

Diplomová práce



České  
vysoké  
učení technické  
v Praze

**F3**

Fakulta elektrotechnická  
Katedra řídicí techniky

## Přesné dokování mobilního robotu s využitím laserových senzorů

**František Brandštýl**

Vedoucí: Ing. Karel Košnar, Ph.D.  
Obor: Kybernetika a robotika  
Studijní program: Kybernetika a robotika  
Květen 2019



## Poděkování

Přál bych si poděkovat panu Ing. Karlu Košnarovi, Ph.D. za jeho cenné rady a pomoc během práce na tomto projektu. Jeho moudré vedení vedlo ke zdárnému dokončení této diplomové práce. Rovněž bych chtěl poděkovat všem členům skupiny mobilní a kolektivní robotiky CIIRC, kteří neváhali nabídnout pomocnou ruku při řešení různých problémů. V neposlední řadě bych rád poděkoval všem svým blízkým, ať už rodině či přátelům, za jejich podporu během celého studia.

## 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

Cílem této diplomové práce bylo vytvoření dokovacího algoritmu pro mobilní robot a jeho ověření v laboratorních podmínkách. Úvodní část práce se zabývá způsoby lokalizace pomocí laserových senzorů a navigace mobilních robotů. V teoretické části se nachází detailní rozbor úlohy a návrh jejího řešení. Dokování robotu bylo rozděleno na několik dílčích částí. Samostatná je část kalibrace senzorů, která je nezbytná z důvodu použití více laserových dálkoměrů. I samotné dokování je rozdělené na více částí tak, aby program fungoval v reálném čase. Celý algoritmus je popsán v kapitole implementace. V závěru práce je popsáno několik experimentů včetně dosažených výsledků. Pro testování byl použit robot TurtleBot a senzory Hokuyo.

**Klíčová slova:** Dokování robotu, laserový dálkoměr, mobilní robot

**Vedoucí:** Ing. Karel Košnar, Ph.D.

## Abstract

This diploma thesis had an aim to design a docking algorithm for mobile robot and validate it in laboratory conditions. The opening part of the thesis gives an overview of localisation techniques with laser scanners and mobile robot navigation. In the theoretical part of the thesis is the detailed analysis of the problem and a proposal of the solution. The algorithm was divided into several parts. The sensor calibration is separated from the rest and is necessary, because multiple scanners were used. The docking part was also split, so it can work realtime. The whole algorithm is described in the implementation part. In the final part of the thesis are described some experiments with accomplished results. For experimentation was used the TurtleBot robot and laser scanners Hokuyo.

**Keywords:** Mobile robot, docking, laser scanner

**Title translation:** Precise docking of mobile robot with laser scanners

# Obsah

<b>1 Úvod</b>	<b>1</b>	<b>3 Použité nástroje</b>	<b>17</b>
1.1 Definice úlohy	2	3.1 Robot Operating System	17
1.2 Lokalizace s pomocí laserových senzorů	2	3.2 Point Cloud Library	18
1.2.1 Autonomní vozidla	4	3.3 Použité balíčky	19
1.2.2 SLAM v uzavřených prostorech	5	3.3.1 Systém zpráv a kanálů v ROS	19
1.3 Navigace mobilních robotů	5	3.3.2 Ovládání laserových dálkoměrů	19
1.3.1 Metoda potenciálových polí	6	3.3.3 Point Cloud Library	20
1.3.2 Sledování bodů	7	3.3.4 Ovládání robotu	20
1.3.3 Všesměrový podvozek	7	3.4 TurtleBot 2e	20
<b>2 Teoretický rozbor</b>	<b>9</b>	3.5 Laserové dálkoměry Hokuyo	21
2.1 Laserové dálkoměry	9	<b>4 Implementace</b>	<b>23</b>
2.2 Iterative Closest Point	10	4.1 Kalibrace senzorů	23
2.3 Návrh řešení	12	4.1.1 Vstupy a výstupy	24
2.3.1 Stanovení cílového bodu	12	4.1.2 Funkce aligning	24
2.3.2 Kalibrace senzorů	13	4.1.3 Funkce filterPoints	25
2.3.3 Dokování robotu	14	4.2 Skládání scanů	25
		4.2.1 Vstupy a výstupy	26
		4.2.2 Funkce combineScans	26

4.3 Zarovnávání . . . . .	26	<b>6 Vyhodnocení experimentů</b>	<b>37</b>
4.3.1 Vstupy a výstupy . . . . .	26	6.1 Vicon . . . . .	37
4.3.2 Vlastní zpráva . . . . .	27	6.2 Kalibrace senzorů . . . . .	38
4.3.3 Funkce aligning . . . . .	27	6.2.1 Vytvoření referenčního scanu	38
4.4 Řízení . . . . .	28	6.3 Dokování v ose doku . . . . .	39
4.4.1 Vstupy a výstupy . . . . .	28	6.3.1 Malá vzdálenost od cílového bodu . . . . .	40
4.4.2 Regulace . . . . .	28	6.3.2 Velká vzdálenost od cílového bodu . . . . .	40
<b>5 Uspořádání experimentů</b>	<b>31</b>	6.4 Dokování mimo osu doku . . . . .	41
5.1 Použitý robot . . . . .	31	6.4.1 Dokování podél levé stěny . . .	41
5.2 Realizace doku . . . . .	31	6.4.2 Dokování podél pravé stěny .	41
5.3 Kalibrace senzorů . . . . .	32	6.5 Celkové vyhodnocení . . . . .	42
5.4 Dokování v ose doku . . . . .	32	<b>7 Závěr</b>	<b>45</b>
5.4.1 Malá vzdálenost od cílového bodu . . . . .	33	<b>A Literatura</b>	<b>47</b>
5.4.2 Velká vzdálenost od cílového bodu . . . . .	34	<b>B Zadání práce</b>	<b>49</b>
5.5 Dokování mimo osu doku . . . . .	34		
5.5.1 Dokování podél levé stěny . . .	34		
5.5.2 Dokování podél pravé stěny .	35		

## Obrázky

1.1 Princip detekce objektů pomocí laserového dálkoměru. Převzato z: [18] .....	3	5.1 Robot vybavený senzory .....	32
1.2 Ukázka scanu okolí autonomního vozidla. Převzato z: [2] .....	4	5.2 Reálná podoba doku .....	32
1.3 Scanování interiéru. Převzato z: [6]	5	5.3 Pozice robotu během vytváření referenčního mračna bodů .....	33
1.4 Metoda potenciálových polí. Převzato z: [7] .....	6	5.4 Pozice robotu při dokování z malé vzdálenosti v ose doku .....	33
2.1 Porovnání 3D a 2D laserového scanu. Převzato z: [17] .....	10	5.5 Pozice robotu při dokování z velké vzdálenosti v ose doku .....	34
2.2 Ukázka zarovnání pomocí ICP. Převzato z: [11] .....	11	5.6 Pozice robotu při dokování podél levé stěny .....	35
3.1 Robot Operating System. Převzato z: [16] .....	18	5.7 Pozice robotu při dokování podél pravé stěny .....	35
3.2 Point Cloud Library. Převzato z: [15] .....	19	6.1 Vliv přesnosti kalibrace na skládání scanů .....	38
3.3 Robot TurtleBot 2e. Převzato z: [14] .....	21	6.2 Vytvoření referenčního mračna bodů .....	39
3.4 Senzor Hokuyo URG-04LX-UG01. Převzato z: [13] .....	21	6.3 Zarovnání mračen bodů s rostoucí vzdáleností .....	39
4.1 Schéma dokovacího algoritmu ..	25	6.4 Výsledky dokování z malé vzdálenosti v ose doku .....	40
4.2 Schéma regulace robotu .....	29	6.5 Výsledky dokování z velké vzdálenosti v ose doku .....	41
		6.6 Výsledky dokování podél levé stěny .....	42

6.7 Výsledky dokování podél pravé stěny .....	42
6.8 Všechny dosažené dojezdové pozice .....	43

## Tabulky

6.1 Odchyly při dokování z malé vzdálenosti v ose doku .....	40
6.2 Odchyly při dokování z velké vzdálenosti v ose doku .....	40
6.3 Odchyly při dokování podél levé stěny .....	41
6.4 Odchyly při dokování podél pravé stěny .....	42
6.5 Shrnutí průměrných odchylek ..	42





# Kapitola 1

## Úvod

Velkým trendem současnosti je automatizace výroby a stále se zvyšující počet úkonů, které místo člověka vykonávají stroje. Například v průmyslu, kde je potřeba manipulovat s těžkými břemeny, se skvěle uplatňují robotická ramena. Dobrým příkladem je například automobilový průmysl. Robotické manipulátory jsou schopny přesně usazovat jednotlivé součásti, čímž se zvyšuje efektivita a rychlost výroby. Zároveň se tím snižuje nebezpečí nehod nebo poškození výrobků.

Současně dochází k velkému rozvoji mobilních robotů. Tyto roboty se dnes běžně používají v domácnostech jako takzvané chytré vysavače nebo sekačky. Zároveň dochází k velikým pokrokům v oblasti samořídících automobilů, které můžeme rovněž považovat za mobilní roboty. Velké uplatnění však nacházejí rovněž v průmyslu. Již dlouho se hovoří o využití robotů k převážení nákladů například ve skladech, čímž by se opět zefektivnil jejich provoz a snížilo riziko nehod při manipulaci s těžkými a objemnými břemeny.

Z této myšlenky vychází téma této diplomové práce. Jejím předmětem je autonomní dokování robotu pomocí laserových senzorů. Předpokladem této úlohy je průmyslový nákladní robot převážející těžké náklady například ve skladišti. Takový robot potřebuje stanici, kam se bude vracet kvůli dobíjení nebo aby uvolnil prostor v době nečinnosti. Zde bude tedy zapotřebí přesného zadokování s co největší přesností.

## 1.1 Definice úlohy

Jak již bylo zmíněno, předmětem této práce není autonomní pohyb na velké vzdálenosti. Pod pojmem dokování je myšleno poježdění robotu směrem k požadované pozici v doku na vzdálenost maximálně 1 až 2 metry. Předpokládáme tedy, že robot je schopen se nějakým způsobem dostat k doku ovšem s velkou odchylkou od požadované pozice. Z toho vyplývá, že výchozí pozice pro dokování není vždy stejná.

Vzhledem k předpokládanému prostředí uvažujeme průmyslový robot o délce přibližně 6 metrů. Tento robot disponuje všesměrovým podvozkem, což do jisté míry snižuje nároky na řízení a zjednodušuje jeho pohyb v okolí doku. Pro účely práce a laboratorního vývoje se však tento robot nehodí. Proto byl v této práci použit TurtleBot 2e od firmy Kobuki, což je malý robot vhodný pro testování v laboratoři. Turtlebot bohužel nemá všesměrové, ale diferenciální řízení. To v tomto případě znamená vyšší složitost řízení, což ale nemá na výsledek velký vliv.

Pro lokalizaci robotu mají být použity laserové dálkoměry. Předpoklad použití robotu pro převoz nákladů však vylučuje jeden laserový dálkoměr s rozsahem 360 stupňů umístěný navrchu robotu. Proto se předpokládá umístění senzorů v rozích robotu tak, že jejich společný záběr je zmíněných 360 stupňů. Z toho vychází první problém, který je potřeba v této práci vyřešit a to je kalibrace jednotlivých senzorů ve smyslu synchronizace detekovaných bodů. Tím získáme celkový obraz okolí robotu v jednotných souřadnicích.

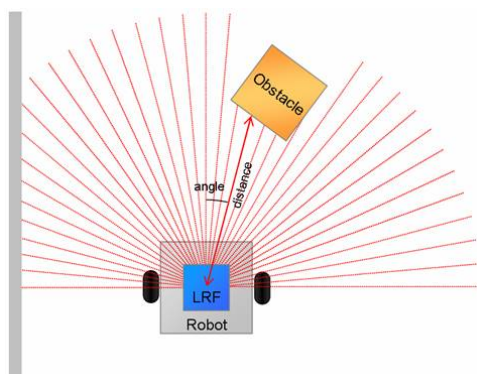
Další částí práce je poté navigace a řízení robotu v reálném čase s cílem dosáhnout co nejnižší odchylky od požadované pozice. Vzhledem k tomu, že dokování by mělo probíhat na krátkou vzdálenost, lze předpokládat, že jediným objektem nacházejícím se v okolí robota bude dok. Z toho důvodu není zapotřebí implementovat systém vyhýbání se překážkám (collision avoidance). Konkrétní postup řešení zmíněných problémů je detailně popsán v dalších kapitolách této práce.

## 1.2 Lokalizace s pomocí laserových senzorů

Laserové senzory se vyznačují vysokou přesností a to i na vzdálenosti jednotek až desítek metrů. V současnosti se využívají v mnoha odvětvích. Co se týče

využití při lokalizaci mobilních robotů, jsou laserové senzory pravděpodobně nejpoužívanější. Jejich výhodou je kromě přesnosti i vysoká frekvence měření. Hodí se proto nejen k mapování okolního prostředí, ale i pro detekci dynamických překážek a objektů. Často se používá také metoda SLAM (Simultaneous localization and mapping), tedy současná lokalizace a mapování. I zde se skvěle uplatňují laserové senzory. V článku [1] je popsáno velké množství způsobů lokalizace používaných v současnosti. Jsou zde popsány i metody využívající laserová data nebo kombinace různých senzorů.

Lokalizaci pomocí laserových dat lze rozdělit na dvě základní skupiny. První z nich je lokalizace ze samotného laserového scanu. V tomto případě se jedná o lokalizaci danou čistě vzdáleností od detekovaných překážek. Dá se říci, že se v tomto případě jedná o relativní lokalizaci, protože robot nemá žádné informace o širším okolí a reaguje pouze na nejbližší objekty detekované laserovým senzorem. Tento přístup se nejčastěji používá k systému vyhýbání se překážkám (collision avoidance).



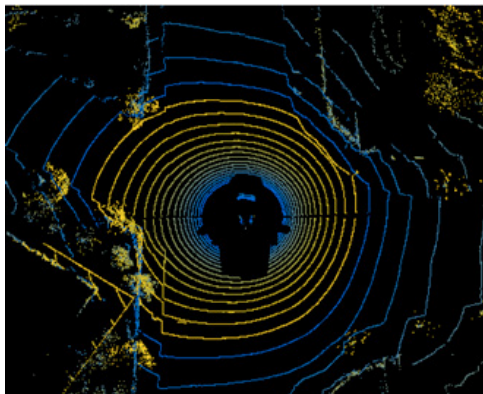
**Obrázek 1.1:** Princip detekce objektů pomocí laserového dálkoměru. Převzato z: [18]

Druhým typem je lokalizace pomocí předem známé mapy. V tomto případě se aktuálně zachycený scan určitým způsobem porovnává s danou mapou. Robot tak dokáže určit svou pozici v prostoru a jedná se tak o absolutní lokalizaci. Díky tomu lze například plánovat trajektorii robotu v mapě. K určování polohy v mapě se používají různé algoritmy jako je například Iterative Closest Point (ICP) nebo Particle filter. Tato mapa může být předem vytvořena pomocí mapování laserovým senzorem nebo je možné použít jiné typy map, ze kterých lze určitým způsobem získat potřebné význačné body (features), které se poté porovnávají s laserovými daty. Některé případy konkrétního využití jsou uvedeny v následující části.

### 1.2.1 Autonomní vozidla

V článku [2] se používá 3D laserový dálkoměr k mapování a lokalizaci autonomního vozidla. Senzor je umístěn na střeše a snímá okolí vozidla. V mapě vytvořené ze sensorových dat se poté detekují okraje silnice. Vzhledem k tomu, že systém je určen pro navigaci ve městech, využívá se toho, že obrubníky na okrajích silnice jsou o 10 až 15 centimetrů vyšší než vozovka. Díky tomu a faktu, že okraje chodníků jsou téměř dokonale svislé, se dají okraje vozovky z laserových dat velmi dobře detekovat.

Poté, co jsou okraje vozovky detekovány, je vytvořen profil vozovky v okolí vozidla. Ten se následně pomocí ICP algoritmu porovnává s detailní digitální mapou a tímto způsobem se vozidlo lokalizuje. Tento algoritmus je určen spíše pro zpřesnění lokalizace vozidla například při použití systému GPS (Global positioning system), který má ve městech mezi výškovými budovami nižší přesnost.



**Obrázek 1.2:** Ukázka scanu okolí autonomního vozidla. Převzato z: [2]

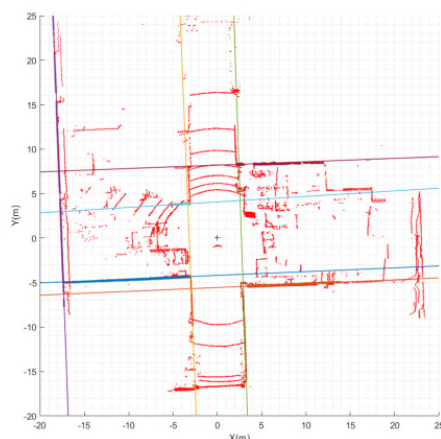
Podle výsledků provedených experimentů je přesnost lokalizace v porovnání s daty ze samotné GPS výrazně lepší. Nedostatky tohoto přístupu jsou však ty, že algoritmus nedokáže detekovat okraje silnice tam, kde jsou buď zastíněny jinými předměty (například zaparkovaná vozidla), nebo tam, kde nejsou na okrajích obrubníky.

Technologie autonomních vozidel se v současnosti velmi rychle rozvíjí. Proto se mnoho výzkumných týmů zabývá i použitím laserových senzorů k navigaci těchto vozidel. Jejich výhodou je, že kromě lokalizace v mapě je lze zároveň využít například k detekci ostatních vozidel nebo chodců ve vozovce. S relativně malými rozdíly se touto tematikou zabývají například články [3], [4], [5].

## 1.2.2 SLAM v uzavřených prostorech

Jak bylo zmíněno výše, laserové senzory se hodí pro úlohu současného mapování a lokalizace. V článku [6] je příklad aplikace této metody k mapování vnitřních prostor. Výstupem této metody může být například mapa prostor určená k pozdější úloze lokalizace robotu při autonomní jízdě. Výsledek mapování může však být použit také jako 3D model vnitřních prostor budovy, který lze použít třeba jako podklad pro stavební úpravy interiéru. Využití laserových senzorů k lokalizaci uvnitř budov je výhodné zejména proto, že zde obvykle bývá slabý signál GPS, což je jinak zřejmě nejjednodušší způsob lokalizace v místech, kde je signál dostupný.

Při úloze současné lokalizace a mapování se využívá význačných bodů detekovaných v datech. Ve vnitřních prostorech jsou takovými body například rohy zdí. Tyto význačné body jsou důležité pro lokalizační část úlohy, kde se porovnávají jejich vzájemné pozice s pozicí robotu. Postupným skládáním jednotlivých scanů se vytváří mapa prostor. Tento proces bývá obvykle výpočetně náročný, zejména v případech použití vícevrstvých laserů, které poskytují ohromná množství dat. Proto se například v citovaném článku [6] snaží tento algoritmus upravit a snížit tak jeho náročnost.



Obrázek 1.3: Scanování interiéru. Převzato z: [6]

## 1.3 Navigace mobilních robotů

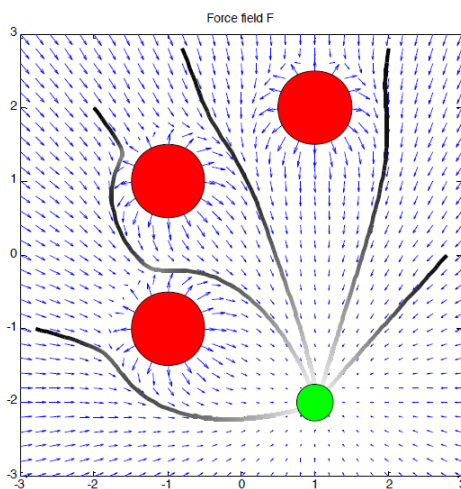
Navigace robotu je způsob, jakým se robot pohybuje v prostoru a jak se dostává na požadované pozice. Ve výsledku je potřeba použít nějaký systém

regulace pro výpočet rychlostí robotu nebo jednotlivých kol. Nejprve je však nezbytné nějakým způsobem reprezentovat pozici robotu v prostoru a také pozice objektů, které se nacházejí v okolí robotu. Nejjednodušším a nejpoužívanějším přístupem je takzvaná mřížka obsazenosti. Jedná se o souřadnicový systém s diskretním krokem, který slouží pro ukládání pozic datekovaných objektů. Každé pole mřížky nese hodnotu pravděpodobnosti toho, že se na dané pozici nachází překážka. Jedná se tedy o mapu okolí robotu.

Ve chvíli kdy je takováto mapa k dispozici, je možné zvolit bod, na který se má robot dostat a spočítat potřebné hodnoty pro jeho řízení. Používají se různé způsoby, jak spočítat potřebné hodnoty a zajistit, aby se robot dostal na správné místo.

### 1.3.1 Metoda potenciálových polí

Tato metoda [7] využívá v jádru jednoduchého systému přitažlivých a odpudivých sil. Přitažlivou sílu má bod, na který se má robot přesunout. Odpudivé síly mají překážky a místa, kterým je potřeba se vyhnout. Všechna tato místa působí na své okolí a vznikají tak potenciálová pole. V každém bodě v prostoru je poté počítána výslednice sil, které v daném místě působí. Robot se poté pohybuje směrem nejnižšího odporu.



Obrázek 1.4: Metoda potenciálových polí. Převzato z: [7]

### ■ 1.3.2 Sledování bodů

Další často používanou metodou je sledování jednotlivých bodů v mapě. Tyto body tvoří trajektorii, která vede do požadovaného bodu. K plánování trajektorie v mřížce obsazenosti se používají známé algoritmy jako například A\*, nebo Dijkstrův algoritmus. Poté je potřeba tuto trajektorii sledovat. Existuje opět několik různých přístupů.

Jeden z nejjednodušších je pohybovat se jednoduše vždy k nejbližšímu bodu. Ve chvíli, kdy je tento bod dosažen (nebo jeho blízké okolí), pokračuje robot k dalšímu bodu. Tento přístup není však příliš efektivní. Proto existuje mnoho vylepšení a úprav. Známa je například metoda Follow the carrot („sledování mrkve“). V tomto případě nesleduje robot bod, který je k němu nejbližší, ale bod, který je vždy v dané vzdálenosti od robotu. Tato změna v algoritmu pomáhá vyhladit trajektorii robotu i plynulost jeho pohybu. K vyhlazení trajektorie se používá také prokládání jejích bodů určitými křivkami (spline) [8].

K dosažení jednotlivých bodů trajektorie se používají regulátory, které v reálném čase počítají rychlost jednotlivých motorů potřebnou pro pohyb směrem k danému bodu. Zřejmě nejnámější a nejčastěji používaný je takzvaný PID regulátor. Regulátor se snaží minimalizovat odchylku, kterou může být například úhel od přímého směru, vzdálenost od cílového bodu a tak dále.

### ■ 1.3.3 Všesměrový podvozek

Navigace všesměrového robotu je obdobná jako u ostatních typů podvozků. Výhodou je, že kola robotu se mohou natočit libovolným směrem a i robot se tak může pohybovat libovolně. V praxi to znamená, že robot může jet jak dopředu, tak kolmo do strany aniž by se muselo měnit vlastní natočení robotu. Další výhodou je, že robot může velmi snadno měnit natočení na místě aniž by se musel pohybovat lineárně. Tyto vlastnosti mu umožňují projet i místy, kde by se standardní robot například s ackermanovým podvozkem nedostal.





## Kapitola 2

### Teoretický rozbor

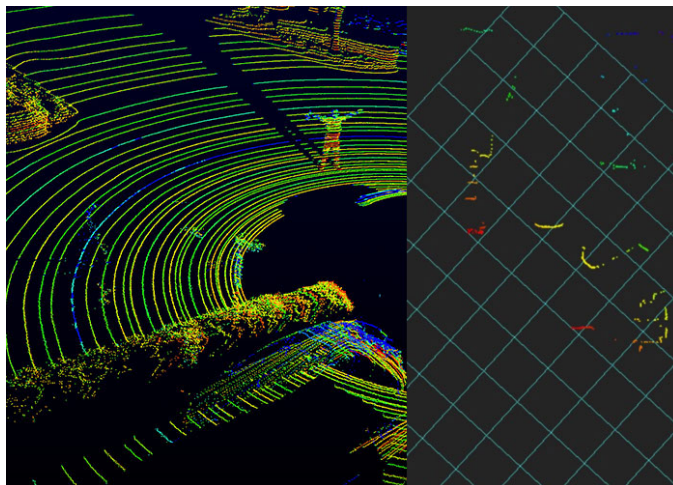
V této části práce je předvedeno navržené řešení problému, jeho rozdělení na podúlohy a popis nástrojů potřebných k jejich řešení. Zadaný problém se v zásadě dělí na dvě základní části. První z nich je kalibrace jednotlivých senzorů a druhou dokování robotu v reálném čase.

Jak už bylo zmíněno dříve, robot je opatřen několika laserovými dálkoměry, které slouží k jeho lokalizaci vzhledem k doku a jeho cílové pozici. V ideálním případě by měl robot mít čtyři senzory a mít tak kompletní záběr 360 stupňů. Pro účely této práce a laboratorních testů byly však použity pouze dva senzory. Hlavním důvodem byl nedostatek těchto senzorů v laboratoři. Pro návrh řešení a jeho realizaci v laboratorních podmínkách však dva senzory postačují.

#### 2.1 Laserové dálkoměry

V odborné literatuře se tento typ senzorů označuje zkráceně LIDAR (Light Detection And Ranging). Už z názvu vyplývá, že využívá rozmítané laserové paprsky k měření vzdálenosti jednotlivých bodů a objektů. Vzdálenost se počítá z doby šíření vyslaného pulsu, který se odrazí od snímaného objektu. Pod pojmem laser jsou myšleny paprsky infračerveného záření, což je elektromagnetické záření o vlnové délce 750 až 1050 nanometrů. Díky tomu, že se toto záření šíří rychlostí světla, pracují laserové dálkoměry s vysokou frekvencí měření. Jejich další výhodou je vysoká přesnost a relativně velká odolnost vůči rušení.

V praxi se využívají senzory, které měří buď pouze v jedné nebo hned v několika rovinách. Výstupem z měření je mračno bodů (Point cloud). V případě laseru snímajícího v jedné rovině získáme 2D obraz okolního prostředí. Při využití vícerovinného lidarů dostáváme 3D obraz okolí senzoru. Na obrázku 2.1 je vidět rozdíl mezi 2D a 3D snímáním. Oba tyto přístupy mají své uplatnění v závislosti na požadavcích konkrétní úlohy.



Obrázek 2.1: Porovnání 3D a 2D laserového scanu. Převzato z: [17]

## 2.2 Iterative Closest Point

Iterative Closest Point (ICP) [10] je algoritmus určený k minimalizaci rozdílu dvou mračen bodů ve smyslu vzdáleností jednotlivých bodů. Jedno z mračen je zpravidla určeno jako referenční a druhé je postupně transformováno tak, aby došlo k minimalizaci vzdáleností korespondujících bodů. Transformací rozumíme rotaci a translaci mračna bodů jako celku, nikoli změnu vzájemné polohy jeho jednotlivých bodů.

Z matematického hlediska můžeme definovat dvě mračna bodů jako  $X = x_1, \dots, x_n$  a  $P = p_1, \dots, p_n$ . Cílem je spočítat translaci  $T$  a rotaci  $R$ , které minimalizují následující rovnici.

$$E(R, T) = \frac{1}{N_p} \sum_{i=1}^{N_p} (x_i - Rp_i - T)^2, \quad (2.1)$$

kde  $x_i$  a  $p_i$  jsou korespondující body z obou mračen. Pro zjednodušení

výpočtu se používá přístup, kdy od všech bodů daného mračna odečteme těžiště tohoto mračna. Těžiště mračna  $X$  lze spočítat jako:

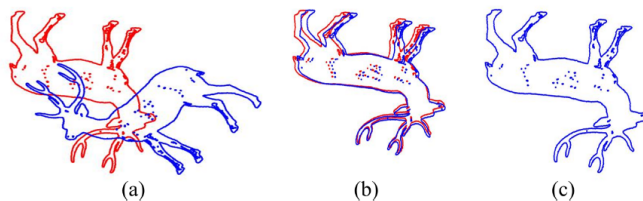
$$\mu_x = \frac{1}{N_x} \sum_{i=1}^{N_x} x_i \quad (2.2)$$

Obdobně lze spočítat těžiště  $\mu_p$  i pro mračno  $P$ . Následně odečteme získané hodnoty od všech bodů v mračnech  $X$  a  $P$  a získáme tak nová mračna bodů  $X'$  a  $P'$ .

$$X' = x_i - \mu_x = x'_i \quad (2.3)$$

$$P' = p_i - \mu_p = p'_i \quad (2.4)$$

S těmito novými mračny se poté vyřeší daná optimalizační úloha například pomocí algoritmu Singular Value Decomposition (SVD). Vyřešením této úlohy je získána požadovaná transformační matice. Existuje více variant algoritmu ICP, které se liší hlavně v tom, jakým způsobem jsou přiřazovány korespondující body. Nejjednodušší je varianta, kdy je ke každému bodu z prvního mračna přiřazen nejbližší bod z druhého mračna. Dále se používá například přiřazení nejbližšího bodu na normále nebo vzdálenost bodu od roviny.



**Obrázek 2.2:** Ukázka zarovnání pomocí ICP. Převzato z: [11]

Průběh algoritmu označujeme jako zarovnávání (aligning). K zarovnávání dochází v cyklech. V každém cyklu jsou znovu vybírány korespondující body. Po přiřazení bodů se řeší optimalizační úloha popsaná výše. Výsledkem je transformace (rotace a translace) jednoho z mračen. Tato transformace se poté aplikuje na dané mračno bodů. Dále se tento proces opakuje v dalších cyklech dokud není splněna ukončovací podmínka. Nutno dodat, že algoritmus je poměrně citlivý na počáteční odhad transformace, který může mít velký vliv na konečný výsledek. Pro zlepšení výsledku výpočtu se také používá odstranění nejhorších párů bodů. Buď se přímo určí hraniční vzdálenost a



details by nemusely odpovídat reálnému doku, který se působením různých vlivů může časem deformovat.

Místo vytváření modelu doku tak lze použít jednodušší variantu. Jako referenci pro algoritmus ICP lze použít přímo scan doku z pohledu robota v cílovém bodě. Výhodou tohoto přístupu je, že scan bude odpovídat aktuálnímu stavu doku. Scan rovněž přesně odpovídá snímku, který by měl robot pořídit, pokud se bude nacházet v cílovém bodě na konci dokování. Tímto přístupem by tedy mělo být docíleno toho, že přesnost lokalizace robota bude velmi vysoká v okolí cílového bodu. Tím by rovněž měla být zvýšena celková přesnost dokování.

Zvolením tohoto přístupu byl zároveň elegantně vyřešen problém stanovení cílového bodu. Ten lze nyní téměř libovolně určit tak, že robot fyzicky postavíme na požadovanou pozici a provedeme na ní vytvoření referenčního scanu. Jistá omezení jsou však kladena kalibrací senzorů, která je popsána v následující části.

### ■ 2.3.2 Kalibrace senzorů

Jak bylo zmíněno v úvodu, kalibrací senzorů je myšleno vzájemné zarovnání scanů z obou senzorů tak, aby vznikl jeden celkový scan v jednotných souřadnicích. Je tedy potřeba oba scany zarovnat tak, aby se odpovídající části překrývaly a správně navazovaly. K tomuto zarovnání lze využít algoritmus ICP. Přibližným změřením vzájemné polohy obou senzorů po jejich umístění na robot lze získat odhad transformace polohy jednoho ze senzorů.

Konkrétní kalibrace se provádí tak, že do zorného pole obou senzorů se umístí předmět, který má dostatek nerovností nebo význačných bodů. Tím se zajistí to, že po zarovnání scanů pomocí ICP algoritmu se překryjí odpovídající tvary. Vhodným objektem je například pravoúhlý dok. Přesnost kalibrace lze zvýšit přidáním dalších nerovností na stěnách doku. Poté je potřeba umístit robot tak, aby se tyto význačné body (například roh doku) nacházely v zorném poli obou senzorů. Nakonec je pomocí ICP algoritmu získána přesná transformace jednoho ze scanů. Po aplikování této transformace a spojení obou scanů vznikne výsledný kompletní scan.

Použití doku pro kalibraci senzorů umožňuje spojit kalibraci a stanovení cílové pozice pro dokování. Pokud provedeme kalibraci na cílovém bodě, získáme zároveň scan sloužící jako reference pro pozdější lokalizaci robota.



lze tímto postupem dosáhnout velké přesnosti pohybu. Další výhodou je, že je mnohem jednodušší ladit nezávisle úhlový a lineární regulátor. Podobný způsob regulace by navíc byl použit i v případě všesměrového podvozku.

Všechny části dokovacího algoritmu běží v reálném čase, aby byly k dispozici aktuální hodnoty a bylo dosaženo co nejvyšší přesnosti dokování. Jak bylo zmíněno dříve, přesnost lokalizace a tedy i dokování by měla být nejvyšší právě v okolí cílového bodu. Pohyb robotu musel být přizpůsoben ještě faktu, že během této práce byly použity pouze dva laserové dálkoměry místo celkových čtyř. Tyto dva dálkoměry byly zaměřeny na oblast před robotem a částečně po obou jeho stranách. Proto se robot musí pohybovat vždy přední stranou otočenou směrem k doku. Jinak by mohlo dojít ke ztrátě kontaktu s dokem a správné dokování by bylo znemožněno.





## Kapitola 3

### Použité nástroje

Tato kapitola popisuje nástroje využité při tvorbě dokovacího algoritmu. Základem je Robot Operating System (ROS), ve kterém byl celý program vytvářen. Dále jsou zmíněny některé konkrétní knihovny nezbytné pro fungování systému. V této kapitole je rovněž popsán robot použitý v experimentální části práce.

#### 3.1 Robot Operating System

Robot Operating System (ROS) je software framework používaný při práci a experimentování s robotickými systémy. Jedná se o volně dostupný systém, jehož hlavní přínos spočívá v možnosti jednoduchého znovupoužití aplikací a programů, které jsou v něm vyvinuty. Každá taková aplikace je zahrnuta v balíčku (package). V systému ROS lze tyto balíčky jednoduše zkompileovat a používat nebo je importovat do daného programu podobně jako knihovny. Tímto způsobem vzniká platforma, kde mohou lidé z celého světa sdílet své objevy a pokroky, navzájem je využívat, rozšiřovat a navazovat na ně. Kromě toho zde existuje množství ovládacích programů a rozhraní pro senzory od jejich výrobců, což značně zjednodušuje práci s nimi.

Další výhodou systému ROS je, že slouží jako komunikační rozhraní umožňující snadné posílání informací mezi jednotlivými částmi větších a rozvětvenějších programů nebo mezi jednotlivými programy tvořícími větší celek. Tyto informace a data se posílají ve formě zpráv (message). Existuje množství

základních zpráv ve formě konkrétních datových typů určených například pro přenos scanu z laserového dálkoměru. Kromě toho lze vytvořit vlastní nový typ zprávy na míru konkrétního programu.



**Obrázek 3.1:** Robot Operating System. Převzato z: [16]

Systém ROS rovněž obsahuje některé pomocné nástroje, které slouží například k hledání chyb, optimalizaci programu nebo k vizualizaci výsledků. Jednotlivé aplikace se spouštějí v takzvaných uzlech (node). V každém uzlu běží daný program nezávisle. Pro komunikaci mezi uzly slouží již zmíněné zprávy, které se odesílají na zvláštní kanály (topic). Každý uzel může přijímat a odesílat zprávy na libovolný počet kanálů.

Podporovanými jazyky pro psaní programů v ROS jsou C++ a Python. Kromě již zmíněných funkcí zajišťovaných přímo systémem ROS, může tedy každý program využívat stejné funkce jako při samostatné kompilaci. Těmito funkcemi je myšleno například načítání vstupů, psaní na výstup, využívání knihoven daného jazyka a tak dále.

## 3.2 Point Cloud Library

Pro práci s mračky bodů byla použita knihovna Point Cloud Library (PCL) [9]. Jedná se o volně přístupnou knihovnu, která obsahuje například funkce pro převádění mračen bodů na různé datové typy nebo základní algoritmy pro zpracování mračen. Jedním z nejdůležitějších algoritmů, který je základem řešení zadaného problému, je dříve zmiňovaný algoritmus ICP.



**Obrázek 3.2:** Point Cloud Library. Převezato z: [15]

## ■ 3.3 Použité balíčky

Celá práce byla vypracována v systému ROS a pro psaní kódu byl použit programovací jazyk C++. Bylo použito několik balíčků, z nichž některé jsou přímo ve výchozím stavu obsažené v tomto systému. Tyto balíčky lze přirovnat ke knihovnám jazyka C++ a není tedy třeba se o nich explicitně zmiňovat. Je však vhodné zmínit některé větší balíčky, které jsou nezbytné pro fungování celého algoritmu.

### ■ 3.3.1 Systém zpráv a kanálů v ROS

Pro posílání zpráv na kanál se používá takzvaný publisher (vydavatel). Tato funkce jednoduše odesílá danou zprávu a není potřeba ji nijak dále upravovat. Naopak pro čtení zpráv slouží subscriber (odběratel). Ten potřebuje pro správné fungování funkci callback. To je funkce, kterou je potřeba implementovat tak, aby správně zpracovávala přijaté zprávy. Například ukládat je do dané proměnné a podobně.

### ■ 3.3.2 Ovládání laserových dálkoměrů

Pro správnou funkci laserových dálkoměrů byl použit balíček „hokuyo\_node“. Jedná se o oficiální nástroj pro práci s laserovými dálkoměry Hokuyo. Tento balíček mimo jiné zajišťuje to, že data ze senzorů jsou k dispozici na zvláštním kanálu v rámci ROS.

### ■ 3.3.3 Point Cloud Library

Tato knihovna byla zmíněna již dříve. Pro účely této práce byla použita její verze pro ROS. Balíček se jmenuje jednoduše „pcl“. Obsahuje množství funkcí pro převody mezi datovými typy i přímo pro práci s mračky bodů. Některé z těchto funkcí jsou zmíněny dále podle toho, kde byly použity.

### ■ 3.3.4 Ovládání robotu

Balíček „TurtleBot“ je oficiální nástroj od výrobce robotu firmy Kobuki. Díky tomuto ovladači je k dispozici množství diagnostických informací o stavu robotu. Nezbytnými funkcemi je získávání odometrických dat robotu a posílání řídicích signálů.

## ■ 3.4 TurtleBot 2e

Jak bylo zmíněno v úvodu, byl v této práci použit mobilní robot TurtleBot 2e od firmy Kobuki. Jedná se o malý mobilní robot, který je přímo určen pro vývoj a experimentování v oblasti mobilní robotiky. Jeho základna je přizpůsobena pro připevnění různých senzorů a malého palubního počítače. Díky napájení z baterie může robot pracovat zcela samostatně a bezdrátově. To se hodí například při úlohách jako je prohledávání či mapování větších prostor.

Robot je postaven na diferenciálním podvozku. To znamená, že má dvě nezávisle řízená kola na levé a pravé straně podvozku. Další dvě neřízená kolečka vepředu a vzadu slouží pouze pro stabilizaci podvozku. Výhodou tohoto uspořádání je, že robot se může snadno otáčet na místě. Díky tomu má také poměrně přesnou odometrii, která umožňuje sledovat jeho polohu a natočení.



Obrázek 3.3: Robot TurtleBot 2e. Převzato z: [14]

### 3.5 Laserové dálkoměry Hokuyo

Pro účely této práce byly použity dva jednorovinné lidary Hokuyo URG-04LX-UG01. Použité senzory Hokuyo jsou schopny detekovat překážku až do vzdálenosti 5,6 metru. Doba potřebná pro pořízení kompletního scanu je přibližně 100 milisekund. To odpovídá frekvenci 10 Hz. Maximální záběr senzoru je 240 stupňů s úhlovým rozlišením 0,352 stupně. Udávaná přesnost je  $\pm 3$  centimetry, ale v kombinaci velkého množství paprsků je reálná přesnost výrazně vyšší.



Obrázek 3.4: Senzor Hokuyo URG-04LX-UG01. Převzato z: [13]



## Kapitola 4

### Implementace

V této kapitole je popsána konkrétní implementace všech dříve navržených postupů. Všechny popsané funkce byly sdruženy ve společném balíku nazvaném `laser_node`. Vytvořené programy jsou k dispozici na přiloženém disku. Algoritmus je rozdělen do několika souborů, které se po spuštění v ROS chovají jako jednotlivé uzly. Každý z těchto uzlů má v této kapitole vlastní sekci, ve které je detailně popsán princip jeho funkce a provázanost s ostatními uzly.

#### 4.1 Kalibrace senzorů

Program kalibrace senzorů je více oddělený od ostatních. Spouští se samostatně a jeho cílem je vytvoření referenčního scanu a transformační matice pro jeden ze senzorů. Samotnou kalibraci je potřeba provést pouze v případě, že není k dispozici potřebný soubor s transformační maticí. Jinak lze tento program spouštět pouze za účelem změny cílové pozice robotu bez toho, aby se ukládala nová transformace. Transformační matice se používají v následujícím tvaru. Kde  $\alpha$  je úhel rotace a  $x$ ,  $y$  posuny v daných souřadnicích.

$$\begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 & x \\ \sin(\alpha) & \cos(\alpha) & 0 & y \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.1)$$

### ■ 4.1.1 Vstupy a výstupy

Vstupem do kalibrace jsou dva kanály, na které posílají svá data jednotlivé laserové dálkoměry. Tyto kanály vytváří dříve zmíněný `hokuyo_node`. Data na těchto kanálech mají typ `LaserScan` a je tedy potřeba je po načtení převést na typ `PointCloud` se kterým se dále pracuje. Pro převod těchto typů se používá funkce z balíčku `laser_geometry`. V programu je také nutné rozlišovat mezi datovým typem `PointCloud` jako sensorovou zprávou, která se používá pro posílání zpráv přes kanály, a datovým typem `PointCloud`, se kterým pracuje knihovna PCL. Tento převod je zajišťován funkcí právě z knihovny PCL a nebude již dále zmiňován.

Výstupem programu jsou tři kanály se zprávami typu `PointCloud`. Jedná se o dva scany ze sensorů a výsledný složený scan po provedení kalibrace. Tyto zprávy slouží pouze pro vizualizaci výsledků kalibrace. Dále se ukládá výsledná transformace a to do souboru „transform.txt“. Posledním výstupem je referenční scan, ten se ukládá do souboru „reference.pcd“.

### ■ 4.1.2 Funkce aligning

Toto je hlavní funkce programu, ve které dochází k zarovnávání dat načtených ze sensorů. Funkce běží v cyklech, aby bylo možno dosáhnout co nejlepších výsledků zarovnání, například mírnou úpravou polohy nebo natočení robotu. Ve funkci se spouští algoritmus ICP z knihovny PCL. Je tedy nutné nastavit vstupy do algoritmu, kterými jsou data ze sensorů. Dále se používá odhad transformace. Ten byl získán změřením reálné polohy obou sensorů na robotu. Konkrétně se jako odhad používá pouze rotace o 225 stupňů. Nastavují se také hodnoty ukončovacích podmínek algoritmu. Konkrétně je omezen maximální počet iterací a pro předčasné ukončení slouží podmínka minimální odchylky transformačních matic.

Po provedení zarovnání algoritmem ICP se načte výsledná transformační matice. Ta se uloží do souboru pomocí funkce `saveTransform`. Tato funkce pouze uloží jednotlivé prvky matice jako text. Hodnoty z výsledné transformační matice se v dalším cyklu použijí jako odhad, což slouží ke zpřesnění výsledků. Ze dvou původních scanů se po aplikování transformace složí celkový scan. Ten se ještě transformuje pomocí matice s naměřenými hodnotami a to proto, aby počátek souřadnic mračna bodů nebyl v poloze senzoru ale uprostřed robotu. Výsledné mračno bodů se dále posílá do funkce `filterPoints`.

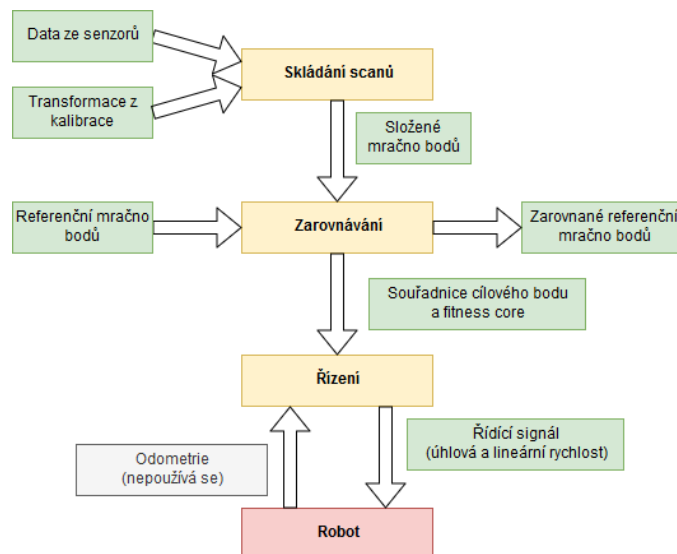


### 4.1.3 Funkce filterPoints

V této funkci se dále upravuje výsledné získané mračno bodů. Jsou z něj odstraněny všechny body, které jsou dále než 0,8 metru od počátku souřadnic. Toto je minimální experimentálně zjištěná vzdálenost, ve které je vidět celý dok. Tuto hodnotu lze změnit v závislosti na velikosti doku nebo vzdálenosti cílové pozice od doku. Důvodem tohoto kroku je to, že ve větší vzdálenosti od robotu se mohou nacházet pohybující se předměty nebo objekty které se později přesouvají. To by mohlo vnášet chyby do pozdějšího porovnávání mračen bodů během dokování robotu. Proto by měl ideálně být jediným objektem v referenčním scanu dok. Po odstranění přebytečných bodů se výsledné mračno uloží do souboru pomocí funkce z knihovny PCL a odešle se na výstup pro možnost vizualizace.

## 4.2 Skládání scanů

Tato funkce už podle názvu slouží ke skládání dat ze senzorů do celkového mračna bodů a to v reálném čase. Na obrázku 4.1 je zobrazeno schéma celého dokovacího algoritmu a propojení jeho jednotlivých částí, které jsou popsány dále.



Obrázek 4.1: Schéma dokovacího algoritmu

### ■ 4.2.1 Vstupy a výstupy

Vstupem jsou stejně jako v případě kalibrace dva kanály s daty ze senzorů. Dále se načítá transformace jednoho ze senzorů a to pomocí funkce `loadTransform`. Tato funkce načítá textová data ze souboru a ukládá je do transformační matice. Výstupem funkce je pouze jeden kanál, na který se posílá složené mračno bodů.

### ■ 4.2.2 Funkce `combineScans`

Stěžejní funkce tohoto programu. Nejprve transformuje jedno z dílčích mračen bodů pomocí transformační matice získané kalibrací. Poté obě mračna složí dohromady a výsledné mračno ještě transformuje tak, aby byl počátek souřadnic ve středu robotu. Celkový postup je velmi podobný postupu při kalibraci s tou výjimkou, že místo zarovnávání algoritmem ICP se rovnou používá známá transformace. Nakonec se stejně jako při kalibraci používá funkce `filterPoints`. Ta opět odstraňuje body ve větší vzdálenosti od robotu. Zvolená vzdálenost záleží na předpokládané odchylce od cílové polohy na začátku dokování (například 2 metry). Body, které jsou příliš daleko by zbytečně ztěžovaly správné zarovnání referenčního mračen bodů.

## ■ 4.3 Zarovnávání

V tomto programu se určuje vzdálenost a rotace robotu vzhledem k cílovému bodu. Díky tomu, že referenční i aktuální mračno bodů mají počátek souřadnic ve středu robotu, je výsledná transformace přímo transformací robotu a není potřeba ji dále upravovat.

### ■ 4.3.1 Vstupy a výstupy

Program načítá z jednoho kanálu celkový scan okolí robotu vytvořený v programu skládání scanů. Ze souboru se pomocí funkce `loadReference` načítá referenční scan vytvořený při kalibraci. Jako výstup slouží dva kanály. Na první se posílají získané souřadnice cílového bodu vzhledem k robotu. Na druhý kanál se posílá zarovnané referenční mračno bodů. Tento kanál slouží pro

vizualizaci výsledku zarovnávání. Pokud se vykreslí přes sebe toto zarovnané referenční mračno bodů a celkové mračno z programu skládání scanů, budou se v ideálním případě překrývat. Tímto způsobem lze ověřit, zda nedošlo například k chybné identifikaci doku.

### ■ 4.3.2 Vlastní zpráva

Pro účely posílání souřadnic cílového doku byl vytvořen vlastní typ zprávy. První tři položky zprávy tvoří souřadnice získané ve funkci zarovnávání. Jedná se o souřadnice  $x$  a  $y$  cílového bodu a úhel natočení vzhledem k referenčnímu mračnu bodů. Je třeba si uvědomit, že zmíněný úhel není úhlem který směřuje na souřadnice cílového bodu. Je to úhel, na který se musí robot otočit v cílovém bodě, aby viděl dok stejně jako při kalibraci.

Poslední položka zprávy je hodnota „fitness score“, která symbolizuje přesnost s jakou si obě mračna bodů po zarovnání odpovídají. Tato hodnota se používá k tomu, aby se odlišilo, zda robot vidí opravdu dok nebo nějaký jiný objekt. Díky tomu by nemělo docházet k chybné identifikaci doku. V případě, že se referenční mračno zarovná k jinému objektu než je dok, měla by hodnota fitness score být výrazně vyšší, což značí nízkou shodu mračen. Přítomnost objektů, které by byly hodně podobné doku, je ze zadání úlohy vyloučená, takže tento způsob kontroly detekce doku by měl být dostatečný.

### ■ 4.3.3 Funkce aligning

Nejdůležitější část tohoto programu, kde se počítá poloha cílového bodu. Pracuje se se dvěma dříve popsanými mračny bodů (reference a aktuální scan okolí robotu), které se použijí jako vstup do algoritmu ICP. Algoritmus opět používá omezení maximálního počtu iterací a jako ukončovací podmínka slouží minimální změna transformace mezi iteracemi. Při zarovnávání mračen bodů se používá odhad. Počáteční odhad lze nastavit podle toho, v jaké vzdálenosti od doku se předpokládá začátek dokování. V dalších cyklech se poté používá výsledek z předchozího výpočtu. Tím se jednak snižuje počet iterací potřebný k zarovnání mračen bodů a také se minimalizuje riziko chybné identifikace doku.

Při zarovnávání se transformuje referenční mračno bodů a aktuální scan je fixní. Důvodem je to, aby výsledné hodnoty transformační matice měly správné znaménko a odpovídaly faktu, že slouží k pohybu robotu. Po ukončení

algoritmu ICP se z finální transformační matice spočítají souřadnice cílového bodu a úhel natočení. Ty se posílají na výstup v podobě zprávy popsané v předchozí části společně s hodnotou fitness score, která určuje přesnost dosaženého zarovnání. Na výstup se rovněž odesílá zarovnané mračno bodů, aby bylo možné kontrolovat správnost zarovnání.

## ■ 4.4 Řízení

Posledním programem dokovacího systému je řízení robotu. Tvoří ho systém regulace, jehož úkolem je dojet s robotem na cílový bod. Jak už bylo řečeno, jedná se spíše o soustavu regulátorů, které pracují nezávisle. Důležitá je kontrola fitness score. Pokud je tato hodnota vyšší než 0,01 (odpovídá průměrné odchylce 0,01 metru mezi nejbližšími body zarovnaných mračen), žádné řízení neprobíhá a robot se nepohybuje. Tím se předchází tomu, aby se robot pokoušel dokovat v případě, kdy nastane chybná identifikace doku.

### ■ 4.4.1 Vstupy a výstupy

Program používá dva vstupní kanály. Na prvním jsou zprávy z programu zarovnávání. Ty obsahují souřadnice cílového bodu, na který musí robot dojet a hodnotu fitness score. Druhý kanál obsahuje zprávy s odometrií robotu. Ta se však nepoužívá v algoritmu dokování, ale slouží k ladění regulátorů a kontrole pozice robotu. Jediným výstupem programu je odesílání řídicích signálů na vstupní kanál robotu. Řídicí zprávy obsahují lineární a angulární rychlost.

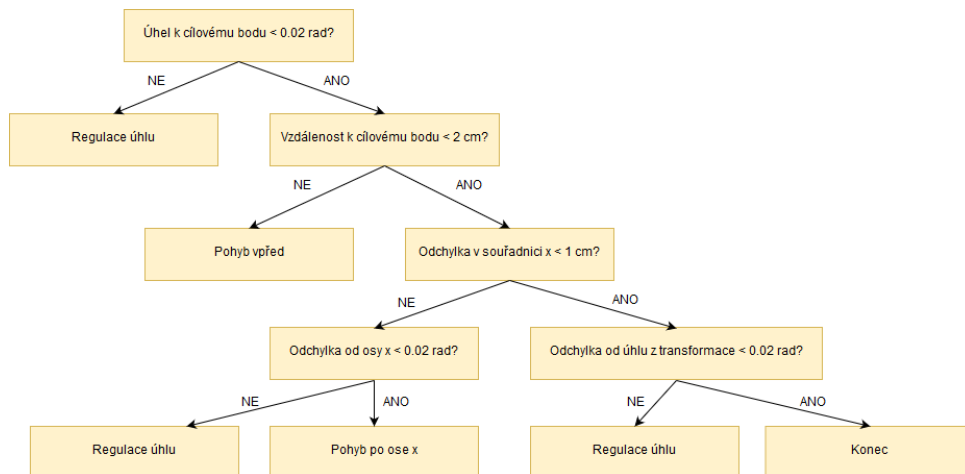
### ■ 4.4.2 Regulace

Regulaci zajišťuje funkce controller. Na začátku regulace se spočítá úhel cílového bodu vzhledem k přímému směru pohybu robotu (tedy osa  $x$ ). Dále se spočítá vzdálenost tohoto bodu od počátku souřadnic a tedy od středu robotu. Dále následuje soustava podmínek, podle jejichž splnění se spouští správný regulátor. Celý proces je navíc rozdělen do několika částí. První část odpovídá tomu, že vzdálenost robotu od cílového bodu je větší než 2 centimetry.

V tom případě se kontroluje, zda je úhel k cílovému bodu menší než 0,02 radiánu. Pokud ano robot se pohybuje konstantní rychlostí 0,07 metru za sekundu směrem k cílovému bodu. Pokud však úhlová odchylka překročí stanovenou hranici, pohyb se zastaví a PI regulátor ji sníží pod danou hodnotu. Tímto způsobem se robot pohybuje, dokud se jeho vzdálenost k cílovému bodu nesníží pod 2 centimetry. Poté nastává druhá část regulace.

Ve druhé části se pracuje pouze s odchylkou v souřadnici x. Experimentálně bylo totiž zjištěno, že odchylka v souřadnici y bývá mnohem menší. Pohybem po ose x se tedy tato odchylka minimalizuje a to pod hranici 1 centimetru. Úhlový regulátor opět zajišťuje minimální odchylku úhlu od osy x a poté se robot pohybuje buď dopředu nebo dozadu podle potřeby. Pokud je odchylka v souřadnici x menší než 1 centimetr, nastává poslední fáze dokování.

V poslední fázi už se reguluje pouze úhel natočení robotu. Robot se natočí na úhel z transformace získané v části zarovnávání. Po natočení na tento úhel by měl být robot ve stejné pozici jako při kalibraci. Tato závěrečná část dokování je částečně nadbytečná. Důvodem je, že při použití všesměrového podvozku by úhel natočení robotu odpovídal celou dobu úhlu z transformace. I při použití diferenciálního podvozku není závěrečné natočení nutné, protože robot se již nachází na cílové pozici a dokování je tedy dokončeno. Z principu úlohy je však vhodnější, když má robot na cílové pozici stejnou polohu jako při kalibraci a to včetně natočení.



Obrázek 4.2: Schéma regulace robotu



## Kapitola 5

### Uspořádání experimentů

V této kapitole jsou popsány experimenty, které byly provedeny pro ověření funkčnosti dokovacího systému a jeho přesnosti. Většina pokusů spočívá v provedení dokování z různých vzdáleností od doku a pod různými úhly. Kromě toho je mezi experimenty předvedena i kalibrace. Součástí této kapitoly je rovněž popis doku vytvořeného za účelem experimentování a podoba robotu včetně laserových senzorů a řídicího počítače.

#### 5.1 Použitý robot

Už dříve bylo zmíněno, že v práci byl použit TurtleBot 2e. Na obrázku 5.1 je vidět robot s připevněnými laserovými dálkoměry Hokuyo URG-04LX-UG01. Pro řízení robotu byl použit počítač Intel NUC.

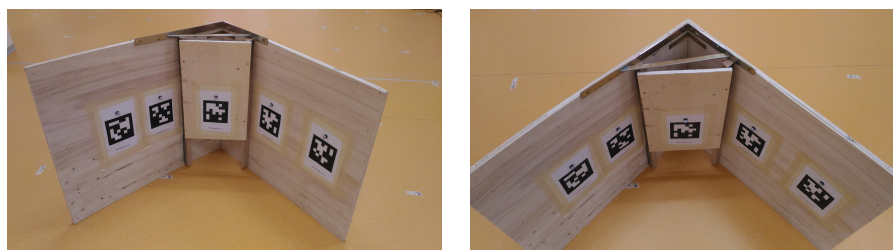
#### 5.2 Realizace doku

Pro reálné pokusy byl z dřevěných desek sestaven dok. Dvě dřevěné desky jsou spojeny v pravém úhlu jak je vidět na obrázku 5.2. Na deskách jsou svislé trámký, které pomáhají při zarovnávání mračen bodů během kalibrace. Každá deska má rozměry přibližně 1x1 metr. Dok však není určen výlučně pro účely této práce, proto jsou na něm ještě umístěny vizuální značky používané



Obrázek 5.1: Robot vybavený senzory

v dalších projektech. Na doku jsou také připevněny reflexní kuličky, které slouží k zaměření pozice doku během testování.



(a):

(b):

Obrázek 5.2: Reálná podoba doku

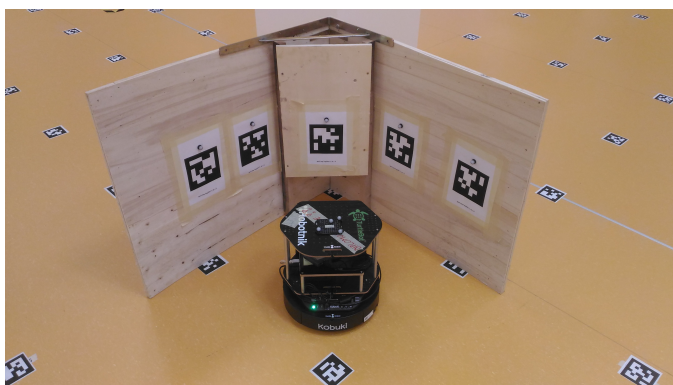
### 5.3 Kalibrace senzorů

Kalibrace je nezbytná pro fungování dokovacího algoritmu a také pro určení cílové polohy robotu. Pro ukázkou byly provedeny dvě kalibrace z různých vzdáleností, na kterých je ukázán rozdíl v přesnosti výsledné transformační matice. Dále byl pořízen referenční scan. Poloha robotu na cílové pozici je vidět na obrázku 6.2. V kapitole vyhodnocení experimentů je obrázek referenčního scanu a pro porovnání i scany z jednotlivých senzorů.

### 5.4 Dokování v ose doku

Při prvním pokusu se samotným dokovacím algoritmem byl robot umístěn přibližně na ose doku (tedy na ose úhlu mezi stěnami doku). Dokování z této



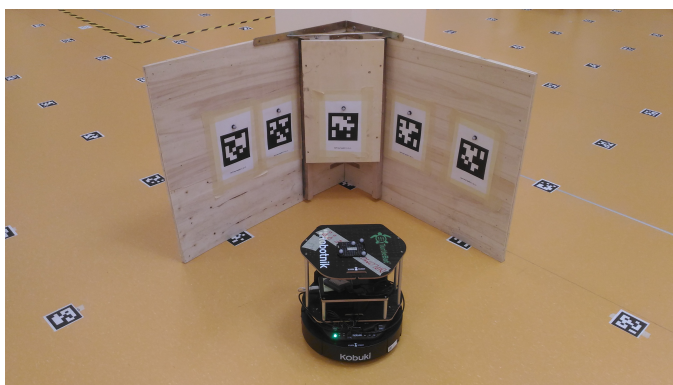


**Obrázek 5.3:** Pozice robotu během vytváření referenčního mračna bodů

výchozí pozice by mělo být nejjednodušší, protože robot má nejlepší záběr doku. Proto by odchylka od cílové pozice měla být u tohoto experimentu nejnižší. Tento pokus byl rozdělen na dvě části. V první části byla výchozí pozice robotu velmi blízko cílového bodu. Ve druhé části už byla vzdálenost větší. Díky tomu by mělo jít z výsledků určit, zda má vzdálenost od cílové pozice vliv na přesnost dokování.

#### ■ 5.4.1 Malá vzdálenost od cílového bodu

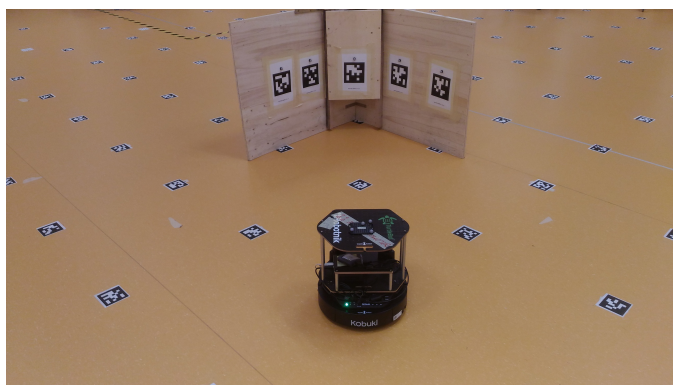
Na obrázku 5.4 je vidět výchozí pozice robotu při prvním pokusu. Vzdálenost od cílové pozice je přibližně 20 - 30 centimetrů.



**Obrázek 5.4:** Pozice robotu při dokování z malé vzdálenosti v ose doku

## 5.4.2 Velká vzdálenost od cílového bodu

Na obrázku 5.5 je zobrazena výchozí pozice robotu během druhého experimentu. V tomto případě se vzdálenost od cílové pozice pohybuje už kolem jednoho metru.



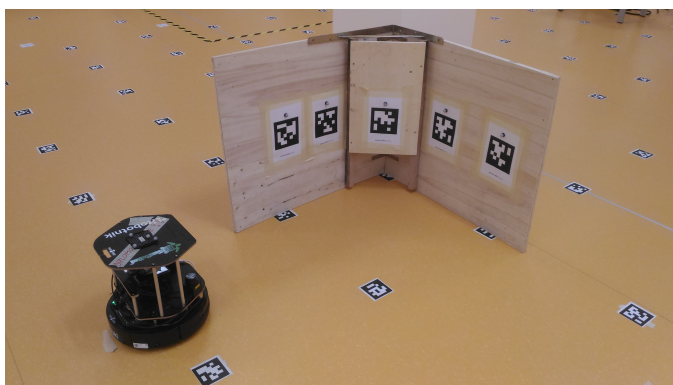
**Obrázek 5.5:** Pozice robotu při dokování z velké vzdálenosti v ose doku

## 5.5 Dokování mimo osu doku

Ve druhé části experimentů s dokovacím algoritmem byla výchozí pozice robotu posunuta mimo osu doku. Byly opět provedeny dva pokusy. Při prvním pokusu byla výchozí pozice blíže k pravé stěně doku a při druhém pokusu blíže k levé stěně. Robot byl umístěn téměř na hranici prostoru, kde je ještě schopný detekovat dok. Díky tomu je vzdálenost od středové osy relativně velká. Tyto pokusy mohou ukázat, zda má odchylka od osy doku vliv na přesnost dokování.

### 5.5.1 Dokování podél levé stěny

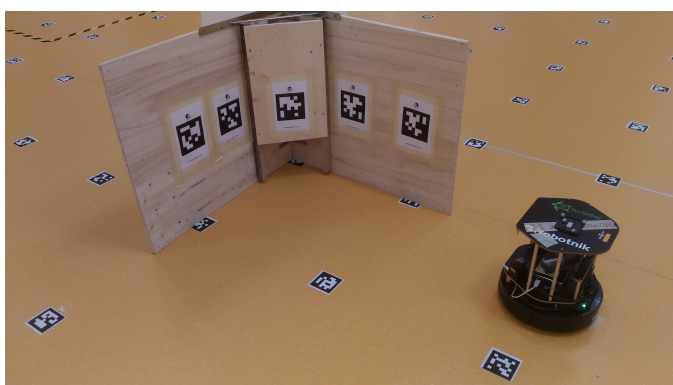
Obrázek 5.6 ukazuje výchozí pozici během třetího experimentu. Vzdálenost od cílového bodu je přibližně stejná jako při druhém experimentu (asi 1 metr), ale robot je posunut výrazně směrem k levé stěně doku.



Obrázek 5.6: Pozice robotu při dokování podél levé stěny

### 5.5.2 Dokování podél pravé stěny

Výchozí pozice robotu při posledním experimentu je zachycena na obrázku 5.7. Situace je podobná jako v předchozím případě, pouze u druhé stěny doku. Tímto by se mělo prokázat, že algoritmus není závislý na výchozím natočení.



Obrázek 5.7: Pozice robotu při dokování podél pravé stěny



## Kapitola 6

### Vyhodnocení experimentů

Tato část práce se zabývá vyhodnocením experimentů popsaných v předchozí kapitole. Na začátku je popsán použitý měřicí systém. Dále jsou ukázány obrázky z kalibrace a zarovnávání mračen bodů. Ve zbytku kapitoly jsou výsledky z provedených dokování. Celkem byly provedeny pokusy ze čtyř různých pozic, přičemž z každé pozice bylo provedeno deset měření.

#### 6.1 Vicon

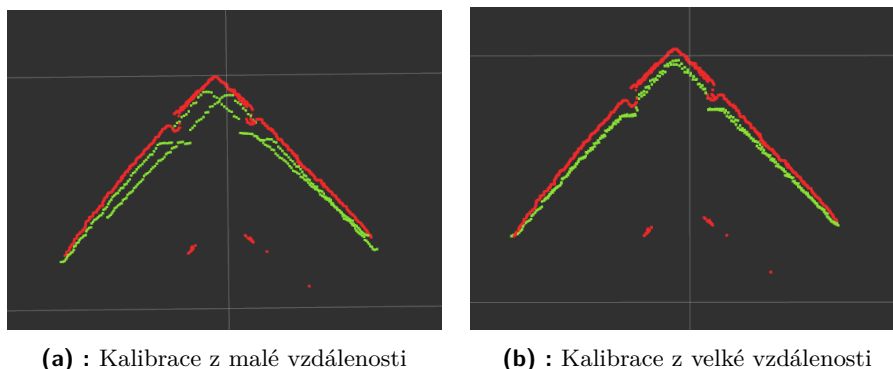
K měření během testování byl použit komerční systém Vicon [12], který je instalován v laboratoři. Tento systém se hojně používá pro takzvaný motion capture (zachycení pohybu). Systém se skládá z velkého množství infračervených kamer. Pro detekci objektů se používají malé kuličky, které se připevní na objekty, jejichž pozici je potřeba měřit. Tyto kuličky se chovají jako pasivní značky, to znamená že pouze odrážejí infračervené světlo a samy nevysílají žádný signál. Díky kombinaci měření z více kamer je tento systém schopen dosáhnout velmi vysoké přesnosti. V případě systému použitého v této práci se přesnost měření pohybuje pod  $\pm 1$  mm.

## 6.2 Kalibrace senzorů

Během experimentování byly provedeny dvě různé kalibrace senzorů. Důvodem je ukázat vliv kalibrace na přesnost skládání scanů během dokování. Byly tedy získány dvě různé transformační matice, které jsou uvedeny dále. Je vidět, že v jejich hodnotách jsou malé rozdíly. Vlevo je matice pořízená při kalibraci v blízkosti doku, vpravo matice při kalibraci z větší vzdálenosti.

$$\begin{bmatrix} -0,57 & 0,81 & 0 & -0,13 \\ -0,81 & -0,57 & 0 & -0,09 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} -0,51 & 0,86 & 0 & -0,15 \\ -0,86 & -0,51 & 0 & -0,13 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.1)$$

Na obrázku 6.1 je porovnání složených mračen bodů při použití různých transformačních matic. Snímky jsou pořízeny ze vzdálenosti přibližně jednoho metru od cílové pozice. Při malé vzdálenosti od doku není rozdíl v kalibraci téměř vůbec znatelný.

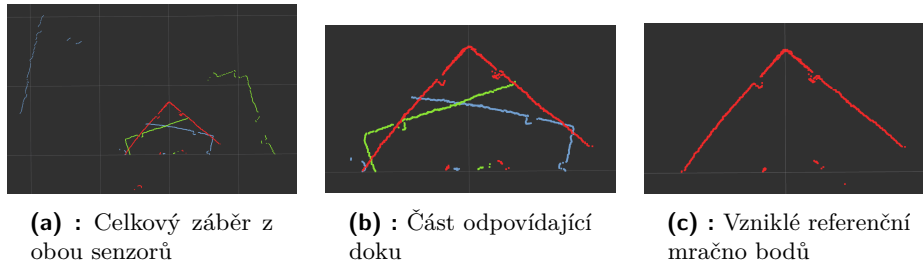


**Obrázek 6.1:** Vliv přesnosti kalibrace na skládání scanů

### 6.2.1 Vytvoření referenčního scanu

Na začátku měření bylo vytvořeno referenční mračno bodů. Na obrázku 6.2 je vidět, jak se ze dvou výchozích mračen bodů skládá referenční mračno. Data ze senzorů jsou označena modře a zeleně. Je vidět že v záběru jsou kromě doku ještě další objekty, které jsou však ve velké vzdálenosti. V tomto případě se jedná zřejmě o stěny místnosti. Výsledné referenční mračno bodů je označeno červeně a tyto vzdálené body již neobsahuje. Je rovněž vidět,

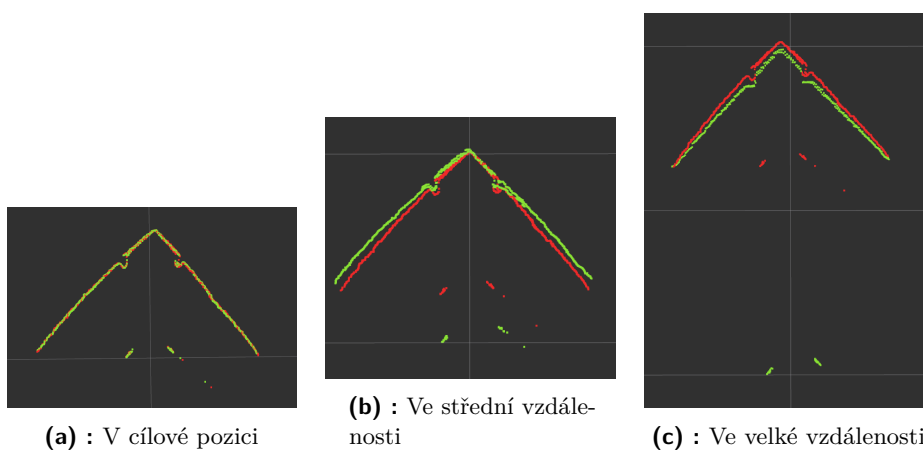
že po zarovnání dat ze senzorů odpovídajících doku, je výsledné referenční mračno bodů transformováno do pohledu z robotu. Tedy tak, že počátek souřadnic je v robotu a osa x směřuje dopředu. Dva malé shluky bodů ve středu jsou okraje robotu, které zasahují do záběru laserů.



Obrázek 6.2: Vytvoření referenčního mračna bodů

## 6.3 Dokování v ose doku

Tato část obsahuje výsledky z dokování v ose doku. Na obrázku 6.3 je ukázáno, jak vypadá zarovnání referenčního mračna bodů s aktuálními daty ze senzorů během dokování. Je zřejmé, že v cílové pozici je shoda maximální a odchylka mračen téměř nulová. S rostoucí vzdáleností od doku se rozdíl obou mračen trochu zvětšuje, ale přesto zůstává zarovnání velmi dobré a pro účely dokování je shoda obou mračen velmi dobrá.



Obrázek 6.3: Zarovnání mračen bodů s rostoucí vzdáleností

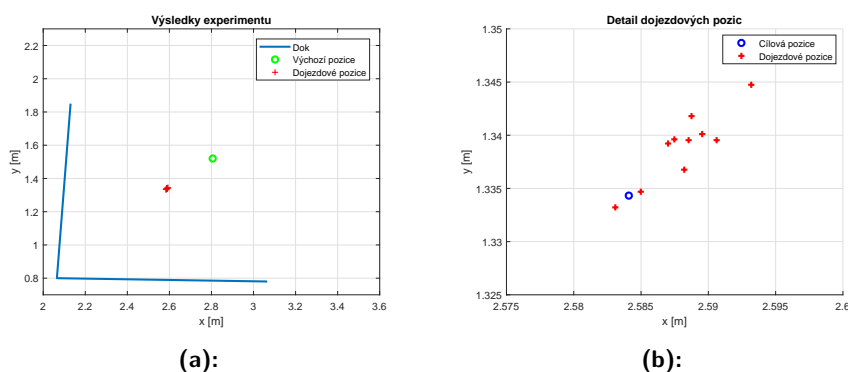
### 6.3.1 Malá vzdálenost od cílového bodu

První měření bylo prováděno z okolí cílového bodu. Na obrázku 6.4 jsou vidět dosažené výsledky. Na prvním obrázku je celková situace experimentu včetně zobrazení doku a výchozí pozice. Na druhém obrázku jsou poté přiblíženy dosažené pozice v porovnání s cílovou pozicí. Stejně jsou výsledky prezentovány i u dalších měření.

V případě tohoto experimentu byla dosažena průměrná odchylka 6,5 mm. Maximální odchylka byla 13,8 mm. V následující tabulce 6.1 jsou hodnoty odchylek jednotlivých pozic dosažených během tohoto experimentu.

Měření č.	1	2	3	4	5	6	7	8	9	10	$\bar{\sigma}$
Odch. [mm]	1,0	4,8	1,5	6,8	6,3	7,9	8,3	5,7	8,8	13,8	6,5

**Tabulka 6.1:** Odchylky při dokování z malé vzdálenosti v ose doku



**Obrázek 6.4:** Výsledky dokování z malé vzdálenosti v ose doku

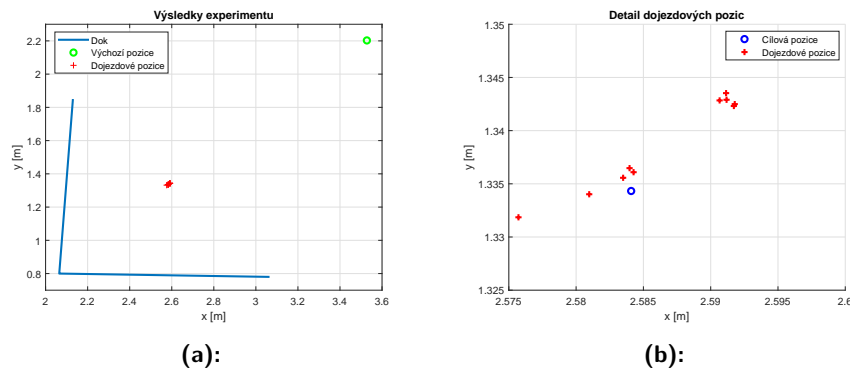
### 6.3.2 Velká vzdálenost od cílového bodu

Při dokování z velké vzdálenosti byla průměrná odchylka 7,3 mm a maximální odchylka 11,6 mm. Výsledky jsou zobrazeny na obrázku 6.5 a v tabulce 6.2.

Měření č.	1	2	3	4	5	6	7	8	9	10	$\bar{\sigma}$
Odch. [mm]	3,1	11,0	11,6	1,8	11,2	8,7	1,4	11,1	2,2	10,8	7,3

**Tabulka 6.2:** Odchylky při dokování z velké vzdálenosti v ose doku





Obrázek 6.5: Výsledky dokování z velké vzdálenosti v ose doku

## 6.4 Dokování mimo osu doku

Následující dvojice experimentů byla provedena mimo osu doku. Oproti očekávání byly dosažené výsledky v těchto případech stejné či dokonce lepší než při dokování v ose. Z toho lze usuzovat, že výchozí pozice nemá na přesnost dokování téměř žádný vliv.

### 6.4.1 Dokování podél levé stěny

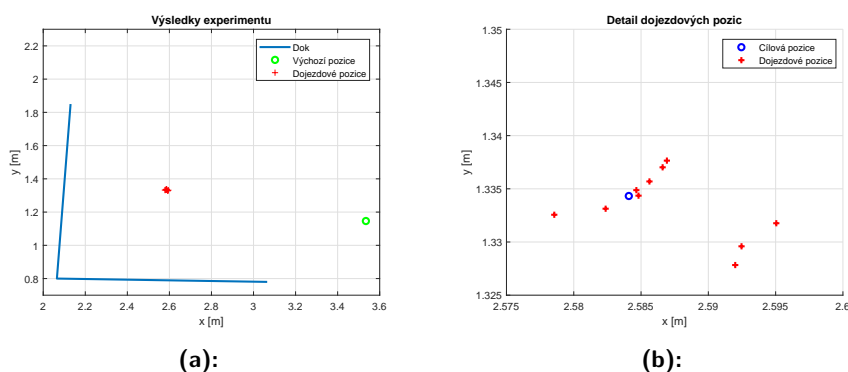
Při dokování podél levé stěny byla výsledná průměrná odchylka 5,1 mm, což je nejméně ze všech provedených experimentů. Maximální odchylka vyšla 11,3 mm. Výsledky jsou v obrázku 6.6.

Měření č.	1	2	3	4	5	6	7	8	9	10	∅
Odch. [mm]	4,4	0,7	2,1	9,6	3,7	5,8	10,2	11,3	0,8	2,1	5,1

Tabulka 6.3: Odchylky při dokování podél levé stěny

### 6.4.2 Dokování podél pravé stěny

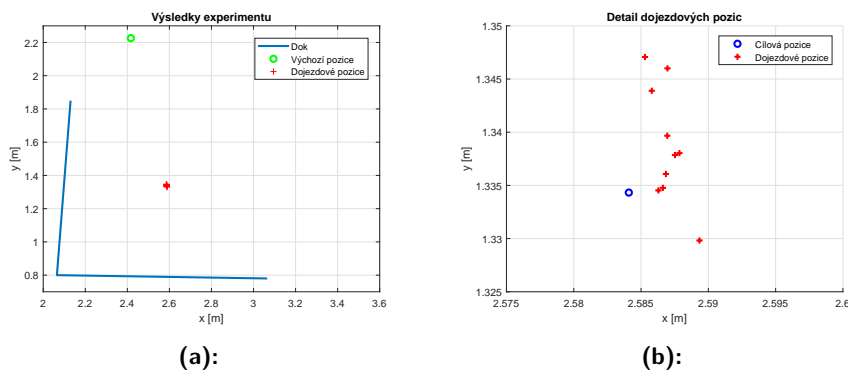
Při posledním pokusu bylo provedeno dokování podél pravé stěny. Průměrná odchylka tohoto experimentu činí 6,6 mm. Maximální naměřená odchylka byla 12,8 mm. Dosažené pozice jsou vidět na obrázku 6.7.



Obrázek 6.6: Výsledky dokování podél levé stěny

Měření č.	1	2	3	4	5	6	7	8	9	10	∅
Odch. [mm]	2,2	12,8	12,0	4,9	5,3	6,9	3,3	2,6	9,7	6,1	6,6

Tabulka 6.4: Odchytky při dokování podél pravé stěny



Obrázek 6.7: Výsledky dokování podél pravé stěny

## 6.5 Celkové vyhodnocení

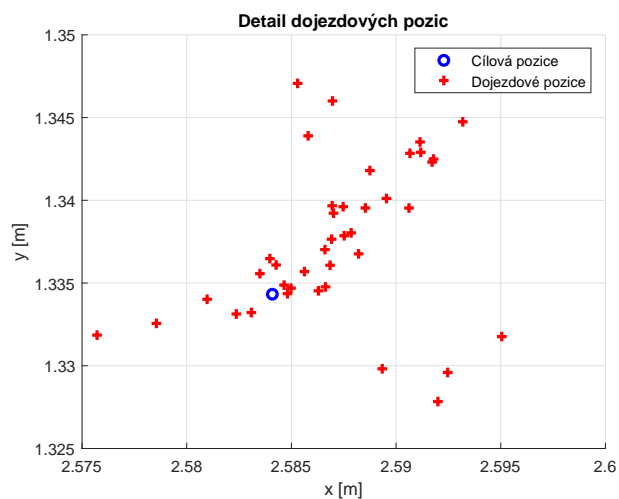
Z předchozích výsledků je vidět, že pozice dosažené z různých výchozích pozic jsou velmi podobné. To samé platí o spočítaných odchytkách, které jsou shrnuty v tabulce 6.5. Nedá se tedy říci, že by některá pozice byla pro dokování výhodnější než jiná. Na závěr byla spočítána celková průměrná odchylka ze všech provedených měření. Tato odchylka je 6,4 mm. Maximální odchylka mezi všemi měřeními byla 13,8 mm.

Experiment	Blízko	Daleko	Vlevo	Vpravo	Celkem
Průměrná odch. [mm]	6,5	7,3	5,1	6,6	6,4

Tabulka 6.5: Shrnutí průměrných odchylek

Při některých pokusech byla tedy překonána odchylka jednoho centimetru.

Průměrná odchylka se však pohybuje relativně hluboko pod touto hranicí. Výsledky dosažené testováním dokovacího algoritmu lze tedy prohlásit za více než uspokojivé. Na posledním obrázku 6.8 jsou ještě vykresleny všechny dosažené pozice během všech pokusů vzhledem k cílové pozici.



**Obrázek 6.8:** Všechny dosažené dojezdové pozice



## Kapitola 7

### Závěr

V úvodu této práce byly popsány některé základní způsoby lokalizace z laserových dat a také způsoby navigace mobilních robotů. Tyto informace sloužily jako základ pro řešení úlohy, kterou se tato práce zabývala. Touto úlohou bylo dokování mobilního robotu s pomocí laserových senzorů. Přesné zadání práce je rovněž uvedeno v úvodu. Prvotním impulsem pro vznik tohoto zadání byl požadavek na vytvoření dokovacího algoritmu pro průmyslové roboty. V této práci byl pro vývoj algoritmu použit malý mobilní robot Turtlebot 2, který se více hodí pro experimentování v laboratorních podmínkách.

Druhá kapitola se zabývá teoretickým rozborem zadaného problému. Konkrétně principem činnosti laserových senzorů a dále metodou ICP, která slouží pro práci s mračny bodů. Následuje konkrétní návrh řešení dokovacího algoritmu včetně kalibrace laserových senzorů, která je nezbytná z důvodu použití více senzorů. Navržený princip řešení využívá předem pořízeného referenčního mračna bodů, které se v reálném čase pomocí metody ICP zarovnává s aktuálními daty. Tímto způsobem se získává transformace, ze které lze spočítat souřadnice cílového bodu vzhledem k robotu.

Další kapitola prezentuje nástroje nezbytné pro vývoj a fungování algoritmu. Z nich je nejdůležitější takzvaný Robot Operating System, což je platforma vytvořená přímo pro účely práce s mobilními roboty a vývoj nových algoritmů. Další nezbytnou součástí je veřejná knihovna Point Cloud Library, která slouží pro práci s laserovými daty a mračny bodů. Zmíněny jsou i některé další balíčky z prostředí ROS.

Čtvrtá kapitola dokumentuje konkrétní implementaci algoritmu. Celkový systém dokování byl rozdělen na několik dílčích částí. Tyto části se samostatně spouštějí jako uzly ve zmíněném prostředí ROS. Relativně samostatnou částí je kalibrace senzorů, která slouží jako příprava pro samotný dokovací algoritmus. Samotné dokování je potom rozděleno na část skládání scanů ze senzorů. Následuje zarovnávání s referenčním mračnem bodů, kde se spočítají souřadnice cílového bodu. Poslední částí je řízení robotu do cílové pozice.

Poslední dvě kapitoly se zabývají návrhem a vyhodnocením experimentů. Nejprve jsou popsány podmínky provedení jednotlivých pokusů. Tyto pokusy sestávaly převážně z provedení dokování z několika různých pozic. Součástí experimentů je i ukázka kalibrace senzorů a realizace doku. V poslední kapitole je poté vyhodnocení jednotlivých experimentů. Dohromady bylo provedeno 40 měření ze 4 různých výchozích pozic. Pro každou výchozí pozici jsou vykresleny dosažené pozice a spočítány průměrné odchylky od požadované cílové pozice. Pro přesné měření polohy robotu byl během testování použit systém Vicon, který je schopen detekovat používané značky s vysokou přesností.

Výsledky dosažené během experimentů jsou velmi pozitivní. Celková průměrná odchylka dosažená během všech čtyřiceti měření byla pouhých 6,4 mm. To svědčí o vysoké přesnosti vytvořeného dokovacího algoritmu. Některé jednotlivé odchylky sice přesáhly hranici jednoho centimetru, bylo jich však pouze několik. Těto vysoké přesnosti bylo dosaženo z velké části díky použití laserových dálkoměrů, které jsou známy právě díky své přesnosti. Naopak je potřeba zmínit, že dosažená přesnost je také závislá na použitém robotu, který není schopen vykonávat nekonečně malé pohyby. Už z toho důvodu jsou dosažené výsledky částečně překvapivé a velmi příznivé.

Po zhodnocení dosažených výsledků lze říci, že vyvinutý algoritmus by bylo možné doporučit pro reálné použití. Dalším nezbytným krokem by tedy bylo otestovat algoritmus na větším průmyslovém podvozku. Jistou výhodou by mohlo být všesměrové řízení, které by zjednodušilo navigaci robotu. Pro další zlepšení a plnou funkčnost algoritmu by bylo potřeba přidat další dva laserové dálkoměry tak, aby jejich celkový záběr pokrýval celé okolí robotu. I bez toho lze však výsledky této práce považovat za úspěch.



## Příloha A

### Literatura

- [1] S. Kuutti, S. Fallah, K. Katsaros, M. Dianati, F. Mccullough and A. Mouzakitis, "A Survey of the State-of-the-Art Localization Techniques and Their Potentials for Autonomous Vehicle Applications,"in IEEE Internet of Things Journal, vol. 5, no. 2, pp. 829-846, April 2018. doi: 10.1109/JIOT.2018.2812300
- [2] Liang, Wang and Zhang, Yihuan and Wang, Jun. (2017). "Map-Based Localization Method for Autonomous Vehicles Using 3D-LIDAR \*", \*This work is supported in part by the National Natural Science Foundation of China under Grant No. 61473209.. IFAC-PapersOnLine. 50. 276-281. 10.1016/j.ifacol.2017.08.046.
- [3] Zhang, Yihuan and Liang, Wang and Wang, Jun and Dolan, John. (2017). "Real-time localization method for autonomous vehicle using 3D-LIDAR."
- [4] R. W. Wolcott and R. M. Eustice, "Visual localization within LIDAR maps for automated urban driving,"2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, 2014, pp. 176-183. doi: 10.1109/IROS.2014.6942558
- [5] A. Y. Hata and D. F. Wolf, "Feature Detection for Vehicle Localization in Urban Environments Using a Multilayer LIDAR,"in IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 2, pp. 420-429, Feb. 2016. doi: 10.1109/TITS.2015.2477817
- [6] Felipe Jiménez, Miguel Clavijo and Javier Juana, "LiDAR-based SLAM algorithm for indoor scenarios,"University Institute for Automobile Research (INSIA), Madrid, Spain, VEHICULAR 2018, ISBN: 978-1-61208-643-9





## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Brandštýl** Jméno: **František** Osobní číslo: **435024**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávající katedra/ústav: **Katedra řídicí techniky**  
Studijní program: **Kybernetika a robotika**  
Studijní obor: **Kybernetika a robotika**

## II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

**Přesné dokování mobilního robotu s využitím laserových senzorů**

Název diplomové práce anglicky:

**Precise docking of mobile robot with laser scanners**

Pokyny pro vypracování:

1. Seznamte se s metodami přesné lokalizace z laserových dat
2. Seznamte se s metodami navigace mobilního robotu (s důrazem na všesměrový podvozek)
3. Navrhněte a implementujte metodu pro přesné dokování mobilního robotu
4. Vyhodnoťte vlastnosti navržené metody

Seznam doporučené literatury:

1. On the position accuracy of mobile robot localization based on particle filters combined with scan matching. Röwekämper, J., Sprunk, C., Tipaldi, G. D., Stachniss, C., Pfaff, P., & Burgard, W. Intelligent Robots and Systems (IROS), 2012
2. A hybrid approach for robust and precise mobile robot navigation with compact environment modeling. Tomatis, N., Nourbakhsh, I., Arras, K., & Siegwart, R. Conference on Robotics and Automation, ICRA. 2001

Jméno a pracoviště vedoucí(ho) diplomové práce:

**Ing. Karel Košnar, Ph.D., inteligentní a mobilní robotika CIIRC**

Jméno a pracoviště druhého(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **14.02.2019**

Termín odevzdání diplomové práce: **24.05.2019**

Platnost zadání diplomové práce:

**do konce letního semestru 2019/2020**

\_\_\_\_\_  
Ing. Karel Košnar, Ph.D.  
podpis vedoucí(ho) práce

\_\_\_\_\_  
prof. Ing. Michael Šebek, DrSc.  
podpis vedoucí(ho) ústavu/katedry

\_\_\_\_\_  
prof. Ing. Pavel Ripka, CSc.  
podpis děkana(ky)

## III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

\_\_\_\_\_  
Datum převzetí zadání

\_\_\_\_\_  
Podpis studenta