



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV AUTOMATIZACE A INFORMATIKY

INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

**NÁVRH A REALIZACE MODELU PODVOZKU MOBILNÍHO
ROBOTU PRO FRAMEWORK ROS**

DESIGN OF MOBILE ROBOT SIMULATION MODEL SUITABLE FOR ROS FRAMEWORK

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Miroslav Čepel

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Stanislav Věchet, Ph.D.

BRNO 2017

Zadání bakalářské práce

Ústav: Ústav automatizace a informatiky
Student: **Miroslav Čepí**
Studijní program: Strojírenství
Studijní obor: Základy strojírenského inženýrství
Vedoucí práce: **doc. Ing. Stanislav Věchet, Ph.D.**
Akademický rok: 2016/17

Ředitel ústavu Vám v souladu se zákonem č.1111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Návrh a realizace modelu podvozku mobilního robotu pro framework ROS

Stručná charakteristika problematiky úkolu:

Cílem práce je sestavit obsáhlou rešerši možných modelů podvozků vhodných pro implementaci v rámci frameworku ROS. Na základě rešeršní studie bude některý z modelů podvozku implementován a simulačně ověřen v simulátoru ROS Stage.

Cíle bakalářské práce:

Seznamte se s aktuálním stavem frameworku ROS.

Prostudujte tvorbu modelů různých typů podvozků mobilních robotů a jejich implementaci v rámci frameworku ROS.

Navrhněte vhodný simulační model pohybu autonomního vozidla dle specifikací.

Navržený systém otestujte.

Seznam doporučené literatury:

THRUN Sebastian, BURGARD Wolfram, and FOX Dieter. Probabilistic Robotics (Intelligent Robotics and Autonomous Agents series). Intelligent robotics and autonomous agents. The MIT Press, August 2005.

CHOSSET Howie, LYNCH Kevin M., HUTCHINSON Seth, KANTOR George, BURGARD Wolfram, KAVRAKI Lydia E., and THRUN Sebastian. Principles of Robot Motion: Theory, Algorithms, and Implementations (Intelligent Robotics and Autonomous Agents). The MIT Press, June 2005.

ROS.org. ROS.org | Powering the world's robots. [online]. 2.11.2016 [cit. 2016-11-02]. Dostupné z: <http://www.ros.org/>.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2016/17

V Brně, dne

L. S.

doc. Ing. Radomil Matoušek, Ph.D.
ředitel ústavu

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

Abstrakt

Tato práce se zabývá problematikou pohybových modelů mobilních robotů. V první části je přiblížen framework ROS a simulátor STAGE. Druhá část navazuje rešerší možných podvozků a zkoumá vhodnost pro implementaci do frameworku ROS. Ve třetí části je vybráno jedno určité uspořádání podvozku, které je pak nasimulováno na základě měření na reálném robotu.

Summary

This thesis focuses on motion models of autonomous robots. In the first section the framework ROS is introduced and creation of models for this framework is mentioned. The second section focuses on various types of robots and discusses their suitability for the framework ROS. Basic kinematic models for the most common robots are shown. In the third section a particular chassis is chosen, a motion model is experimentally determined and the results are tested on a control set of data.

Klíčová slova

Mobilní robot, Pohybový model, Ackermannův podvozek, ROS

Keywords

Mobile robot, Motion model, Ackermann chassis, ROS

ČEPL, M. *Návrh a realizace modelu podvozku mobilního robotu pro framework ROS*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2017. 56 s. Vedoucí Stanislav Věchet.

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně dle pokynů vedoucího a s použitím uvedené odborné literatury.

Miroslav Čepl

Tímto chci poděkovat doc.Ing. Stanislavu Věchetovi, Ph.D. za vedení, cenné rady a přátelský přístup při zpracování této bakalářské práce.
Děkuji své rodině a přátelům za podporu a pochopení během tvorby.

Miroslav Čepl

Obsah

1	Úvod	13
2	Framework ROS	15
2.1	ROS – Robot operating system	15
2.1.1	Základní vlastnosti	15
2.1.2	Distribuce	15
2.2	Hlavní moduly	16
2.2.1	Simulátor STAGE	16
2.2.2	RVIZ	16
2.3	Tvorba modelu podvozku	16
3	Rešerše pohybových modelů	19
3.1	Úvod do pohybových modelů	19
3.2	Roboty pohybující se po souši	20
3.2.1	Kolové roboty	20
3.2.2	Pásové podvozky	26
3.2.3	Kráčející roboty	27
3.2.4	Plazivé roboty	28
3.2.5	Šplhající roboty	28
3.2.6	Skákající roboty	28
3.2.7	Hybridní roboty	29
3.2.8	Pravděpodobnostní modely	29
3.3	Mobilní roboty pohybující se po vodě, ve vzduchu nebo ve vesmírném prostoru	32
3.3.1	Mobilní roboty pohybující se ve vodě	32
3.3.2	Mobilní roboty pohybující se ve vzduchu	33
3.3.3	Mobilní roboty pohybující se ve vesmírném prostoru	35
4	Motivace k práci	37
5	Konkrétní řešení	39
5.1	Účel experimentu a popis měřeného objektu	39
5.1.1	Popis měřeného podvozku	39
5.2	Návrh měření a realizace	41
5.2.1	Návrh experimentu	41
5.2.2	Průběh měření	42
5.3	Zpracování výsledků a zhodnocení	42
6	Závěr	49
7	Seznam použité literatury	51
8	Seznam použitých zkratk a symbolů	53
9	Seznam příloh	55

1. Úvod

Tato práce se věnuje problematice modelování pohybu podvozků pro mobilní roboty. Cílem je vytvořit přehled možných uspořádání podvozků a uvést jejich základní vlastnosti.

Nejdříve je představen Framework ROS. Tento systém je vhodný pro implementaci řízení mobilních robotů a je volně šiřitelný. Jsou zmíněny jeho základní vlastnosti a představeny hlavní moduly. Také je nastíněna tvorba modelů pro tento framework.

V navazující části práce je provedena rešerše různých typů robotů a posouzena jejich vhodnost pro Framework ROS. U nejčastěji používaných podvozků je nastíněna kinematika pohybu.

V praktické části je vybrán reálný podvozek a provedena analýza pohybu na základě řídicích signálů. Výsledky by měly sloužit k přesnějšímu řízení pohybu podvozku nebo k předpovědi jeho pohybu. Výsledky by měly být použitelné ve frameworku ROS.

2. Framework ROS

V této části je představen framework ROS. Kapitola 2.1 se zabývá hlavními aspekty. V kapitole 2.2 je poté uveden integrovaný simulátor STAGE a zmíněn modul RVIZ, sloužící k přehledné vizualizaci dat. Kapitola 2.3 se věnuje tvorbě modelu pro framework ROS.

2.1. ROS – Robot operating system

Robot operating system (volně přeloženo: Operační systém pro roboty), zkráceně ROS, je volně šiřitelná otevřená platforma pro simulaci a řízení robotických systémů. Logo projektu je na obr.2.1. Hlavní myšlenka tohoto projektu je sdružit vývojáře aplikací pro robotické systémy. Nejedná se o opravdový operační systém, ale má jeho znaky, jako je poskytování služeb a nástrojů pro tvorbu, správu nebo diagnostiku. Primárně je vyvíjen pro operační systém Linux, podpora pro jiné systémy je jen částečná.[1]



Obrázek 2.1: Logo frameworku ROS [1]

2.1.1. Základní vlastnosti

Filozofie frameworku ROS je poskytnout pouze potřebné zázemí pro tvorbu projektu. Tato vlastnost umožňuje integraci do různých systémů. Platforma je také optimalizována pro růst komplexnosti a velikosti systému. K dispozici jsou rozsáhlé knihovny, které jsou neustále dopňovány a aktualizovány. Pro implementaci kódu je možné použít několik moderních programovacích jazyků, jako je Python, C++ nebo Lisp. Framework je založen na principu balíčků, komunikace je typem publisher/ subscriber. V praxi to znamená, že každý balíček je samostatný, jednotlivé funkce pak komunikují přes témata (topic), kam data posílají, nebo je odebírají. Samotnou komunikaci zprostředkovává framework. Tato koncepce umožňuje snadnou implementaci cizího kódu, čehož využívá početná uživatelská základna a vývoj je rychlý. Framework má ovšem i svá úskalí, pro začínajícího uživatele není snadný na pochopení. Také není primárně stavěný na real-time řízení.[1]

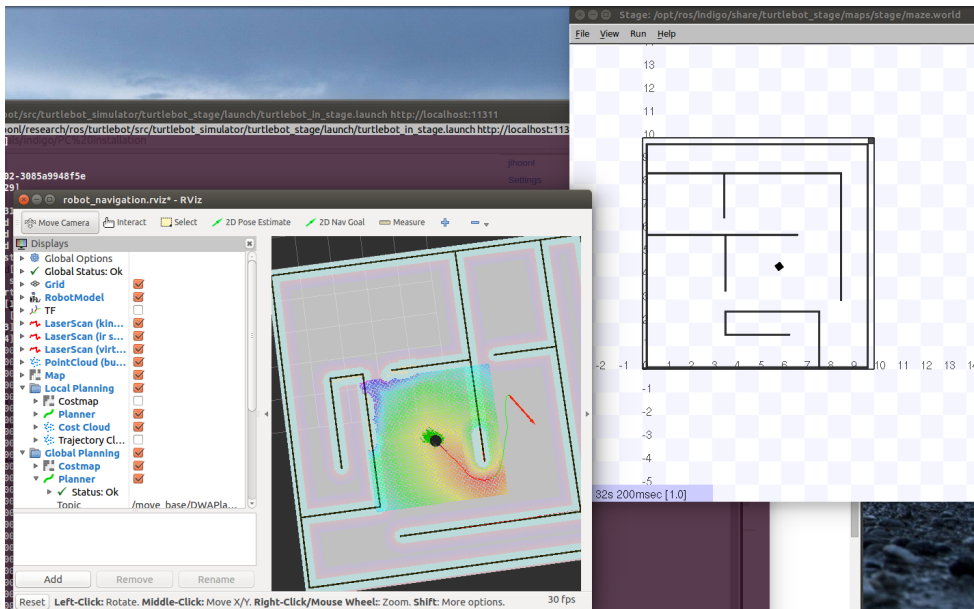
2.1.2. Distribuce

Jádro frameworku je spolu s nejdůležitějšími knihovnami pravidelně vylepšováno a vydáváno jako ucelený balíček – distribuce. V současné době existuje deset distribucí, ovšem jen nejnovější jsou podporovány. Těmi jsou: ROS Kinetic Kame (vydáno 23. května 2016), ROS Jade Turtle (vydáno 23. května 2015) a ROS Indigo Igloo (vydáno 22. července 2014). V době psaní této práce bylo naplánováno vydání nejnovější verze ROS Lunar Loggerhead, datum vydání 23. května 2017. Každá distribuce má vlastní nástroje a knihovny. Balíčky pro určitou distribuci jsou s ní kompatibilní, pro balíčky určené jiné distribuci není kompatibilita zaručena. Novější distribuce nabízí lepší služby a obecně robustnější podporu pro hardware i software.[1]

2.2. Hlavní moduly

2.2.1. Simulátor STAGE

STAGE je simulátor vyvinutý pro framework ROS. Umožňuje vytvoření virtuálního světa, ve kterém se pohybují mobilní roboti. Zajišťuje interakci mapy a senzorů robota. Tento nástroj je ideální pro testování algoritmů aniž by byla potřeba reálný robot a senzory. Pro generování světa se nejčastěji používá bitmapa, kde logická nula reprezentuje volné místo a logická jednička reprezentuje překážku. Načtená mapa je pak dvourozměrná. Poté jsou nahrány modely robotů. Každá součást může mít specifické vlastnosti, například senzor může mít zadanou přesnost, rozsah měřených hodnot atd.[1] Na obr.2.2 vpravo lze vidět příklad vygenerovaného světa.



Obrázek 2.2: Ukázka prostředí RVIZ (vlevo) a STAGE (vpravo)[1]

2.2.2. RVIZ

Simulátor STAGE bohužel nenabízí pokročilé vizuální prostředí. K tomu je určen modul RVIZ. Dokáže interpretovat data získaná jak simulací, tak i z reálných senzorů. Běh simulace není nijak narušen, data jsou získána ze stejných uzlů, které používá simulace nebo reálný robot. Modul také umožňuje interakci, jako například řízení robotu v simulaci, online nastavování parametrů nebo řízení běhu simulace, například zastavení chodu pro vyhodnocení právě nastalé situace.[1] Na obr.2.2 vlevo lze vidět příklad vizualizace.

2.3. Tvorba modelu podvozku

Tvorba modelu ve frameworku ROS je komplexní záležitost. Míra komplexity závisí na požadavcích na model. Také závisí na požadované funkci, existují dva přístupy implementace:

Tvorba balíčků

Tento přístup je možný, pokud je možné podvozek simulovat řadou rovnic nebo algoritmem. Vytvoří se balíček, který obsluhuje určitý uzel. Do tohoto uzlu vstupují potřebné údaje, například o požadovaném pohybu, balíček vezme data z uzlu, provede potřebné operace a vrátí požadovaná data, například hodnotu signálu pro motory. Balíček lze vytvořit ručně, ale snadnější je použít připravený modul frameworku.[1] Samotný skript uvnitř balíčku je možné psát v jazyce C++ nebo python, pro každý jazyk existuje patřičný soubor. Výběr jazyka je na uživateli. Dalším krokem je připojení balíčku na tok dat. Pro přenos jsou používány zprávy (message). Je možné použít standardní formát, nebo lze nadefinovat vlastní formát zprávy. Posledním krokem je připojení balíčku na uzly. K vstupnímu uzlu se balíček připojuje jako subscriber – odběratel. K výstupnímu uzlu se balíček připojuje jako publisher – přispěvatel. Vstupní a výstupní uzly jsou obecně různé, ale může jít i o uzel totožný.

Tvorba geometrického modelu

Pro potřeby simulace lze definovat geometrický model podvozku. Má dvě části:

- vizuální – slouží pro vizualizaci pro operátora,
- kolizní – slouží pro definování kolizní geometrie pro interakci s prostředím a ostatními roboty.

Tyto dva modely nemusí být vizuálně shodné, obvykle je vizuální model tvarově složitější, protože probíhá pouze vykreslování na obrazovku. Naopak kolizní model je jednoduchou reprezentací robota, často je skutečný tvar nahrazován zjednodušeným modelem, například válcem, pro snížení výpočetní náročnosti. Obě části modelu jsou tvořeny pomocí xml souborů. Základ tvoří takzvaná základna. Určuje hlavní souřadný systém robota. K základně jsou připojeny další komponenty. Vazba může být pevná nebo pohyblivá. Lze definovat osy rotace jednotlivých dílů. Každý díl má vlastní souřadný systém. Pro přepočítání na hlavní SS slouží transformační matice. Více o tvorbě je v [2].

3. Rešerše pohybových modelů

V části 3.1 je krátký úvod do této kapitoly. Dále jsou v části 3.2 rozebrány roboty pohybující se po souši. Roboty pohybující se v jiných prostředích jsou přiblíženy v části 3.3.

3.1. Úvod do pohybových modelů

Pohybové modely obecně slouží k popisu pohybu zkoumaného předmětu v prostoru. V případě podvozku se v ideálním případě jedná o deterministický matematický popis, který na základě pohybu jednotlivých částí robotu spočítá výsledný pohyb těžiště nebo referenčního bodu. Obecně se pohybem zabývá kinematika, podrobněji studuje pohyb dynamika. S ohledem na rozsah práce bude kinematika pohybu zmíněna pouze u některých podvozků. Existuje mnoho možností, jak realizovat podvozek robotu. Tato práce se zaměřuje na podvozky používané ve frameworku ROS. Velká část tohoto frameworku je postavena na blíčcích, které předpokládají všesměrový, holonomní podvozek. Tohoto předpokladu se poté využívá při řešení lokalizace a plánování pohybu. I přes toto omezení je možné implementovat i neholonomní robot. V tomto případě je potřeba upravit algoritmy, aby se zamezilo konfliktním stavům. Detailněji jsou rozebrány podvozky, které jsou do frameworku ROS implementovány nejčastěji, nejvíce pozornosti je věnováno kolovým podvozkům. Pro přehlednost bylo vybráno následující dělení robotů podle [3]. Lze rozdělit roboty podle prostředí na pohyb:

- po souši,
- ve vodě,
- ve vzduchu,
- ve vesmírném prostoru,
- hybridní – kombinace předchozích.

Podle [3] je možné dále dělit roboty pohybující se po souši na:

- kolové,
- pásové,
- kráčejší,
- plazivé,
- šplhající,
- skákající,
- hybridní.

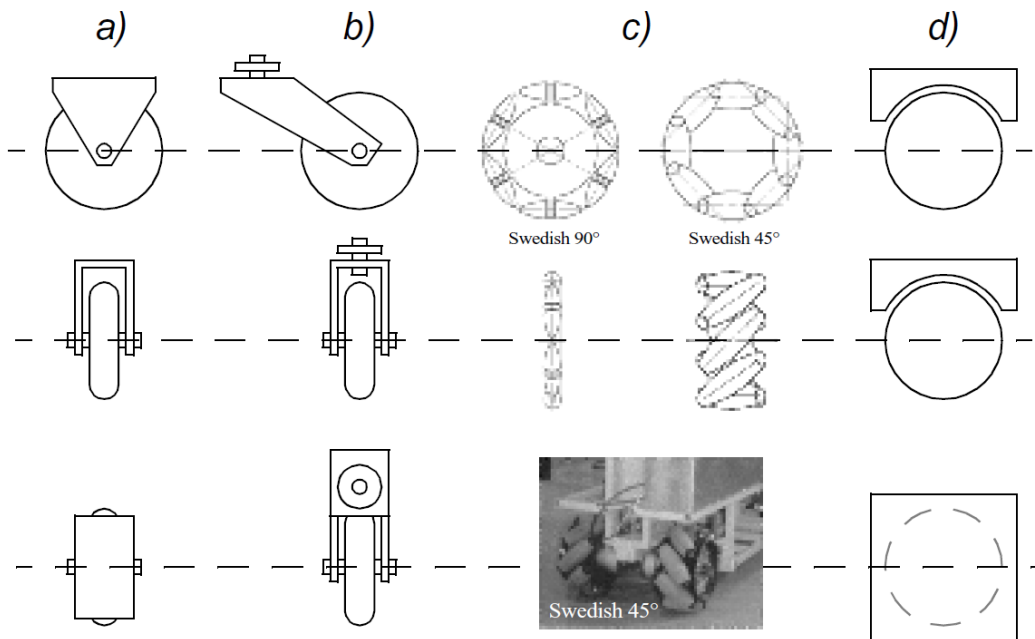
Tato dělení budou použita v následujících částech. Vzhledem k povaze frameworku ROS byly také zmíněny pravděpodobnostní modely, představené v [4].

3.2. Roboty pohybující se po souši

V této části bylo prozkoumáno co nejvíce možných uspořádání robotů. Roboty vhodné k implementaci do frameworku ROS byly popsány detailněji.

3.2.1. Kolové roboty

Mobilní roboty s kolovým podvozkem patří mezi nejvíce rozšířené platformy. Kolové podvozky jsou oproti ostatním relativně jednoduché, dostupné jak cenově, tak i z konstrukčního hlediska. Existuje celá řada možných uspořádání. Z toho pak vyplývají vlastnosti podvozku.



Obrázek 3.1: Základní typy kol [5]

Typy kol

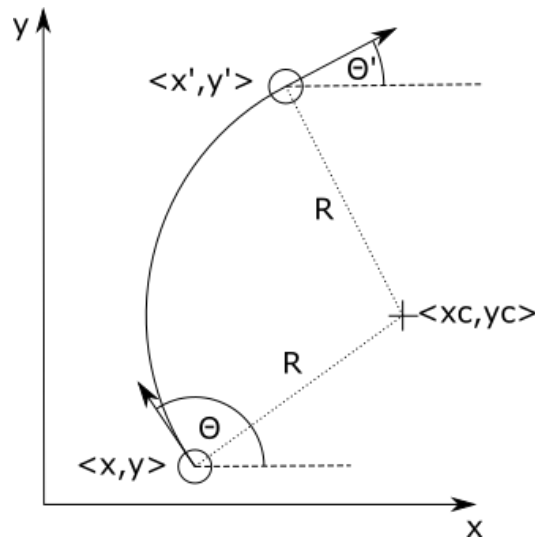
Kolo je základní prvek podvozku. Stavba kola dále určuje možné konfigurace a vlastnosti. Existuje několik typů kol. [5] zmiňuje čtyři hlavní skupiny:

1. Standardní kolo
Nákres standardního kola je možné vidět na obr.3.1 a). Horizontální i vertikální osa rotace prochází středem kola. Kolo má dva stupně volnosti. Ideální pohyb je ve směru valení, jinak dochází ke skluzu a je potřeba kolo natočit ve směru pohybu.[5]
2. Vlečené kolo
Oproti standardnímu kolu je jeho vertikální osa posunuta mimo střed kola. Kolo má dva stupně volnosti. Při pohybu se samo natáčí ve směru pohybu kvůli excentricitě vertikální osy. Nákres je vidět na obr.3.1 b). [5]
3. Všesměrové článkové kolo
Základní tvar je shodný se standardním kolem, obvod kola je ovšem tvořen valivými

články. Osa rotace článku svírá s osou kola nenulový úhel, nejčastěji 45° nebo 90°. Kolo má tři stupně volnosti, uspořádání článků dovoluje pohyb všemi směry. Nákres je na obr.3.1 c). Mezi nevýhody patří vyšší cena a menší odolnost vůči nečistotám. [5]

4. Všesměrové sférické kolo

Kolo je tvořeno koulí, schopnou pohybu do všech směrů bez omezení, má tři stupně volnosti. Spojení kulové části a podvozku nejčastěji zajišťují další valivé elementy. K nevýhodám patří složitá konstrukce a špatná průchodnost terénem. Nákres je na obr.3.1 d). [5]



Obrázek 3.2: Nákres pohybu po kružnici

Jednokolové podvozky

Konkrétní podvozek vznikne spojením základny a určitého počtu kol. Nejméně lze mít jedno kolo. Takový robot je ovšem velmi nestabilní. Pokud není těžiště robotu pod osou kola a kolo nemá dostatečnou šířku, pak je nutné zajistit dynamickou stabilitu. Z těchto důvodů se toto uspořádání v mobilní robotice téměř nepoužívá. Při vzpřímeném postoji se robot pohybuje po přímce, rychlost odpovídá obvodové rychlosti kola. Při náklonu robotu se trajektorie změní na kružnici. Pro ustálený pohyb musí platit, že odstředivá síla vyrovná vodorovnou složku reakce podložky. Při pohybu musí být zaručena i dynamická stabilita ve směru pohybu. Pokud je možné určit poloměr pohybu, pak úhlová rychlost pohybu je:

$$\omega = \frac{v}{R} \quad (3.1)$$

kde v je obvodová rychlost a R je poloměr pohybu. Ze znalosti v , ω , původní polohy x, y , natočení θ a délky pohybu Δt lze určit výslednou pozici x', y', θ' pomocí rovnice pohybu po kružnici v kartézských souřadnicích [4]:

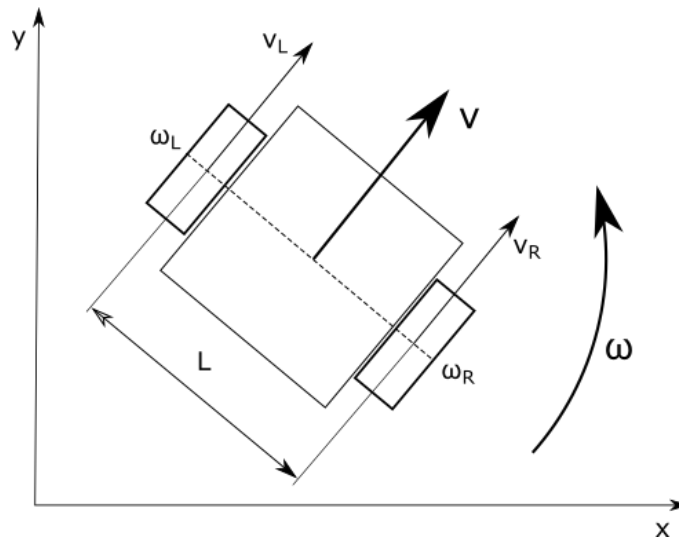
$$\begin{pmatrix} x' \\ y' \\ \omega' \end{pmatrix} = \begin{pmatrix} x \\ y \\ \omega \end{pmatrix} + \begin{pmatrix} -\frac{v}{\omega} \sin \theta + \frac{v}{\omega} \sin(\theta + \omega \Delta t) \\ \frac{v}{\omega} \cos \theta - \frac{v}{\omega} \cos(\theta + \omega \Delta t) \\ \omega \Delta t \end{pmatrix} \quad (3.2)$$

Schéma pohybu je na obr.3.2, bod o souřadnicích x_c, y_c je střed otáčení.

Tyto podvozky je možné do frameworku ROS implementovat, ale není to optimální řešení. Z důvodu nestability se neustále mění poloha robotu a je potřeba neustálý výpočet polohy. To je náročné na výpočetní výkon a případné ladění systému. Ve většině aplikací je vhodné tento podvozek nahradit staticky stabilním podvozkem.

Dvoukolový (diferenciální) podvozek

Dále je možné mít dvoukolový robot. I zde je těžké zajistit statickou stabilitu, buďto je potřeba mít těžiště robotu pod osou kol nebo stabilitu zajistit dynamicky. Z tohoto důvodu je podvozek často doplňován o pomocná všesměrová kola, tato konfigurace bude zmíněna později. Výhoda dvoukolového podvozku je v jeho jednoduchosti, je potřeba malý počet pohyblivých částí. Naproti tomu je ale potřeba velké množství senzorů pro zajištění stability a kvalitní řídicí algoritmus. [5] Oproti jednokolovému robotu stačí řešit statickou rovnováhu pouze ve směru pohybu. Schéma jednoduchého diferenciálního robotu je na obr.3.3.



Obrázek 3.3: Schéma dvoukolového podvozku [6]

$$v = \omega R \quad (3.3)$$

Pohyb se odvíjí od otáček jednotlivých motorů. Je-li obecný pohyb daný rovnicí 3.3, pak lze rychlost jednotlivých kol spočítat z otáček ω a jejich poloměru pomocí rovnic:

$$v_L = \omega_L r = \left(R - \frac{L}{2}\right)\omega \quad (3.4)$$

$$v_R = \omega_R r = \left(R + \frac{L}{2}\right)\omega \quad (3.5)$$

kde v jsou rychlosti a r je poloměr kol, L je vzdálenost kol. Přepočtem pak lze dostat rovnice pro v a ω :

$$v = \frac{r(\omega_R + \omega_L)}{2} \quad (3.6)$$

$$\omega = \frac{r(\omega_R - \omega_L)}{L} \quad (3.7)$$

Jedná se o pohyb po kružnici, platí rovnice 3.2. I tento podvozek je možné implementovat do frameworku ROS. Problém je opět se stabilitou, je potřeba řešit dynamickou stabilitu, která se ovšem mění, když je robot v klidu nebo pohybu. Ve většině aplikací je vhodné tento podvozek nahradit staticky stabilním podvozkem.

Vícekolové roboty

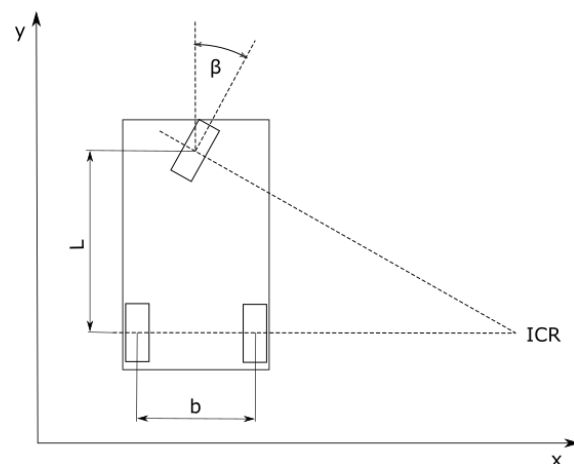
Nejčastěji mají podvozky tři nebo čtyři kola. Pokud kola neleží na jedné přímce a těžiště leží v prostoru mezi koly, tak je zaručena statická stabilita na rovném, nenakloněném povrchu. Existuje mnoho možných uspořádání, zde budou zmíněny nejčastější varianty. Pro dosažení statické stability stačí pouze tři kola, čtyřkolové varianty ale nabízí lepší stabilitu při pohybu a vyšší únosnost. Vlastnosti jednotlivých uspořádání budou zmíněny u konkrétních podvozků.

Diferenciální podvozek s pomocnými koly

Základ tvoří dvě kola, která mají společnou osu rotace. Doplněna jsou o pomocná všesměrová kola, zajišťující stabilitu. Nejčastěji jsou použita vlečná kola, méně často pak sférická kola. Platí stejné rovnice jako u dvoukolového diferenciálního podvozku, tedy 3.2, 3.6 a 3.7. Podvozek má stejné vlastnosti jako dvoukolová varianta, navíc je stabilní. Mezi výhody podvozku patří již zmíněná schopnost rotace na místě, dále je to jednoduchá konstrukce a jednoduché řízení. Toto uspořádání je vhodné pro framework ROS, umožňuje snadnou implementaci s ostatními částmi, protože neobsahuje restrikce pohybu z hlediska plánování trajektorie.

Ackermannův podvozek

Nejčastější uspořádání je s dvěma standardními koly a jedním otočným kolem. Hnané může být libovolné kolo, nejčastěji jsou hnaná dvě standardní kola, směrové kolo hnané není. Hnaná kola pak musí být spojena diferenciálem nebo musí mít samostatné osy. Pohled na základní uspořádání je na obr.3.4 Pokud je nežádoucí prokluz kol, tak musí



Obrázek 3.4: Schéma tříkolového Ackermannova podvozku [7]

být splněna Ackermannova podmínka [8] a to, že osy rotace všech kol musí směřovat do okamžitého středu pohybu (ICR). Pokud všechna kola směřují stejným směrem, tak je trajektorie podvozku přímka. Pokud je směrové kolo natočeno pod určitým úhlem, tak se

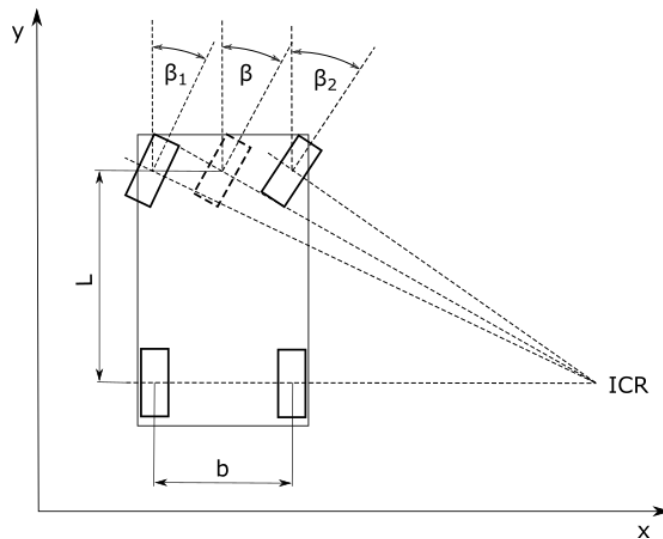
jedná o pohyb po kružnici. Za předpokladu, že zadní náprava zajišťuje pohyb rychlostí v , tak lze dopočítat úhlovou rychlost:

$$\omega = \frac{v \tan \beta}{L} \quad (3.8)$$

kde L je vzdálenost přední a zadní nápravy. Poté opět platí rovnice pohybu po kružnici 3.2. Limitní případ nastává v momentě, kdy je úhel směrového kola 90° . Pohyb je pak závislý na uspořádání hnaných kol. Pokud je mezi nimi diferenciál, tak je umožněna rotace téměř na místě (rotace s malým poloměrem), kola se otáčejí proti sobě, v opačném případě hrozí zamezení pohybu nebo ke smyku kol. Mezi výhody podvozku patří jednoduchost a obvykle vyšší únosnost než u diferenciálního podvozku. Při neuvažování smyku lze dobře předvídat pohyb robotu. Hlavní nevýhoda je v nemožnosti rotace na místě (obvykle nelze dosáhnout velkého úhlu směrového kola). Z tohoto důvodu je podvozek málo vhodný pro framework ROS. Při plánování pohybu je také potřeba počítat s omezeními podvozku, jako je minimální poloměr otáčení. Také je potřeba řešit výslednou polohu robotu, protože nelze docílit výsledné orientace pouhou rotací na konci pohybu. Tento problém se stává velmi složitým při řešení pohybu v limitovaných prostorách.

Čtyřkolový Ackermannův podvozek

Jedná se o podobné uspořádání jako u předchozího podvozku, náčrt je na obr.3.5. Směrové kolo je zde nahrazeno dvěma koly. Při pohybu bez prokluzu musí platit Ackermannova podmínka [8].



Obrázek 3.5: Čtyřkolový ackermannův podvozek [7]

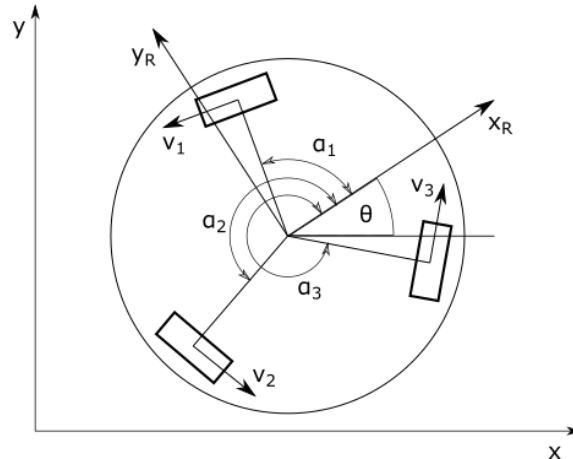
Při modelování pohybu lze tento podvozek nahradit podvozkem tříkolovým. Platí rovnice 3.8. Zavedením virtuálního kola uprostřed přední nápravy lze spočítat potřebné úhly natočení jednotlivých kol:

$$\beta_1 = \cot^{-1} \left(\cot \beta - \frac{b}{2L} \right) \quad (3.9)$$

$$\beta_2 = \cot^{-1} \left(\cot \beta + \frac{b}{2L} \right) \quad (3.10)$$

Všesměrové podvozky

Všesměrové podvozky využívají k pohybu všesměrových kol. To jim umožňuje měnit polohu v jakémkoliv směru, nebo provést jízdu i ve směru kolmém k ose pohybu kol. Na obr.3.6 je typické uspořádání tříkolové varianty. Kola jsou rozmístěna po kruhu, nejčastěji jsou od sebe natočena o 120° .



Obrázek 3.6: Všesměrový podvozek [6]

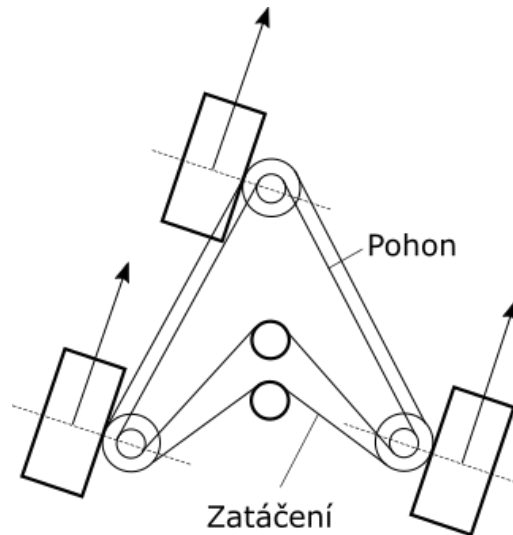
[6] pro tento typ podvozku udává následující rovnici:

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = \begin{bmatrix} \frac{v_1}{r} \\ \frac{v_2}{r} \\ \frac{v_3}{r} \end{bmatrix} = \frac{1}{r} \begin{bmatrix} -\sin(\theta + \alpha_1) & \cos(\theta + \alpha_1) & R \\ -\sin(\theta + \alpha_2) & \cos(\theta + \alpha_2) & R \\ -\sin(\theta + \alpha_3) & \cos(\theta + \alpha_3) & R \end{bmatrix} \begin{bmatrix} \dot{x}_G \\ \dot{y}_G \\ \dot{\theta} \end{bmatrix} \quad (3.11)$$

Kde ω je úhlová rychlost jednotlivých kol, v je rychlost jednotlivých kol, r je poloměr kol, θ je natočení přední části robotu, úhly α značí natočení os kol vůči lokálnímu souřadnému systému x_R, y_R a R je matice rotace mezi lokálním a hlavním souřadným systémem. Rotace a translace je docíleno protiběžným chodem kol, pohyb pak probíhá odvalováním válečkových segmentů kol. Konkrétní kinematická závislost se pak odvíjí od konkrétního uspořádání podvozku a vlastností kol. Všesměrové podvozky poskytují nejlepší manévrovatelnost ze zmíněných podvozků. Podvozek je vhodný pro implementaci do frameworku ROS. Nevýhody jsou v potřebě složitějšího řídicího algoritmu. Také je zvýšená potřeba přesnosti lokalizace, protože vlivem tření může dojít k nečekaným chybám při pohybu. Konstrukce je také drahá a kola nejsou vhodná do terénu.

Synchronní podvozek

Synchronní uspořádání je specifické, všechna kola jsou směrová, ale jsou navzájem propojena. Synchronní natáčení kol umožňuje pohyb do všech směrů, ale samotný podvozek si zachovává konstantní orientaci. Velmi často jsou všechna kola i hnaná pro zamezení nechtěného smyku. Na obr.3.7 je náčrt uspořádání s třemi koly, lze ovšem použít i vyšší počet kol pro zvýšení únosnosti a stability.



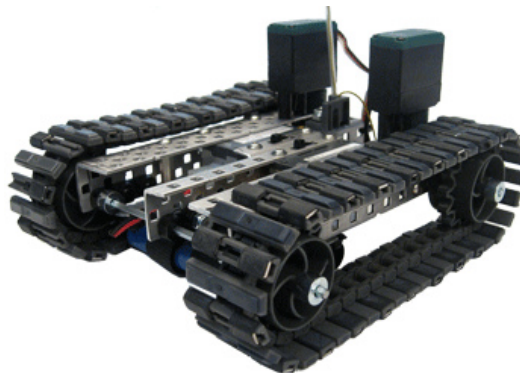
Obrázek 3.7: Synchronní podvozek [7]

Ostatní uspořádání kolových robotů

Pro tvorbu podvozku lze použít libovolnou kombinaci kol (při zachování pohyblivosti). Obvyklé jsou podvozky s více než čtyřmi koly pro zvýšení únosnosti nebo zlepšení prostupnosti terénu. Více informací lze najít v [5] a [9]

3.2.2. Pásové podvozky

Pásové podvozky jsou principiálně velmi podobné diferenciálním podvozkům. Jak už název napovídá, jsou k pohybu použity pásy. Nejčastěji jsou nataženy mezi dvěma koly. Pro zvýšení únosnosti mohou být mezi hlavními koly i kola doplňková. Rozdíl je v rozměru kontaktní plochy mezi podvozkem a povrchem země. Kontakt je po celém povrchu spodní části pásu. Proto vzniká při pohybu tření a pohyb je energeticky náročný. Výhodou je vysoká prostupnost terénem a dobrá stoupavost. [10] Příklad pásového podvozku je na obr.3.8.



Obrázek 3.8: Ukázka pásového podvozku [11]

Při zanedbání smyku jsou kinematické rovnice shodné s rovnicemi diferenciálního podvozku, tedy 3.2, 3.6 a 3.7. Při reálném pohybu ovšem vlivem smykového odporu vznikají odchylky. Velikost a charakter odchylek je závislý na aktuálním rozložení váhy robotu a charakteru terénu, velký vliv má také poměr mezi rozvorem a délkou pásů. Díky po-

dobným vlastnostem jako má diferenciální podvozek, tedy schopností otáčení na místě a relativně jednoduchému základnímu kinematickému modelu je tento podvozek vhodný k implementaci do frameworku ROS. Kvůli smyku je ovšem potřeba přesný lokalizační aparát. V nepříznivém prostředí se ovšem mění vlastnosti a podvozek je těžko ovladatelný, proto se často používá k řízení teleoperátor. [10]

3.2.3. Kráčející roboty

Kráčející roboty pro pohyb používají nohy. Principiálně se snaží napodobit biologické systémy, nejčastěji hmyz, savce nebo lidi. Vlastnosti jednotlivých robotů velmi závisí na počtu noh a konkrétním uspořádání. Obecně se tento typ vyznačuje dobrou prostupností terénu. Kráčející roboty zvládají velmi členitý terén, jsou schopné překonat relativně vysoké překážky nebo prohlubně. Naopak hlavní nevýhoda je ve velkém počtu stupňů volnosti, což znamená vysokou složitost řídicího algoritmu. Také hardware je složitý a nákladný, může docházet k oscilacím a podvozek se špatně oživuje. [10] Na obr.3.9 je vidět typický šesti-nohý robot.



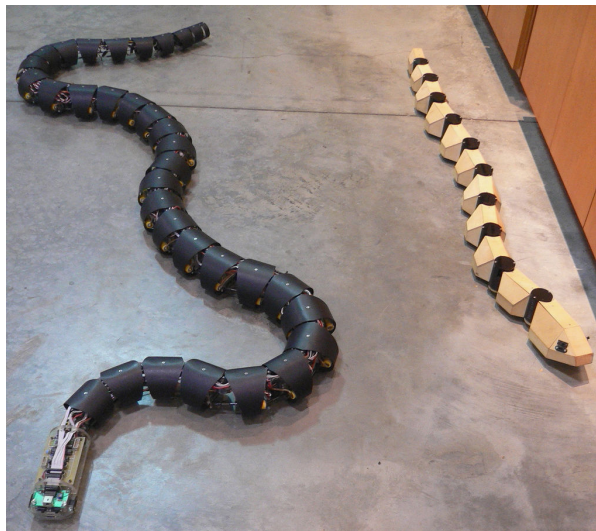
Obrázek 3.9: Příklad kráčejícího robotu [12]

Kráčející roboty obvykle mají 1 až 8 noh. Jednonohé roboty jsou zastoupeny minimálně. Dvounohé roboty nejčastěji kopírují stavbu lidského těla a snaží se napodobit chůzi. Stability při pohybu mohou dosahovat přesunem těžiště a nebo mohou spoléhat na dynamickou stabilitu. [5] Třínohé a pětínohé roboty jsou spíše kuriozita, ale je například snaha naučit šestínohé roboty chodit s menším počtem noh v případě poruchy. [13] Nejčastější jsou čtyř a šestínohé roboty, protože nabízí dostatečnou stabilitu i při pohybu. Kinematika pohybu se odvíjí od algoritmu chůze. Robot může spoléhat na dynamickou stabilitu, nebo naopak může vhodným přesunem těžiště dosáhnout statického stavu, kdy tři nohy tvoří stabilní postoj a zbytek nohou se přesouvá na další pozici. [5] Více než šestínohé systémy vynikají stabilitou i při pohybu, ale s rostoucím počtem nohou roste počet možných stavů robota a řídicí algoritmus je velmi složitý. Obecně je také problém s deterministickým chováním, neboť koncové polohy končetiny lze dosáhnout více způsoby. [5] uvádí, že počet všech možných úkonů robota N , jako je zvednutí nebo položení nohy, je úměrný počtu nohou se závislostí $N = (2k - 1)!$, kde k je počet nohou. Je vidět, že s rostoucím počtem nohou prudce roste složitost pohybu. Kráčející roboty jsou schopné otáčení na místě a při dostatečném počtu nohou nabízí statickou stabilitu. Jsou proto vhodnými kandidáty pro implementaci do frameworku ROS. Z důvodu vysoké náročnosti na řídicí algoritmus ovšem příliš výhodné nejsou. Složitost řídicího algoritmu chůze může

být srovnatelná s algoritmy lokalizace, což je překážka u systému s malým výpočetním výkonem.

3.2.4. Plazivé roboty

Tyto roboty kopírují pohyby hadů. Skládají se z několika válcových článků, mezi sebou jsou články spojeny pohyblivou vazbou. Hlavní výhodou je možnost použití v úzkých prostorách, tyto roboty se proto používají na inspekční práce v potrubích nebo těžko přístupných místech. Roboty se mohou pohybovat plazením, jednotlivé články také mohou mít kola pro rychlejší pohyb. Při zajištění dostatečné tuhosti konstrukce je robot schopen překonávat i výškové překážky. Nevýhodou je poměrně malá rychlost pohybu a velký počet pohyblivých částí.[10] Tyto roboty jsou vhodné pro operace s teleoperátorem. Ve většině případů jsou propojeny kabelem s řídicí obsluhou. Pohyb robotu je složitý, vhodnost pro framework ROS je proto velmi malá. Vzhledem k velkému množství částí by byl simulační model velmi složitý.



Obrázek 3.10: Ukázka plazivých robotů [14]

3.2.5. Šplhající roboty

Jde o modifikaci kráčejších robotů. Na koncích noh jsou umístěny mechanismy umožňující pohyb po vertikálních plochách. Velmi rozšířené jsou přísavky, které ovšem vyžadují rovný povrch pro získání dostatečné přísavné síly. Druhá možnost je použití elektromagnetů, ale je potřeba, aby podložka byla z feromagnetického materiálu. Z hlediska pohybu se opět jedná o složité systémy, implementace je náročná. Také ve většině případů se předpokládá, že země je pevná a vertikální objekty představují překážky. Nelze tedy zaručit stabilitu algoritmů.

3.2.6. Skákající roboty

Tyto podvozky jsou charakteristické použitím mechanismů, které jsou schopné vydat velké množství energie za krátký čas. Nejčastěji jsou k tomuto účelu použity pružiny nebo píсты.

Skok udělí robotu dostatečnou kinetickou energii, aby se odpoutal od země. Při zanedbání odporu vzduchu pak robot opisuje parabolickou trajektorii. Výhoda těchto robotů je schopnost překonat překážky, které jsou několikanásobné oproti velikosti robotu. [10] Implementace do frameworku ROS je teoreticky možná, ale velmi omezená. Skok představuje energeticky náročný úkon, robot je tedy stavěn co nejlehčí, aby se snížila potřebná energie. To znamená malý použitelný výkon řídicí jednotky a tedy i zhoršené lokalizační schopnosti.

3.2.7. Hybridní roboty

Tyto roboty kombinují více způsobů pohybu. Výsledný podvozek pak má vlastnosti od každého typu pohybu. Popis takového systému je velmi závislý na konkrétní konfiguraci. Například robot Handle firmy Boston Dynamics používá pro pohyb kola, ale jejich blokáci dokáže přejít do chůze nebo může dokonce skákat [15].

3.2.8. Pravděpodobnostní modely

Doposud zmíněné modely byly závislé na konkrétním uspořádání podvozků robotů. [4] zavádí nový způsob modelování pohybu. Zjednodušeně se jedná o aplikaci pravděpodobnostních rovnic na klasické modely, zmíněné v předcházejících kapitolách. Vychází z předpokladu, že daný jev, například rychlost robota nebo řídicí signál, se může v realitě mírně lišit od žádané hodnoty a tím vzniká nepřesnost pohybu. Tyto nepřesnosti podléhají zákonům pravděpodobnosti a řídí se určitým pravděpodobnostním rozdělením. Znamená to tedy, že reálná hodnota se bude pohybovat kolem hodnoty očekávané, pravděpodobnost výskytu určité hodnoty bude klesat s rostoucí odchylkou od teoreticky přesné hodnoty.

Pravděpodobnostní teorie je aplikovatelná na objekty v trojrozměrném prostoru. Z důvodů jednoduchosti a použitelnosti kniha pracuje s dvourozměrným prostorem. Stejně jako u předchozích modelů je stav robota definován trojicí parametrů:

$$x_T = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} \quad (3.12)$$

kde x a y jsou souřadnice robota a θ je jeho natočení, měřené od osy x . Pravděpodobnostní přístup vyšetřuje, s jakou pravděpodobností robot dojde na určité místo po odeslání příkazu pohybu. Tento mechanismus slouží k doplnění algoritmu autonomního pohybu, více informací o lokalizaci a plánování je v [4]. [4] předkládá dva hlavní modely: Odometrický model (z ang. Odometry model) a rychlostní model (z ang. Velocity model). Oba mají své klady a zápory, také vhodnost použití je rozdílná, následující část přiblíží oba modely.

Rychlostní (velocity) model

Předpokladem je, že je možné robot řídit pomocí dvou příkazů. Konkrétně se jedná o translaci v , kdy kladná hodnota představuje pohyb vpřed, a rotaci ω , kdy kladná hodnota představuje rotaci proti směru hodinových ručiček. Výhoda modelu je možnost predikce pohybu, což se využívá u plánování trajektorie.

Při přesném pohybu je předpokládáno deterministické chování. Řídící příkaz pak vypadá následovně:

$$u = \begin{pmatrix} v \\ \omega \end{pmatrix} \quad (3.13)$$

Při nenulových příkazech je výsledná trajektorie pohybu kružnice o poloměru:

$$R = \frac{v}{\omega} \quad (3.14)$$

Pro hodnoty ω blíží se k nule roste poloměr do vysokých hodnot, při $\omega = 0$ přerostl poloměr nade všechny meze a jedná se o pohyb po přímce. Schéma pohybu je na obr.3.2. Pro pohyb po kružnici opět platí rovnice 3.2.

Ve skutečnosti jsou řídicí příkaz i samotný pohyb zatížené odchylkami a chybami. Hlavní rozdíl oproti deterministickým modelům je v zavedení fiktivních příkazů [4]:

$$\begin{pmatrix} \hat{v} \\ \hat{\omega} \end{pmatrix} = \begin{pmatrix} v \\ \omega \end{pmatrix} + \begin{pmatrix} \varepsilon_{\alpha_1|v|+\alpha_2|\omega|} \\ \varepsilon_{\alpha_3|v|+\alpha_4|\omega|} \end{pmatrix} \quad (3.15)$$

K původnímu příkazu omega je přičtena náhodná hodnota z pravděpodobnostního rozdělení. Nejčastěji se používá Gaussovo normální rozdělení nebo trojúhelníkové rozdělení. Střed rozdělení je v nule.

Výpočet výsledného stavu robota je pak shodný jako při určení přesného pohybu, ale při každém použití je vrácena poloha s určitou odchylkou. Výpočet ukazuje rovnice [4]:

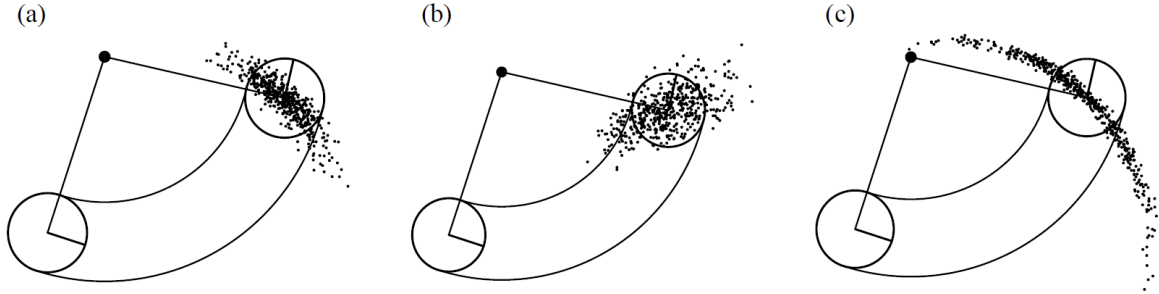
$$\begin{pmatrix} x' \\ y' \\ \omega' \end{pmatrix} = \begin{pmatrix} x \\ y \\ \omega \end{pmatrix} + \begin{pmatrix} -\frac{\hat{v}}{\hat{\omega}} \sin\theta + \frac{\hat{v}}{\hat{\omega}} \sin(\theta + \hat{\omega}\Delta t) \\ \frac{\hat{v}}{\hat{\omega}} \cos\theta - \frac{\hat{v}}{\hat{\omega}} \cos(\theta + \hat{\omega}\Delta t) \\ \hat{\omega}\Delta t \end{pmatrix} \quad (3.16)$$

Ze vztahu 3.15 je vidět, že velikost odchylky je určena konstantami α , které jsou specifické pro určitý robot, a velikostí řídicího příkazu. Rychlejší pohyb tedy za stejný časový úsek vyvolá vyšší nepřesnost, ale na delší dráze. Samotný model má dvě použití. Prvním z nich je určení pravděpodobnosti výsledné pozice robota na základě původního stavu a hodnoty příkazu. Toho se využívá v souvislosti s pravděpodobnostním algoritmem plánování trasy, kdy robot předpokládá svoji původní pozici, na základě řídicího signálu spočítá možné následující stavy a následně se je snaží ověřit pomocí senzorů. Více opět v [4]. Druhé použití modelu je možnost generovat vzorek (sample) polohy. Soubor takovýchto vzorků pak slouží podobně jak v prvním případě k doplnění algoritmu plánování pohybu. Rozdíl je v tom, že se již nejedná o kontinuální rozdělení pravděpodobnosti a musí být použity jiné algoritmy. Na obr.3.11 lze vidět různé rozložení vzorků v závislosti na parametrech alfa. Zkušenosti z praxe ukazují, že pohyb po kružnici nemusí být vždy s konstantním poloměrem. Dochází pak k odchylce výsledné orientace robota. Proto byly navíc zavedeny parametry α_5 a α_6 , které jsou součástí fiktivní rotace na konci pohybu [4]:

$$\hat{\gamma} = \varepsilon_{\alpha_5|v|+\alpha_6|\omega|} \quad (3.17)$$

Výsledný výpočet má pak tvar [4]:

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} -\frac{\hat{v}}{\hat{\omega}} \sin\theta + \frac{\hat{v}}{\hat{\omega}} \sin(\theta + \hat{\omega}\Delta t) \\ \frac{\hat{v}}{\hat{\omega}} \cos\theta - \frac{\hat{v}}{\hat{\omega}} \cos(\theta + \hat{\omega}\Delta t) \\ \hat{\omega}\Delta t + \hat{\gamma}\Delta t \end{pmatrix} \quad (3.18)$$



Obrázek 3.11: Příklady rozložení vzorků pro různé hodnoty alfa - rychlostní model [4]

Odometrický model

Tento model využívá informací z enkodérů umístěných na kolech robotu. Tímto způsobem lze určit pohyb robotu. Výhoda enkodérů je jejich vyšší přesnost oproti rychlostnímu modelu. Nevýhoda je ve skutečnosti, že získané informace jsou dostupné až po vykonání pohybu. To není problém pro určení výsledné pozice, ale znemožňuje to přesné plánování trajektorie. Více informací o enkodérech je možné získat v [16]. Správně by se data z enkodérů měla brát jako měření. Pro potřeby modelu je s nimi nakládáno jako s požadovanými hodnotami pohybu. Problém odometrického modelu je nemožnost určit pohyb robotu v absolutní souřadné soustavě. Odometrie udává pouze relativní posun vůči vnitřnímu souřadnému systému robotu, jehož poloha vůči okolí je neznámá. [4] simuluje pohyb pomocí dvou rotací a jedné translace. První rotace je mezi původní orientací a spojnicí předchozích a následujících souřadnic robotu. Následuje translace po této spojnici. Pohyb je ukončen druhou rotací, která robotu udělí výsledný směr. Schéma pohybu je na obr.3.12.

Příkaz k pohybu je pak definovaný jako rozdíl dvou po sobě jdoucích stavů \bar{x}_{t-1} a \bar{x}_t :

$$u_t = \begin{pmatrix} \bar{x}_{t-1} \\ \bar{x}_t \end{pmatrix} \quad (3.19)$$

Teoreticky přesné hodnoty posuvů jsou [4]:

$$\delta_{rot1} = atan2(\bar{y}' - \bar{y}, \bar{x}' - \bar{x}) - \bar{\theta} \quad (3.20)$$

$$\delta_{trans} = \sqrt{(\bar{x} - \bar{x}')^2 + (\bar{y} - \bar{y}')^2} \quad (3.21)$$

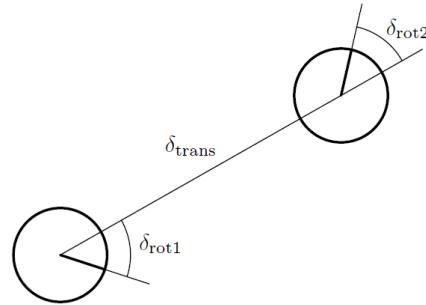
$$\delta_{rot2} = \bar{\theta}' - \bar{\theta} - \delta_{rot1} \quad (3.22)$$

Pravděpodobnostní přístup pak stejně jako u velocity modelu přidává určitou odchylku. Konkrétně se jedná o odečtení „rušení“ o nulové střední hodnotě s rozptylem daným parametry α_1 až α_4 . Tyto parametry jsou charakteristické pro každý robot, fiktivní posuvy pak jsou [4]:

$$\hat{\delta}_{rot1} = \delta_{rot1} - \varepsilon_{\alpha_1|\delta_{rot1}|+\alpha_2|\delta_{trans}|} \quad (3.23)$$

$$\hat{\delta}_{trans} = \delta_{trans} - \varepsilon_{\alpha_3|\delta_{trans}|+\alpha_4|\delta_{rot1}+\delta_{rot2}|} \quad (3.24)$$

$$\hat{\delta}_{rot2} = \delta_{rot2} - \varepsilon_{\alpha_1|\delta_{rot2}|+\alpha_2|\delta_{trans}|} \quad (3.25)$$

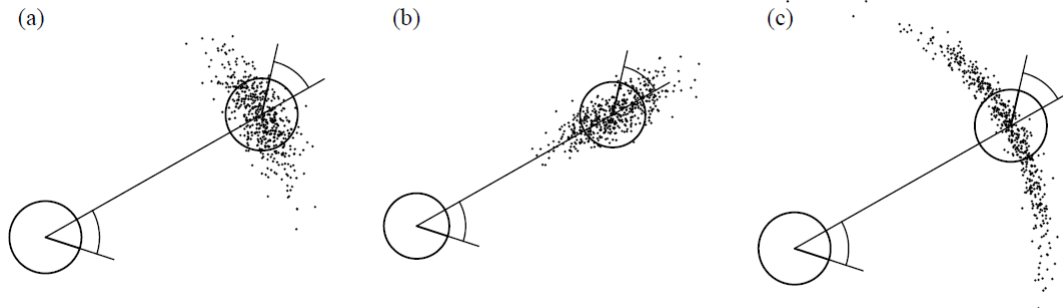


Obrázek 3.12: Schéma odometrického modelu [4]

Výsledkem je rovnice [4]:

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} \hat{\delta}_{trans} \cos(\theta + \hat{\delta}_{rot1}) \\ \hat{\delta}_{trans} \sin(\theta + \hat{\delta}_{rot1}) \\ \theta + \hat{\delta}_{rot1} + \hat{\delta}_{rot2} \end{pmatrix} \quad (3.26)$$

Tímto způsobem lze opět dostat vzorek, mračno těchto vzorků tvoří částicové rozdělení pravděpodobnosti. Příklady pro různé hodnoty alfa lze vidět na obr. 3.13. Při použití pravděpodobnostních rovnic z [4] lze také spočítat kontinuální pravděpodobnostní rozložení pro daný pohyb, výsledkem není mračno bodů, ale funkce nad dvourozměrným prostorem.



Obrázek 3.13: Příklady rozložení vzorků pro různé hodnoty alfa - odometrický model [4]

3.3. Mobilní roboty pohybující se po vodě, ve vzduchu nebo ve vesmírném prostoru

3.3.1. Mobilní roboty pohybující se ve vodě

Pro pohyb ve vodě se nejčastěji používají lodě nebo ponorky. Popis pohybu je velmi obtížný, ve většině případů se voda samovolně pohybuje, popřípadě ji ovlivňuje samotný pohyb robotu. Také je těžké zajistit přesnou lokalizaci pohybu, musí být použity speciální voděodolné senzory. Vlnění hladiny také může tvořit odlesky, což způsobuje chybné měření. Z tohoto důvodu budou zmíněny jen základní principy pohybu ve vodě.

Lodě

Lodě nebo plavidla jsou charakteristická tím, že využívají vztlakových sil pro setrvání na hladině. Potopení je ve většině případů nežádoucí. Pro pohyb se pak používají například vodní šrouby nebo plachty. Zcela obecně musí platit Newtonův druhý pohybový zákon. Tíha lodě je vykompenzována vztlakem ponořené části trupu, při zvýšení zatížení dojde k samoregulaci, kdy se trup více ponoří a vykompenzuje novou hmotnost. Pro rovinu hladiny je pak potřeba znát veškeré síly, které na loď působí. Výsledný pohyb je pak dán součtem těchto vlivů, mezi základní patří síly motorů, pohyby vody (vlny a vodní proudy) nebo síla větru [17]. V [17] je možné najít více informací o popisu pohybu malé loďky. Na obr.3.14 lze vidět prototyp soutěžní autonomní loďky.



Obrázek 3.14: Ukázka robotické loď [18]

Ponorky

Oproti lodím je žádoucí úplné ponoření do vody. Ve většině případů je kontakt se dnem vodní masy nežádoucí, ponorky jsou proto v kontaktu pouze s vodou. Výšková poloha je zajištěna vyvážením vztlakové a tíhové síly. Ponorky obsahují vaky, které jsou schopné měnit množství vytlačené vody a tím mohou regulovat vertikální pohyb. Stejně jako u lodí platí druhý Newtonův pohybový zákon, největší vliv mají vodní proudy. Ponorky jsou využívány k inspekčním pracím a průzkumným úkolům. Nejčastěji jsou ovládány teleoperátorem. Velmi často mají ve výbavě sonar, který zajišťuje základní informace o okolí robotu. [10] Framework ROS se primárně zaměřuje na roboty pohybující se na pevné zemi, lodě a ponorky proto nejsou vhodné k implementaci. Většina podporovaných senzorů není vodotěsná, je tedy i potřeba implementace senzorického systému.

3.3.2. Mobilní roboty pohybující se ve vzduchu

Pohyb ve vzduchu, neboli let, je charakteristický nemožností dlouhodobého staticky stabilního stavu. Téměř po celou dobu letu je potřeba kompenzovat vnější vlivy, není možné „zastavit“. Zákonitosti letu jsou také silně závislé na velikosti létajícího objektu a aktuálním stavu atmosféry. Z tohoto důvodu je velmi složité ovládat mobilní letouny. Ve většině případů je také potřeba velmi rychlá odezva řídicího systému. Tato podmínka jde bohužel proti snaze postavit velmi lehký letoun s omezeným vybavením.

Letadla

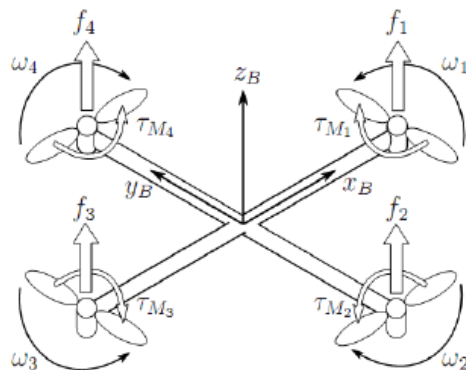
Stavba letadla se odvíjí od konkrétního účelu mise. Velmi často jde o kompromis a snahu o získání vlastností, na kterých nejvíce záleží. [19] nabízí základní přehled mechaniky letu. Pohyb je závislý na mnoha proměnných, velmi závisí na konkrétním okamžiku letu. Implementace letounů do frameworku ROS je náročná. Velmi často je potřeba z důvodu hmotnostního omezení přesunout výpočetní výkon mimo letadlo, což zvyšuje nároky na stabilní přenos informací mezi letounem a řídicí jednotkou. V omezených prostorách je potřeba rychlá odezva systému. Na obr. 3.15 je autonomní letoun, vyvinutý na platformě ROS.



Obrázek 3.15: Experimentální letoun Erleplane [20]

Multikoptéry

V poslední době se rozmáhá popularita multikoptér, konkrétně jsou nejvíce používány kvadrokoptéry nebo hexakoptéry. Jedná se o stroje, které přímo využívají tah motorů k vytvoření dostatečné síly pro udržení stroje ve vzduchu. K pohybu se tedy nevyužívá vzltlak vznikající na profilu křídla jako u letadla. Hlavní výhodou multikoptér je v možnosti „zastavení“ ve vzduchu. Naopak oproti letadlům mají multikoptéry nižší účinnost a tedy i kratší letový čas. Nákres základního uspořádání kvadrokoptéry je na obr.3.16.



Obrázek 3.16: Schéma čtyřrotorové multikoptéry [21]

Opět musí platit druhý Newtonův pohybový zákon. Kombinací sil f_1 až f_4 z obr.3.16 lze dosáhnout požadovaného pohybu. Opět lze rozlišit několik základních případů pohybu. Základní případ je prosté udržení stroje ve vzduchu. Součet sil musí vyrovnat tíhu. Zároveň se musí vyrovnat momenty, aby nedošlo k přetočení a pádu. Mezi další případy lze zahrnout vše od prostého otáčení kolem jednotlivých os až po složité manévry, vyžadující složitou souhru všech motorů. Velmi často lze žádaný pohyb realizovat více způsoby. Více informací o mechanice letu multikoptér je v [21]. S rostoucím výkonem dostupných multikoptér roste množství použitelných senzorů a výpočetního výkonu. Implementace

do frameworku ROS je velmi náročná, systém musí reagovat velmi pohotově. Kvůli povaze pohybu je těžké dosáhnout přesných a rychlých měření, algoritmy musí být stavěny velmi robustně. Velké omezení je z hlediska legislativy, za standardních okolností nelze provozovat kompletně autonomní letoun a je nutný přímý dohled operátora.

3.3.3. Mobilní roboty pohybující se ve vesmírném prostoru

Téměř všechny objekty ve vesmíru vyrobené člověkem se dají považovat za roboty. Plní různé úkoly, od jednoduchých satelitů po vozítka prozkoumávající cizí planety. Jen malá část robotů může být ovládána pomocí teleoperátora, zejména jde o lunární vozítka a satelity na oběžné dráze kolem Země. Většina je stavěna na téměř plnou autonomnost [10]. Pro vozítka pohybující se na cizích planetách platí stejné modely jako pro Zemské protějšky. Rozdíl je v nemožnosti korekce chyb v případě nešťastné události. Také se začíná projevovat velká doba odezvy. Proto je téměř nemožné řídit vozítka na Marsu přímo, je potřeba částečná autonomnost. Teleoperátor pak nastavuje úkoly, samotné splnění je na vozítku.[10]

Při pohybu kolem planety opisuje družice eliptickou dráhu. Ve speciálním případě se může jednat o kružnici. Při přeletu mezi tělesy nebo obecně při pohybu ve sluneční soustavě může družice opisovat parabolickou nebo hyperbolickou trajektorii [22]. Framework ROS je nevhodný pro tato použití, protože při pohybu vesmírem je vyžadována extrémní spolehlivost. Jakákoliv chyba může znamenat extrémně vysokou ztrátu.

4. Motivace k práci

Praktická část práce je zaměřena na popis pohybu podvozku modelářského auta. Tyto podvozky byly použity pro vývoj experimentálního semiautonomního konvoje. Získané výsledky by měly vést ke zpřesnění pohybu jednotlivých autonomních aut a lepší regulaci vedoucího auta, ovládaného člověkem.

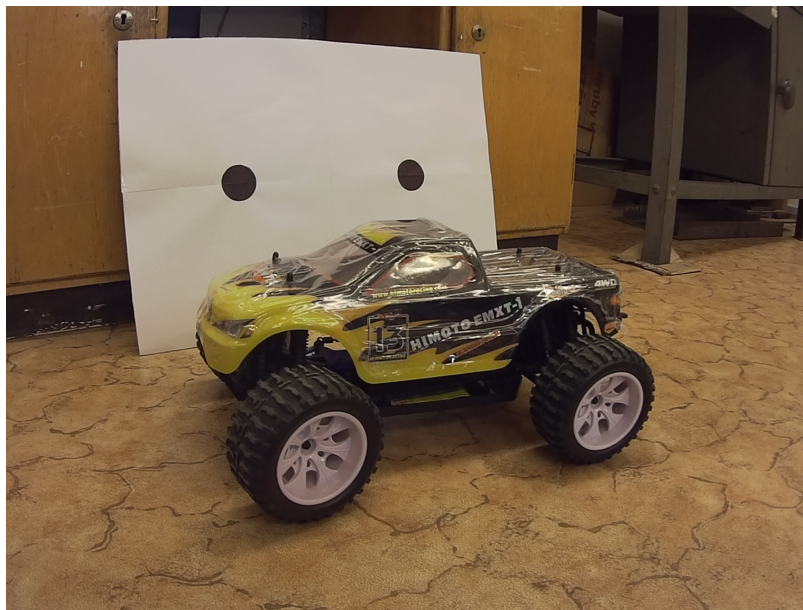
Zvolené podvozky nejsou k tomuto účelu příliš vhodné, protože jsou určeny pro mnohem vyšší rychlosti, než je potřeba pro konvoj, což má za následek špatnou možnost regulace při nízké rychlosti jízdy. Také není možnost měřit otáčky kol bez konstrukčního zásahu do podvozku. Cílem praktické části je tedy i zhodnotit dosažitelnou přesnost simulace.

5. Konkrétní řešení

Konkrétní řešení má za cíl vytvořit funkční model reálného podvozku. V části 5.1 jsou popsány cíle experimentu a měřený podvozek. Dále je v 5.2 navržen experiment pro získání dat potřebných pro model. Následuje zpracování dat v části 5.3.

5.1. Účel experimentu a popis měřeného objektu

Pro tuto práci byl jako předloha vybrán podvozek modelářského RC auta. Takovéto podvozky byly použity při realizaci experimentálního konvoje vozidel, kdy základ tvoří vedoucí auto, které je následováno auty autonomními. Vedoucí auto je ovládáno na dálku a udává trajektorii konvoje. Ostatní auta se snaží následovat, aniž by narazily do překážek, jako je například roh zatáčky. Vedoucí auto je možné ovládat původním dálkovým ovladačem, tato metoda ovšem nezaručuje plynulou jízdu, proto je potřeba hledat jiné řešení. Podvozky mají předinstalovaný hardware, který se vhodným signálem dá řídit. Pro jednoduchost bylo zvoleno ponechání všech komponent až na RC přijímač, který byl nahrazen platformou Arduino (pouze pro účely měření). Je proto potřeba najít vztah mezi potřebným vstupním signálem ponechaného hardwaru a odezvou podvozku. Samotný podvozek má řadu vůlí a mechanické vlastnosti závisí na velkém množství parametrů. Druhým cílem experimentu je tedy určit míru nepřesnosti pohybu podvozku. Pro popis pohybu je vhodný velocity model z kap.3.2.8, experimentem budou určeny konstanty α .

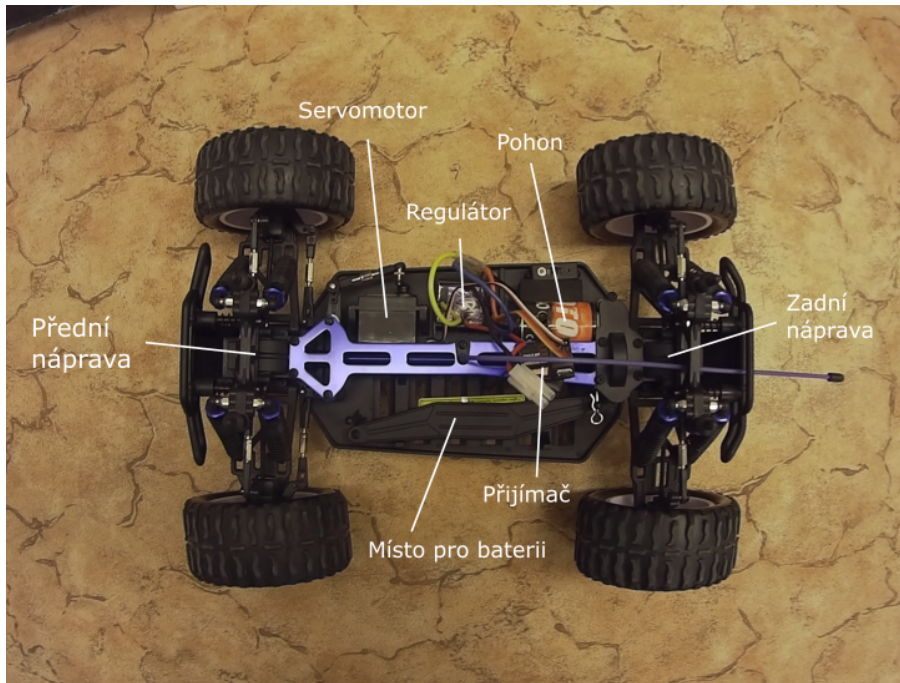


Obrázek 5.1: Původní podvozek

5.1.1. Popis měřeného podvozku

Jde o klasickou konstrukci dálkově řízeného auta v měřítku 1:10 značky Tamiya. Na obr.5.1 je pohled na původní modelářské auto, na obr.5.2 je detail podvozku včetně popisu hlavních částí. V základní konfiguraci je pohyb řízen dálkovým ovladačem, ovládaný je elektromotor a natáčení přední nápravy. V podvozku je pak umístěn přijímač, který přenos dekóduje

a vytvoří z něj dva standardní servo signály. Jeden je pro regulátoru otáček stejnosměrného motoru zajišťující pohyb vpřed a druhý pro servomechanismus přední nápravy. Celý systém je napájen dvoučlánkovým li-pol akumulátorem. Tato platforma vyniká stabilitou na nerovném povrchu, má jednoduchou konstrukci a vysokou únosnost. Také je vhodná pro pohyb ve vnějším prostředí a může dosahovat velkých rychlostí. Důvodem výběru bylo doporučení vedoucího práce, jednoduché řízení a dostupnost v laboratoři.

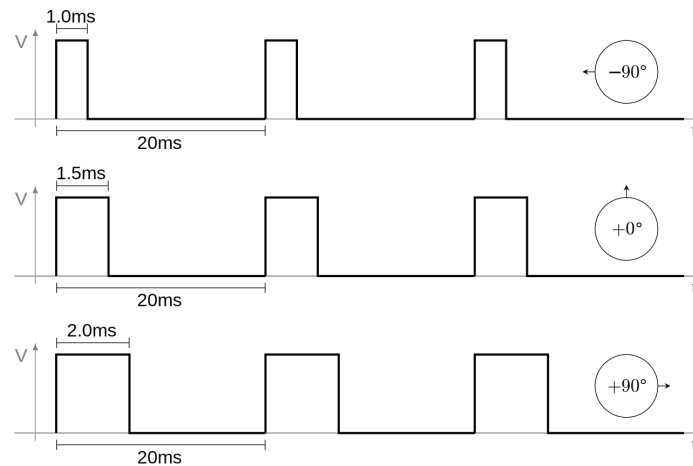


Obrázek 5.2: Popis základních částí podvozku

Servo signál

V následujícím textu bude mnohokrát použit termín servo signál, signál pro regulátor nebo řídicí signál pro servo/ regulátor. Jedná se o název téhož druhu signálu, používaného v modelářství a bude dobré tento termín více přiblížit. V podstatě se jedná o určitý druh pulzně šířkové modulace. Perioda signálu je pevně stanovena na 50 Hz, čemuž odpovídá délka periody 20ms. Obdélkový signál pak má střídu od 5 do 10%, což odpovídá délce 1 až 2ms. Střed signálu je při střídě 1,5ms. Tento rozsah pak odpovídá pohybu páčky servomotoru, obvyklá je možnost pohybu od 90° až do 180°, nebo v případě regulátoru rychlosti od plné jízdy vzad až po plný chod motoru vpřed. Speciální jednotky pak umožňují využití širšího pásma střídy, nebo nabízí jiné rozsahy pohybu páčky.[3] Typický průběh servo signálu je na obr. 5.3.

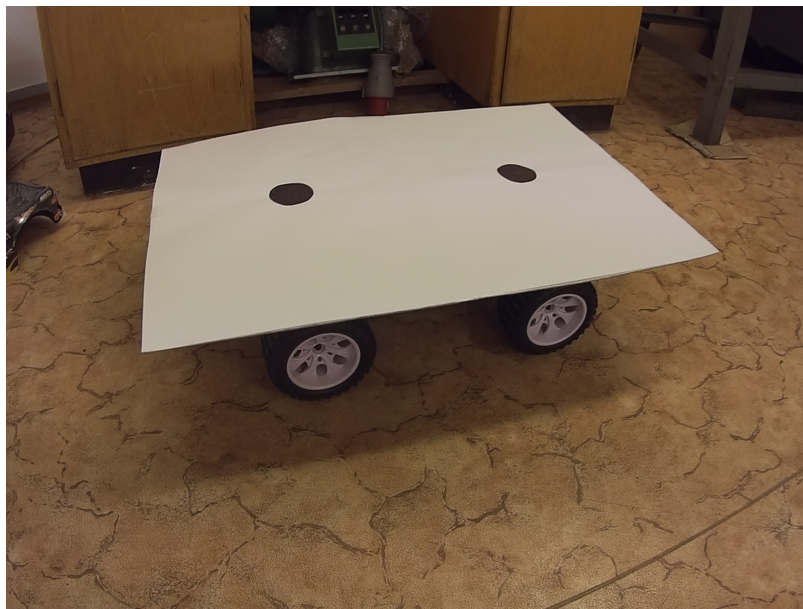
Pro vytvoření pohybového modelu jsou k dispozici dvě možnosti přiblížené v kapitole 3.2.8. Odometrický model vyžaduje data o ujeté vzdálenosti a natočení nápravy. To bohužel u zvoleného podvozku není možné, jelikož nemá zpětnou vazbu. Je proto potřeba použít velocity model, který pracuje s rychlostmi. Přesné hodnoty je těžké získat, ovšem s přiměřenou přesností je možné tyto data zjistit měřením bez nutnosti konstrukčního zásahu do podvozku.



Obrázek 5.3: Průběh standardního servo signálu [23]

5.2. Návrh měření a realizace

Cílem experimentu je získat reálná data pro zvolený podvozek. Velocity model počítá s rychlostí a úhlovou rychlostí. Je proto potřeba navrhnout měření, které poskytne závislost rychlostí na řídicích povelích.



Obrázek 5.4: Testovací podvozek

5.2.1. Návrh experimentu

Měření musí být realizovatelné v laboratorních podmínkách s dostupnými přístroji. Je vhodné, aby správnost výsledných dat bylo možné graficky ověřit. Měření by také nemělo být příliš časově náročné a následné zpracování by nemělo zabrat nadměrný čas vlivem vysoké výpočetní složitosti. Pro jednoduchost a dostupnost přístrojů bylo zvoleno následující uspořádání: přijímač rádiového signálu byl nahrazen řídicí deskou Arduino. Zde byl poté generován přesný řídicí signál pro regulátor otáček a servomotor nápravy. Tím je zaručena opakovatelnost. Poloha podvozku byla snímána kamerou. Pro dostupnost a jed-

noduchost instalace a obsluhy byla zvolena akční kamera. Umístěna byla na stropě ve velké vzdálenosti pro zmenšení paralaxy. Pro následnou detekci byl podvozek opatřen rovnou deskou, kde na bílém pozadí byly vytištěny dva černé kruhy, jejichž poloha odpovídala středům náprav.

5.2.2. Průběh měření

Samotné měření probíhalo tak, že byl vyslán povel pro Arduino jednotku, která uvedla auto do pohybu na přesně stanovenou dobu. Kamera celou dobu pořizovala záznam. Data byla ukládána ve formě videa s pevně nastavenou frekvencí 10 snímků za sekundu. Do textového souboru pak byly ručně ukládány informace o povelích pohybu pro pozdější synchronizaci s videozáznamem. Po každé jízdě byl podvozek vrácen na počáteční lokaci a měření se opakovalo.

Celý experiment měl čtyři části. V první části byla měřena závislost úhlové rychlosti pro různý signál pro servomotor, při konstantním signálu pro regulátor. Obvyklý servo signál má délku 1000 až 2000 μs , s ohledem na časovou náročnost bylo pro servomotor zvoleno devět hodnot s konstantním rozestupem. Celé měření probíhalo v šesti segmentech. Jednotlivé segmenty se dále dělily na devět sérií po pěti jízdách, kdy každá pětice odpovídá jednomu předem zvolenému signálu. Celkově tedy pro každý z devíti příkazů existuje 30 měření. Celkový počet byl zvolen pro další statistické zpracování.

Druhá část byla zaměřená na zjištění závislosti rychlosti jízdy na signálu pro regulátor. Signál pro servomotor byl konstantní s hodnotou zjištěnou pro přímočarou jízdu.

Ve třetí části byl naměřen kontrolní segment k první části, ale s jemnějším krokem signálu pro servomotor. Toto měření slouží pro kontrolu výsledků a nebude použito při výpočtu modelu.

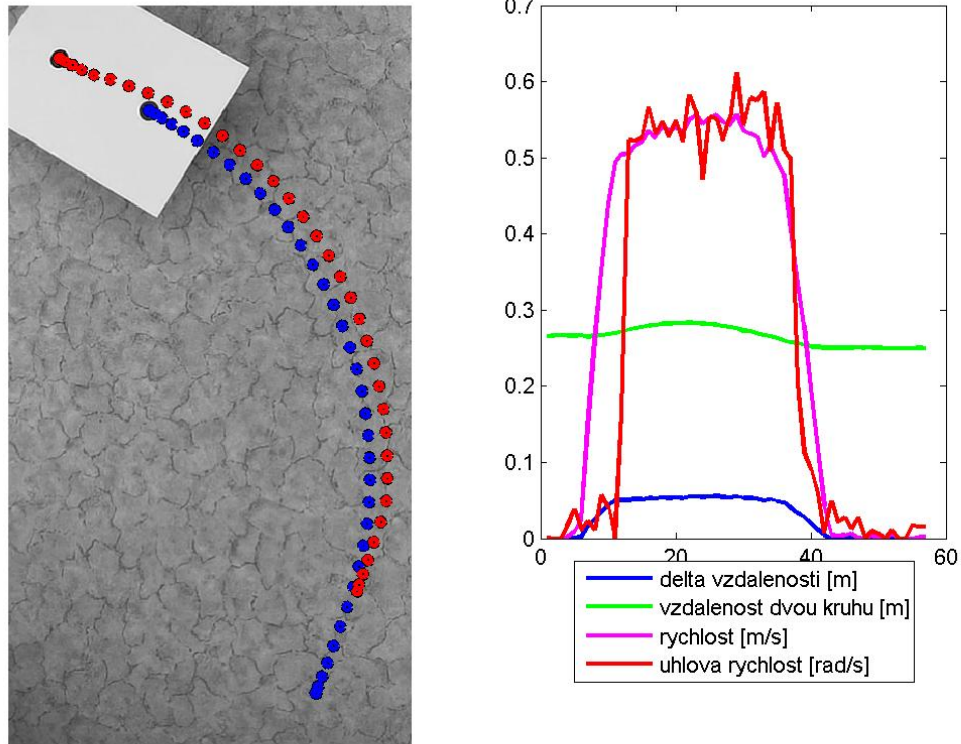
V poslední části byly naměřeny úhly natočení kol v celém rozsahu pohybu servomotoru. Toto měření slouží k porovnání s modelem, který by tyto hodnoty měl potvrdit.

5.3. Zpracování výsledků a zhodnocení

Výstupem z měření byly videosoubory a jeden textový soubor s povely. Videosoubory byly pomocí videoeditoru rozstříhány na jednotlivé jízdy a každému souboru byl přiřazen odpovídající povel pohybu. S ohledem na množství dat byl pro další zpracování využit program Matlab. Vytvořený skript otevřel video soubor, našel polohu dvou kruhů (tedy středů náprav podvozku) a porovnáváním souřadnic dvou po sobě jdoucích snímků určil úhlovou a dopřednou rychlost. Tento způsob byl zvolen pro snížení chyby paralaxou. Rozdíl mezi dvěma po sobě jdoucími snímky je dostatečně velký pro výpočet, ale zároveň dostatečně malý, takže chyba paralaxou mohla být zanedbána. Vzdálenost dvou černých teček je pevně daná a proto je možné kalibrovat velikost pixelu pro jednotlivý snímek.

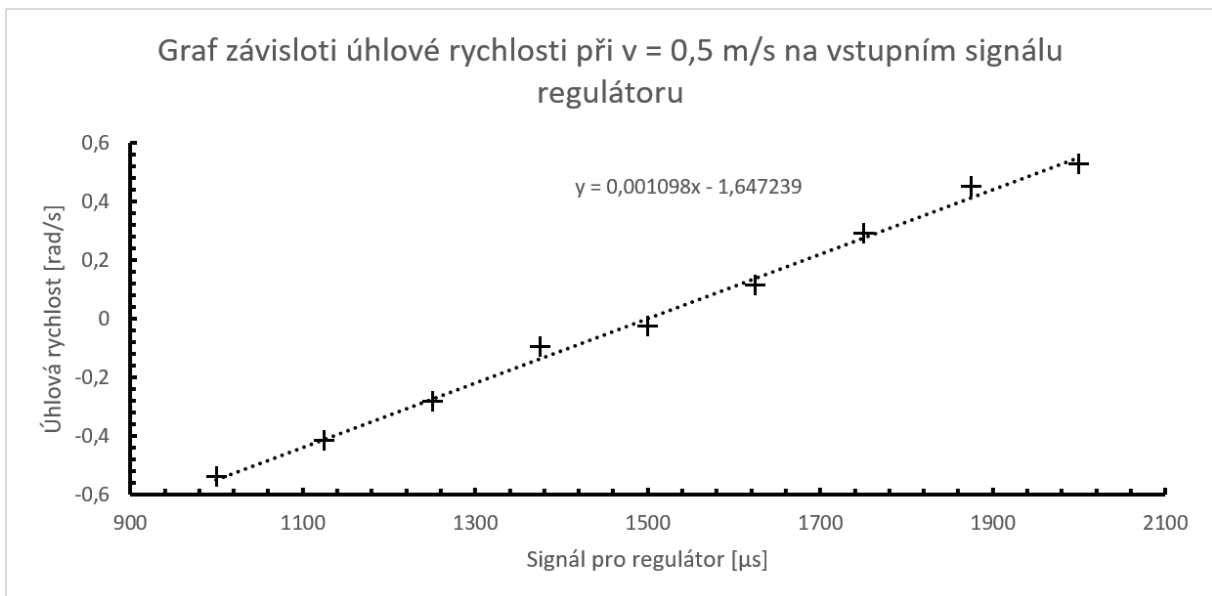
Na určeném úseku pak byly z těchto hodnot určeny průměrné velikosti rychlostí pro danou jízdu. Data pak byla zapsána do textového souboru a byl uložen grafický výstup pro kontrolu výstupu skriptu. Příklad grafického výstupu skriptu lze vidět na obr.5.5

Data z textového souboru pak byla načtena do tabulkového editoru Excel, kde proběhlo další zpracování. Na základě dat z první části experimentu byla počítána závislost úhlové rychlosti (rotace) na signálu pro konstantní rychlost. Na základě průměrné rych-



Obrázek 5.5: Příklad grafického výstupu z Matlab skriptu

losti ze všech měření byla určena referenční rychlost $v_{ref} = 0,5\text{m/s}$. Pro tuto rychlost pak byly přepočteny všechny úhlové rychlosti, aby bylo možné jednotlivá měření porovnat.

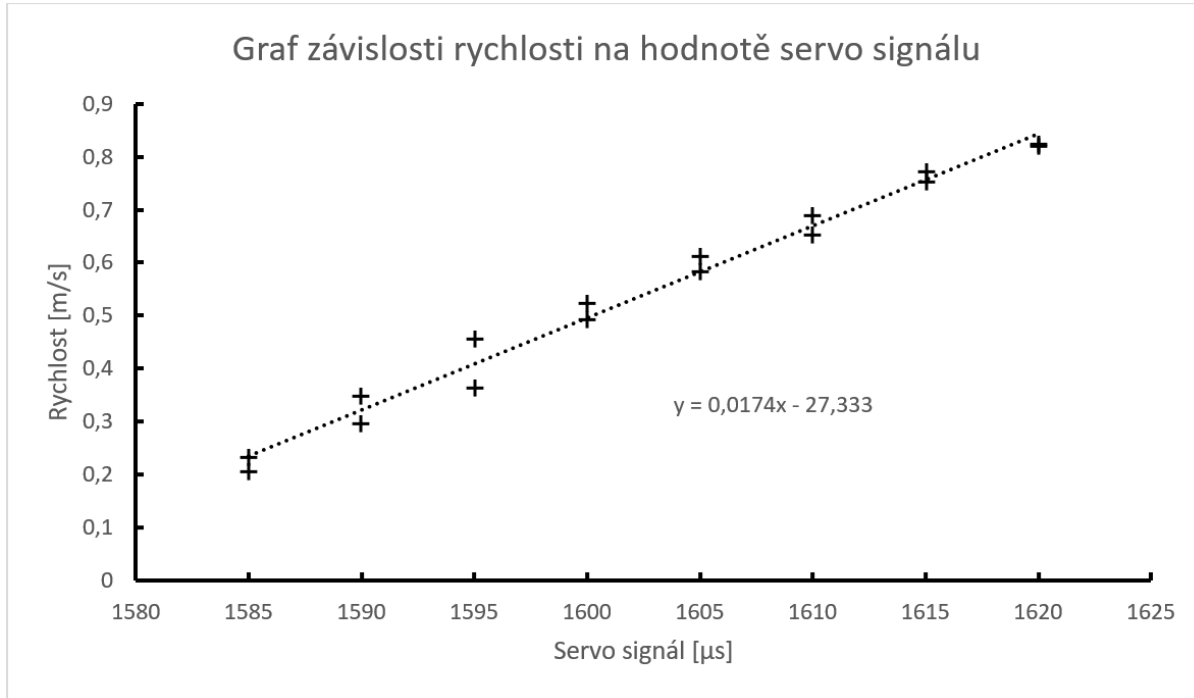


Obrázek 5.6: Graf závislosti úhlové rychlosti na servosignálu

V grafu na obr.5.6 lze vidět závislost průměrné úhlové rychlosti na řídicím povelu. Závislost je přibližně lineární, kladné hodnoty odpovídají rotaci ve směru proti hodinovým ručičkám. Rovnici přímky z grafu pak lze přepsat do tvaru:

$$\omega = 0,001098S_{servo} - 1,64739 \quad (5.1)$$

kde ω je úhlová rychlost, S_{servo} je velikost signálu pro servomotor a rovnice platí pro v_{ref} a pro signál o rozsahu 1000-2000 μs . Střední odchylka se pohybuje kolem $0,6^\circ/s$, nejvýše však $1^\circ/s$. Dalším statistickým zpracováním jednotlivých jízd byly zjištěny konstanty pro pravděpodobnostní model: $\alpha_1=0.0032$; $\alpha_2=0$; $\alpha_3=0.00005766$; $\alpha_4=0.0013761$. Konstanty α_5 a α_6 určeny nebyly.



Obrázek 5.7: Graf závislosti rychlosti na velikosti signálu

Z druhé části je možné zjistit závislost obvodové rychlosti na signálu pro regulátor. Navržené měření a zpracování není vhodné pro určování vysokých rychlostí a proto bylo pro zpracování zvoleno pouze okolí referenčního bodu. Získanou závislost lze vidět na obr.5.7

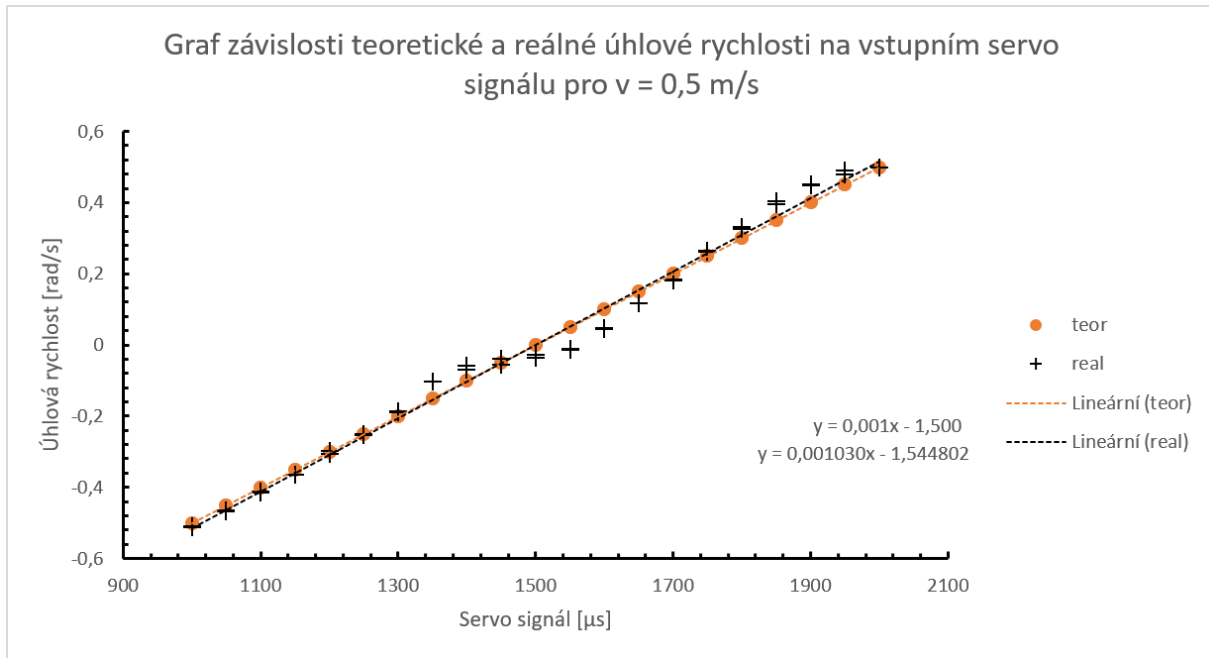
Nejnižší hodnota signálu, kdy se auto samo rozjelo a udrželo stálou rychlost byla 1585 μs , závislost pak lineárně pokračuje až do hodnoty 1620 μs . Rovnici přímky z grafu lze přepsat do tvaru:

$$v = 0,0174S_{reg} - 27,333 \quad (5.2)$$

kde v je rychlost translace a S_{reg} je velikost signálu pro regulátor, rovnice platí pro signál o rozsahu 1585-1620 μs . Ze získané přímky lze zjistit hodnotu signálu 1600 μs pro referenční rychlost 0,5 m/s, což potvrzuje správnou volbu referenční hodnoty použité v první části pro všechna měření.

$$v = R\omega \quad (5.3)$$

Zjištěné závislosti z prvních dvou částí slouží k vytvoření velocity modelu. Rovnice 5.1 platí pouze pro jednu rychlost, při předpokladu pohybu po kružnici s konstantním poloměrem lze použitím vztahu 5.3 vypočítat úhlovou rychlost pro naměřený rozsah rychlostí.



Obrázek 5.8: Graf porovnání naměřených a spočtených hodnot

Výsledné rovnice pak mají tvar:

$$v = 0,0174S_{reg} - 27,333 \quad (5.4)$$

$$\omega_{ob} = \frac{\omega v}{v_{ref}} = \frac{(0,001098S_{servo} - 1,64739)(0,0174S_{reg} - 27,333)}{0,5} \quad (5.5)$$

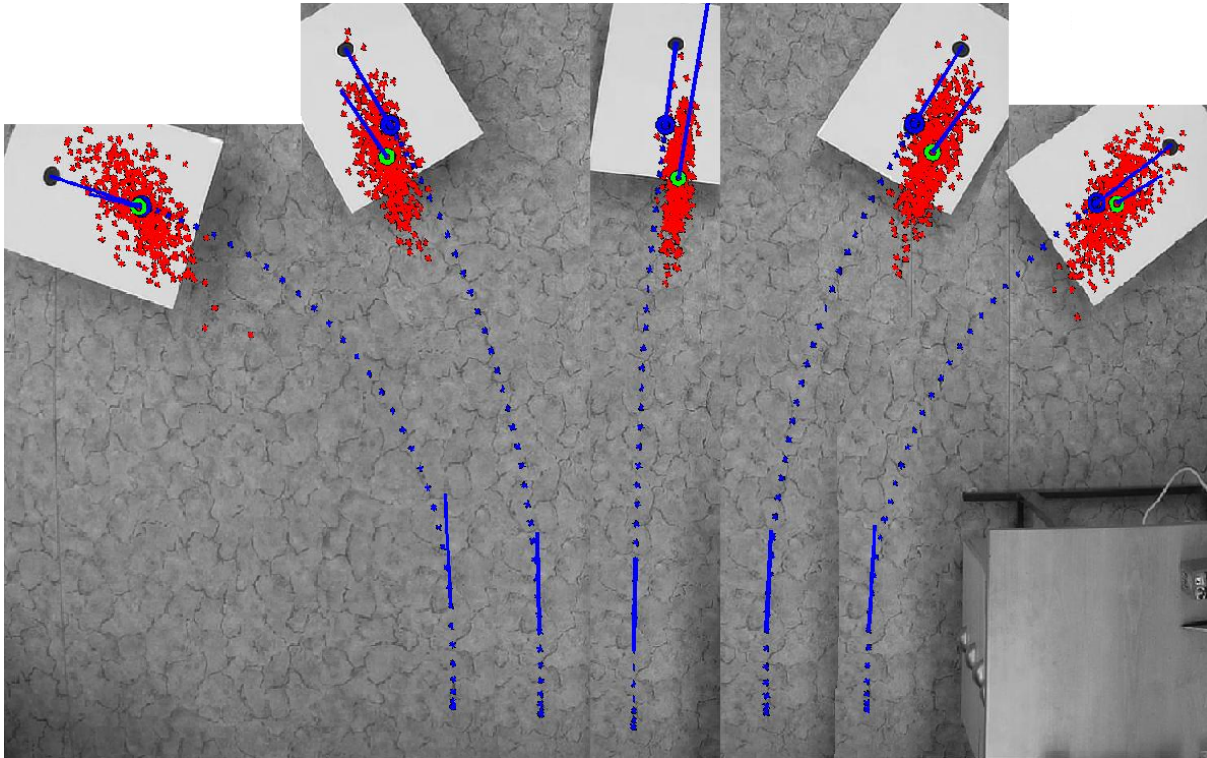
Zde je ω_{ob} obecná úhlová rychlost, platí pro konkrétní rychlost v . Inverzí vstupních a výstupních veličin lze řešit opačný problém, tedy jaký signál použít na základě požadované rotace a translace. Jednoduchým vyjádřením ze vzorců 5.4 a 5.5 lze dojít ke vztahům:

$$S_{reg} = \frac{v + 27,333}{0,0174} \quad (5.6)$$

$$S_{servo} = \frac{0,5\omega - 94,095}{0,0629} \quad (5.7)$$

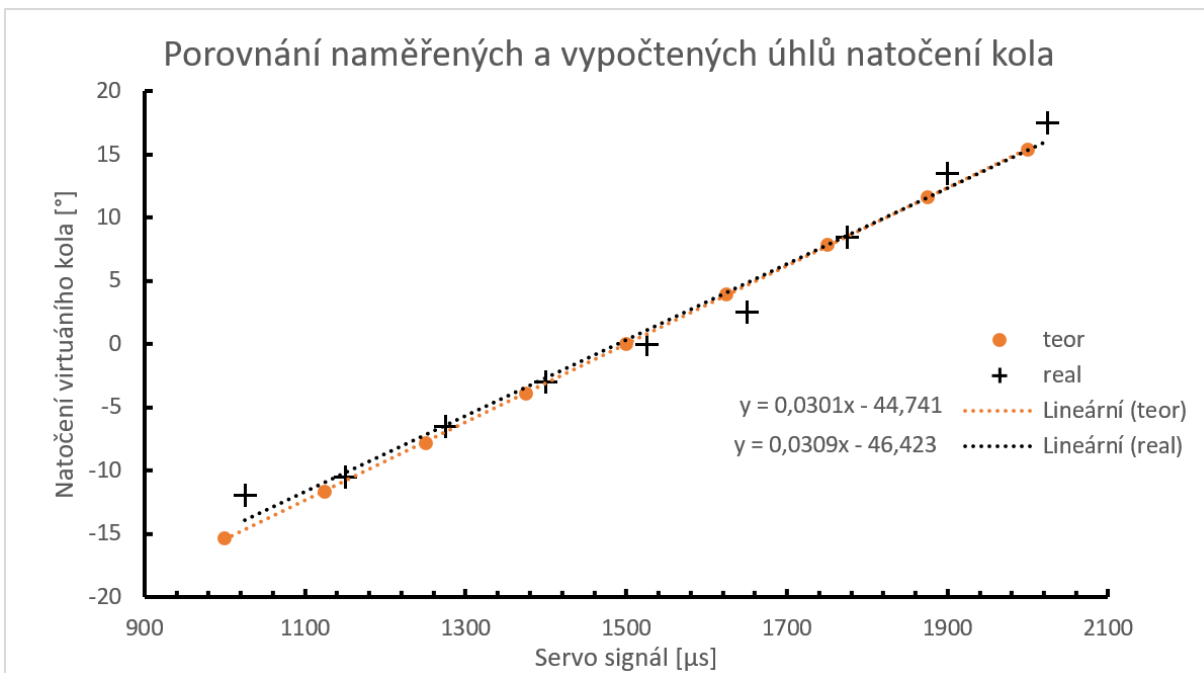
Dosažením vypočtených konstant α do rovnic 3.11 velocity modelu je získán sámplovací model pohybu.

Získaný model byl otestován na kontrolním vzorku dat ze třetí části měření. Nejprve byla provedena kalibrace, kdy jízda byla analyzována algoritmem Matlabu a získané hodnoty rotace a translace byly dosazeny do modelu přesného pohybu po kružnici 3.2. Porovnáním s grafickým výstupem bylo zjištěno, že hodnota rotace pro pohyb po kružnici musí být o 5 % menší, než hodnota zjištěná skriptem, aby se teoretická i reálná poloha bodu překrývala. To je pravděpodobně způsobeno paralaxou obrazu, kdy výpočet kružnice přes celý obraz dá zkreslený výsledek. Pro každou jízdu pak byly do velocity modelu vloženy hodnoty příkazů, vypočtená rotace byla snížena o 5 % a hodnoty byly dosazeny do teoreticky přesného modelu. Takto bylo vygenerováno 500 bodů z pravděpodobnostního modelu. Na obr.5.8 je porovnání vypočtené teoretické rotace a hodnot určených skriptem z Matlabu.



Obrázek 5.9: Ukázka přesnosti vzorkovacího algoritmu

Příklad grafického výstupu je vidět na obr.5.9, kde je ukázán výstup pro různé úhlové rychlosti. Modré body představují reálný pohyb, zelený kruh je spočítaná teoretická poloha a červené body jsou z pravděpodobnostního modelu. Velký modrý kruh, tedy výsledná poloha auta, by měl ležet mezi mrakem červených bodů s pravděpodobností větší než 95%.



Obrázek 5.10: Porovnání naměřených a vypočtených hodnot natočení přední nápravy

Nakonec byl model porovnán s daty ze čtvrté části. Pro hodnoty servo signálu byly na základě geometrie podvozku přepočtem určeny odpovídající úhly natočení kol přední nápravy. Ty byly pak porovnány s naměřenými hodnotami. Porovnání lze vidět na obr. 5.10. Je vidět, že oba průběhy jsou lineární, odchylka směrnice je menší než 3 %, což svědčí o správnosti modelu.

Výsledkem experimentu jsou rovnice 5.6, 5.7 a hodnoty parametrů α . Tyto závislosti je možné použít pro přesné dálkové řízení nebo se dají implementovat do frameworku ROS.

6. Závěr

V kapitole 2 byl představen framework ROS a byly zmíněny jeho základní vlastnosti. Mezi hlavní moduly patří simulátor Stage, sloužící k simulaci virtuálního světa. Modul RVIZ slouží k přehledné vizualizaci. Také byl nastíněn postup tvorby modelů pro tento framework.

V řešeršní části 3 byla popsána široká škála robotů. Zvýšená pozornost byla věnována kolovým robotům, kde byla u některých nastíněna i základní kinematika pohybu. U všech robotů byla diskutována jejich vhodnost pro framework ROS. Představeny byly i pravděpodobnostní modely, navržené v [4].

V praktické části 5 byl vybrán podvozek modelářského auta s ackermannovým řízením. Pomocí kamer byl nasnímán pohyb pro různé hodnoty vstupních signálů. Byl proveden rozbor pohybu a na základě výsledků byly sestaveny rovnice popisující pohyb. Rovnice popisují nejpravděpodobnější předpokládaný pohyb auta, reálně se může pohyb odchýlit o několik centimetrů na metr dráhy. Přesnější simulaci dává pravděpodobnostní model. Pro podvozek byly určeny potřebné konstanty, pomocí kterých lze generovat pravděpodobnostní vzorek pohybu. Model byl otestován na kontrolním vzorku dat.

Získané rovnice mohou být použity pro přesné dálkové řízení vedoucího auta experimentálního semiautonomního konvoje, tedy lze nahradit nepřesný modelářský ovladač. Při použití téměř identických podvozků je možné aplikovat pravděpodobnostní model na sledující auta a tím zvýšit plynulost jízdy. Získané rovnice lze implementovat do frameworku ROS. Měly by být zařazeny za výstup z modulu, který určuje rychlost a směr pohybu jednotlivých aut.

7. Seznam použité literatury

- [1] ROS.org. ROS.org — Powering the world's robots. [online]. 20.5.2017 [cit. 2017-05-20]. Dostupné z: <http://www.ros.org/>.
- [2] HOFFMANN, D. Tvorba simulačních modelů mobilních robotů pro framework ROS. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2017. Vedoucí bakalářské práce doc. Ing. Stanislav Věchet, Ph.D.
- [3] NOVÁK, Petr. Mobilní roboty: pohony, senzory, řízení. Praha: BEN - technická literatura, 2004, 247 s. ISBN 80-7300-141-1
- [4] THRUN Sebastian, Wolfram BURGARD, Dieter FOX. Probabilistic Robotics (Intelligent Robotics and Autonomous Agents series). Intelligent robotics and autonomous agents. The MIT Press, August 2005.
- [5] SIEGWART, Roland, R. Illah NOURBAKHSI. Introduction to autonomous mobile robots. Massachusetts: MIT Press, 2004, 321 s. ISBN 0-262-19502-X
- [6] BARRERA, Alejandra. Advances in Robot Navigation. InTech, 2011, 250 s. ISBN 978-953-307-346-0
- [7] MCKERROW, Phillip John. Introduction to Robotics (Electronic Systems Engineering Series). Addison-Wesley Pub (Sd), 1991, ISBN 13: 9780201182408
- [8] SAJDL, Jan. Ackermannova podmínka. In: autolexicon.net. [online]. [cit. 2017-05-10]. ISSN 1804-2554. Dostupné z: <http://www.autolexicon.net/cs/articles/ackermannova-podminka/>
- [9] CAMPION, G., G. BASTIN a B. DANDREA-NOVEL. Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. Robotics and Automation, IEEE Transactions on [online]. USA: IEEE, 9602, 12(1), 47-62 [cit. 2017-05-13]. DOI: 10.1109/70.481750. ISSN 1042-296X.
- [10] KÁRNÍK, Ladislav, Radek KNOFLÍČEK a Jozef NOVÁK-MARCINČIN. Mobilní roboty. Opava: Márfy Slezsko, 2000, 212 s. : il. ISBN 80-902746-2-5.
- [11] VEX Robotics. Magbot M2 [online], 2009 [cit. 2017-05-7]. Dostupné z: <https://www.flickr.com/photos/vexrobotics/3996439338/>
- [12] Wyatt915. A 6-legged robot from Parallax, Inc. [online]. 13.11.2008 [cit. 2017-05-02]. Dostupné z: https://commons.wikimedia.org/wiki/File:Legged_robot.jpg
- [13] CULLY, Antoine, Jeff CLUNE, Danesh TARAPORE a Jean-Baptiste MOURET. Robots that can adapt like animals. Nature [online]. 2015, 521(7553), 503 [cit. 2017-05-20]. DOI: 10.1038/nature14422.
- [14] JURVETSON, Steve. Robo-snakes [online]. 30.7.2005 [cit. 2017-05-7]. Dostupné z: <https://www.flickr.com/photos/jurvetson/32689486>

- [15] HORČÍK, Jan. Boston Dynamics představili nového robota na kolečkách: umí skákat i nosit bedny. In: hybrid.cz [online]. 1.3.2017 [cit. 2017-05-8]. Dostupné z: <http://www.hybrid.cz/boston-dynamics-predstavili-noveho-robotu-na-koleckach-umi-skakat-i-nosit-bedny>
- [16] ĎAĎO, Stanislav a Marcel KREIDL. Senzory a měřicí obvody. Praha: ČVUT, 1996, 315 s. ISBN 80-01-01500-9
- [17] BROWNING, A. W., A Mathematical Model To Simulate Small Boat Behaviour. Doctorate Thesis (Doctorate). Bournemouth University, 1990.
- [18] WILLIAMS, John. 2016 Interanational RoboBoat Competition 160709-N-PO203-213.jpg [online]. 7.12.2016 [cit. 2017-05-09]. Dostupné z: https://commons.wikimedia.org/wiki/File:2016_Interanational_RoboBoat_Compensation_160709-N-PO203-213.jpg
- [19] KOSTROUN, Tomáš. Mechanika letu [online] 2013 [cit. 2017-05-20]. Dostupné z: http://aa.fd.cvut.cz/wp-content/uploads/2013/06/mechanika_letu.pdf
- [20] Erle Robotics. Erle Plane [online]. [cit. 2017-05-20]. Převzato z: <http://erlerobotics.com/blog/erle-plane/>
- [21] FATAN, M., B.L. SEFIDGARI a A.V. BARENJI. An adaptive neuro PID for controlling the altitude of quadcopter robot. In: 2013 18th International Conference on Methods and Models in Automation and Robotics, MMAR 2013 [online]. 2013, s. 662-665 [cit. 2017-05-05]. ISBN 9781467355063.
- [22] BRABEC, Petr, Filip BARTŮŇEK. Pohyby těles v centrálním gravitačním poli Země - družice a sondy. In: <http://www.fyzika007.cz/> [online]. [cit. 2017-05-01]. Dostupné z: <http://www.fyzika007.cz/gravitacni-pole/pohyby-teles-v-centralnim-gravitacnim-poli-zeme—druzice-a-sondy>
- [23] TAUNER, Stefan. Servo Diagram [online]. 24.12.2013 [cit. 2017-05-06]. Dostupné z: https://commons.wikimedia.org/wiki/File:Servo_Diagram.svg

8. Seznam použitých zkratek a symbolů

Zkratky:

ROS	Robot Operating System
Li-Pol	Lithium-polymer
ICR	Instant centre of rotation (Okamžitý střed otáčení)
SS	Souřadný systém
RC	Radio control

Symboly:

α, β, θ	Úhly [°][rad]
α	Parametr pravděpodobnostního modelu
v	Rychlost [m/s]
ω	Úhlová rychlost (rotace) [°/s][rad/s]
ε	Pravděpodobnostní rozdělení
L,b	Vzdálenosti [m]
x,y	Souřadnice robotu [m]
x,y	Značení os souřadného systému
r	Poloměr [m]
f	Síla [N]
R	Poloměr otáčení [m]
S	Velikost servo signálu [μs]
R	Matice pro přepočítání mezi souřadnými systémy
t	Čas [s]

9. Seznam příloh

1. Příloha 1 Obsah CD

Příloha 1 - Obsah CD

Příložené CD obsahuje elektronickou verzi bakalářské práce, skripty, soubory vytvořené při zpracování a reprezentativní vzorek dat z měření. Struktura adresářů:

- / Kořenový adresář obsahující elektronickou verzi práce a ostatní adresáře
- /matlab Adresář obsahující skripty pro program Matlab
- /data Adresář obsahující reprezentativní vzorek z měření
- /zpracovani Adresář obsahující výstupy ze skriptů a soubory stvořené při zpracování dat