet ke krátkodobým výpadkům dat ze zpracování obrazu. Při návrhu metody toto bylo zohledněno tak, že byly upřednostněny přístupy, které nejsou kriticky závislé na znalosti polohy vedoucího vozidla v každém okamžiku.

Zpřesnění výsledků ze zpracování obrazu je možné dosáhnout i tak, že bude využito více kamer. Příkladem takového přístupu je systém popsaný v [9], kde je dosaženo stereovize použitím dvou kamer, které jsou orientovány rovnoběžně. Předpoklad o menší chybě oproti použití jedné kamery je v článku experimentálně ověřen.

3.3.2 Senzor natočení

Přestože informaci o vzájemném natočení vozidel je možné získat ze zpracování obrazu kamery, nebudou tyto údaje příliš přesné. Daleko přesněji můžeme úhel natočení zjistit využitím kombinace kompasu a gyroskopu. V námi používaném mobilním telefonu je již zabudováno softwarové spojení těchto senzorů pod názvem *senzor natočení*.

Ukázka dat naměřených při pomalé jízdě rovně je na obrázku 3.5. Na naměřených hodnotách je znatelný šum a mírný drift, který však přičítáme faktu, že vozidlo je při měření ovládáno člověkem a jízda tak nikdy není zcela rovná. Proto nás nejvíce zajímá až odchylka v časovém rozmezí $20 - 30s$. Změna hodnot až o tři stupně byla pravděpodobně způsobena objekty v okolí, které mohly ovlivnit senzory, především kompas. I přesto bylo usouzeno, že senzor natočení je možné použít, obzvláště za předpokladu, že bude přistoupeno k použití fúze dat z něj a ze zpracování obrazu. To by mělo vést k dalšímu zpřesnění měření.



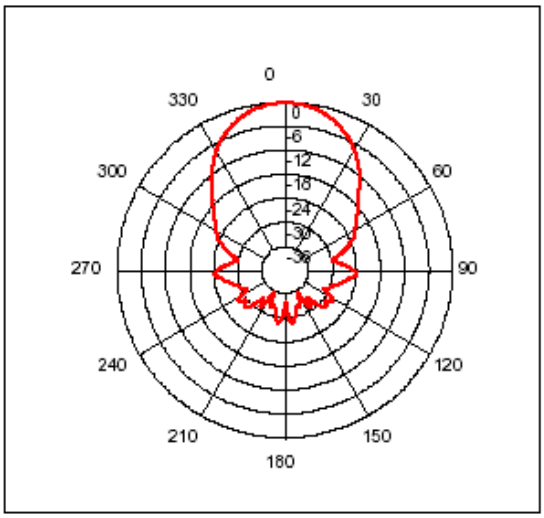
Obrázek 3.5: Data ze senzoru natočení při pomalé jízdě rovně (nahoře výřez osy y)

Jako jediný ze senzorů nám tento udává hodnoty vztažené k absolutnímu souřad-

nému systému. Jelikož nás ale zajímá pouze rozdíl jednotlivých natočení, je volba nulového úhlu naprosto nepodstatná.

3.3.3 Ultrazvukový dálkoměr

Ultrazvukový dálkoměr získává pomocí odrazu tlakových vln ultrazvukových frekvencí údaje o vzdálenosti k objektu, od kterého se vlnění odráží. Tento senzor byl do systému zařazen z důvodů neuspokojivých výsledků zjišťování vzdálenosti ze zpracování obrazu. Senzor nebyl doposud důkladně vyzkoušen, ale ze specifikací ultrazvukových dálkoměrů předpokládáme, že volba tohoto senzoru oproti třeba infračerveným dálkoměrům byla vhodná pro naši aplikaci. V našem konvoji předpokládáme vzdálenosti mezi vozidly od jednotek centimetrů po přibližně dva metry (horní hranice se bude zvyšovat se zvyšující se rychlostí příštích iterací konvoje), což odpovídá parametrům ultrazvukových dálkoměrů používaných v robotice. Problémem se může ukázat omezený úhel snímání vzdálenosti, ale pravděpodobně přísnější jsou zatím omezení plynoucí z návrhu trajektorie v kapitole 4 a zorného úhlu kamery. Na obrázku 3.6 jsou zobrazeny výsledky měření provedeného výrobcem námi použitého senzoru SRF08. Senzor je blíže popsán v práci Vojtěcha Kolomazníka.



Obrázek 3.6: Citlivost ultrazvukového dálkoměru SRF08 v závislosti na úhlu od osy [16]

3.3.4 Trojosý akcelerometr

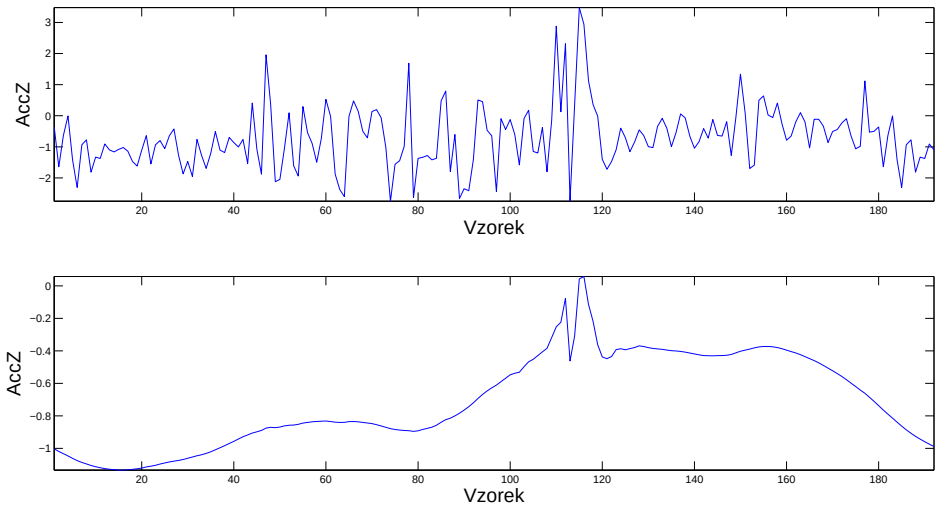
Zatím jediným senzorem, který využíváme k odhadu rychlosti, je trojosý akcelerometr. Použitý senzor v mobilním telefonu má již zabudovanou kompenzaci tíhového zrychlení pomocí gyroskopu. Teoreticky bychom měli být schopni integrací naměřených hodnot získat aktuální rychlost vozidla. Bohužel, data získaná z akcelerometru jsou

vždy výrazně zašuměná. Existuje řada způsobů, jak toto potlačit. Naměřením či dobrým odhadem rušení v měření, kdy je senzor v klidu, získáme rozložení šumu, které lze přibližně popsat normálním rozdělením. Podaří-li se nám pak naměřit i několik charakteristik odezvy na skok požadované rychlosti, můžeme využít Kálmánova filtru k výraznému zlepšení odhadu rychlosti.

Námi použitý podvozek je však příliš stísněný k připojení snímačů otáček na hnací hřídel či jednotlivá kola. Měření odezvy na skok tak bude rovněž zatížené značnou chybou. V dalším vývoji projektu je přidání IRC snímačů nutností, bude to tedy zohledněno při výběru nového podvozku.

Šum senzoru je ovšem menším problémem ve srovnání s nepřesnostmi vznikajícími v důsledku pohybu vozidla. Část tvoří kmity vyvolané jízdou po povrchu, který nikdy nebude dokonale rovný. Toto by mělo být alespoň částečně kompenzováno tlumením v podvozku. Další chyba vzniká proto, že v důsledku zrychlování, zpomalování či zatáčení se náš lehký podvozek naklání. Toto by pravděpodobně šlo potlačit kompenzací pomocí gyroskopu.

Problémem plynoucím z dosavadního použitého hardwaru je obtížnost správného umístěni a fixace mobilního telefonu, ve kterém je senzor zabudován. Ten má již nezanedbatelnou hmotnost a závěs je tak také ovlivňován dynamickými jevy. Zároveň bylo zjištěno, že v důsledku několika běžících aplikací a použití operačního systému Android není ani při použití vlastního vlákna zaručeno snímání dat se stejnou periodou. To značně komplikuje zpracování dat například filtrací. Oba tyto problémy by měl vyřešit samostatný akcelerometr, který v budoucnosti pravděpodobně rozšíří desku plošných spojů vyvíjenou v rámci bakalářské práce Vojtěcha Kolomazníka. Na druhou stranu to znamená, že bude třeba napsat vlastní software ke skloubení akcelerometru a gyroskopu.



Obrázek 3.7: Měření pomocí akcelerometru - nahoře hrubá data, dole po filtraci vlnkovou transformací v programu MATLAB

Na obrázku 3.7 vidíme data naměřená akcelerometrem v mobilním telefonu. Při tomto experimentu byla zjišťována odezva na skokovou změnu žádané hodnoty PWM signálu do motoru podřízené jednotky. Tato odezva by posloužila k návrhu Kálmánova filtru pro odhad rychlosti na základě posílané hodnoty PWM a dynamických vlastností systému. Na hrubých datech je velice znatelný šum a výsledky měření se tak jeví prakticky nepoužitelné. Po filtraci vlnkovou transformací nástrojem *wavemenu* v programu MATLAB ale vidíme hodnoty, které v rámci pokusu dávají smysl. Pomineme-li obrácenou orientaci osy y (odpovídá obrácené fyzické orientaci akcelerometru), vidíme zpočátku největší hodnotu zrychlení, které odpovídá rozjezdu vozidla. Postupně se zrychlení zmenšuje, protože rychlost vozidla se blíží požadované hodnotě. Ostré špičky v průběhu zrychlení ukazují okamžik, kdy bylo vozidlo zvednuto ze země a experiment ukončen. Zbylá data již nemají žádnou výpovědní hodnotu. Z tohoto měření by se tedy dalo usuzovat, že v moment ukončení experimentu ještě nebyla rychlost ustálená - zrychlení nedosáhlo nulové hodnoty. Je ovšem otázkou, zda má použitá filtrace a její parametry vůbec fyzikální význam. Zdržujeme se tedy vyřčení rozsáhlejších závěrů a pokus bude opakován po připojení IRC snímačů.

Kromě těchto snímačů bychom údaje o rychlosti mohli získat také využitím znalosti vzdálenosti a natočení obou vozidel v různých časových okamžicích. Každé vozidlo má vlastní odhad rychlosti a toto může být konfrontováno se změnou geometrického uspořádání konvoje. Předpokládáme, že fúzí údajů z akcelerometru, IRC snímačů a výpočtů na základě poloh a údajů komunikovaných mezi vozidly získáme dostatečně přesný odhad rychlosti.

Rychlost hraje přitom velice důležitou roli v námi navržené metodě plánování trasy (kapitola 4), protože na ní stojí využití dead reckoning, tedy odhadu současné polohy vozidla (v našem případě nás zajímá spíše odhad ujeté dráhy) z minulé polohy a velikosti a směru okamžité rychlosti.

4 Návrh metody

Vzhledem k současnému stavu konvoje byla zvolena metoda plánování trajektorie pomocí kružnicových oblouků spojujících polohy a orientace vozidel v okamžiku výpočtu, použitá z citovaných zdrojů například v [9]. Metoda byla zvolena především z těchto důvodů:

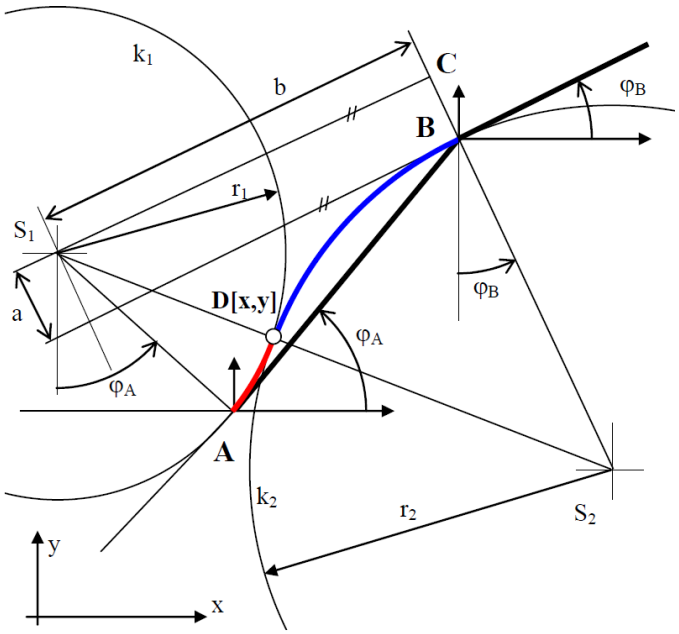
- K výpočtu a následování trajektorie je zapotřebí pouze znalost rychlosti sledujícího vozidla, relativní polohy (úhel a vzdálenost) vedoucího vozidla a absolutního natočení obou vozidel
- Nízké nároky na výpočetní výkon - poměrně jednoduché vztahy analytické geometrie řešitelné pouze s využitím goniometrických funkcí a polynomů

4.1 Předpoklady a omezení

Při použití této metody je třeba mít na paměti, že správně funguje pouze za určitých předpokladů:

- Rychlosti vozidel jsou dostatečně malé, aby bylo k řízení možné použít pouze kinematický model podvozku a nezabývat se podrobnější dynamikou systému
- Úhlová rychlost natáčení přední nápravy je dostatečně velká, ideálně nekonečná
- Vzdálenosti mezi vozidly jsou natolik malé, že sestavená trajektorie velice dobře odpovídá skutečné trajektorii projeté vedoucím vozidlem
- Po projetí sledujícího vozidla poslední vypočtenou trajektorií se vedoucí vozidlo nachází v zorném poli kamery, stejně tak v efektivním dosahu zpracování obrazu a ultrazvukového senzoru vzdálenosti
- Vozidla jezdí po dostatečně rovném povrchu, který se dá dostatečně přesně nahradit dvourozměrnou plochou

Člověk řídící vedoucí jednotku je tak omezen především nároky na rychlost vozidla a komplikovanost trajektorie. Rychlá a zběsilá jízda klikatou trajektorií nerovným terénem není s takovýmto řízením spolehlivě realizovatelná.



Obrázek 4.1: Výpočet kruhových oblouků pro segment cesty definovaný dvojicí bodů A, B a úhlem navazujícího segmentu [11]

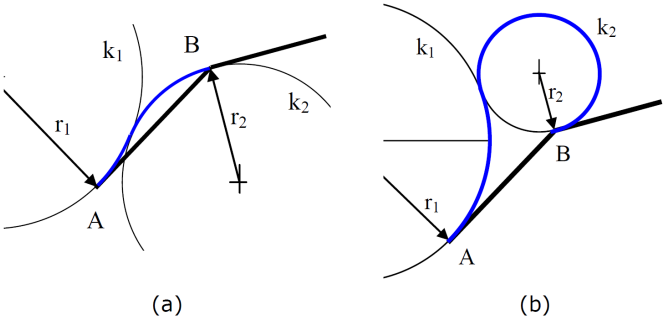
4.2 Sestavení trajektorie

4.2.1 Výpočet parametrů obloukové trajektorie

Podrobný nákres geometrie použité pro výpočet trajektorie je na obrázku 4.1. Předpokládejme, že známe polohy bodů A a B v našem souřadném systému. Stejně tak v obou bodech známe i úhel, který svírá tečna trajektorie s osou x. Naším úkolem je nalézt středy S_1 a S_2 a poloměry r_1 a r_2 kružnic k_1 a k_2 tak, aby splňovaly požadavky na tečny v bodech a spojitou návaznost na sebe. Takových poloměrů je nekonečné množství, jeden z nich tedy musíme vhodně zvolit. Prozatím řekněme, že jsme určitým způsobem zvolili poloměr r_1. K výpočtu ostatních parametrů využijeme podle obrázku 4.1 následující rovnice [11]:

$$\begin{aligned}
(x_D - x_{S_1})^2 + (y_D - y_{S_1})^2 &= r_1^2 \\
(x_D - x_{S_2})^2 + (y_D - y_{S_2})^2 &= r_2^2 \\
x_{S_2} &= x_B + r_2 \cos\varphi_b \\
y_{S_2} &= y_B + r_2 \sin\varphi_b \\
y_D - y_{S_1} &= \frac{y_{S_2} - y_{S_1}}{x_{S_2} - x_{S_1}}(x_D - x_{S_1})
\end{aligned} \tag{4.1}$$

Těchto pět rovnic nám postačuje k výpočtu našich pěti neznámých (x_D, y_D, r_2, x_{S_2} a y_{S_2}), ale vzhledem k tomu, že rovnice kružnic jsou kvadratické, získáme dvě možná řešení, z nichž pouze jedno odpovídá našim představám, viz obrázek 4.2.



Obrázek 4.2: Dvě řešení kvadratických rovnic, přijatelné (a) a nepoužitelné (b) [11]

K získání jednoznačného řešení tedy využijeme pravoúhlý trojúhelník $\triangle S_1 S_2 C$, jehož přepona je součtem poloměrů obou kružnic a obě odvěsny a i b dopočítáme pomocí rovnic přímek rovnoběžných či kolmých k výslednému směru trajektorie. Známe-li tedy a, b i r_1, můžeme z rovnice

$$b^2 + (a + r_2)^2 = (r_1 + r_2)^2 \tag{4.2}$$

vyjádřit r_2 takto [11]:

$$r_2 = \frac{-r_1^2 + b^2 + a^2}{-2a + 2r_1} \tag{4.3}$$

Je třeba si ovšem dát pozor na konstrukci tohoto trojúhelníku, protože se může stát, že kvůli vzájemné poloze a orientaci bodů nebude jedna z odvěsen rovna $(a+r_2)$, ale $(-a + r_2)$. To nastane v případě, že průsečík tečny v bodě B a kolmice na tečnu v bodě A bude od bodu A dále, než bod S_1.

Vraťme se k volbě vhodné velikosti poloměru r_1. Z důvodů plynulosti jízdy je výhodné, aby oba oblouky měly pokud možno co nejpodobnější velikost poloměru. Experimentálně bylo zjištěno, že taková velikost vyjádřená jako zlomek vzdálenosti mezi body A a B úzce souvisí s relativním úhlem segmentu φ, tedy s úhlem, který svírá tečna v bodě A se spojnicí bodů A a B. Simulačně bylo nalezeno optimální nastavení poloměru pro několik hodnot úhlu a výsledek proložen polynomem druhého stupně ve tvaru:

$$p = -1,1191\varphi^2 + 4,384|\varphi| \tag{4.4}$$

Zvolíme-li, že d je rovno vzdálenosti bodů A a B, pak je poloměr r_1 vypočten ze vztahu:

$$r_1 = \frac{d}{p} \tag{4.5}$$

Je nutno dodat, že například rovnice odvozené v [9] již v sobě zahrnují rovnost obou poloměrů, ale to je právě jediný případ, ve kterém mají rovnice unikátní řešení. Námi použité vztahy umožňují návrh trasy o různých poloměrech oblouků, což se

může v budoucnosti ukázat jako výhodné pro speciální případy trajektorií. Zároveň dosahují vztahy 4.3 a 4.5 velice podobných výsledků jako rovnice odvozené v [9].

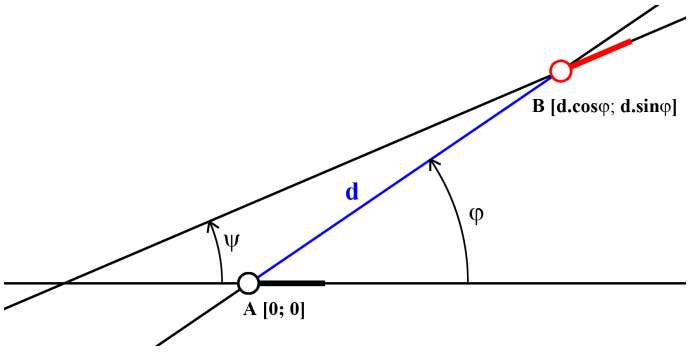
Po zjištění poloměrů zatáčení dokážeme zjistit souřadnice všech významných bodů, jmenovitě S_1, S_2 a průsečíku jejich spojnice s trajektorií, D. Úhly $\angle AS_1D$ a $\angle DS_2B$ a poloměry r_1 a r_2 využijeme k výpočtu délky jednotlivých segmentů. Vozidlo poté pomocí dead reckoning usuzuje, ve kterém segmentu se právě nachází a projelo-li již celou trajektorii. To znamená, že po vypočtení trajektorie je prováděn výpočet uražené vzdálenosti, kdy se přírůstek dráhy uražené v každém výpočetním kroku zjišťuje jednoduchou rovnicí $s = vdt$, kde v je okamžitá rychlost vozidla a dt výpočetní periodou. Hodnota těchto přírůstků nasčítaných od začátku projíždění aktuální vypočtené trajektorie je pak porovnávána s délkou segmentů.

4.2.2 Výpočet v relativních souřadnicích

Předešlé výpočty byly uvažovány pro absolutní souřadnicový systém. Naše podřízené jednotky však nemají možnost získávat přesná data o své absolutní poloze a mapovat okolí. Jedinými našimi vstupy jsou rychlost v, vzdálenost vozidel d, úhel odchylky vozidel φ a absolutní natočení obou jednotek, jejichž rozdílem získáme celkovou změnu natočení ψ po projetí aktuálního úseku trajektorie. To nám naštěstí nevadí, protože výstupy z plánování trasy jsou pouze poloměry zatáčení a délky jednotlivých oblouků, tedy hodnoty nezávislé na souřadnicovém systému.

Na obrázku 4.3 je znázorněna transformace vstupů tak, aby bylo možné použít odvozené rovnice. Podřízená jednotka je vždy umístěna do počátku souřadnicového systému s nulovým úhlem natočení, poloha vedoucí jednotky je vypočtena přes vzdálenost d a úhel φ a změna natočení ψ je vypočtena z rozdílu absolutních natočení obou vozidel. Takové umístění bodu A zároveň umožňuje zjednodušení výpočtu, protože z několika rovnic přímek nám díky nulovým souřadnicím vypadne množství členů.

Vzájemný vztah ψ a φ hraje podstatnou roli při určování směru zatáčení a typu trajektorie, což je podrobně rozebráno v následující sekci.

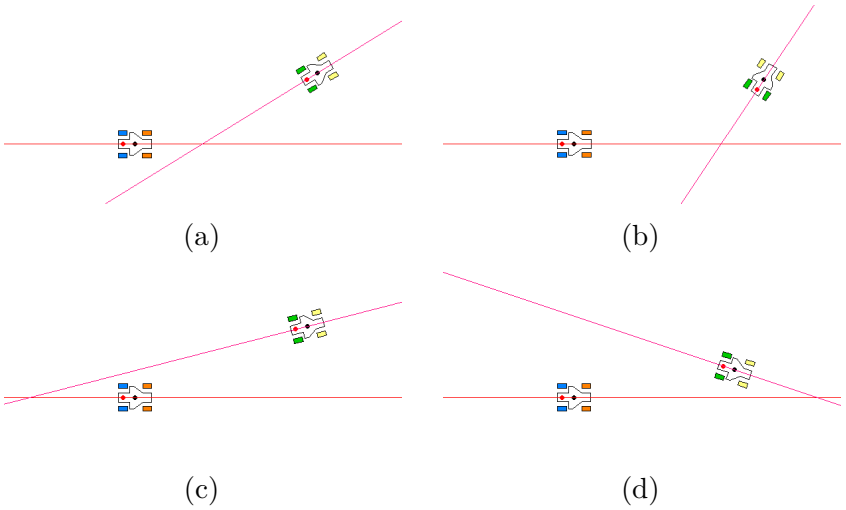


Obrázek 4.3: Znázornění známých vstupů pro použití výpočtu podle obrázku 4.1

4.2.3 Typ trajektorie v závislosti na orientaci vozidel

Přestože předpokládáme nekonečnou úhlovou rychlost natáčení přední nápravy, reálná situace samozřejmě tak příznivá není. Především přechod mezi dvěma kružnicemi je problémový kvůli náhlé změně požadovaného poloměru zatáčení. Tento problém se dá částečně vyřešit ovlivňováním rychlosti vozidla tak, aby pro menší poloměry zatáčení vozidlo zpomalilo a při zatáčení tak nevyjelo daleko mimo trajektorii, mohlo by však dojít ke ztrátě kontaktu s vedoucím vozidlem, například zpožděním a vyjetím z efektivního dosahu zpracování obrazu kamery. Bylo proto usouzeno, že kde to bude možné, bude použito trajektorie složené pouze z jedné kružnice a rovného úseku. Chyba našeho předpokladu by se tak měla omezit, nebo přinejmenším nezhoršit, stejně tak potřebný výpočetní čas bude kratší.

Na obrázku 4.4 jsou znázorněny čtyři možné situace, které mohou reálně v našem konvoji nastat. Vidíme, že v případech 4.4a a 4.4b je možné trasu složit pouze z jednoho kružnicového oblouku a rovného úseku v závislosti na tom, které z vozidel je blíže průsečíku orientací obou vozidel. V případech 4.4c a 4.4d je nemožné do koncového bodu a natočení dojet pouze zatáčením na jednu stranu, je tedy třeba trajektorii složit ze dvou kružnic.



Obrázek 4.4: Možné orientace vozidel v konvoji

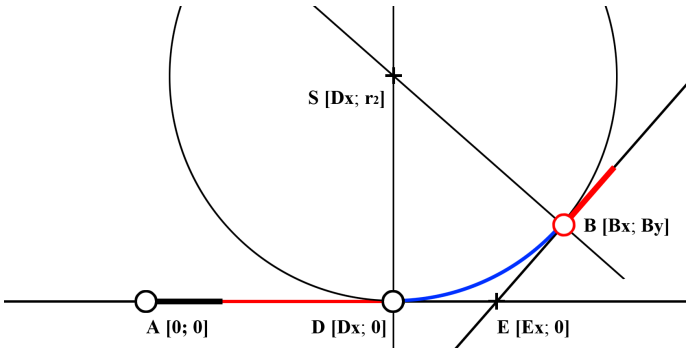
Při plánování trajektorie z kružnicového oblouku a rovného úseku je však větší riziko, že navržený oblouk bude mít menší poloměr, než je minimální poloměr zatáčení vozidla. V takovém případě nezbývá, než vypočítat trasu ze dvou kružnicových oblouků.

4.2.4 Výpočet parametrů trajektorie úsečka + oblouk

Znázornění sestavení trajektorie z úsečky a kruhového oblouku vidíme na obrázku 4.5. O tom, který z úseků bude rovný, a který zakřivený, rozhoduje poloha bodu E, tedy průsečíku orientací obou vozidel. Ve vyobrazeném případě je tento bod blíže

koncovému bodu B, úsečka tedy bude prvním úsekem. Dalším krokem je nalezení polohy bodu D, tedy bodu, kde se setkávají oba úseky trajektorie. Z geometrie situace je zřejmé, že vzdálenost mezi body D a E musí být shodná se vzdáleností mezi body E a B. Zároveň v tomto případě leží bod D na ose x.

Sestrojíme-li nyní kolmice v bodech D a B, jejich průsečík nám dá bod S, tedy střed kružnice. V případě takovéto trajektorie tedy není poloměr kružnice volným parametrem a bude vždy největším možným pro dané geometrické uspořádání. To zaručí co nejmenší změnu poloměru zatáčení vozidla, čímž minimalizujeme chyby způsobené našimi předpoklady. Je také patrné, že výpočet tohoto typu trajektorie je velice nenáročný i ve srovnání se složitější variantou trajektorie.



Obrázek 4.5: Znázornění výpočtu trajektorie úsečka + oblouk

4.3 PI regulátor natočení

Bylo již zmíněno, že sestavená trajektorie je tím věrnější skutečné trajektorii vedoucího vozidla, čím je vzdálenost mezi vozidly v okamžik výpočtu menší. Se zmenšující se vzdáleností se ale může stát, že trajektorie sestavená naší metodou nebude vyhovovat omezení na minimální poloměr zatáčení vozidla. Pravděpodobně hlavní příčinou tohoto problému jsou nepřesnosti senzorů, kdy se například vedoucí vozidlo jeví sledujícímu blíže, než ve skutečnosti je. Jelikož je horní mez vzdálenosti mezi vozidly diktována rozlišením obrazu kamery, nemůžeme tento problém lehce obejít. Trajektorie s menšími poloměry zatáčení, než je reálně možné, samozřejmě není pro naše vozidlo splnitelná a při jejím sledování by docházelo k výraznému vyjíždění ze zatáček, které by vedlo například ke ztrátě kontaktu s vedoucím vozidlem. Bylo proto rozhodnuto v takovém případě přepnout řízení do módu ovládaného PI regulátorem natočení přední nápravy.

Řízení realizované výhradně PI regulátorem je sice výpočetně nenáročné a velice spolehlivé, trajektorie sledujícího vozidla ale věrně kopíruje požadovanou trajektorii pouze při rovné jízdě. Při zatáčení dochází k odchýlení z ideální stopy, které by mohlo při jízdě v těsných prostorech (například úzké koridory či těsné objíždění rohu) vést ke srážce s objekty v okolí vozidla. Na obrázku 4.6 je srovnání trajektorií projetých pomocí PI regulátoru a pomocí výše navržené metody v simulaci v programu MATLAB.

V nouzových situacích, kdy si naše metoda sama neporadí, je ale využití PI regulátoru menším zlem, než úplné vyjetí ze stopy následkem nevytočení ostré zatáčky. Jakmile je jednotka opět navedena na vedoucí vozidlo, můžeme přejít zpět k použití dead reckoning a obloukové trajektorie.

Zároveň bylo simulačně zjištěno, že při jízdě rovně je také výhodné přepnout řízení na použití PI regulátoru. Při vhodné volbě mezního úhlu vozidel je tak výpočet nenáročný a trajektorie přesná. Naopak oblouková trajektorie může jízdu rovně zbytečně komplikovat.

(a) (b)

Obrázek 4.6: Srovnání jízdy pomocí PI regulátoru natočení (a) a pomocí navržené metody plánování trajektorie (b)

4.4 PI regulátor vzdálenosti

Dosud jsme se zabývali laterálním řízením, tedy především tvarem trajektorie. V naší metodě je laterální (příčné) řízení naprosto oddělené od longitudinálního (podélného). Rychlost vozidla při sledování trajektorie vystupuje pouze jako volný parametr a je ovládána zvlášť, což je umožněno především zanedbáním dynamických jevů díky malým rychlostem a setrvačným hmotám vozidel našeho konvoje.

Naším záměrem bylo, aby se vozidla sledovala s pokud možno konstantními rozestupy. Byl tedy navržen PI regulátor vzdálenosti tak, aby pomocí ovládání rychlosti dodržoval předem stanovený odstup. V ideálním konvoji by byla taková vzdálenost brána jako délka trajektorie mezi vozidly, ale jednou z (v tomto případě) nevýhod naší metody plánování trajektorie je, že je vždy vypočtena jen aktuální část. Během projíždění tohoto segmentu bychom museli kvůli takovému řešení počítat trajektorii v každém okamžiku, či ji jinak odhadovat (například opět na základě dead reckoning). Kvůli faktu, že jízda konvoje je již poměrně omezená jinými předpoklady a nebude tak příliš komplikovaná, bylo rozhodnuto použít jako směrodatnou čistě vzdálenost vzdušnou čarou, tedy údaje z ultrazvukového dálkoměru a zpracování obrazu kamery.

Oba PI regulátory byly laděny pouze podle jejich chování v simulaci, protože bylo usouzeno, že není třeba důkladného matematického popisu dynamiky systému. Parametry vozidla ovlivňující dynamiku také závisí na ostatních částech systému, které jsou neustále ve vývoji.

5 Implementace a simulace

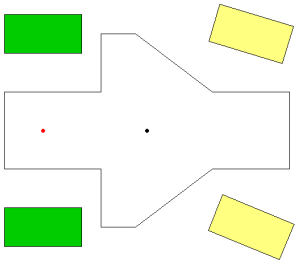
5.1 Simulace v programu MATLAB

V rámci splnění zadání bakalářské práce byla k návrhu a testování metody sestavování trajektorie vytvořena simulace v programu MATLAB. Základem simulace je hlavní cyklus, který řeší pohybové diferenciální rovnice. Přestože program MATLAB obsahuje množství řešičů diferenciálních rovnic, bylo vzhledem k jednoduchosti úlohy rozhodnuto použít vlastní naprogramovanou Eulerovu metodu s konstantním časovým krokem $0,01s$. Řešené rovnice vycházejí z kinematického modelu Ackermannova podvozku a v kódu jsou realizovány takto:

```
1 robot.r = robot.size.length/tan(robot.psi); % poloměr zatáčení
2
3 robot.fidot = robot.v / robot.r; % úhlová rychlost natáčení v ...
    souřadném systému
4
5 robot.x = robot.x + robot.v*dT*cos(robot.fi);
6 robot.y = robot.y + robot.v*dT*sin(robot.fi);
7 robot.fi = robot.fi + robot.fidot*dT;
```

K získání smysluplných výsledků a přehledu o situaci bylo navrženo také grafické rozhraní, které vykresluje v reálném čase půdorysný pohled na dvourozměrnou plochu znázorňující terén. Současně se na této ploše nacházejí dvě vozidla, kterým byl pro jednoznačnost znázornění navržen tvar podobný použitým reálným jednotkám. Tento je znázorněn na obrázku 5.1. V simulaci je možné rozměry vozidla a jeho částí snadno nastavit podle potřeby pro případ použití nových podvozků, se kterými se do budoucna počítá. Z obrázku je také patrné, že kola přední nápravy jsou správně znázorněna s odlišnými úhly natočení tak, aby splňovala Ackermannovu podmínku, kdy při zatáčení musí všechna kola jet po soustředných kružnicích. Zároveň je kromě středu vozidla (černá tečka) znázorněn i střed zadní nápravy (červená tečka). Poloha právě tohoto bodu je považována za polohu vozidla. Přestože se taková volba může zdát neintuitivní, je velice výhodná pro výpočty. Bod totiž leží jak na ose zadní nápravy, tak na podélné ose vozidla. Pohybuje se tak vždy po trajektorii, jejíž poloměr zatáčení vychází z úhlu natočení virtuálního prostředního kola přední nápravy, se kterým počítáme.

Ke zjišťování polohy všech bodů vozidla byla s výhodou využita optimalizovaná práce s maticemi, kterou program MATLAB nabízí. Celé vozidlo je sestaveno z bodů definovaných pro nulový úhel natočení, každý bod či tvar složený z více bodů je přitom sloupcový vektor či obecně dvouřádková matice, kde první řádek tvoří



Obrázek 5.1: Grafická reprezentace jednotky konvoje

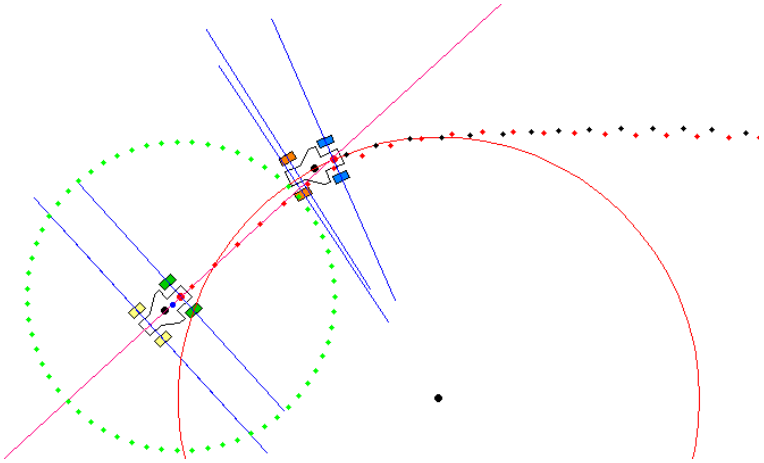
souřadnice na ose x a druhý souřadnice na ose y. K výpočtu polohy těchto bodů pro jiný úhel natočení vozidla pak stačí tyto matice vynásobit transformační maticí pro rotaci a posunout na aktuální polohu vozidla.

```
% transformační matice
robot.C = [cos(robot.fi), -sin(robot.fi); sin(robot.fi), ...
    cos(robot.fi)];
```

K účelům návrhu a testování metody plánování trajektorie obsahuje grafické rozhraní také řadu dalších prvků. Na obrázku 5.2 je ukázka ze simulace, na které je vidět:

- Dvě jednotky semiautonomního konvoje - vedoucí, řízená člověkem (žlutá + zelená) a podřízená, autonomní (oranžová + modrá)
- Osy všech kol, sloužící k ověření správného zatáčení
- Okamžité trajektorie obou vozidel. Ve vyobrazeném případě jede vedoucí jednotka rovně, okamžitou trajektorií je tedy přímka (růžová). Podřízená jednotka právě zatáčí, okamžitou trajektorií je kružnice (červená) o poloměru odpovídajícím poloměru zatáčení vozidla. Je také znázorněn střed zatáčení (černý bod)
- Zelená tečkovaná kružnice znázorňující požadovanou vzdálenost, ve které se podřízená jednotka snaží udržet
- Body, které vozidla za sebou "pokládají"v daných časových intervalech. Vedoucímu vozidlu náleží červené body, sledujícímu černé. Tyto body slouží ke znázornění projetých trajektorií, lze podle nich tedy posuzovat přesnost navržené metody

V hlavním cyklu jsou kromě řešení diferenciálních rovnic prováděny také další úkony, které však zpravidla chceme vykonávat s jinou periodou. U regulátorů například proto, abychom ověřili funkčnost i při nízkých vzorkovacích frekvencích snímaných dat, jinde například z důvodů menších nároků na výpočet. To se týká především grafického rozhraní, protože program MATLAB není k takovému účelu



Obrázek 5.2: Grafické rozhraní simulace v programu MATLAB

primárně určen a při vysoké frekvenci vykreslování by docházelo ke zbytečnému zpomalování celé simulace. Tento problém samozřejmě odpadá při implementaci do reálné aplikace.

Zavedeme-li omezení, že perioda daného úkonu je celočíselným násobkem výpočetní periody hlavního cyklu, můžeme takové kusy kódu provádět pomocí funkce modulo takto (příkladem je část kódu starající se o výpočet úhlu natočení přední nápravy pomocí PI regulátoru):

```
% PI pro uhel
if mod(T, dTreg) == 0
    [robot.psi, pipsi.e] = PI_psi(angle, pipsi, target.psimax);
end
```

Samotný PI regulátor je realizován v další ukázce kódu. Protože obsahuje pouze složky P a I, stačí při každém spuštění funkce znát pouze aktuální odchylku od žádané hodnoty a naintegrovanou hodnotu odchylky za celý běh simulace. Dále je ošetřen nežádoucí windup efekt integrace saturací hodnoty integrálu. Na závěr je provedena také saturace výstupu tak, aby odpovídal reálným omezením vozidla, v tomto případě maximálnímu úhlu natočení nápravy.

```
function [output, e] = PI_psi(uhel, pipsi, psimax)

global dTreg;

% e=error vector ([k], integral)

error = atan2(sin(uhel), cos(uhel));

e(2) = pipsi.e(2) + error*dTreg;

if (abs(e(2)) > 10)
```

```
    e(2) = sign(e(2))*10;
end

e(1) = error;

output = pipsi.kp*e(1) + pipsi.ki*e(2);

if output > psimax
    output = psimax;
elseif output < -psimax
    output = -psimax;
end
```

Důležitou součástí simulace je možnost sledovat, jak navržená metoda funguje při reálných omezeních, jako je například nepřesnost senzorů či konečná rychlost natáčení přední nápravy. Právě poslední zmíněný problém je realizován v kódu níže, kde je změna úhlu natočení diskretizována podle naměřené maximální úhlové rychlosti natáčení přední nápravy vozidla.

```
if robot.psiwnt > robot.psi
    robot.psi = min(robot.psi + koef*robot.psidotmax*dT, robot.psiwnt);
elseif robot.psiwnt < robot.psi
    robot.psi = max(robot.psi - koef*robot.psidotmax*dT, robot.psiwnt);
end
```

Jak bylo zmíněno v kapitole o návrhu metody, předcházející problém s konečnou rychlostí zatáčení je možné omezit ovládáním rychlosti tak, aby při malých poloměrech zatáčení vozidlo zpomalilo.

```
robot.vmax = min(abs(R1)/(0.2+abs(R1)) * target.vmax, target.vmax);
```

Pro simulaci nepřesností senzorů je ke skutečným údajům o vzdálenosti, úhlu, atp. přidáván Gaussovský šum s volitelnými hodnotami směrodatné odchylky a střední hodnoty.

```
function [y] = makeNoisy(x, std, additive)

y = x + std*randn(1, length(x)) + additive;
```

Veškeré výpočty regulátorů a navrhování trasy počítají s takto zašuměnými daty. Protože nepřesné, rychle se měnící hodnoty nepříznivě ovlivňují funkci především regulátorů, bylo do simulace přidáno i filtrování pomocí mediánového filtru volitelného řádu. Tato úprava dat bude zajisté potřeba i v reálné aplikaci a vede sice k vyhlazení šumu, ale vnáší do systému zpoždění úměrné řádu filtru a výpočetní periodě.

```
% příklad simulace šumu a filtrace dat pro senzor vzdálenosti

fakeDist = makeNoisy(target.distance, 0.05, 0.01);
```

```
4
5           % mediánový filtr
6           for i=1:(rad-1)
7               fakeDistVec(i) = fakeDistVec(i+1);
8           end
9   fakeDistVec(end) = fakeDist;
10
11  fakeDist = median(fakeDistVec);
```

Dalšími z volitelných parametrů jsou dva mezní úhly pro přepínání mezi jízdou pomocí PI regulátoru natočení či podle navržené metody plánování trasy. Velikosti těchto parametrů jsou obecně různé, což vede k jakési "hysterezi", která zabraňuje rychlým překmitům z důvodu nepřesnosti senzorů. Zároveň je po přepnutí módu zabráněno další změně, dokud neuplyne určitá odmlka - v základním nastavení $0,3s$.

```
1   if abs(relangle) > pilimit && pimode && cnt>30 % mezní uhel pro PI
2       pimode = 0;
3       first = 0;
4
5       distanceDone = 0;
6       distanceToTravel = target.dd/2;
7       seg1 = 10;
8       R1 = 9999;
9
10      disp('SWITCH TO PATH');
11
12  end
```

Metoda plánování trasy z oblouků či úseček je implementována přesně podle návrhu představeném v kapitole 4. Z našich čtyř vstupů vypočítáme délky dvou segmentů a jejich poloměry křivosti. Pro svou délku je kód k nalezení až v přiložených souborech.

```
1   function [R1, seg1, R2, seg2] = makePathSegmentNew(distance, angle, ...
        alfa, beta)
```

Funkce zároveň umožňuje vykreslení každé sestavené části v grafickém rozhraní, čímž můžeme ověřit správnost metody.

5.1.1 Vyhodnocení výsledků simulace

Většina simulací byla prováděna v rychlostech okolo $2m/s$, což je rychlost vyšší, než reálně očekáváme v současné fázi vývoje. Ukázalo se, že i při těchto rychlostech je navržená metoda k řízení vhodná, ovšem pouze za splnění předpokladů o nepříliš členité jízdě. Tato omezení ale pravděpodobně nebudou tím nejvíce limitujícím faktorem - daleko přísněji je členitost trajektorie omezena zorným úhlem kamery, ze které zatím získáváme pouze data o úhlu mezi vozidly. Uspokojivě fungují také oba PI regulátory, a to i při nízkých vzorkovacích frekvencích. Filtrování vstupních dat

se ovšem ukázalo jako nutnost. Simulace byly prováděny zatím pouze s vedoucím vozidlem a jednou podřízenou jednotkou, což odpovídá současnému stavu konvoje, není však mnoho důvodů se domnívat, že s více vozidly bude metoda nepoužitelná. V takovém případě bude ale vhodné navrhnout komplexnější longitudinální řízení, ve kterém si budou jednotky předávat informace o zrychlování a zpomalování tak, aby byl konvoj schopen reagovat jako celek. Potřebná komunikace prostřednictvím wi-fi je již ve vývoji.

Je poměrně obtížné kvantifikovat přesnost metody pro různá nastavení, protože konvoj vykazuje velice stabilní chování pro řadu postupně se zvyšujících hodnot rychlosti a směrodatných odchylek šumu senzorů. Teprve po dosažení jistých kritických hodnot (popsaných níže) se chování skokově zhoršuje za hranice použitelnosti.

V simulaci byla vyzkoušena řada různých nastavení šumů senzorů a maximální povolené rychlosti vozidla. Bylo zjištěno, že snímání vzdálenosti vozidla i se značnými odchylkami od skutečné hodnoty nezpůsobuje selhání metody. Hodnoty jsou totiž dostatečně vyhlazeny mediánovým filtrem. Konvoj nevykazoval nežádoucí chování ani pro směrodatnou odchylku měření vzdálenosti $15cm$, což je více než 10% obvyklé vzdálenosti mezi vozidly. Při vyšších hodnotách již není PI regulátor schopný udržet plynulou jízdu. Zvyšováním řádu filtrace či vhodným nastavením regulátoru bychom mohli dosáhnout zlepšení plynulosti, ale výsledné zpomalení by již značně omezilo schopnost sledující jednotky reagovat na náhlé změny rychlosti vedoucího vozidla.

Zhoršení kvality snímání úhlové odchylky mezi vozidly negativně ovlivnilo zejména sestavování trajektorie, kde jsou jednotlivé oblouky navrhovány tečně k natočením vozidel a trajektorie sestavená z nepřesného měření pak logicky neodpovídá skutečné trase vedoucího vozidla. Protože je ke správnému návrhu trasy zapotřebí co nejaktuálnějších hodnot, není vhodné vnášet do systému zpoždění, například filtrací (kterou tak využijeme jen pro PI regulátory). U snímání vzdálenosti se použití nefiltrovaných hodnot při návrhu trasy (možná překvapivě) výrazně neprojevilo a limitujícím faktorem byl PI regulátor rychlosti. V případě snímání úhlu byla situace obrácená a nepřesnosti vedly k oscilacím sledujícího vozidla kolem ideální trajektorie. Jízda po požadované trase bez oscilací byla dosažena pro hodnoty směrodatné odchylky měření úhlu mezi vozidly menší než 2°. O něco odolnější je návrh trajektorie vůči chybám v měření jednotlivých natočení vozidel, ze kterých je počítáno relativní natočení jednotek. Zde se jako nepřístupná velikost směrodatné odchylky ukázala hodnota přibližně 5°.

Přesnější stanovení maximální rychlosti skutečného konvoje vyžaduje extenzivní naměření přesností použitých senzorů, které zatím neproběhlo. Pro podkritické hodnoty odchylek uvedené výše však byl konvoj v simulaci schopen jezdit i rychlostmi přesahujícími $3m/s$. Pro jízdu vyšší rychlostí, která je jednou z priorit pro další vývoj, je ale třeba zvážit, zda rozšířit stávající metodu, nebo přistoupit ke zcela jinému řešení. Návrh trasy z křivky se spojitou druhou derivací, tedy takové, která mění poloměr zakřivení spojitě, by jistě pomohl omezit problémy plynoucí z konečné rychlosti natáčení přední nápravy. Při vyšších rychlostech bychom však pravděpodobně již museli také uvažovat dynamické jevy, jejichž modelování by značně zkomplikovalo výpočty.

5.2 Implementace do robustní aplikace

Implementace do systému podřízené jednotky semiautonomního konvoje je spíše záležitostí ostatních prací zabývajících se tímto konvojem, protože posílání a částečně i zpracování dat z jednotlivých částí je řešeno v každé z nich zvlášť. Hardwarová nízkoúrovňová jednotka navržená Vojtěchem Kolomazníkem se stará především o řízení motorů podle vstupních signálů. Výsledek práce Martina Garaje umožňuje komunikaci mezi úrovněmi řízení jednotky.

Při testování naší podřízené jednotky byl jako hlavní výpočetní systém doposud využit mobilní telefon s operačním systémem Android. Většina aplikací je tak psána v jazyku Java, mezi nimi například i zpracování obrazu pomocí knihovny OpenCV. Kód realizující návrh trasy v simulátoru byl tedy přepsán z jazyku MATLAB do jazyku Java. Jelikož není třeba překládat veškeré grafické funkce ze simulace a zbudou tak pouze základní goniometrické funkce a polynomy, je tento překlad víceméně pouze otázkou záměny syntaxe.

V budoucnosti se počítá s rozšířením simulace v programu MATLAB a výsledné aplikace tak, aby obstarávala funkci vysokoúrovňové řídící jednotky s rozsáhlejšími možnostmi řízení. V době dokončení této práce ještě nebylo možné navrženou metodu plně ověřit na reálném vozidle, protože nebyly dokončeny všechny náležitosti systému. Již nyní je ale jisté, že plánovanou změnou hardwarového vybavení jednotek se výrazně zlepší především dostupný výpočetní výkon pro zpracování obrazu kamery, která je kvůli omezenému rozlišení a tedy i přesnosti zatím pravděpodobně nejslabším článkem systému. Z kamery budou také získávána data nejen o úhlu mezi jednotkami, ale i o relativním natočení vůči sobě a o vzdálenosti. Zároveň pořízení snímačů natočení kol umožní přesnější odhad okamžité rychlosti, než je zatím možné získat pouze z akcelerometru. Fúzí údajů o stejné veličině z různých senzorů pak získáme přesnější informace o stavu vozidla.

6 Závěr

Tato práce se v první části zabývala prozkoumáním a popisem existujících metod návrhu trasy pro autonomní vozidla v konvoji. S ohledem na zamýšlenou funkci semiautonomního konvoje byl kladen důraz na metody vyvinuté pro vozidla, která nespoléhají na lokalizaci a mapování okolí. Byly prozkoumány aplikace konceptů, jako jsou neuronové sítě, fuzzy regulace, stavové řízení a lineární zpětnovazební regulátory.

Následoval rozbor semiautonomního konvoje. Byly popsány dostupné senzory, hardware a software. Vzhledem k cílům práce na konvoji bylo rozhodnuto, že použití Ackermannova kinematického modelu čtyřkolového vozidla s pevnou zadní nápravou je dostatečně přesné. Zanedbání dynamických jevů značně usnadnilo další postup návrhu aplikace, ale vneslo do systému nepřesnosti, se kterými bylo třeba počítat.

Poté byla prezentována samotná metoda návrhu trajektorie pro podřízené jednotky semiautonomního konvoje. Vzhledem k jednoduchosti výpočtu a implementace byla zvolena metoda, kdy laterální a longitudinální řízení jsou plně odděleny. Longitudinální řízení obstarává PI regulátor vzdálenosti, který se snaží udržet zvolené rozestupy mezi vozidly. Laterální řízení využívá dead reckoning k jízdě po trajektorii navržené z kruhových oblouků a úseček.

Zvolená metoda byla simulačně ověřena v grafickém prostředí navrženém v rámci této práce v programu MATLAB. Simulace obsahuje množství volitelných parametrů a grafických funkcí, sloužících k posouzení správnosti navržené metody. Při vývoji byl kladen důraz na využitelnost v budoucím pokračování projektu. Zároveň jsou simulovány i nepřesnosti senzorů a zpoždění vzniklé filtrováním vstupních dat. Je tedy možné posuzovat vliv těchto chyb na jízdu vozidla. Ze simulací bylo usouzeno, že navržená metoda je pro současný stav konvoje vhodná, ale pravděpodobně ji bude třeba dále rozšířit k dosažení pokročilejších cílů projektu.

Implementace do reálného systému spočívala v přepsání navržené metody do jazyka Java tak, aby mohla spolupracovat s ostatními aplikacemi semiautonomního konvoje, jako jsou wi-fi komunikace či zpracování obrazu, které bylo realizováno pomocí knihovny OpenCV. V čas dokončení práce nebyl semiautonomní konvoj plně v provozu, experimentální ověření (které nebylo zamýšlenou součástí této bakalářské práce) metody plánování trasy tak bude provedeno v blízké budoucnosti. Výsledky experimentů budou porovnávány s hlavním výsledkem této práce, tedy simulací v prostředí MATLAB, která tvoří základ pro vysokoúrovňovou řídící jednotku celého semiautonomního konvoje.

Seznam použitých zdrojů

[1] AVANZINI, P.; THUILOT, B.; DALLEJ, T.; aj.: On-line reference trajectory generation for manually convoying a platoon of automatic urban vehicles. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009. IROS 2009.*, Říjen 2009, ISBN 978-1-4244-3804-4, s. 1867–1872.

[2] BENHIMANE, S.; MALIS, E.; RIVES, P.; aj.: Vision-based Control for Car Platooning using Homography Decomposition. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005. ICRA 2005.*, Duben 2005, ISBN 0-7803-8914-X, s. 2161–2166.

[3] CONSOLINI, L.; MORBIDI, F.; PRATTICHIZZO, D.; aj.: Leader-follower formation control of nonholonomic robots with input constraints. *Automatica*, ročník 44, č. 5, Září 2007: s. 1343–1349, doi:10.1016/j.automatica.2007.09.019.
URL `<http://dx.doi.org/10.1016/j.automatica.2007.09.019>`

[4] DONG, W.; GUO, Y.; FARRELL, J.: Formation control of nonholonomic mobile robots. *American Control Conference*, Červen 2006, doi:10.1109/acc.2006.1657616.
URL `<http://dx.doi.org/10.1109/acc.2006.1657616>`

[5] FRITZ, H.: Longitudinal and lateral control of heavy duty trucks for automated vehicle following in mixed traffic: experimental results from the CHAUFFEUR project. In *Proceedings of the 1999 IEEE International Conference on Control Applications*, Srpen 1999, ISBN 0-7803-5446-X, s. 1348–1352.

[6] IDSC: Vehicle Dynamics and Design. [Online], 2014, [cit. 2014-05-06].
URL `<http://www.idsc.ethz.ch/Courses/vehicle_dynamics_and_design/11_0_0_Steering_Theroy.pdf>`

[7] JARZEBOWSKA, E.: Leader follower tracking control design for task-based missions. *International Journal of Vehicle Autonomous Systems*, ročník 9, 2011: s. 203–279, doi:10.1504/ijvas.2011.041385.
URL `<http://dx.doi.org/10.1504/ijvas.2011.041385>`

[8] JUNG-MIN, Y.; JONG-HWAN, K.: Sliding Mode Control for Trajectory Tracking of Nonholonomic Wheeled Mobile Robots. *IEEE Transactions on Robotics and Automation*, ročník 15, č. 3, Červen 1999: s. 578–587, ISSN 1042-296X.

[9] KEHTARNAVAZ, N.; GRISWOLD, N. C.; LEE, J.: Visual control of an autonomous vehicle (BART) - the vehicle-following problem. *IEEE Transactions on Vehicular Technology*, ročník 40, č. 3, Srpen 1991: s. 654–662, ISSN 0018-9545.

[10] KHATIB, M.; JAOUNI, H.; CHATILA, R.; aj.: Dynamic path modification for car-like nonholonomic mobile robots. In *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, Duben 1997, ISBN 0-7803-3612-7, s. 2920 – 2925.

[11] KREJSA, J.; VĚCHET, S.: Navigace mobilních robotů, interní dokument VUT.

[12] LAVALLE, S.: Planning Algorithms. [Online], 2006, [rev. 2012-04-20], [cit. 2014-05-06].
URL `<http://planning.cs.uiuc.edu/>`

[13] NG, K.; TRIVEDI, M.: A Neuro-Fuzzy Controller for Mobile Robot Navigation and Multirobot Convoying. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, ročník 28, č. 6, Prosinec 1998: s. 829–840, ISSN 1083-4419.

[14] PIAZZI, A.; BIANCO, C. L.; BERTOZZI, M.; aj.: Quintic G2 splines for the iterative steering of vision-based autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, ročník 3, č. 1, Srpen 2002: s. 27–36, ISSN 1524-9050.

[15] SHIN, K.; SADAYUKI, T.: Lateral and Longitudinal Control Algorithms for Visual Platooning of Autonomous Vehicles. *Seoul 2000 FISITA World Automotive Congress*, Červen 2000.

[16] SRF08: Ultrasonic range finder. [Online], 2014, [cit. 2014-05-06].
URL `<http://www.robot-electronics.co.uk/htm/srf08tech.shtml>`

[17] STERGIOU, C.; SIGANOS, D.: Neural Networks. [Online], [cit. 2014-05-06].
URL `<http:`
`//www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html>`

[18] SUDIN, S.; COOK, P.: Two-vehicle look-ahead convoy control systems. *2004 IEEE 59th Vehicular Technology Conference. VTC 2004-Spring (IEEE Cat. No.04CH37514)*, ročník 5, Květen 2004: s. 2935 – 2939.

Seznam obrázků