Bakalářská práce

České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra kybernetiky

Experimentální prostředí pro skupinovou robotiku

Matěj Beránek

Vedoucí: Ing. Libor Přeučil, CSc.
Obor: Robotika
Studijní program: Kybernetika a robotika
Květen 2017

Poděkování

Chtěl bych poděkovat vedoucímu práce Ing. Liboru Přeučilovi, CSc. za podporu a zpětnou vazbu při vypracovávání této práce. Dále bych rád poděkoval Ing. Janu Chudobovi, Ing. Martinu Dörflerovi a Dr. Gaël Pierre Marie Écorchardovi za cenné rady po dobu tvoření práce.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, dne .....

.....
podpis autora práce

Abstrakt

Tato práce se zabývá sestrojením experimentálního prostředí pro testování robotických hejn. V práci je shrnuto několik robotických testovacích prostředí a robotických experimentů. Dále je navržen a sestrojen robot se všemi potřebnými komponentami pro řízení a určování polohy. Poté je navrženo několik testovacích situací, které ověří funkčnost algoritmů pro udržování formace robotů. Nakonec je hejno navržených robotů testováno s vybraným algoritmem.

Klíčová slova: hejno robotů, AprilTags, Turtlebot, experimentální prostředí, formace robotů

Vedoucí: Ing. Libor Přeučil, CSc.

Abstract

This thesis focuses on building experimental environment for swarm robotics. This thesis summarizes several robot test environments and a few robotic experiments. Then, a robot is designed and built with all the neccessary components for controlling and getting position. Then several testing situations are designed to verify the functionality of algorithms for maintaining robot formation. Finally, robotic swarm is tested with chosen algorithm.

Keywords: swarm, AprilTags, Turtlebot, robotics testbed, robotic formation

Title translation: Testbed for Swarm Robotics

Obsah

Úvod **1**

1 Experimentální robotika ve světě **3**

1.1 Experimentální robotika obecně . 3

1.2 Experimentální prostředí ve světě 4

1.2.1 NIST 4

1.2.2 Experimentální prostředí pro vesmírná tělesa 5

1.2.3 SyRoTek 6

1.3 Hejno(swarm) - Základní poznatky a motivace pro skupinovou robotiku 8

1.4 Příklad konkrétních robotických experimentů 10

1.4.1 Šestinohý robot 10

1.4.2 Experimentální Lego robot . . 11

1.4.3 S-bots (Selfassessment Robotics) 12

2 Návrh experimentálního prostředí pro skupinovou robotiku **15**

2.1 Realizace reálného robotu - Hardware 15

2.1.1 Turtlebot 2 16

2.1.2 Hokuyo URG-04LX-UG01 – Laser scanner 17

2.1.3 Intel NUC5i5RYK 18

2.1.4 Olověný akumulátor Shimastu NP12-12 19

2.1.5 Basler daA1280-54um - kamera 19

2.1.6 AprilTags 20

2.2 Realizace reálného robotu - Software 21

2.2.1 ROS 22

2.2.2 Celková řídicí smyčka jednotky swarmu 22

2.3 Návrh experimentálního prostředí 23

2.3.1 Návrh vyhodnocování experimentů 24

2.3.2 Matematické aparáty pro vyhodnocování algoritmů 26

2.3.3 Testovací případy 1 - Leader pohybující se po trajektorii přímky 29

2.3.4 Testovací případy 2 - Leader pohybující se po trajektorii křivky 31

3 Reálný experiment **35**

3.1 Parametry provedeného měření . 35

3.2 Jízda robotů po trajektorii přímky 36

3.3 Jízda robotů po trajektorii křivky 37

3.4 Jízda robotů kolem rohové překážky 38

3.5 Jízda robotů kolem sloupu 40

3.5.1 1. testování 40

3.5.2 2. testování 41

3.6 Jízda skrz zúžení 43

3.7 Jízda s překážkou ve tvaru klínu 45

3.8 Jízda s překážkou ve tvaru L . . . 46

3.9 Zhodnocení algoritmu 48

Závěr **51**

3.10 Dosažené výsledky 51

3.11 Další kroky 52

A Literatura **53**

B Zadání práce **57**

Obrázky

1.1 Ukázka pracovního prostředí pro robotiku uvnitř technologického centra NIST [2] 5

1.2 Experimentální prostředí pro vesmírná tělesa [3] 5

1.3 Ukázka robotu a celkový vzhled arény z projektu SyRoTek [4] 7

1.4 Vazby a komunikace mezi jednotlivými komponenty arény SyRoTek [4] 7

1.5 Šestinohý robot se dvěma rameny [5] 10

1.6 Senzory použité na robotu [6] . . 11

1.7 Hotový LEGO robot se senzory [6] 12

1.8 Swarm robotů S-bot pro experimentální selfassessment robotics [7] 13

2.1 Výsledný sestavený robot 16

2.2 Turtlebot s okótovanými rozměry. Převzato z: [9] 16

2.3 Turtlebot - reálný vzhled s doporučeným vybavením z výroby. Převzato z: [10] 17

2.4 Laserový senzor vzdálenosti Hokuyo URG-04LX-UG01. Převzato z: [14] 18

2.5 Počítač Intel NUC5i5RYK. Převzato z: [12] 18

2.6 Kamera Basler daA1280-54um a použitý objektiv 20

2.7 Ukázka různých typů značek AprilTags. Převzato z: [18] 21

2.8 Osmiůhelníkový prstenec se značkami AprilTags [23] 21

2.9 Možné využití robotického swarmu - sklízení pole. Převzato z: [22] 23

2.10 Relativní vzdálenost od vedoucího robotu (d) a relativní vektor polohy $[x, y]$ 25

2.11 Minimální vzdálenost od překážky d_P 25

2.12 Odchylka robotu Δ_R od ideální pozice vůči leaderovi 28

2.13 Testovací případy 1 - Bez překážek 29

2.14 Testovací případy 1 - Sloupová překážka 30

2.15 Testovací případy 1 - Úžina . . . 30

2.16 Testovací případy 1 - Překážka tvaru L 31

2.17 Testovací případy 2 - Bez překážek 32

2.18 Testovací případy 2 - Rohová překážka 32

2.19 Testovací případy 2 - Překážka ve tvaru klínu 33

3.1 Grafy vývoje vzdáleností jednotlivých robotů při jízdě vpřed 36

3.2 Foto z pokusu při jízdě vpřed . . 37

3.3 Grafy vývoje vzdáleností jednotlivých robotů při jízdě se zatáčkami 37

3.4 Grafy vývoje vzdálenosti pravého robotu od vedoucího robotu a od překážky při jízdě kolem rohové překážky 38

3.5 Vývoj vzdálenosti levého robotu od vedoucího robotu 39

3.6 Foto z pokusu při jízdě kolem rohové překážky (nosný sloup budovy) 39

3.7 Grafy vývoje vzdálenosti levého robotu od vedoucího robotu a od překážky při jízdě kolem sloupu . . 40

3.8 Vývoj vzdálenosti pravého robotu od vedoucího robotu 41

3.9 Grafy vývoje vzdálenosti levého robotu od vedoucího robotu a od překážky při jízdě kolem sloupu . . 42

3.10 Vývoj vzdálenosti pravého robotu od vedoucího robotu 42

3.11 Foto z druhého měření objíždění sloupu 43

3.12 Grafy vývoje vzdálenosti levého robotu od vedoucího robotu a od překážky při jízdě zúženým místem 43

3.13 Grafy vývoje vzdálenosti pravého robotu od vedoucího robotu a od překážky při jízdě zúženým místem 44

3.14 Foto z pokusu projetí zúženého místa 44

3.15 Grafy vývoje vzdálenosti pravého robotu od vedoucího robotu a od překážky při jízdě na překážku ve tvaru klínu 45

3.16 Grafy vývoje vzdálenosti levého robotu od vedoucího robotu a od překážky při jízdě na překážku ve tvaru klínu 45

3.17 Foto z pokusu projetí kolem překážky ve tvaru klínu 46

3.18 Grafy vývoje vzdálenosti pravého robotu od vedoucího robotu a od překážky při jízdě na překážku ve tvaru klínu 47

3.19 Vývoj vzdálenosti levého robotu od vedoucího robotu 47

3.20 Foto z pokusu z překážkou ve tvaru L 48

Tabulky

2.1 Parametry řídicího počítače Intel NUC5i5RYK 19

Úvod

Kvůli náročnosti některých činností, které jsou pro člověka nezvladatelné, je nutné si pomáhat pomocí robotů. Neustále vznikají nové úlohy, které musí roboty řešit. S nimi přicházejí nová řešení, která je z důvodu uvedení do reálného provozu nutné otestovat. Kvalitní testování musí být provedeno pro všechny druhy robotů - od obrovských průmyslových robotů až po malé nanoboty.

Práce je zaměřena hlavně na pozemní mobilní roboty. Mobilní roboty jsou schopny dosáhnout větší flexibility než pevně připevněná chapadla. V průmyslu se často setkáváme s nebezpečnými nebo špatně dosažitelnými místy pro člověka. V těchto situacích lze využít mobilní roboty. Čím dál více problémů se nachází mimo schopnosti jednotlivých mobilních robotů, proto se robotika začíná zabývat jejich skupinami. Skupina více robotů spolupracující na jednom úkolu se stává výkonnější jednotkou, než individuální robot se spoustou funkčností. Proto se zavádí nový pojem – swarm, neboli hejno robotů.

Pro potřeby testování robotického hejna není ve světě mnoho technik. Swarm robotů je sám o sobě velmi flexibilní a je možné ho využít prakticky v jakékoliv situaci. Je proto potřeba si uvědomit, jakou vlastnost daného swarmu potřebujeme konkrétně testovat.

Tato práce se zabývá návrhem testování schopností algoritmu udržet formaci a přizpůsobit ji případným překážkám. Následně je testování provedeno na vybraném algoritmu. Pro toto experimentální prostředí je nutné vybrat vhodné roboty pro testování, tedy hardware i software jednotlivého člena swarmu. Poté je nutné vybrat situace a navrhnout měřené veličiny, které dostatečně ohodnotí schopnost udržet požadovanou formaci.

První Kapitola se zabývá především teorií robotického swarmu, testovacími arénami a prostředími používanými ve světě, a několika příklady konkrétního testování a měření.

Druhá část práce rozebírá samotný návrh testovacího prostředí. První částí této kapitoly je popis konkrétního robotu použitého při testování. Je zde popsán hardware i software použitý k vytvoření jednoho člena robotického hejna. Druhá část kapitoly se věnuje celkovému návrhu testování, tedy jaké veličiny měřit, jakým způsobem je měřit a vyhodnotit, a jaké testovací případy pro tento účel sestavit.

Třetí část práce popisuje reálný experiment na testovacím systému navrženém v Kapitole 2. Je zde realizován návrh takového systému a v něm je testován swarm dvou navržených robotů. Dále jsou v této kapitole uvedeny výsledky měření a následné vyhodnocení použitého algoritmu z hlediska jeho spolehlivosti při udržování formace.

Kapitola 1

Experimentální robotika ve světě

1.1 Experimentální robotika obecně

Kvůli potřebě testovat a zdokonalovat vyvinuté roboty a mechanismy vzniká spousta testovacích laboratoří a arén, které pomáhají testovat jednotlivé vlastnosti robotů a měřit je. Abychom mohli roboty testovat po celém světě, bylo vytvořeno několik standardů, které určují jak kterou vlastnost měřit, jaké prostředí pro testování postavit. Díky těmto standardům jsme schopni roboty otestovat, změřit jejich vlastnosti a ty následně kvantitativně vyhodnocovat. Pokud bylo prostředí sestaveno podle vybraného standardu, následné testování správně změřeno a vyhodnoceno, jsou tyto výsledky relevantní po celém světě a mohou být srovnávány s ostatními výsledky pořízenými za stejných podmínek.

Příkladem návrhu, jak správně otestovat mapovací algoritmy robotů, je článek od vědců z Itálie [1]. Důležité pro kvalitní experiment je jeho možná opakovatelnost a vyhodnotitelnost. Pro kvalitní opakování experimentu je potřebné

- Testovaný systém musí být testován na veřejně přístupných datech
- Jsou-li potřebné určité parametry pro nastavení algoritmu a robotů, musí být uvedeny jejich konkrétní hodnoty při provedeném experimentu
- Měly by být uvedeny všechny výsledky, nejen ty dobré - tímto je možno lépe pochopit testovaný systém.

Pro správné vyhodnocení je nutné dodržet následující :

- Uvést všechna data o mapovaném prostředí - velikost, rozložení atd.
- Testovat systém pro různé hodnoty parametrů algoritmu a robotů - zde je možné vyhodnotit robustnost systému
- Uvést chování robotu, pokud by pokus vedl k nekonečnému zacyklení - opět se hodnotí robustnost systému
- Uvést přesnost mapovaného obrazu oproti reálnému prostředí - hodnotí se přesnost systému

1.2 Experimentální prostředí ve světě

1.2.1 NIST

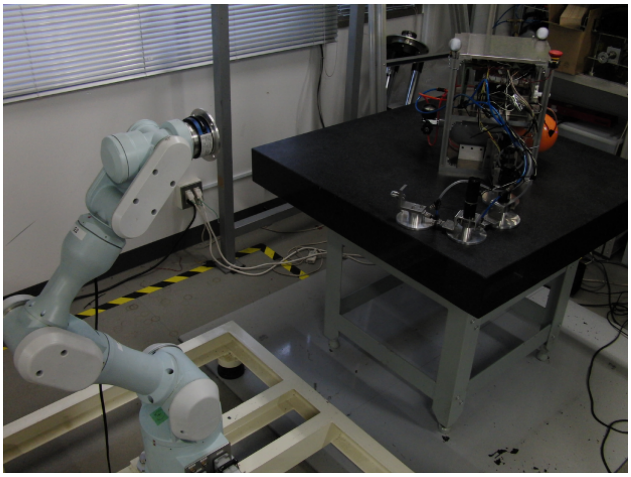
National Institute of Standards and Technology [2] je obrovské technologické centrum v USA, které se zabývá všemi obory v oblasti technologie, robotiku nevyjímaje. Centrum pod svým jménem vydává obrovské množství vědeckých článků ze všech technologických oborů, ať už se jedná o konkrétní technologické výzkumy či obecné popisy standardních postupů měření různých veličin. Velká část je zaměřena na standardizované testování robotů. Experimentální prostředí tu zahrnují například testování "search and rescue" v oblastech podobných městskému prostředí, zneškodnění bomby, různých vojenských pozemních operací a další. Všechny metody měření jsou založeny na požadavcích profesionálů, kteří budou tyto roboty následně využívat v praxi. Testované schopnosti, na které se institut zaměřuje, jsou :

- Mobilita robotu
- Manipulace s okolím
- Interakce mezi člověkem a robotem
- Robotická interakce s okolím
- Síla a výdrž pohonu robotu

Obrázek 1.1: Ukázka pracovního prostředí pro robotiku uvnitř technologického centra NIST [2]

Experimenty jsou prováděny na ploše o velikosti 1524 m^2, kde jsou k dispozici všechna výše zmíněná prostředí pro testování všech kritických vlastností robotu.

1.2.2 Experimentální prostředí pro vesmírná tělesa



Obrázek 1.2: Experimentální prostředí pro vesmírná tělesa [3]

Pokud je vyžadováno experimentální prostředí zaměřené na jedno konkrétní využití, nemusí být toto prostředí nutně obrovské. V Japonsku vznikla experimentální aréna [3] pro testování vesmírných přístrojů, tzn. družic, satelitů atd., která nepotřebuje obrovské letištní haly k plné funkčnosti. Využívá principu proudění vzduchu, kterým se dostatečně kvalitně simuluje stav beztíže. V tomto laboratorním prostředí lze měřit moment sil, hybnost, rychlost a další charakteristické veličiny robotu, který se nachází ve stavu beztíže. K provádění pokusů je zde využito dvou manipulátorů. První je pevně připev-

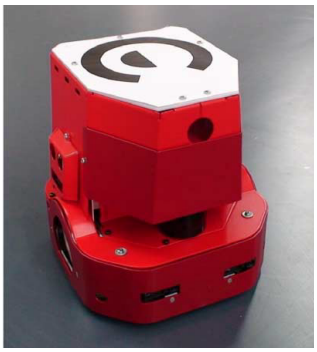
něn k zemi. Druhý manipulátor je připevněn na testovací konstrukci, která má simulovaný stav beztíže. Při experimentech se testuje interakce obou manipulátorů a měří se charakteristické veličiny druhého z nich. Do testovací konstrukce je možno připevnit jakýkoliv manipulátor, se kterým se následně mohou provádět operace potřebné ve vesmíru.

1.2.3 SyRoTek

Na univerzitě ČVUT v Praze, Fakultě elektrotechnické, vznikl projekt SyRoTek [4]. Jedná se o robotickou arénu, která je se všemi svými komponentami přístupná online na webových stránkách fakulty. Cílem tohoto projektu bylo zpřístupnit reálný hardware studentům, vědeckým pracovníkům školy a přidružené univerzitě v Buenos Aires. Projekt byl inspirován několika projekty, ve kterých bylo zpřístupněno přes internet ovládání experimentálních robotů, žádný z nich však neposkytoval mobilní roboty, vybavené velkým počtem senzorů, v aréně s překážkami, na kterých se tak dají testovat algoritmy pro vyhýbání se překážkám (collision avoidance), plánování trasy nebo třeba mapování okolí.

Aréna je vybavena jednou lokalizační kamerou, která identifikuje jednotlivé roboty na ploše arény a poskytuje informace o jejich absolutní poloze. Dále jsou zde 3 kamery pořizující videozáznam o dění v aréně, aby měl uživatel ovládající roboty přehled o dění v aréně. Robot použitý v aréně se nazývá S1R, skládá se z podvozku, ovládacího počítače a laserových a ultrazvukových senzorů pro měření vzdálenosti od okraje arény a od roviny arény. Navíc je robot připraven pro zapojení dalších externích senzorů. Na horní straně robotu je umístěna značka, kterou je schopna lokalizační kamera rozpoznat a určit z ní přesnou polohu robotu i jeho natočení. Dále je aréna vybavena pohyblivými překážkami, které může uživatel také ovládat ze svého počítače - je tedy možné si arénu přizpůsobit pro vlastní testovací potřeby.

Komunikace s arénou je zajištěna přes hlavní počítač (server). Server zajišťuje komunikaci s uživatelem přes internet. Zároveň přes WiFi komunikuje s jednotlivými roboty v aréně a posílá jim příkazy od uživatele. Pokud se uživatel od serveru odpojí, počítač vyšle příkazy robotům pro zajetí zpět do dokovací stanice. Na tomto počítači jsou také uloženy a zálohovány všechny kódy od koncových uživatelů, je tedy jednoduché sledovat vývoj algoritmu nebo spolupracovat v týmu s více lidmi. K hlavnímu počítači je také připojen

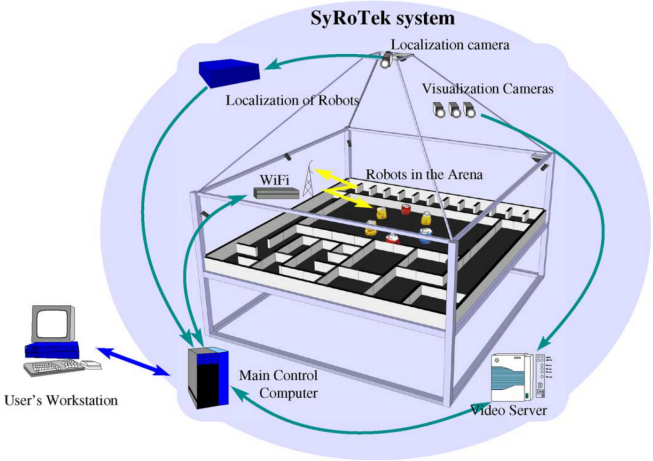


(a) : Robot S1R

(b) : Experimentální aréna SyRoTek

Obrázek 1.3: Ukázka robotu a celkový vzhled arény z projektu SyRoTek [4]

Videoserver, který zpracovává video z kamer nad arénou, a poskytuje je hlavnímu počítači. Ten video posílá přímo koncovému uživateli, který tak má přehled o dění v aréně v reálném čase.



Obrázek 1.4: Vazby a komunikace mezi jednotlivými komponenty arény SyRoTek [4]

Prostředí připravené ke komunikaci s roboty je Player/Stage, což je hojně využívaný rámec (framework) pro ovládání robotů. Stage je simulátor, na kterém se může implementovaný kód vyzkoušet a ověřit. Pokud je uživatel spokojen s výsledkem simulace, může stejný kód pomocí Player prostředí poslat přímo reálným robotům. Jako druhá možnost připojení k robotům je operační systém ROS, kde jsou všechny roboty se senzory připraveny k použití jako uzly (viz podkapitola 3.2.1).

K připojení k hlavnímu počítači se v této aréně mohou využít tři způsoby.

- Přes webové stránky fakulty, kde má uživatel k dispozici videa z arény, podpůrné publikace, články a prezentace sloužící jako podpora při vytváření různých algoritmů.
- Přes Netbeans IDE. V tomto IDE je připraveno prostředí Player/Stage, všechen output je přesměrován na výstup IDE v počítači uživatele, takže má uživatel přístup k uceleným informacím o aréně a jejím stavu.
- Přes příkazový řádek Unixového stylu. Jedná se ale o přístup pro zkušenější uživatele, jelikož příkazový řádek neposkytuje žádnou podporu pro koncového uživatele.

V aréně bylo provedeno již nespočet různých experimentů a měření, je využívána například přímo v předmětu Praktická robotika, kde studenti využívají všech možností arény pro plnění zadaných úkolů. Projekt SyRoTek samozřejmě nemusí využívat jen studenti tohoto předmětu, byly na něm testovány projekty v rámci bakalářských i diplomových prací, nebo ho využívají přímo vědečtí pracovníci ke své odborné činnosti. Koncept arény SyRoTek byl také inspirací pro tuto bakalářskou práci, ať už výběrem podobného typu robotu, senzorů, nebo výběrem situací pro testování vlastních robotů.

1.3 Hejno(swarm) - Základní poznatky a motivace pro skupinovou robotiku

Swarm robotů je velice důležitý pojem v oboru řízení robotů. Z hlediska řízení se jedná o jeden z novějších přístupů, který se neustále zdokonaluje. Tento přístup je založen na pozorování různých zvířat a analýze jejich chování ve skupině. Swarm jako takový můžeme pozorovat například u ryb, ptáků, či hmyzu. To je cíl, kterého se robotika snaží dosáhnout – roboty, které společně ve swarmu zvládají řešit složité problémy, které by sami řešit nemohli. Důležitá je proto interakce mezi jednotlivými členy swarmu a také interakce s prostředím, ve kterém se swarm nachází. Tato interakce může být rozdělena na pasivní a aktivní.

Pasivní interakce robotů, ať už s prostředím či s ostatními členy formace, bývá zajištěna pomocí různých senzorů. V případě prostředí se k interakci většinou používají senzory vzdálenosti, jimiž jsme schopni rozpoznat blížící se překážku nebo nerovnost terénu. Nevýhodou těchto senzorů je přesnost a šum, hovoříme-li o senzorech ultrazvukových, nebo cena, jedná-li se o senzory

laserové. Pro interakci mezi jednotlivými členy můžeme použít senzory obrazové, potřebujeme-li více informace, než pouze binární informaci o výskytu neznámého objektu. V případě použití kamery je potřeba využít sofistikovanější algoritmy, které jsou schopny z videa či obrázku sestavit potřebnou informaci, například o tom, zdali se v zorném úhlu nachází jiný člen formace, či nikoliv.

Aktivní interakce členů je zajištěna síťovou komunikací. Roboty tedy musí být vybaveny zařízením, které je schopno síťovou komunikaci vytvořit a udržovat. Tato komunikace se pak využívá pro sdílení důležitých informací o poloze ostatních robotů a poloze překážek, nebo jestli je nějaký člen ztracen bez kontaktu s ostatními a bez možnosti pokračovat v práci. V tomto případě je možné použití systému GPS, kdy každý člen má informace o své poloze, kterou je pak schopen sdělit ostatním.

Samotných algoritmů pro udržování swarmu je několik.

- **Neudržovat pevnou formaci** - V tomto případě není důležité udržovat přesně danou formaci, ve které se swarm pohybuje. Swarm se může pohybovat zdánlivě chaoticky, mezi členy mohou být pouze vazby stejné, jako pro překážky, aby nedocházelo ke kolizím. Tento typ swarmu je možné využít třeba při přesunu na dané místo.

  Konkrétním příkladem by mohly být roboty, které se dokáží spojit do většího robotu. Zde při přesouvání na konkrétní místo není potřeba dojet ve formaci, důležité je pouze se na místo dostat.

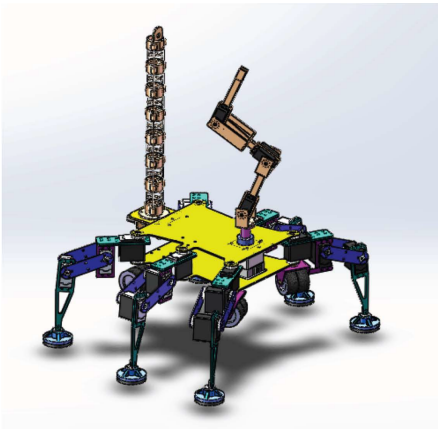
- **Udržovat pevnou formaci - Pružinový algoritmus [24]** - Pokud je potřeba udržet swarm v dané formaci, využívá se znalostí fyziky pro zajištění vazeb mezi jednotlivými roboty. V tomto případě se mohou vazby vytvořit pomocí virtuálních pružin. Robot si ke členům swarmu vytvoří pružiny, které mohou mít různé tuhosti podle potřeb swarmu, a podle těchto pružin se pak dokáže pohybovat. U tohoto přístupu je nepříjemné hlavně to, že pokud je pružin ve formaci velké množství, celá tato formace se stává nestabilním systémem a mohlo by dojít k narušení formace.

- **Udržovat pevnou formaci - Algoritmus využívající potenciálového pole [24]** - Využijeme-li pro udržování formace potenciálová pole, můžeme na konkrétních pozicích vytvářet potenciálové jámy, které udržují člena na správném místě. K vytvoření jámy je potřeba pouze pozice, kde má být umístěna. Je tedy velice jednoduché potenciálová pole implementovat. Tato pole zajišťují mnohem větší stabilitu celé formace a celkově lepší a přesnější řízení jednotlivých robotů.

1.4 Příklad konkrétních robotických experimentů

1.4.1 Šestinohý robot

Výzkumníci v Číně [5] vyzkoušeli nový přístup v mobilní robotice. Jelikož se čím dál častěji setkáváme s prostředími, kde není možné projet robotem s koly, vědci využili obdobu "robotického pavouka", jako základní platforma pro mobilní robot. Výhodou robotu je také možnost využít kola na spodní straně svého těla v případě rovného terénu, pokud nohy otočí směrem nad sebe. Robot je dále vybaven dvěma rameny, kde jedno je obecný 4-DOF manipulátor s chapadlem na jeho konci, a druhé funguje jako stativ pro kameru. Druhé rameno není ovládáno obyčejnými klouby, uvnitř ramene je pouze několik drátů. Pomocí stahování jednotlivých drátů je pohybováno celým ramenem.

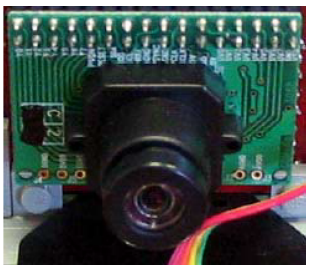


Obrázek 1.5: Šestinohý robot se dvěma rameny [5]

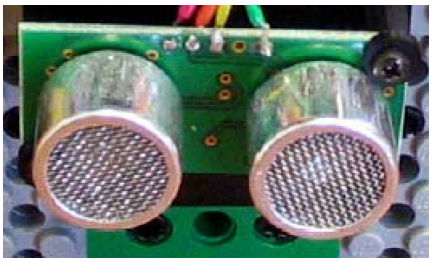
Pro tento robot bylo navrženo několik experimentů, nejzajímavější samozřejmě testování mobility robotu v nerovném terénu. Bylo umístěno několik překážek, testovalo se jakou rychlostí je robot schopen se pohybovat, aniž by ztratil rovnováhu. Zajímavé bylo testování i pohybu na kolech, když byly překážky odstraněny. Několik pokusů ještě otestovalo schopnost přesné manipulace hlavním ramenem, když je robot postaven na nohách a není tedy stoprocentně stabilní.

1.4.2 Experimentální Lego robot

Na univerzitě v Louisville se vědci pokusili sestavit kvalitní experimentální robot [6] pomocí stavebnice LEGO Mindstorms Robotic Invention System 2.0 kit. Touto stavebnicí je pouze sestaveno šasi robotu, k jeho ovládání je použita deska Handy Board. Robot je vybaven optickým senzorem CMUcam a ultrazvukovým senzorem vzdálenosti SRF04. Hlavním důvodem pro sestrojení tohoto robotu bylo připravit studentům univerzity robot, na kterém mohou zkoušet své algoritmy pro sledování objektu a collision avoidance.



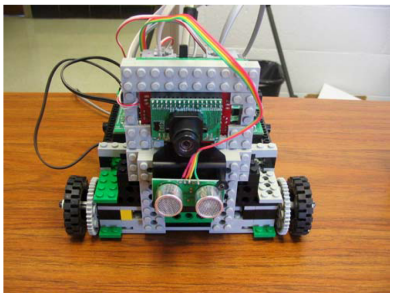
(a) : Kamera CMUcam



(b) : Senzor SRF04

Obrázek 1.6: Senzory použité na robotu [6]

Pro robot byl vyvinut algoritmus pro sledování oranžového předmětu. Pro ověření funkčnosti algoritmu bylo provedeno několik testování. První pokus testoval kamery a schopnost robotu sledovat předmět. Ultrazvukový senzor byl v tomto testu vypnut, ale robot dokázal sledovat předmět bez problémů. Pro další test byl zapnut ultrazvukový senzor vzdálenosti, který tomuto sledování přidává třetí rozměr (vzdálenost sledovaného objektu od robotu). Sledování bylo rychlejší a přesnější, bylo tedy vyhodnoceno jako úspěšné. Poslední testovaný případ byl navrhnut pro testování schopnosti robotu vyhnout se průhledné překážce. Průhledná překážka byla v tomto případě zvolena tak, aby bylo možné oranžový objekt skrz ni stále pozorovat, ale ultrazvukový senzor ji je schopen detekovat. I závěrečný test byl úspěšný, vědci došli k závěru, že robot je použitelný pro další možná testování.



Obrázek 1.7: Hotový LEGO robot se senzory [6]

1.4.3 S-bots (Selfassessment Robotics)

Výzkumníci ze Švýcarska a Belgie zpracovali problém hejna robotů [7] z pohledu schopnosti propojit jednotlivé členy swarmu do většího a funkčního celku, který následně zvládá řešit úkoly, na které každý člen samostatně nemá prostředky a schopnosti. Například, objeví-li se před tímto hejnem propast, kterou není možné nějakým způsobem objet nebo se jí vyhnout a pokračovat jiným směrem, roboty se spojí do větší pevně propojené skupiny a díky tomu propast snadno přejedou. Článek se velmi detailně zabývá pouze problematikou samotného propojování robotů.

Pro testování byly využity roboty s-bots [8], které mají využití především v laboratorních experimentech, hlavně kvůli jejich velikosti (průměr 116 mm), která dovoluje experimentovat s větším počtem robotů na malé pracovní ploše. Důležitá vlastnost S-bots je jejich svítící obal po celém obvodu těla, který dokáže měnit barvy, čehož je využito při identifikaci ostatních členů z obrazu kamery. V neposlední řadě je každý S-bot vybaven "chapadlem", kterým se S-bot může přichytit dalšího robotu a vytvořit s ním pevné spojení. S-bot je navíc vybaven spoustou senzorů vzdálenosti.

Základní myšlenka celého skládání spočívá právě v různobarevných znameních, která roboty hledají pomocí omnikamery (důležitá je barva obrazu, není tedy potřeba přesné zobrazení z kamery). Pokud tělo jiného robotu svítí červeně, značí to, že je připraven k propojení s jiným robotem, svítí-li robot modře, znamená to, že se k němu mají ostatní chovat jako k překážce a nemají se k němu napojovat. Dostane-li se již S-bot na správnou vzdálenost k červeně svítícímu robotu či tělesu (vzdálenost detekuje senzory vzdálenosti), pokusí se k němu přichytit svým chapadlem. Chapadlo (gripper) je vybaveno senzorem světla na jedné čelisti a LED diodou na té druhé, před sevřením čelistí se tedy změří hodnota ze senzoru při vypnuté i zapnuté LED diodě. Pokud jsou tyto hodnoty stejné, znamená to, že v čelistech se něco nachází, jelikož rozsvícená dioda neovlivnila hodnotu ze senzoru. Díky tomuto ověření se zamezuje špatnému napojení. Jakmile se robot připojí, rozsvítí své tělo červeně a ostatní roboty se k němu mohou připojovat. Velká výhoda tohoto

swarmovacího algoritmu je použití stejného kódu pro všechny roboty, žádný není nadřazený ostatním a tímto způsobem je jednoduché experimentovat s větším počtem robotů.

(a) : S-bot s chapadlem **(b) :** Swarm robotů S-bots

Obrázek 1.8: Swarm robotů S-bot pro experimentální selfassessment robotics [7]

Celý tento algoritmus byl ověřen v simulaci, než byl swarm vyzkoušen v praxi. Pro vyzkoušení tohoto swarmu v reálném prostředí bylo navrženo několik měření. Nejprve byl testován samotný S-bot a jeden statický objekt svítící červeně. Pokus byl prováděn pro různé počáteční úhly robotu vůči tomuto objektu a pro různé vzdálenosti od objektu. Z jakéhokoliv počátečního stavu se robot úspěšně připojil k testovanému objektu. Další série pokusů byla stejná, pouze statický objekt byl nahrazen statickým robotem. Samotný S-bot totiž neumožňuje připojení na 100% svého těla, jsou podmínky pro připojení jsou tedy obtížnější. I tato série pokusů ale byla vyhodnocena se 100% úspěšností. Pro poslední a také nejkomplexnější testování bylo použito 16 robotů, z nichž se jeden stal po nějaké době statickou základnou a ostatní se díky tomu měli postupně propojit. Z vyhodnocení tohoto testování vyšlo, že 97.55% propojení bylo provedeno, jen někdy se robot nebyl schopen připojit, protože se zasekl a nemohl se dostat na požadované místo.

Všechny výše zmíněné experimenty pak byly provedeny i v nerovném terénu. Terény byly použity dva - nerovnosti v něm byly rovnoměrně a nerovnoměrně rozmístěny. Pokaždé však úspěšnost připojení přesahovala hranici 90%, což vědci považují za úspěch a plánují s tímto projektem pokračovat.

Kapitola 2

Návrh experimentálního prostředí pro skupinovou robotiku

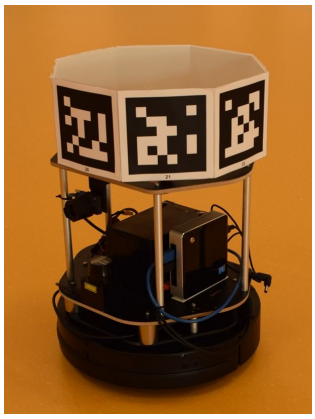
2.1 Realizace reálného robotu - Hardware

Pro realizaci člena formace pro experimentální skupinovou robotiku je nutné vybrat malý, mobilní robot, který bude mít dostatečnou nosnost, aby uvezl ostatní prvky. Těmito prvky jsou myšleny především hlavní počítač a senzory, které budou na robotu umístěny. Jak je zmíněno výše v práci, roboty při skupinové robotice by měly být schopny pasivní i aktivní interakce s prostředím a ostatními členy, z tohoto důvodu bude na robotu umístěna kamera, laserový dálkoměr a počítač s napájecí baterií. Počítač musí být kompaktní a zároveň dostatečně výkonný, aby dokázal pracovat na operačním systému Ubuntu. Dále by měl dokázat pracovat s robotickou nadstavbou ROS a zároveň by neměl vyžadovat připojení dalších zařízení pro navázání bezdrátové komunikace. Pro potřebu rozpoznávání kamerou je na robotu připevněn osmiúhelníkový prstenec se značkami AprilTags.

Na základě těchto požadavků byl sestaven robot pomocí následujících komponent:

- Turtlebot 2 (robot) [9]
- Intel NUC (počítač s operačním systémem, ovládá všechny ostatní komponenty) [11]
- Hokuyo URG-04LX-UG01 (laserový dálkoměr) [13]

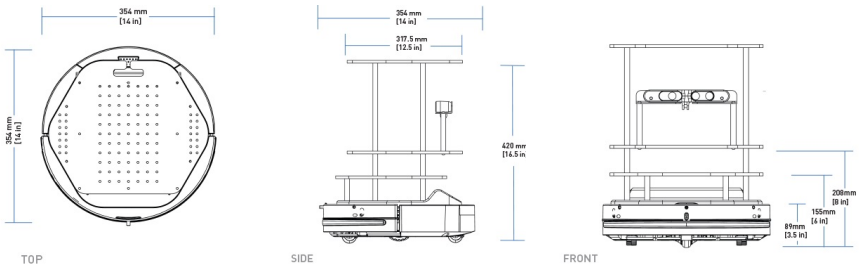
- Basler daA1280-54um (kamera) [15]
- Olověná baterie



Obrázek 2.1: Výsledný sestavený robot

2.1.1 Turtlebot 2

Jako robot byl vybrán Turtlebot 2. Tento robot je velmi často používán pro laboratorní experimenty, hlavně kvůli jeho cenové dostupnosti a malým rozměrům. Jelikož byl Turtlebot vyvíjen při spolupráci s firmou Willow Garage, která se podílela na vývoji robotického operačního systému ROS, je plně kompatibilní s tímto systémem. Díky této spolupráci je Turtlebot již při zakoupení připraven s integrovanou verzí ROS - je velmi jednoduché a intuitivní navázat spojení přes sběrnici USB s externím počítačem, kterým je robot ovládán.



(a) : Pohled shora **(b) :** Pohled z boku **(c) :** Pohled zepředu

Obrázek 2.2: Turtlebot s okótovanými rozměry. Převzato z: [9]

Turtlebot má válcovitý tvar s průměrem 354 mm a výškou 89 mm. Jeho váha činí cca 6.3 kg a je schopen uvézt dalších 5 kg bez jakýchkoliv změn jeho vlastností. Mobilitu robotu zajišťují dva motory pro nezávislý pohyb obou

kol. Díky umístění kol na stranách robotu se může otáčet kolem své vlastní osy, což umožňuje vysokou pohyblivost, kterou normální čtyřkolové roboty nedisponují. Robot je schopen pohybovat se rychlostí až 0.65 m/s. V příslušenství robotu se dále nachází tři patra a uchycovací šroubky s podpěrami, na které se umísťují všechny ostatní komponenty. Robot má vepředu zabudované tři vlastní senzory tlaku, tzv. bumpery, dále je možné využít "wheel drop" senzorů, které jsou schopny detekovat změnu výšky terénu, po kterém se Turtlebot pohybuje(např. schody). Robot je vybaven baterií s kapacitou 2200 mAh a čtyřmi elektrickými konektory (5 V (1 A), 19 V (1 A), 12 V(1.5 A) a 12 V(5 A)). Bohužel jsou tyto konektory napájeny pouze ve chvíli, kdy je Turtlebot připojen adaptérem k síti. Kvůli této komplikaci je nutné počítač napájet z externí baterie.

Obrázek 2.3: Turtlebot - reálný vzhled s doporučeným vybavením z výroby. Převzato z: [10]

2.1.2 Hokuyo URG-04LX-UG01 – Laser scanner

Pro detekci překážek musel být vybrán vhodný senzor vzdálenosti. Na robotu je použit laserový dálkoměr Hokuyo URG-04LX-UG01, který je v porovnání s konkurencí cenově dostupný, kompaktní a lehký (robot uveze pouze 5kg nákladu). Senzor komunikuje s počítačem přes rozhraní USB, konkrétně využívá konektor Mini USB. Tento senzor je navíc přes toto rozhraní napájen, což je velkou výhodou mezi konkurencí.

Senzor rozmítá laserový paprsek v záběru 240°, s krokem 360°/1024 (cca

0.352°). Každý scan je proveden v časovém intervalu 100 ms, což je pro reakci na překážku v reálném čase dostačující. Senzor má dosah až 5.6 metru, jeho přesnost se pohybuje kolem 30 mm do vzdálenosti 1 metru, poté chyba poměrně narůstá s vzdáleností (3% z naměřené vzdálenosti), jsou-li detekční podmínky ideální, tzn. detekovaná překážka není ze skla nebo jiného materiálu špatně odrážejícího paprsky.

Obrázek 2.4: Laserový senzor vzdálenosti Hokuyo URG-04LX-UG01. Převzato z: [14]

Všechny senzory Hokuyo mají navíc plnou podporu v operačním systému ROS, komunikace s tímto senzorem je tedy jednoduchá.

2.1.3 Intel NUC5i5RYK

Obrázek 2.5: Počítač Intel NUC5i5RYK. Převzato z: [12]

Pro řízení robotu i senzorů byl vybrán počítač Intel NUC s operačním systémem Linux. Počítač je celkově velmi malý a zároveň výkonný, což je hlavní důvod k výběru právě tohoto počítače. Počítač je vybaven čtyřmi USB porty 3.0, což je dostačující pro připojení laserového senzoru, kamery a robotu. Navíc je jeden port volný pro případné připojení další kamery či

senzoru. Dále se díky portu mini HDMI lze připojit k monitoru. Počítač je napájen standardními 19 V DC, napájen je externí baterií dodávající 12 V DC s měničem napětí.

Specifikace	
Procesor	Intel Core i5-5250U
RAM	DDR3L SODIMM 16 GB
Grafická karta	Intel HD Graphics 6000
Porty	4x USB 3.0
Síť	Intel Wireless-AC 7265, antény(IEEE 802.11ac, Bluetooth)

Tabulka 2.1: Parametry řídicího počítače Intel NUC5i5RYK

2.1.4 Olověný akumulátor Shimastu NP12-12

Pro napájení počítače je využit olověný akumulátor od firmy Shimastu. Tento akumulátor je schopen dodávat napětí 12 V, měničem napětí je dosaženo hodnoty 19 V pro požadované napájení počítače. Nevýhodou akumulátoru je větší váha 3.5 kg. Nedochází zde ovšem k poklesům poskytovaného výkonu, což může zapříčinit náhlé vypnutí ovládacího počítače. Kapacita tohoto akumulátoru je 12 Ah.

2.1.5 Basler daA1280-54um - kamera

Pro rozpoznávání obrazu je na robotu připevněna kamera Basler daA1280-54um vybavená objektivem s ohniskovou vzdáleností 4 mm. Kamera nevyžaduje externí napájení, napájení i veškerá komunikace je vedena skrze sběrnici USB 3.0. Kamera poskytuje rozlišení až 1280x960 pixelů. Obraz z kamery je pouze černobílý, ale při použití značek AprilTags toto omezení nemá žádný vliv na požadovaný výsledek rozpoznávání. Robot může být vybaven více kamerami tohoto typu, pokud je požadováno sledovat více členů formace a ne pouze vedoucí robot. Kamera s výše zmíněným objektivem má totiž zorný úhel pouhých 70°.

(a) : Basler daA1280-54um. Převzato z: [16]

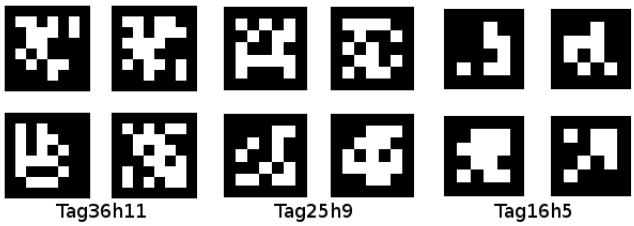
(b) : Objektiv kamery. Převzato z: [23]

Obrázek 2.6: Kamera Basler daA1280-54um a použitý objektiv

Pro ovládání kamery je použit software Pylon od firmy Basler. Tento software zahrnuje veškeré ovladače kamer Basler a je tedy zajištěna plná funkčnost kamery. Navíc je tento software přístupný k úpravám a koncový uživatel si tedy může upravovat vlastnosti podle vlastních potřeb.

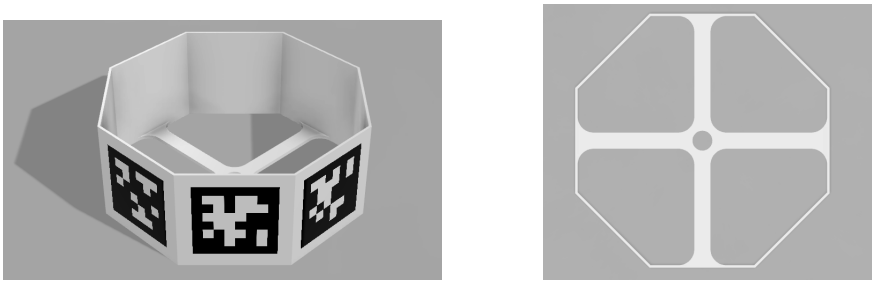
2.1.6 AprilTags

Pro potřeby rozpoznávání pomocí vybrané kamery je také potřeba zavést značení, které lze jednoduše identifikovat v obrazu z kamery. K tomuto účelu byly vybrány značky AprilTags [17], které fungují na podobném principu jako QR kódy. AprilTags zajišťují 6DoF identifikaci jako několik dalších pasivních značek pro vizuální rozpoznávání. Dle [23] značky AprilTags vycházejí pro naši potřebu jako nejvhodnější.



Obrázek 2.7: Ukázka různých typů značek AprilTags. Převzato z: [18]

Pro náš konkrétní případ jsou značky rozmístěny na prstenci ve tvaru osmiúhelníku [23], na každé stěně tohoto prstence je umístěn jeden tag s unikátní adresou. Prstenec je umístěn na všech členech swarmu včetně leadera, díky čemuž je jeden člen schopen rozeznat nejen polohu člena jiného, ale také jeho natočení vůči sobě.



(a) : Pohled ze strany **(b) :** Pohled shora

Obrázek 2.8: Osmiůhelníkový prstenec se značkami AprilTags [23]

2.2 Realizace reálného robotu - Software

Jako operační systém počítače je vybrán systém Unixového typu, konkrétněji Ubuntu 14.04. Novější verze Ubuntu mají problémy s funkčností systému ROS, proto byla zvolena stabilnější verze OS. Pro zpracování dat z kamery je potřeba použít knihovny OpenCV a knihovny Basler Pylon. Jak je již v této práci zmíněno, veškerou obsluhu robotu i senzoru vzdálenosti zařizuje systém ROS. V našem případě je použita verze "Indigo", protože podporuje komunikaci s knihovnami OpenCV. Řídicí kód je psán v jazyce C++.

2.2.1 ROS

ROS je operační systém od firmy Willow Garage zajišťující komunikaci robotů s jejich řídicími počítači a senzory, jejich synchronizaci a minimální chybovost přenosu dat [19]. Komunikace je založena na principu uzlů. Každá část robotu(motory, vestavěné senzory), externí senzory a další členy komunikace tvoří svůj vlastní uzel grafu, kterému jsou přiřazeny komunikační hrany (topics), po kterých mohou být posílána data. Tato síť pak funguje na principu posílání zpráv (messages) z uzlu do uzlu, kde každý z nich musí umět tuto zprávu přeložit a zpracovat z ní data. Například se do zprávy mohou zakódovat rychlosti motorů, kterými by se měl robot pohybovat, tato zpráva je poté odeslána do uzlu obsluhujícímu motory, který následně tuto zprávu přeloží a přiřadí motorům požadovanou rychlost.

2.2.2 Celková řídicí smyčka jednotky swarmu

Celkový program pro řízení jednotky swarmu je psán v jazyku C++, který využívá knihoven ROS pro komunikaci s Turtlebotem. Pro zpracování obrazu je použit program pro rozpoznávání značek AprilTags a jejich následné vyhodnocení [23]. Samotný pseudokód algoritmu (Algorithm 1) je uvedený zde.

Algorithm 1: Řídicí algoritmus robotu

1 initialization, start session with all ROS nodes;
2 run new thread with AprilTags recognition; **while** *everything ok, robot and sensors communicate* **do**
3 cameraVals ← load camera values vector;
4 laserVals ← load laser values vector;
5 $[v_l, v_a]$ = f(cameraVals, laserVals) ;
6 Send $[v_1, v_2]$ to robot control center;
7 **end**

Program spuštěný na řádku 2 automaticky vytváří fotky z kamery. Po vytvoření fotky se tento obraz zpracuje pomocí AprilTags detektoru, který nalezne značku AprilTag v obraze, pokud se tam vyskytuje. Po rozpoznání obrazu uloží všechny hodnoty k rozpoznanému AprilTagu (x, y, z, yaw, pitch, roll, distance, ID) do globální struktury, kterou následně předá řídicímu programu. Jedna tato aktualizace trvá asi 0.3 s, vytváření fotky z obrazu kamery je časově náročný proces.

Funkce na řádku 5 je již samotný algoritmus udržení formace [24], který převádí vstupní vektory hodnot z kamery a z laserového dálkoměru na jednotlivé rychlosti robotu. Rychlost v_l je rychlostí lineární a tedy dopřednou, rychlost v_a je rychlost angulární (úhlová). Lineární rychlost je omezena na hodnotě 0.3 m/s, jelikož při této rychlosti je možné pořád s robotem kvalitně manipulovat v případě výskytu překážky, nebo prudkých změn pohybu leadera.

2.3 Návrh experimentálního prostředí

Pro testování robotického swarmu je důležité především zjistit, jaká konkrétní vlastnost by měla být testována. Práce se zabývá testováním robotického swarmu z hlediska schopnosti udržet předem danou formaci.

Z reálného hlediska má přístup udržování formace velký význam. Je-li potřeba posekat trávu na velké ploše, kterou může být například golfové hřiště, fotbalový stadion, rozlehlá louka atd., může být výhodně využit robotický swarm. Aplikuje-li se na sekací stroj (sekačka, traktor) veškerá technika potřebná k vytvoření hejna těchto strojů, tzn. senzory, kamera, značky pro kameru, je možné tyto stroje nechat pracovat samostatně a řídit pouze vedoucí stroj. Tímto přístupem se minimalizují náklady na lidskou obsluhu, navíc je omezen vliv lidské chyby na výsledek práce.

Obrázek 2.9: Možné využití robotického swarmu - sklízení pole. Převzato z: [22]

Dalším využitím tohoto typu robotiky by mohlo být sklízení pole. Při sklí-

zení pole je nutné udržovat kombajn a traktor s přívěsem od sebe v konstantní vzdálenosti, aby sklizené plodiny dopadaly na přívěs. Bylo by tedy výhodné, pokud by byl řízen pouze kombajn jakožto vedoucí stroj a traktor by jej pouze následoval. V dnešní době již existují autonomní kombajny a traktory, které jezdí pomocí systému GPS. Na poli není běžný výskyt překážek, které by tento signál rušily, proto je použití GPS v tomto případě možné.

Formace robotů by mohla být také využita při použití sněžných pluhů. Při náhlé změně počasí, kdy za krátký čas napadne velké množství sněhu, je nutné tento sníh rychle uklidit z veškerých komunikací a také z letištních ploch. Právě v případě runwaye je potřebné sníh odklízet rychle a efektivně z velké plochy. S výhodou by zde mohlo být využito formace sněhových pluhů, kde by byl potřebný pouze jeden řidič a zbytek pluhů by ho pouze následoval. Při vhodně zvoleném počtu pluhů by tato formace dokázala odklidit celou plochu pouze jedním průjezdem, čímž by se značně snížil potřebný čas pro vyčištění plochy a zvýšil se čas, kdy runway může být efektivně použita pro přistávání letadel.

Obecně lze problém udržení formace shrnout do několika pravidel, která je nutné pro správné fungování formace dodržovat:

- **Udržovat konstantní pozici ve formaci** - Při sledování vedoucího robotu je nutné udržovat konstantní vzdálenost, aby tvar formace, a tudíž její funkce, nebyl narušen. Zde lze uvažovat o možných tolerovaných odchylkách. Je-li systém založen na principu aktivní i pasivní interakce, je možné tyto odchylky minimalizovat na desítky centimetrů. Pokud je formace založena na principu pouze pasivní interakce, odchylky jsou větší.

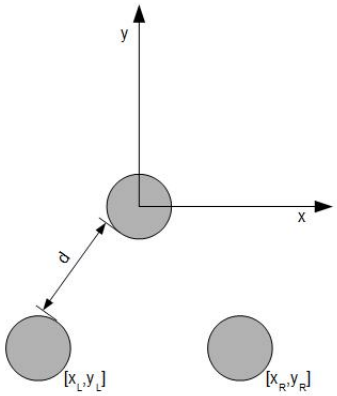
- **Reagovat na okolí** - Dále je nutné zajistit robustnost této formace a odolnost vůči vnějším vlivům. Formace by neměla být narušena kolizí jednoho nebo více jejích členů s okolím nebo mezi sebou navzájem. V algoritmu je tedy nutné implementovat systém collision avoidance, díky kterému jsou členy formace schopni reagovat na okolní objekty a zabránit kolizím. Je také potřeba nejen zabránit kolizi, ale vrátit se zpět na své místo v určené formaci.

2.3.1 Návrh vyhodnocování experimentů

Po analýze problému udržování formace bylo vybráno měření a testování následujících kritérií.

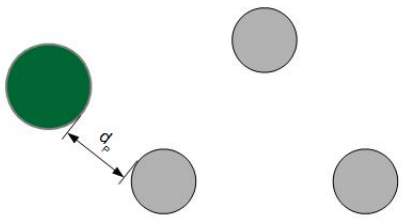
- **Relativní vzdálenost vůči hlavnímu robotu** - Při využití udržování formace v praxi se počítá s kvalitním sledováním cíle a udržením se ve správné relativní poloze. Hodnota relativní vzdálenosti vůči hlavnímu robotu je proto nejdůležitějším kritériem, kterým lze algoritmus popisovat a vyhodnocovat.

- **Relativní poloha vůči hlavnímu robotu** - Poloha vůči hlavnímu robotu je velmi podobnou informací jako relativní vzdálenost. Obě kritéria se velmi dobře doplňují a lze z nich názorně určit chování celé formace.



Obrázek 2.10: Relativní vzdálenost od vedoucího robotu (d) a relativní vektor polohy $[x, y]$

- **Minimální vzdálenost od překážky** - Pokud není prostředí, ve kterém se formace pohybuje, bezporuchové, je nutné nahlížet také na interakci robotů s okolím. Pokud se v trajektorii některého robotu vyskytuje překážka, robot musí svou trajektorii přizpůsobit. Proto je nutné měřit hodnotu minimální vzdálenosti robotu od překážky, která popisuje kvalitu algoritmu collision avoidance.



Obrázek 2.11: Minimální vzdálenost od překážky d_P

- **Mezní body udržení formace** - Tímto kritériem je myšlena velikost překážky, která již nevratně roztrhne formaci robotů. Pokud formace narazí na velkou překážku a její člen se bude překážce vyhýbat směrem od zbytku formace, pozorujeme, jestli se je robot schopen dostat zpět na své místo ve formaci.

Při jednotlivých testech se bude nahlížet na tato tři kritéria, kterými pak bude konkrétní algoritmus vyhodnocen. Pro měření vzdálenosti od hlavního robotu bude využito hodnot naměřených pomocí detektoru značek AprilTags. Pro vzdálenost od překážky se nejvíce hodí data z laserového senzoru, nejjednodušší je sledovat nejmenší hodnotu z celého změřeného vektoru vzdáleností. Pro mezní bod musí být vytvořena překážka s možností měnit svou velikost, která bude zvětšována, dokud se formace swarmu nevratně nepoškodí.

Všechna tato testování by měla kvůli své relevanci proběhnout v ideálních podmínkách, tedy :

- **Kvalitní povrch** - Povrch, po kterém se roboty pohybují, by měl být z takového materiálu, aby bylo zamezeno prokluzu kol robotu. Pokud je podlaha kluzká (rozlitá voda, naleštěna voskem atd.), je testování nepřesné, protože není zajištěno přesné řízení robotů.

- **Detekovatelné překážky** - V případě, že je k testování potřeba využití překážek, je nutné zařídit detekovatelnost těchto překážek. Pokud je použitý senzor pro detekci překážek laserový, musí být použity překážky z materiálu, které laserové paprsky bezchybně odrazí. Tímto postupem se minimalizuje chyba měření pomocí senzoru. Zároveň je potřeba vybrat překážky z materiálu, který robot nepoškodí při kolizi, pokud tato situace nastane.

2.3.2 Matematické aparáty pro vyhodnocování algoritmů

V této práci je nutné vyhodnotit kvalitu testovaných algoritmů. V předchozí kapitole bylo navrženo několik kritérií, která musí být nějak zpracována. Pro toto vyhodnocování jsou použity 2 statistické aparáty.

Charakteristika polohy

Jako první statistický aparát byla vybrána pětihodnotová charakteristika polohy. Tato charakteristika pracuje s pěti matematickými pojmy - minimum, maximum, aritmetický průměr, modus a medián.

- Minimum je minimální hodnota ze souboru hodnot.
- Maximum je maximální hodnota ze souboru hodnot.
- Aritmetický průměr (Mean average) je průměrnou hodnotou ze souboru. Vzorec pro výpočet této hodnoty je

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, \tag{2.1}$$

  kde $\overline{x}$ je aritmetickým průměrem všech hodnot x_i, n je počet hodnot x_i.
- Modus (Mode average) je hodnota s největším počtem výskytů v souboru hodnot. Pro tuto hodnotu platí vztah

$$P|X = \widehat{x}| \geq P|X = x_i|, \tag{2.2}$$

  kde $P|X = \widehat{x}|$ je pravděpodobnost znaku $\widehat{x}$ a $P|X = x_i|$ je pravděpodobnost znaku x_i.
- Medián (Median average) je hodnota, která dělí vzestupně seřazený soubor hodnot na dvě stejné poloviny. Platí tedy, že nejméně 50% hodnot ze souboru je menších nebo rovných a nejméně 50% hodnot je větších nebo rovno mediánu.

Střední kvadratická odchylka

Druhým statistickým aparátem je střední kvadratická odchylka, neboli rozptyl. Pro tuto odchylku platí vztah

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}[x_i - E(x)]^2, \tag{2.3}$$

kde σ^2 je hledaný rozptyl, n je počet hodnot x_i a $E(x)$ je střední hodnotou souboru hodnot.

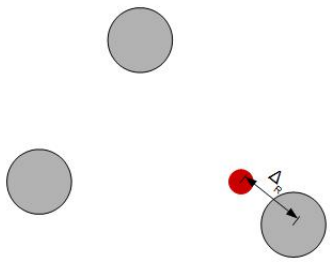
Užití aparátů pro vyhodnocení kvality algoritmů

Zmíněné statistické aparáty jsou využity pro vyhodnocení některých kritérií z podkapitoly 2.3.1.

- Pro vyhodnocování relativní vzdálenosti od vedoucího robotu je využita charakteristika polohy. Tato charakteristika je v tomto případě nejideálnějším hodnocením, protože je z ní ihned vidět nejmenší a největší vzdálenost od vedoucího robotu. Dále všechny tři průměry vyjadřují průměrnou vzdálenost, ve které se robot pohybuje od leadera. Samozřejmě by pro vyhodnocení relativní vzdálenosti bylo možné využít i střední kvadratické odchylky, tato odchylka má však mnohem větší význam u druhého kritéria.
- Druhé kritérium, relativní pozice vůči leaderovi, je vyhodnoceno pomocí přizpůsobené střední kvadratické odchylky. Využití charakteristiky polohy zde nemá moc velký význam, protože by musely být obě souřadnice x a y vyhodnocovány samostatně. Pokud si upravíme vzorec 2.3 do tvaru

$$\sigma_j^2 = \frac{1}{n}\sum_{i=1}^{n}\Delta_i^2, \tag{2.4}$$

kde σ_j^2 je střední kvadratická odchylka j-tého robotu od ideální pozice, n je počet jednotlivých hodnot polohy a Δ_i je vzdálenost robotu od ideální pozice.



Obrázek 2.12: Odchylka robotu Δ_R od ideální pozice vůči leaderovi

Pro celkovou hodnotu kvality udržení formace pak využijeme středních kvadratických odchylek všech členů formace a dle vzorce 2.5 vytvoříme celkovou střední kvadratickou odchylku formace.

$$\sigma_S^2 = \sqrt{\sum_{j=1}^{m}(\sigma_j^2)^2} \tag{2.5}$$

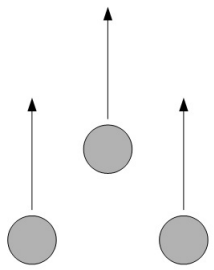
V rovnici 2.5 je σ_S^2 střední kvadratickou odchylkou formace a m je počet robotů (bez vedoucího robotu) ve formaci.

- Třetím kritériem je minimální vzdálenost od překážky. Pro toto kritérium není použito žádné statistické zpracování, jelikož nám stačí najít minimální hodnotu, kterou srovnáme s požadovaným minimem.

2.3.3 Testovací případy 1 - Leader pohybující se po trajektorii přímky

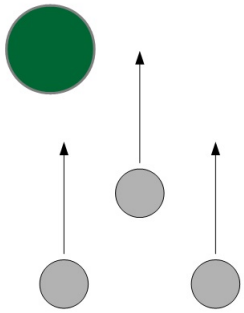
První testovací případy měří kvalitu sledování leadera, který se pohybuje vpřed.

Prvním případem je testování formace v prázdné oblasti, kde na sledující roboty nebudou působit žádné vnější poruchy. Při tomto testu je zaručeno nejkvalitnější měření relativní polohy vůči hlavnímu robotu.



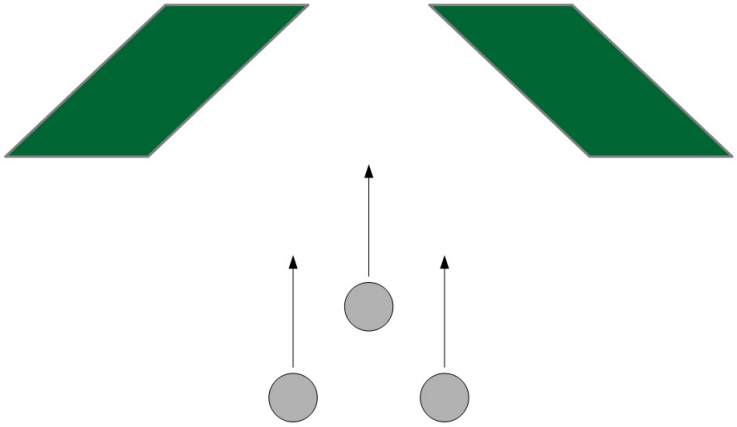
Obrázek 2.13: Testovací případy 1 - Bez překážek

Druhým případem je přidání překážky do místa pohybu robotu, která může reprezentovat například sloup či kmen stromu. Robot ovlivněný překážkou by se jí měl vyhnout a rychle se vrátit zpět do své správné pozice ve formaci. V této situaci je počítáno i s třetím kritériem, kterým je vzdálenost od překážky. Robot by měl dodržet tuto minimální hodnotu, ale zároveň kvůli velikosti překážky by neměl dlouho setrvávat mimo svou ideální vzdálenost od vedoucího robotu.



Obrázek 2.14: Testovací případy 1 - Sloupová překážka

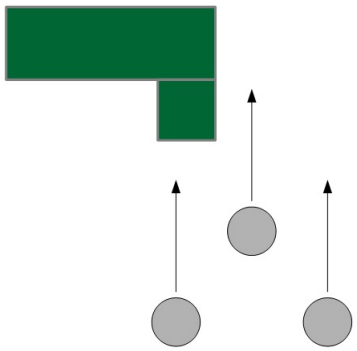
Třetím případem je vytvoření úžiny, kterou leader projede a zbytek formace se za něj musí zařadit. Úžina musí být sestrojena stylem, aby bylo možné robotem projet tak, že bude dodržena minimální vzdálenost od překážky. Pokud toto kritérium není dodrženo, porovnávání hodnoty vzdálenosti od překážky s určenou minimální vzdáleností není relevantní.



Obrázek 2.15: Testovací případy 1 - Úžina

Posledním případem je vytvoření překážky ve tvaru písmena L, která bude realizovat velkou překážku, například zeď. Překážka je v tomto případě umís-

těna jako na obrázku 2.13. Zde se testuje kvalita udržování formace, tedy její mezní body. Pokud robot bude objíždět překážku směrem vně od formace, bude měřeno, jak velkou překážku je schopen objet, a znovu najít zbytek formace do které se bude schopen zařadit zpět na správné místo. Bude-li robot objíždět překážku směrem dovnitř do formace, vybírá si horší cestu, ale zároveň cestu správnou - v tomto případě by nemělo nastat roztržení formace a tudíž je algoritmus navrhnut ideálně, aby nedocházelo k odtržení robotů.

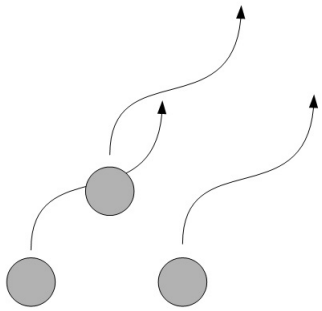


Obrázek 2.16: Testovací případy 1 - Překážka tvaru L

2.3.4 Testovací případy 2 - Leader pohybující se po trajektorii křivky

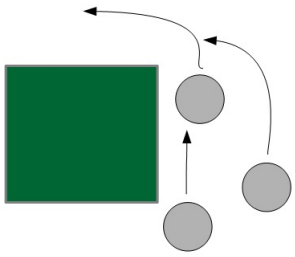
Druhá sada testovacích případů se zabývá sledováním leadera, který ale může zatáčet a tudíž je hůře detekovatelný pro sledující roboty.

Prvním případem je pouze prázdná oblast, ve které se hlavní robot pohybuje po trajektorii tvaru nějaké křivky. V tomto případě se změří, jak dobře je sledující robot schopen zareagovat na změnu směru hlavního robotu.



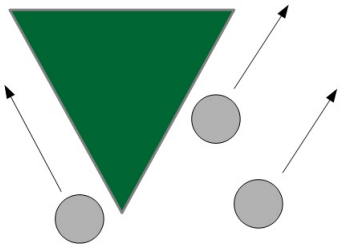
Obrázek 2.17: Testovací případy 2 - Bez překážek

Druhý případ přidá do testovací oblasti větší překážku, která reprezentuje například roh budovy. Leader se pohybuje podél této překážky, na jejím rohu zatočí směrem za překážku. V tomto případě je testována schopnost sledujícího robotu reagovat na ztrátu hlavního robotu ze zorného pole kamery a schopnost ho opět najít a zařadit se do formace.



Obrázek 2.18: Testovací případy 2 - Rohová překážka

Třetí případ je vytvoření překážky ve tvaru písmena V. Formace směřuje pohyb směrem na špičku této překážky se snahou oddělit hlavní robot a sledující robot na rozdílné strany této překážky. Pokud se formaci podaří rozdělit, testuje se, jestli je sledující robot schopen opět se navázat na hlavní robot a pokračovat společně s formací. Pokud k oddělení nedojde, algoritmus je zvolen správně, aby zabraňoval roztržení formace.



Obrázek 2.19: Testovací případy 2 - Překážka ve tvaru klínu

V prvních dvou případech má význam měřit relativní vzdálenost od hlavního robotu, v druhém a třetím případě pak vzdálenost od překážky. Poslední případ je navrhnut pro otestování schopnosti držet formaci při delším oddělení.

Kapitola 3

Reálný experiment

Pro reálný experiment byl vybrán pružinový algoritmus z práce [24]. Reálný experiment byl proveden se dvěma sledujícími roboty a vedoucím robotem ve formě AprilTags prstence taženým člověkem, jelikož nebylo k dispozici více robotů. Testování je tedy zatíženo chybou způsobenou nedokonalostí pohybu vedoucího robotu - rychlost nebyla konstantní a je možné, že byla překročena maximální rychlost robotu (0.3 m/s). Experiment byl navíc negativně ovlivněn nedostatkem kamer, každý robot byl vybaven pouze jednou kamerou sledující vedoucí robot. Algoritmus tedy není implementován přesně dle práce, jelikož ten požaduje vytvoření pružin i mezi sousedními roboty. Dále bylo nutné kameru natočit směrem do formace, protože při kameře směřující rovně byl prstenec v ideální poloze robotu na kraji obrazu a rychle se značka ze záběru ztrácela. Natočení kamery ovšem snižuje přesnost jízdy robotu a proto bylo nutné kompromisně zvolit její natočení o 10°.

3.1 Parametry provedeného měření

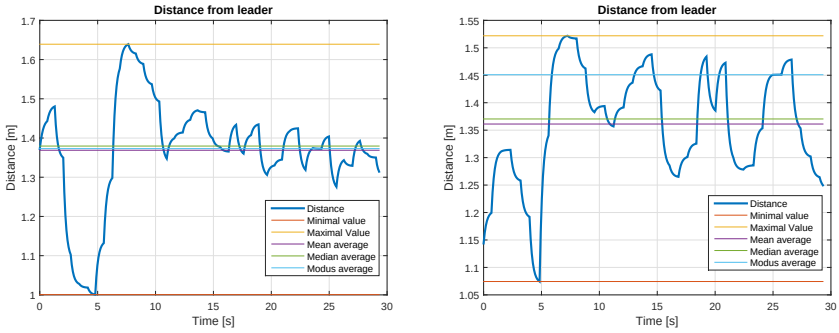
Experiment byl proveden za osvětlení 650 lux-690 lux. Kamera byla kalibrována na vzdálenost 1 m. Klidová délka pružin je 1 m. Rozlišení kamery bylo nastaveno na 1280x960. Minimální vzdálenost od překážky byla stanovena na 0.2 m. Formace byla zvolena ve tvaru šipky či letky.

Ideální pozice obou robotů byly nastaveny v souřadné soustavě z obrázku

(2.10) na $x = -1\,\mathrm{m}$ a $y = \pm 0.5\,\mathrm{m}$. Testovaná formace využívá pouze pasivní interakce robotů, maximální možná odchylka vzdálenosti byla stanovena na 50 %, jelikož větší odchylka od ideální pozice by narušovala správné fungování formace. Prvním experimentem bylo zjištěno, že průměrná vzdálenost robotu od vedoucího robotu je 1.36 m. Vzdálenosti při jízdě by se tedy měly vyskytovat v intervalu 0.68 m - 2.04 m.

3.2 Jízda robotů po trajektorii přímky

První pokus byl proveden při rovnoměrné jízdě vpřed bez překážek. Z této jízdy je následně zjištěna ideální vzdálenost robotu od vedoucího robotu, se kterou se vyhodnocují další měření. Tato hodnota byla zvolena jako průměrná vzdálenost při jízdě vpřed.



(a) : Vývoj vzdálenosti levého robotu od robotu vedoucího

(b) : Vývoj vzdálenosti pravého robotu od robotu vedoucího

Obrázek 3.1: Grafy vývoje vzdáleností jednotlivých robotů při jízdě vpřed

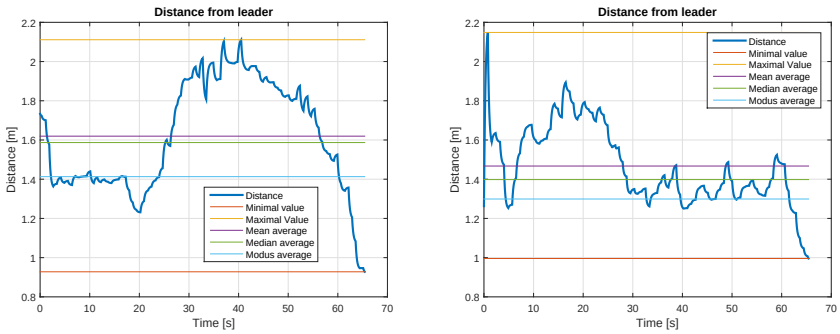
V obrázku 3.1. je vidět, že oba roboty se chovají při jízdě rovně velmi podobně. Průměrná vzdálenost robotu od vedoucího robotu je 1.36 m. Z grafů je také vidět, že rozpoznávání obrazu kamerou běží na nízké frekvenci, což vysvětluje "zuby" v průběhu vzdáleností.

Střední kvadratická odchylka formace má pro tento test hodnotu $\sigma_S^2 = 0.1301$. Jak lze vidět, formace při jízdě po rovině má velmi malou odchylku a je tedy vysoce kompaktní.

Obrázek 3.2: Foto z pokusu při jízdě vpřed

3.3 Jízda robotů po trajektorii křivky

Dalším pokusem byl pohyb vedoucího robotu po trajektorii ve tvaru křivky. Trajektorie byla zatočena nejprve doleva ve směru jízdy, posléze pak přesměrována doprava. Při zatáčce doleva tedy musí pravý robot vykonat trajektorii delší, než vedoucí robot, a naopak.



(a) : Vývoj vzdálenosti levého robotu od robotu vedoucího

(b) : Vývoj vzdálenosti pravého robotu od robotu vedoucího

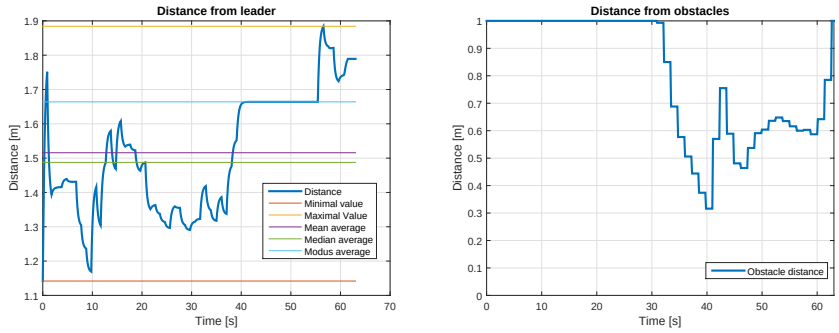
Obrázek 3.3: Grafy vývoje vzdáleností jednotlivých robotů při jízdě se zatáčkami

V grafech z obrázku 3.3. je vidět vývoj jednotlivých vzdáleností, které odpovídají očekávaným předpokladům. Kolem 30 s je vidět změna trajektorie z levotočivé na pravotočivou zatáčku. Oba roboty při zatáčení se pohybují ve vzdálenosti kolem 1.9 m, algoritmus tedy splňuje kritérium 50 % odchylky od ideální vzdálenosti.

Střední kvadratická odchylka vychází při tomto testu $\sigma_S^2 = 0.4222$. V porovnání s odchylkou v testu 3.2 lze pozorovat, že roboty při trajektorii křivky se neudržují v tak blízkém okolí své ideální pozice. Přesto formace nebyla přerušena a udržovala stále svůj tvar.

3.4 Jízda robotů kolem rohové překážky

V tomto testu byla vyzkoušena funkce algoritmu collision avoidance a také funkčnost kalmanova filtru. Podle výše popsaného pokusu (obrázek 3.13.) bylo připraveno testovací prostředí. Pokus byl proveden tak, že vedoucí robot zatáčel kolem sloupu po směru hodinových ručiček. Robot na pravé straně se musel překážce vyhnout, robot na straně levé projížděl obyčejnou zatáčku doprava.

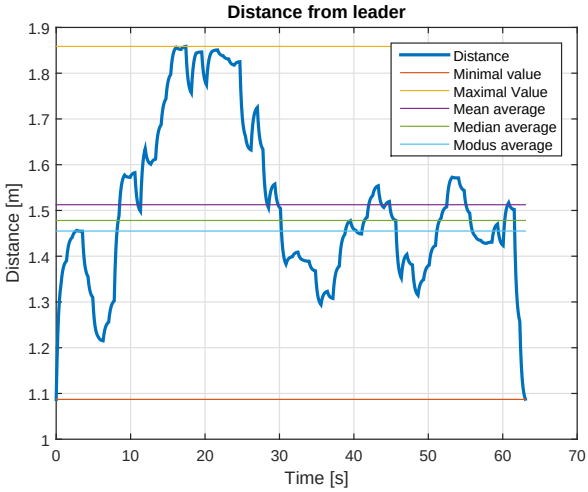


(a) : Vývoj vzdálenosti pravého robotu od robotu vedoucího

(b) : Vývoj vzdálenosti pravého robotu od překážky

Obrázek 3.4: Grafy vývoje vzdálenosti pravého robotu od vedoucího robotu a od překážky při jízdě kolem rohové překážky

Z grafu na obrázku 3.4.(b) je patrné, že robot v čase 32 s začal zaznamenávat překážku, které se následně musel vyhnout. Z grafu je také vidět, že vzdálenost robotu od překážky neklesla pod 0.3 m, bylo tedy splněno kritérium pro minimální vzdálenost. Z grafu na obrázku 3.4.(a) je vidět v čase 40 s nástup funkce kalmanova filtru, jelikož byl ztracen vedoucí robot ze záběru kamery. Robot poté jel 16 s pouze s hodnotami kalmanova filtru, než znovu objevil v záběru kamery vedoucí robot. V grafu je funkce kalmanova filtru



Obrázek 3.5: Vývoj vzdálenosti levého robotu od vedoucího robotu

značena konstantní vzdáleností robotů, vypočtené průměrné hodnoty jsou tímto vlivem zkresleny, přesto by průměrná vzdálenost robotu nepřekročila hodnotu 1.9 m.

Z grafu na obrázku 3.5. je vidět, že levý robot v čase 10 s - 30 s pouze zatáčel doprava bez vlivu jakékoliv překážky, výsledek je tedy srovnatelný s průběhem hodnot z obrázku 3.3.(a) v časech 25 s - 60 s.

Obrázek 3.6: Foto z pokusu při jízdě kolem rohové překážky (nosný sloup budovy)

Při jízdě kolem rohu byla hodnota střední kvadratické odchylky $\sigma_S^2 = 0.5644$. Tuto odchylku způsobuje pravý robot, který se musel vyhýbat překážce. Při

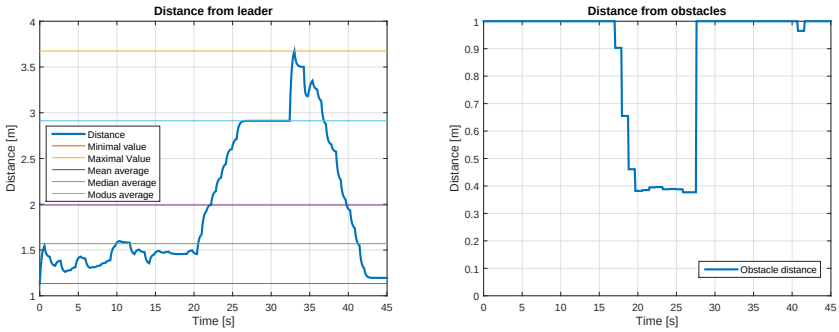
srovnání s testem 3.3 však tato hodnota není o tolik vyšší a formace nebyla překážkou velmi ovlivněna.

3.5 Jízda robotů kolem sloupu

Toto měření bylo sestaveno podle návrhu na obrázku 3.9. Při prvním testování byl nalezen problém v algoritmu, kdy, v případě menší překážky vyskytující se přímo před robotem, se robot zastaví, a do formace se dokáže zařadit až po delší době. Proto bylo toto měření provedeno dvakrát, kdy podruhé byla trajektorie robotu přizpůsobena tak, aby překážku měl více po straně. Druhé měření testovalo, jestli nastane stejný problém jako v měření prvním.

3.5.1 1. testování

Při prvním pokusu překážka ovlivňovala pouze levý robot, pravý robot měl trajektorii stejnou jako při testu z podkapitoly 3.2.



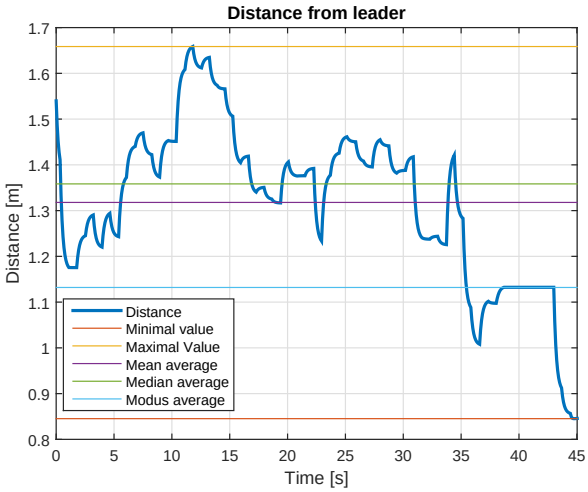
(a) : Vývoj vzdálenosti levého robotu od robotu vedoucího

(b) : Vývoj vzdálenosti levého robotu od překážky

Obrázek 3.7: Grafy vývoje vzdálenosti levého robotu od vedoucího robotu a od překážky při jízdě kolem sloupu

Z grafu (3.7.(b)) je vidět, že robot začal registrovat překážku kolem 17 s. Z grafu (3.7(a)) lze najít hodnoty vzdálenosti od vedoucího robotu až 3.6 m,

což nesplňuje výše uvedený interval (podkapitola 3.1) povolené vzdálenosti od vedoucího robotu. V čase 26 s - 32 s byl vedoucí robot opět ztracen ze záběru kamery, robot tedy využíval hodnoty z kalmanova filtru. Vzdálenost od překážky byla dodržena, v tomto experimentu se robot nepřiblížil více než na 0.38 m.

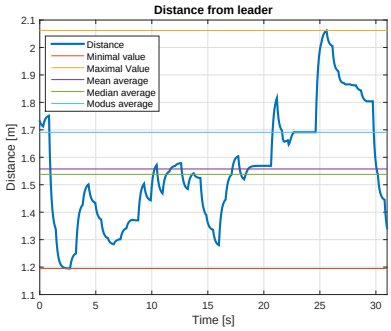


Obrázek 3.8: Vývoj vzdálenosti pravého robotu od vedoucího robotu

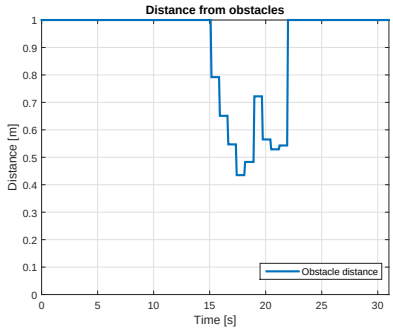
Druhý robot se pohyboval standardně, jak je vidět z grafu 3.8. Celkově lze pokus prohlásit za neúspěšný, protože je velmi pravděpodobné, že by se levý robot ztratil, kdyby vedoucí robot ve vzdálenosti 3.7 m od překážky nezastavil.

3.5.2 2. testování

Při druhém pokusu byla překážka posunuta více do strany. Při pohledu na graf 3.9. je zřejmé, že robot se překážce vyhnul lépe než v prvním případě. Bohužel maximální hodnota vzdálenosti od vedoucího robotu činí 2.06 m, která spadá již mimo interval určený v podkapitole 3.1. Jelikož je hodnota maxima jen o 0.02 m vyšší a doba trvání je jen zlomek sekundy, je možné tuto hodnotu zanedbat a výsledek považovat za dostačující. Minimální vzdálenost se pohybuje kolem hodnoty 0.4 m, stejně jako v prvním případě.

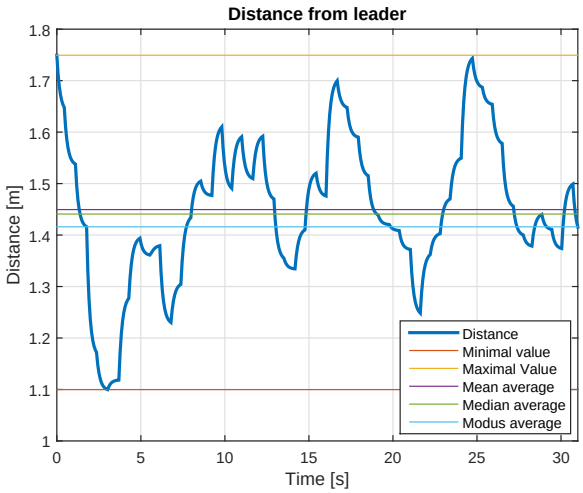


(a) : Vývoj vzdálenosti levého robotu od robotu vedoucího



(b) : Vývoj vzdálenosti levého robotu od překážky

Obrázek 3.9: Grafy vývoje vzdálenosti levého robotu od vedoucího robotu a od překážky při jízdě kolem sloupu



Obrázek 3.10: Vývoj vzdálenosti pravého robotu od vedoucího robotu

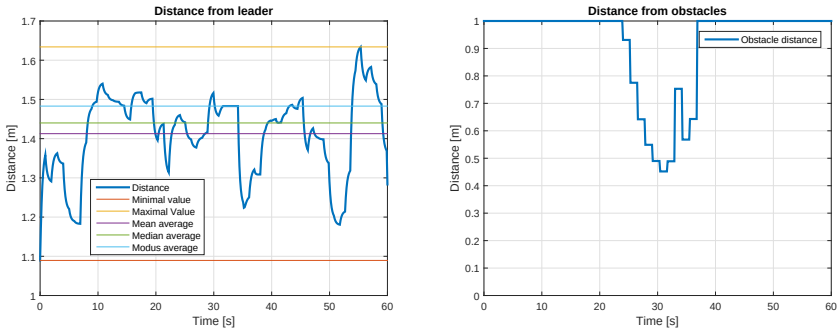
V grafu 3.10. lze pozorovat, že průměrná vzdálenost od vedoucího robotu je skoro o 0.1 m vyšší než v ideálním případě. Tato skutečnost je velmi pravděpodobně zapříčiněna nekonstantním pohybem vedoucího robotu.

Střední kvadratická odchylka formace vyšla $\sigma_S^2 = 1.6123$ pro první pokus, pro druhý $\sigma_S^2 = 0.4013$. Lze pozorovat, že při prvním pokusu byla odchylka 4x větší než v pokusu druhém. Zároveň ale druhý pokus vyhnutí se sloupu vychází lépe než test v podkapitole 3.3. Celkově lze výsledek testu vyhnutí se sloupu považovat za nedostatečný. Minimální vzdálenost od překážky byla dodržena, maximální vzdálenost od vedoucího robotu však nikoliv a střední kvadratická odchylka byla velmi vysoká.

Obrázek 3.11: Foto z druhého měření objíždění sloupu

3.6 Jízda skrz zúžení

Následujícím testem (obrázek 3.11.) je testována především minimální vzdálenost od překážky a schopnost robotů zařadit se zpět do formace po vychýlení. Proto zde nebude přikládána velká váha maximální hodnotě vzdálenosti od vedoucího robotu, jelikož ji lze považovat za irelevantní, pokud roboty problém úspěšně vyřeší a formace se neroztrhne.

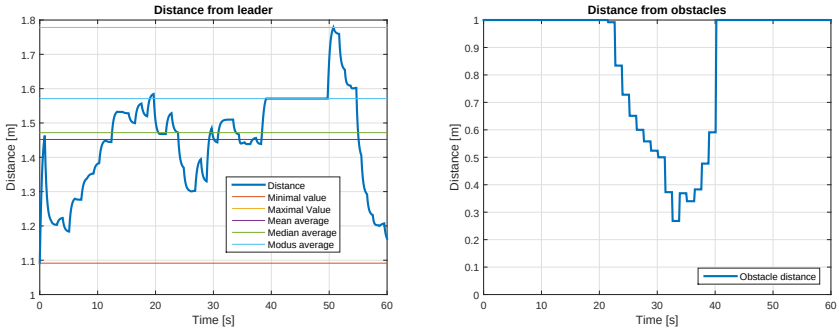


(a) : Vývoj vzdálenosti levého robotu od robotu vedoucího

(b) : Vývoj vzdálenosti levého robotu od překážky

Obrázek 3.12: Grafy vývoje vzdálenosti levého robotu od vedoucího robotu a od překážky při jízdě zúženým místem

Z grafu 3.12. lze vyčíst, že na pozici levého robotu tato překážka měla minimální vliv. Průměrná hodnota vzdálenosti od vedoucího robotu je jen o 0.06 m vyšší a minimální vzdálenost od překážky neklesla pod hodnotu 0.4 m.



(a) : Vývoj vzdálenosti levého robotu od robotu vedoucího

(b) : Vývoj vzdálenosti levého robotu od překážky

Obrázek 3.13: Grafy vývoje vzdálenosti pravého robotu od vedoucího robotu a od překážky při jízdě zúženým místem

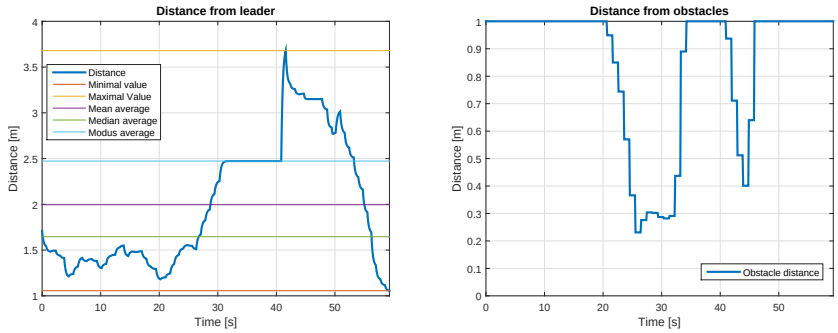
V případě robotu pravého, který se zařadil při průjezdu za robot levý, lze pozorovat vyšší vliv překážky. Od 39 s do 50 s je robot řízen hodnotami pouze z kalmanova filtru, což bylo způsobeno zakrytím výhledu druhým robotem. Celkově ale robot nepřekročil hodnotu vzdálenosti od vedoucího robotu 1.8 m, což je naprosto dostačující. Hlavním měřítkem je však vzdálenost od překážky, která dosáhla minima v hodnotě 0.26 m.

Obrázek 3.14: Foto z pokusu projetí zúženého místa

Průjezd úžinou vychází z hlediska kompaktnosti formace nejlépe ze všech testovacích případů (bez jízdy po rovině). Střední kvadratická odchylka pro tento pokus je $\sigma_S^2 = 0.3178$. Celkový test projetí zúženým místem dopadl úspěšně, všechny požadavky byly splněny a kompaktnost formace vyšla velmi dobře.

3.7 Jízda s překážkou ve tvaru klínu

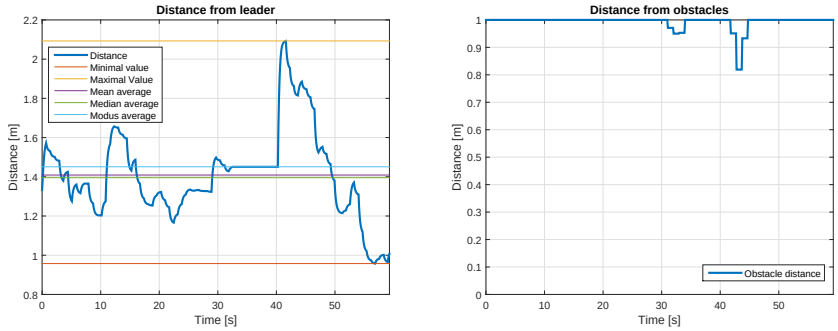
Podle obrázku 3.15 byla postavena překážka ve tvaru klínu. Trajektorie vedoucího robotu byla vedena vlevo od překážky, pravý robot byl tedy nucen se překážce vyhnout, levý následoval pouze trajektorii po křivce.



(a) : Vývoj vzdálenosti pravého robotu od robotu vedoucího

(b) : Vývoj vzdálenosti pravého robotu od překážky

Obrázek 3.15: Grafy vývoje vzdálenosti pravého robotu od vedoucího robotu a od překážky při jízdě na překážku ve tvaru klínu



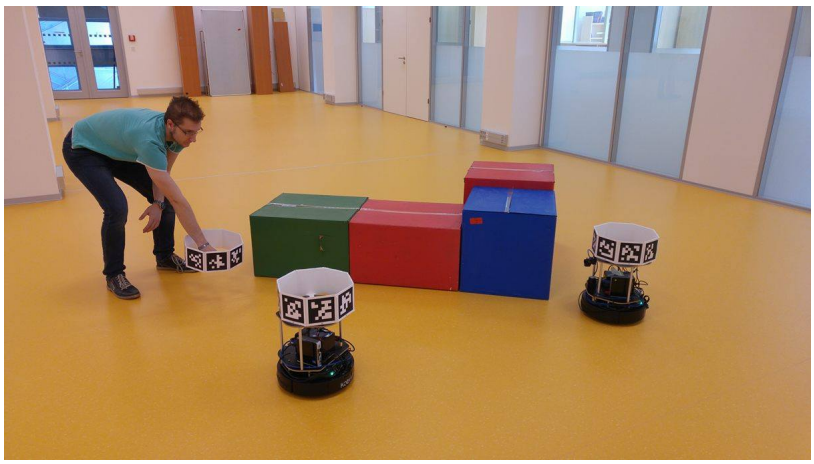
(a) : Vývoj vzdálenosti levého robotu od robotu vedoucího

(b) : Vývoj vzdálenosti levého robotu od překážky

Obrázek 3.16: Grafy vývoje vzdálenosti levého robotu od vedoucího robotu a od překážky při jízdě na překážku ve tvaru klínu

Ve 20 s pravý robot detekoval překážku (graf 3.15.) a začal na ni reagovat. Ve 30 s ztratil vedoucí robot ze záběru, ale s hodnotami z kalmanova filtru pokračoval asi 10 s než vedoucí robot opět objevil. Robot si vybral vyhnutí se překážce směrem dovnitř formace (obrázek 3.17.) a proto nedošlo k roztržení formace. Minimální vzdálenost od překážky také nebyla překročena.

Levý robot jel standardní jízdu, v čase 32 s byl ovlivněn překážkou (graf 3.16.) kvůli které ztratil vedoucí robot ze záběru kamery a byl nucen jet dle kalmanova filtru. Proto jeho maximální hodnota vzdálenosti od vedoucího robotu je 2.09 m po nalezení robotu zpět v obraze.

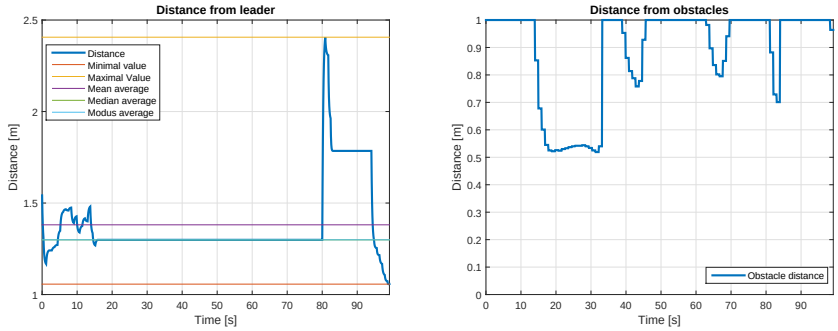


Obrázek 3.17: Foto z pokusu projetí kolem překážky ve tvaru klínu

Střední kvadratická odchylka pro toto testování vychází $\sigma_S^2 = 1.6605$, což je nejvyšší hodnota ze všech testování. Pravý robot, který byl překážkou ovlivněn měl odchylku $\sigma_R^2 = 1.6506$, protože setrvával dlouho vzdálený od své ideální pozice. Formace byla při tomto testu velmi poškozena, nedošlo však k jejímu roztržení. Proto celkový test je vyhodnocen jako úspěšný.

3.8 Jízda s překážkou ve tvaru L

Závěrečný pokus byl proveden podle návrhu (obrázek 3.12.). Tento tvar překážky je nejhorším tvarem, jaký může v reálné situaci nastat. Na překážku byla formace navedena tak, že pravý robot směřoval na ni, levý jel pouze rovně za vedoucím robotem bez kontaktu s překážkou.

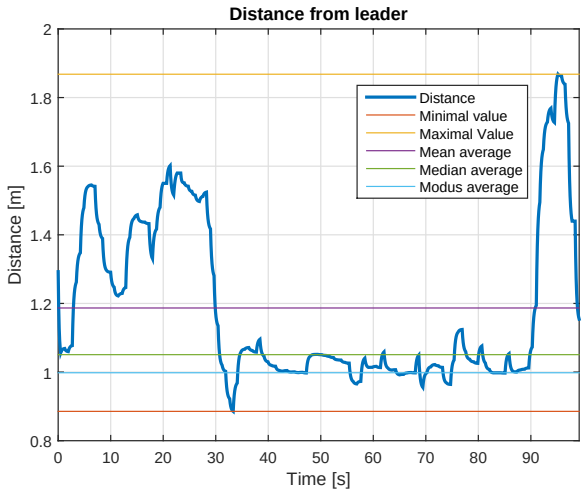


(a) : Vývoj vzdálenosti pravého robotu od robotu vedoucího

(b) : Vývoj vzdálenosti pravého robotu od překážky

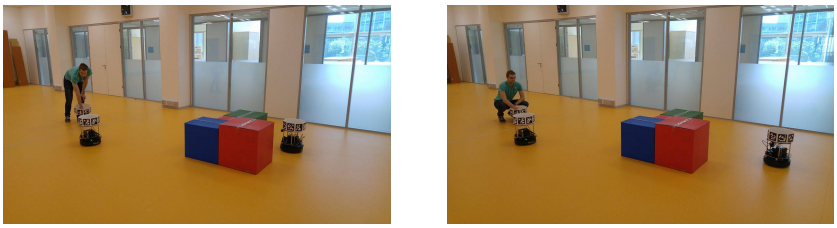
Obrázek 3.18: Grafy vývoje vzdálenosti pravého robotu od vedoucího robotu a od překážky při jízdě na překážku ve tvaru klínu

Jak lze vidět v grafu 3.18.(a), robot zajel za překážku a ztratil vedoucí robot ze záběru kamery už v 15 s. Robot se za překážkou zastavil, poté začal kroužit doprava a překážku nakonec objel směrem do formace. Vedoucí robot byl zastaven v čase 30 s, takže pravý robot se nakonec byl schopen vrátit zpět do formace.



Obrázek 3.19: Vývoj vzdálenosti levého robotu od vedoucího robotu

Levý robot byl od času 30 s zastaven za vedoucím robotem, kde osciloval kolem své ideální pozice (kvůli nízké frekvenci dat obdržených z kamery a detekci vedoucího robotu algoritmem collision avoidance), v čase 90 s se vedoucí robot rychle rozjel.



(a) : Foto v čase 30 s **(b) :** Foto v čase 50 s

Obrázek 3.20: Foto z pokusu z překážkou ve tvaru L

Počítání střední kvadratické odchylky od ideální pozice v tomto případě není relevantní, protože kvůli konstantním datům z pravého robotu tato odchylka vychází velmi malá, přestože robot setrvával přes minutu daleko mimo svou ideální pozici.

Celkové zhodnocení tohoto testu je nedostatečné. Kdyby vedoucí robot nezastavil na minutu, ztracený pravý robot by se nikdy do formace zpět nezařadil. I v tomto případě bylo zařazení do formace spíše náhodné. Pozitivum z testu je skutečnost, že se robot opět snažil vyhnout překážce směrem do formace.

3.9 Zhodnocení algoritmu

Závěrečné zhodnocení algoritmu se skládá z vyhodnocení všech testů dohromady. Algoritmus navržený v práci [24] obstál v 5 ze 7 navržených scénářů. Formace se při jízdě bez přítomnosti překážek nerozpadla a neměnila výrazně svůj tvar. Toto vyjadřují i čísla v kapitolách 3.2 a 3.3. Dále v žádném z testovaných případů nedošlo ke kolizi s překážkou ani k překročení minimální vzdálenosti 0.2 m. Formace byla roztržena pouze překážkou tvaru L, kde díky specifickému tvaru překážky se vedoucí robot ztratí ze záběru kamery velmi rychle. Ideální také nebylo objíždění malé překážky (například strom, sloup), které bylo velmi pomalé a robot na tuto překážku reagoval přehnaně.

Při vyhodnocení středních kvadratických odchylek bylo dosaženo velmi vysokých čísel pouze u testu se sloupem a testu s překážkou ve tvaru klínu. Obě tyto hodnoty se pohybovaly kolem $\sigma_S^2 = 1.6$, každá z nich však měla při pokusu jinou váhu. Při jízdě kolem sloupu by měla formace zachovat tvar skoro bez narušení a proto by odchylka měla být velmi malá. Naopak u

překážky ve tvaru V je předpokládáno s větším zásahem do formace a tedy i s vyšší odchylkou, proto je tento test vyhodnocen jako úspěšný. Všechny ostatní testy, ve kterých byla střední kvadratická odchylka formace počítána dopadly úspěšně a odchylka byla malá.

Algoritmus je možné tedy považovat za kvalitní, většinu problémů v reálném použití řeší dostačujícím způsobem. Udržování formace je implementováno robustně, je potřeba velmi složitá překážka, aby došlo k poškození formace. Celý algoritmus by navíc fungoval lépe, kdyby byl implementován přesně dle návrhu a bylo použito více kamer (nebyly k dispozici) pro vytvoření více pružinových vazeb.

Závěr

3.10 Dosažené výsledky

Práce zpracovává problematiku testování swarmu mobilních robotů. Byly vytvořeny dva exempláře mobilních robotů, které jsou opatřeny všemi senzory potřebnými pro možnost využití při testech. Celkově je možné s těmito roboty vytvořit malou formaci, pokud je vedoucí robot (konkrétně prstenec AprilTags) tažen člověkem, nebo jiným typem robotu. Řídící smyčka člena formace je psána v jazyce C++. Zdrojový kód je možné upravit podle typu algoritmu, který je potřeba testovat.

První část práce se zabývala problematikou experimentálních prostředí, jejich použitím ve světě a realizací některých experimentů. Bylo zjištěno, že problémem skupinové robotiky a jejím testováním se nezabývá tolik prací, kolik by tato problematika vyžadovala.

Druhá část práce pojednává o navrženém experimentálním prostředí včetně jednotlivých robotů. Roboty byly vybrány podle cenové dostupnosti a schopnosti jednoduché manipulace ve vnitřním prostředí. Dále se v této části zabýváme celkovou problematikou skupinové robotiky a jejím využitím, na což navazuje návrh pro testování klíčových vlastností této problematiky. Celkový navržený systém (robot i scénáře pro testovaní) jsou jednoduše realizovatelné, což je nutností při vytváření experimentálního prostředí, aby mohla být prostředí reprodukována a výsledky testování různých algoritmů porovnávány.

Ve třetí a také závěrečné části byl proveden experiment s výše navrženými roboty. Zvolený algoritmus byl testován ve všech navržených scénářích z druhé části této práce. Některé navržené scénáře se prokázaly jako značně nereálné (například překážka ve tvaru L (obrázek 3.13.)), nicméně je nutné algoritmy testovat i v krajních situacích. Vybraný pružinový algoritmus úspěšně splnil 5 modelových situací ze 7 testovaných, lze ho tedy považovat za funkční a mohl by být použit v praxi.

Díky této práci tedy bylo vytvořeno experimentální prostředí pro skupinovou robotiku zaměřené na testování schopnosti udržet danou formaci.

3.11 Další kroky

V rámci této práce byly vyvinuty dva mobilní pozemní roboty, které jsou ovládány vlastní počítačovou jednotkou. Pro kvalitnější testování robotického swarmu by bylo potřeba realizovat minimálně další dva roboty, aby mohla být formace větší a testování více podobné reálným aplikacím. V budoucnu budou roboty osazeny dalšími kamerami, jelikož v době dokončení bakalářské práce ještě kamery nebyly k dispozici. Dále je možné zapracovat na síťové komunikaci mezi jednotlivými roboty, počítače jsou na tuto funkci připraveny díky integrované síťové kartě. Pro zjednodušení interakce uživatele a robotu by mohlo být vyvinuto vývojové prostředí, ve kterém by se měnil pouze samotný testovaný algoritmus. Ve stávající podobě je totiž nutné měnit algoritmus v souboru kompletního kódu robotu, je tedy možné upravovat i kódy týkající se zpracování dat ze senzorů, kamery a robotu.

Příloha A

Literatura

[1] AMIGONI, Francesco, Simone GASPARINI a Maria GINI. Good Experimental Methodologies for Robotic Mapping: A Proposal. In: Proceedings 2007 IEEE International Conference on Robotics and Automation [online]. IEEE, 2007, s. 4176-4181 [cit. 2017-05-07]. DOI: 10.1109/ROBOT.2007.364121. ISBN 1-4244-0602-1. ISSN 1050-4729. Dostupné z: http://ieeexplore.ieee.org/document/4209739/

[2] Robotics Test Facility | NIST. National Institute of Standards and Technology | NIST [online]. Dostupné z: https://www.nist.gov/laboratories/tools-instruments/robotics-test-facility

[3] UYAMA, Naohiro, Havard LUND, Koki ASAKIMORI, Yuki IKEDA, Daichi HIRANO, Hiroki NAKANISHI a Kazuya YOSHIDA. Integrated experimental environment for orbital robotic systems, using ground-based and free-floating manipulators. In: 2010 IEEE/SICE International Symposium on System Integration [online]. IEEE, 2010, s. 108-113 [cit. 2017-05-07]. DOI: 10.1109/SII.2010.5708310. ISBN 978-1-4244-9316-6. Dostupné z: http://ieeexplore.ieee.org/document/5708310/

[4] KULICH, Miroslav, Jan CHUDOBA, Karel KOSNAR, Tomáš KRAJNIK, Jan FAIGL a Libor PREUCIL. SyRoTek—Distance Teaching of Mobile Robotics. IEEE Transactions on Education [online]. 2013, 56(1), 18-23 [cit. 2017-05-07]. DOI: 10.1109/TE.2012.2224867. ISSN 0018-9359. Dostupné z: http://ieeexplore.ieee.org/document/6341862/

[5] CHANG, Qingkai, Xiaolong LIU, Wenfu XU, Lei YAN a Bingsong YANG. The design and experiments of a small wheel-legged mobile robot system with two robotic arms. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) [online]. IEEE, 2016, s. 2590-2595 [cit.

2017-05-07]. DOI: 10.1109/IROS.2016.7759403. ISBN 978-1-5090-3762-9. Dostupné z: http://ieeexplore.ieee.org/document/7759403/

[6] DINH, Huan a Tamer INANC. Low cost mobile robotics experiment with camera and sonar sensors. In: 2009 American Control Conference [online]. IEEE, 2009, s. 3793-3798 [cit. 2017-05-07]. DOI: 10.1109/ACC.2009.5160449. ISBN 978-1-4244-4523-3. Dostupné z: http://ieeexplore.ieee.org/document/5160449/

[7] GRO, Roderich, Michael BONANI, Francesco MONDADA a Marco DORIGO. Autonomous Self-Assembly in Swarm-Bots. IEEE Transactions on Robotics [online]. 2006, 22(6), 1115-1130 [cit. 2017-05-07]. DOI: 10.1109/TRO.2006.882919. ISSN 1552-3098. Dostupné z: http://ieeexplore.ieee.org/document/4020359/

[8] Swarm-bots project. Swarm-bots project [online]. Copyright © [cit. 07.05.2017]. Dostupné z: http://www.swarm-bots.org/index.php@main=3&sub=31&conpage=sbot.html

[9] TurtleBot 2 - Open source personal research robot. [online]. Dostupné z: http://www.clearpathrobotics.com/turtlebot-2-open-source-robot

[10] Haifa University robotics & Big data lab. [online]. Dostupné z: https://sites.hevra.haifa.ac.il/rbd/2015/11/23/turtlebot-navigation-and-face-recognition/?lang=en

[11] Intel® NUC Kit NUC5i5RYK. Laptop, Notebook, Desktop, Server and Embedded Processor Technology - Intel [online]. Copyright © Intel Corporation [cit. 07.05.2017]. Dostupné z: http://www.intel.com/content/www/us/en/products/boards-kits/nuc/kits/nuc5i5ryk.html

[12] Sleviště.cz. [online]. Dostupné z: http://www.sleviste.cz/pc-sestavy/intel-nuc-5i5ryk-boxnuc5i5ryk/

[13] Scanning Rangefinder Distance Data Output/URG-04LX-UG01 Product Details | HOKUYO AUTOMATIC CO., LTD.. HOKUYO AUTOMATIC CO., LTD. [online]. Copyright © 2014 HOKUYO AUTOMATIC CO., LTD. All Rights Reserved. [cit. 07.05.2017]. Dostupné z: https://www.hokuyo-aut.jp/search/single.php?serial=166

[14] ACRONAME. [online]. Dostupné z: https://acroname.com/products/HOKUYO-URG-04LX-UG01-LASER?sku=R325-URG-04LX-UG01

[15] Basler dart - Area Scan Cameras. 302 Found [online]. Copyright © 2017 Basler AG [cit. 07.05.2017]. Dostupné z: https://www.baslerweb.com/en/products/cameras/area-scan-cameras/dart/

[16] Soda Vision Eshop. [online]. Dostupné z: http://www.sodavision.com/shop/camera/basler-dart-daa1280-54um-scsb/

[17] OLSON, Edwin. AprilTag: A robust and flexible visual fiducial system. In: 2011 IEEE International Conference on Robotics and Automation [online]. IEEE, 2011, s. 3400-3407 [cit. 2017-04-19]. DOI: 10.1109/ICRA.2011.5979561. ISBN 978-1-61284-386-5. Dostupné z: http://ieeexplore.ieee.org/document/5979561/

[18] April robotics laboratory. [online]. Dostupné z: https://april.eecs.umich.edu/software/apriltag.html

[19] M. Quigley, K. Conley, B.P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng, "ROS: an open-source Robot Operating System,"ICRA Workshop on Open Source Software, 2009. Dostupné z: http://www.willowgarage.com/papers/ros-open-source-robot-operating-system

[20] Swarm robotics - Wikipedia. [online]. Dostupné z: https://en.wikipedia.org/wiki/Swarm_robotics

[21] Iñaki Navarro and Fernando Matía, “An Introduction to Swarm Robotics,” ISRN Robotics, vol. 2013, Article ID 608164, 10 pages, 2013. doi:10.5402/2013/608164. Dostupné z: https://www.hindawi.com/journals/isrn/2013/608164/

[22] Agrojournal. [online]. Dostupné z: https://www.agrojournal.cz/clanky/bez-ridice-farmar-si-postavil-autonomni-traktor-189

[23] ŽÁKOVEC, Martin. Vizuální navigace mobilních robotů ve formaci. Praha, 2017. Bakalářská práce. ČVUT FEL. Vedoucí práce Ing. Libor Přeučil CSc.

[24] NOVOTNÝ, Martin. Robustní postupy řízení formací robotů. Praha, 2017. Bakalářská práce. ČVUT FEL. Vedoucí práce Ing. Libor Přeučil CSc.

České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra kybernetiky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: Matěj B e r á n e k

Studijní program: Kybernetika a robotika (bakalářský)

Obor: Robotika

Název tématu: Experimentální prostředí pro skupinovou robotiku

Pokyny pro vypracování:

1. Seznamte se s problematikou laboratorní experimentace v oblasti malých pozemních mobilních robotů pro vnitřní prostředí.
2. Navrhněte experimentální systém pro ověřování algoritmů řízení mobilních robotů ve formacích. Do návrhu, spolu s mechanickým řešením nástavby robotu, zahrňte vybavení palubními senzory, řídícím počítačem s vhodným OS unixového typu a systémem napájení kompatibilním s užitým robotem. Využijte prostředků vývojového prostředí ROS (Robot Operating System) a komerčních robotů TurtleBot.
3. Shora navržený systém realizujte v minimálně 2 exemplářích, navrhněte základní scénáře pro experimentaci v oblasti problematiky udržování formací robotů a ověřte jeho funkčnost spolu se vhodným zadaným algoritmem. Dosažený výsledek dokumentujte krátkým videozáznamem.

Seznam odborné literatury:

[1] E. Rohmer, S. P. N. Signgh, M. Freese, "V-REP: a Versatile and Scalable Robot Simulation Framework," IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2013

[2] M. Quigley, K. Conley, B.P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng, "ROS: an open-source Robot Operating System," ICRA Workshop on Open Source Software, 2009

[3] B. Shucker, J. K. Bennett, "Virtual spring mesh algorithms for control of distributed robotic macrosensors," Department of computer science university of Colorado in Boulder, May 2005

Vedoucí bakalářské práce: Ing. Libor Přeučil, CSc.

Platnost zadání: do konce letního semestru 2017/2018

L.S.

prof. Dr. Ing. Jan Kybic
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 10. 1. 2017