

**Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra kybernetiky**

BAKALÁŘSKÁ PRÁCE

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd

Akademický rok: 2014/2015

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: Ondřej VANÍČEK
Osobní číslo: A13B0265P
Studijní program: B3918 Aplikované vědy a informatika
Studijní obor: Kybernetika a řídicí technika
Název tématu: Návrh řízení pro zřetězenou řadu mobilních robotů
Zadávací katedra: Katedra kybernetiky

Z á s a d y p r o v y p r a c o v á n í :

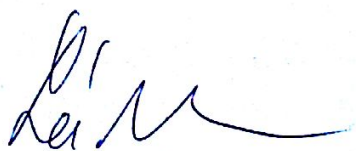
1. Navrhnete matematický model popisující zřetězené řady mobilních robotů.
2. Navrhnete algoritmus pro řízení řady mobilních robotů.
3. Otestujete navržený algoritmus řízení.

Rozsah grafických prací: dle potřeby
Rozsah pracovní zprávy: 30-40 stránek A4
Forma zpracování bakalářské práce: tištěná
Seznam odborné literatury:

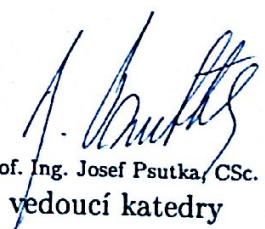
- Klančar, G., Matko, D., & Blažič, S. (2008). Wheeled Mobile Robots Control in a Linear Platoon. *Journal of Intelligent and Robotic Systems*, 54(5), 709731. doi:10.1007/s10846-008-9285-7
- Martinec, D., & Hurák, Z. (2011). Vehicular platooning experiments with LEGO MINDSTORMS NXT. In 2011 IEEE International Conference on Control Applications (CCA) (pp. 927932). IEEE. doi:10.1109/CCA.2011.6044393

Vedoucí bakalářské práce: Ing. Miroslav Flídr, Ph.D.
Katedra kybernetiky

Datum zadání bakalářské práce: 1. listopadu 2014
Termín odevzdání bakalářské práce: 15. května 2015



Doc. RNDr. Miroslav Lávička, Ph.D.
děkan



Prof. Ing. Josef Psutka, CSc.
vedoucí katedry

V Plzni dne 1. listopadu 2014

Prohlášení

Předkládám tímto k posouzení a obhajobě bakalářskou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím odborné literatury a pramenů, jejichž úplný seznam je její součástí.

V Plzni dne _____

podpis

Poděkování

Tímto bych rád poděkoval vedoucímu bakalářské práce, panu Ing. Miroslavu Flídřovi, PhD., za odborné vedení práce, ochotu při poskytování rad i připomínek a mnohé cenné rady, které vedly k vypracování této práce. Dále bych chtěl poděkovat svým rodičům za podporu během celého studia.

Abstrakt

Tato bakalářská práce se zabývá návrhem řízení zřetěžené řady mobilních robotů. V první části práce je provedeno odvození potřebných matematických modelů, k nimž je následně prováděn návrh řídicích algoritmů. Navržené algoritmy jsou posléze simulačně ověřeny při řízení odvozených modelů. Druhá část práce se zabývá návrhem experimentální platformy určené k praktickému ověření navržených algoritmů v laboratorním prostředí včetně volby řídicí jednotky, senzorů a komunikačního rozhraní. Na závěr jsou pak představeny výsledky testovacích experimentů s reálnými roboty.

Klíčová slova

řízení kolon, pružinový model, sledování trajektorie, diferenciálně řízený robot, Arduino, MATLAB, Simulink

Abstract

This bachelor thesis is focused on control design of mobile robot platoon. Necessary mathematical models are derived in the first part of the work and the design of control algorithms is carried out afterwards. Proposed algorithms are then verified by simulation of controlling derived models. The second part of this thesis is focused on design of an experimental platform dedicated to practical verification of proposed algorithms in laboratory environment including selection of control unit, sensors and communication interface. The results of test experiments with real robots are presented at the end of the work.

Keywords

platoon control, mass-spring-damper model, trajectory tracking, differentially driven robot, Arduino, MATLAB, Simulink

Obsah

1	Úvod	1
2	Matematické modelování	2
2.1	Matematický model diferenciálně řízeného robotu	2
2.1.1	Souřadné systémy	3
2.1.2	Kinematické rovnice	3
2.2	Matematický model kolony vozidel	4
3	Návrh řídicího algoritmu	6
3.1	Řízení diferenciálně řízeného robotu	6
3.2	Řízení lineární kolony vozidel	10
3.3	Řízení pohybu členů kolony v rovině	11
3.3.1	Přímková aproximace sledované trajektorie	11
3.3.2	Polynomiální aproximace sledované trajektorie	12
4	Simulační ověření navržených řídicích algoritmů	14
4.1	Pohyb diferenciálně řízeného robotu	14
4.2	Sledování referenční trajektorie	15
4.3	Řízení lineární kolony vozidel	17
4.4	Řízení zřetěžené řady diferenciálně řízených robotů	18
4.4.1	Přímková aproximace sledované trajektorie	18
4.4.2	Polynomiální aproximace sledované trajektorie	19
5	Návrh experimentální platformy	20
5.1	Platforma Arduino	20
5.2	Použité šasi	21
5.3	Podvozek a řízení pohybu	21
5.3.1	Pulzně šířková modulace	23
5.4	Použité senzory	23
5.4.1	Enkodéry	23
5.4.2	Ultrazvukový dálkoměr	25
5.5	Komunikační prostředky	27
5.5.1	Bezdrátové moduly	27
6	Experimentální ověření navržených algoritmů	28
6.1	Regulace rychlosti pohybu vozidla	28
6.2	Řízení lineární kolony vozidel	30
7	Závěr	31

Seznam obrázků

1	Schéma modelovaného robotu	2
2	Schématické znázornění pružinového modelu kolony	5
3	Blokové schéma řízení robotu pro sledování referenční trajektorie	6
4	Výstup funkce <i>ATAN2</i> v závislosti na směru pohybu robotu	7
5	Simulace pohybu robotu pro konstantní vstupy - pohyb v rovině	14
6	Simulace pohybu robotu pro konstantní vstupy - časový vývoj proměnných	14
7	Regulace pohybu po přímce - pohyb v rovině	15
8	Regulace pohybu po přímce - vývoj stavové odchylky	15
9	Regulace pohybu po zakřivené trajektorii	16
10	Regulace pohybu po zakřivené trajektorii - vývoj stavové odchylky	16
11	Lineární kolona - rychlosti členů	17
12	Lineární kolona - relativní vzdálenosti	17
13	Řízení kolony v rovině - přímková aproximace	18
14	Řízení kolony v rovině - přímková aproximace	18
15	Řízení kolony v rovině - polynomiální regrese	19
16	Jednotka Arduino UNO	20
17	Použité šasi - <i>Magician chassis</i>	21
18	Použitý motor s enkodérem	22
19	Schéma řadiče Pololu DRV8835	22
20	Časový diagram PWM modulace	23
21	Magnetický enkodér	24
22	Výstup magnetických enkodérů	24
23	Ultrazvukový dálkoměr Devantech SRF05	26
24	Diagram použití ultrazvukového dálkoměru	26
25	Bezdrátový modul nRF24L01	27
26	Regulace rychlosti otáčení kol robotu	28
27	Regulace rychlosti pohybu robotu	29
28	Řízení lineární kolony	30

1 Úvod

V posledních desetiletích jsme svědky bouřlivého rozvoje v mnoha průmyslových odvětvích, který má přímý vliv na život každého člena dnešní společnosti. Velmi značný je například vývoj v oblasti automobilového průmyslu, který vedl ke zvýšení dostupnosti automobilů a tím pádem i zvýšení jejich počtu. Dle údajů Českého statistického úřadu a Svazu automobilového průmyslu [1] vzrostl mezi lety 1989 a 2014 počet registrovaných osobních vozidel v České republice více než dvojnásobně na téměř 5 mil. v současné době provozovaných vozidel. Ještě větší nárůst v témže období zaznamenala kategorie nákladních vozidel, kde došlo k více než 2,5-násobnému nárůstu na současný počet bezmála 700 tis. vozidel.

Takto výrazný nárůst počtu provozovaných vozidel samozřejmě znamená zvýšení nároků na kapacitu silniční sítě, jejíž rozvoj však probíhá výrazně pomaleji a díky tomu jsou řidiči (*nejen*) na českých komunikacích čím dál častěji svědky dopravních komplikací, které jsou z velké části způsobeny přetížením dopravních tepen. Řešením tohoto problému, kterému je v současné době věnována značná pozornost, je přechod k automaticky řízeným vozidlům. To by mělo přinést eliminaci chyby lidského faktoru, jako je nepozornost, dlouhá reakční doba, volba nepřiměřeně vysoké či naopak nízké jízdní rychlosti a podobně.

Cílem této práce je navrhnout a otestovat algoritmus řízení zřetěžené řady mobilních robotů, tedy laboratorního modelu plynule jedoucí kolony vozidel, která si mezi sebou udržují bezpečnou vzdálenost a pohybují se stejnou rychlostí po stejné trajektorii. Takovéto formace by v praxi přinesly nejen zvýšení bezpečnosti a plynulosti provozu na pozemních komunikacích, ale i úsporu paliva z důvodu aerodynamických vlivů a eliminaci nákladů na likvidaci škod po nehodách.

První část této práce je věnována určení matematických modelů systémů, které jsou předmětem této práce. V první řadě se jedná o model testovacího vozidla, tedy diferenciálně řízeného robotu, a dále pak o tzv. pružinový model kolony vozidel, který představuje požadované chování formace vozidel, respektive mobilních robotů. Další kapitola je pak věnována návrhu řídicích algoritmů pro získané matematické modely, které jsou zaměřeny jak na řízení samotného robotu, tak i na řízení pohybu skupiny robotů tvořících zřetěženou řadu, nebo-li kolonu. Navržené řídicí algoritmy jsou následně simulačně ověřeny pro několik různých požadavků na chování jednotlivých částí i celého systému. Další část této práce se zabývá návrhem experimentální platformy, tedy diferenciálně řízeného robotu osazeného řídicí jednotkou, všemi potřebnými čidly, akčními prvky i komunikačním rozhraním. V poslední kapitole je pak prezentováno výsledné chování reálných robotů i jejich kolony při pohybu v laboratorním prostředí při experimentech.

2 Matematické modelování

Prvním krokem návrhu řízení je nalezení matematického modelu řízeného systému. V této kapitole budou odvozeny rovnice popisující jednak pohyb jednotlivých vozidel ve zřetězené řadě, nebo-li koloně, kterými byli pro účely této práce zvoleni diferenciálně řízení roboti, a dále bude uveden model ideálního chování kolony, tzv. pružinový model.

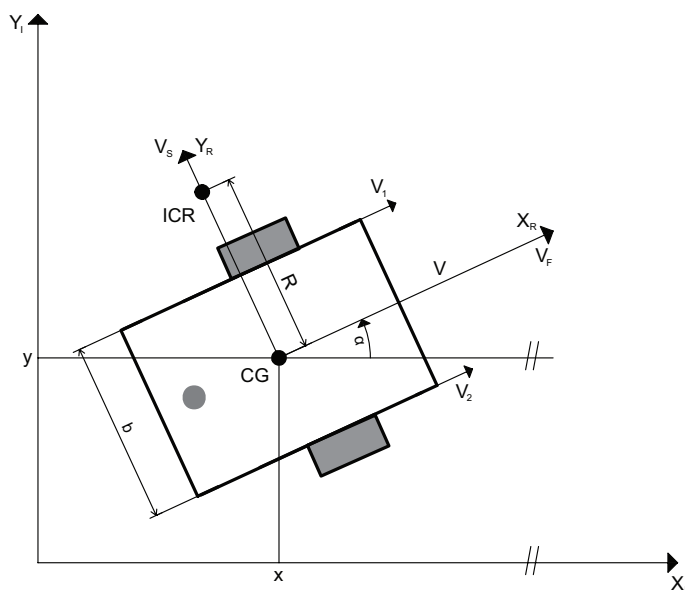
2.1 Matematický model diferenciálně řízeného robotu

Jako cílová platforma pro testování navržených algoritmů bylo zvoleno robotické vozidlo s diferenciálně řízeným podvozkem. Základní principem tohoto řízení je nezávislé ovládání rychlosti otáčení jednotlivých kol (resp. pásů atp.) na každé straně vozidla. Samotné řízení vozidla je pak výrazně flexibilnější, než běžně známé řízení pohybu natáčením kol na jedné z náprav (tzv. Ackermannův podvozek, který známe například z osobních automobilů), není totiž limitován maximálním úhlem natočení kol a umožňuje tak otáčení prakticky "na místě". V neposlední řadě je také tato technika méně náročná na hardware.

V praxi se pro svou flexibilitu a v případě pásové varianty podvozku i velkou prostupnost terénem tento druh podvozku používá zejména pro vojenské účely, ale uplatnění nachází i v civilní sféře. Je hojně využíván ve stavebnictví, zemědělství a v neposlední řadě i na vozidlech zajišťujících úpravu tratí pro zimní sporty.

Pro samotný matematický popis pohybu takto řízeného robotu provedeme několik zjednodušení. Jednak budeme uvažovat, že se robot bude pohybovat výlučně po rovině a že jeho kola budou v neustálém kontaktu s podložkou. Dále pak budou zanedbány třecí síly kol a další pasivní síly působící proti pohybu robotu. Odvození uvedeného matematického modelu bylo použito z materiálu uvedeného v seznamu použité literatury pod číslem [2].

Popisovaným systémem je tedy robot, k jehož šasi jsou v přední části připevněna dvě nezávisle poháněná kola a jehož stabilitu zajišťuje opěrná konstrukce v zadní části. Místo dotyku opěrné konstrukce s podloužkou zprostředkovává ocelová kulička, která se volně otáčí v pouzdře a minimalizuje tak tření mezi opěrným bodem robotu a podložkou. Ilustrační schéma je znázorněno na Obrázku 1.



Obrázek 1: Schéma modelovaného robotu

2.1.1 Souřadné systémy

Pohyb robotu lze tedy mimo jiné reprezentovat pomocí polohy jeho těžiště v osách x , y a úhlu natočení α . V inerciální soustavě tedy polohu robotu popisuje vektor ξ_I , který je tvořen výše uvedenými složkami.

$$\xi_I = \begin{bmatrix} x \\ y \\ \alpha \end{bmatrix} \quad (2.1)$$

Pro zjištění rychlosti v jednotlivých osách resp. rychlosti otáčení robotu kolem své osy je zapotřebí složky tohoto vektoru zderivovat podle času.

$$\dot{\xi}_I = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\alpha} \end{bmatrix} \quad (2.2)$$

Stav samotného robotu budeme však reprezentovat v souřadném systému, který označíme písmenem R . Rychlost pohybu robotu bude vždy ve směru os tohoto systému a její složky \dot{x}_R a \dot{y}_R budou odpovídat rychlost v_f a v_s , zatímco jako ω označíme úhlovou rychlost otáčení kolem bodu ICR. Vektor ξ_R pak bude vypadat následovně (2.3)

$$\dot{\xi}_R = \begin{bmatrix} \dot{x}_R \\ \dot{y}_R \\ \omega \end{bmatrix} \quad (2.3)$$

Vztah obou souřadných systémů je pak dán rotační maticí, která pro převod ze souřadného systému R do souřadného systému I bude označena jako R_{RI} (2.4)

$$R_{RI} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

Dodejme, že tato matice slouží pouze pro převádění mezi souřadnými systémy v oblasti rychlosti, což ovšem pro naše účely postačuje. Převod s pomocí rotační matice bude tedy prováděn podle rovnice (2.5).

$$\dot{\xi}_I = R_{RI} \cdot \dot{\xi}_R \quad (2.5)$$

2.1.2 Kinematické rovnice

Vstupními veličinami hledaného modelu tedy budou úhlové rychlosti otáčení kol $\dot{\phi}_1$ a $\dot{\phi}_2$ na obou stranách robotu. Výstupní veličinou pak bude poloha těžiště robotu v interciální soustavě (souřadnicích $[x,y]$) a úhel natočení robotu α od směru osy x , viz Obrázek 1.

Tečné rychlosti otáčení jednotlivých kol jsou pak zřejmě dány vztahy

$$\begin{aligned} v_1 &= r \cdot \dot{\phi}_1 \\ v_2 &= r \cdot \dot{\phi}_2 \end{aligned} \quad (2.6)$$

a při splnění předpokladů, že se robot může pohybovat jen dopředu nebo dozadu a smýkavý pohyb do boku není umožněn, lze říct, že rychlost v_s bude nulová a v_f pak bude rovna rychlosti (2.7).

$$v = \frac{v_1 + v_2}{2} \quad (2.7)$$

Za situace, kdy se kola na každé straně otáčejí různou a zároveň konstantní rychlostí, se robot pohybuje kolem středu otáčení (ICR). Pro tuto vlastnost platí rovnice (2.8)

$$\begin{aligned} v &= R \cdot \omega \\ v_1 &= \left(R + \frac{b}{2}\right) \cdot \omega \\ v_2 &= \left(R - \frac{b}{2}\right) \cdot \omega \end{aligned} \quad (2.8)$$

kde v je rychlost pohybu těžiště robotu, R je poloměr otáčení a ω úhlová rychlost otáčení těžiště robotu kolem středu ICR a b značí rozchod kol robotu.

Nyní tedy můžeme z rovnic (2.8) a (2.6) vyjádřit rychlost pohybu robotu v těžišti v a úhlovou rychlost otáčení těžiště robotu kolem ICR ω v závislosti na úhlových rychlostech otáčení kol $\dot{\phi}_1$ a $\dot{\phi}_2$, viz. rovnice (2.9)

$$\begin{aligned} v &= \frac{(\dot{\phi}_1 + \dot{\phi}_2)}{2} \\ \omega &= \frac{(\dot{\phi}_1 - \dot{\phi}_2)}{2} \end{aligned} \quad (2.9)$$

Tyto rychlosti pak můžeme zapsat do vektoru (2.10), který představuje rychlost robotu v 'jeho' souřadném systému.

$$\dot{\xi}_R = \begin{bmatrix} v \\ 0 \\ \omega \end{bmatrix} \quad (2.10)$$

Vztah nutný pro popis pohybu robotu v inerciální soustavě I je dán rovnicí (2.5).

$$\dot{\xi}_I = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v \\ 0 \\ \omega \end{bmatrix} = \begin{bmatrix} v \cdot \cos\alpha \\ v \cdot \sin\alpha \\ \omega \end{bmatrix} \quad (2.11)$$

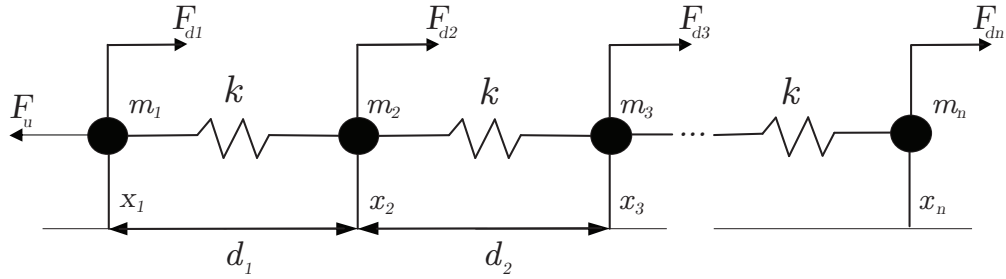
Diferenciální rovnice (2.11) tak popisuje časový vývoj stavu (polohy) odvozeného modelu v inerciální soustavě I prostřednictvím souřadnic a úhlu natočení. S použitím vztahů (2.6) pak získáváme stavové rovnice (2.12).

$$\begin{aligned} \dot{x} &= r \cdot \frac{\dot{\phi}_1 + \dot{\phi}_2}{2} \cdot \cos\alpha \\ \dot{y} &= r \cdot \frac{\dot{\phi}_1 + \dot{\phi}_2}{2} \cdot \sin\alpha \\ \dot{\alpha} &= r \cdot \frac{\dot{\phi}_1 - \dot{\phi}_2}{b} \end{aligned} \quad (2.12)$$

2.2 Matematický model kolony vozidel

V odborné literatuře se můžeme setkat s několika různými způsoby, jak modelovat a řídit lineární kolony vozidel. Tím nejčastějším i nejzákladnějším ale bezesporu je tzv. pružinový model [4]. V této kapitole bude představeno odvození modelu.

Jak je patrné z názvu, tento přístup je inspirován jednou z nejzákladnějších mechanických interakcí, a sice pružinovou silou. Účelem je nalézt takové řízení jednotlivých členů zřetězené řady vozidel, aby se svým chováním co nejvíce blížili níže znázorněné soustavě hmotných bodů, které jsou pospojovány pomocí pružin a tlumičů.



Obrázek 2: Schématické znázornění pružinového modelu kolony

Rozeberme si nyní tento model. Na jednotlivé členy řady, v této variantě tedy hmotné body, působí následující síly (2.13)

$$\begin{aligned}
 F_1 &= m_1 \ddot{x}_1 = F_u - F_{d1} = F_u - b\dot{x}_1 \\
 F_2 &= m_2 \ddot{x}_2 = F_{k1} - F_{d2} = k(d_1 - d_0) - b\dot{x}_2 \\
 F_3 &= m_3 \ddot{x}_3 = F_{k2} - F_{d3} = k(d_2 - d_0) - b\dot{x}_3 \\
 &\vdots \\
 F_n &= m_n \ddot{x}_n = F_{k_{n-1}} - F_{dn} = k(d_{n-1} - d_0) - b\dot{x}_n
 \end{aligned} \tag{2.13}$$

kde d_0 představuje klidovou délku pružiny, respektive požadovanou vzdálenost mezi členy zřetězené řady, vzdálenosti $d_i = x_{i+1} - x_i$ a koeficient b lze chápat jako součinitel odporu prostředí.

Při aplikaci této strategie na reálnou kolonu je však nutné zajistit přítomnost "virtuální pružiny" prostřednictvím řízení jednotlivých vozidel, neboť v takovém případě pro každé vozidlo platí pohybová rovnice (2.14),

$$m_i \ddot{x}_i + b_i \dot{x}_i = u_i \tag{2.14}$$

kde u_i představuje řízení i -tého vozidla. Návrh takového řízení bude diskutován v kapitole (3.2).

V rámci této práce bude představeno i řízení zřetězené řady vozidel pohybujících se v rovině, avšak pro tento případ není nutné odvozovat žádný speciální model, neboť se jedná pouze o rozšíření modelu (2.13) ze skalárních veličin na vektorové, viz. (2.15)

$$\vec{F}_n = m_n \ddot{\vec{x}}_n = \vec{F}_{k_{n-1}} - \vec{F}_{dn} = k(\vec{d}_{n-1} - \vec{d}_0) - b\dot{\vec{x}}_n \tag{2.15}$$

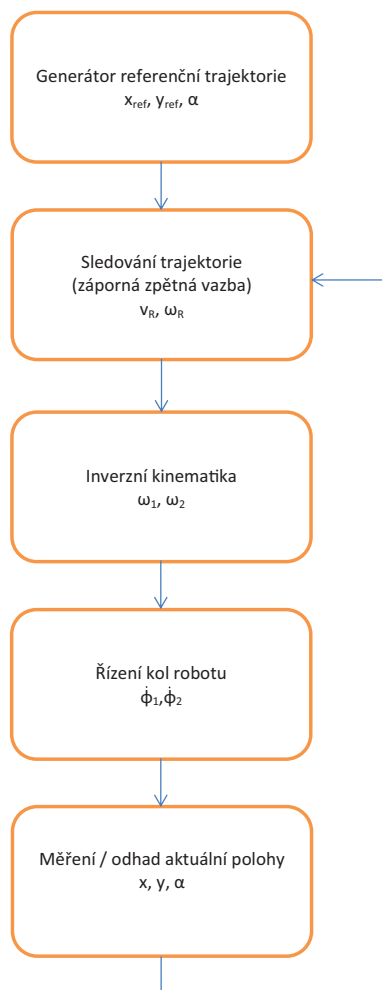
3 Návrh řídicího algoritmu

Dalším logickým krokem je návrh řídicího algoritmu pro odvozené matematické modely. V této kapitole bude nejprve diskutováno zpětnovazební řízení pohybu diferenciálně řízeného robotu v rovině. Dále pak bude představena regulace lineární kolony vozidel, tj. pružinového modelu z kapitoly (2.2) a nakonec bude navrženo řízení pohybu vozidel tvořících zřetězenou řadu v rovině.

3.1 Řízení diferenciálně řízeného robotu

Obecně můžeme problém řízení vozidla, které se pohybuje v prostoru, převést na problém sledování referenční trajektorie, která může být určena nejrůznějšími způsoby. Jako příklady uveďme trajektorii letadla určenou GPS souřadnicemi a nadmořskou výškou nebo trasu osobního automobilu určenou jízdním pruhem na vozovce (a případně údaji z GPS nebo jiného pozičního systému).

Vzhledem k tomu, že cílová platforma, tedy robotické vozidlo laboratorních rozměrů řízené uvnitř místnosti, nemůže ze své podstaty a zamýšleným místem provozu využívat údaje z měření pomocí systému typu GPS, bylo zapotřebí využít inerciální souřadný systém pevně spojený s podložkou (resp. podlahou laboratoře). Referenční trajektorie bude tedy posloupností bodů v prostoru $[x, y]$, která bude v simulačním i experimentálním oddílu generována vektorovou funkcí času. Na následujícím obrázku (3) je znázorněno blokové schéma takového řídicího systému.

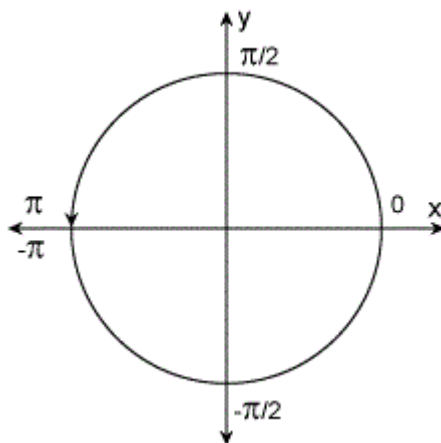


Obrázek 3: Blokové schéma řízení robotu pro sledování referenční trajektorie

Pro sledování referenční trajektorie je tedy nutné znát kromě jejích vlastních parametrů i aktuální stav (polohu $[x, y]$ a úhel natočení α) robotu. Tyto informace jsou v simulačním modelu samozřejmě jednoduše dostupné, avšak na reálném systému je k získání odhadu stavu nutné použít mimo jiné i data z enkodérů umístěných na obou motorech, gyroskopu a akcelerometru. Rozeberme si nyní jednotlivé bloky schématu z Obrázku 3

Prvním blokem je generátor referenční trajektorie. Ta bude pro simulační i experimentální účely, jak již bylo zmíněno, generována vektorovou funkcí času. Výstupem této funkce tedy je v každém časovém okamžiku (resp. kroku simulace) bod popsany souřadnicemi $[x, y]$ v inerciální soustavě. Pro úplnost uveďme, že k získání třetího potřebného parametru, tedy úhlu natočení robotu, je využita čtyřkvadrantová varianta inverzní goniometrické funkce tangens (\tan^{-1}), která je v MATLABu resp. Simulinku označována jako $ATAN2(Y, X)$ a jejíž výstup je určen na základě dvou parametrů Y a X , což jsou v tomto případě tečné rychlosti ve směru osy y resp. x viz. rovnice (3.1). Výstup této funkce v závislosti na orientaci robotu pak ilustruje Obrázek 4.

$$\alpha_{ref} = ATAN2(\dot{y}_{ref}, \dot{x}_{ref}) \quad (3.1)$$



Obrázek 4: Výstup funkce $ATAN2$ v závislosti na směru pohybu robotu

Druhý blok schématu je označen jako sledování referenční trajektorie. Tento úkon lze provádět mnoha způsoby, avšak pro účely této práce bylo zvoleno řízení pomocí stavového regulátoru. Nyní bude představen zákon řízení [3].

Regulační odchylka stavového regulátoru je stanovena podle rovnice (3.2)

$$e = \begin{bmatrix} e_x \\ e_y \\ e_\alpha \end{bmatrix} = \begin{bmatrix} x_{ref} - x \\ y_{ref} - y \\ \alpha_{ref} - \alpha \end{bmatrix} \quad (3.2)$$

Z důvodu využití dvou souřadných systémů (viz. kapitola 2.1.1) je nyní nutné převést tuto odchylku z inerciálního souřadného systému I do souřadného systému robotu R . K tomu se použije zpětná rotační matice, která má tvar (3.3)

$$R_{IR} = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

Používaná transformovaná odchylka tedy má tvar

$$e_t = \begin{bmatrix} e_{xt} \\ e_{yt} \\ e_{\alpha t} \end{bmatrix} = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} e_x \\ e_y \\ e_\alpha \end{bmatrix} \quad (3.4)$$

Pro výpočet zpětnovazebního řízení je dále zapotřebí určit referenční rychlosti pohybu robota. Tyto rychlosti odpovídají rovnicím (3.5) a (3.6).

$$v_{ref} = \sqrt{\dot{x}_{ref}^2 + \dot{y}_{ref}^2} \quad (3.5)$$

$$\omega_{ref} = \frac{\dot{y}_{ref} \cdot \dot{x}_{ref} - \ddot{x}_{ref} \cdot \dot{y}_{ref}}{\dot{x}_{ref}^2 + \dot{y}_{ref}^2} \quad (3.6)$$

Aplikací lineární transformace získáme vztahy (3.7) a (3.8) pro řídicí rychlosti v_r a ω_r

$$v_r = v_{ref} \cdot \cos(e_{\alpha t}) - u_1 \quad (3.7)$$

$$\omega_r = \omega_{ref} - u_2 \quad (3.8)$$

kde u_1 a u_2 jsou vstupy dynamického modelu transformované odchylky ve tvaru (3.9)

$$\dot{e}_t = \begin{bmatrix} 0 & \omega_r & 0 \\ -\omega_r & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot e_t + \begin{bmatrix} 0 \\ \sin(e_{\alpha t}) \\ 0 \end{bmatrix} \cdot v_r + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (3.9)$$

Nyní, po linearizaci rovnice (3.9) v okolí referenční trajektorie získáváme vztah (3.10) pro zpětnovazební řízení v závislosti na transformované odchylce e_t a rychlosti v_{ref}

$$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} -k_1 \cdot e_{xt} \\ -k_2 \cdot \text{sign}(v_{ref}) \cdot e_{yt} - k_3 \cdot e_{\alpha t} \end{bmatrix} \quad (3.10)$$

Ve vztahu (3.10) se vyskytují dosud nezmiňované konstanty $k_1 - k_3$, které je nyní potřeba určit. Určení těchto koeficientů by mohlo být provedeno metodou pokus-omyl, ale sofistikovanějším řešením bude jejich určení z požadovaného tvaru charakteristického polynomu uzavřené regulační smyčky. Jeho požadovaný tvar tedy definujeme takto

$$(p + 2\xi\omega_n) \cdot (p^2 + 2\xi\omega_n p + \omega_n^2), \quad (3.11)$$

kde parametr ξ , představující tlumení, je číslo z intervalu $(0, 1)$ a parametr ω_n , představující přirozenou frekvenci, je libovolné číslo $\omega_n > 0$. Uzavřená regulační smyčka tedy bude mít reálný pól v $p_1 = -2\xi\omega$ a dva komplexně sdružené póly v $p_{2,3} = -\xi\omega_n \pm i\omega_n\sqrt{\xi^2 - 1}$.

Na základě uvedených skutečností zvolíme koeficienty $k_1 - k_3$ následujícím způsobem

$$k_1 = k_3 = 2\xi\omega_n \quad (3.12)$$

$$k_2 = \frac{\omega_n^2 - \omega_{ref}^2}{|v_{ref}|} \quad (3.13)$$

V tomto případě ovšem nesmíme zapomenout na skutečnost, že koeficient k_2 by v případě snižování rychlosti v_{ref} rostl nade všechny meze (tj. generované řízení u by se blížilo k nekonečnu). Tento zádrhel je vyřešen položením $\omega_n = \omega_n(t) = \sqrt{\omega_{ref}^2(t) + bv_{ref}^2(t)}$, z čehož vyplývá

$$k_1 = k_3 = 2\xi\sqrt{\omega_{ref}^2(t) + bv_{ref}^2(t)} \quad (3.14)$$

$$k_2 = b \cdot |v_{ref}| \quad (3.15)$$

kde člen b představuje dodatečný stupeň volnosti.

Kombinací vztahu (3.7), (3.8) a (3.10) tak získáváme výsledný zákon řízení

$$\begin{bmatrix} v_r \\ \omega_r \end{bmatrix} = \begin{bmatrix} v_{ref} \cos(\alpha_{ref} - \alpha) + k_1 [\cos(\alpha) \cdot (x_{ref} - x) + \sin(\alpha) \cdot (y_{ref} - y)] \\ \omega_{ref} + k_2 \text{sign}(v_{ref}) [\cos(\alpha) \cdot (x_{ref} - x) - \sin(\alpha) \cdot (y_{ref} - y)] + k_3 (\alpha_{ref} - \alpha) \end{bmatrix}$$

Třetí blok v řídicím schématu (Obrázek 3) je nazván *Inverzní kinematika*, a sice z toho důvodu, že se jedná o opačný výpočet, než ve vztazích (2.6) a (2.8). Z rychlostí v_r a ω_r , vypočtených na základě zpětné vazby, je zapotřebí vypočítat rychlosti otáčení obou kol ω_1 a ω_2 .

Dosazením vztahu (2.6) do rovnice (2.8) a úpravou získáváme potřebné vztahy pro *požadované* rychlosti jednotlivých kol.

$$\omega_1 = \frac{v_r + \frac{b}{2} \cdot \omega_r}{r} \quad (3.16)$$

$$\omega_2 = \frac{v_r - \frac{b}{2} \cdot \omega_r}{r} \quad (3.17)$$

Požadované úhlové rychlosti otáčení jednotlivých kol jsou tedy závislé na vypočtených hodnotách v_r a ω_r , které jsou časově proměnné, a na konstantách b a r , které představují rozchod a poloměr kol řízeného robotu.

Předposledním blokem řídicího schématu je regulátor otáček motoru. Úkolem tohoto subsystému je zajištění rychlé a přesné změny otáček akčních členů. Na tento problém lze aplikovat mnoho přístupů, z nichž každý bude mít své výhody, ale i nevýhody. Pro účely této práce jsem se omezil na návrh zpětnovazebního PD regulátoru s jedním stupněm volnosti pro stejnosměrný elektrický motor s obecným přenosem (3.18).

$$F_{u \rightarrow \omega}(p) = \frac{K}{(J \cdot L)p^2 + (L \cdot b + R \cdot J)p + (K^2 + b \cdot R)} \quad (3.18)$$

V této přenosové funkci se vyskytuje několik fyzikálních parametrů konkrétního motoru, které zpravidla nejsou známy, a proto je nutné před samotným návrhem regulátoru provést identifikaci systému s předpokládanou strukturou (3.19).

$$F_{u \rightarrow \omega}(p) = \frac{b_0}{a_2 p^2 + a_1 p + a_0} \quad (3.19)$$

K identifikovanému systému je pak možné navrhovat regulátor. Jak již bylo zmíněno, pro účely této práce byl zvolen zpětnovazební PD regulátor s jedním stupněm volnosti. Hlavní předností tohoto přístupu, kromě jeho jednoduchosti, je nepřítomnost tzv. *wind-up* efektu, který působí nepříznivě v jednoduchých regulačních smyčkách s PID regulátorem. Přenos uvažovaného PD regulátoru s použitím konstanty zesílení K a časových konstant T_D a τ , které slouží k umístění nuly a pólu je následující

$$F_{PD}(p) = K \frac{(T_D + \tau)p + 1}{\tau p + 1} \quad (3.20)$$

a zkráceně lze zapsat také jako (3.21)

$$F_{PD}(p) = \frac{d_1 p + d_0}{p + c_0} \quad (3.21)$$

Koeficienty $d_{1,0}$ a c_0 jsou tedy návrhovými parametry regulátoru, které lze volit. Jejich volba závisí na požadavcích na chování systému v uzavřené regulační smyčce a není

zdaleka jednoznačná, viz. materiál [6], podle něhož byly rovněž určeny přenosové funkce v tomto odstavci. K návržení parametrů regulátoru otáčení kol robotů, které jsou předmětem této práce, byl použit nástroj PID tuning, který je součástí softwaru MATLAB od společnosti MathWorks, Inc.

Posledním blokem schématu na (Obr. 3) je měření, resp. výpočet aktuální polohy robotu. Tento úkon je prováděn časovou integrací rychlostí ze vztahů (2.12).

$$\begin{aligned}x(t) &= \int_0^t r \cdot \frac{\dot{\phi}_1(t) + \dot{\phi}_2(t)}{2} \cdot \cos\alpha(t) dt \\y(t) &= \int_0^t r \cdot \frac{\dot{\phi}_1(t) + \dot{\phi}_2(t)}{2} \cdot \sin\alpha(t) dt \\ \alpha(t) &= \int_0^t r \cdot \frac{\dot{\phi}_1(t) - \dot{\phi}_2(t)}{b} dt\end{aligned}\tag{3.22}$$

Tento způsob určování polohy můžeme aplikovat, protože je v souladu s předpoklady z kapitoly o matematickém modelování (2.1), tedy že kola robotu jsou neustále v kontaktu se zemí a nedochází ke smýkavému pohybu robotu do stran ani k prokluzu kol.

Po zjištění aktuálních souřadnic robotu v inerciální soustavě je tedy možné uzavřít zpětnou vazbu, dosadit do rovnice (3.2) a řídit pohyb robotu na základě regulační odchylky e v bloku *Sledování trajektorie*.

3.2 Řízení lineární kolony vozidel

Účelem této sekce je návržení takového řídicího algoritmu, který zajistí požadované chování zřetězené řady mobilních robotů, jak bylo diskutováno v kapitole (2.2). Pro daný účel je uvažována znalost jednak aktuální pozice a rychlosti daného člena kolony a také znalost rychlosti a pozice přímých předchůdců každého člena. Řídicí algoritmus a odvození přenosových funkcí lineární kolony vozidel bylo inspirováno materiálem uvedeným v seznamu použité literatury pod číslem [4].

Navrhované řízení, tedy vstupní signál u_i v rovnici (2.14), je obdoba klasické PID regulace a je popsáno vztahem (3.23)

$$u_i = P_i(x_{i-1} - x_i - \tau_d \dot{x}_i) + I_i \left(\int x_{i-1} - x_i - \tau_d \dot{x}_i dt \right) + D_i(\dot{x}_{i-1} - \dot{x}_i)\tag{3.23}$$

kde bezpečnostní prodlevu τ_d lze nahradit konstantní hodnotou bezpečnostního rozestupu $d_0 = \tau_d \dot{x}_i$.

Dále zavedme následující substituce

$$\begin{aligned}d_i &= x_{i-1} - x_i \\v_i &= \dot{x}_i\end{aligned}$$

a s použitím Laplaceovy transformace můžeme zapsat pohybovou rovnici řízeného vozidla následujícím způsobem (3.24).

$$(m_i p + b_i)v_i(p) = (P_i + I_i \frac{1}{p} + D_i p)d_i(p) - (\tau_d P_i + \tau_d I_i \frac{1}{p})v_i(p)\tag{3.24}$$

Rovnice (3.24) je potom výchozím vztahem pro popis chování jednotlivých členů kolony v závislosti na chování svých přímých předchůdců. Přenosová funkce z hodnoty vzdálenosti

i-tého vozidle d_i na jeho rychlost v_i plyne přímo z rovnice (3.24) a je následující (3.25).

$$F_{d_i \rightarrow v_i}(p) = \frac{D_i p^2 + P_i p + I_i}{m_i p^2 + (b_i + \tau_d P_i + D_i) p + \tau_d I_i} \quad (3.25)$$

Dále uvažujme, že první derivace vzdálenosti od předchůdce podle času je zároveň rozdílem rychlostí předchůdce a následovníka, tedy

$$\frac{\delta d_i}{\delta t} = \dot{x}_{i-1} - \dot{x}_i$$

což v Laplaceově formě lze zapsat jako (3.26)

$$p d_i(p) = v_{i-1}(p) - v_i(p) \quad (3.26)$$

S použitím rovnice (3.26) potom můžeme zapsat

$$F_{v_{i-1} \rightarrow d_i}(p) = \frac{m_i p^2 + (b_i + \tau_d P_i + D_i) p + \tau_d I_i}{m_i p^3 + (b_i + \tau_d P_i + D_i) p^2 + (\tau_d I_i + P_i) p + I_i} \quad (3.27)$$

a hledaný vztah spojující rychlost předchůdce s rychlostí následovníka získáme kombinací rovnic (3.27) a (3.25).

$$F_{v_{i-1} \rightarrow v_i}(p) = F_{v_{i-1} \rightarrow d_i}(p) \cdot F_{d_i \rightarrow v_i}(p) = \frac{D_i p^2 + P_i p + I_i}{m_i p^3 + (b_i + \tau_d P_i + D_i) p^2 + (\tau_d I_i + P_i) p + I_i} \quad (3.28)$$

Další důležitou přenosovou funkcí je dozajista závislost vzdálenosti od předchůdce na vzdálenosti předchůdce od svého referenčního vozidla, tedy $F_{d_{i-1} \rightarrow d_i}(p)$. Lze ukázat, že za předpokladu, že $m_i \approx m_{i-1}$ a $\{P_i, I_i, D_i\} \approx \{P_{i-1}, I_{i-1}, D_{i-1}\}$, je přenosová funkce $F_{d_{i-1} \rightarrow d_i}(p)$ stejná jako $F_{v_{i-1} \rightarrow v_i}(p)$.

Výsledky simulací takto řízené lineární kolony jsou v kapitole (4.3).

3.3 Řízení pohybu členů kolony v rovině

Vzhledem k tomu, že ve skutečnosti není řízení kolony pouze jednorozměrným problémem, tj. regulovanou veličinou jednotlivých vozidel není pouze jejich rychlost, ale i směr jízdy, je nutné dále navrhnout algoritmus řízení vedoucí k vytvoření kolony vozidel pohybujících se volně v rovině.

Cílem každého člena *pohybující se* kolony, který nejede na prvním místě (dále jen *následovníka*) by v každém časovém okamžiku mělo být, aby se po uplynutí výše zmíněné bezpečnostní prodlevy τ_d nacházel přesně na místě, kde se v danou chvíli nachází vozidlo jedoucí před ním (dále jen *předchůdce*). Předpokladem tohoto přístupu k řízení následovníků je ovšem dostatečně přesná znalost celé trajektorie jednotlivých předchůdců (resp. celého úseku trajektorie mezi předchůdcem a jeho následovníkem). Navržením řídicího algoritmu pro vozidla osazená dostatečně výkonným počítačem a stereo kamerou určenou k odhadu trajektorie předchůdce na základě relativní vzdálenosti a úhlu natočení se zabývá např. [11]. Vzhledem k možnostem platformy použité v rámci této práce, tedy osmi-bitového čipu s omezenou pamětí, je však nutné učinit určité zmírnění nároků.

3.3.1 Přímková aproximace sledované trajektorie

V první fázi se tedy omezíme na předpoklad, že referenční vozidlo, tedy předchůdce, se mezi svojí aktuální polohou a pozicí následovníka pohyboval po přímce.

V předchozích kapitolách bylo zmíněno, že robot provádí odhad své polohy *pouze* na základě časové integrace rychlostí otáčení svých kol. Tento přístup bývá v literatuře označován jako *Odometrie* a pokud budeme uvažovat diskrétní řízení pomocí číslicového mikrokontroléru s periodou T_s , poloha robotu v dalším časovém okamžiku $(k+1)T_s$ bude ze znalosti $\mathbf{X}(kT_s) = [x(kT_s), y(kT_s)]^T$ vypočtena Eulerovskou integrací jako

$$\mathbf{X}((k+1)T_s) = \mathbf{X}(kT_s) + \begin{bmatrix} \cos(\alpha(kT_s)) \\ \sin(\alpha(kT_s)) \end{bmatrix} \cdot v(kT_s) \cdot T_s \quad (3.29)$$

kde $\alpha(kT_s)$ představuje aktuální úhel natočení robotu v inerciálních souřadnicích a $v(kT_s)$ je okamžitá rychlost robotu. Odhad polohy předchůdce bude prováděn na základě měření vzdálenosti D_P a azimutu (tj. úhlu odchýlení od tečny k pohybu následovníka) α_P podle vztahu

$$\mathbf{X}_P(kT_s) = \mathbf{X}(kT_s) + \begin{bmatrix} \cos(\alpha(kT_s) + \alpha_P(kT_s)) \\ \sin(\alpha(kT_s) + \alpha_P(kT_s)) \end{bmatrix} \cdot D_P(kT_s) \quad (3.30)$$

S využitím těchto vztahů lze zapsat vztah pro výpočet bodu referenční trajektorie v čase kT_s jako

$$\mathbf{X}_{ref}((k+1)T_s) = \frac{\mathbf{X}_P(kT_s) - \mathbf{X}(kT_s)}{|\mathbf{X}_P(kT_s) - \mathbf{X}(kT_s)|} \cdot (|\mathbf{X}_P(kT_s) - \mathbf{X}(kT_s)| - d_0) + \mathbf{X}(kT_s) \quad (3.31)$$

což geometricky představuje bod ležící na přímce procházející body $\mathbf{X}(kT_s)$ a $\mathbf{X}_P(kT_s)$ ve vzdálenosti d_0 "za" pozicí předchůdce $\mathbf{X}_P(kT_s)$. Takto vypočtená posloupnost bodů pro $k = \{1, 2, \dots\}$ je pak přivedena na vstup algoritmu pro následování referenční trajektorie z kapitoly 3.1, čímž je zajištěno chování skupiny robotů jako zřetězené řady. Výsledky simulací s tímto algoritmem jsou v pasáži 4.4.1.

3.3.2 Polynomiální aproximace sledované trajektorie

Další možnost výpočtu referenční trajektorie na základě měření polohy sledovaného vozidla představuje výpočet regresní funkce z několika zjištěných pozic se známým časovým odstupem [5].

Tato metoda opět provádí výpočty nad globální polohou v inerciální soustavě pevně spojené se zemí se souřadným systémem $[x, y]$. Aktuální poloha robotu bude tedy, stejně jako v předchozím případě, vypočtena pomocí odometrické rovnice (3.29) a podobně i poloha předchůdce bude odhadována na základě vztahu (3.30).

Hlavním předností oproti výše představené metodě je předpoklad, že se předcházející vozidlo nepohybuje po trajektorii složené z přímek, ale obecně po částech kuželoseček. Tento přístup zajišťuje významný posun ve smyslu přesnosti následování trajektorie sledovaného vozidla. Na druhou stranu je důležité zmínit, že algoritmus, který bude představen na následujících řádkách je vypočetně i paměťově náročnější než předchozí metoda, protože vyžaduje uchovávání určité *historie* pohybu sledovaného vozidla, tj. několika minulých měření.

Hledaná referenční trajektorie může tedy být vyjádřena na intervalu $t \in (T_1, T_2)$ v polynomiální parametrické formě jako

$$\mathbf{X}_P(t) = \begin{bmatrix} \hat{x}_p(t) \\ \hat{y}_p(t) \end{bmatrix} = \begin{bmatrix} a_2^x t^2 + a_1^x t + a_0^x \\ a_2^y t^2 + a_1^y t + a_0^y \end{bmatrix} \quad (3.32)$$

kde funkce $\hat{x}_p(t)$ a $\hat{y}_p(t)$ jsou aproximacemi funkcí $x(t)$ a $y(t)$, které představují skutečnou trajektorii sledovaného vozidla na zmiňovaném časovém intervalu. Koeficienty polynomů a_i^x, a_i^y jsou pak získány aplikací metody nejmenších čtverců na několik vzorků polohy

sledovaného vozidla na intervalu $t \in (T_1, T_2)$. Ukazuje se, že polynom druhého stupně může (s ohledem na omezení dynamických možností pohybu robotu jako takového) na vhodně zvoleném, resp. dostatečně krátkém, časovém intervalu velmi dobře aproximovat pohyb sledovaného vozidla.

Referenční trajektorie použitá jako vstup zpětnovazebního řízení pohybu robotu z kapitoly 3.1 pak bude při diskrétním řízení vypočtena dle vztahu (3.33).

$$\mathbf{X}_{ref}(kT_s) = \begin{bmatrix} a_2^x kT_s^2 + a_1^x kT_s + a_0^x \\ a_2^y kT_s^2 + a_1^y kT_s + a_0^y \end{bmatrix} \quad (3.33)$$

Mezi další vztahy potřebné k řízení patří pak rovnice (3.34), která slouží pro výpočet úhlu natočení sledovaného vozidla v inerciální soustavě.

$$\alpha_{ref} = ATAN2 \left(\frac{2a_2^y kT_s + a_1^y}{2a_2^x kT_s + a_1^x} \right) \quad (3.34)$$

V této rovnici opět figuruje již dříve popisovaná čtyřkvadrantová varianta goniometrické funkce arcus tangens, jejím argumentem je podíl časových derivací trajektorie, tj. rychlostí, ve směru jednotlivých os.

Dále je ještě vhodné vyjádřit vztahy pro přímovazební část řízení, tedy tedy pro v_{ref} a ω_{ref} z rovnic (3.5) a (3.6). Z matematického hlediska se sice jedná o prosté provedení časových derivací a dosazení do známých vztahů, avšak z výpočetního hlediska se jedná o velkou úsporu času při přímém použití zjištěných koeficientů namísto redundantnímu výpočtu časových derivací v každém kroku.

Tečná rychlost referenční trajektorie v_{ref} je tedy určena z rovnice (3.35) a úhlová rychlost otáčení robotu z rovnice (3.36).

$$v_{ref} = \sqrt{(2a_2^x kT_s + a_1^x)^2 + (2a_2^y kT_s + a_1^y)^2} \quad (3.35)$$

$$\omega_{ref} = \frac{(2a_2^x kT_s + a_1^x)^2 \cdot 2a_2^y - (2a_2^y kT_s + a_1^y)^2 \cdot 2a_2^x}{(2a_2^x kT_s + a_1^x)^2 + (2a_2^y kT_s + a_1^y)^2} \quad (3.36)$$

Posledním významným vztahem je rovnice regulační odchylky pro zpětnovazební část řízení. Tato rovnice opět odpovídá vztahu (3.2), avšak namísto referenční trajektorie coby deterministické časové funkce zde vystupují regresně vypočtené hodnoty ze vztahů (3.33) a (3.34).

$$e = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{X}_{ref} - \mathbf{X} \\ \alpha_{ref} - \alpha \end{bmatrix} \quad (3.37)$$

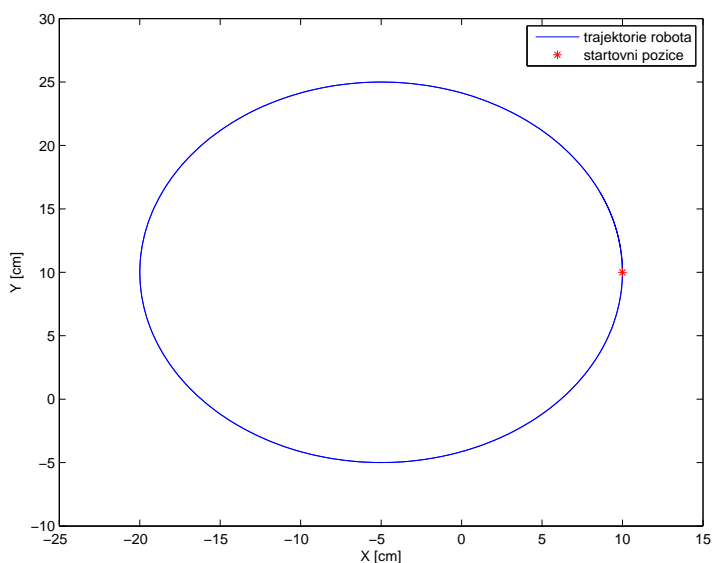
Užitím těchto vztahů pro řízení jednotlivých členů kolony je tedy zajištěno požadované chování. Výsledky simulací řízení několika robotů tímto algoritmem jsou uvedeny v kapitole 4.4.2.

4 Simulační ověření navržených řídicích algoritmů

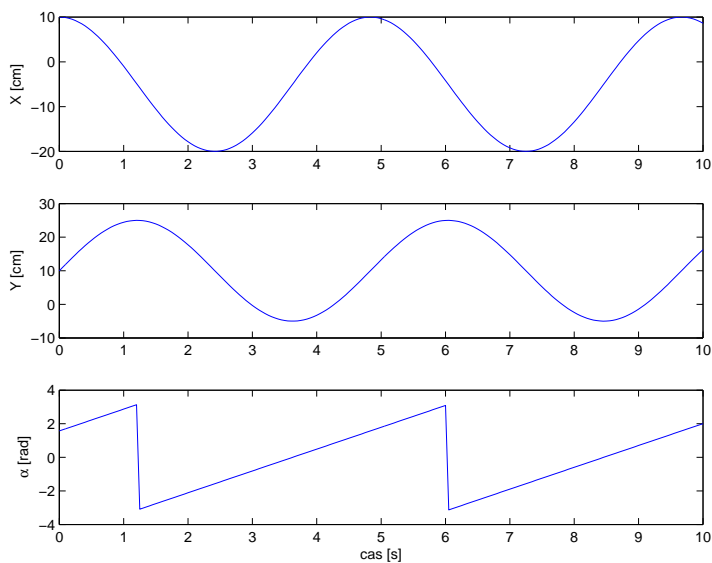
Tato kapitola je věnována prezentaci výsledků simulací vázaných na jednotlivé sekce v předchozích dvou kapitolách. Tyto simulace byly prováděny za použití programu Simulink, který je nadstavbou matematického softwaru MATLAB od společnosti MathWorks Inc.

4.1 Pohyb diferencálně řízeného robotu

Na následujících dvou obrázcích je znázorněno chování modelu robotu, pokud jsou kola řízena na konstantní, avšak různé úhlové rychlosti otáčení $\omega_1 = 8\text{rad} \cdot \text{s}^{-1}$ pro pravé kolo a $\omega_2 = 4\text{rad} \cdot \text{s}^{-1}$ pro levé kolo. Obrázek 5 znázorňuje pohyb robotu v rovině $[x, y]$, zatímco Obrázek 6 ukazuje vývoj jednotlivých stavových proměnných v závislosti na čase.



Obrázek 5: Simulace pohybu robotu pro konstantní vstupy - pohyb v rovině

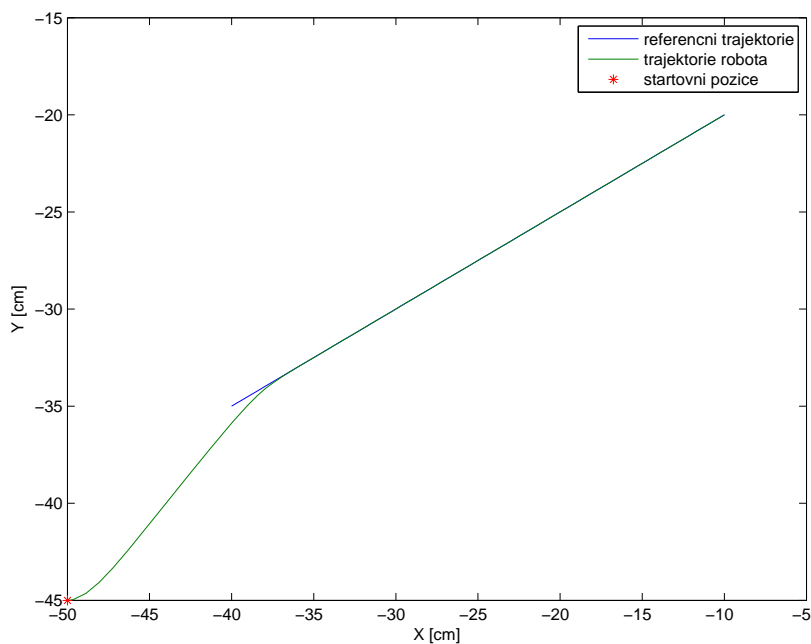


Obrázek 6: Simulace pohybu robotu pro konstantní vstupy - časový vývoj proměnných

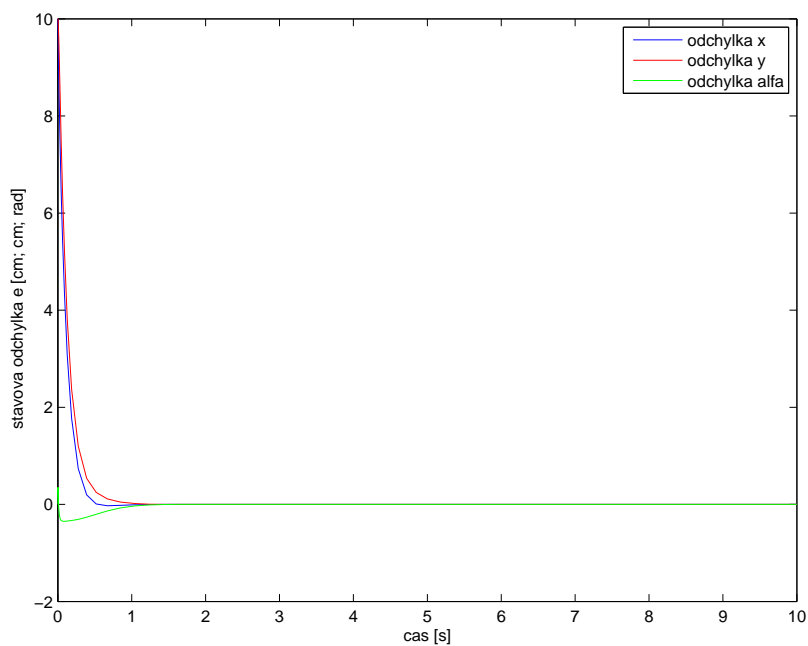
Z obrázků je patrné, že matematický model robotu vykazuje očekávané chování, tj. opisuje kruhovou trajektorii. Úhel jeho natočení se mění způsobem odpovídajícím funkci $ATAN2$ z rovnice 3.1.

4.2 Sledování referenční trajektorie

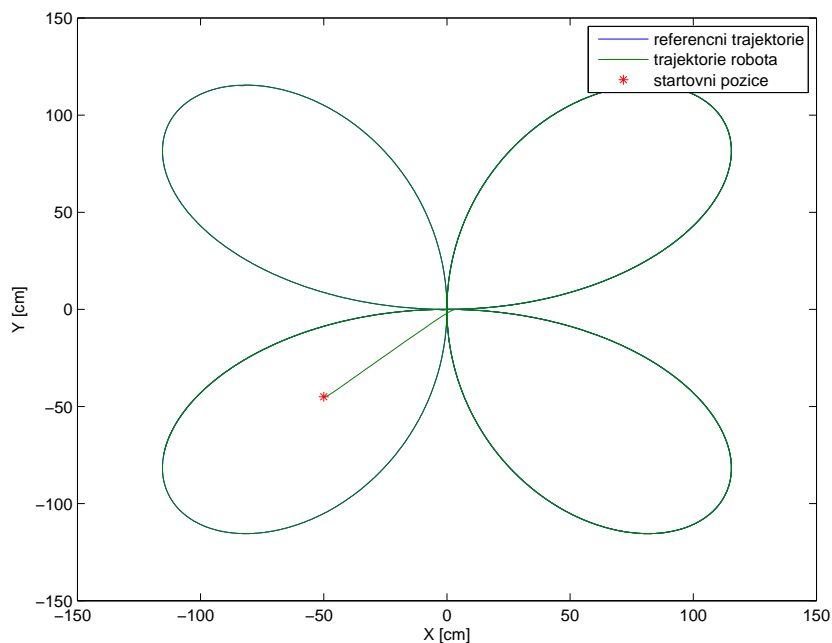
Dalším důležitým aspektem řízení kolony vozidel je bezesporu řízení jednotlivých členů tak, aby následovali určenou trajektorii. Výsledky simulací, kdy byl model robotu z kapitoly 2.1 řízen pomocí stavového regulátoru z kapitoly 3.1 jsou na následujících obrázcích. Pro úplnost uvádím, že jako generátoru referenční trajektorie bylo v obou případech užito vektorové funkce času, tj. $[x_{ref}, y_{ref}] = \vec{f}(t)$. První dvojice obrázků znázorňuje pohyb robotu v případě, že je referenční trajektorie rovná přímce. Druhý pár potom ukazuje řízení pohybu po složitější trajektorii, v tomto případě tzv. růžičkové funkci.



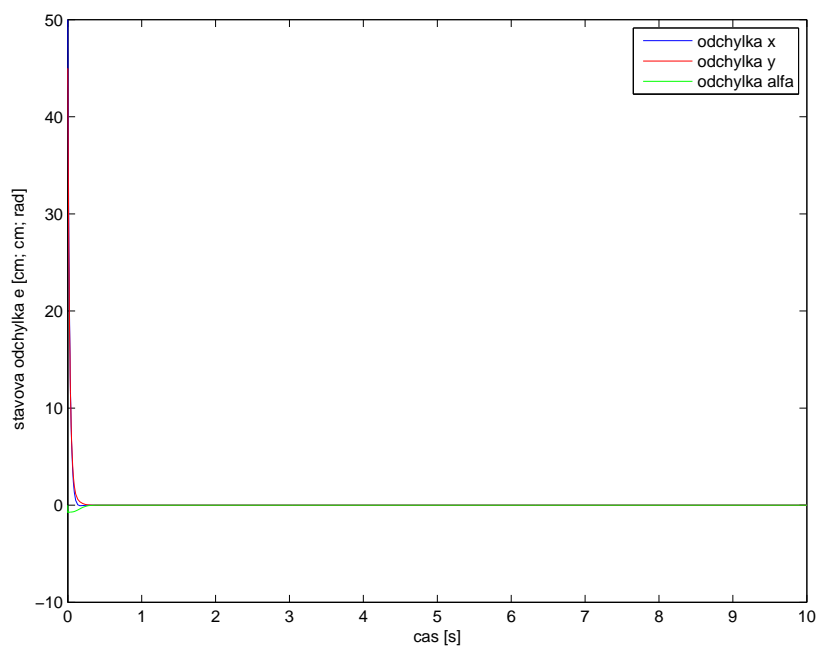
Obrázek 7: Regulační pohyb po přímce - pohyb v rovině



Obrázek 8: Regulační pohyb po přímce - vývoj stavové odchylky



Obrázek 9: Regulace pohybu po zakřivené trajektorii

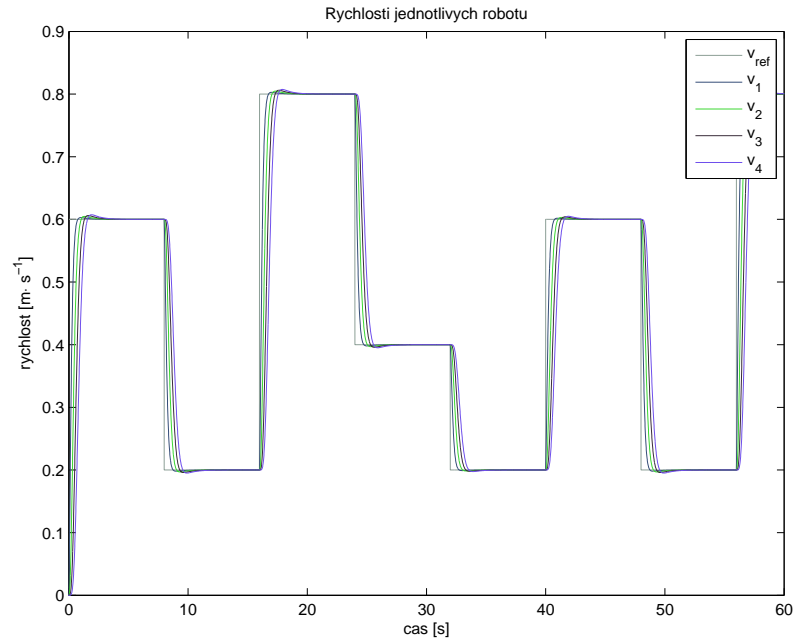


Obrázek 10: Regulace pohybu po zakřivené trajektorii - vývoj stavové odchylky

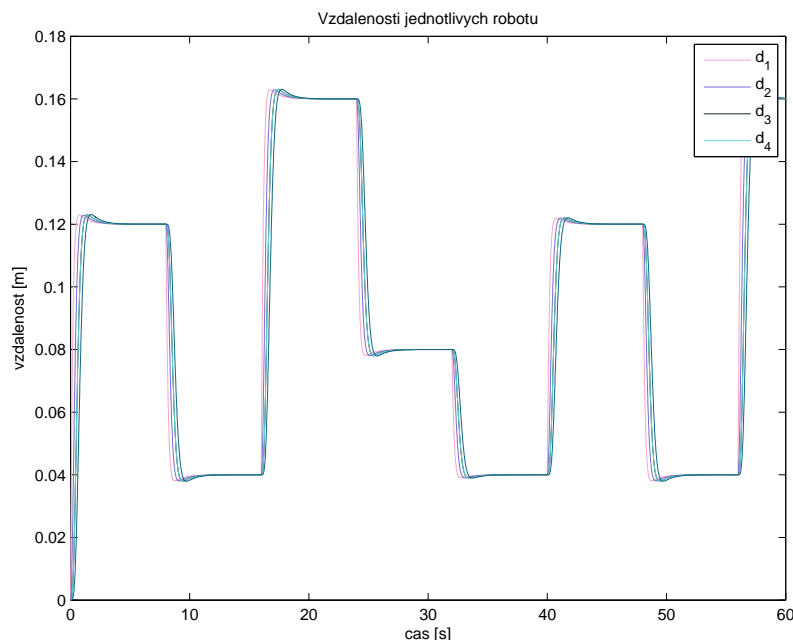
Z obrázků je patrné, že navržený regulátor zajišťuje v ustáleném stavu nulovou odchylku sledovaných proměnných, avšak dlužno poznamenat, že v simulacích nebyly aplikovány žádné šумы či poruchy, které v nesterilním prostředí provozu robotů s velkou pravděpodobností budou přítomny. Uvážíme-li však, že v praxi bude trajektorie všech "následujících" členů kolony, tedy druhého a každého dalšího, generována na základě *odhadu* polohy sledovaného předchůdce, nejví se již tento nedostatek natolik významným, neboť zásadnějším ukazatelem kvality řízení v koloně je *vzájemná* poloha jednotlivých členů.

4.3 Řízení lineární kolony vozidel

V této sekci jsou prezentovány výsledky simulací řízení lineární kolony z kapitoly 3.2. Referenčním signálem reprezentujícím rychlost je schodovitá funkce s hodnotami, jichž by reálné rychlosti uvažovaných vozidel mohly dosahovat. Chování kolony z pohledu okmažitých rychlostí členů a vzdáleností mezi členy je na Obrázku 11 a Obrázku 12.



Obrázek 11: Lineární kolona - rychlosti členů



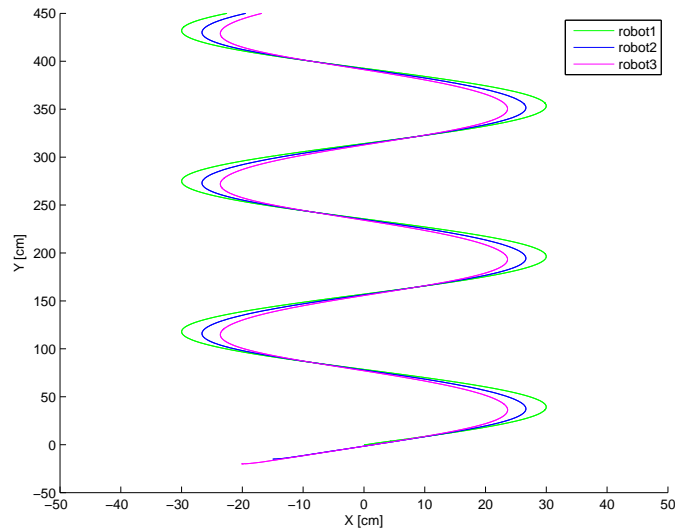
Obrázek 12: Lineární kolona - relativní vzdálenosti

Z výše uvedených grafů je patrné, že navržený způsob řízení zajišťuje uspokojivé chování celé kolony a bylo by možno jej použít například jako v automobilovém průmyslu hojně používaný adaptivní tempomat.

4.4 Řízení zřetěžené řady diferenciálně řízených robotů

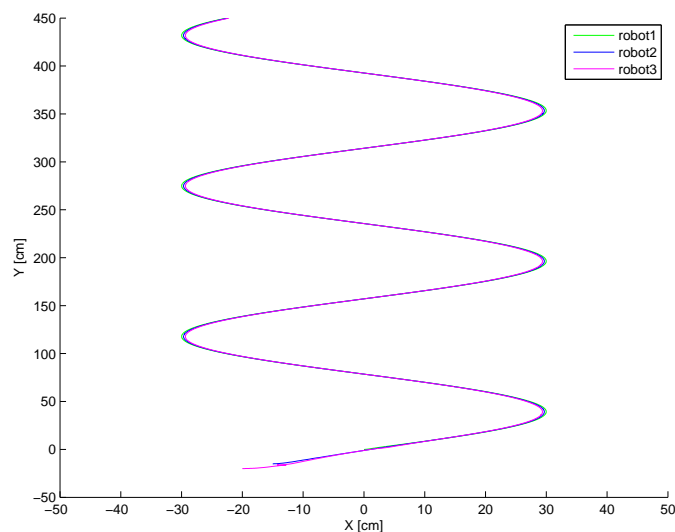
4.4.1 Přímková aproximace sledované trajektorie

V této sekci budou prezentovány výsledky simulací řízení zřetěžené řady mobilních robotů s použitím řídicího algoritmu navrženého v sekci 3.3.1. V této simulaci bylo implementováno řízení tří robotů, z nichž první sleduje sinovou referenční trajektorii podél osy Y a další dva jsou řízeny podle algoritmu z rovnice (3.31) s parametrem bezpečné vzdálenosti $d_0 = 10 \text{ cm}$.



Obrázek 13: Řízení kolony v rovině - přímková aproximace

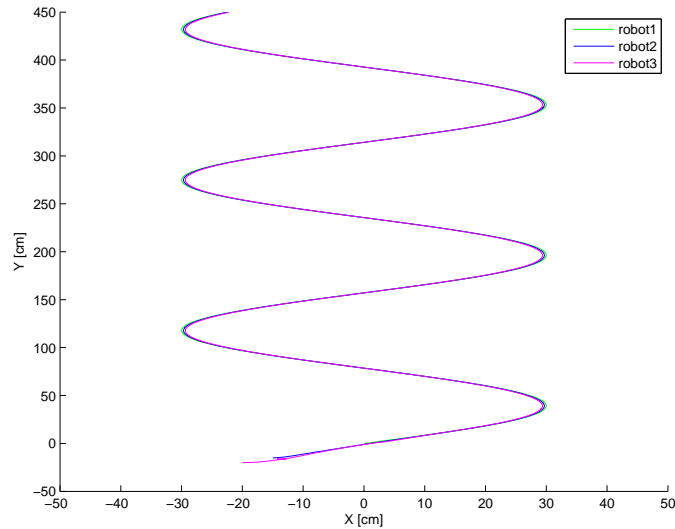
Z výsledků je patrné, že tento přístup s sebou přináší chybu sledování (zejména v místech náhlých změn směru), která je propagována zřetěženou řadou a s rostoucím počtem členů kolony by její váha geometricky rostla. Na druhou stranu je třeba říct, že s klesajícím koeficientem d_0 by se tato chyba naopak významně zmenšovala, což dokládá Obrázek 14, jakožto výstup téže simulace s parametrem $d_0 = 1 \text{ cm}$. Snížení chyby je však docíleno na úkor bezpečnosti, což obecně v oboru řízení vozidel není žádoucí.



Obrázek 14: Řízení kolony v rovině - přímková aproximace

4.4.2 Polynomiální aproximace sledované trajektorie

V této sekci budou prezentovány výsledky simulací řízení zřetěžené řady mobilních robotů s použitím řídicího algoritmu navrženého v sekci 3.3.2. Podobně jako v předchozí sekci, i tentokrát je simulováno řízení kolony o třech členech, z nichž první sleduje sinovou referenční trajektorii podél osy Y a další dva jsou řízeny podle algoritmu z rovnice (3.37)



Obrázek 15: Řízení kolony v rovině - polynomiální regrese

Výsledky simulací kolony takto řízených mobilních robotů již vykazují uspokojivé chování, tedy následování trajektorie vedoucího robotu se zanedbatelnou chybou při dodržování bezpečné vzdálenosti, která je v tomto případě kvantifikována hodnotou bezpečnostní časové prodlevy $\tau_d = 1 \text{ sec}$.

Přesnost následování trajektorie referenčního robotu následovníky by mohla být při experimentech s reálnými vozidly zvýšena navýšením počtu vzorků pro výpočet regresního polynomu v generátoru referenční trajektorie ze stávajících tří na libovolně větší počet. Tento úkon s sebou však přináší úměrný nárůst výpočetní náročnosti a je proto nutné nalézt vhodný počet v závislosti na charakteru řízené platformy, zejména řídicí jednotky a senzorů odhadujících pozici předchůdce.

5 Návrh experimentální platformy

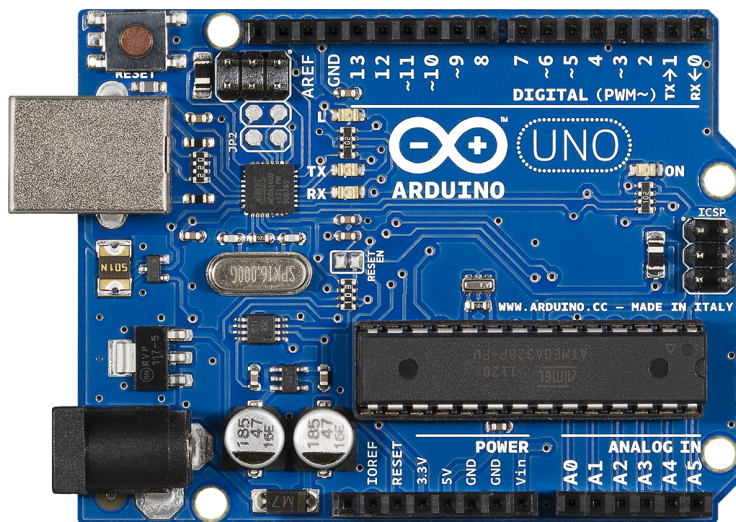
Další kapitola této práce je věnována návrhu platformy pro experimentální ověření navržených řídicích algoritmů. V následujících odstavcích bude představena platforma vytvořená pro tyto účely v laboratorním prostředí, která sestává z řídicí jednotky, šasi, podvozku s elektrickým pohonem a sady senzorů.

5.1 Platforma Arduino

Nejdůležitější součástí každého automatizovaného systému je zřejmě řídicí jednotka, která odpovídá požadavkům a potřebám daného problému. Pro účely této práce bylo zapotřebí najít jednotku, která by byla schopna sbírat a vyhodnocovat informace ze senzorů, podle těchto informací generovat řízení a to aplikovat na fyzické výkonové prvky, tj. motory. Po zvážení těchto požadavků byl vybrán mikrokontrolér typu Arduino, který bude nyní v krátkosti představen.

Arduino je open-source platforma založená na mikrokontrolerech ATMega od firmy Atmel. Arduino může být použito k vytváření samostatných interaktivních zapojení nebo může být připojeno k aplikacím běžícím na běžném počítači.

Desky arduino obsahují 8-bitové mikrokontrolery řady AVR od firmy Atmel a množství dalších podpůrných obvodů. Oficiální vydání Arduina, které vyrábí a prodává Italská firma Smart Projects, používají čipy ATMega8, ATMega168, ATMega328, ATMega1280 a ATMega2560. Každá deska má většinu I/O pinů přístupných přes precizní patice, do kterých se jednoduše připojují další obvody, kterým se ve světě Arduina říká Shiedly. Na deskách bývá několik diod, resetovací tlačítko, konektory pro ICSP programování, napájecí konektor, oscilátor a modul zprostředkávající komunikaci po USB. Základní verze Arduina, Arduino Uno (na Obrázku 16), poskytuje celkem 14 I/O digitálních pinů a 6 analogových pinů. Z toho šest digitálních pinů je také možné použít na softwarově řízený PWM výstup.



Obrázek 16: Jednotka Arduino UNO

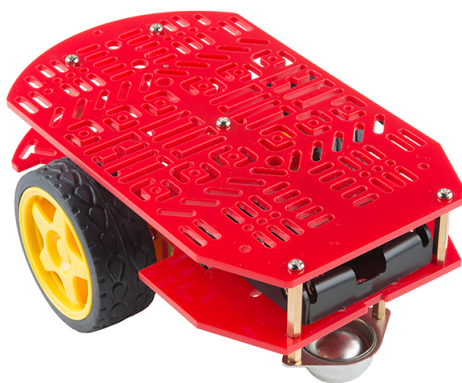
Hlavní mikrokontrolér, který je uživatelsky programovatelný, má bootloader (kód, který se po spuštění postará o základní nastavení mikrokontroleru, jako jsou interní časovače, nastavení rozhraní USART a další) a nastavené potřebné fuses bajty (těmi se nízkourovňově nastavují některé vlastnosti čipu). Díky tomu se uživatel nemusí starat o detaily a své programy píše v jazyce podobném C/C++.

Ačkoliv je Arduino připojeno k počítači pomocí rozhraní USB, je softwarově simulována komunikace přes sériovou linku (RS-232). Ve starších deskách se pro tyto účely používaly FTDI čipy, v novější desce Uno toto obstarává ATmega8U2 s předprogramovaným firmware (který je, jako vše ostatní, volně dostupný v podobě zdrojových kódů).

Technická data i ilustrační obrázek ze sekce 5.1 byly čerpány z webových stránek výrobce těchto zařízení uvedené v seznamu použité literatury pod číslem [7].

5.2 Použité šasi

Při výběru vhodné kostry celého vozidla byl kladen důraz zejména na jednoduchost konstrukce, robustnost a maximální modularitu. Na základě těchto kritérií bylo zvoleno tzv. *Magician chassis*, které se skládá ze dvou hustě perforovaných plastových desek spojených hliníkovými distančními válci. Velkou předností tohoto šasi je zejména vysoká modularita, možnost bezproblémově uchytnout všemožné potřebné komponenty do předpřipravených otvorů. Prázdné šasi, osazené pouze základním podvozkem a pouzdem na baterie je vyobrazeno na Obrázku 17.



Obrázek 17: Použité šasi - *Magician chassis*

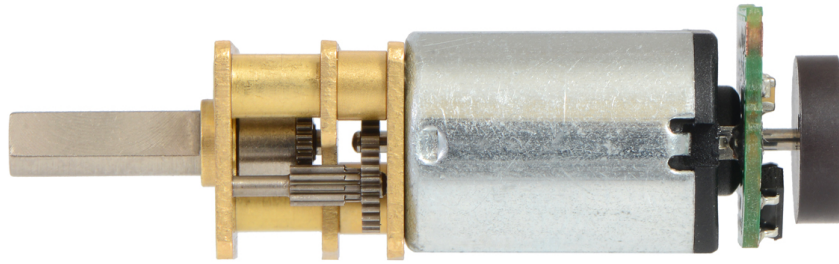
Na takto sestavenou kostru bylo namontováno Arduino Uno i ostatní komponenty potřebné komponenty, které budou diskutovány dále.

5.3 Podvozek a řízení pohybu

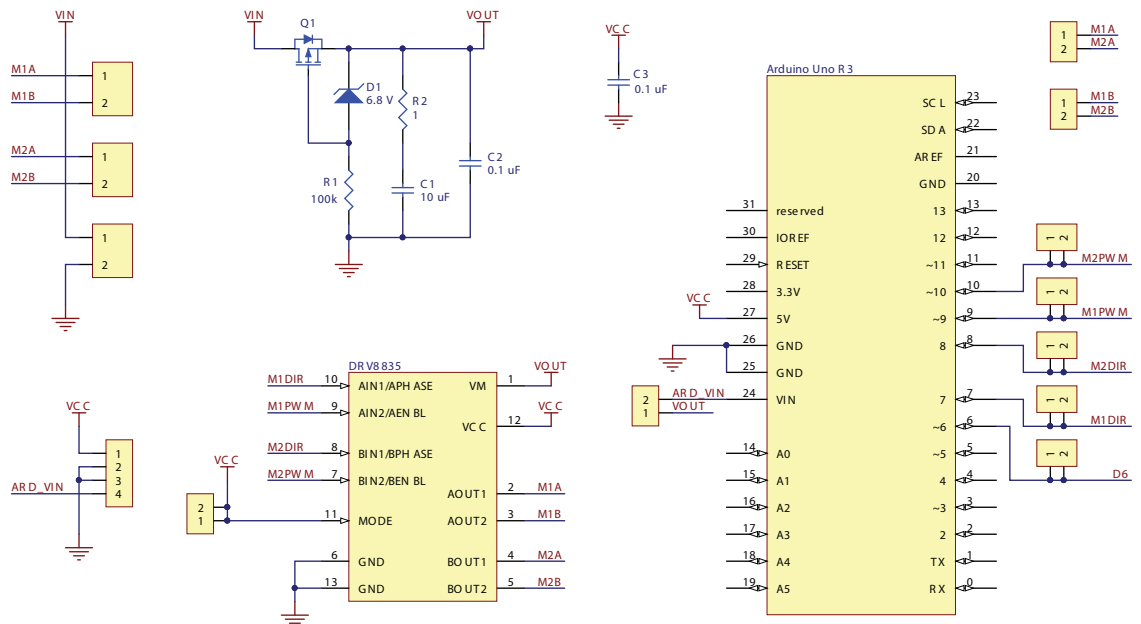
Oproti Obrázku 17 došlo na použitých robotech k nahrazení širokých plastových kol užšími s gumovým návkem. Jejich hlavním přínosem je eliminace prokluzu při kontaktu s hladkou podložkou a tím pádem redukce chyby v integrovaném odhadu polohy a orientace robotu v prostoru.

Tato kola jsou poháněna malými stejnosměrnými elektrickými motory, které pracují v napěťovém rozmezí 3 - 9 Voltů. Realizace motoru s převodovou soustavou a použitým Hallovým enkodérem je na Obrázku 18. Motory jsou řízeny prostřednictvím nadstavbového řadiče připojeného na centrální výpočetní jednotku, tedy mikrokontrolér Arduino Uno. Schéma použitého řadiče motorů je na Obrázku 19 a popis jeho funkce je uveden níže.

Napětí z baterie je přes stabilizátor přivedeno na porty VIN a GND modulu DRV8835, jehož funkce je řízena pomocí Arduino GPIO pinů 6 až 9, do nichž je přímo uchycen. Hodnota digitálních pinů 7 a 8 stanovuje směr otáčení jednotlivých motorů (PHASE),



Obrázek 18: Použitý motor s enkodérem



Obrázek 19: Schéma řadiče Pololu DRV8835

zatímco piny 6 a 9 pulzně-šířkovou modulací (PWM, viz. podkapitola 5.3.1) regulují vlastní výstupní napětí (ENABLE) a tím rychlost otáčení obou motorů. Fungování řadiče motorů je znázorněno v následující tabulce.

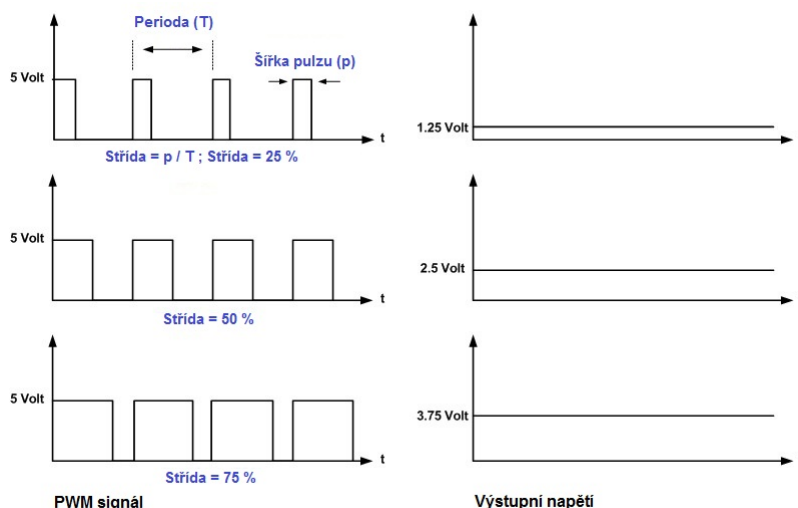
Logika řízení v základní PHASE/ENABLE módu				
xPHASE	xENABLE	MxA	MxB	operační mód
0	PWM	PWM	L	pohyb vpřed rychlostí PWM
1	PWM	L	PWM	pohyb vzad rychlostí PWM
X	0	L	L	brzdění (výstupy uzemněny)

Obrázek 19 stejně jako výše uvedená tabulka řídicí logiky jsou čerpány z materiálu uvedeného v seznamu použité literatury pod číslem [8].

5.3.1 Pulzně šířková modulace

Tento druh modulace je jedním z nejrozšířenějších přístupů ke generování analogového signálu v oblasti elektroniky a řídicí techniky. Často se můžeme setkat také s označením PWM, které pochází z anglického označení *Pulse Width Modulation*. PWM je tedy diskretní modulace, která se používá pro přenos analogového signálu pomocí dvouhodnotového signálu.

Přenosový signál tedy může nabývat pouze dvou hodnot: zapnuto (logická 1) a vypnuto (logická 0). Hodnota přenášeného signálu je pak "zakódována" jako poměr mezi stavy zapnuto/vypnuto. Tento poměr se nazývá *střída* a může nabývat hodnot 0 - 100 %. Doba mezi dvěma přechody ze stavu vypnuto do stavu zapnuto se nazývá *perioda* a ku příkladu na používané platformě Arduino má délku dvou milisekund. Ukázka realizace PWM signálu je na Obrázku 20.



Obrázek 20: Časový diagram PWM modulace

Další velmi důležitou součástí podvozku jsou enkodéry. Tato čidla jsou uchycena přímo na oba motory a jejich princip i účel bude rozebrán v následující sekci.

5.4 Použité senzory

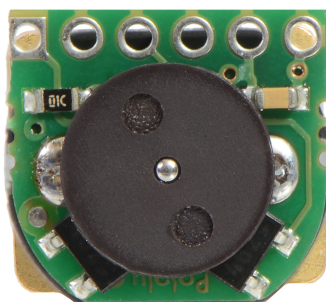
Na robotech použitých pro experimentální ověření matematických a simulačních závěrů byly použity dva druhy senzorů. Jedná se jednak o trojici ultrazvukových dálkoměrů typu SRF05 a dále jsou to dva páry Hallových sond připevněné na elektromotorech.

5.4.1 Enkodéry

Enkodéry jsou obecně čidla, která slouží k čítání otáček motorů. K tomuto účelu může sloužit několik různých senzorů. Mezi nejpoužívanější patří jednotky, v nichž je proti sobě postaven fototranzistor a dioda, která emituje infračervené světlo - do štěrbin mezi těmito dvěma součástkami je pak vloženo perforované kolo, které při otáčení střídavě zastíňuje světelný tok, a tím na výstupu fototranzistoru vytváří pulzy.

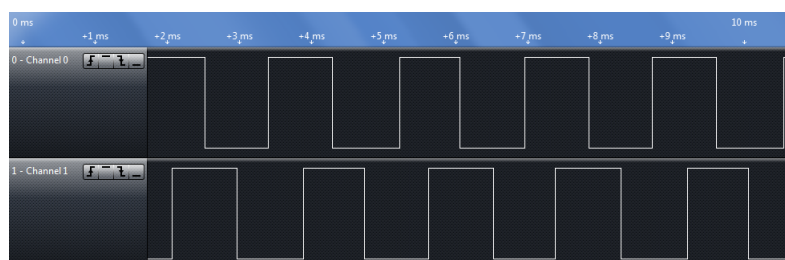
Druhým často používaným řešením je nasazení čidel využívajících tzv. Hallův jev. Jedná se o polovodičovou destičku, skrz níž prochází proud a v níž při vložení do magnetického dochází k přeskupení náboje a tím pádem i ke změně napětí. Napětí na Hallově sondě tedy určuje, zda-li se sonda nachází v magnetickém poli, či nikoliv, a toho je právě využito v magnetických enkodérech.

V rámci experimentů k této práci bylo využito kompletní řešení od firmy Pololu, které je navrženo pro kompaktní uchycení na použité motory (viz. Obrázek 18) a skládá se z dvojice Hallových sond uchycených na integrovaném obvodu a zmagnetovaného kotouče, na němž jsou tři dvojice magnetických pólů. Tento kotouč je uchycen na vývodové hřídeli elektromotoru a při svém otáčení působí změnami magnetického pole na Hallovy sondy. Čelní pohled na zkompletovanou jednotku je na Obrázku 21.



Obrázek 21: Magnetický enkodér

Velkou výhodou přítomnosti hned dvou Hallových sond na použitém modulu je možnost zjišťovat kromě rychlosti otáčení i směr rotace. Toho se dosahuje porovnáním fázového rozdílu obou signálů. Průběh obou signálů naměřených při otáčení motoru na 50 % výkonu bez zátěže je na Obrázku 22. V tomto případě je fázový předstih zhruba třetina periody, což odpovídá postavení obou sond na desce (viz. Obrázek 21), na straně levého enkodéru označeného jako *Channel 0* a tudíž se z čelního pohledu Obrázku 21 motor otáčí proti směru hodinových ručiček.



Obrázek 22: Výstup magnetických enkodérů

Z průběhu obou signálů na Obrázku 22 je patrná přítomnost hysterezního obvodu s operační zesilovačem na enkodérové jednotce, který redukuje přítomnost šumu a záskmitů na naprosté minimum. Jedná se tedy o binární signál, který má hodnotu logické 1 (resp. HIGH, 5 V), pokud se nad danou Hallovou sondou nachází magnetický pól a hodnotu logické 0 (resp. LOW, 0 V), pokud se Hallova sonda nachází mimo *dostatečně silné* magnetické pole. Díky tomu je možné jednoduše implementovat spolehlivý čítač pulzů bez nutnosti softwarového filtrování signálu.

Signály z enkodérů jsou snímány prostřednictvím digitálních vstupů mikrokontroléru a jsou procesorem asynchronně obsluhovány prostřednictvím tzv. externích přerušení. Tato

problematika je detailně rozebrána v materiálu uvedeném v seznam použité literatury pod číslem [9]. V rámci této práce budou představeny pouze základní principy.

Využití tohoto přístupu je vzhledem k charakteru signálu nezbytné, neboť časová prodleva mezi jednotlivými pulzy se mění s měnící se rychlostí otáčení motorů a je zároveň relativně krátká (přibližně 1 ms). Proto by při deterministickém opakovaném čtení hodnot napětí na příslušných digitálních vstupech (s určitou *delší* periodou) mohlo dojít k "promeškání" několika pulzů. Naopak při zvolení příliš krátké periody by docházelo ke zbytečnému plýtvání cenným výpočetním časem. Namísto toho se příslušná část kódu vykoná pouze ve chvíli, kdy na příslušném pinu dojde ke změně logické hodnoty.

Samotná implementace měření rychlosti otáčení motorů vyvinutá pro účely této práce využívá faktu, že ačkoliv jsou výstupem každé enkodérové jednotky dva signály, ve skutečnosti se jedná v podstatě o jeden signál s různým fázovým posunem. Díky tomu není zapotřebí vyvolávat přerušování při změně logické hodnoty (výskytu tzv. hrany) v obou signálech, ale stačí sledovat hrany v jednom signálu a v rámci obsluhy přerušování¹ přečíst hodnotu druhého signálu za účelem zjištění směru otáčení. Navržená ISR tedy provádí pouze inkrementaci nebo dekrementaci příslušné celočíselné proměnné `counti`, která je určena k čítání pulzů během jedné periody výpočtu rychlosti podle vztahu (5.1), kde T_v představuje periodu výpočítávání rychlosti a N_p je počet pulzů vygenerovaných za jednu otáčku kola.

$$\omega_i = 2\pi \frac{\text{count}_i}{T_v \cdot N_p} \quad (5.1)$$

Při běhu programu pak v každém čase $t = kT_v, k = 1, 2, \dots$ dojde k výpočtu rovnice (5.1) a vynulování čítačů `counti`, čímž je vypočítána *průměrná* úhlová rychlost otáčení i-tého motoru. Vzhledem k rozlišení enkodérů $N_p = 308$, zvolené časové periodě $T_v = 0.1$ s a dynamickým vlastnostem užití platformy však lze tuto rychlost pro řídicí účely považovat za rychlost *okamžitou* a využít ji mj. pro odhad polohy robotu podle rovnic (3.22).

5.4.2 Ultrazvukový dálkoměr

Hlavním účelem této práce je vytvoření systému řízení kolony vozidel, a proto je nezbytné, aby jednotliví členové této kolony měli informace o vzájemné poloze se svým předchůdcem. K tomuto účelu slouží trojice ultrazvukových dálkoměrů umístěných na přední části robotu.

Tato čidla měří vzdálenost k objektu před sebou na základě časové prodlevy mezi vysláním ultrazvukového signálu a příjmem odražené "ozvěny". Na základě tohoto časového údaje je pak vypočtena předpokládaná vzdálenost ze známého vztahu pro rychlost $s = v \cdot t$, kde v je známá rychlost zvuku a t změřený časový úsek.

Použité čidlo typu SRF05 od firmy Devantech [10] je vyobrazeno na Obrázku 23. Pro komunikaci s MCU se využívá pětice pinů v horní části desky. Piny umístěné u dolní hrany desky jsou určeny k programování přítomného mikročipu a při běžném používání se nezapojují. Z technických parametrů této jednotky bych jmenoval frekvenci vydávaného ultrazvuku, která činí 40 kHz, a dále pak deklarovaný rozsah měření, který je od 1 cm do 4 m.

Ultrazvukový signál je emitován z levého tubusu a šíří se do prostoru v kuželovitém prostoru s vyzařovacím úhlem zhruba 55°. Testování ukázalo, že takto vzniklá detekční zóna se ve vzdálenosti zhruba 1,5 metru od čidla začíná opět sužovat a čtyři metry vzdálený objekt je detekovatelný pouze v blízkém okolí myšlené osy kolmé na desku senzoru. Těchto vlastností bylo s výhodou využito k účelu detekování vzdálenosti a úhlu vychýlení

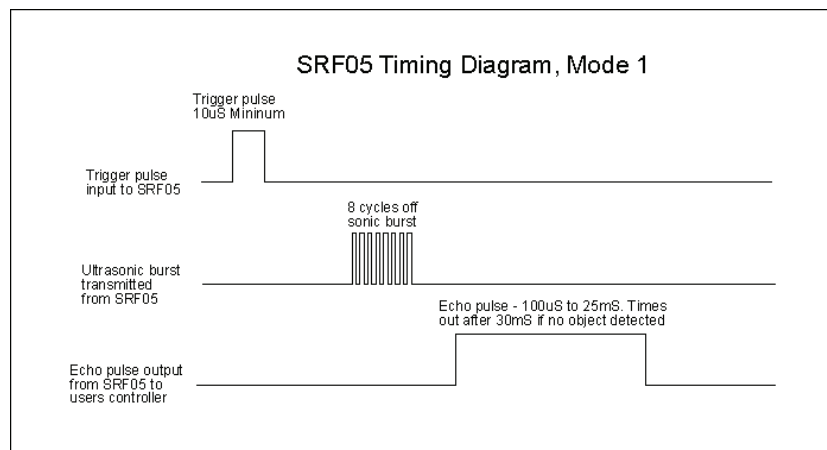
¹jedná se o část kódu označovanou jako ISR (z angl. Interrupt Service Routine), která je spouštěna pokaždé, když na zvoleném pinu dojde k určené události



Obrázek 23: Ultrazvukový dálkoměr Devantech SRF05

předchůdce (tj. vozidla jedoucího v koloně před daný vozidlem) od přímého směru. Na přední část šasi byly umístěny tři jednotky SRF05, z nichž prostřední míří přímo ve směru jízdy, a dvě další s odchylkou $\pm 30^\circ$ od přímého směru. Díky tomu je možné odhadovat nejen vzdálenost, ale i směr pohybu vpředu jedoucího vozidla.

Samotné měření vzdálenosti pak probíhá periodicky ve stanovených intervalech T_m , kdy se postupně provede aktivace trigger signálu na jednotlivých dálkoměrech a je změřena doba zachycení ozvěny dle Obrázku 24.



Obrázek 24: Diagram použití ultrazvukového dálkoměru

Na základě změřené doby zachycení odezvy t_o [μs] je pak za použití známé rychlosti šíření (ultra)zvuku v prostoru vypočtena vzdálenost od nejbližšího objektu v "zorném poli" čidla dle vztahu d [cm] = $\frac{t_o}{58}$. Logika odhadu relativního odchýlení předchůdce od osy pohybu následovníka je uvedena v následující tabulce, v níž hodnoty ve sloupcích $d_{1,2,3}$ představují výstupy jednotlivých ultrazvukových dálkoměrů², \hat{d} značí odhad vzdálenosti předchůdce a $\hat{\alpha}$ je odhadovaný úhel vychýlení předchůdce od osy pohybu následovníka.

d_1	d_2	d_3	\hat{d}	$\hat{\alpha}$
$\sim d$	d_{max}	d_{max}	d	-30°
$\sim d$	$\sim d$	d_{max}	d	-15°
d_{max}	$\sim d$	d_{max}	d	0°
d_{max}	$\sim d$	$\sim d$	d	$+15^\circ$
d_{max}	d_{max}	$\sim d$	d	$+30^\circ$

²senzor s číslem 1 je vychýlen z přímého směru o 30° doleva, senzor s číslem 2 míří přímo ve směru pohybu robotu a senzor s číslem 3 hledí o 30° doprava od osy pohybu robotu

Pro úplnost dodejme, že případy, kdy všechny senzory "nevidí nic", tzn. na výstupech všech tří je d_{max} , jsou v implementaci vyřešeny uchováním informace o posledním známém směru vychýlení. Podle toho je pak předpokládáno vychýlení o -90° , pokud byl předchůdce naposledy zaznamenán levým senzorem, nebo o $+90^\circ$, pokud byl naopak předchůdce naposledy zjištěn pravým senzorem.

Je také důležité zmínit, že přesnost měření je ovlivněna materiálem, kterým je pokryt objekt v zorném poli senzoru. Senzor například vůbec nezaznamená přítomnost plyšové hračky, protože od měkkého povrchu se ultrazvuk neodrazí, ale je pohlcen. Dalším rizikem je, že pokud je překážka *nevhodně* natočena, může dojít k odrazu signálu mimo přijímač senzoru a překážka může být rovněž "přehlédnuta".

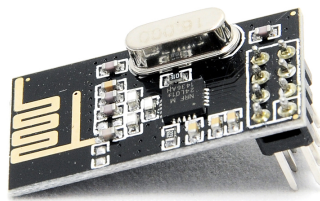
5.5 Komunikační prostředky

Přestože je navrhované řízení robotů autonomní a komunikace mezi jednotlivými členy kolony není uvažována, je komunikace s mikrokontroléry nutná za účelem diagnostiky a prezentace dat. Nejjednodušším způsobem komunikace mezi stolním počítačem a jednotkou Arduino je sériová komunikace prostřednictvím USB kabelu, avšak vzhledem k počtu současně řízených robotů a nutnosti jejich volného pohybu bylo nutné experimentální platformu doplnit o bezdrátové jednotky za účelem bezproblémového shromažďování diagnostických dat.

5.5.1 Bezdrátové moduly

Vzhledem k účelu potřebné bezdrátové komunikace nebylo nutné využít žádné speciálně sofistikované řešení, neboť se v tomto případě nejedná o přenos řídicích, ale pouze diagnostických dat, a proto není kladen přílišný důraz na dobu odezvy ani kapacitu.

Z těchto důvodů bylo využito jednoduchých modulů typu nRF24L01 (viz. Obrázek 25), které jsou s rozměry $1.5 \times 2.5 \text{ cm}$ jednoduše umístitelné na šasi mobilních robotů a nezvyšují významným způsobem hmotnost ani energetickou náročnost celé platformy. Tyto moduly pracují ve volném frekvenčním pásmu 2.4 GHz, jejich maximální šířka pásma je 2 Mbps a s mikrokontroléry komunikují prostřednictvím standardní SPI sběrnice.



Obrázek 25: Bezdrátový modul nRF24L01

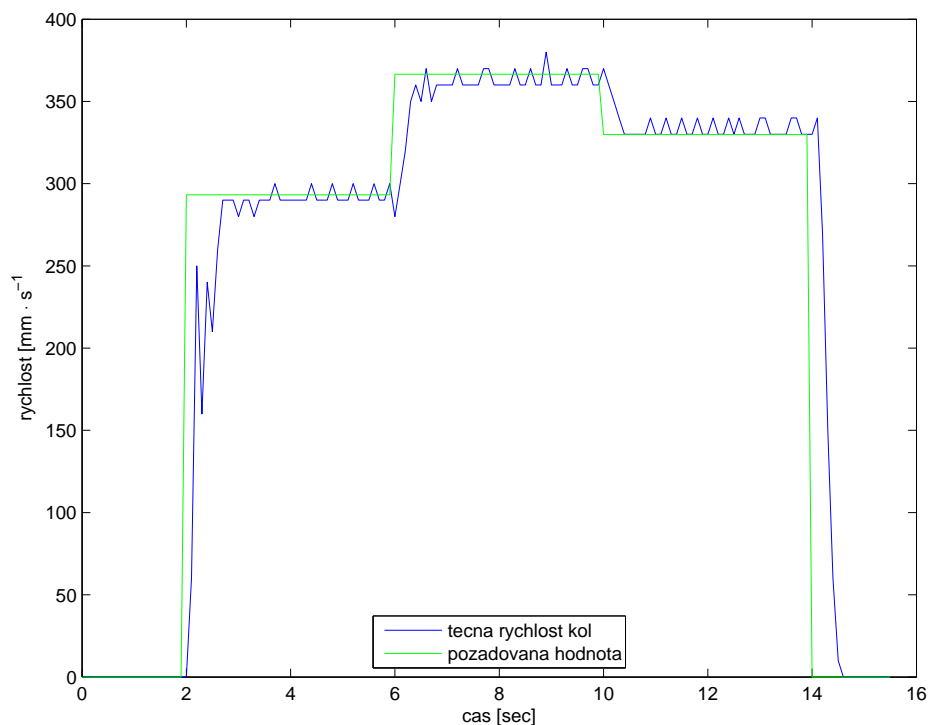
Moduly mají několik operačních módů, z nichž byly pro účely potřebné komunikace využity pouze základní Tx (transmit, vysílání) a Rx (receive, přijímání). K vytvoření sítě bylo tedy využito pěti mikrokontrolérů Arduino vybavených těmito moduly, z nichž čtyři pracovaly v režimu Tx a se zvolenou periodou zasílaly diagnostická data o svém pohybu páté jednotce, která je pomocí sériové linky předávala do stolního počítače ke zpracování.

6 Experimentální ověření navržených algoritmů

Poslední část této práce je věnována prezentaci výsledků experimentů, které byly prováděny na navržené platformě v laboratorním prostředí za účelem ověření navržených řídicích algoritmů.

6.1 Regulace rychlosti pohybu vozidla

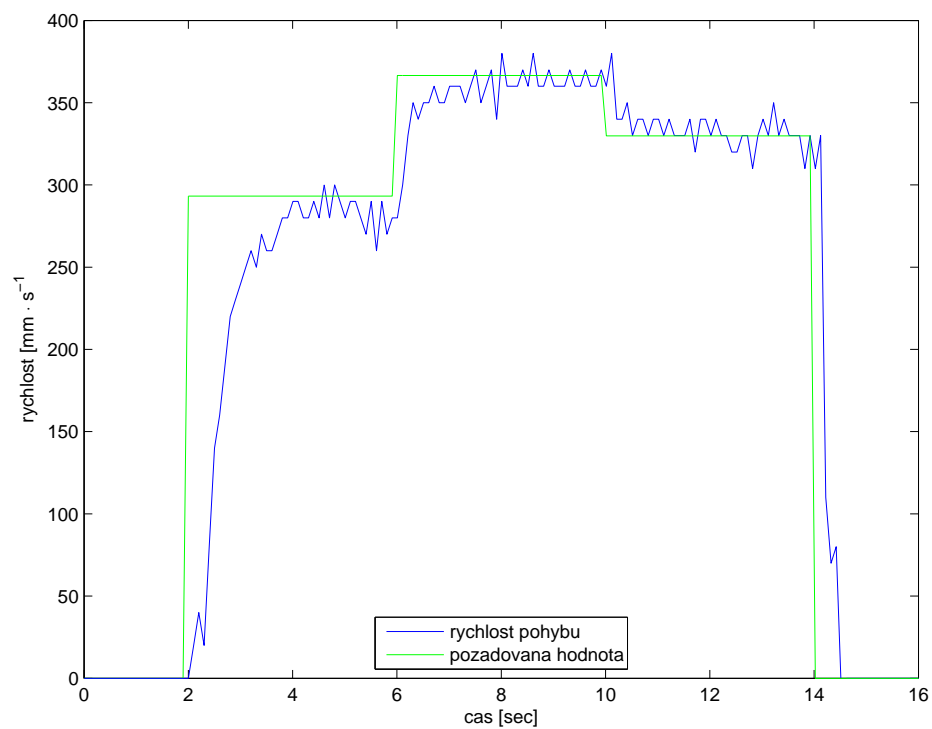
První velmi důležitou součástí řízení pohybu veškerých kolových vozidel je regulace rychlosti otáčení hnacích kol. K těmto účelům bylo použito diskretizovaného PD regulátoru, který byl navržen v rámci kapitoly 3.1. Chování výstupu uzvařené regulační smyčky, tedy tečné rychlosti otáčení kol *bez zátěže*, je znázorněno na následujícím obrázku.



Obrázek 26: Regulace rychlosti otáčení kol robotu

Z průběhu sledované veličiny lze pozorovat určitá odchylna od požadované hodnoty v ustáleném stavu a drobná kmitavost. Tyto jevy jsou zapříčiněny jednak způsobem měření výstupu (rychlosti), které má omezenou přesnost a pro zvolenou periodu vzorkování $50ms$ je diskrétní ve své hodnotě s krokem $10mm \cdot s^{-1}$, a dále pak omezenou přesností výpočtů s čísly v pohyblivé řádové čárce na použitém osmibitovém procesoru.

Velmi významným faktorem v oblasti řízení pohybu mechanických soustav je však působení třecí síly. Na následujícím obrázku je proto ilustrováno reálné chování robotu při skokových změnách referenční rychlosti během jeho pohybu po testovací dráze. Vliv tření je viditelný na první pohled při porovnání doby regulace na Obrázku 26 a na Obrázku 27, na němž je také patrné nerovnoměrné tření z důvodu střídavého smýkání a odvalování opěrné kuličky po podložce, která navíc není dokonale rovná a hladká.

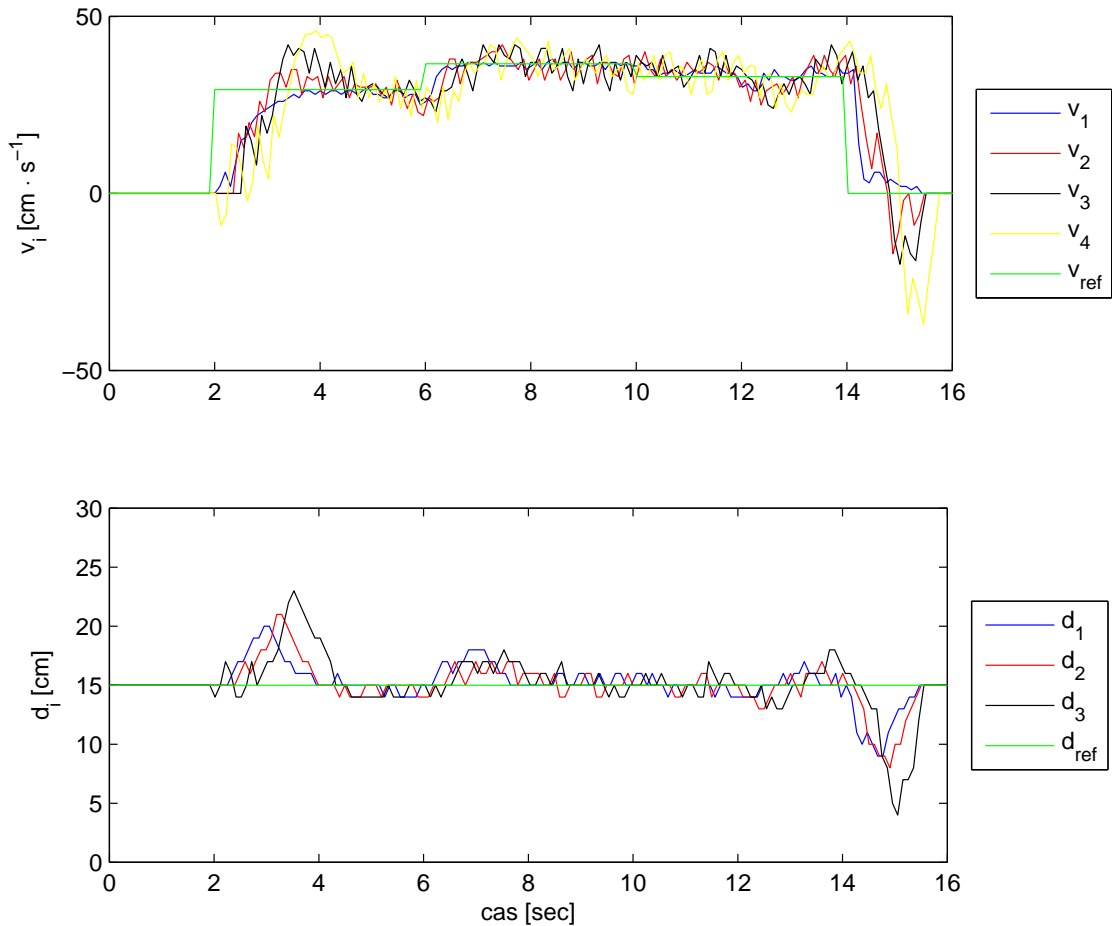


Obrázek 27: Regulace rychlosti pohybu robotu

Navržený regulátor však vykazuje uspokojivé chování i pro takto ovlivňovaný systém, a proto je možné nyní přistoupit k finálnímu experimentu, kterým je řízení lineární kolony reálných robotů.

6.2 Řízení lineární kolony vozidel

Finálním krokem této práce bylo experimentální ověření řídicího algoritmu navrženého v kapitole 3.2, který byl ve své diskrétní podobě implementován na zvoleném mikrokontroléru Arduino Uno. Sledovanou řídicí veličinou byla tentokrát vzdálenost od předchůdce d , jejíž požadovaná hodnota byla nastavena na hodnotu $d_{ref} = 15\text{ cm}$. Chování takto řízené kolony 4 robotů je ilustrováno na Obrázku 28.



Obrázek 28: Řízení lineární kolony

Na Obrázku 28 lze znovu pozorovat určité kmitavé chování, které lze přičítat jednak již zmiňovanému nerovnoměrnému tření na dráze, ale dále také rozlišovací schopností použitých senzorů, které mají výrobcem deklarovanou přesnost na 2 cm a poskytují hodnoty zaokrouhlené na celé centimetry. Tyto dva faktory lze tedy považovat za hlavní zdroje jistých nepřesností v chování experimentálního systému, který však z celkového hlediska vykazuje požadované chování.

7 Závěr

Předmětem této práce byla komplexní analýza řízení zřetězené řady mobilních robotů od vytvoření matematických modelů přes návrh regulačních algoritmů a jejich simulačního ověření až po návrh experimentální platformy a otestování řídicích algoritmů na reálných vozidlech.

V první fázi byly vytvořeny matematické modely potřebné k návrhu řídicích algoritmů. K jejich určení byly využity znalosti a dovednosti získané v úvodních kurzech matematické analýzy a fyziky na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Dalším krokem bylo navržení řídicích algoritmů k odvozeným matematickým modelům, k čemuž již bylo využito znalostí z oboru teorie systémů. Řídicí algoritmy byly v příslušné kapitole navrženy pro všechny vrstvy řízeného systému, tedy od regulátoru otáček kol robotu, přes řízení pohybu celého robotu po zvolené referenční trajektorii až po řízení skupiny robotů za účelem vytvoření zřetězené řady, nebo-li kolony.

V následující kapitole byly prezentovány výsledky simulací, jimž byla během vytváření této práce věnována značná pozornost. V této části bylo nejprve ilustrováno chování samotného modelu diferenciálně řízeného robotu při přímovazebním řízení a následně byly ověřeny všechny navržené řídicí algoritmy, tedy regulátor sledování referenční trajektorie, algoritmus řízení pohybu lineární kolony vozidel po přímce i obě varianty řídicích algoritmů pro pohyb kolony v rovině po libovolné trajektorii.

Významnou součástí této práce pak byl návrh experimentální platformy, který obnášel volbu a sestavení šasi robotu, na které byly následně montovány potřebné řídicí, měřicí a komunikační prvky. Zvolenou řídicí jednotkou byl mikrokontrolér Arduino Uno, k němuž byly následně voleny potřebné komponenty. Na základě požadavků na přesnost a kvalitu měření byly vybrány jednotlivé senzory, k nimž byla následně implementována aplikační vrstva ve vývojovém prostředí Arduino pro účely komunikace se zvolenou řídicí jednotkou. Dále pak byla implementována a realizována bezdrátová síť pro přenos diagnostický dat z jednotlivých řídicích jednotek do stolního počítače.

Poslední část této práce pak byla věnována prezentaci výsledků laboratorních experimentů, které byly prováděny s vytvořenými roboty. Při těchto experimentech byla ověřena funkčnost navržených algoritmů na reálném zařízení. Výsledky vykazovaly z důvodu tření a dalších negativních vlivů prostředí i vlastností užitých senzorů určité nepřesnosti, které však jsou akceptovatelné.

V rámci budoucí práce je plánováno navržení filtračních algoritmů za účelem zpřesnění měření a minimalizace šumu, dále pak práce na optimalizaci algoritmu řízení kolony v rovině a jeho rozšíření např. o vyhýbání se překážkám apod. V rámci posunu směrem k reálnému provozu a nasazení v běžných vozidlech by rovněž bylo vhodné přejít od malé laboratorní platformy s řídicí jednotkou s omezeným výpočetním výkonem na rozměrnější vozidla, která by byla schopná nést dostatečně přesná čidla (např. stereo kameru) pro měření polohy sledovaného vozidla a výpočetní jednotku schopnou takto získaná rozsáhlejší data zpracovat.

Použitá literatura

- [1] Složení vozového parku v ČR <http://www.autosap.cz/sfiles/a1-9.htm>
- [2] Sypták, M. (2009). Modelování a řízení kolových robotů.
- [3] De Luca, A., Oriolo, G., Vendittelli, M. (2004). Control of Wheeled Mobile Robots: An Experimental Overview.
- [4] Liang, K. Y. (2011). Linear quadratic control for heavy duty vehicle platooning.
- [5] Klančar, G., Matko, D., Blažič, S. (2008). Wheeled Mobile Robots Control in a Linear Platoon. Journal of Intelligent and Robotic Systems.
- [6] Melichar, J. (2011), Lineární systémy: Učební text.
- [7] Webové stránky projektu Arduino, <http://www.arduino.cc/>
- [8] Webová prezentace jednotky Pololu DRV8835, <http://www.pololu.com/product/2511>
- [9] Fu-Ching, Y., Wen-Kai, H., Jing-Kun, Z., Ing-Jer, H. (2008). Automatic Verification of External Interrupt Behaviors for Microprocessor Design.
- [10] Datasheet jednotky Devantech SRF05 <http://www.robot-electronics.co.uk/htm/srf05tech.htm>
- [11] Gehrig, S. K., Stein, F. J. (1998), A trajectory-based approach for the lateral control of car following systems.