

Západočeská univerzita v Plzni

Fakulta aplikovaných věd

Katedra kybernetiky

DIPLOMOVÁ PRÁCE

PLZEŇ, 2013

TOMÁŠ ČECHURA

Prohlášení

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

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

V Plzni dne

.....

Tomáš Čechura

Poděkování

Tímto bych chtěl poděkovat vedoucímu bakalářské práce Prof. Ing. Milošovi Schlegelovi, CSc. za metodické vedení a věcné připomínky při zpracování této práce. Velké poděkování patří také celé mojí rodině za podporu během celého studia, a zvláště pak mé přítelkyni, za neustálý tvůrčí dohled.

Anotace

Tato diplomová práce se zabývá vývojem nové koncepce mobilního robotického manipulátoru určeného pro nedestruktivní testování svarů potrubí v jaderných elektrárnách. Nejprve je odvozen matematický model a simulační model v prostředí MATLAB / Simulink / SimMechanics. Model je následně řízen s využitím vizuální zpětné vazby. Vlastnosti řídicího algoritmu jsou testovány na sadě trajektorií, která prověřuje jeho robustnost. Nakonec je zmíněna praktická ukázka regulace podtlaku pomocí řídicího systému REX na reálném modelu podtlakové komory.

Klíčová slova

vizuální zpětná vazba, matematický model, Lagrangeova metoda, řízení podtlaku, NDT, mobilní robot, osculační kružnice, poloměr křivosti

Annotation

This Thesis deals with the development of the mobile robotic manipulator's new conception designated for non-destructive testing of pipelines welds in the nuclear power station. First of all, the mathematical and the simulation models are derived in the MATLAB/Simulink/SimMechanics setting. Afterwards, the model is controlled by the visual feedback use. The controlling algorithm qualities are tested on the set of trajectories, which verifies its robustness. After all, the practical demonstration of the negative pressure regulation, provided by controlling system REX on the real model of the vacuum chamber, is mentioned.

Key words

visual feedback, mathematical model, Lagrange method, control of negative pressure, NDT, mobile robot, osculating circle, radius of curvature

Obsah

1	Úvod	1
1.1	Současné manipulátory pro NDT	2
1.2	Přehled technologií pro přilnavost robotů	3
1.3	Konstrukce robotu	5
2	Matematický model	6
2.1	Kinematický model	6
2.2	Dynamický model	9
2.2.1	Celková kinetická energie robotu	9
2.2.2	Lagrangeova metoda odvození pohybových rovnic	9
2.2.3	Aplikace Lagrangeovy metody	10
2.2.4	Stavový popis	11
3	Model v prostředí Simulink/SimMechanics	13
4	Regulace úhlových rychlostí kol robotu	15
4.1	Úvod do problému	15
4.2	Rozvazbující regulátor	15
4.3	Regulace úhlových rychlostí kol	17
4.4	Verifikace odvozených modelů	18
5	Vizuální zpětná vazba	20
5.1	Obrazové zpracování trajektorie	21
5.2	Uzavření vizuální zpětné vazby	23
6	Metody určení požadovaného poloměru R^*	26
6.1	Řízení s využitím oskulačních kružnic	26
6.1.1	Zhodnocení metody	27
6.2	Řízení s využitím oskulačních kružnic s korekcí odchylky	30
6.3	Kritérium kvality řízení	32
6.4	Metoda korekce natočení robotu	33
7	Komplexní testování vizuální zpětné vazby	37
7.1	Sada testovacích trajektorií	37
7.2	Trajektorie linie - výsledky	39
7.3	Trajektorie oblouk - výsledky	40
7.4	Trajektorie vlny - výsledky	41
7.5	Shrnutí testování	42

8	Regulace podtlaku v reálném modelu	43
8.1	Motivace	43
8.2	Reálný model	43
8.3	Implementace řídicího systému	44
8.4	Zhodnocení regulace podtlaku	47
9	Závěr	48
	Seznam obrázků	49
	Reference	50
	Příloha	52
A	Skript předzpracování obrazu	52

1 Úvod

Tato práce si klade za úkol navrhnout zcela novou koncepci robotického manipulátoru pro nedestruktivní testování svarů (NDT) s velkým důrazem kladeným na řízení pomocí vizuální zpětné vazby. Robot bude určen pro diagnostické účely svarů potrubí v jaderných elektrárnách (JE), kde dochází k jejich periodickému testování. Zvláště v jaderném průmyslu je více než žádoucí, aby případné defekty byly odhaleny ještě před případnou havárií. Technologie NDT také pomáhá prodlužovat životnost JE nad rámec jejich plánované doby provozu.

Další motivací k vývoji nového konceptu mobilního robotického manipulátoru je snaha co nejvíce zkrátit dobu provádění NDT svarů v JE. Tyto kontroly probíhají většinou při plánovaných odstávkách JE v omezeném čase, proto urychlení kontroly svarů potrubí může mít i obrovský ekonomický dopad.

Koncept robota musí splňovat celou řadu požadavků. Mezi nejdůležitější patří schopnost pohybu po potrubí. Předpokládejme, že většina cílových potrubí je vyrobena z nerezové oceli, která není obvykle feromagnetická. Z tohoto pohledu můžeme danou úlohu přibližně převést na úlohu návrhu robotu, který bude schopný pohybu po předem nakreslené trajektorii po vertikální stěně místnosti či po jiném neferomagnetickém materiálu ve vertikální poloze.

Dalším požadavkem je určitá nosnost robotu, protože jeho tělo ponese pravděpodobně ultrazvukovou měřicí sondu či jiné diagnostické zařízení. Cílem této práce však není navrhnout použitou technologii pro samotné testování svarů, ale mobilní zařízení, které tuto technologii ponese. Důraz při vývoji manipulátoru bude také kladen na jeho rychlost pohybu tak, aby došlo ke zlepšení oproti současným konvenčním průmyslovým skenerům.

Návrhem koncepce mobilního robotu se v této diplomové práci rozumí provedení vhodné volby lokomočního ústrojí robotu společně s odvozením a analýzou jeho simulačního modelu. Vyvinutý simulační model bude následně použit pro návrh řídicího systému modelu robotu, který bude zabezpečovat sledování požadované trajektorie pomocí vizuální zpětné vazby. V závěru práce se budeme věnovat praktickému návrhu a reálné ukázce regulace vyvíjeného podtlaku.

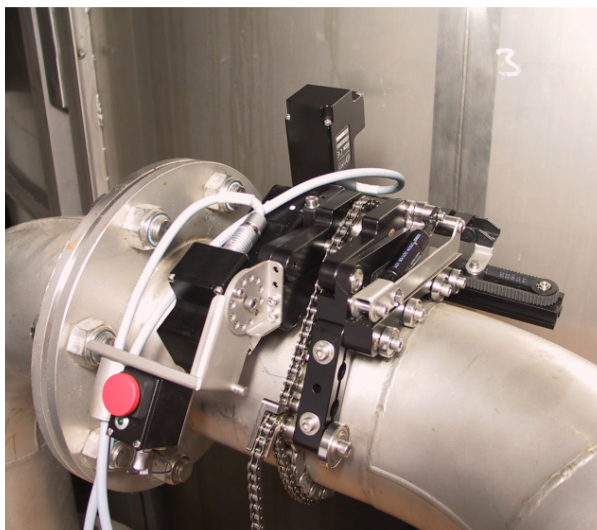
1.1 Současné manipulátory pro NDT

V průmyslových aplikacích se stále běžně setkáváme s použitím ručních defektoskopů. Při kontrole svarů v JE je kladen důraz na jejich automatizaci, aby se co nejvíce omezil vliv lidského faktoru na výsledky defektoskopické zkoušky svarů.

V současné době se za automatický průmyslový skener pro NDT považuje mobilní robot s připevněnou diagnostickou jednotkou, který je operátorem posazen na požadované místo potrubí v blízkosti testovaného svaru. Manipulátor pak dle pokynů operátora nebo automaticky provede diagnostiku svaru danou technologií a poskytne veškerá data k následnému vyhodnocení.

Technologií zajišťujících připevnění robotického manipulátoru k potrubí se obecně používá několik. Mezi hlavní patří přichycení pomocí řetězů či jiných pásů omotaných kolem potrubí nebo použití magnetických kol, pokud to materiál potrubí dovoluje. Mezi méně časté způsoby patří různé přísavky, protože jejich použití v průmyslovém prostředí je značně ovlivněno všudypřítomnými nečistotami a prachem.

Pro ilustraci uvádíme na obrázku 1 příklad dvou automatických průmyslových skenerů NDT používaných v současné praxi. Jedná se o skener *APS – 6* [1] využívající napnutý řetěz kolem potrubí od firmy *ForceTechnology*. Technologii magnetických kol naopak používá robot s označením *WeldROVER* [2] od firmy *Olympus*.



(a) Force Technology APS-6 [1]



(b) Olympus WeldROVER [2]

Obrázek 1: Ukázka současných průmyslových skenerů potrubních svarů

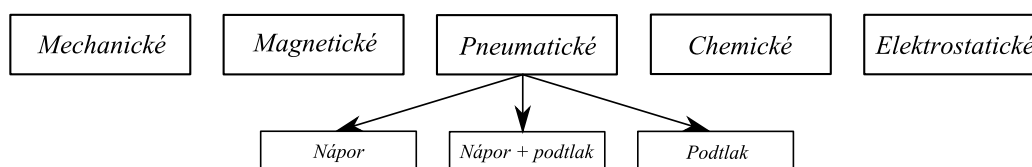
Každý průmyslový robot má svá omezení a podmínky, za kterých lze

použít, i když se výrobci vždy snaží, aby byl daný diagnostický manipulátor co nejvíce variabilní. Mezi omezení patří především rozsah průměru potrubí, na kterém můžeme defektoskopické kontroly provádět. Dalšími důležitými ukazateli jsou rozměry robotu a lineární rychlost, při které může probíhat snímání svaru.

1.2 Přehled technologií pro přilnavost robotů

Technologií, které zabezpečují různé formy přitlačné síly, je celá řada. Přilnavost je stěžejní pro pohyb robotu po vertikální podložce. Musí být natolik silná, aby vyrušila účinky gravitační síly a umožnila tak robotu jeho požadovaný pohyb.

Technologie rozdělíme podle účinků přitlačné síly, jak je uvedeno na obrázku 2. Každou samostatně popíšeme a uvedeme vhodný příklad použití.



Obrázek 2: Přehled technologií pro přilnavost robotů

Mechanická adheze obecně využívá struktury a tvaru povrchu, případně dovede vhodným způsobem povrch měnit. Příkladem může být *SpinybotII* [3], který je opatřen množstvím mikroskopických háčků. Háčky se zachycují za drobné nerovnosti povrchu a robot je tak schopen šplhat např. po betonu, cihlové zdi či omítce.

Rozšířenějším přístupem je bezpochyby využití magnetismu. V dnešní době existuje mnoho sériově vyráběných robotických manipulátorů s magnetickými kolečky, stejně jako již zmíněný *OlympusWeldROVER* (obrázek 1b). Toto řešení je velmi účinné a spolehlivé, avšak použitelné pouze na ferromagnetické povrchy.

Pneumatické účinky přitlačné síly vznikají rozdílem tlaků. Tento jev využívá další velmi rozšířená skupina šplhajících robotů. Skupinu můžeme dále rozdělit na roboty využívající podtlak, nápor či jejich kombinací.

Prvním příkladem použití pneumatické adheze bylo použití klasických přísavek upravených pro úchop - dnes se běžně používají v potravinářském průmyslu např. k oddělování plátků. Přísavky mohou být ale použity pouze v režimu zapnuto/vypnuto - roboty využívající přísavky jsou obecně krácející. Naproti tomu stojí podtlakové komory s vhodným těsněním mezi komorou a povrchem, aby byl zajištěn volný pohyb robotu stálým únikem podtlaku.

Příkladem může být robot *Alicia3* [4]. Tento princip bychom rádi použili jako základní a dále ho vylepšili o regulaci podtlaku a tím se vyhnuli složitému návrhu vhodného těsnění.

Nápor působící proti gravitační síle působící na robot vznikne roztočením vrtule, která je umístěna na těle robotu. Robot je tak pevně přitisknut k vertikální podložce a může se po ní volně pohybovat. Koncept šplhajícího robotu byl už dříve postaven a otestován v Japonsku [5]. Analýzou použití kombinace podtlaku a náporu se zabývá článek [6].

V počátcích použití chemického způsobu přichycení robotu k vertikální podložce se používala lepicí páska. Po několika málo přichycení však došlo k nenávratné ztrátě potřebné přitlačné síly nevyhnutelnou přítomností prachu a jiných nečistot. Robot *StickyBot* [7] se inspiroval přilnavými polštářky, které mají na svých prstech gekoni. Jedná se o kombinaci chemického a mechanického přístupu.

Princip elektrostatické adheze byl poprvé přenesen do praxe s použitím na šplhajících robotech institut *SRIinternational*. Praktické výsledky uvedené v práci [8] dosahují vynikajících výsledků. Hodnoty přitlačných sil a udávaná spotřeba elektrické energie jsou velice slibné. Zatím se nám bohužel nepodařilo k této technologii blíže dostat. Pro ilustraci uvádíme alespoň příklad použití jimi vyvinuté elektroadhezní podložky (viz obrázek 3). Podložka je umístěna na vertikální dřevěné desce. Závaží, které podložka drží, váží přibližně 34 kg (75 lbs). Tato technologie má obsáhlé možnosti využití nejen pro šplhající roboty.



Obrázek 3: Demonstrace použití elektroadheze

1.3 Konstrukce robotu

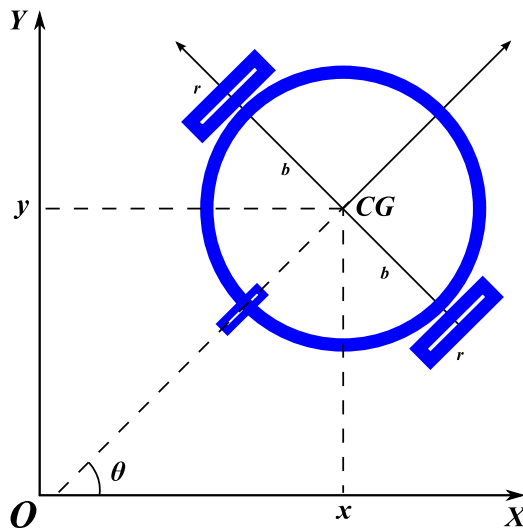
V počátku je nutné přesně a konkrétně definovat robot uvažovaný v této práci.

Z dostupných technologií, které umožňují pohyb po vertikálním neferomagnetickém materiálu, jsme vybrali princip podtlakového přisávání. Podvozek robotu je diferenciálně řízený s dvěma poháněnými koly a jedním podpurným. Výhodou tohoto podvozku je jeho konstrukční jednoduchost a zároveň velká obratnost a variabilita. Sledování trajektorie bude řízeno s využitím vizuální zpětné vazby uzavřené přes kameru. Kamera bude vhodným způsobem umístěna na robotu a v každém kroku simulace poskytne snímek prostoru před robotem společně s částí sledované trajektorie.

Velký důraz je kladen na přesnou specifikaci podvozku robotu, jehož kompletní analýza a zpětnovazební řízení budou odvozeny v teoretické části této práce.

Jedná se o mobilní robot se třemi koly - dvě z nich leží v ose robotu, každé na jedné straně a jsou nezávisle poháněna. Třetí kolo je pouze podpurné, jeho úkolem je zajišťovat stabilitu robotu a v podstatě nijak neomezuje jeho pohyblivost. Model v tomto případě idealizujeme a neuvažujeme žádné dynamické účinky podpurného kola - zanedbáváme tření a moment setrvačnosti.

Na obrázku 4 je schématicky znázorněna konstrukce robotu. Předpokládejme některé hlavní parametry robotu. Těžiště robotu (CG) se nachází v průsečíku os souměrnosti robotu. Poháněná kola o poloměru r jsou souměrně umístěna ve vzdálenosti b od těžiště. Natočení robotu v původním souřadném systému je dáno úhlem θ . Jeho poloha je pak vyjádřena souřadnicemi x a y .



Obrázek 4: Schéma uvažovaného mobilního robotu

2 Matematický model

Pro potřeby simulace bude nutné odvodit odpovídající matematický model ideálního mobilního robotu s nezávislým pohonem dvou kol v rovině. Obdobným tématem se zabývá práce [9]. Problém návrhu můžeme rozdělit na dvě části - kinematický a dynamický popis robotu.

V kinematické části se budeme zabývat pouze prostorovými vlastnostmi robotu, které jsou dány jeho vlastní geometrickou konstrukcí a konfigurací podvozku bez ohledu na hmotnosti jednotlivých částí robotu či jejich dynamické vlastnosti. Uvažujeme tedy tvar robotu, jeho polohu a rychlost.

Naproti tomu stojí dynamický popis, který popisuje chování robotu v důsledku silového působení. K tomuto popisu musíme bezpodmínečně znát hmotnosti jednotlivých částí robotu a jejich momenty setrvačnosti.

Uvažujme následující kinematická omezení: Pohyb robotu je možný pouze ve směru osy souměrnosti. Poháněná kola neprokluzují při styku s podložkou.

2.1 Kinematický model

Základem pro kinematický model bude vztah mezi pohybem těžiště robotu a úhlovými rychlostmi levého a pravého kola. Rychlosti kol označme po řadě $\dot{\theta}_L$ a $\dot{\theta}_R$.

Z předpokladu neprokluzu kol můžeme pomocí geometrických vztahů odvodit vzorce pro výpočet jednotlivých komponent rychlosti těžiště robotu:

$$\begin{aligned}\dot{x} &= \frac{r}{2}(\dot{\theta}_R + \dot{\theta}_L)\cos\theta \\ \dot{y} &= \frac{r}{2}(\dot{\theta}_R - \dot{\theta}_L)\sin\theta,\end{aligned}\tag{2.1}$$

kde \dot{x} a \dot{y} jsou rychlosti těžiště robotu ve směru osy x resp. y .

Dále bychom rádi znali vztah mezi rychlostmi jednotlivých poháněných kol a úhlovou rychlostí otáčení robotu kolem své osy. Obrázek 5 ukazuje geometrický význam obvodových rychlostí kol robotu. Z podobnosti trojúhelníků můžeme psát vztah pro úhlovou rychlost otáčení robotu:

$$\dot{\theta} = \frac{v_R - v_L}{2b},\tag{2.2}$$

kde v_R a v_L jsou obvodové rychlosti kol robotu.

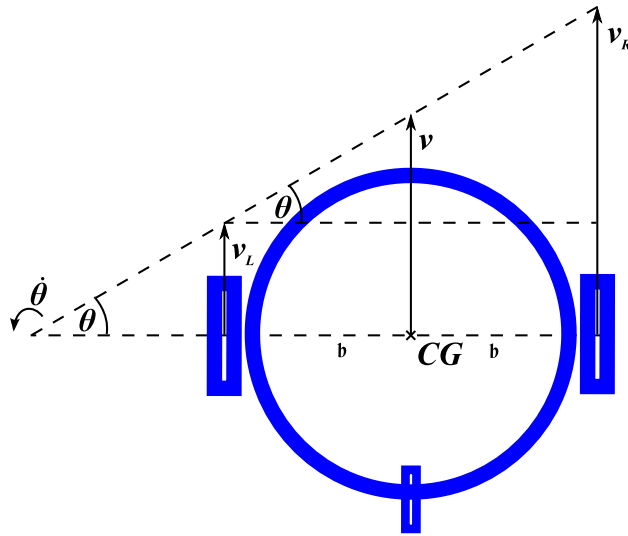
Vztah mezi obvodovou a úhlovou rychlostí rotačního pohybu kol je:

$$v_L = r\dot{\theta}_L; \quad v_R = r\dot{\theta}_R\tag{2.3}$$

2 MATEMATICKÝ MODEL

Dosazením rovnic 2.3 do vztahu pro úhlovou rychlost otáčení robotu 2.2 a elementárními úpravami dostaneme vztah, vyjadřující závislost úhlové rychlosti otáčení robotu na úhlových rychlostech kol:

$$\dot{\theta} = \frac{r}{2b}(\dot{\theta}_R - \dot{\theta}_L) \quad (2.4)$$



Obrázek 5: Geometrická interpretace obvodových rychlostí kol

Poznámka 2.1 Pokud by z různých konstrukčních či jiných důvodů nebylo těžiště robotu umístěno ve středu otáčení, bylo by nutné přetransformovat rychlosti středu osy kol \dot{x} a \dot{y} na rychlosti těžiště například pomocí následující transformace:

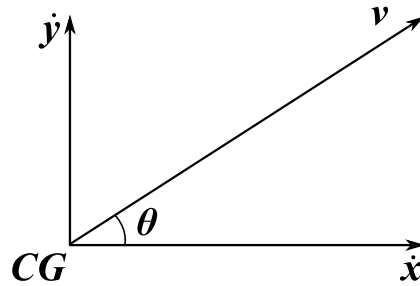
$$\dot{x}_c = \dot{x} + d\dot{\theta} \sin \theta \quad (2.5)$$

$$\dot{y}_c = \dot{y} - d\dot{\theta} \cos \theta,$$

kde parametr d je vzdálenost středu otáčení robotu a těžiště. Stále však uvažujeme souměrnost robotu, tudíž těžiště musí ležet na ose souměrnosti.

Z hlediska budoucího řízení modelu nás bude také zajímat vztah pro výpočet lineární rychlosti robotu. Ten určíme použitím Pythagorovy věty dle obrázku 6:

$$v^2 = \sqrt{\dot{x}^2 + \dot{y}^2} \quad (2.6)$$



Obrázek 6: Geometrické znázornění lineární rychlosti robotu

Po dosazení za \dot{x} a \dot{y} a upravení pomocí goniometrických funkcí a následném odmocnění dostaneme vztah:

$$v = \frac{r}{2}(\dot{\theta}_R + \dot{\theta}_L) \quad (2.7)$$

Vyjádřením úhlových rychlostí obou kol ze vztahů pro lineární rychlost robotu a jeho úhlovou rychlost otáčení dostaneme tyto rovnice:

$$\begin{aligned} \dot{\theta}_R &= \frac{1}{r}v + \frac{b}{r}\dot{\theta} \\ \dot{\theta}_L &= \frac{1}{r}v - \frac{b}{r}\dot{\theta} \end{aligned} \quad (2.8)$$

Kinematický model tedy můžeme shrnout do následující soustavy rovnic:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta}_R \\ \dot{\theta}_L \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ \frac{1}{r} & \frac{b}{r} \\ \frac{1}{r} & -\frac{b}{r} \end{bmatrix} \begin{bmatrix} v \\ \dot{\theta} \end{bmatrix} \quad (2.9)$$

Poznámka 2.2 *Tento model nám říká, jak se bude měnit rychlost robotu v rovině a úhlové rychlosti jeho kol v závislosti na zvolené lineární rychlosti a úhlové rychlosti otáčení robotu. S použitím integrace můžeme získat polohu robotu a úhly natočení jednotlivých kol.*

2.2 Dynamický model

Dynamické pohybové rovnice odvodíme pomocí Lagrangeovy metody. Pro její následnou aplikaci je nutná znalost celkové kinetické a potenciální energie mobilního robotu.

Potenciální energie je rovna nule, neboť předpokládáme rovinný pohyb robotu.

2.2.1 Celková kinetická energie robotu

Celkovou kinetickou energii robotu můžeme rozdělit na tyto složky: translační kinetická energie robotu T_t , rotační kinetická energie robotu T_r a rotační kinetická energie poháněných kol T_w :

$$T = T_t + T_r + T_w \quad (2.10)$$

Uvedené kinetické energie můžeme vyjádřit takto:

$$T_t = \frac{1}{2}mv^2 \quad (2.11)$$

$$T_r = \frac{1}{2}J\dot{\theta}^2 \quad (2.12)$$

$$T_w = \frac{1}{2}J_w\dot{\theta}_R^2 + \frac{1}{2}J_w\dot{\theta}_L^2, \quad (2.13)$$

kde m je hmotnost celého robotu, J je moment setrvačnosti celého robotu vzhledem k ose otáčení a J_w je moment setrvačnosti kola.

2.2.2 Lagrangeova metoda odvození pohybových rovnic

Lagrangeova metoda využívá pro sestavení pohybových rovnic celkovou kinetickou a potenciální energii soustavy a všechny působící momenty sil. Předpis pro Lagrangeovy rovnice je následující:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \left(\frac{\partial L}{\partial q_i} \right) = Q_i, \quad (2.14)$$

kde $1 \leq i \leq n$, $L = T - V$ je Lagrangian, T - kinetická energie vzhledem k inerciální soustavě, V - potenciální energie, n - počet stupňů volnosti, q_i - zobecněné souřadnice, Q_i - zobecněné síly

2.2.3 Aplikace Lagrangeovy metody

Lagrangián je definován jako rozdíl celkové kinetické a celkové potenciální energie. V našem případě je tedy Lagrangián přímo roven celkové kinetické energii.

$$L = T \quad (2.15)$$

Nyní máme vše připravené na použití Lagrangeovy metody. Jako zobecněné souřadnice uvažujeme úhly natočení poháněných kol robotu θ_L a θ_R .

Předpis pohybových rovnic podle Lagrangeovy metody je následující:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_R} \right) - \left(\frac{\partial L}{\partial \theta_R} \right) = M_R \quad (2.16)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_L} \right) - \left(\frac{\partial L}{\partial \theta_L} \right) = M_L, \quad (2.17)$$

kde M_R a M_L jsou momenty působící na pravé, resp. levé poháněné kolo robotu.

Dosazením a matematickými úpravami získáme upravené Lagrangeovy rovnice pro působící moment na pravé resp. levé kolo:

$$M_R = \frac{1}{8} \frac{\left(2 (mb^2 + J) \ddot{\theta}_R + 2 (mb^2 - J) \ddot{\theta}_L \right) r^2 + 8 J_w b^2 \ddot{\theta}_R}{b^2} \quad (2.18)$$

$$M_L = \frac{1}{8} \frac{\left(2 (mb^2 - J) \ddot{\theta}_R + 2 (mb^2 + J) \ddot{\theta}_L \right) r^2 + 8 J_w b^2 \ddot{\theta}_L}{b^2} \quad (2.19)$$

Pohybové rovnice (2.18) a (2.19) jsou funkcí pouze druhých derivací natočení poháněných kol θ_R a θ_L . Dostáváme tedy lineární dynamický popis robotu a nebude třeba jeho linearizace.

Vyřešením těchto rovnic vzhledem ke druhým derivacím natočení poháněných kol $\ddot{\theta}_R$ a $\ddot{\theta}_L$ získáváme následující rovnice:

$$\ddot{\theta}_R = \frac{(J r^2 - b^2 r^2 m) M_L + (4 J_w b^2 + b^2 r^2 m + J r^2) M_R}{2 b^2 r^2 m J_w + 4 b^2 J_w^2 + J r^4 m + 2 J r^2 J_w} \quad (2.20)$$

$$\ddot{\theta}_L = \frac{(J r^2 - b^2 r^2 m) M_R + (4 J_w b^2 + b^2 r^2 m + J r^2) M_L}{2 b^2 r^2 m J_w + 4 b^2 J_w^2 + J r^4 m + 2 J r^2 J_w} \quad (2.21)$$

2.2.4 Stavový popis

Úplný model mobilního robotu získáme spojením dříve odvozeného kinematického a dynamického modelu do jedné sady rovnic. Jako vektor stavu zvolíme:

$$\begin{aligned}
 x_1 &= \theta_R & \dot{x}_1 &= x_3 \\
 x_2 &= \theta_L & \dot{x}_2 &= x_4 \\
 x_3 &= \dot{\theta}_R & \dot{x}_3 &= \ddot{\theta}_R \quad (2.20) \\
 x_4 &= \dot{\theta}_L & \dot{x}_4 &= \ddot{\theta}_L \quad (2.21) \\
 x_5 &= \theta & \dot{x}_5 &= \frac{r}{2b}x_3 - \frac{r}{2b}x_4 \\
 x_6 &= s & \dot{x}_6 &= \frac{r}{2}x_3 + \frac{r}{2}x_4 \\
 x_7 &= x & \dot{x}_7 &= \dot{x}_6 \cos x_5 \\
 x_8 &= y & \dot{x}_8 &= \dot{x}_6 \sin x_5
 \end{aligned} \tag{2.22}$$

Stavový popis 2.22 je na první pohled nelineární, protože derivace polohy robotu nejsou pouze lineární funkcí ostatních složek stavu. Z důvodu sestavení modelu v prostředí Simulink jsme rozdělili tento nelineární model na jeho lineární a nelineární část. Lineární část můžeme zapsat v podobě matic:

$$\begin{aligned}
 \dot{x} &= Ax + Bu \\
 y &= Cx,
 \end{aligned} \tag{2.23}$$

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 \\ b_{31} & b_{32} \\ b_{41} & b_{42} \end{bmatrix}, C = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, D = 0 \tag{2.24}$$

$$b_{31} = \frac{4 J_w b^2 + b^2 r^2 m + J r^2}{2 b^2 r^2 m J_w + 4 b^2 J_w^2 + J r^4 m + 2 J r^2 J_w} \tag{2.25}$$

$$b_{32} = \frac{J r^2 - b^2 r^2 m}{2 b^2 r^2 m J_w + 4 b^2 J_w^2 + J r^4 m + 2 J r^2 J_w} \tag{2.26}$$

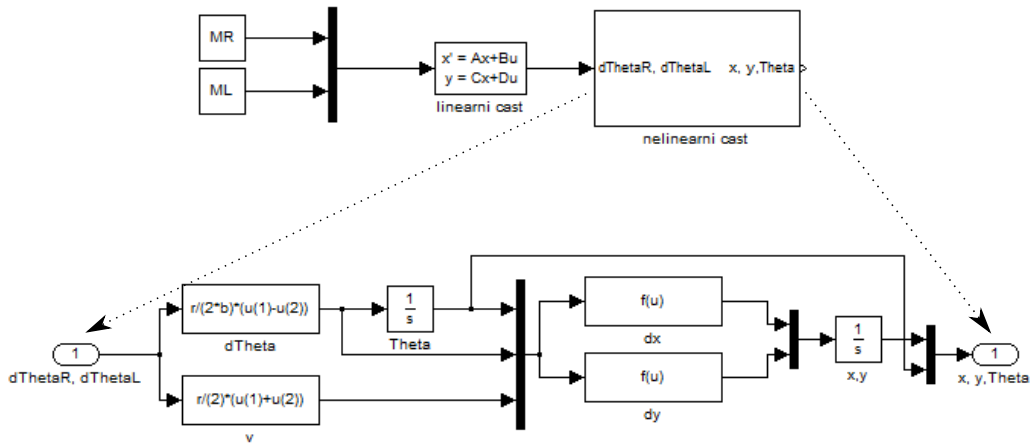
$$b_{41} = \frac{J r^2 - b^2 r^2 m}{2 b^2 r^2 m J_w + 4 b^2 J_w^2 + J r^4 m + 2 J r^2 J_w} \tag{2.27}$$

$$b_{42} = \frac{4 J_w b^2 + b^2 r^2 m + J r^2}{2 b^2 r^2 m J_w + 4 b^2 J_w^2 + J r^4 m + 2 J r^2 J_w} \tag{2.28}$$

2 MATEMATICKÝ MODEL

Obrázek 7 znázorňuje implementaci matematického modelu v prostředí Simulink. Jako vstup modelu jsou použity momenty působící na kola. Výstup stavového popisu jsou úhlové rychlosti kol, které jsou dále zpracovány v nelineární části. Ta se skládá pouze z již dříve odvozených rovnic. Výstupem modelu je pak poloha robotu a jeho natočení.

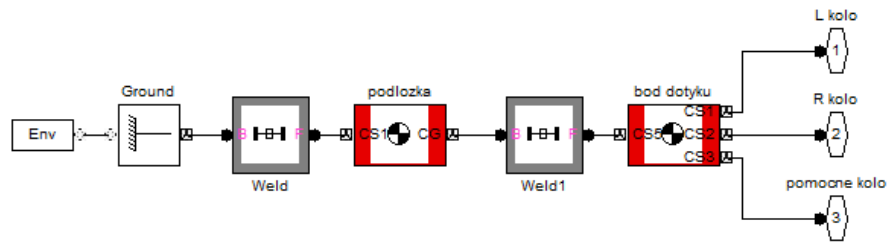
Poznámka 2.3 *Nelineární transformaci úhlových rychlostí kol na polohu a natočení robotu použijeme i pro odvození modelu v prostředí Simulink/SimMechanics.*



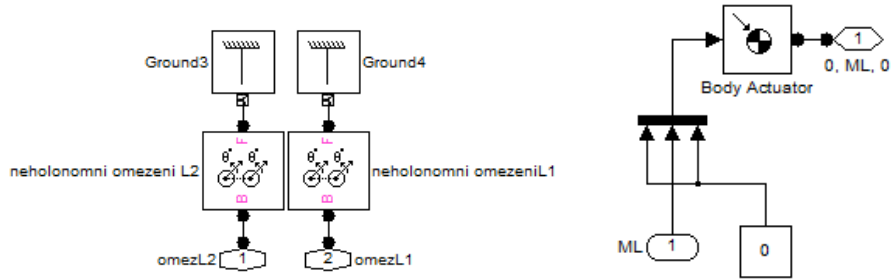
Obrázek 7: Matematický model v prostředí Simulink

3 MODEL V PROSTŘEDÍ SIMULINK/SIMMECHANICS

Jako vstup modelu slouží momenty působící na levé a pravé kolo M_L a M_R v ose otáčení kol. Výstupem jsou pak úhlové rychlosti jednotlivých kol $\dot{\theta}_L$ a $\dot{\theta}_R$. Ve spojení s již implementovanou nelineární částí matematického modelu tak dostáváme kompletní popis robotu s výstupem polohy a natočení.



(a) Subsystém podložky robotu

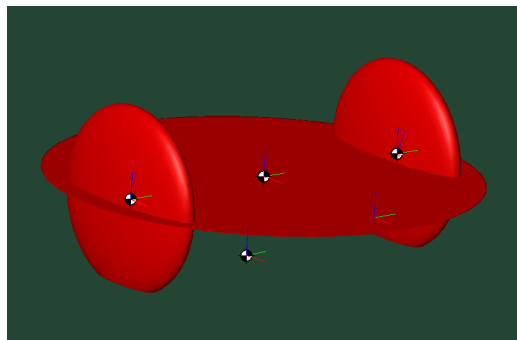


(b) Neholonomní omezení na levé kolo

(c) Momentové působení v ose rotace kola

Obrázek 9: Subsystémy hlavního modelu

Prostředí SimMechanics nabízí i různé formy vizualizace. Vizualizace znázorněná na obrázku 10 zobrazuje jednotlivá těla modelu jako ekvivalentní elipsoidy podle zadaných hmotností a momentů setrvačnosti.



Obrázek 10: Vizualizace modelu robotu v prostředí Simulink/SimMechanics

4 Regulace úhlových rychlostí kol robotu

4.1 Úvod do problému

Důvod, proč nás zajímá právě regulace úhlových rychlostí kol, je velice prostý. Robot s nezávisle poháněnou dvojicí kol se nativně pohybuje po kružnicích s proměnným poloměrem R . Pohyb do stran není možný z důvodu neholonomních omezení. Jak si dále ukážeme, poloměr zakřivení pohybu robotu R je závislý pouze na úhlových rychlostech poháněných kol robotu.

Poloměr R může obecně pro podvozky s diferenciálně řízenými koly nabývat čtyř různých stavů: R je nekonečný, pokud úhlové rychlosti kol budou totožné - robot pojedje rovně; R je nulový, pokud se budou úhlové rychlosti kol lišit pouze ve znaménku - robot se otáčí na místě kolem své osy; R je roven $2b$ (případně $-2b$), pokud jedno kolo stojí a druhé se pohybuje; R nabývá různých jiných hodnot pro ostatní případy. Z tohoto pohledu je velice výhodné využít k budoucímu řízení robotu právě tento poloměr R kružnice, po které se bude robot pohybovat.

Vztah pro výpočet poloměru R vychází z geometrické podstaty. Obvodové rychlosti kol jsou vzájemně rovnoběžné, dokonce jsou rovnoběžné i s celkovou lineární rychlostí robotu - to vyplývá z konstrukce robotu, kdy kola mají totožnou osu otáčení a těžiště robotu na ní dle předpokladu leží.

Z obecného vztahu mezi obvodovou a úhlovou rychlostí plyne:

$$R = \frac{v}{\dot{\theta}} \quad (4.1)$$

Po dosazení z rovnic (2.7) a (2.4):

$$R = b \frac{\dot{\theta}_R + \dot{\theta}_L}{\dot{\theta}_R - \dot{\theta}_L} \quad (4.2)$$

4.2 Rozvazující regulátor

Stavový popis robotu i jeho fyzikální podstata ukazují na vzájemné ovlivnění působících momentů. Jinými slovy - moment aplikovaný na dané kolo vyvolá změnu úhlové rychlosti na obou kolech zároveň. Robot je z tohoto pohledu souměrný, jak je patrné ze souměrnosti koeficientů matice B (2.25)-(2.28) vzhledem k levému a pravému kolu.

Navrhujeme tedy rozvazující regulátor pro uvažovaný TITO systém podle metody popsané v [10].

Nejprve musíme nalézt přenos systému od působících momentů M_L a M_R na úhlové rychlosti kol $\dot{\theta}_L$ a $\dot{\theta}_R$. Přenosovou funkci můžeme určit z již odvozeného stavového popisu (2.24) pomocí vztahu publikovaného např. v [12]:

$$F(s) = C(sI - A)^{-1}B + D \quad (4.3)$$

Po dosazení a úpravě:

$$F(s) = \begin{bmatrix} \frac{4J_w b^2 + b^2 r^2 m + J r^2}{(2b^2 r^2 m J_w + 4b^2 J_w^2 + J r^4 m + 2J r^2 J_w)s} & \frac{J r^2 - b^2 r^2 m}{(2b^2 r^2 m J_w + 4b^2 J_w^2 + J r^4 m + 2J r^2 J_w)s} \\ \frac{J r^2 - b^2 r^2 m}{(2b^2 r^2 m J_w + 4b^2 J_w^2 + J r^4 m + 2J r^2 J_w)s} & \frac{4J_w b^2 + b^2 r^2 m + J r^2}{(2b^2 r^2 m J_w + 4b^2 J_w^2 + J r^4 m + 2J r^2 J_w)s} \end{bmatrix} \quad (4.4)$$

V přenosové funkci vystupují pouze nenulové koeficienty matice B (2.25)-(2.28), proto můžeme zkráceně psát:

$$F(s) = \begin{bmatrix} \frac{b_{31}}{s} & \frac{b_{32}}{s} \\ \frac{b_{41}}{s} & \frac{b_{42}}{s} \end{bmatrix} \quad (4.5)$$

Algoritmus pro nalezení rozvazbovacího regulátoru popsany v [10]:

(1) Začneme s $K = I$, potom $D = adj(G)$, kde D je rozvazbovací matice (decoupler) a K je pomocná diagonální matice.

(2) Odstraňme největší společné dopravní zpoždění z každého sloupce matice D vynásobením odpovídajících diagonálních členů v matici K inverzním dopravním zpožděním.

(3) Odstraňme společné nuly a póly sloupců matice D vynásobením odpovídajících diagonálních členů matice K jejich inverzemi.

(4) $D = adj(G) \cdot K$

Pomocná matice K má v našem případě tvar:

$$K(s) = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} \quad (4.6)$$

Rozvazbovací matici D dopočteme:

$$D = \begin{bmatrix} b_{42} & -b_{41} \\ -b_{32} & b_{31} \end{bmatrix} \quad (4.7)$$

Přenosová funkce rozvazbené smyčky je vyjádřena jako součin $G \cdot D$:

$$G \cdot D = \begin{bmatrix} \frac{b_{31}b_{42} - b_{32}b_{41}}{s} & 0 \\ 0 & \frac{b_{31}b_{42} - b_{41}b_{42}}{s} \end{bmatrix} \quad (4.8)$$

Výsledná přenosová funkce (4.8) je diagonální, což zabezpečí správné rozvazbení původního systému.

4.3 Regulace úhlových rychlostí kol

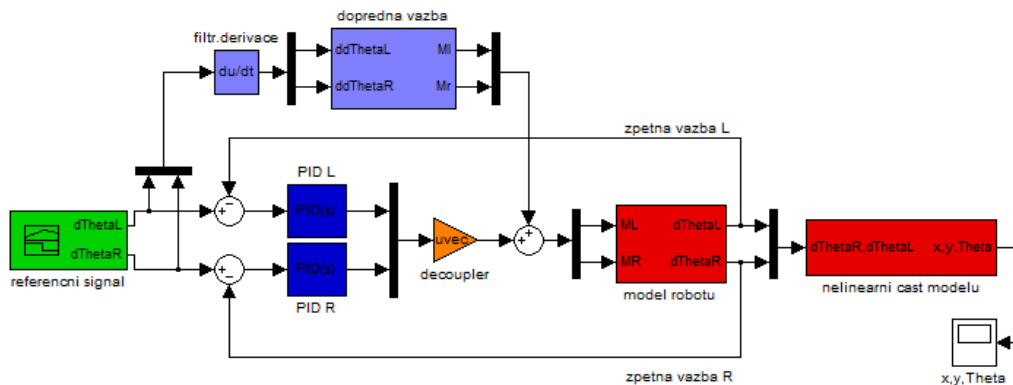
Po odstranění vazeb v modelu robotu můžeme přistoupit k návrhu regulátorů úhlových rychlostí. Je třeba navrhnout dva nezávislé regulátory, pro každé kolo jeden. Díky konstrukční symetrii robotu mohou být tyto regulátory totožné.

Zvolili jsme standardní regulátor PID. Přesné nastavení PID regulátorů není cílem této práce, proto jsme návrh jeho parametrů provedli s pomocí nástroje, který je standardní součástí prostředí Matlab/Simulink. Konkrétní hodnoty budou uvedeny dále v kapitole 4.4 společně s použitými konkrétními parametry robotu.

Znalostí vztahů mezi úhlovými zrychleními kol $\ddot{\theta}_R$ a $\ddot{\theta}_L$ a působícími momenty M_R a M_L (2.18) a (2.19) můžeme s výhodou aplikovat jako dopřednou vazbu pro regulaci úhlových rychlostí kol.

Na referenční signál úhlových rychlostí kol musíme použít filtrovanou derivaci, díky které budeme mít k dispozici požadovaná úhlová zrychlení kol $\ddot{\theta}_R$ a $\ddot{\theta}_L$. Ty následně použijeme pro výpočet odhadu působících momentů M_R a M_L prostřednictvím rovnic (2.18) a (2.19). Získané hodnoty zavedeme kladnou dopřednou vazbou za rozvazbovací regulátor. Použitím dopředné vazby můžeme ovlivnit činnost rozvazbovacího regulátoru, avšak získané výsledky s uzavřenou dopřednou vazbou jsou v porovnání s výsledky bez ní řádově lepší.

Výsledná regulační smyčka je téměř totožná pro oba odvozené modely. Pouze zaměníme odpovídající subsystém na obrázku 11 označený jako *model robotu*.



Obrázek 11: Regulace úhlových rychlostí kol

4.4 Verifikace odvozených modelů

Pro účely verifikace obou sestavených modelů robotu a kvůli následným simulačním experimentům s modely musíme přesně nadefinovat veškeré parametry robotu.

Parametry robotu:

$m_p = 10[kg]$ hmotnost těla robotu

$m_w = 0.1[kg]$ hmotnost kola

$b = 0.05[m]$ vzdálenost kola od osy robotu

$r = 0.025[m]$ poloměr kola

$l = 0.01[m]$ šířka kola

$d = 0[m]$ vzdálenost těžiště robotu od středu osy kol

$J = \frac{1}{12}m_p(4b^2 + 4b^2)[kg \cdot m^2]$ moment setrvačnosti robotu ve středu os kol

$J_w = \frac{1}{2}m_w r^2 [kg \cdot m^2]$ moment setrvačnosti kola

$m = m_p + 2m_w [kg]$ celková hmotnost

Parametry PID regulátoru:

$P : 0.0075$

$I : 0.0663$

$D : -3.53 \cdot 10^{-6}$

N (filtrace derivační složky): 2122.98

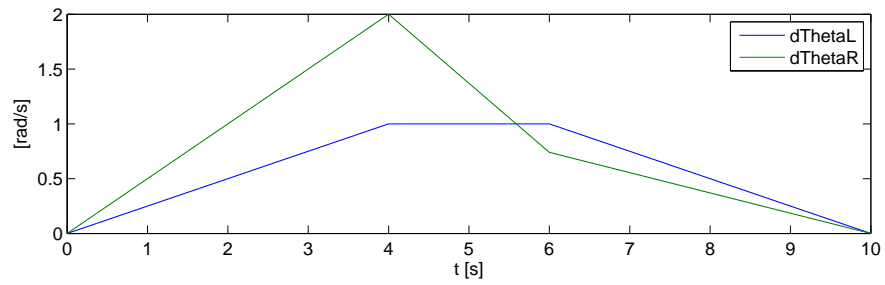
Verifikace probíhala s náhodně volenými referenčními signály pro požadované úhlové rychlosti kol $\dot{\theta}_L$ a $\dot{\theta}_R$. Dále uvedeme výsledky verifikace pro jeden konkrétní vstupní signál (obrázek 12a).

Graf na obrázku 12b je záznam trajektorie robotu v rovině XY s použitím modelu odvozeného v prostředí Simulink/SimMechanics. Chyba mezi matematickým modelem a modelem v prostředí SimMechanics je znázorněna na obrázku 12c. Vidíme, že odchylky jednotlivých rovinných souřadnic robotu jsou zanedbatelné. Řádově obdobné výsledky jsme obdrželi i u ostatních srovnávacích simulací s různými budícími signály.

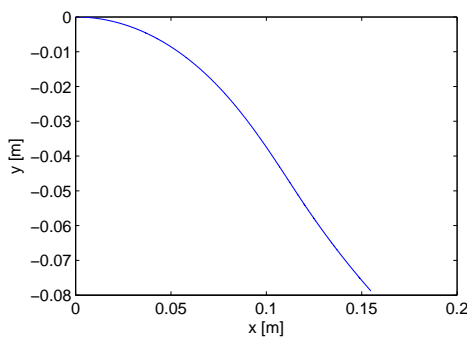
Pro kontrolu uvádíme i časový průběh chyby regulace úhlových rychlostí kol s použitím modelu odvozeného v prostředí Simulink/SimMechanics. Při srovnání s referenčním signálem je zřejmé, že chyba se skokově změní při skokové změně referenčního signálu. Brzy ale zcela odezní.

Poznámka 4.1 Máme tedy k dispozici model rovinného robotu odpovídající zadaným požadavkům, který nám poslouží pro testování a simulaci jeho chování s uzavřením zpětné vazby. Pro účely těchto simulací budeme výhradně používat model sestrojený v prostředí Simulink/SimMechanics.

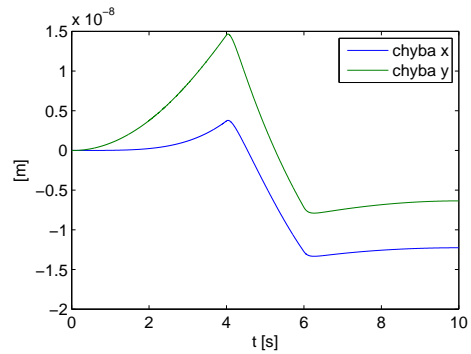
4 REGULACE ÚHLOVÝCH RYCHLOSTÍ KOL ROBOTU



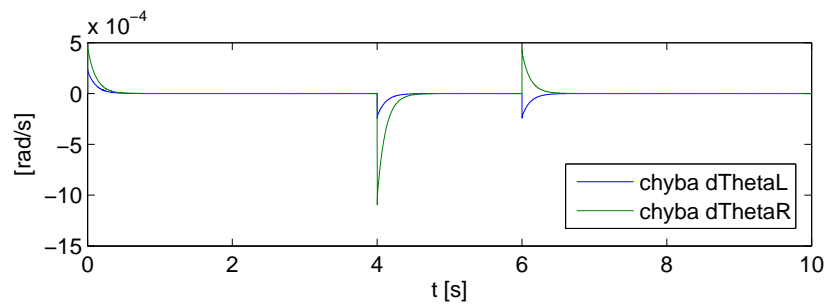
(a) Zvolený referenční signál pro verifikaci



(b) Výstup modelu v prostředí SimMechanics - XY graf



(c) Časový průběh odchylky odvozených modelů



(d) Časový průběh chyby PID regulace úhlové rychlosti obou kol

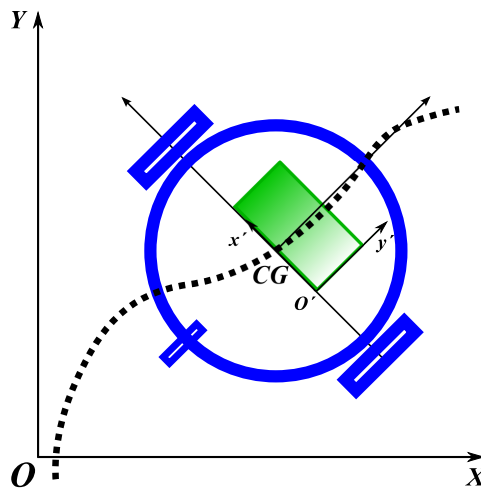
Obrázek 12: Verifikace modelů

5 Vizuální zpětná vazba

Stěžejním částí práce je implementace sledování zadané trajektorie rovinným mobilním robotem s dvěma nezávisle řízenými koly. Prostor před těžištěm robotu bude snímán kamerou, která poskytne obrazovou informaci k dalšímu zpracování. Na základě získaných snímků bude řídicí algoritmus v každém kroku vyhodnocovat svoji polohu vzhledem k žádané trajektorii a bude generovat odpovídající akční zásahy, aby bylo zajištěno co nejpřesnější sledování požadované trajektorie.

Předpokládáme, že v každém časovém okamžiku máme k dispozici snímek s výřezem žádané trajektorie odpovídající zeleně označené oblasti, přičemž se těžiště robotu nalézá v přední části vymezené oblasti a souřadný systém je zaveden jako na obrázku 13.

Zorné pole se může lišit v závislosti na použité kameře. Pro simulační účely uvažujeme jeho rozměry řádově odpovídající nákresu na obrázku 13.



Obrázek 13: Ilustrace zorného pole robotu

Jelikož nemáme k dispozici reálný model robotu s kamerou, na kterém bychom mohli vyvíjené algoritmy testovat v praxi, bylo nutné obrazovou informaci získat jiným způsobem. Požadovaná rovinná trajektorie byla nasnímaná jako jeden velký celek, kdy v každém kroku simulace je zhotoven výřez odpovídající aktuální poloze a natočení robotu.

Aby byl umožněn co nejjednodušší přechod od simulačního modelu robotu k reálnému, byly použity naskenované trajektorie namalované lihovým fixem na papír (výřez na obrázku 14), které tím pádem obsahují různé šумы a další nepřesnosti. Snažíme se tím co nejvíce přiblížit reálné situaci a eliminovat tak předpoklady, které by znemožňovaly použití robotu v praxi.

5.1 Obrazové zpracování trajektorie

Získanou obrazovou informací (obrázek 14) se budeme snažit maximálně využít pro správné řízení modelu robotu. Snažíme se snímek zpracovat takovým způsobem, aby výsledkem byly dva vektory hodnot - souřadnice x a y každého jednotlivého bodu trajektorie na daném snímku. Převedení obrazové informace na matematický popis využijeme pro vyhodnocení snímků a vygenerování odpovídajících akčních zásahů.

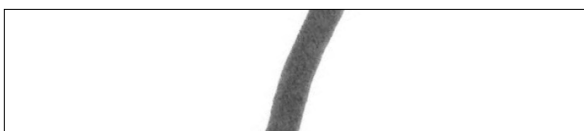


Obrázek 14: Pořízený snímek trajektorie

Každý získaný snímek musíme nejprve upravit určitou posloupností operací, kterou si definujeme.

Poznámka 5.1 *Tato posloupnost je navržena výhradně pro námi zhotovené zkušební trajektorie. Dále odvozené řídicí algoritmy budou ovšem fungovat i pro jiné typy snímků, pokud budou upraveny do náležitého tvaru. Z toho důvodu tato posloupnost může sloužit jako návod pro úvodní úpravu trajektorie.*

Nejprve ze získaného snímku ve standardním formátu *jpeg* v barevném prostoru RGB vybereme pouze jeden barevný kanál, ve kterém bude použita barva trajektorie nejvíce kontrastní. V našem případě můžeme pro červenou trajektorii zvolit buď zelený, nebo modrý barevný kanál. Obrázek 15 znázorňuje výběr zeleného barevného kanálu.



Obrázek 15: Výběr zeleného kanálu z RGB

Vzniklý šedotónový snímek nyní převedeme prahováním na černobílý. Metod prahování je celá řada a používají se obecně jako předzpracování

obrazu pro různé úlohy rozpoznávání. Pro nalezení optimálního globálního prahu použijeme metodu Otsu popsanou v [13]. Otsu metoda pracuje na principu statistického vyhodnocení údajů z histogramu šedotónového obrazu. Ukázka takto zpracovaného snímku je na obrázku 16.



Obrázek 16: Prahování obrazu - metoda Otsu

Musíme také pamatovat na šum v obraze vzniklý různými světelnými podmínkami při snímání či neexaktním nakreslení trajektorie. Proto ještě použijeme morfológickou operaci otevření s kruhovým strukturujícím elementem. Algoritmus můžeme nalézt např. v [14]. V počítačovém zpracování obrazu se obecně používá k morfológické redukci šumu.



Obrázek 17: Otevření

Z požadavků na převod trajektorie na matematický popis je výhodné proložit rozpoznané trajektorii křivku, která bude jejím ztenčením do jedné linie. Použijeme algoritmus popsaný v [15], pomocí kterého můžeme ztenčit různé objekty do linií.



Obrázek 18: Ztenčení trajektorie do linie

Nakonec použijeme funkci *find* implementovanou v programovém prostředí Matlab. Výstupem předzpracování obrazové trajektorie budou dva

stejně dlouhé vektory souřadnic x a y , pomocí kterých je plně popsána poloha trajektorie v aktuálním zorném poli robotu. Tyto hodnoty už můžeme dále vyhodnocovat s použitím matematického a geometrického přístupu.

Poznámka 5.2 *Všechny algoritmy použité na obrazové zpracování trajektorie jsou standardně implementovány v programovém prostředí Matlab v toolboxu ImageProcessing.*

Skript, který zajišťuje veškeré předpřipravení obrazové informace je uveden v příloze A.

Poznámka 5.3 *Před vlastním uzavřením vizuální zpětné vazby musíme provést kalibraci snímků trajektorie. Jedná se o převod mezi obrazovou informací získanou v pixelech a informací o poloze robotu v metrech ze sensorů v simulačním modelu. Tuto kalibraci můžeme provést např. tak, že pomocí kamery umístěné na robotu nasnímáme předmět či trajektorii známých rozměrů. Z obrazové informace pak snadno dopočteme vhodný kalibrační poměr.*

V našem případě používáme kalibraci $100px = 1cm$.

5.2 Uzavření vizuální zpětné vazby

Získaná obrazová informace musí být náležitě použita k řízení robotu. Z obrazového zpracování trajektorie dostaneme informaci o průchodu požadované trajektorie zorným polem robotu. Na základě těchto znalostí navrhne algoritmus, který vzniklou situaci vyhodnotí a jeho snahou bude se dané trajektorii limitně co nejvíce blížit. Zároveň by měl být navržený řídicí algoritmus stabilní a co nejvíce robustní.

Velká většina řídicích algoritmů diferenciálně řízených robotů sledujících zadanou trajektorii je založena na informaci, na které straně od trajektorie se robot v danou chvíli nachází. Podle toho se přibrzdí či urychlí jedno kolo tak, aby se odchylka vyrovnala. Toto je pouze hlavní myšlenka, kterou se každý snažil vylepšit po svém. My se pokusíme použít zcela jiný přístup, který už vychází ze samotné konstrukce robotu. Přirozeným pohybem diferenciálně řízených robotů je totiž pohyb po kružnici (viz odvození v kapitole 4.1).

Vizuální zpětná vazba využívá informace o poloze a natočení robotu z odvozeného modelu. Přesná posloupnost jednotlivých operací vyhodnocujících polohu robotu je uvedena na obrázku 19.

Nejprve je nutné získané hodnoty polohy x a y naměřené v základních jednotkách SI metrech převést na pixely. Tato konverze je závislá na rozlišení a zorném poli použité kamery a jejím umístění na robotu. Jedná se pouze o jednoduchý přepočet, který si můžeme usnadnit např. nasnímáním trajektorie známé délky. Její délku v pixelech odečteme ze snímku, reálnou délku známe, snadno tak dopočteme kalibrační poměr.

Další velmi důležitou částí je obrazové zpracování trajektorie, blíže popsané v kapitole 5.1.

Následuje výpočet požadovaného poloměru křivosti R^* , kterým se podrobně zabývá kapitola 6, kde se zabýváme různými metodami.

Poslední operací je převod požadovaného poloměru křivosti R^* na požadované úhlové rychlosti $\dot{\theta}_R$ a $\dot{\theta}_L$. Již v kapitole 4.1 bylo ukázáno, že poloměr křivosti kružnice, po které se robot v daném okamžiku pohybuje, je závislý pouze na úhlových rychlostech kol $\dot{\theta}_R$ a $\dot{\theta}_L$ a konstantní vzdálenosti kola od osy robotu b .

Z předpokladu konstantní lineární rychlosti robotu můžeme určit závislost mezi jednotlivými úhlovými rychlostmi robotu. Vyjádříme vztahy pro výpočet obou úhlových rychlostí kol:

$$\dot{\theta}_R = \frac{2v}{r} - \dot{\theta}_L \quad (5.1)$$

$$\dot{\theta}_L = \frac{2v}{r} - \dot{\theta}_R, \quad (5.2)$$

kde v je lineární rychlost robotu.

Poznámka 5.4 *Lineární rychlost robotu v chápeme jako parametr simulace, který si můžeme zvolit. Poloměr kol r je dán konstrukcí robotu.*

Do rovnice 6.1 dosadíme rovnici 5.1 pro závislost mezi úhlovými rychlostmi a vyjádříme konečný vztah pro úhlovou rychlost pravého kola $\dot{\theta}_R$:

$$\dot{\theta}_R = \frac{v}{r} \left(1 + \frac{b}{R} \right) \quad (5.3)$$

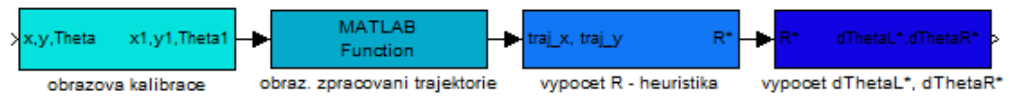
Dosazením z rovnice 5.3 do rovnice 5.2 a následnými elementárními úpravami dostaneme konečnou podobu vzorce pro výpočet úhlové rychlosti levého kola $\dot{\theta}_L$:

$$\dot{\theta}_L = \frac{v}{r} \left(1 - \frac{b}{R} \right) \quad (5.4)$$

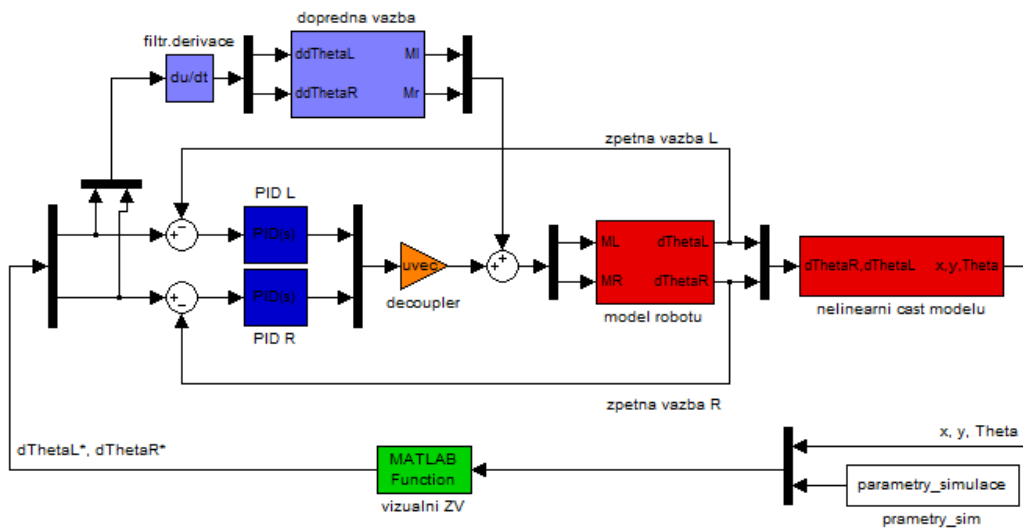
Poznámka 5.5 *Nyní máme k dispozici kompletní aparát, kterým můžeme řídit model robotu po požadovaném poloměru křivosti. Zbývá tedy nalézt požadovaný poloměr křivosti pohybu robotu a můžeme uzavřít kompletní vizuální zpětnou vazbu. Tímto problémem se podrobně zabývá následující kapitola 6.*

Regulační smyčka s uzavřenou vizuální zpětnou vazbou bude vypadat podle obrázku 20.

5 VIZUÁLNÍ ZPĚTNÁ VAZBA



Obrázek 19: Vizualní zpětná vazba - posloupnost operací

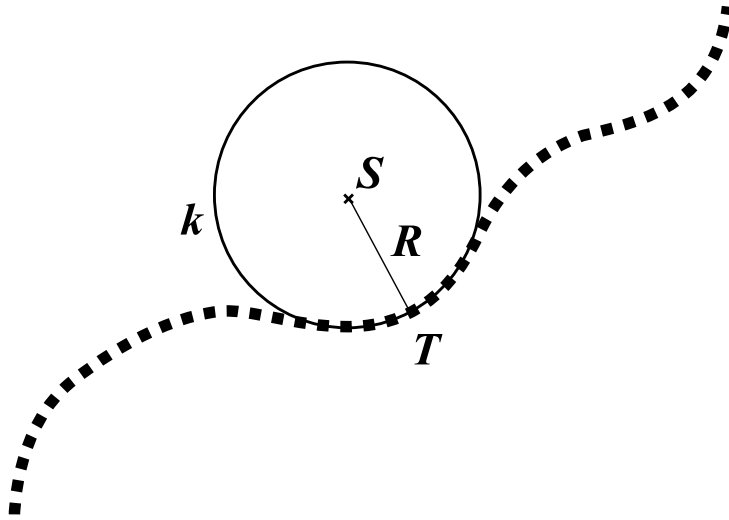


Obrázek 20: Regulační smyčka s vizualní zpětnou vazbou

6 Metody určení požadovaného poloměru R^*

6.1 Řízení s využitím oskulačních kružnic

Hlavní ideou řídicího algoritmu daného mobilního robotu bylo využití oskulačních kružnic. Oskulační kružnice (obrázek 21) je většinou definována jako kružnice procházející daným bodem T a dvojicí dalších bodů, které jsou umístěny nekonečně blízko bodu T . Střed S oskulační kružnice k leží na vnitřní části normály ke křivce v bodě T . Pro nás nejdůležitější vlastnost, kterou využijeme, je fakt, že poloměr křivosti R oskulační kružnice je shodný s poloměrem křivosti křivky v bodě T .



Obrázek 21: Oskulační kružnice

Nás bude konkrétně zajímat příklad nalezení poloměru křivosti u rovinné křivky, trajektorie, kterou chceme robotem co nejvěrněji sledovat. Pro výpočet poloměru křivosti v daném bodě křivky použijeme vzorec uvedený např. v [16]:

$$R = \frac{(\dot{x}^2 + \dot{y}^2)^{\frac{3}{2}}}{|\dot{x}\ddot{y} - \dot{y}\ddot{x}|}, \quad (6.1)$$

kde $x = x(t)$ a $y = y(t)$ jsou rovnice parametrického vyjádření křivky a tečkovaná notace znamená derivaci podle parametru t .

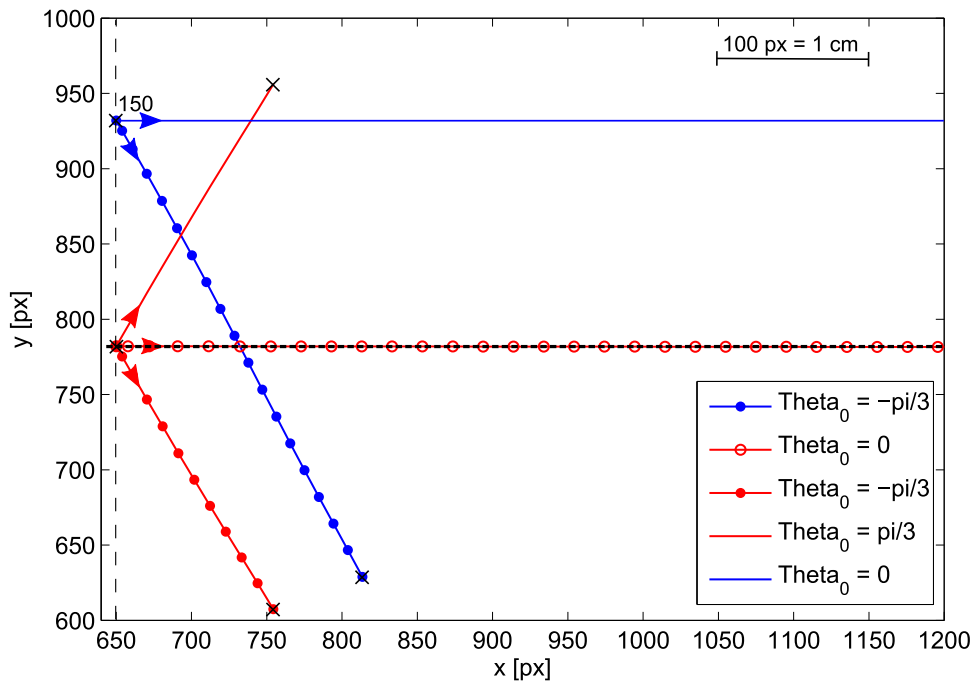
Poznámka 6.1 *Výhodou je, že poloměr může být kladný i záporný, proto není třeba ze směru křivky dále určovat, na které straně od křivky se bude oskulační kružnice nacházet.*

Ze zpracování obrazové informace (kapitola 5.1) máme k dispozici body trajektorie. Získanými body trajektorie proložíme polynom druhého řádu. K této křivce najdeme poloměr oskulační kružnice v bodě, kde trajektorie vstupuje do aktuálního snímku - jinými slovy $y = 0$. Takto získaný poloměr použijeme jako požadovaný poloměr křivosti R^* .

6.1.1 Zhodnocení metody

Po uzavření vizuální zpětné vazby a několika prvotních experimentech provedených na simulačním modelu v prostředí SimMechanics bylo zřejmé, že tato metoda nedosahuje požadovaných výsledků na kvalitu regulace.

Je to způsobeno tím, že navržený algoritmus prokládá oskulační kružnici, která nejlépe aproximuje část trajektorie na aktuálním snímku. Neuvažuje však odchylku robotu od trajektorie či jiné natočení robotu než ve směru směrnice trajektorie v aktuálním snímku v místě těžiště (směr pohybu robotu a směrnice sledované trajektorie nejsou rovnoběžné). Projev tohoto problému je ukázán na sledované trajektorii tvaru přímé linie na obrázku 22.



Obrázek 22: Ukázka chyby metody oskulačních kružnic

Obrázek znázorňuje chování robotu řízeného metodou oskulačních kružnic, kdy sledovanou trajektorií je rovná čára. Uvažované počáteční podmínky po-

lohy a natočení robotu vhodně ilustrují úskalí použití této metody. Pokud je robot zprvu natočen jiným směrem než je směrnice trajektorie, robot pojedí rovně - algoritmus určí jako požadovaný poloměr zatočení nekonečný poloměr.

Dalším problémem je, že metoda využívající oskulačních kružnic nemá žádný aparát, který by reagoval na odchýlení robotu od trajektorie. To má za následek neustálé integrování relativně malých chyb v každém kroku simulace. Tyto malé chyby jsou způsobeny většinou diskretizací simulace, kdy řídicí algoritmus není schopen zareagovat na spojitě změny poloměru křivosti trajektorie. V praktickém použití přibude ještě celá řada dalších nepřesností a chyb, s kterými si tento algoritmus nebude umět poradit.

Jako další demonstrační trajektorii jsme zvolili spirálu - tedy křivku se spojitě se zmenšujícím poloměrem křivosti - aby byla patrná absence integračního členu regulace.

Poznámka 6.2 *Trajektorie byly pro účely testování vygenerovány pomocí počítače, tedy bez jakýchkoliv přidaných šumů. V této práci při testování řídicích algoritmů klademe větší důraz na přesný tvar trajektorie než na reálné podání barev a světelných podmínek.*

Obrazové zpracování trajektorie bylo popsáno i s ohledem na reálné snímání kamerou připevněnou na robotu.

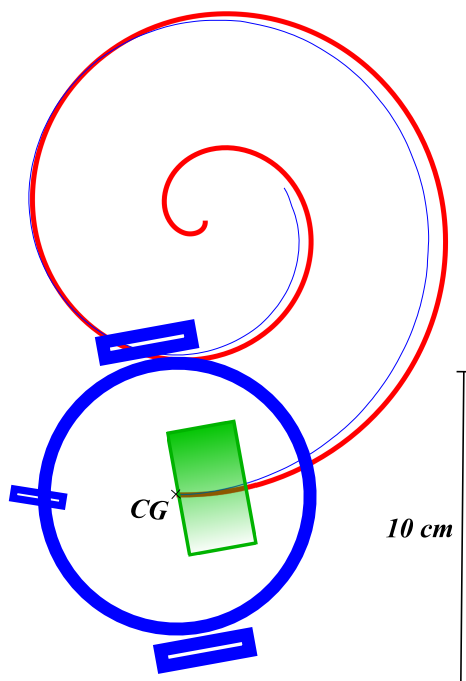
Výsledný pohyb robotu je vizualizován na obrázku 23. Pro ilustraci je uveden i robot v přibližném měřítku odpovídajícímu sestavenému matematickému a simulačnímu modelu.

Parametry simulace jsou:

$T[s]$	0.01	perioda vzorkování
$v[m.s^{-1}]$	0.2	lineární rychlost robotu
$t[s]$	3	čas simulace

Tabulka 1: Parametry simulace

Poznámka 6.3 *Metoda řízení pomocí oskulačních kružnic je obecně nestabilní. Není zaručena nulová regulační odchylka robotu od trajektorie v ustáleném stavu, proto se může stát, že robot sjede z požadované trajektorie úplně - trajektorie zmizí ze zorného pole robotu. Závisí to také na konfiguraci použité kamery a jejím zorném úhlu. Zorné pole robotu ale nemůže být zase přehnaně velké. Mohlo by dojít k situaci, kdy by požadovaná trajektorie v místě s malým poloměrem křivosti zasahovala do snímku z kamery dva a vícekrát a výpočet poloměru oskulační kružnice by selhal.*



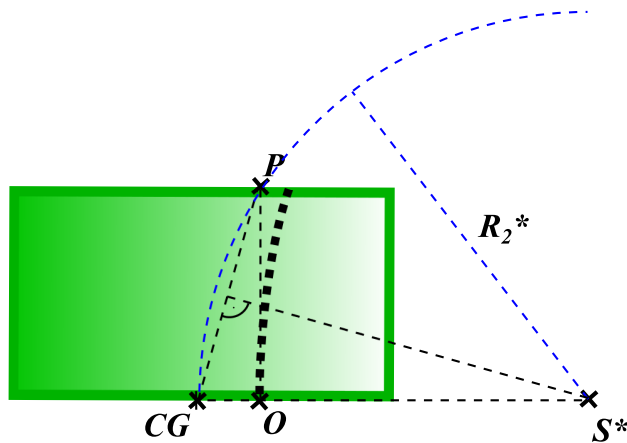
Obrázek 23: Pohyb robotu po spirále s využitím pouze oskulačních kružnic (červená - požadovaná trajektorie, modrá - trajektorie robotu)

Poznámka 6.4 Zatím se nezabýváme počátečním postavením robotu a oblastí atraktivity ke křivce. Bližší analýza tohoto problému je uvedena v kapitole 7. Uvedené experimenty uvažují ideální postavení robotu - tzn. robot stojí přesně na trajektorii a je natočen odpovídajícím směrem.

6.2 Řízení s využitím oskulačních kružnic s korekcí odchylky

Pokusíme se vylepšit metodu řízení robotu pomocí oskulačních kružnic o korekci regulační odchylky polohy robotu. Pokusíme se nalézt další požadovaný poloměr křivosti, tentokrát R_2^* , pouze z informace o odchylce těžiště robotu od požadované trajektorie. Ten společně s původním poloměrem křivosti z proložené oskulační kružnice R_1^* (viz kapitola 6.1) budeme vážit vhodným způsobem koeficientem α .

Způsob výpočtu R_2^* má geometrický základ podložený obrázkem 24. Bod O se nachází na hraně zorného pole, kde požadovaná trajektorie vchází do snímku. Bod P je pravoúhlým průmětem bodu O do protější strany snímku. Požadovaný střed křivosti S^* nalezneme jako průsečík osy úsečky mezi těžištěm CG a bodem P a přímkou procházející těžištěm a bodem O zároveň. Poloměr křivosti R_2^* je pak vzdáleností středu křivosti S^* od těžiště robotu.



Obrázek 24: Geometrický význam poloměru křivosti R_2^*

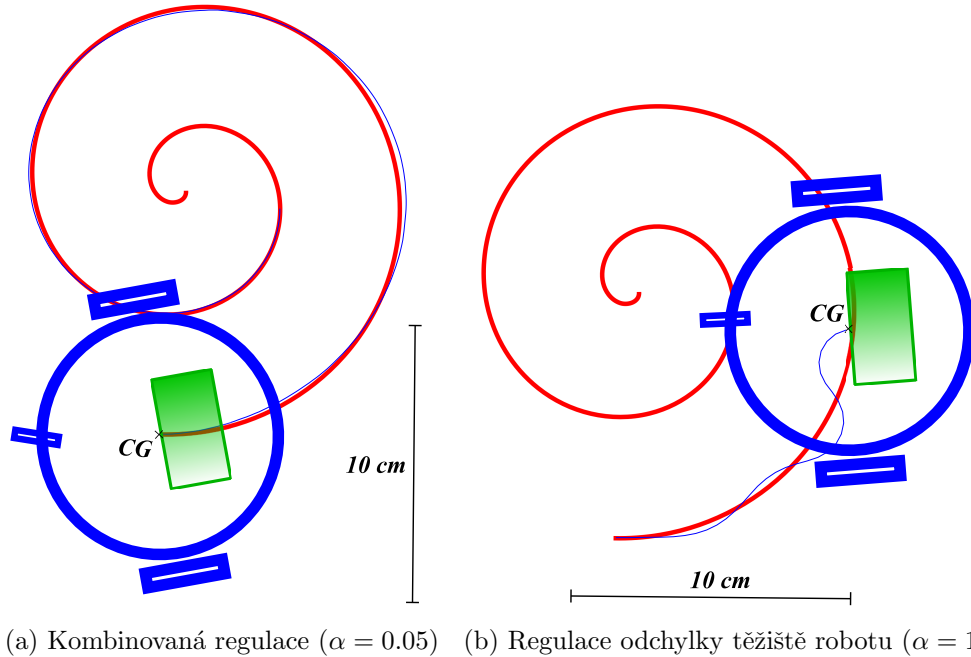
Pro výpočet celkového požadovaného poloměru křivosti R^* zavedeme ještě pojem křivost k . Křivost k je definována jako obrácená hodnota poloměru křivosti R :

$$k = \frac{1}{R} \quad (6.2)$$

Předpis pro výsledný požadovaný poloměr křivosti R^* je převrácená hodnota z křivostí k_1 a k_2 vážených koeficientem α :

$$R^* = \frac{1}{(1 - \alpha) k_1^* + \alpha k_2^*} \quad (6.3)$$

Celá řada experimentů s takto uzavřenou vizuální zpětnou vazbou ukázala, že pro naši konfiguraci robotu a řídicího systému dostáváme nejkvalitnější výsledky přibližně pro $\alpha = 0.05$ (viz obrázek 25a). I z uvedeného obrázku je na první pohled patrné zlepšení kvality regulace oproti výsledkům řídicího algoritmu využívajícího pouze oskulační kružnice (obrázek 23).



Obrázek 25: Pohyb robotu po spirále s využitím oskulačních kružnic a korekce s koeficientem $\alpha = 0.05$ (červená - požadovaná trajektorie, modrá - trajektorie robotu)

Pro srovnání uvádíme i průběh simulace při použití pouze regulace odchylky těžiště robotu - viz obrázek 25b. Toho docílíme, pokud parametr α bude roven jedné.

Z různých experimentů se simulačními modely jsme si všimli zajímavé skutečnosti. Jak jsme si již ukázali, řídicí algoritmus založený na principu oskulačních kružnic obstojně reguluje směr robotu, nikoliv však odchylku těžiště robotu od požadované trajektorie. Naproti tomu implementovaná regulace odchylky se snaží danou odchylku udržet nulovou. To se ale děje za cenu nedodržení směru vývoje sledované trajektorie a robot stále více kmitá kolem požadované trajektorie. Proto kombinace obou řídicích algoritmů dosahuje obecně lepších výsledků - kloubí do sebe totiž informaci ze současnosti (odchylka těžiště robotu od trajektorie) i z budoucnosti (oskulační kružnice aproximující budoucí průběh trajektorie).

Poznámka 6.5 *Parametry simulace jsou totožné jako v předešlém případě řízení pouze pomocí oskulačních kružnic.*

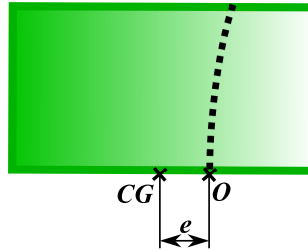
6.3 Kritérium kvality řízení

Pro účely srovnání kvality jednotlivých algoritmů řízení musíme navrhnout kritérium podle našich požadavků. Z dosud uvedených výsledků je patrné jisté zlepšení v návrhu řídicího algoritmu, ale pro další vývoj se nám bude hodit přesné kritérium kvality řízení, které umožní objektivní porovnání algoritmů.

Hlavním úkolem této práce je se zaměřit na sledování požadované trajektorie. Kritérium kvality řízení tedy stanovíme jako integrál z absolutní hodnoty odchylky těžiště robotu od požadované trajektorie:

$$J = \int_0^t |e| dt, \quad (6.4)$$

kde e je odchylka těžiště od bodu O - viz obrázek 26.



Obrázek 26: Geometrický význam odchylky e v zorném poli robotu

algoritmus	α	$J[px^2]$
oskulační kružnice	0	95.5034
oskul. kruž. + korekce e	0.05	36.7740
korekce e	1	alg. selhal

Tabulka 2: Srovnání algoritmů

Poznámka 6.6 *Tímto kritériem budeme porovnávat navržené řídicí algoritmy mezi sebou. Výsledky v tabulce 2 jen podpořily naše prvotní domněnky o zlepšení kvality řízení robotu s algoritmem uvažujícím i odchylku těžiště robotu od požadované trajektorie.*

6.4 Metoda korekce natočení robotu

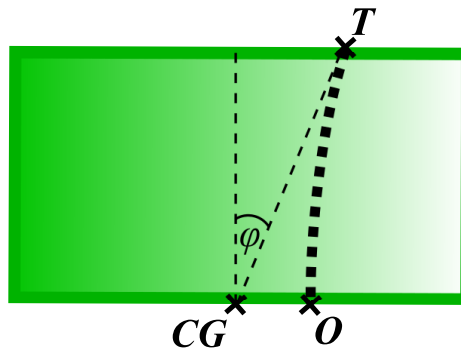
Řízení modelu robotu na základě použití oskulačních kružnic může být jistým startovacím bodem pro návrh více sofistikovaných algoritmů řízení. Z tohoto důvodu zde byla odvozena základní myšlenka algoritmu.

Dalším přínosem může být snaha o eliminaci chyb a nepřesností v již odvozených algoritmech. Metoda určení požadovaného poloměru R^* s využitím oskulačních kružnic selhává, pokud je regulační odchylka těžiště robotu od trajektorie nenulová nebo pokud směr pohybu robotu není shodný se směrnici trajektorie v místě těžiště robotu. Vše bylo blíže popsáno v kapitole 6.1.1.

Dalším problémem je spojitě se měnící poloměr křivosti křivky, kdy v případě odchylky robotu od trajektorie dochází ke zkreslení poloměru křivosti.

Snahou bude navrhnout takový přístup k určení požadovaného poloměru R^* tak, aby tyto nedostatky eliminoval. Při návrhu jsme prošli samozřejmě celou řadou slepých uliček. Jedním z nápadů, vylepšující metodu oskulačních kružnic, byla regulace natočení robotu s geometrickým opodstatněním.

Tento přístup určení požadovaného poloměru R^* využívá informaci z co možná největší budoucnosti, kterou nám snímek poskytne. Označme bod, kde trajektorie vystupuje z aktuálního snímku, jako bod T . Požadovaný poloměr R^* budeme generovat vhodným způsobem tak, aby v daném kroku simulace došlo k odregulování odchylky natočení robotu - tedy natočení o úhel φ . Snaha je udržet bod T na snímku v místě pravoúhlého průmětu těžiště.



Obrázek 27: Geometrický význam metody korekce natočení robotu

Známe tedy úhel φ , o který chceme robot natočit v dalším kroku simulace. Mezi lineární rychlostí a úhlovou rychlostí otáčení robotu kolem vlastní osy platí již dříve použitý vztah (4.1). Vyjádříme z něj úhlovou rychlost otáčení robotu:

$$\dot{\theta} = \frac{v}{R} \quad (6.5)$$

Simulace probíhá diskrétně, proto můžeme derivaci úhlu natočení robotu zapsat jako přírůstek úhlu natočení za přírůstek času:

$$\frac{\Delta\theta}{\Delta t} = \frac{v}{R} \quad (6.6)$$

Změnu času Δt můžeme nahradit periodou simulace T . Z rovnice (6.6) následně vyjádříme poloměr křivosti pohybu:

$$R = \frac{vT}{\Delta\theta} = R_3^* \quad (6.7)$$

Takto určeným poloměrem křivosti R_3^* nahradíme ve výpočtech poloměr křivosti R_1^* vypočtený s pomocí oskulační kružnice. Regulaci odchylky těžiště robotu od sledované trajektorie ponecháme beze změny, jen je třeba znovu heuristicky nalézt vhodný vážící koeficient α .

Při simulaci různých stavů, do kterých se může robot v praxi dostat, a hledání nejvhodnějšího koeficientu α jsme narazili na další nedostatek. Pokud jsme volili takové α , které dosahovalo nejlepších výsledků pro regulaci modelu z ideálních počátečních podmínek, řídicí algoritmus selhával, pokud byly použity počáteční podmínky robotu s větším posunem či natočením. Algoritmus selhával tím více, čím byla směrnice trajektorie v zorném poli robotu větší (jinými slovy čím byl směr pohybu robotu rozdílnější než směr sledované trajektorie). Algoritmus v takových případech vygeneroval tak malý poloměr křivosti pohybu robotu, že došlo ke ztrátě trajektorie ze zorného pole robotu.

Řešením se zdá být filtr, který upravuje účinnost požadovaného poloměru R_3^* v závislosti na velikosti směrnice sledované trajektorie v zorném poli robotu. Poloměr křivosti R_3^* tak vynásobíme číslem vyšším než 1 (přesně vyšším o absolutní hodnotu směrnice trajektorie), což má za následek větší poloměr pohybu robotu.

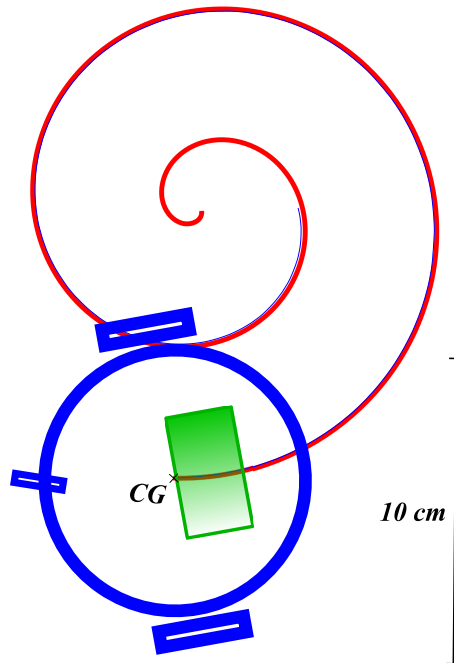
Výsledný zákon řízení je tedy úpravou rovnice (6.3):

$$R^* = \frac{1}{\frac{1-\alpha}{1+|t|}k_3^* + \alpha k_2^*}, \quad (6.8)$$

kde t je směrnice sledované trajektorie v bodě O ; $k_3^* = \frac{1}{R_3^*}$; $k_2^* = \frac{1}{R_2^*}$.

Poznámka 6.7 *Koeficient α jsme určili s přihlédnutím k různým simulačním experimentům jako $\alpha = 0.2$.*

Pro první náhled porovnání s předešlými metodami řízení robotu uvádíme na obrázku 28 simulační výsledky algoritmu při testování na trajektorii tvaru spirály a stejných parametrech simulace jako při testování předešlých algoritmů řízení.



Obrázek 28: Pohyb robotu po spirále s využitím vylepšeného algoritmu korekce natočení robotu o korekci odchylky polohy robotu a filtraci podle směrnice sledované trajektorie.

(červená - požadovaná trajektorie, modrá - trajektorie robotu)

algoritmus	α	$J[px^2]$
korekce natočení	0	32.3506
korekce natočení + korekce e	0.2	16.5150
korekce natočení + filtr t	0	30.9739
korekce natočení + korekce e + filtr t	0.2	14.2140
oskulační kružnice	0	95.5034
oskul. kruž. + korekce e	0.05	36.7740
korekce e	1	alg. selhal

Tabulka 3: Srovnání algoritmů

Tabulka 3 nabízí přehled srovnání všech odvozených algoritmů a jejich modifikací vzhledem k zavedenému kritériu. Metoda korekce natočení robotu si v tomto testu vedla obecně lépe než metody odvozené s využitím oskulačních kružnic.

Přídavná korekce odchylky polohy těžiště robotu od sledované trajektorie opět vylepšila výsledky základní metody určení požadovaného poloměru křivosti. Filtrace poloměru křivosti podle směrnice sledované trajektorie přispěla nejen v lepším výsledkům z hlediska kritéria J , ale výraznou měrou také vylepšila celkovou robustnost a stabilitu uzavřené regulační smyčky. Podrobné simulační výsledky budou diskutovány v navazující kapitole.

Poznámka 6.8 *Vzhledem k simulačním výsledkům v dalších provedených experimentech budeme nadále používat výhradně tuto metodu vylepšenou o korekci odchylky polohy robotu a filtraci podle směrnice sledované trajektorie.*

7 Komplexní testování vizuální zpětné vazby

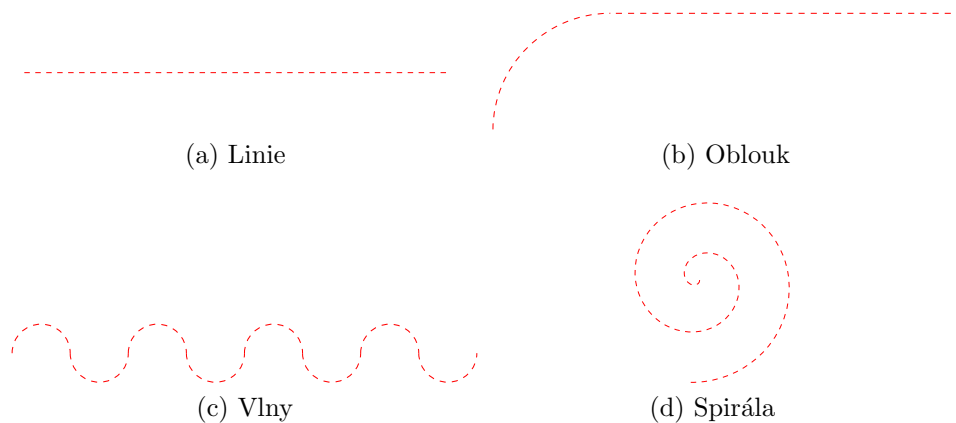
Prozatím odvozené metody řízení robotu poskytují nadějně výsledky v idealizovaných případech. Uvažujeme totiž ideální počáteční postavení robotu. Pokud bychom rádi takto vyvinuté algoritmy použili i v praxi, musíme se zaměřit také na robustnost použitých algoritmů řízení a jejich schopnost odolávat nepřesnostem tak obvyklých v praktickém použití.

V této kapitole budeme testovat chování simulačního modelu řízeného navrženými algoritmy na různých testovacích trajektoriích a s různými neidealizovanými počátečními podmínkami robotu.

Snahou je otestovat stabilitu výše odvozených metod řízení robotu. Analýza stability uzavřené regulační smyčky může být obecně provedena dvěma způsoby - matematicky a simulačně. Provedení matematické analýzy stability by bylo značně obtížné, hlavně z důvodu nelinearity a obtížného popisu uzavřené vizuální zpětné vazby. Simulační ověření stability má také svá úskalí. Sada provedených simulačních experimentů musí být vyčerpávající, aby nám poskytla dostatek informací o chování uzavřené regulační smyčky.

7.1 Sada testovacích trajektorií

Sada testovacích trajektorií byla zvolena tak, aby co nejvíce vystihovala kritické situace, které mohou vzniknout během sledování rozličných druhů trajektorií. Cílem je vybrat takové testovací trajektorie, pomocí kterých získáme komplexní informace o stabilitě navržených řídicích algoritmů.



Obrázek 29: Sada testovacích trajektorií

Základní úlohou sledování trajektorie je jízda robotu po rovné linii - obrázek 29a. Pokud nebude robot s daným řídicím algoritmem schopen sle-

dovat rovnou linií a neudrží ji ve svém zorném poli, nemá smysl testovat jeho chování na dalších složitějších trajektoriích.

Další testovací trajektorie prověřuje chování řídicích algoritmů při průjezdu robotu zatáčkou konstantního poloměru.

Trajektorie tvaru vln - spojených půlkružnic za sebe do řady (viz obrázek 29c) - má za účel prověřit, na jak velké poloměry zatočení umí ještě algoritmus odpovídajícím způsobem zareagovat. Zvolením několika různých poloměrů použitých půlkružnic byla otestována rozlišovací schopnost dané vizuální zpětné vazby.

Mimo různých tvarů testovacích trajektorií se také zaměříme na prozkoumání oblasti atraktivity různými počátečními podmínkami. Takovou oblast budeme chápat jako množinu počátečních podmínek polohy a natočení robotu vzhledem ke sledované trajektorii, kdy máme zaručeno její správné sledování.

Poznámka 7.1 *Řídicí algoritmy neuvažují žádným způsobem, jak je sledovaná trajektorie orientovaná. Tato informace není ani nijak čitelná z vlastní trajektorie (např. šipkami apod.). Proto předpokládáme, že odchylka natočení robotu od směru trajektorie v dané počáteční poloze robotu je v otevřeném intervalu $(-\frac{\pi}{2}, \frac{\pi}{2})$ tak, aby řídicí algoritmus bez další informace jednoznačně pochopil požadovaný směr.*

Vybraný algoritmus otestujeme při sledování výše uvedené sady trajektorií. Snahou je najít takové počáteční podmínky umístění robotu vzhledem k sledované trajektorii, kdy bude daný řídicí algoritmus ještě schopen spolehlivě navigovat robot k trajektorii.

Počátečními podmínkami rozumíme počáteční polohu v rovině xy a úhel natočení robotu vzhledem ke kladné poloose x .

Obecné parametry simulace jsou obdobné jako v předchozích ukázkách řízení pomocí jednotlivých algoritmů. Ke všem prezentovaným simulacím využíváme model odvozený v této práci v prostředí Matlab / Simulink / SimMechanics.

$T[s]$	0.01	perioda vzorkování
$v[m s^{-1}]$	0.2	lineární rychlost robotu
$t[s]$	různý	čas simulace

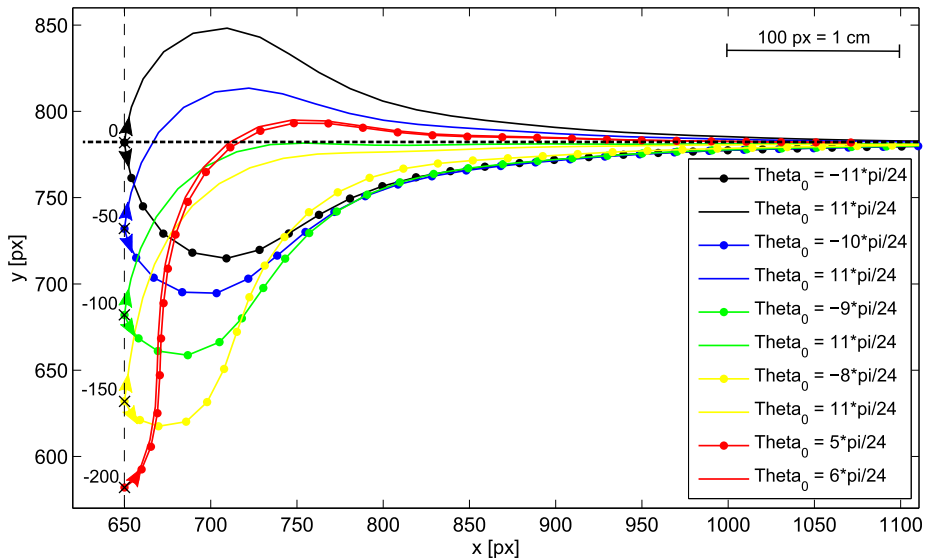
Tabulka 4: Parametry simulace

7.2 Trajektorie linie - výsledky

Analýza řídicího algoritmu při sledování trajektorie tvaru přímé linie spočívá v našem případě v nalezení různých počátečních podmínek, pro které se model robotu přiblíží ke sledované trajektorii a bez vnějších zásahů si ji udrží.

Osová souměrnost testovací trajektorie tvaru linie nám dovoluje zkoumat posun robotu od trajektorie jen záporným směrem po ose y . Při posunu v kladném směru bychom obdrželi totožné výsledky jen osově převrácené podle sledované trajektorie.

Obrázek 30 znázorňuje výsledky simulačních experimentů provedených pro pět různých možností posunu počáteční polohy robotu ve směru osy y : 0, -50, -100, -150 a -200 px. Pro každou takovou konfiguraci budeme hledat co největší množinu počátečních úhlů natočení robotu tak, aby robot byl schopen sledovat požadovanou trajektorii. Šipky naznačují směr pohybu robotu.



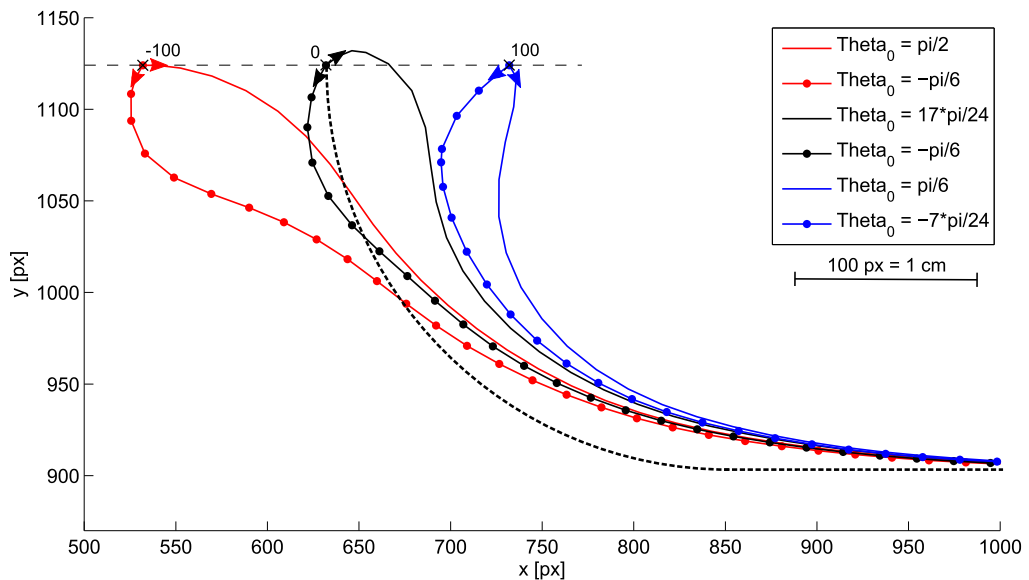
Obrázek 30: Sledování linie - počáteční posunutí 0, -50, -100, -150 a -200 px

Sledovaná trajektorie je vyznačena tlustou černou přerušovanou čarou ve směru osy x . Z každé výchozí pozice vede dvojice trajektorií stejné barvy. Tato dvojice trajektorií ohraničuje množinu všech trajektorií, které mají pro danou počáteční polohu robotu smysl - pro které máme zajištěno za daných podmínek sledování požadované trajektorie robotem. Každá taková trajektorie odpovídá jednomu počátečnímu úhlu natočení robotu pro danou počáteční polohu. Dostáváme tak uzavřený interval všech vyhovujících počátečních natočení robotu. Uvedené hodnoty v legendě obrázku 30 vyjadřují odchylky

natočení robotu od počáteční směrnice sledované trajektorie. Tvoří tak hranici přípustného intervalu odchylek.

7.3 Trajektorie oblouk - výsledky

Další zvolenou testovací trajektorií je trajektorie tvaru oblouku. Opět budeme hledat co největší množinu počátečních podmínek vyhovujících řídicímu algoritmu. Tentokrát volíme tři možné varianty posunu počáteční polohy robotu - vzhledem ke tvaru trajektorie ve směru osy x - vně oblouku (-100 px), na oblouku (0 px) a uvnitř oblouku (100 px).



Obrázek 31: Sledování oblouku - počáteční posunutí -100, 0 a 100 px

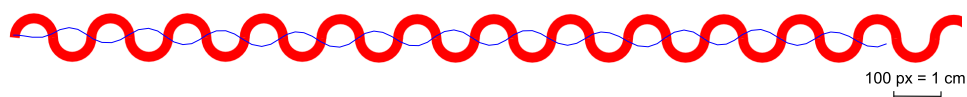
Popis obrázku 31 je obdobný jako v případě obrázku 30 uvedeném v předchozí podkapitole.

Zajímavým výsledkem je výrazně menší interval vyhovujících odchylek natočení robotu pro polohu uvnitř oblouku než vně. Před provedením této sady simulací jsme intuitivně předpokládali, že výsledky budou mít opačný charakter. Hlavní důvod je následující: při zkoumání počátečních podmínek uvnitř oblouku dochází mnohem častěji k pohybu robotu kolmo na sledovanou trajektorii. Algoritmus pak není schopen rozlišit, na kterou stranu sledované trajektorie má robot jet nebo trajektorii kolmo překříží a ztratí ji ze svého zorného pole.

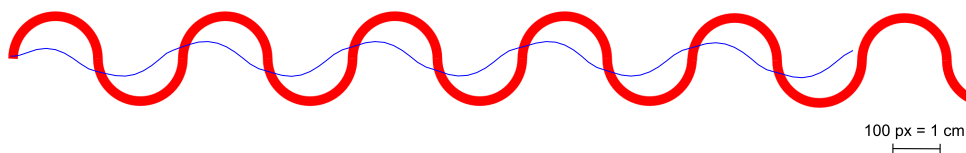
7.4 Trajektorie vlny - výsledky

Rozlišovací schopnost řídicího algoritmu prověřuje trajektorie tvaru vln - do řady spojené půlkružnice stejného poloměru.

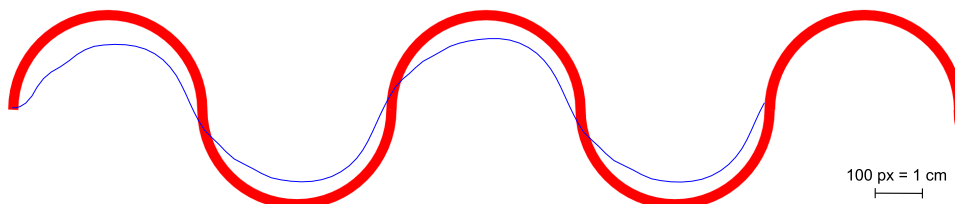
Obrázek 32 ukazuje výsledky simulací při sledování vlnitých trajektorií s různými poloměry půlkružnic.



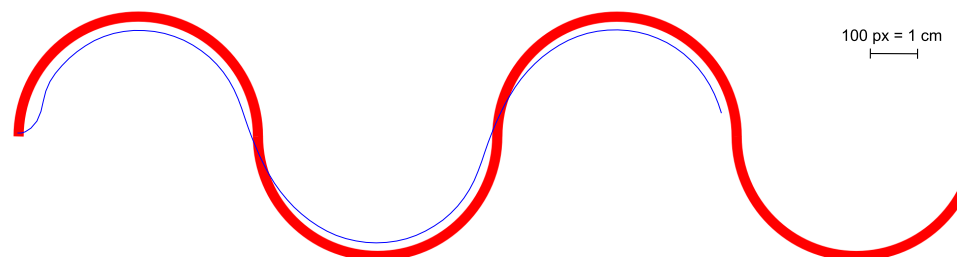
(a) Poloměr 50 px



(b) Poloměr 100 px



(c) Poloměr 200 px



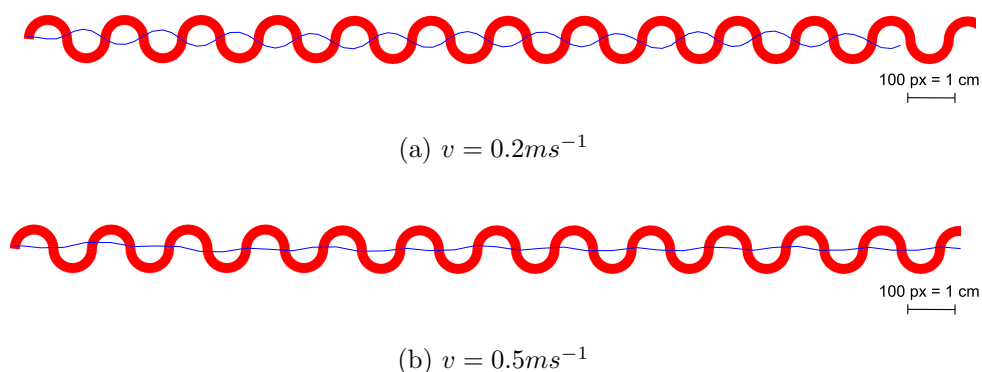
(d) Poloměr 250 px

Obrázek 32: Sledování vlnitých trajektorií - různé poloměry

Pro velmi malé poloměry zatočení není algoritmus schopen zareagovat odpovídajícím způsobem tak, aby robot tyto zatáčky svým pohybem kopíroval. Hlavními faktory, které tento fakt ovlivňují jsou zvolená lineární rychlost

robotu a délka zorného pole. Délka zorného pole nám říká, do jak velké budoucnosti se při generování akčních zásahů díváme. Pokud bychom tedy předpokládali použití námi vyvinuté vizuální zpětné vazby pro sledování trajektorií s velmi malými poloměry křivosti, můžeme odpovídajícím způsobem zvolit velikost zorného pole robotu a jeho lineární rychlost.

Částečný vliv změny zvolené lineární rychlosti robotu na kvalitu řízení je ukázán na obrázku 33.



Obrázek 33: Vliv změny zvolené lineární rychlosti robotu na kvalitu regulace

7.5 Shrnutí testování

Testovací sada trajektorií a simulace, které jsme s její pomocí provedly, nám nabízí praktický pohled na vlastnosti uzavřené vizuální zpětné vazby. Cílem bylo ověřit vhodnost navržené metody korekce natočení robotu se všemi jejími vylepšeními a odhalit co nejvíce omezení, která nás limitují v jejím praktickém použití.

Podarilo se nám identifikovat konkrétní omezení řídicího algoritmu. Mezi nejzásadnější patří volba rozměru zorného pole robotu. Pokud je zorné pole moc velké a poloměr křivosti sledované trajektorie v daném místě dostatečně malý, mohou se v aktuálním snímku vyskytovat dvě různé části sledované trajektorie. Námi navržená vizuální zpětná vazba nemá vyvinutý žádný algoritmus, který by dovedl správně vyhodnotit i tento stav.

Dále musíme dávat pozor na orientaci sledované trajektorie. Navržený algoritmus neumí rozeznat směr trajektorie, proto musí být počáteční natočení robotu takové, aby byla orientace trajektorie jednoznačně rozpoznána.

8 Regulace podtlaku v reálném modelu

8.1 Motivace

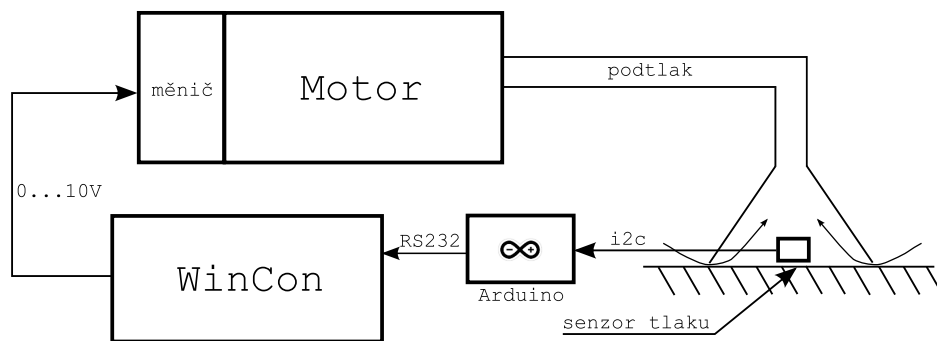
Úkolem této kapitoly je provedení experimentu s regulací podtlaku na reálném modelu podtlakové komory. Snahou je v praxi ověřit elementární myšlenky koncepce podtlakového přísávání pro aplikaci v mobilních manipulátorech.

Všechny současně používané technologie, se kterými jsme se seznámili, vyráběly konstantní podtlak. Ochranu proti úplnému přísátí robotu k podložce řešily různými technologiemi těsnění kolem podtlakové komory či použitím štětinek. Z našeho pohledu se jedná o velmi nespolehlivá řešení, která bychom rádi nahradili právě regulací podtlaku na požadovanou mez.

Hlavním cílem je sestavení regulační smyčky tlaku. Budeme využívat informací z tlakového čidla, které budou zpracovány pomocí průmyslového PC WinCon s nainstalovaným řídicím systémem REX.

8.2 Reálný model

Sestavení reálného modelu proběhlo v laboratoři za použití motoru, který jsme zapojili tak, aby vytvářel podtlak. Na motor jsme připojili hadici od vysavače a pevně spojili s plastovým trychtýřem o průměru 16 cm. Trychtýř představuje velmi zjednodušeně tělo mobilního robotu, kde může v budoucnu přibýt diferenciálně poháněná dvojice kol a kamera k využití ve vizuální zpětné vazbě.



Obrázek 34: Schéma reálného modelu

Dovnitř trychtýře jsme umístili senzor atmosférického tlaku MPL115A2I2C. Dle specifikace [17] je schopen snímat tlak už od 50 kPa (přibližně polovina běžného atmosférického tlaku).

Z důvodu absence ovladače na komunikaci I2C pro WinCon jsme senzor tlaku zapojili ještě přes Arduino Uno, které bylo použito jako převodník na

běžnou sériovou linku.

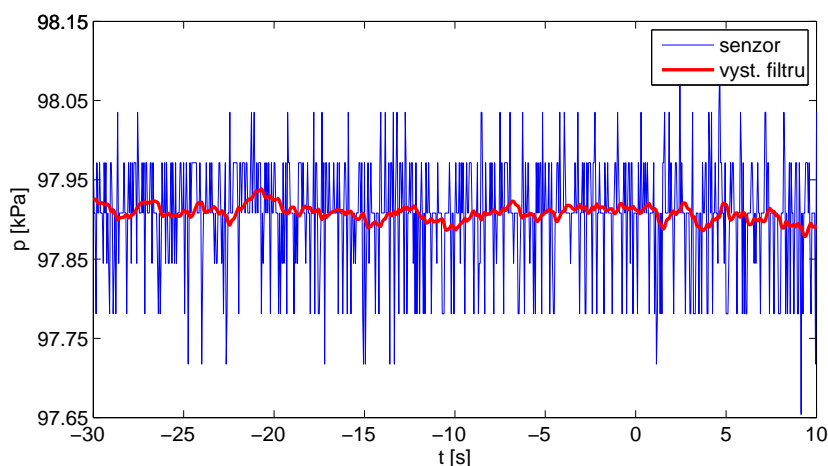
Další nezbytnou součástí modelu je průmyslový počítač WinCon. Uvnitř je nainstalovaný řídicí systém REX. Řídicí algoritmus vyhodnocuje vstup ze senzoru tlaku a následně generuje odpovídající akční zásah v rozmezí 0 až 10 V ke zpracování v měniči přiřazeném k motoru. Detailnější popis implementace řídicího systému bude proveden dále.

8.3 Implementace řídicího systému

Nejdříve jsme se zaměřili na signál ze senzoru tlaku. Jak je patrné z obrázku 35, výstup senzoru atmosférického tlaku je značně zašumělý. Proto před vlastní zpracování signálu vřadíme jednoduchý filtr prvního řádu, kterým odfiltrujeme daný šum. Návrh parametrů filtru bylo nutné provést s ohledem na dynamiku celého řízeného systému. Bylo samozřejmě možné obdržet ještě hladší křivku při konstantním tlaku, ovšem při výraznější změně tlaku (např. požadovaná hodnota tlaku se skokově změnila) výstup filtru dostatečně rychle nekopíroval dynamiku změny tlaku. Použitý filtr je tedy jakýmsi kompromisem - filtr prvního řádu s časovou konstantou $\tau = 1$ a statickým zesílením $k_0 = 1$:

$$F = \frac{1}{s + 1} \quad (8.1)$$

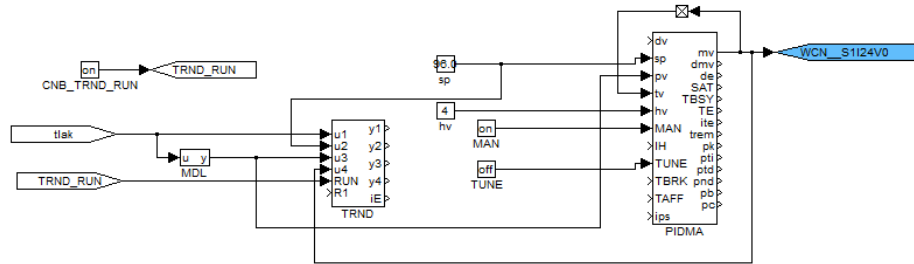
Poznámka 8.1 Předpokládejme, že okolní atmosférický tlak je po dobu provádění měření konstantní.



Obrázek 35: Naměřená data ze senzoru při konstantním tlaku - filtrace

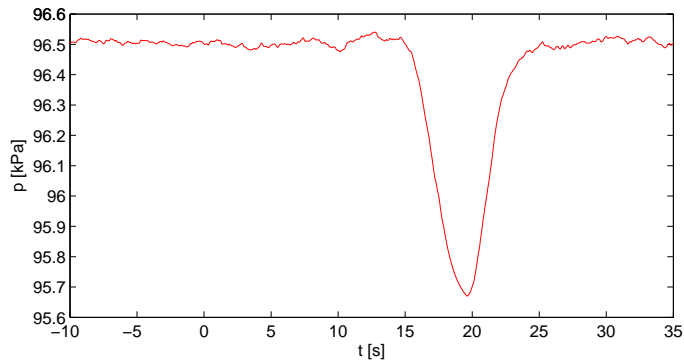
8 REGULACE PODTLAKU V REÁLNÉM MODELU

Dále uvedeme navržené regulační schéma v prostředí REX. Srdcem sestaveného řídicího systému je blok PIDMA, kde jsme s výhodou použili již implementovaný momentový autotuner pro nastavení parametrů regulátoru. Po úvodní konfiguraci parametrů bloku PIDMA jsme mohli spustit vlastní experiment samonastavení. Strukturu regulátoru jsme zvolili PI. Experimenty jsme samozřejmě provedli i pro konfiguraci PID regulátoru, ale získané výsledky byly obdobné v porovnání s PI regulátorem.

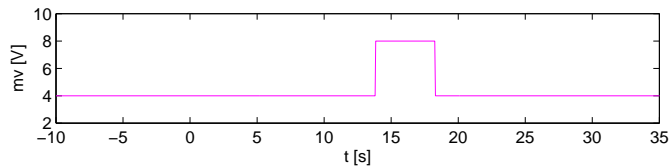


Obrázek 36: Schéma v prostředí REX

Průběh procesu ladění je znázorněn na obrázku 37. Bližší popis principu ladění regulátorů s momentovým autotunerem je uveden v Referenční příručce funkčních bloků systému REX [18].



(a) řízená veličina - tlak



(b) akční veličina - napětí

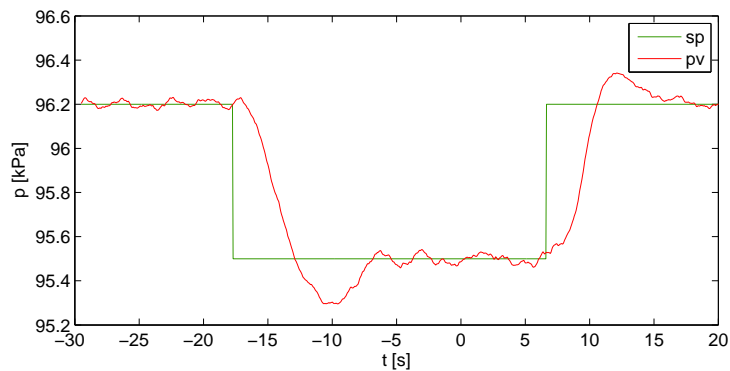
Obrázek 37: Průběh ladění regulátoru pomocí bloku PIDMA

Poznámka 8.2 Horní graf obrázku 37 ukazuje průběh regulované veličiny - tlaku. Dolní graf pak zachycuje akční zásah vygenerovaný momentovým autotunerem v bloku PIDMA.

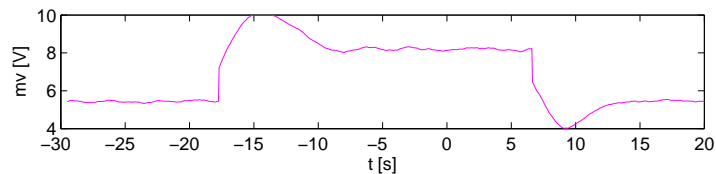
Po úspěšném naladění parametrů regulátoru (viz tabulka 5) jsme přešli k samotné regulaci. Jako referenční signál jsme zvolili skokovou změnu tlaku nejdříve z 96.2 kPa na 95.5 kPa a po ustálení skok zpět na tlak 96.2 kPa - viz signál sp v obrázku 38. Výsledné grafy regulované veličiny a akční veličiny jsou uvedené na obrázku 38.

k	2.4742
T_i	1.2096
b	1.0

Tabulka 5: Naladěné parametry regulátoru



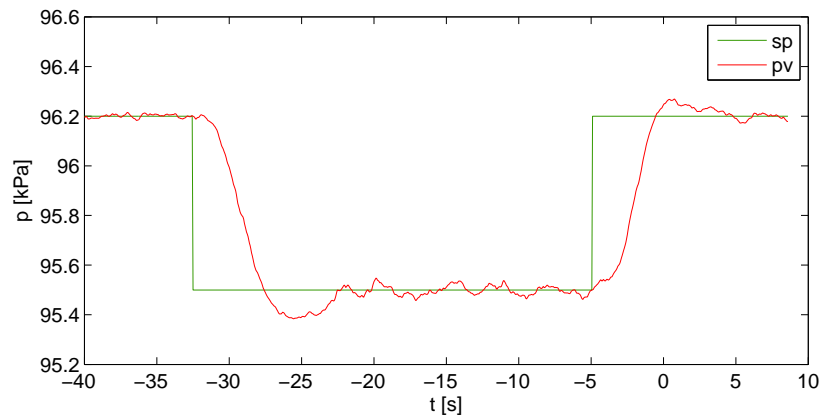
(a) sp - požadovaná hodnota, pv - řízená veličina



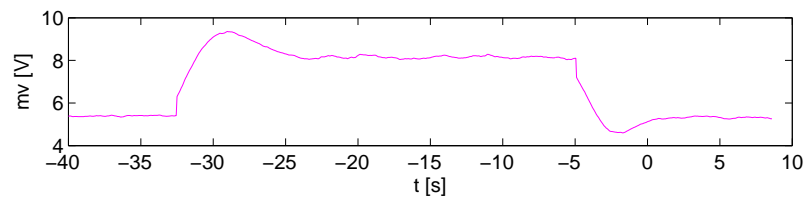
(b) akční veličina

Obrázek 38: Odezva na skokovou změnu požadované hodnoty, $b=1$

Dále jsme projevili snahu zmírnit překmitý, ke kterým docházelo při skokové změně referenčního signálu. Vhodnou změnou váhového koeficientu proporcionální složky b se nám podařilo překmitý mírně vylepšit - viz obrázek 39 - regulace při hodnotách váhového koeficientu proporcionální složky $b = 0.5$.



(a) sp - požadovaná hodnota, pv - řízená veličina



(b) akční veličina

Obrázek 39: Odezva na skokovou změnu požadované hodnoty, $b=0.5$

8.4 Zhodnocení regulace podtlaku

V této kapitole jsme položili základ pro regulaci podtlaku na požadovanou hodnotu za použití reálného modelu. Tato úloha je jednou z dílčích úloh pro následnou analýzu mobilního robotu s podtlakovým přísáváním.

Regulace podtlaku byla mimo jiné ovlivněna použitým motorem. Tento motor nebyl primárně určen na výrobu podtlaku, nicméně pro základní provedení této úlohy byl dostačující. Dynamika daného motoru se promítla do vlastní regulace a tím ovlivňovala její výsledky.

9 Závěr

Cílem této práce bylo vyvinout zcela novou koncepci robotického manipulátoru vhodného pro nedestruktivní testování svarů potrubí v jaderných elektrárnách. Výsledkem je koncept podvozku robotu s dvěma diferenciálně poháněnými koly, který je schopen s uvedenými předpoklady a omezeními sledovat požadovanou trajektorii. Podtlaková komora robotu má za úkol udržet robot na povrchu potrubí a umožnit mu volný pohyb po potrubí. Na těle robotu je umístěna kamera snímající prostor před robotem. Řídící systém tak dostává vizuální informaci o poloze robotu vůči sledované trajektorii.

Praktická ukázka regulace vyvíjeného podtlaku v podtlakové komoře byla implementována a je popsána v závěrečné části této práce. Informace ze senzoru tlaku umístěného v podtlakové komoře jsou zpracovávány pomocí řídicího systému REX v průmyslovém počítači WinCon. Řídící systém vygeneruje odpovídající akční zásah, který je přiveden do měniče motoru používaného k vyvíjení podtlaku.

Komplexní softwarová analýza podvozku vedla k sestavení matematického modelu a simulačního modelu v prostředí MATLAB / Simulink / SimMechanics. Dynamický popis byl získán aplikací Lagrangeovy metody. Verifikace těchto dvou modelů pomohla k ověření jejich správného odvození. Model v prostředí SimMechanics byl dále použit při veškerých simulacích a při testování vyvinutého řídicího algoritmu robotu. Dále byla odvozena regulační smyčka úhlových rychlostí kol. Dynamické rovnice jsme použili pro uzavření pomocné dopředné vazby. Regulaci úhlových rychlostí robotu zajišťuje dvojice PID regulátorů v sériovém spojení s rozvazbovacím regulátorem za pomoci dopředné vazby.

Hlavní částí řídicího algoritmu robotu a stěžejní kapitolou této práce je návrh nelineární vizuální zpětné vazby. Heuristicky bylo nalezeno několik různých variant zákona řízení zpracovávajících vizuální informaci obdrženou z kamery na těle robotu. Vybraný algoritmus s nejlepšími výsledky jsme dále testovali na definované sadě sledovaných trajektorií. Kapitola testování řídicího algoritmu se zaměřuje na nalezení vhodných počátečních podmínek polohy a natočení robotu, při kterých je algoritmus schopen řídit robot podle požadované trajektorie. Získané poznatky nám pomohly usoudit o stabilitě a robustnosti navržené vizuální zpětné vazby.

Koncept robotického manipulátoru je zpracován a připraven pro další simulaci či praktické experimenty. Zatím jsme nenarazili na žádnou skutečnost, která by stála v cestě ke správné funkčnosti výsledného reálného zařízení, ale tato cesta bude ještě dlouhá. Nicméně předvedená diplomová práce poskytuje nejen prvotní seznámení s daným problémem, ale i pokročilou analýzu konkrétní vizuální zpětné vazby a kinematiky lokomočního ústrojí robotu.

Seznam obrázků

1	Ukázka současných průmyslových skenerů potrubních svarů . . .	2
2	Přehled technologií pro přilnavost robotů	3
3	Demonstrace použití elektroadheze	4
4	Schéma uvažovaného mobilního robotu	5
5	Geometrická interpretace obvodových rychlostí kol	7
6	Geometrické znázornění lineární rychlosti robotu	8
7	Matematický model v prostředí Simulink	12
8	Modelu robotu v prostředí Simulink/SimMechanics	13
9	Subsystémy hlavního modelu	14
10	Vizualizace modelu robotu v prostředí Simulink/SimMechanics	14
11	Regulace úhlových rychlostí kol	17
12	Verifikace modelů	19
13	Ilustrace zorného pole robotu	20
14	Pořízený snímek trajektorie	21
15	Výběr zeleného kanálu z RGB	21
16	Prahování obrazu - metoda Otsu	22
17	Otevření	22
18	Ztenčení trajektorie do linie	22
19	Vizuální zpětná vazba - posloupnost operací	25
20	Regulační smyčka s vizuální zpětnou vazbou	25
21	Oskulační kružnice	26
22	Ukázka chyby metody oskulačních kružnic	27
23	Pohyb robotu po spirále s využitím pouze oskulačních kružnic	29
24	Geometrický význam poloměru křivosti R_2^*	30
25	Pohyb robotu po spirále s využitím oskulačních kružnic a korekce	31
26	Geometrický význam odchylky e v zorném poli robotu	32
27	Geometrický význam metody korekce natočení robotu	33
28	Pohyb robotu po spirále s využitím korekce natočení	35
29	Sada testovacích trajektorií	37
30	Sledování linie - počáteční posunutí 0, -50, -100, -150 a -200 px	39
31	Sledování oblouku - počáteční posunutí -100, 0 a 100 px	40
32	Sledování vlnitých trajektorií - různé poloměry	41
33	Vliv změny zvolené lineární rychlosti robotu na kvalitu regulace	42
34	Schéma reálného modelu	43
35	Naměřená data ze senzoru při konstantním tlaku - filtrace . . .	44
36	Schéma v prostředí REX	45
37	Průběh ladění regulátoru pomocí bloku PIDMA	45
38	Odezva na skokovou změnu požadované hodnoty, $b=1$	46
39	Odezva na skokovou změnu požadované hodnoty, $b=0.5$	47

Reference

- [1] Force Technology. APS - 6 [online]. ©2013 [cit. 2013-04-26] Dostupné z: <http://www.forcetechnology.com/en/Menu/Products/P-scan/Scanners/General/aps6.htm>
- [2] Olympus. WeldROVER Scanner [online]. ©2013 [cit. 2013-04-26] Dostupné z: <http://www.olympus-ims.com/cs/scanners/weldrover/>
- [3] Kim, S. *Spinybot II: Climbing hard walls with compliant microspines*. A. T. Asbeck, M. R. Cutkosky, W. R. Provancher. V Proc. Int. Conf. Adv. Robot., 2005, pp. 601–606.
- [4] Longo, D. *The Alicia3 climbing robot for automatic wall inspection*. G. Muscato. IEEE Robotics and Automation Magazine, Vol. 13, N.1, pp. 42-50, March 2006.
- [5] Nishi, A. *A wall-climbing robot using thrust force of propeller, (Mechanism and control in a strong wind)*. H. Miyagi. JSME Int. J., Series C. 37-1, 172-178 (1994).
- [6] Jiang, Z. *Study on pneumatic wall climbing robot adhesion principle and suction control*. J. Li, X. Gao, N. Fan, and B. Wei. in Robotics and Biomimetics. IEEE International Conference on, ser. ROBIO, Bangkok, Thailand, Feb 2008, pp. 1812 –1817.
- [7] Sangbae, K. *Smooth vertical surface climbing with directional adhesion*. M. Spenko, S. Trujillo, B. Heyneman, D. Santos, M. R. Cutkosky. in Robotics and Biomimetics. IEEE Trans. Robot., vol. 24, no. 1, pp. 65–74, Feb. 2008.
- [8] Prahlad, H. *Electroadhesive robots—Wall climbing robots enabled by a novel, robust, and electrically controllable adhesion technology*. R. Pelrine, S. Stanford, J. Marlow, R. Kornbluh. Proc. IEEE Int. Conf. Robot. Autom., Pasadena, CA, May 19–23, 2008, pp. 3028–3033.
- [9] Yamamoto, Yoshio *Coordinating Locomotion and Manipulation of a Mobile Manipulator*. Philadelphia.
- [10] Nordfeld, Pontus *Decoupler and PID controller design of TITO systems*. Hägglund, Tore. Lund: Lund University, 2008.
- [11] Schlegel, Miloš. *Systémy a modely*. Učební text. Plzeň: Západočeská univerzita, 2010.

REFERENCE

- [12] Melichar, Jiří *Lineární systémy*. Učební text. Plzeň: Západočeská Univerzita, 2010.
- [13] Otsu, Nobuyuki *A threshold selection method from gray-level histograms*. IEEE Transactions on Systems, Man and Cybernetics 9(1), 1979.
- [14] Gonzalez, Rafael C. *Digital Image Processing, Second Edition*. Richard E. Woods. New Jersey: University of Tennessee, 2002. ISBN-10: 0201180758
- [15] Lam, L. *Thinning Methodologies - A Comprehensive Survey*. Lee Seong Whan, Suen Y. Ching. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 14, No. 9, 1992, str. 879
- [16] Gray, A. *Modern Differential Geometry of Curves and Surfaces with Mathematica, 2nd edition* Boca Raton, FL: CRC Press, 1997. ISBN-10: 1584884487
- [17] Freescale Semiconductor. Data Sheet: Technical Data. Digital Barometer MPL115A2 [online]. ©2013 [cit. 2013-04-26] Dostupné z: http://www.freescale.com/files/sensors/doc/data_sheet/MPL115A2.pdf
- [18] REX Controls. Funkční bloky systému REX. Referenční příručka. ©2013 [cit. 2013-04-26] Dostupné z: <http://www.rexcontrols.cz/media/documents/manuals/cz/BRef.CZ.pdf>

Příloha

A Skript předzpracování obrazu

```
1 close all
2 clear all %vycisteni workspace
3
4 trajektorie='trajektorie.jpg'; %navez souboru s
   porizenou trajektorií
5 neupravena = imread(trajektorie); %nacteni
   trajektorie
6
7 im_g=neupravena(:,:,2); %vyber G kanalu z
   RGB
8
9 level = graythresh(im_g); %nalezeni hranice
   pomoci Otsu metody
10 im_thr=im_g<floor(level*256); %prahovani
11
12 se = strel('disk',7); %strukturovací
   element
13 im_o=imopen(im_thr,se); %otevrení (
   morfologická operace)
14
15 im_thin = bwmorph(im_thr,'thin',Inf); %ztenceni
   trajektorie
16
17 figure
18 imshow(im_thin); %vykreslení
19
20
21 imsave %uložení
```