

Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra měření

**Software pro robotickou jednotku, která
měří osvětlenost v interiérech v síti
kontrolních bodů**

Petr Kůrka

Školitel: Ing. Tomáš Drábek
Květen 2020

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Kůrka** Jméno: **Petr** Osobní číslo: **468799**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávací katedra/ústav: **Katedra měření**
Studijní program: **Kybernetika a robotika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Software pro robotickou jednotku, která měří osvětlenost v interiérech v síti kontrolních bodů

Název bakalářské práce anglicky:

A software for a robotic unit that measures indoor illumination in a control point network

Pokyny pro vypracování:

Na základě dostupných zdrojů knižních, časopiseckých a internetových se seznamte s problematikou měření osvětlenosti. Proveďte rešerši v oblasti měření osvětlenosti a lokace robota v interiéru včetně používaných senzorů. Na předložené robotické jednotce pomocí existujících knihoven implementujte algoritmus pro mapování místností a najíždění do kontrolních bodů v místnosti na základě zadaných souřadnic, kde robot bude provádět měření osvětlenosti. Proveďte reálné experimenty pro různé místnosti.

Seznam doporučené literatury:

- [1] Česká technická norma ČSN 36 0011 - 1. Měření osvětlenosti vnitřních prostorů - Část 1: Základní ustanovení. Český normalizační institut, 2005, 13 s.
- [2] Česká technická norma ČSN 36 0011 - 3. Měření osvětlenosti vnitřních prostorů - Část 1: Měření umělého osvětlení. Český normalizační institut, 2005, 11 s.
- [3] Česká technická norma ČSN EN 12 464 - 1 (360450). Světlo a osvětlení - Osvětlení pracovních prostorů - Část 1: Vnitřní pracovní prostory. Český normalizační institut, 2004, 52 s.
- [4] Drábek, T.: Robotická jednotka pro měření osvětlenosti v interiérech v rovnoměrně optimalizované síti kontrolních bodů. Diplomová práce ČVUT FEL, 2014, 54 s.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Tomáš Drábek, katedra měření FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **21.01.2020**

Termín odevzdání bakalářské práce: **22.05.2020**

Platnost zadání bakalářské práce:

do konce letního semestru 2020/2021

Ing. Tomáš Drábek
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

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

_____ Datum převzetí zadání

_____ Podpis studenta

Poděkování

Rád bych poděkoval vedoucímu bakalářské práce Ing. Tomášovi Drábkovi za odborné vedení a přínosné konzultace. Dále pak Ing. Vladimírovi Petříkovi, Ph.D. za zasvěcení do tajů ROS.

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, 15. května 2020

Abstrakt

Úkolem této bakalářské práce bylo vytvoření softwaru pro robotickou jednotku, která měří osvětlenost v interiérech v síti kontrolních bodů. V úvodu práce se nachází popis měření osvětlenosti. V další části je uvedena teoretická rešerše v oblasti vhodných senzorů pro robotickou jednotku. Následující kapitola obsahuje popis mobilní platformy a dále pak práce pokračuje teoretickou rešerší v oblasti lokalizace mobilních robotů v interiéru.

Těžištěm bakalářské práce je vytvoření softwaru pro mobilní platformu se schopností mapování, orientace a pohybu v neznámém prostředí. Nedílnou součástí práce bylo ověřování–testování algoritmů a hardwarového vybavení robotické jednotky. V předposlední kapitole se nachází souhrn všech experimentů vykonaných s mobilní platformou. V závěru práce jsou uvedeny další možnosti rozšíření a zlepšení programového vybavení.

Klíčová slova: mobilní robotická jednotka; měření osvětlenosti; lokalizace; mapování; senzory

Školitel: Ing. Tomáš Drábek
Katedra měření,
FEL ČVUT v Praze

Abstract

The task of this bachelor's thesis was to create software for a robotic unit that measures lighting in interiors in a network of control points. In the introduction, there is a description of lighting measurements. The next part presents a theoretical search in the field of suitable sensors for a robotic unit. The following chapter contains a description of the mobile platform and then the work continues with a theoretical search in the field of localization of mobile robots in the interior.

The focus of the bachelor's thesis is the creation of software for a mobile platform with the ability to map, orient and move in an unknown environment. An integral part of the work was verification–testing of algorithms and hardware of the robotic unit. The penultimate chapter contains a summary of all experiments performed with the mobile platform. At the end of the work are given other possibilities of expansion and improvement of the software.

Keywords: mobile robotic unit; lighting measurement; localization; mapping; sensors

Title translation: A software for a robotic unit that measures indoor illumination in a control point network

Obsah

1 Úvod	1	5.2.3 Feature-based přístup	28
2 Měření osvětlenosti	3	5.3 Shrnutí	29
2.1 ČSN 360011-1 – základní ustanovení [1]	3	6 Popis navrženého softwaru pro robotickou jednotku	31
2.2 ČSN 360011-3 – měření umělého osvětlení [2]	5	6.1 ROS	31
2.3 Měření osvětlenosti – shrnutí	5	6.1.1 Urg node	32
3 Používané senzory pro lokalizaci v interiéru	7	6.1.2 Cartographer	32
3.1 Přehled senzorů	8	6.1.3 Move base	32
3.1.1 Lidar	8	6.1.4 My robot node	33
3.1.2 Ultrazvukové senzory	9	6.1.5 Set goal	36
3.1.3 Infračervené senzory	10	6.1.6 Rviz	36
3.1.4 Kamera	10	6.2 Software vývojové desky	36
3.1.5 MEMS	11	6.3 Odhad parametrů	36
3.1.6 Odomerické senzory	11	6.3.1 Převod translační rychlosti	37
3.1.7 Taktilní senzory	12	6.3.2 Převod rotační rychlosti	41
3.1.8 Bluetooth	12	6.3.3 Shrnutí	46
3.1.9 Wi-Fi	13	6.4 Ladění parametrů	47
3.1.10 Magnetometr	13	6.4.1 Odometrie	47
3.1.11 RGB-D kamery	14	6.4.2 Ladění robotické jednotky jako celku	48
3.2 Shrnutí	14	6.5 Odstraňování softwarových problémů	48
4 Robotická jednotka	17	6.5.1 Příklady komunikace mezi Arduinem a MD25	49
4.1 Senzory robotické jednotky	17	6.5.2 Tutorial – Navigation Stack (Mobe base)	49
4.1.1 Lidar UTM-30LX	17	6.5.3 Komunikace vývojové desky přes I2C	49
4.1.2 Luxmetr RadioLux 111	18	6.5.4 Deadzony motorů a DWA planner	50
4.1.3 Přesnosti jednotlivých senzorů	19	7 Experimenty	51
4.2 Motory EMG30	20	7.1 Experimenty přesnosti nájezdu do kontrolních bodů	51
4.3 Výdrž robotické jednotky	20	7.1.1 První experiment	51
4.4 Úpravy robotické jednotky	20	7.1.2 Druhý experiment	54
4.4.1 Chladič lidaru	20	7.2 Zátěžový test stability	56
4.4.2 Přední pěnový nárazník	21	7.3 Série kontrolních bodů na přímce	58
4.4.3 Ochrana vývojové desky Arduino Uno před rušením	21	7.4 Závěrečné testování	59
4.4.4 Ochrana vývojové desky Arduino Uno proti poklesům napětí	22	7.5 Shrnutí	61
5 Lokalizace robotické jednotky v interiéru	25	8 Závěr	63
5.1 Lokalizace robotické jednotky se znalostí počáteční pozice	25	Přílohy	65
5.2 Lokalizace robotické jednotky s neznalostí počáteční pozice	27	8.1 Příloha A	66
5.2.1 Topological přístup	27	8.2 Příloha B	67
5.2.2 Grid-based přístup	28	8.3 Příloha C	68
		8.4 Příloha D	69

8.5 Příloha E	69
Literatura	75

Obrázky

2.1 Rozložení kontrolních bodů – ČSN 36 0011 - 1 [1]	4	7.11 Závěrečné testování – měřená místnost	60
4.1 Robotická jednotka – schéma komunikace	17	7.12 Závěrečné testování – mapování	60
4.2 Robotická jednotka – hliníková kostra	18	7.13 Závěrečné testování – nájezd do kontrolních bodů	61
4.3 Robotická jednotka – zapojení..	18	8.1 Kompletní propojení jednotlivých uzlů v ROS	66
4.4 RadioLux 111 [3]	19	8.2 Přehled Cartographeru [5]	67
4.5 Chladič lidarů	21	8.3 Přehled Move base [6]	68
4.6 Pěnový nárazník/podpěra	22		
5.1 Srovnání Cartographeru vůči jiným algoritmům – Filipenko, Maksim, Ilya Afanasyev [4]	29		
6.1 Zátěžové testy – pravý motor...	40		
6.2 Zátěžové testy – levý motor	40		
6.3 Lineární regrese závislosti počtu impulzů za [s] na rychlosti motorů	41		
6.4 Měření orientace robotické jednotky pomocí MEMS	42		
6.5 Závislost rychlosti rotace na interní rychlosti motorů	44		
6.6 Závislost rychlosti rotace na interní rychlosti motorů	45		
6.7 Lineární regrese závislosti rotační rychlosti na diferenci rychlostí motorů	46		
7.1 Měření nájezdů do kontrolního bodu	52		
7.2 Diference dojezdu do počátku – osy x,y	53		
7.3 Diference dojezdu do počátku – rotace	53		
7.4 A figure	54		
7.5 Průběh experimentu	54		
7.6 Diference dojezdu do počátku – osy x,y	55		
7.7 Diference dojezdu do počátku – rotace	56		
7.8 Zátěžový test – výsledná mapa .	57		
7.9 Zátěžový test – průběh mapování	58		
7.10 Měření na chodbě FEL ČVUT v Praze	59		

Tabulky

4.1 UTM-30LX	19
4.2 Použitelné celkové chyby luxmetru [3]	19
4.3 Maximální spotřeba jednotlivých zařízení [7] [8]	20
6.1 Pravý motor – bez zátěže	38
6.2 Levý motor – bez zátěže	38
6.3 Měření 1 - závislost úhlu natočení ve stupních (bílá) na rychlosti motorů (šedá)	42
6.4 Měření 2 - závislost úhlu natočení ve stupních (bílá) na rychlosti motorů (šedá)	43
6.5 Závislost rotační rychlosti [rad/s] (bílá) na rychlosti motorů (šedá) ..	44
6.6 Testování translace.....	47
7.1 Diference dojezdu do počátku vůči skutečnému počátečnímu bodu ...	52
7.2 Diference dojezdu do počátku vůči skutečnému počátečnímu bodu ...	55
8.1 Pravý motor – koberec	69
8.2 Pravý motor – chodba.....	70
8.3 Pravý motor – lino.....	70
8.4 Levý motor – koberec	71
8.5 Levý motor – chodba.....	72
8.6 Levý motor – lino.....	72

Kapitola 1

Úvod

V dnešní době průmyslové revoluce si roboti neustále osvojují nová pracovní nasazení, která ještě před několika lety nebyla možná. Oproti lidem jsou schopni provádět jednoduché, neustále se opakující činnosti s velkou rychlostí, a to po velmi dlouhou dobu. Díky těmto vlastnostem začínají dominovat výrobě a dalším odvětvím již od 80. let dvacátého století [9]. Roboti mohou být ideálními pomocníky, kdy lidem připadá kreativní činnost plánování a robotům se pak přenechává samotná realizace plánu. Tento způsob dělení práce je pro lidstvo v určitých případech velice výhodný (v opačných případech může vést ke ztrátě velkého množství pracovních pozic) [10]. Tento poznatek má hluboké dopady do všech odvětví, v našem případě je to hlavně měření umělého vnitřního osvětlení budov, za účelem dosažení ideálních pobytových podmínek pro jejich lidské obyvatel.

V současné době se měření osvětlenosti provádí lidmi. Jde o zdloouvavou, nákladnou činnost vyžadující speciálně školený personál [11]. Právě zde, by robotická jednotka mohla pomoci. Samotné měření osvětlenosti po vytyčení kontrolních bodů je neustále se opakující činnost, s vysokými požadavky na přesnost a rychlost provedení samotného měření. Jedná se tedy o ideální práci pro robotickou jednotku, která by byla v dané činnosti mnohem rychlejší a efektivnější než člověk. Měření osvětlenosti by pak mohla vypadat následujícím způsobem.

Role člověka:

- obsluha robotické jednotky
- rozhodování, které místnosti jsou vhodné pro změření robotickou jednotkou
- komunikace se zákazníkem požadujícím dané měření
- zajišťování informací od správce dané budovy
- udělování příkazů robotické jednotce

Role robotické jednotky:

- mapování neznámého prostředí za účelem provedení měření umělého osvětlení

- provedení samotného měření osvětlenosti

Tento druh spolupráce člověka a robotické jednotky má několik výhod:

1. dojde ke zrychlení celého procesu
2. sníží se požadavky na kvalifikaci lidí provádějících dané měření
3. lidé se nebudou muset tolik zabývat nudnou mechanickou činností samotného měření, které přenechají robotické jednotce, a sami se budou moci zabývat složitějšími místnostmi (schodiště atd.)

Člověk robotické jednotce zadá měření vhodné místnosti a mezitím se může věnovat například měřením jiných místností pro robotickou jednotku nevhodných.

Kapitola 2

Měření osvětlenosti

Denní světlo má přímý vliv na životní funkce člověka [12]. Jeho dostatek je naprosto klíčový pro vykonávání, ať už běžných, či specializovaných úkonů. Uvnitř budov nemusí být denní světlo vždy k dispozici a nebo nemusí dosahovat optimálních úrovní. V takových případech je potřeba jej doplňovat nebo nahrazovat světlem umělým. Snahou je, aby bylo za jakýchkoliv podmínek dosaženo optimálního světelného mikroklimatu. Zabývat se touto problematikou je důležitější, než by se na první pohled mohlo zdát. Osvětlení vnitřních prostor se podílí z 20% na spotřebě celosvětové elektrické energie [11]. Vhodné osvětlení pracovních prostor má dále pozitivní dopad na efektivitu práce lidí, pracujících v daném prostoru.

Problematikou osvětlení vnitřních pracovních prostorů se zabývá norma [13]. Pro nás je však důležitější norma zabývající se samotným měřením osvětlení vnitřních prostorů, a to konkrétně její první [1] a třetí [2] část. Robotická jednotka musí zajišťovat měření vyhovující normě měření osvětlení vnitřních prostorů, aby byla schopna poskytnout relevantní data.

Měření na základě normy [2] mohou být dále zkoumána normou [13]. Až poté se dá rozhodnout, zda je určitý interiér schopen poskytovat vyhovující světelné podmínky, či nikoliv. Nyní rozebereme požadavky norem [1] a [2]. Nebudeme se zabývat celým zněním, vypíšeme pouze klíčová ustanovení.

Následující dvě sekce jsou výkladem [1],[2] a v určitých pasážích, z důvodu zachování exaktnosti formulací, je použito přímo vět či pasáží z [1],[2]. Toto upozornění nahrazuje uvedení [] za každou takovou větou, které by činilo text nepřehledným.

2.1 ČSN 360011–1 – základní ustanovení [1]

Tato norma se zabývá základním ustanovením pro měření denního, umělého a sdruženého osvětlení ve vnitřních prostorech budov. Norma udává důvody, proč měřit osvětlení ve vnitřních prostorech budov. Hlavními důvody jsou:

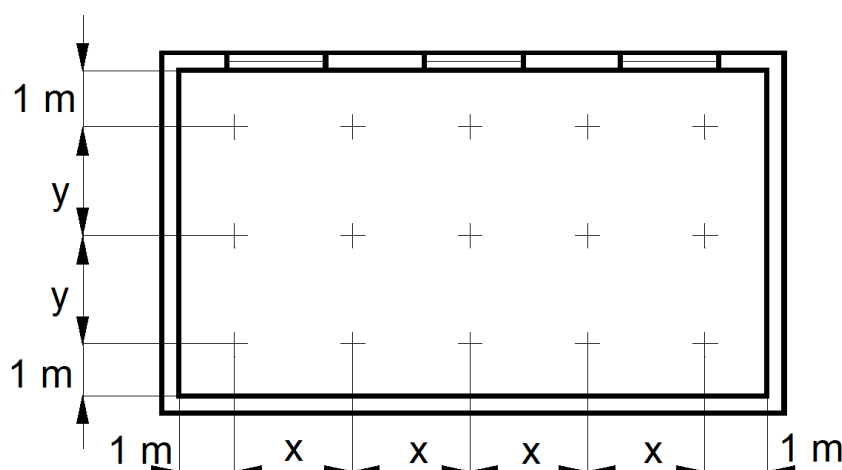
- ověření splnění podmínek a hodnot osvětlovací soustavy na základě projektu (jedná se převážně o nově zřízené budovy)
- zjištění podmínek osvětlení a zrakové pohody během užívání osvětlovacích soustav, ověření zda odpovídají normě

- porovnání různých řešení osvětlení vnitřních prostorů z hlediska zrakových podmínek, hospodárnosti a energetických úspor

Norma podle účelu měření, a z toho vyplývajících požadavků na přesnost, rozeznává tři úrovně přesnosti:

- **přesné** – určené pro posouzení náročných vnitřních prostorů atd., odhad rozšířené nejistoty měření je
 $U \leq 8(\%)$
- **provozní** – určené pro ověřování správnosti navržených a realizovaných podmínek osvětlení atd., odhad rozšířené nejistoty měření je
 $8 < U \leq 14(\%)$
- **orientační** – určené pro ověřování základních atd., odhad rozšířené nejistoty měření je
 $14 < U \leq 20(\%)$

Norma dále určuje měřené veličiny v podobě úrovně osvětlení kontrolních bodů. U umělého osvětlení a doplňujícího umělého osvětlení, se vzhledem k jejich relativní stálosti vyjadřuje jako E v luxech (lx). Kontrolní body musí být rozmístěny v pravidelné pravoúhlé síti (viz obrázek 2.1) po celé srovnávací rovině v celém vnitřním prostoru nebo v jeho funkčně vymezených částech a na pracovních místech. Výška vodorovné srovnávací roviny vnitřních prostorů je 0,85 m nad podlahou, není-li podle konkrétní funkce vnitřního prostoru stanovena výška jiná.



Obrázek 2.1: Rozložení kontrolních bodů – ČSN 36 0011 - 1 [1]

Krajní řady kontrolních bodů na vodorovné srovnávací rovině je nutné umístit 1 m od vnitřních povrchů stěn. Ostatní kontrolní body se rozmístí v pravidelných vzdálenostech s takovou hustotou, aby s ohledem na výšku vnitřního prostoru a další okolnosti byl dostatečně zachycen prostorový průběh. Pokud

některému z bodů pravidelné sítě brání k jeho změření překážka, bod se vynechá a tato okolnost se uvede ve zprávě o měření.

Při jiném než pravoúhlém půdorysném tvaru vnitřního prostoru se kontrolní body rozmístí tak, aby jejich síť byla co nejpravidelnější a co nejrovnoměrněji pokrývala celý vnitřní prostor nebo jeho funkčně vymezenou část.

Norma také určuje přístroje vhodné pro měření osvětlení – fotoelektrické luxmetry a jasoměry. Dále pak uvádí způsob zpracování naměřených dat, odhad standardní nejistoty měření, vyhodnocení měření a informace o tom, jaké údaje musí obsahovat zpráva o měření.

2.2 ČSN 360011-3 – měření umělého osvětlení [2]

Tato norma se zabývá ustanovením pro měření umělého osvětlení ve vnitřních prostorech budov. Dále určuje měřené veličiny a přístroje, kterými se dané veličiny měří. Popisuje se zde také příprava před měřením a samotný postup měření. Při měření umělého osvětlení je nutné vyloučit vliv denního osvětlení. Měření se proto provádí v noci, a nebo se zatemněnými okny. Před měřením je důležité zapnout umělé osvětlení s dostatečným předstihem, aby světelné zdroje dosáhly stabilizovaného stavu (dosáhnutí provozní teploty atd.).

Následuje postup pro vyhodnocení, odhad nejistoty měření a informace, jaké údaje má obsahovat zpráva o měření.

2.3 Měření osvětlenosti – shrnutí

Z předchozího popisu norem můžeme vyvodit základní požadavky na naši robotickou jednotku.

1. robotická jednotka musí mapovat neznámé prostředí
2. robotická jednotka se musí ve zmapovaném prostoru zorientovat, aby byla schopna dopravit měřící přístroj (luxmetr) do předem určených kontrolních bodů
3. luxmetr umístěný na robotické jednotce musí být ve výšce 0,85 m od země

Kapitola 3

Používané senzory pro lokalizaci v interiéru

V této kapitole se podíváme na běžně používané senzory, které by mohla robotická jednotka použít pro svou lokalizaci v interiéru. Nebude zde však souhrn všech možných senzorů, neboť jich je obrovské množství. Zavedeme proto následující omezující podmínky, aby jsme z velké množiny senzorů vybrali pouze vyhovující malou podmnožinu.

Hledané senzory by měly být:

- rozšířené
- ověřené praxí
- cenově přípustné
- kompaktní
- s nízkou spotřebou, při napájecím napětí 12 V
- nenáročné na obsluhu

Hledané senzory by neměly být:

- experimentální
- nepoužitelné v interiéru budov
- jakýmkoliv způsobem nevhodné pro mobilní robotickou jednotku

Při uvážení daných kritérií se nám seznam možných senzorů značně zredukuje na následující:

1. lidar
2. ultrazvukové senzory
3. infračervené senzory
4. kamera
5. MEMS
6. odomerické senzory

- úhlové rozlišení bývá velmi vysoké oproti jiným senzorům (například ultrazvukovým), konkrétně $0,25^\circ$ pro konkrétní senzor [7]
- data z lidarů mohou být přímo interpretována jako vzdálenost objektu v určitém úhlu

Nevýhody [15]:

- získáme pouze 2D sken okolního prostředí. Pro 3D sken prostředí musí být senzor namontován na pohyblivý kloub
- lidary jsou stále velmi drahé (v řádu deseti tisíců až sto tisíců Kč)
- některé průhledné nebo pohltivé materiály (například sklo) mohou být pro lidar neviditelné

■ 3.1.2 Ultrazvukové senzory

Ultrazvukové senzory (sonar) měří svoji vzdálenost vůči jiným objektům na základě odrazu ultrazvukových vln [16]. Funkce těchto senzorů je velmi podobná funkčním principům lidarů. Máme opět dva hlavní principy funkce – pulzní a fázový. Pro podrobný popis viz předchozí sekce.

Oproti lidarům mají však ultrazvukové senzory mnohem menší dosah okolo 35 ft (10,7 m) a přesnost $\pm 0,1$ ft (3 cm) [16]. Rychlost šíření zvuku je také mnohem pomalejší, než rychlost šíření světla. Musíme dbát na korekci rychlosti robotické jednotky vůči okolním objektům. Při pohybu totiž může docházet k Dopplerově jevu. Nyní shrneme výhody a nevýhody těchto senzorů.

Výhody [16]:

- nižší cena (v řádu stovek až tisíců Kč)
- dobrá dostupnost

Nevýhody [16]:

- směrová citlivost závisí na úhlu nasměrování měřeného objektu
- může docházet k několikanásobným odrazům ultrazvuku v prostředí, senzor pak podává nesprávné informace o vzdálenosti objektů
- vzhledem k relativně velkému prostorovému úhlu, jakým se ultrazvuk šíří do prostředí, je nemožné provést přesné měření malých izolovaných objektů (ze stejného důvodu je i úhlová rozlišovací schopnost nízká)
- měření pomocí ultrazvuku je zdlouhavé, neboť rychlost šíření zvuku je okolo 340 m/s (měření objektů ve vzdálenosti 6 m pak trvá okolo 35 ms [17])

- snímané prostředí musí být dostatečně osvětlené [18]
- vyšší cena v řádu tisíců (pro kamery umožňující skutečně funkční a spolehlivou lokalizaci)

■ 3.1.5 MEMS

Jedná se o velký soubor senzorů. Pro mobilní robotiku jsou zajímavé hlavně ty, jež jsou schopné měřit zrychlení. Dvojitou integrací zrychlení se dá dopočítat uražená vzdálenost robotické jednotky [19]. Se znalostí uražené vzdálenosti je pak možné provést lokalizaci.

Lokalizace pomocí MEMS (Micro Electro Mechanical Systems) se vyznačuje dobrou krátkodobou přesností. Avšak s narůstajícím časem dochází k neustálému zvětšování kumulativní chyby v lokalizaci, způsobené šumem senzorů. Přesná pozice robotické jednotky se proto musí neustále obnovovat pomocí jiných senzorů.

Výhody [19]:

- levné senzory v řádu stovek Kč
- dostatečná krátkodobá přesnost
- výborná odolnost vůči rušení, neboť senzory nepotřebují ke své činnosti přímý kontakt s okolním prostředím

Nevýhody [19]:

- s narůstajícím časem dochází k neustálému zvětšování kumulativní chyby v lokalizaci
- je potřeba použít jiné senzory k periodickému obnovování pozice robotické jednotky
- nelze určit globální počáteční pozici

■ 3.1.6 Odomerické senzory

Jedná se o velkou skupinu senzorů (například nejrůznější encodery, inkrementální čítače atd.), jež jsou schopny přímo měřit otáčení příslušných částí robotické jednotky, zaručujících pohyb v prostředí.

Vezměme například motor, u něhož budeme mít na základě odometrických senzorů znalost o úhlu natočení kol. Ze znalosti úhlu natočení a poloměru kol, jsme schopni určit vzdálenost, kterou robotická jednotka ujede a určit její pozici [20]. Jako další příklad můžeme uvést optický senzor standardní počítačové myši, jenž se dá také použít pro přímé odometrické měření ujeté vzdálenosti [21]. Tento druh lokalizace je velmi podobný předchozí kapitole s tím rozdílem, že zde je přímo měřena ujetá vzdálenost. V tomto případě chyba nezáleží na době měření senzoru, ale na délce ujeté vzdálenosti a četnosti proklouznutí kol [15]. Počáteční přesnost lokalizace je vcelku vysoká, ovšem s narůstající ujetou vzdáleností dochází k narůstání kumulativní chyby v přesnosti lokalizace robota.

Nevýhody[22]:

- nutnost umístování Bluetooth beaconů (majáků) do prostoru
- chyba v určení pozice ($0,427 \pm 0.229$) m [22]
- možné výpadky signálu

■ 3.1.9 Wi-Fi

Funkční popis principu lokalizace robota na základě Wi-Fi je popsán v kapitole Lokalizace robotické jednotky v interiéru. Výhodou těchto senzorů je opět jejich snadná dostupnost. Wi-Fi v dnešní době disponuje téměř každý notebook. Stejně jako v předchozím případě, by bylo možné využít interní Wi-Fi přijímač na notebooku naší robotické jednotky.

Výhody[23],[22]:

- nízká cena v řádu stovek Kč
- dostupnost signálu i bez přímé viditelnosti vysílače a přijímače
- jednoduchá dostupnost senzoru
- oproti Bluetooth větší dosah

Nevýhody[23]:

- oproti Bluetooth vysoká spotřeba [22]
- v moderních budovách bývá silná interference Wi-Fi signálů z mnoha vysílačů
- možné výpadky signálu

■ 3.1.10 Magnetometr

Magnetometr je přístroj schopný měřit magnetické pole. Existují dva hlavní přístupy při lokalizaci robota v interiéru, na základě aktivního a pasivního magnetického pole[24].

V případě aktivního magnetického pole, jsou do prostoru umístovány magnetické beacony (majáky), které aktivně vytvářejí požadované magnetické pole. Tímto postupem je možné dosáhnout chyby v lokalizaci robota menší než 0,25 m [24].

V případě pasivního magnetického pole, je měřeno magnetické pole, které je v prostoru již přítomno.

Magnetometr pak měří magnetické pole v daném prostoru. Na základě porovnání teoretického modelu magnetického pole s naměřenými daty, je možné provést lokalizaci robota.

Výhody[24]:

- dostupnost magnetického pole i bez přímé viditelnosti vysílače a přijímače

Robotická jednotka dále musí být schopna určit svoji počáteční globální pozici, kdy musí být zaručeno, že s narůstajícím časem a ujetou vzdáleností, nebude docházet k nárůstu chyby v určení pozice. Některé senzory však nejsou schopny svým principem funkce tyto požadavky zajistit. Musíme tedy vyloučit MEMS, odometrické a taktilní senzory. Tyto senzory mohou být dobrým doplňkem pro jiné, avšak pro samotnou lokalizaci se nehodí.

Vzhledem k tomu, že lokalizace bude prováděna uvnitř budov, je nutné, aby měl hledaný senzor dostačující dosah. Infračervené senzory mají dosah pouze do 1 m, což je pro lokalizaci ve většině běžných místností nedostačující.

Senzor by měl dále mít dostatečnou přesnost, aby bylo možné určit pozici robotické jednotky v řádu jednotek centimetrů. To bohužel neplatí pro Wi-Fi a Bluetooth, neboť přesnost lokalizace těchto senzorů je silně závislá na rušení signálu okolním prostředím. Vzhledem k tomu, že se ve většině moderních budov nachází velké množství Wi-Fi/Bluetooth zařízení, dochází k silnému rušení, výpadkům signálu. To je pro lokalizaci robotické jednotky v interiéru nepřijatelné. Nemůže totiž docházet k proměnlivé přesnosti lokalizace, na základě rušení okolním prostředím. Přesnost pozice určené pomocí Wi-Fi/Bluetooth není dobrá ani v ideálních podmínkách (viz předchozí kapitola).

Po těchto úvahách nám zbývají následující senzory:

- lidar
- ultrazvukové senzory
- kamery bez použití vizuálních značek
- RGB-D kamery

Je zřejmé, že senzor s největším dosahem a přesností je lidar [16]. Některé studie přímo naznačují, že stejné lokalizační algoritmy by měly dosáhnout vyšších přesností při použití lidarů, než při použití ultrazvukových senzorů [27]. Data z lidarů mohou být navíc přímo interpretována jako vzdálenost objektu v určitém úhlu. To samé můžeme říct o ultrazvukových senzorech. Zde však může nastat problém s vícero odrazy. Data již nejsou tak zřejmá jako u lidarů a navíc úhlové rozlišení se z principu funkce ultrazvukových senzorů nemůže lidarů zdaleka vyrovnat [16]. To samé už vůbec nelze říci o klasické kameře a RGB-D kameře, neboť vyhodnocení obrazu z těchto senzorů, je velmi výpočetně náročné [25] [15]. Pro správnou funkci kamer, je navíc potřeba zajistit správné osvětlení okolního prostředí, což je při měření osvětlenosti nepřijatelné. Při snížené viditelnosti, by bylo nutné okolní prostředí dosvítit, čímž by docházelo k přímému rušení měření osvětlenosti. RGB-D kamery mají navíc velmi malý dosah při měření vzdálenosti do 5 m a za zhoršených světelných podmínek prakticky nejsou schopny měřit [25]. Měření osvětlenosti může navíc probíhat v novostavbách, kde čerstvě natřené stěny zpravidla nemají jasnou strukturu, jedná se spíše o táhlé jednobarevné plochy. V takovém prostředí opět nejsou RGB-D kamery schopny správně fungovat, neboť potřebují jasnou strukturu okolních objektů.

Kapitola 4

Robotická jednotka

Tato robotická jednotka vychází z [3], kde nosný systém je tvořen hliníkovým rámem (viz obrázek 4.2). Na kovové kostře jsou uchyceny dva motory EMG30 [8], které jsou připojeny k řídicí jednotce motorů MD25 [28]. MD25 je potom připojena přes I2C k hlavní řídicí jednotce – Arduino Uno [29] (viz obrázek 4.1).



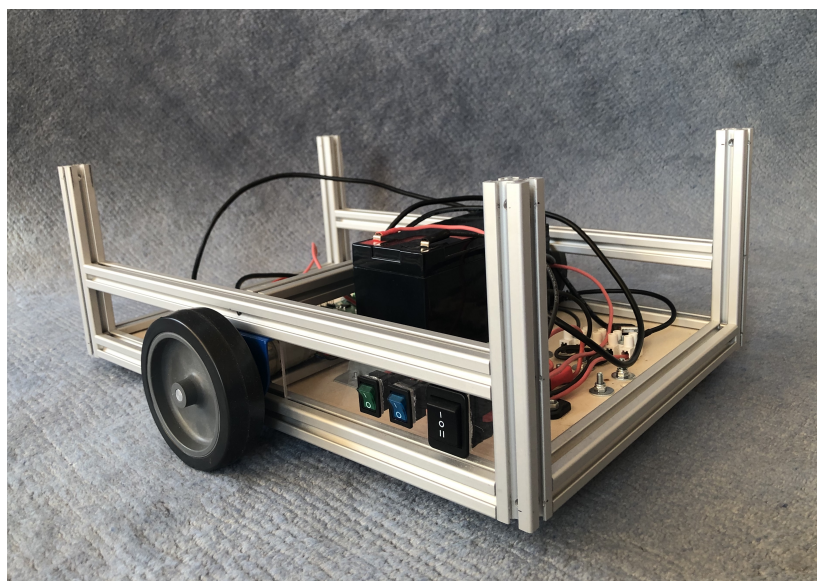
Obrázek 4.1: Robotická jednotka – schéma komunikace

Na kostře je dále uchycen 12 V akumulátor, spínače, dobíjecí konektor a pojistky (viz obrázek 4.3). PC (laptop) se vkládá na hliníkový rám nad Arduino Uno, kdy robotická jednotka si během činnosti vozí PC sebou. Vývojovou desku je možné připojit přes USB k PC, kdy PC zadává příkazy, které Arduino Uno vykonává. Pro více podrobností viz kapitola *Software robotické jednotky*.

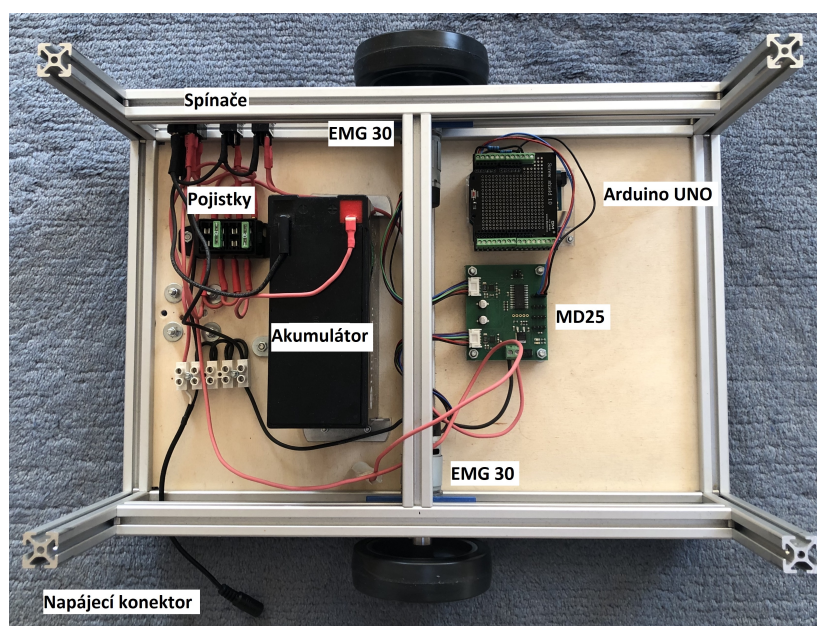
4.1 Senzory robotické jednotky

4.1.1 Lidar UTM-30LX

Robotická jednotka disponuje lidarem UTM-30LX [7]. Pro více informací viz kapitola *Používané senzory pro lokalizaci v interiéru - Lidar*. UTM-30LX měří na základě fázového principu [30], je tedy ideální pro měření v interiérech budov.



Obrázek 4.2: Robotická jednotka – hliníková kostra



Obrázek 4.3: Robotická jednotka – zapojení

4.1.2 Luxmetr RadioLux 111

Robotická jednotka měří osvětlenost luxmetrem PRC Krochmann – Radio-Lux 111, který je připojen přes sériovou linku přímo do palubního počítače. Pro více informací viz [3].



Obrázek 4.4: RadioLux 111 [3]

4.1.3 Přesnosti jednotlivých senzorů

I přesto, že součástí této práce není určování celkové přesnosti robotické jednotky, je důležité se informovat o přesnosti jednotlivých senzorů. V reálném světě není možné dosáhnout absolutně přesných senzorů a s tímto faktem je nutné počítat. Softwarová část robotické jednotky musí být dostatečně robustní na to, aby byla schopná odolat určitému množství šumu. Informace o přesnosti jednotlivých senzorů je také důležitá např. při ladění parametrů SLAM. Je vhodné upřednostňovat senzory s vyšší přesností vůči ostatním senzorům. Pro naši robotickou jednotku to konkrétně znamená, že data z lidaru (především při rotaci) mají větší váhu vůči datům z encoderů (odometrie). Následuje soupis přesností jednotlivých senzorů, na základě informací od výrobců.

Informace o lidaru UTM-30LX můžeme nalézt v tabulce 4.1.

Tabulka 4.1: UTM-30LX

Maximální dosah [m]	Přesnost (0,1 – 10) m [mm]	Úhlové rozlišení [deg]
60	±30	0,25

Luxmetr RadioLux 111 splňuje nejmenší přípustnou celkovou chybu v přesnostech měření. Celkové chyby luxmetru pro různé stupně přesnosti měření lze nalézt v tabulce 4.2 [3].

Tabulka 4.2: Použitelné celkové chyby luxmetru [3]

Přesnost měření	Přípustná celková chyba [%]
Přesné	±5
Provozní	±10
Orientační	±15

Motory EMG30 [8] obsahují encodery, které mají rozlišovací schopnost jednoho stupně.

4.2 Motory EMG30

Robotická jednotka je osazena motory EMG30, které disponují encodery (dvojice hallových sond) [8], pomocí kterých je možné určit směr otáčení spolu s momentální polohou natočení hřídele motorů. Encodery mají rozlišení 360 pulsů na jedno plné otočení hřídele kolem své osy.

4.3 Výdrž robotické jednotky

Nyní zkusíme odhadnout nejhorší možnou výdrž robotické jednotky při plném zatížení všech jednotlivých zařízení. V reálném světě bude výdrž mnohem lepší, než zde odhadovaná spodní hranice.

Tabulka 4.3: Maximální spotřeba jednotlivých zařízení [7] [8]

Zařízení	Spotřeba [A]
UTM-30LX	1
EMG30	2,5

V tabulce 4.3 jsou vypsány maximální spotřeby jednotlivých zařízení. Robotická jednotka je napájena olověným akumulátorem NPG8-12, 12 V 8 Ah, Shimastu [3]. Při maximálním možném zatížení (neustálá plná rychlost motorů, které se nemohou otáčet) bude mít robotická jednotka výdrž okolo jedné hodiny a 20 minut. Tento čas by měl být dostatečný pro provedení několika měření. Při reálném použití by výdrž akumulátoru měla být více než dostačující.

Výdrž palubního počítače HP zBook 15u G5 je přes několik hodin (i s napájením Arduina Uno), a tak by se na celkové výdrži robotické jednotky neměla vůbec projevit. Luxmetr je napájen ze samostatného zdroje.

4.4 Úpravy robotické jednotky

Při práci s robotickou jednotkou jsem provedl několik úprav.

4.4.1 Chladič lidarů

Při delším používání robotické jednotky za účelem mapování okolního prostředí, docházelo k silnému zahřívání spodního kovového podstavce lidarů.

Na základě doporučení výrobce [7] byla spodní kovová část lidarů namontována na hliníkový chladič o dostatečné velikosti, který odvádí odpadní teplo, vznikající z vysokorychlostně rotujících pohyblivých částí senzoru (viz obrázek 4.5).

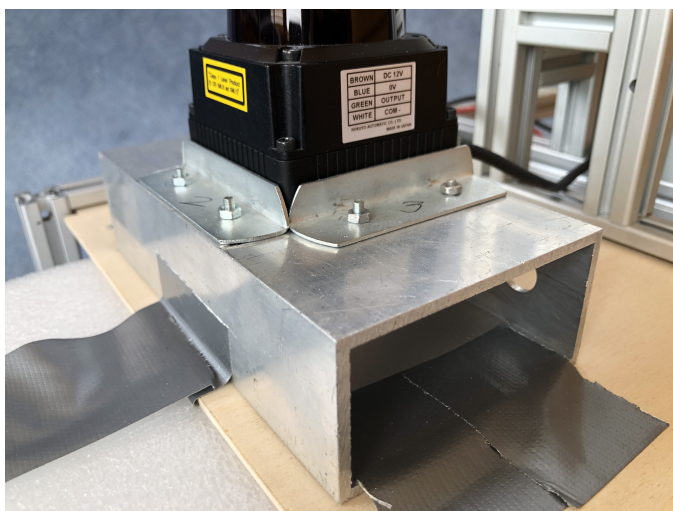
4.4.2 Přední pěnový nárazník

Po zvládnutí najíždění do kontrolních bodů, bylo možné zvýšit celkovou rychlost pohybu robotické jednotky. Pro větší bezpečnost byl do přední části upevněn pěnový nárazník (viz obrázek 4.6) plnící dvě hlavní funkce:

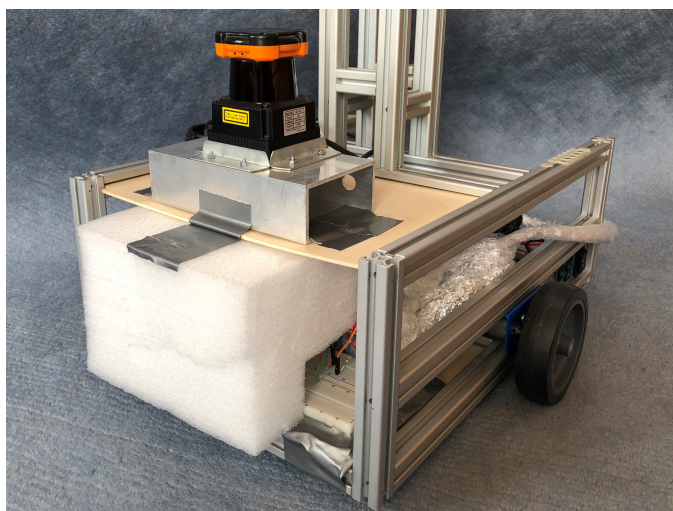
1. v případě nárazu absorbovat kinetickou energii robotické jednotky
2. po namontování hliníkového chladiče pod lidar, došlo k výraznému zatížení vrchní nosné části lidarů. – pěnový nárazník v tomto případě funguje jako potřebná podpora kompenzující váhu lidarů s chladičem

4.4.3 Ochrana vývojové desky Arduino Uno před rušením

Po skončení základních testů robotické jednotky a prvních pokusech při jejím plném zatížení (popojíždění mezi kontrolními body), se vyskytl problém se silnou nestabilitou vývojové desky. Na základě interní LED bylo zjištěno, že Arduino Uno přestane vykonávat hlavní řídicí smyčku programu a doslova zamrzne (v případě, že jsou aktivní motory EMG30 [8]). Motory při funkci vytvářejí rušení, které se projevuje v nestabilitě spojení I2C mezi Arduinem Uno [29] a MD25 [28]. Tato nestabilita měla za následek neustálé zamrzání vývojového kitu, na základě čehož se stala robotická jednotka nepoužitelnou.



Obrázek 4.5: Chladič lidarů



Obrázek 4.6: Pěnový nárazník/podpěra

Odstraňování tohoto problému bylo v domácích podmínkách (vzhledem ke karanténě kvůli koronaviru) značně zdlouhavé a frustrující. Někdy robotická jednotka byla funkční i 20 minut bez problémů, jindy docházelo k zamrznutí vývojové desky každé 2 minuty. Zdroj problému se ukázal jak v softwaru, systémových knihovnách Arduina Uno, které nejsou ochráněné proti rušení na I2C, tak v hardwaru robotické jednotky. Více informací ohledně odstraňování softwarových problémů je možné najít v sekci *Odstraňování softwarových problémů*.

Po odstranění softwarových problémů se situace stability značně zlepšila, ale přesto, ve speciálních případech (skoková změna rychlosti motorů z nuly na plnou rychlost), docházelo k zamrznutí vývojové desky. Propojení Arduina Uno a MD25 s pull-up rezistory, které na svých internetových stránkách doporučuje výrobce MD25, se ukázalo být nedostatečné [31]. Po řadě domácích experimentů se jako řešení ukázalo zapojit diodu na společnou zem v propustném směru od vývojové desky k MD25 [28], spolu s rezistory o hodnotě $220\ \Omega$ na každé z žil vedení I2C, včetně země.

Toto zapojení, spolu s odizolováním kovového notebooku od hliníkové kostry robotické jednotky a umístěním Arduina Uno do Faradayovy klece, vyřešilo problém zamrzávání vývojové desky.

■ 4.4.4 Ochrana vývojové desky Arduino Uno proti poklesům napětí

V pozdních fázích testování byl objeven zvláštní problém. Při dlouhodobé funkci robotické jednotky docházelo k občasnému silnému rozpadnutí mapy. Tento fakt nepříjemně ovlivňoval stabilitu SLAM. I přesto, že na první pohled se problém zdál být softwarového charakteru (špatné parametry SLAM), ve výsledku se jednalo o problém hardwarový.

Bylo zjištěno, že při dlouhodobé komunikaci (mezi PC a vývojovou deskou)

občas dojde k vyslání náhodné sekvence bytů ze strany Arduina Uno. To se projeví chybným čtením odometrie. Při lokalizaci pak dochází ke sporu mezi odometrií a SLAMem, což má za následek rozpad mapy. Při pokusech o zabezpečení komunikace mezi PC a vývojovou deskou (pomocí CRC) se četnost rozpadu mapy snížila, ale přesto nevymizela.

Po delším internetovém hledání bylo zjištěno, že nestandardní chování Arduina Uno může mít tyto příčiny.

1. přetečení paměti Arduina Uno způsobené uložením velkého množství textových řetězců
2. občasným poklesem napětí napájení vývojové desky

Bylo jasné, že první možnost nedávala smysl, neboť program pro Arduino Uno používal minimu textových řetězců.

Nestandardní chování vývojové desky bylo skutečně způsobeno fluktuacemi napětí. To bylo překvapivé zjištění, neboť Arduino Uno fungovalo na první pohled bez problémů. Nedochovalo k restartům ani ke změně periody blikání interní LED, signalizující vykonávání hlavní smyčky programu. Navíc řídicí jednotka motorů MD25 byla již v základu vybavena blokovacími kondenzátory, které měly fluktuacím napětí zabránit.

Po zapojení dvou keramických kondenzátorů na výstup napájení 5 V vývojové desky a společnou zem I2C (spolu s obalením Arduina Uno a všech připojených kabelů hliníkovou fólií), došlo k vyřešení problému. Silné rozpadání mapy již dále nenastávalo.

Kapitola 5

Lokalizace robotické jednotky v interiéru

Problém lokalizace v interiéru můžeme rozdělit do dvou kategorií a to na základě znalosti či neznalosti počáteční globální pozice robotické jednotky po její inicializaci (spuštění) [15].

V případě, že je známa počáteční globální pozice, je problém jednodušší. Robotická jednotka nemusí nutně znát mapu okolního prostředí, protože by ji měla být schopna jednoduše vytvořit. Při prozkoumávání okolního prostředí jsou získávány malé úseky mapy, které lze jednoduše složit do celkové mapy díky neustálé znalosti globální pozice.

V případě, že není známa počáteční globální pozice, je problém složitější. Robotická jednotka není schopna určit svou počáteční globální pozici a zároveň nezná mapu svého okolního prostředí. Jedná se o tzv. *strange loop* problém [32] (sám do sebe zacyklený problém), neboť k určení své pozice je nutné nejdříve znát mapu okolního prostředí, zároveň však k vytvoření mapy je potřeba nejdříve znát svou pozici. Jak je vidět, jedná se o mnohem komplikovanější situaci než v předchozím případě. K jejímu vyřešení je potřeba provádět určování pozice robotické jednotky simultánně s vytvářením mapy okolního prostředí.

V rámci lokalizace je ještě obtížnější variantou tzv. *kidnapped robot problem* [33], kdy již dobře lokalizovaná robotická jednotka je přenesena na jiné místo, aniž by o tom byla informována. Jedná se však o skutečné extrémní případ, který se často používá k testování schopnosti zotavení lokalizačních algoritmů z fatální chyby. Vzhledem k tomu, že naši robotickou jednotku nebude nikdo během vykonávání měření přenášet na jiná místa, se touto lokalizační variantou nebudeme dále zabývat.

Problémy lokalizace se znalostí/neznalostí počáteční pozice spolu s jejich základními funkčními principy budou popsány v následujících dvou částech.

5.1 Lokalizace robotické jednotky se znalostí počáteční pozice

V tomto případě je známa počáteční pozice robotické jednotky, díky označeným stálým bodům v prostoru, vůči kterým lze svou pozici poměřovat. Při pohybu je možné pozici orientačně monitorovat pomocí např. odomet-

Nevýhody:

1. nutnost umísťovat do prostoru umělé značky
2. potřeba školeného personálu, který je schopen umělé značky do prostoru správně umístit

5.2 Lokalizace robotické jednotky s neznalostí počáteční pozice

V tomto případě není známa počáteční pozice robotické jednotky a zároveň není k dispozici mapa okolního prostředí. Tento problém je již známý delší dobu a v literatuře se označuje termínem SLAM (Simultaneous Localization and Mapping). Existují tři hlavní způsoby řešení tohoto problému, které úzce souvisí s druhem map reprezentující okolního prostředí [15].

1. *topological* přístup
2. *grid-based* přístup
3. *feature-based* přístup

Každý z těchto přístupů má své výhody a nevýhody, které spolu se základními funkčními principy rozebereme v následujících třech částech této kapitoly.

5.2.1 Topological přístup

V tomto případě je mapa okolního prostředí reprezentována formou grafu. Jedná se o graf skládající se z uzlů propojených hranami. V extrémních případech nemusí tato struktura vůbec odpovídat reálné geometrické reprezentaci okolního prostředí [15]. Jednotlivá zmapovaná místa jsou do grafu ukládána formou uzlů. Vztahy mezi jednotlivými místy jsou reprezentovány formou hran grafu. Jinak řečeno, robotická jednotka je pouze schopna říci, v jakém místě (uzlu grafu) se nachází a jak se z místa A dostane do místa B. Přesnou geometrickou reprezentaci prostředí v tomto případě neuvažuje.

Výhodou daného přístupu je, že nedochází ke kumulování globální chyby při lokalizaci, jako u jiných přístupů, neboť v tomto přístupu je robotická jednotka vždy navigována lokálně (z místa do místa). Nemůže tak dojít například k chybné lokalizaci vůči velmi vzdálenému místu A, které z místa B, kde se momentálně robotická jednotka nachází, není přímo viditelné. Jak je vidět, daný přístup je hlavně výhodný pro mapování rozsáhlých prostředí, kdy je potřeba redukovat množství ukládaných informací na minimum.

Velkou nevýhodou daného přístupu však zůstává, že rozpoznávání jednotlivých míst (uzlů grafu) může být v reálném světě velmi náročné. Určení, zda bylo určité místo již navštíveno, může být velmi problematické. Robotická jednotka pak může například uvěřit, že místo, které již před tím navštívila a zmapovala, je nové a nezmapované. Tím dojde ke značným problémům, neboť jednomu reálnému místu je přiřazeno více míst v interní reprezentaci mapy.

5.3 Shrnutí

Výše zmiňované tři přístupy jsou jakési hlavní stavební kameny, na kterých stojí celá řada SLAM algoritmů.

Naše robotická jednotka bude používat lokalizační algoritmus na základě Grid-based přístupu. I přes svou velkou výpočetní náročnost a značnou spotřebu paměti mají SLAM algoritmy tohoto typu nejlepší vlastnosti pro přesnou lokalizaci v interiéru. To na základě následujících parametrů:

1. Vytvářejí detailní mapy s jasnou geometrickou interpretací okolního prostředí.
2. V případě interiérů budov nikdy nedojde k mapování velmi rozsáhlých prostorů. V praxi se spíš bude jednat o jednotlivé místnosti budov. Tím je zaručeno, že nedojde k problémům s nedostačující pamětí nebo nezvládnutelnou výpočetní náročností.
3. Jejich princip funkce umožňuje rozlišovat jednotlivé body v místnosti s určitou přesností. Tím je skutečně možné měřit osvětlenost v různých místech interiéru budov. Jiné lokalizační algoritmy (např. na principu topological) tuto vlastnost nemají. Tím jsou pro naši potřebu nepoužitelné.

V praxi využívá robotická jednotka lokalizační algoritmus Cartographer [5] od Google. Tento algoritmus byl vybrán na základě testů, ze kterých vyplynulo, že se v současné době jedná o nejlepší volně dostupný lokalizační algoritmus [4]. Z převzaté tabulky (obrázek 5.1) je jasně viditelné, že Cartographer v přesnosti naprosto přesahuje všechny ostatní algoritmy.

System	RMSE (m)	Mean (m)	Median (m)	Std. (m)	Min (m)	Max (m)
Cartographer	0.024	0.017	0.013	0.021	0.001	0.07
LSD SLAM	0.301	0.277	0.262	0.117	0.08	0.553
ORB SLAM (mono)	0.166	0.159	0.164	0.047	0.047	0.257
DSO	0.459	0.403	0.419	0.219	0.007	0.764
ZEDfu	0.726	0.631	0.692	0.358	0.002	1.323
RTAB map	0.163	0.138	0.110	0.085	0.004	0.349
ORB SLAM (stereo)	0.190	0.151	0.102	0.115	0.004	0.414
S-PTAM (no loop cl.)	0.338	0.268	0.244	0.206	0.001	0.768
S-PTAM (loop cl.)	0.295	0.257	0.242	0.145	0.006	1.119

Obrázek 5.1: Srovnání Cartographeru vůči jiným algoritmům – Filipenko, Maksim, Ilya Afanasyev [4]

Kapitola 6

Popis navrženého softwaru pro robotickou jednotku

Robotická jednotka sestává ze dvou hlavních hardwarových platforem, pro které byl vytvářen potřebný software.

1. počítač HP zBook 15u G5
2. vývojová deska Arduino Uno

Pro počítač s 64-bitovým procesorem intel (i7-8550u) a nainstalovaným operačním systémem Ubuntu ve verzi 18.04.4 (Linux), byl vybrán flexibilní rámec ROS (Robot Operating System) [39] jako vhodný základ pro další práci v oblasti softwaru na robotické jednotce.

Software pro vývojovou desku byl psán v oficiálním programátoru Arduino IDE [29] v programovacím jazyce C, určeném pro programování mikroprocesoru atmega328 [40], který Arduino Uno obsahuje.

6.1 ROS

Robotický operační systém (ROS) [39] funguje na principu uzlů (nodes). Komplexní funkčnost softwaru robota je rozdělena na několik jednodušších kompaktních částí. Tyto jednoduché části jsou poté přiřazeny jednotlivým uzlům, které potřebnou činnost vykonávají. Ve výsledku pak můžeme mít například uzel řešící pohyb robota, mapování, navigaci, čtení odometrie nebo jakoukoliv jinou činnost. Možnosti jsou nepřehledné. Ve výsledku se z těchto jednoduchých stavebních kamenů složí celý komplexní systém.

Jednotlivé uzly spolu komunikují pomocí tzv. topiců. Pokud například jeden uzel vytvoří mapu okolního prostředí, pak ji může sdílet ostatním uzlům pomocí topicu /map. Uzly, které mají o tuto informaci zájem, ji pak mohou odebírat pomocí subscription (odběru).

Ros má několik výhod, na základě kterých byl zvolen.

1. jedná se o open-source projekt s přístupným zdrojovým kódem zdarma
2. ROS obsahuje spoustu užitečných knihoven a nástrojů používaných v robotice

3. Jednotlivé uzly jsou na sobě nezávislé. Jeden uzel může být například napsaný v programovacím jazyce C, druhý v C++ a třetí v Pythonu – Přesto pod ROsem nedojde ke konfliktu a jednotlivé uzly spolu budou bez problému komunikovat a vykonávat činnost, kterou mají

V této sekci budou dále rozepsány jednotlivé uzly fungující v ROS, které robotická jednotka používá. U jednotlivých uzlů nebude soupis všech topiců a subscriptions, neboť jich je velké množství. Snaha bude spíše o celkový přehledný popis základních funkčních principů a vztahů mezi uzly. Pro jejich celé skutečné zapojení viz příloha A.

■ 6.1.1 Urg node

Urg node [41] je driver sloužící k převodu surových dat z lidaru UTM-30LX do světa ROS. Tento uzel vytvoří topic `/scan`, ve kterém se nachází všechny potřebné informace laserového měření vzdálenosti převedené do standardů ROS.

■ 6.1.2 Cartographer

Cartographer [5] je uzel sloužící k lokalizaci robotické jednotky a současnému mapování okolního prostředí (viz *Lokalizace robotické jednotky v interiéru – Shrnutí*). Tento uzel se skládá z velkého množství poduzlů. Cartographer je sám o sobě velmi komplexní systém, proto zde nebude do detailů rozebírán. Pro více informací viz oficiální dokumentace [5]. Základní schéma systému je možné nahlédnout v příloze B.

V našem případě jsou vstupem do systému data z lidaru spolu s informacemi z odometrie. Výstupem je pak vytvářená mapa na topicu `/map` spolu s informacemi o lokaci robotické jednotky v této mapě.

Cartographer má velké množství nastavení, které se provádí přepisováním příslušných souborů. Pro více informací viz sekce *ladění parametrů* této kapitoly.

Součástí uzlu Cartographer jsou v našem případě také tzv. tf. Jedná se o transformace souřadnicových systémů vůči námi zvolenému souřadnicovému systému. Transformace obsahují informace o pozici kol, lidaru atd.. Tyto informace jsou velmi potřebné, neboť určují fyzickou stavbu robotické jednotky. Topic `/tf` proto využívá celá řada dalších uzlů.

■ 6.1.3 Move base

Uzel Move base [6] slouží k navigaci robotické jednotky. Jeho hlavním vstupem je požadovaný cíl. Skládá se z lokálního a globálního plánovače (viz příloha C).

- globální plánovač navrhuje trajektorii mezi aktuální pozicí a požadovaným cílem

- lokální plánovač vytváří informace o tom, jak by se robotická jednotka měla pohybovat – jeho úkolem je plnit lokální cíle z celkové globální trajektorie

Existuje několik různých druhů lokálního plánovače. Po pár marných pokusech, o zprovoznění jedné z implementací, se až DWA planner [6] ukázal jako skutečně funkční. DWA planner bohužel neumožňuje nastavit minimální rychlost otáčení robotické jednotky. Tím vznikl problém, který se naštěstí podařilo softwarově odstranit. Pro více informací viz sekce *Odstraňování softwarových problémů* této kapitoly.

Dalšími vstupy tohoto uzlu jsou mapa, data z lidarů, transformace t_f a informace z odometrie. Jeho výstupem jsou informace o požadované translační a rotační rychlosti. Řízení robotické jednotky je řešeno zpětnovazebně, kdy požadovaný pohyb je neustále kontrolován pomocí informací z odometrie. Odometrie také slouží pro předpovídání dalšího pohybu robota.

Move base má opět velké množství nastavení, které se provádí přepisováním příslušných souborů. Pro více informací viz sekce *ladění parametrů* této kapitoly.

■ 6.1.4 My robot node

My robot node je klíčový uzel napsaný Pythonu přímo pro naši robotickou jednotku. Jeho hlavním cílem je propojit neznámý hardware robotické jednotky se standardizovaným světem ROS. Je tvořen dvěma vlákny.

Hlavní vlákno má na starosti:

- komunikaci s vývojovou deskou
- výpočet odometrie pro celý ROS systém
- přepočítání požadované rychlosti robotické jednotky do interní rychlosti motorů
- hlídání deadzone motorů
- výpis informací o funkci systému pro uživatele (terminál)
- ukončování programu

Vedlejší vlákno má na starosti:

- příjem informací o požadované rychlosti na topicu `/cmd_vel` (Move base)
- předávání informací bezpečně hlavnímu vláknu

V následujících částech této sekce rozebereme jednotlivé funkce My robot node.

■ Komunikace mezi PC a vývojovou deskou

Palubní počítač je propojen s vývojovou deskou pomocí USB (sériová komunikace o rychlosti 9600 Bd). Předávání informací probíhá na základě posílání jednoduchých ASCII znaků. Byl vytvořen následující komunikační protokol:

- (*) – bootování Arduina Uno dokončeno
- (o) – pozdrav (kontrola správné funkce)
- (v) – napětí na akumulátoru
- (c) – hodnota encoderů
- (s) – okamžité zastavení obou motorů
- (l) – volba levého motoru
- (r) – volba pravého motoru
- (m) – požadovaná rychlost motoru
- (!) – hlášení chyb
- (0123456789) – předávané číselné hodnoty

Hlavní řídicí smyčka My robot node neustále komunikuje s vývojovou deskou, za účelem dosažení potřebného chování robotické jednotky. Pro více informací o různých způsobech komunikace viz následující sekce.

■ Ovládání rychlosti motorů

My robot node každých 25 ms zasílá informaci o požadované rychlosti motorů vývojové desce. Komunikace probíhá následujícím způsobem.

1. (m) – číselná hodnota požadované rychlosti motoru
2. (l) – výběr levého motoru
3. (m) – číselná hodnota požadované rychlosti motoru
4. (r) – výběr pravého motoru

Nové požadované rychlosti jsou nejdříve ukládány do dočasných proměnných. Výběr pravého motoru (r) slouží zároveň jako synchronizační mechanismus. Až po jeho obdržení, dojde k současné změně rychlostí u obou motorů najednou.

■ Čtení odometrie

Čtení odometrie bylo nejdříve zajišťované vývojovou deskou, která každých 100 ms zaslala příslušné informace My robot node. Při reálném používání robotické jednotky bylo zjištěno, že občas dochází k současnému vyslání požadavků na změnu rychlostí motorů (ze strany My robot node), spolu s informacemi o odometrii (ze strany vývojové desky). Tento kolapsový stav protichůdného vysílání měl za následek vyřazení vývojové desky z provozu (pro více informací o zamrzání Arduina Uno viz *Odstraňování softwarových problémů*).

Současná verze programu funguje automaticky spolu s ovládáním rychlosti motorů. Po obdržení výběru pravého motoru (r), vývojová deska okamžitě odpovídá informacemi o odometrii. Tím je dosaženo bezkonfliktního vysílání ve stylu pinpongového míčku. Vysílání požadované rychlosti motorů má periodu 25 ms a trvá v nejhorším případě 0,625 ms. Vysílání odometrických dat má tak časové okno okolo 24 ms, což je naprosto dostačující i s rezervou. Tím je zaručeno, že nebude docházet k žádným problémům při vysílání.

Po obdržení odometrických informací z encoderů dojde k výpočtu nové pozice robotické jednotky. Pro přesné informace o výpočtu a lepší přehlednost viz příloha D. My robot node také obsahuje ochranu proti přetečení encoderů, spolu s měřením času mezi jednotlivými měřeními.

■ Komunikace s uživatelem

My robot node je terminálový program. Komunikace s uživatelem je proto zajišťována výpisem příslušných zpráv na terminál. Pro přehlednost jsou zprávy z My robot node a vývojové desky odděleny následujícím způsobem:

- Arduino: – tato hlavička označuje všechny zprávy z Arduina Uno
- PC: – tato hlavička označuje všechny zprávy z My robot node

■ Další funkce

My robot node v případě potřeby umožňuje čtení napětí na akumulátoru robotické jednotky nebo reset encoderů.

■ Ochrana vývojové desky při bootování

Bylo zjištěno, že v čase bootování vývojové desky, nesmějí být přes sériovou linku posílány jakékoliv příkazy na změnu rychlosti motorů. Nedodržení této podmínky má za následek nestandardní chování Arduina Uno, které může vyvolat mimovolnou aktivaci motorů. Pro zvýšení bezpečnosti startu byl vytvořen následující postup:

1. po připojení vývojové desky přes USB je vyvolán reset Arduina Uno
2. počítačový program poté čeká na symbol (*) – bootování Arduina Uno dokončeno – do této doby nevysílá jakékoliv znaky přes sériovou linku

3. po obdržení (*) PC program zasílá pozdrav (o)
4. vývojová deska posílá zpětný pozdrav
5. po obdržení zpětného pozdravu My robot node zahajuje běžnou činnost
6. v případě jakéhokoliv problému, je obsluha vyzvána k restartu Arduina Uno a proces se opakuje

6.1.5 Set goal

Set goal je uzel psaný v Pythonu, sloužící pro zadávání cílů Move base. V místě inicializace robotické jednotky je vytvořen počáteční bod $x,y = [0,0]$. Požadovaný cíl se pak určuje v metrech, vůči počátečnímu bodu. Zadávají se souřadnice x,y a rotace podle osy z v cíli. Tento program vychází z volně dostupného internetového tutoriálu [42].

6.1.6 Rviz

Rviz [43] je grafická aplikace – uzel sloužící k vizualizaci robotů a jejich činnosti. V našem případě je vizualizována mapa, robotická jednotka, tf transformace, naplánovaná trajektorie a překážky. V Rviz je také možné graficky zadávat nové cíle robotické jednotce.

6.2 Software vývojové desky

Software vývojové desky slouží jako převodník mezi PC a řídicí jednotkou motorů. Výhoda tohoto zapojení je v tom, že Arduino Uno obstarává kompletní komunikaci s MD25. Tím nedochází ke zdržování My robot node, který vždy zašle pouze příslušný příkaz, a pak se může věnovat jiným věcem. V podstatě se jedná o jakési další samostatné vlákno PC aplikace.

Software je tvořený hlavní nekonečnou smyčkou, která obstarává komunikaci, převod odometrických dat do vhodného tvaru a případné výpočty.

Průběh komunikace byl již popsán v předchozím textu této kapitoly, viz *My robot node*.

6.3 Odhad parametrů

Uzel Move base řídí robotickou jednotku s diferenciálním pohonem ve 2D prostoru pomocí těchto parametrů:

- translační (dopřednou) rychlost v [m/s]
- rotační rychlost ω [rad/s]

Řídicí jednotka motorů MD25 [28] však nativně toto řízení nepodporuje. Je schopna pouze základního přiřazování rychlostí jednotlivým motorům v rozsahu od -128 do 127. Tato interní reprezentace rychlosti nemá přímou

spojitost s obecným fyzikálním popisem, který používá Move base. Skutečná rychlost a rotační rychlost robotické jednotky dále závisí na obvodu kol a jejich vzájemné vzdálenosti vůči středu hliníkové konstrukce. V rámci uzlu robot node tak bylo nutné vytvořit převodník, jež převádí obecnou translační rychlost [m/s] a rotační rychlost [rad/s] do interní reprezentace rychlostí robotické jednotky.

■ 6.3.1 Převod translační rychlosti

Vzhledem k tomu, že technická dokumentace motorů [8] popisuje pouze minimální a maximální počet otáček za minutu v nezatíženém stavu, nebylo možné teoreticky odhadnout, jakou rychlostí v [m/s] se robotická jednotka začne pohybovat translačním pohybem při zátěži, pokud budou oba motory operovat na stejné rychlosti interní reprezentace MD25 [28]. Parametry pro převody proto musely být odhadnuty experimentálně.

■ Měření

Parametry převodu byly odhadovány na základě měření o následujícím průběhu. Současně byly spuštěny oba motory EMG30 po dobu 5 sekund v rozsahu rychlostí interní reprezentace od -50 do 50 s krokem 5. Výstupem měření bylo počet pulzů encoderů jednotlivých motorů za sekundu, pomocí kterých, spolu se znalostí poloměru kol, bylo možné určit rychlost robotické jednotky v [m/s]. Pro validitu bylo každé z jednotlivých měření provedeno třikrát, a to ve čtyřech zátěžových konfiguracích:

- s motory bez zátěže
- koberec
- lino
- dlaždice

Celkově tak bylo uskutečněno okolo 250 měření.

■ Výsledky měření

V následujících tabulkách 6.1, 6.2 jsou vyneseny výsledky měření levého a pravého motoru bez zátěže. Označení [IMP/5s] značí počet naměřených impulzů encoderů za dobu pěti sekund. Ostatní měření je možné dohledat v příloze E.

Tabulka 6.1: Pravý motor – bez zátěže

Interní rychlost [–]	Měření I [IMP/5s]	Měření II [IMP/5s]	Měření III [IMP/5s]
-50	2351	2350	2342
-45	2092	2098	2100
-40	1844	1842	1850
-35	1606	1612	1610
-30	1349	1353	1350
-25	1188	1188	1184
-20	948	946	950
-15	700	700	700
-10	448	448	448
-5	216	212	216
0	0	0	0
5	-214	-216	-215
10	-448	-448	-447
15	-700	-700	-698
20	-950	-950	-950
25	-1188	-1187	-1189
30	-1354	-1352	-1351
35	-1606	-1611	-1605
40	-1848	-1850	-1849
45	-2100	-2092	-2100
50	-2341	-2351	-2349

Tabulka 6.2: Levý motor – bez zátěže

Interní rychlost [–]	Měření I [IMP/5s]	Měření II [IMP/5s]	Měření III [IMP/5s]
-50	2350	2350	2347
-45	2092	2098	2099
-40	1842	1843	1852
-35	1605	1613	1612
-30	1358	1359	1349
-25	1186	1186	1184
-20	944	946	950
-15	700	700	700
-10	443	445	441
-5	224	223	222
0	0	0	0
5	-221	-220	-222
10	-446	-446	-446
15	-700	-700	-697
20	-951	-950	-950

25	-1188	-1188	-1189
30	-1351	-1352	-1354
35	-1605	-1611	-1605
40	-1848	-1848	-1849
45	-2100	-2092	-2100
50	-2341	-2350	-2350

Jak je vidět, výsledky měření u obou motorů jsou téměř totožné (viz tabulka 6.1, 6.2), což je nutný předpoklad pro správnou funkci robotické jednotky.

■ Výsledný odhad parametrů

Data měření byly zprůměrovány a následně vyděleny pěti, aby jsme získali počet impulzů za sekundu u jednotlivých zátěžových testů. Výsledky jsou vyneseny do následujících grafů (viz obrázek 6.1, 6.2).

Jak je vidět, jedná se o lineární funkci, kdy záporné rychlosti v interní reprezentaci motorů mají za následek kladný počet pulzů a kladné rychlosti v interní reprezentaci motorů mají za následek záporný počet pulzů. Funkce je tedy klesající (tento poznatek je důležitý v rovnici 6.2). Dále si můžeme všimnout, že na druhu zátěže záleží velmi málo. Regulátor otáček motorů v MD25 [28] funguje dobře. Levý a pravý motor se od sebe příliš neliší, což se dalo očekávat.

Tuto funkci můžeme nafitovat pomocí lineární regrese (viz obrázek 6.3). Z naměřených dat nebudeme uvažovat experimenty bez zátěže motorů, neboť pro reálné využití robotické jednotky jsou tato data irelevantní. Zbylá data z levého a pravého motoru zprůměrujeme a v Matlabu dopočteme parametry lineární regrese

$$f(x) = -9,326 \cdot x + 0,275 \quad (6.1)$$

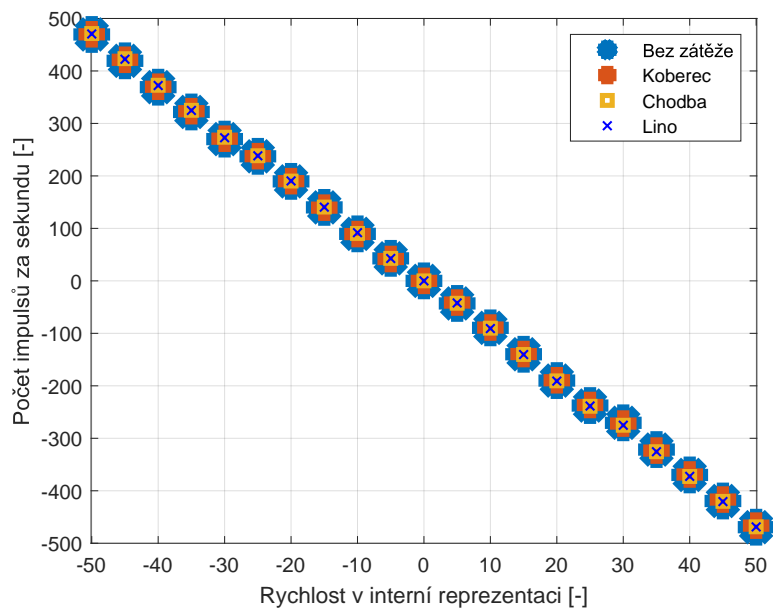
, kde f značí funkční závislost impulzů encoderů za sekundu na interní rychlosti x . Pro převod impulzů za sekundu na skutečnou translační rychlost robotické jednotky v [m/s] platí následující vztah

$$v(f) = -\frac{2\pi r f}{T_f} \quad (6.2)$$

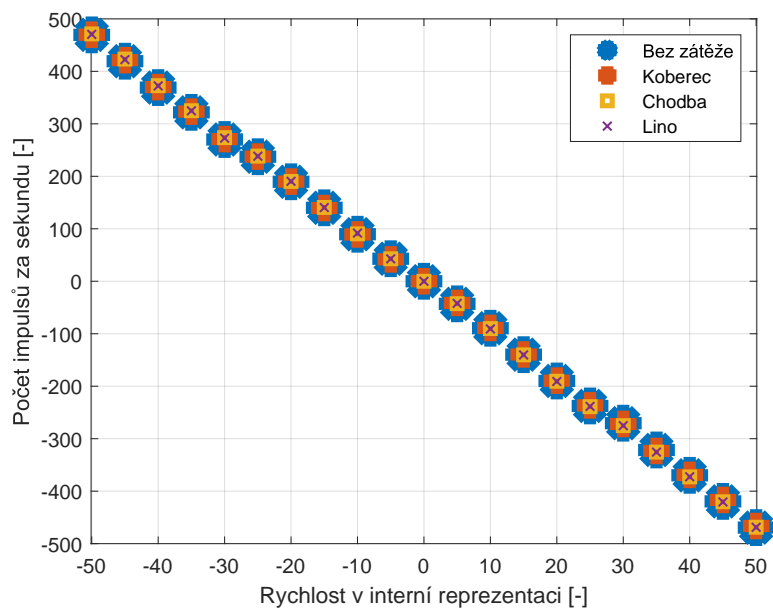
, kde v je rychlost, r je poloměr kol ($r = 4,88$ cm) robotické jednotky, f je naměřený počet impulzů a T_f ($T_f = 360$) je počet impulzů encoderů na jedno plné otočení kola. V rovnici (6.2) musí být mínus, neboť přímá data z encoderů jsou s opačným znaménkem, oproti dopředné orientaci robota. Dosazením rovnice (6.1) do (6.2) a následnou úpravou, získáme výsledný vztah pro převedení skutečné rychlosti v [m/s] do interní rychlosti robotické jednotky

$$x(v) = \frac{vT_f}{(2 \cdot 9,326)\pi r} - \frac{0,275}{9,326} \quad (6.3)$$

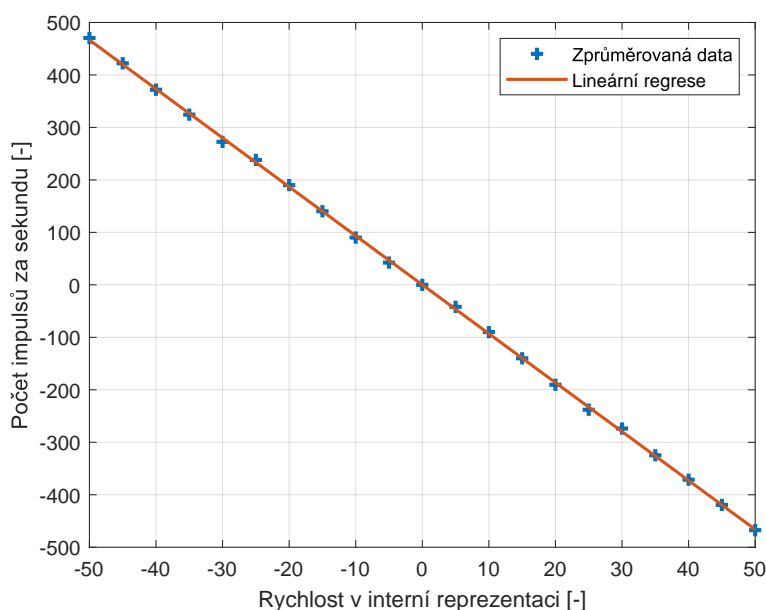
, kde x je hledaná interní rychlost motorů a v je zadaná rychlost v [m/s]. Při reálném použití bude nutné výsledky z rovnice (6.3) zaokrouhlovat na celá čísla.



Obrázek 6.1: Zátěžové testy – pravý motor



Obrázek 6.2: Zátěžové testy – levý motor



Obrázek 6.3: Lineární regrese závislosti počtu impulsů za [s] na rychlosti motorů

6.3.2 Převod rotační rychlosti

Požadovanou rotační rychlost robota v [rad/s] je nutné převést do interní reprezentace rychlostí robotické jednotky. Rotační rychlost robota je závislá na vzájemné vzdálenosti kol d , jejich poloměru r a rozdílu rychlostí levého a pravého motoru.

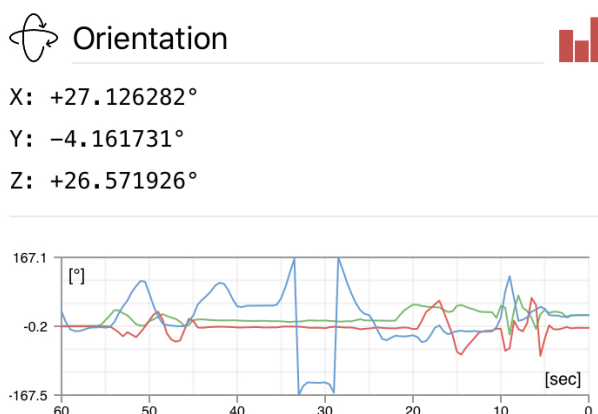
Měření

Vzhledem k tomu, že rychlost rotace robotické jednotky ovlivňuje spoustu těžko odhadnutelných parametrů (např. prokluzování kol ovlivňované vzájemnou rychlostí motorů), bylo nutné závislost rychlosti rotace robotické jednotky v závislosti na rychlostech motorů odhadovat experimentálně.

Rychlost rotace je závislá na interní rychlosti jak levého, tak pravého motoru, výsledná závislost proto musí být trojdimenzionální funkce, kdy na osách x a y je interní rychlost obou motorů a na ose z je rychlost rotace robotické jednotky. Za účelem odhadnutí této závislosti byl uskutečněn experiment o následujících parametrech.

- orientace robotické jednotky byla měřena pomocí MEMS [44] senzoru smartphonu s vysokou přesností podporující plně AR – rozšířenou realitu (viz obrázek 6.4)
- oba motory byly spouštěny v rozsahu rychlostí od -30 do 30 s krokem po pěti – byla vyzkoušena veškerá možná vzájemná působení levého a pravého motoru

- každé z měření bylo opakováno dvakrát pro větší přesnost, celkově tak bylo dosaženo 338 měření
- v každém z měření byl měřen úhel natočení za dobu 4 vteřin



Obrázek 6.4: Měření orientace robotické jednotky pomocí MEMS

■ Výsledky měření

Všechna měření byla prováděna po dobu 4 sekund, kdy horizontální šedý pruh je rozsah rychlostí pro levý motor a vertikální šedý pruh je rozsah rychlostí pro pravý motor. V tabulkách 6.3 a 6.4 je vynesena již předem zmiňovaná závislost natočení robota na rychlosti motorů interní reprezentaci.

Tabulka 6.3: Měření 1 - závislost úhlu natočení ve stupních (bílá) na rychlosti motorů (šedá)

/	-30	-25	-20	-15	-10	-5	0	5	10	15	20	25	30
-30	1	-19	-46	-74	-102	-129	-152	-175	-202	-231	-258	-285	-305
-25	19	0	-26	-54	-83	-108	-134	-156	-185	-211	-239	-266	-285
-20	46	26	0	-29	-56	-82	-105	-130	-157	-185	-213	-238	-259
-15	75	55	28	0	-28	-54	-79	-102	-128	-156	-183	-211	-228
-10	104	83	55	28	1	-26	-51	-73	-100	-128	-157	-182	-203
-5	129	109	83	55	26	0	-24	-46	-73	-101	-130	-157	-175
0	154	135	106	79	50	24	0	-24	-50	-78	-107	-132	-153
5	176	158	129	101	74	46	22	0	-27	-54	-82	-109	-128
10	202	182	157	129	100	75	51	27	0	-27	-55	-81	-102
15	229	211	184	155	128	101	77	54	28	0	-27	-55	-74
20	256	236	212	184	155	129	105	82	55	29	1	-25	-45
25	283	266	238	211	182	155	132	109	82	55	26	0	-19
30	303	285	257	229	202	175	152	129	103	75	46	20	0

Tabulka 6.4: Měření 2 - závislost úhlu natočení ve stupních (bílá) na rychlosti motorů (šedá)

/	-30	-25	-20	-15	-10	-5	0	5	10	15	20	25	30
-30	0	-19	-46	-74	-101	-128	-153	-176	-203	-230	-258	-286	-304
-25	19	0	-27	-55	-83	-109	-132	-157	-183	-210	-240	-265	-285
-20	47	26	0	-28	-56	-83	-107	-129	-156	-183	-212	-239	-258
-15	75	55	28	0	-29	-54	-79	-100	-128	-156	-184	-209	-232
-10	102	82	55	28	0	-26	-50	-74	-101	-128	-155	-182	-201
-5	129	110	83	55	27	0	-23	-46	-73	-102	-129	-156	-175
0	152	134	106	78	51	23	0	-23	-50	-78	-106	-133	-152
5	176	158	129	101	74	45	24	0	-26	-56	-83	-109	-127
10	202	182	157	129	100	74	51	26	0	-28	-56	-82	-102
15	229	211	185	157	128	100	78	54	28	0	-28	-54	-74
20	256	238	212	183	155	129	106	83	55	27	0	-26	-46
25	283	264	238	210	180	157	132	109	83	53	26	0	-20
30	304	285	256	230	202	176	153	128	103	73	47	21	1

Jak můžeme vidět v tabulkách 6.3 a 6.4, diagonála (z levého horního rohu do spodního pravého rohu) je vždy přibližně rovna 0. Na diagonále jsou experimenty, kdy oba motory byly spuštěny se stejnou rychlostí (ať již kladnou nebo zápornou), právě proto musí být diagonální hodnoty nulové. Měření dále dodržuje konvenci pravotočivé soustavy, tedy směr otáčení proti směru hodinových ručiček je, brán jako kladný. Tento fakt je hezky vidět v levém spodním rohu těchto tabulek, kdy pravý motor má interní rychlost 30 a levý motor má interní rychlost -30. Jak můžeme vidět, robot v tomto případě začne rotovat proti směru hodinových ručiček, což odpovídá základní intuici. Na tabulkách je dále zajímavé, že pruhy hodnot měření rovnoběžné s hlavní diagonálou (z levého horního rohu do spodního pravého rohu) mají vždy přibližně podobné hodnoty. Tento fakt bude více rozveden v následující části této kapitoly.

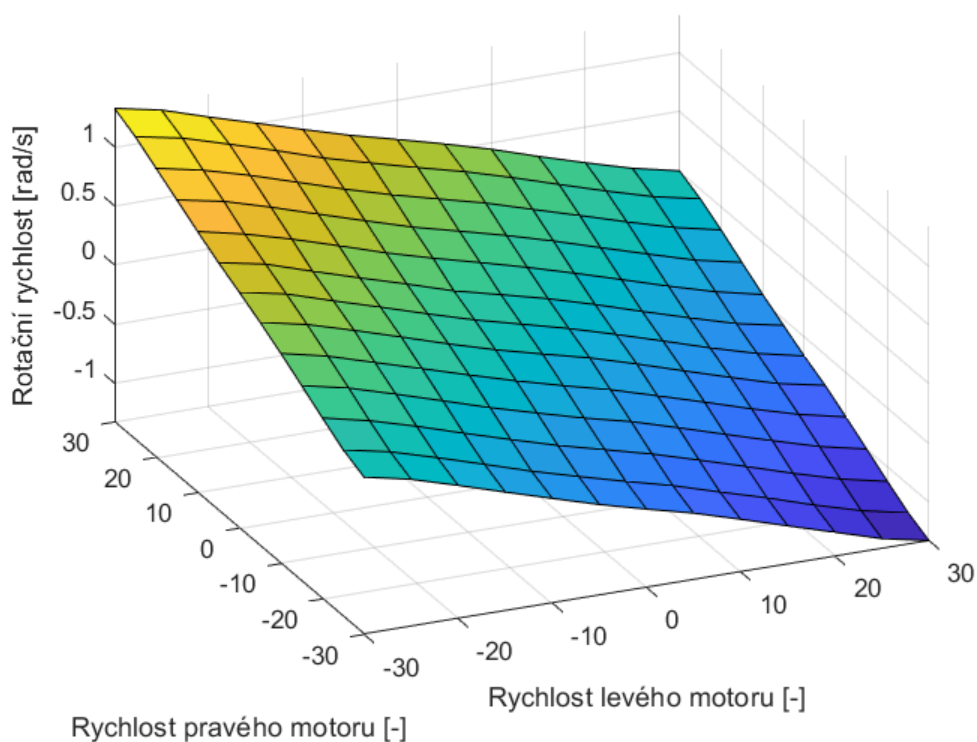
Po zprůměrování obou měření, vydělení čtyřmi a převedení na [rad/s] získáme hledanou závislost rychlosti rotace robotické jednotky v radiánech za sekundu na velikosti interní rychlosti levého a pravého motoru, viz tabulka 6.5.

■ Výsledný odhad parametrů

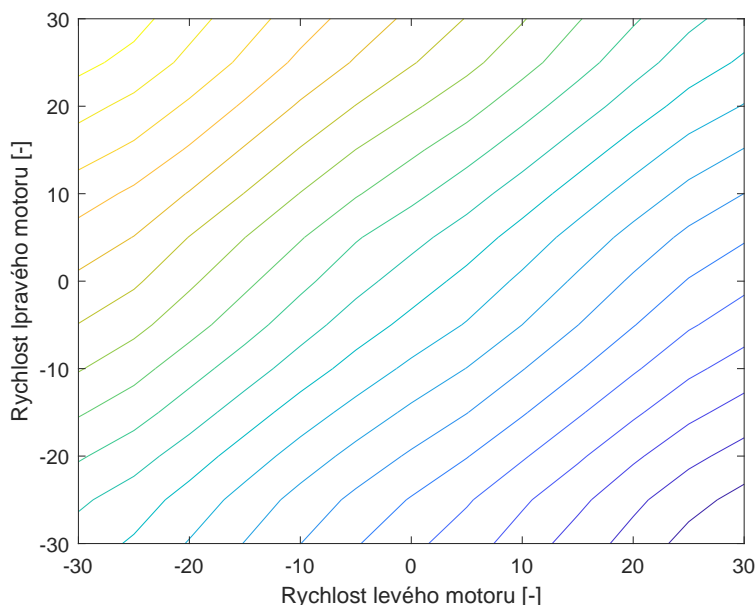
Po vynesení výsledků získáme následující 3D graf (viz obrázek 6.5). Pro lepší názornost jsou jednotlivé body měření propojeny pomocí malých plošek.

Tabulka 6.5: Závislost rotační rychlosti [rad/s] (bílá) na rychlosti motorů (šedá)

/	-30	-25	-20	-15	-10	-5	0	5	10	15	20	25	30
-30	0.00	-0.08	-0.20	-0.32	-0.44	-0.56	-0.67	-0.77	-0.88	-1.01	-1.13	-1.25	-1.33
-25	0.08	0.00	-0.12	-0.24	-0.36	-0.47	-0.58	-0.68	-0.80	-0.92	-1.05	-1.16	-1.24
-20	0.20	0.11	0.00	-0.12	-0.24	-0.36	-0.46	-0.57	-0.68	-0.80	-0.93	-1.04	-1.13
-15	0.33	0.24	0.12	0.00	-0.12	-0.24	-0.35	-0.44	-0.56	-0.68	-0.80	-0.92	-1.00
-10	0.45	0.36	0.24	0.12	0.00	-0.11	-0.22	-0.32	-0.44	-0.56	-0.68	-0.79	-0.88
-5	0.56	0.48	0.36	0.24	0.12	0.00	-0.10	-0.20	-0.32	-0.44	-0.57	-0.68	-0.76
0	0.67	0.59	0.46	0.34	0.22	0.10	0.00	-0.10	-0.22	-0.34	-0.47	-0.58	-0.67
5	0.77	0.69	0.56	0.44	0.32	0.20	0.10	0.00	-0.12	-0.24	-0.36	-0.48	-0.56
10	0.88	0.79	0.69	0.56	0.44	0.33	0.22	0.12	0.00	-0.12	-0.24	-0.36	-0.45
15	0.10	0.92	0.81	0.68	0.56	0.44	0.34	0.24	0.12	0.00	-0.12	-0.24	-0.32
20	1.12	1.03	0.93	0.80	0.68	0.56	0.46	0.36	0.24	0.12	0.00	-0.11	-0.20
25	1.24	1.16	1.04	0.92	0.79	0.68	0.58	0.48	0.36	0.24	0.11	0.00	-0.09
30	1.32	1.24	1.12	1.00	0.88	0.77	0.67	0.56	0.45	0.32	0.20	0.09	0.00

**Obrázek 6.5:** Závislost rychlosti rotace na interní rychlosti motorů

Jak je vidět, jedná se o část roviny. Jde o lineární funkci, kdy rychlost robotické jednotky je přímo úměrná rozdílu rychlostí levého a pravého motoru. Pro ještě větší přehlednost vyneseme stejný graf ve 2D pomocí ekvipotenciál (viz obrázek 6.6).



Obrázek 6.6: Závislost rychlosti rotace na interní rychlosti motorů

Jak je opět vidět, nezáleží na dílčích rychlostech motorů, ale pouze na rozdílu jejich rychlostí. V tomto grafu (6.6) si však ještě můžeme všimnout, že funkce v krajních případech není úplně lineární (hlavně v levém horním a v pravém dolním rohu, obvod grafu). Při vyšších rychlostech a větší dynamice robotické jednotky, dochází k výraznějšímu prokluzování kol, a tedy k nelineárnímu chování systému. Maximální rychlost robotické jednotky proto při reálné aplikaci omezíme.

Na základě těchto výsledků můžeme aproximovat rychlost rotace robotické jednotky v závislosti na rozdílu interních rychlostí levého a pravého motoru pomocí lineární funkce. Pokud nebude diference v rychlostech motorů příliš velká, nebude docházet k nelineárnímu chování a lineární aproximace bude skutečně validní.

Měření, která mají stejný rozdíl v rychlostech motorů (již zmiňované diagonály tabulky 6.5 z předchozí sekce), sečteme a zprůměrujeme. Ve výsledku získáme lineární závislost (viz obrázek 6.7), kterou budeme opět fitovat pomocí lineární regrese stejným způsobem, jako v předchozí kapitole.

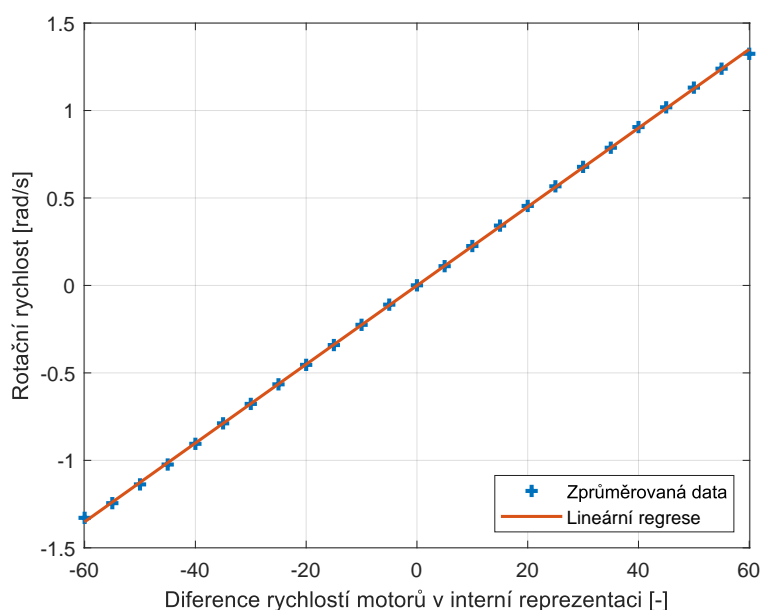
Po dopočtení parametrů lineární regrese získáme

$$g(y) = 0,023 \cdot y - 0,001 \quad (6.4)$$

, kde g značí funkční závislost rotační rychlosti robotické jednotky v [rad/s]

na diferenci rychlostí motorů y . Pro převedení získáme hledanou závislost difference rychlostí motorů na rotační rychlosti (6.5).

$$y(g) = \frac{g + 0,001}{0,023} \quad (6.5)$$



Obrázek 6.7: Lineární regrese závislosti rotační rychlosti na diferenci rychlostí motorů

6.3.3 Shrnutí

Výsledkem této sekce je převodní vztah z translační rychlosti v [m/s] a rotační rychlosti v [rad/s] na interní rychlost řídicí jednotky motorů MD25 [28]. Budeme vycházet z následujícího obecného vzorce (6.6) a (6.7), který předpokládá lineární závislosti mezi těmito proměnnými (tento fakt již byl potvrzen experimentálním měřením v předchozích kapitolách, použití tohoto vzorce je proto skutečně validní).

$$V_p(v_t, v_r) = a \cdot v_t + \frac{1}{2}b \cdot v_r \quad (6.6)$$

$$V_l(v_t, v_r) = a \cdot v_t - \frac{1}{2}b \cdot v_r \quad (6.7)$$

,kde

- V_p je interní rychlost pravého motoru
- V_l je interní rychlost levého motoru

- v_t je požadovaná rychlost robotické jednotky v [m/s]
- v_r je požadovaná rotační rychlost robotické jednotky v [rad/s]
- a, b jsou neznámé lineární koeficienty převodu

. Po dosazení experimentálně nalezených hodnot z předchozích částí této kapitoly za koeficienty a, b získáme finální převodní vztah (6.8) a (6.9).

$$V_p(v_t, v_r) = \frac{v_t \cdot T_f}{(2 \cdot 9,326)\pi \cdot r} - \frac{0,275}{9,326} + \frac{1}{2} \cdot \frac{(v_r + 0,001)}{0,023} \quad (6.8)$$

$$V_l(v_t, v_r) = \frac{v_t \cdot T_f}{(2 \cdot 9,326)\pi \cdot r} - \frac{0,275}{9,326} - \frac{1}{2} \cdot \frac{(v_r + 0,001)}{0,023} \quad (6.9)$$

,kde

- $T_f = 360$ je počet impulzů encoderů na jedno plné otočení kola kolem své osy
- $r = 4,88$ [cm] je poloměr kol

6.4 Ladění parametrů

6.4.1 Odometrie

V případě odometrie se dají ladit dva parametry, poloměr kol robotické jednotky a jejich vzájemná vzdálenost. Tyto hodnoty byly naměřeny metrem a použity pro výpočet pozice (pro způsob výpočtu viz příloha D). Pro orientační ověření správnosti byl zvlášť testován výpočet translace a rotace.

Výpočet translace byl testováno tak, že motory robotické jednotky byly současně zapnuty po určitou dobu. Po vypršení časového intervalu se porovnala skutečná ujetá vzdálenost (měřená pomocí pásma), vůči hodnotám naměřeným pomocí odometrie. Byla naměřena následující data (viz tabulka 6.6).

Tabulka 6.6: Testování translace

Skutečná ujetá vzdálenost [m]	Vypočtená vzdálenost [m]	Rozdíl [cm]
19,3821	19,375	0,71
18,8244	18,85	-2,56
19,2342	19,251	-1,68
18,9241	18,934	-0,99
19,3619	19,35	1,19

Měření probíhala na chodbách FEL, kde je podlaha pokrytá dlaždicemi.

To byl problém, neboť při jízdě robotické jednotky po tomto povrchu, docházelo k nadskakování kol a přichycování zadního kolečka do spár. To mělo za následek vybočování robotické jednotky z kurzu. Ve výsledku jsme dostali průměrný rozdíl mezi odometrickou a měřenou vzdáleností v absolutní hodnotě okolo 1,4 cm.

Výpočet rotace byl orientačně testován vykonáním jedné celé otáčky. Cílem bylo, aby se robot trefoval na jednu a tu samou značku po každé otočce. Po doladění parametru vzájemné vzdálenosti kol, se robotická jednotka začala dobře trefovat na své původní místo (značku).

Po vyzkoušení odometrie spolu s lidarovým SLAMem však nastal problém. Odometrická vypočtená pozice špatně odpovídala pozici určené pomocí SLAM. Bylo jasné, že ladění při vzájemné translaci a rotaci v praxi nebylo možné.

Byl proto zvolen jiný postup ladění, vycházející z přesnosti lidarů a Cartographeru (viz *Robotická jednotka a Lokalizace robotické jednotky v interiéru*). Robotická jednotka byla umístěna do malé místnosti o vhodných parametrech, aby byla možná lokalizace pouze za pomoci lidarů. V Rviz pak bylo možné zobrazit rozdíl pozice určené Cartographerem a odometrií. Parametry pak byly empirickou metodou laděny tak dlouho, dokud nebylo dosaženo shody i v případě delších a komplikovaných trajektorií pohybu.

Výsledkem byl stabilní SLAM, kdy nedocházelo k rozporu mezi odmetrií a Cartographerem (bez odometrie).

6.4.2 Ladění robotické jednotky jako celku

Robotická jednotka je jako celek komplexní systém umožňující ladění velkého množství parametrů. I přes snahu vymyslet teoretický způsob jejich ladění se ukázalo, že chování robotické jednotky bylo do značné míry nepředpověditelné (z důvodu hardwaru od různých výrobců a několika zdrojů softwaru, spolu s velkým množstvím dalších problémů viz *Odstraňování softwarových problémů*).

Byla proto zvolena empirická metoda ladění, spočívající v postupném zkoušení samostatných parametrů a pečlivým dokumentováním jejich vlivů na funkci robotické jednotky. Dokumentace byla prováděna formou textových poznámek, fotek a pomocných videí.

Ve výsledku byla dosažena dobrá funkčnost a spolehlivost robotické jednotky. Pro informace o výsledné přesnosti viz kapitola *Experimenty*.

6.5 Odstraňování softwarových problémů

Při programování bylo objeveno několik zásadních chyb v základních hardwarových knihovnách jednotlivých komponentů robotické jednotky, které způsobovaly vysokou nespolehlivost – nepoužitelnost robotické jednotky. Odstraňování těchto chyb bylo značně zdoluhavé (někdy až v rámci týdnů) a tvořilo významný časový úsek z celkového času stráveného nad touto prací. Proto v následujících několika částech rozebereme některé z nejzávažnějších chyb a způsoby jejich odstranění.

■ 6.5.1 Příklady komunikace mezi Arduinem a MD25

Na stránce výrobce MD25 je možné nalézt ukázky softwaru pro Arduino, umožňující komunikaci Arduina s MD25 přes I2C [31]. Po použití této šablony však není možné dosáhnout některých funkcionalit MD25, jako například vypnutí automatické regulace motorů, vypnutí bezpečnostního zastavení motorů nebo reset encoderů. Pro aktivaci těchto funkcionalit se používá tzv. *command register*, zkráceně CMD. Ve zdrojovém kódu – definici adresy CMD je chyba. Adresa CMD je 0x10, nikoliv 0x00, jak má na svých stránkách výrobce. Pro přepsání této adresy bylo dosaženo plné funkčnosti MD25.

■ 6.5.2 Tutorial – Navigation Stack (Mobe base)

Pro základní nastavení Navigation Stacku [45] je možné použít oficiální tutoriál [46]. V tutoriálu jsou však syntaktické chyby, které se na první pohled těžko hledají. Vzhledem k tomu, že se jedná o oficiální tutoriál, byl výskyt chyb tohoto druhu značně neočekávaný. Konkrétně se jedná o zapomenutou čárku ve výrazu

- footprint: [[x0, y0], [x1, y1], ... , [xn, yn]]

v souboru `costmap_common_params.yaml`. Dále se pak jedná o přebytečné lomítko ve výrazu

- `global_frame: /map`

v souboru `global_costmap_params.yaml`.

■ 6.5.3 Komunikace vývojové desky přes I2C

Při ladění robotické jednotky se ukázal velký problém nestability vývojové desky Arduino Uno. Spolu s hardwarovými problémy (viz *Ochrana vývojové desky Arduino Uno před rušením*) vznikla těžko rozluštitelná série chyb, mající za následek vždy jednu a tu samou věc – zamrznutí vývojové desky. I přes problém s hardwarovým rušením na I2C nebyl důvod k tomu, aby docházelo k výpadku funkce celého Arduina Uno. Bylo zřejmé, že se musí též jednat o softwarový problém.

Nestabilita vývojové desky se projevovala již od začátku práce na robotické jednotce. S minimálním zatížením I2C však výpadek nastával jen velmi vzácně, a proto mu nebyla věnována patřičná pozornost. Přes značnou snahu nalézt chybu ve vlastním programu `Arduino_node`, nebyla žádná chyba nalezena. Vzhledem ke koronavirové karanténě a absolutnímu nedostatku laboratorních přístrojů v domácích podmínkách, bylo nutné zahájit debugging Arduina Uno pomocí interní LED. Tato značně pracná a zdlouhavá metoda ladění byla založená na rozsvícení a zhasínání interní LED. Při zamrznutí vývojové desky, bylo na základě svitu LED možné odhadnout, v jaké části kódu k zamrznutí došlo. Výsledek ladění byl celkem nečekaný. K zamrznutí docházelo v oficiální knihovně `Wire.h` určené pro komunikaci Arduina Uno přes I2C.

Po detektivní práci, se správným podmětem hledání na internetu, se ukázalo, že nestabilita vývojové desky, související s knihovnou Wire.h [47], je dlouhodobě známá věc [48]. Knihovna Wire.h není ochráněna proti rušení na I2C. Soubor twi.c obsahuje několik neochráněných čekajících while smyček, které při zarušení komunikace po I2C probíhají do nekonečna. Jedná se o významné bezpečnostní riziko, neboť při jakémkoliv zarušení I2C komunikace (v reálném světě naprosto běžně) nebo při závadě/utržení libovolného z I2C senzorů dojde ke kompletnímu zamrznutí celého vývojového kitu, který se dostane do nekonečné while smyčky.

I přes dlouhodobou známost tohoto problému, neexistuje jednoznačný způsob, jak danou problematiku řešit. Opravy nalezené na internetu se nezdály věrohodné, a proto nezbývala jiná možnost, než knihovnu Wire.h opravit vlastnoručně.

Do problematických nekonečných while smyček v souboru twi.c byly umístěny timery ve formě int proměnných, které počítají počet průchodů danou smyčkou. Knihovna je přepsána takovým způsobem, že při překročení určitého množství opakování je smyčka ukončena, aby byla detekována chyba přenosu a nedošlo k zamrznutí Arduina Uno.

Po odstranění tohoto problému se vývojová deska stala konečně stabilní a imunní vůči rušení na I2C.

6.5.4 Deadzony motorů a DWA planner

Při počátečním ladění DWA planneru (move base) docházelo k silnému rozkmitávání robotické jednotky. Někdy došlo k tak silným oscilacím, že se začala převracet celá hliníková kostra.

Jak se ukázalo, kombinace zpětnovazebního systému řídicí jednotky motorů MD25, spolu se zpětnovazebním řízením DWA planneru, dělalo problém. Po vypnutí zpětnovazebního řízení MD25 se situace zlepšila, ale objevil se nový problém.

Při otáčení robotické jednotky, posílá někdy DWA planner příliš slabé řídicí signály motorům. Motory tak nejsou schopny překonat své dead zony a robotická jednotka zůstane stát. Jak se ukázalo, DWA planner neumožňuje nastavení minimální rotační/translační rychlosti a dokonce nepodporuje rotaci na místě [6]. Situace byla řešena v uzlu My robot node. V případě, že se má robotická jednotka pohybovat, ale místo toho je pomocí odometrie detekované stání, je řídicí signál DWA planneru několikanásobně zesílen, dokud nedojde k opětovnému pohybu. Tímto způsobem se i částečně povedl vyřešit problém rotace na místě.

Kapitola 7

Experimenty

V této kapitole je možné nalézt všechny experimenty vykonané s robotickou jednotkou, spolu s jejich popisem a interpretací naměřených výsledků.

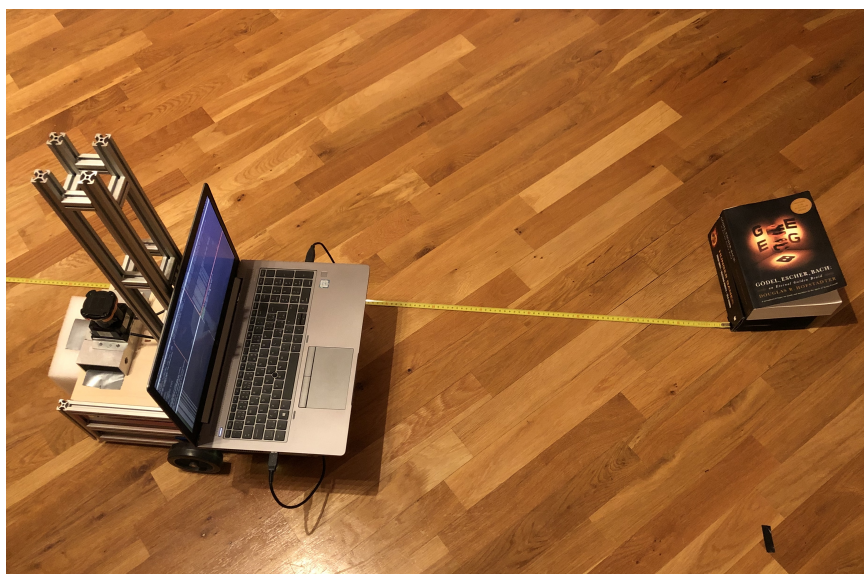
7.1 Experimenty přesnosti nájezdu do kontrolních bodů

Experimenty přesnosti nájezdu do kontrolního bodu byly uskutečněny v domácích podmínkách (kvůli koronavirové karanténě). Robotická jednotka byla umístěna do prostředku místnosti a pomocí elektrikářské pásky byla označena poloha jejích kol. Následně bylo robotické jednotce umožněno plné zmapování okolního prostředí, aby byla potlačena nejistota SLAM. Po zmapování místnosti začalo samotné měření, které probíhalo následujícím způsobem.

1. Robotické jednotce byl zadán náhodný cíl, dosažitelný bod v místnosti o libovolné orientaci.
2. Robotické jednotce byl zadán nový cíl, počáteční bod souřadnicové soustavy $x,y = [0, 0]$ s orientací shodnou s kladným směrem osy x .
3. Po dojetí robotické jednotky do počátku, byla pomocí elektrikářské pásky zaznamenána poloha kol. Poté se robotické jednotce zadal cíl trochu stranou, aby nezavázela samotnému měření. Ze vzájemné polohy počátečních a nových značek, bylo možné určit středy robotické jednotky a následně naměřit diferenci v osách x,y a orientaci vůči počátečnímu bodu. Po provedení měření byly značky označující dojezdový bod odstraněny (značky označující střed souřadnicové soustavy byly vždy zachovány) a následně se měření opakovalo bodem (1.).

7.1.1 První experiment

První experiment je zachycen na obrázku 7.1, kde je v pravém dolním rohu možné vidět značku z elektrikářské pásky spolu s knihami, které fungovaly jako opěrný bod metru. Pro měření byl dále použit úhломěr s několika standardními pravoúhlými pravítky.



Obrázek 7.1: Měření nájezdů do kontrolního bodu

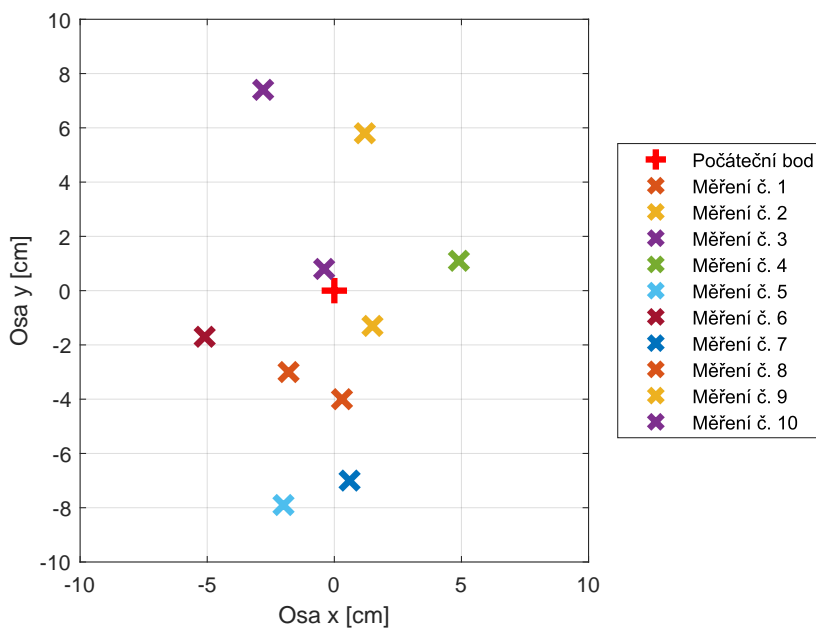
I přes to, že byl experiment prováděn se značnou pečlivostí, jedná se spíše o orientační měření. Určování rovnoběžných čar s počáteční pozicí bylo pomocí pravoúhlých pravítek zdlouhavé. V domácích podmínkách bylo celkem problematické zajistit přesné měření, a to hlavně při určování rotace robotické jednotky vůči počátečnímu bodu.

Výsledky měření je možné vidět v tabulce 7.1. Jedná se o deset měření, kde v jednotlivých sloupcích nalezneme rozdíly na osách x,y a diferenci v orientaci robotické jednotky vůči její počáteční pozici.

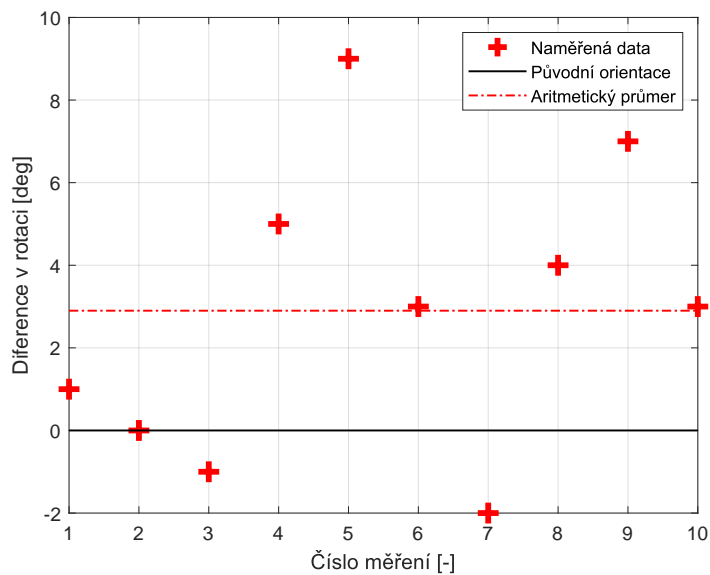
Tabulka 7.1: Diference dojezdu do počátku vůči skutečnému počátečnímu bodu

Diference na ose x [cm]	Diference na ose y [cm]	Diference rot z [deg]
-1.8	-3.0	1
1.5	-1.3	0
-0.4	0.8	-1
4.9	1.1	5
-2.0	-7.9	9
-5.01	-1.7	3
0.6	-7.0	-2
0.3	-4.0	4
1.2	5.8	7
-2.8	7.4	3

Pro lepší názornost byla data z tabulky 7.1 vynesena do dvou grafů, viz obrázek 7.2 a 7.3.



Obrázek 7.2: Diference dojezdu do počátku – osy x,y



Obrázek 7.3: Diference dojezdu do počátku – rotace

Na obrázku 7.2 je prostorové znázornění naměřených poloh vůči počátku ve 2D. Jak je vidět, naměřené polohy se zdají být náhodně rozmístěny okolo počátku. To je dobrým předpokladem pro další přesnější měření. Ideálně by

hustota naměřených bodů měla odpovídat trojdimenzionálnímu gaussovskému rozdělení.

Na obrázku 7.3 je zachycen rozdíl v orientaci robotické jednotky vůči počátku pro jednotlivá měření. Zde je situace horší. Naměřená data se zdají být také náhodně rozmístěna, ale ne podél počátku. Jak můžeme vidět, aritmetický průměr naměřených hodnot se pohybuje okolo $+2,9$ stupňů, což jasně poukazuje na systematickou chybu.

Výhodou této chyby je však fakt, že může být jednoduše softwarově odstraněna odečtením chybové konstanty od požadované orientace. Po odečtení této konstanty by se měl aritmetický průměr difference rotace pohybovat v okolí nuly. Tato hypotéza bude potvrzována pomocí dalších přesnějších měření.

7.1.2 Druhý experiment

Druhý experiment přesnosti nájezdu do kontrolního bodu byl uskutečněn po finálním doladění robotické jednotky. Probíhal stejným způsobem jako experiment první, s tím rozdílem, že mu byla věnována ještě větší pozornost. I nadále však přetrvává fakt, že se jedná spíše o orientační měření. Především určení rotace robotické jednotky vůči počátku, bylo značně komplikované. Na obrázku 7.5 je možné vidět průběh experimentu – umístování kontrolních značek a jejich vzájemné poměrování.



Obrázek 7.5: Průběh experimentu

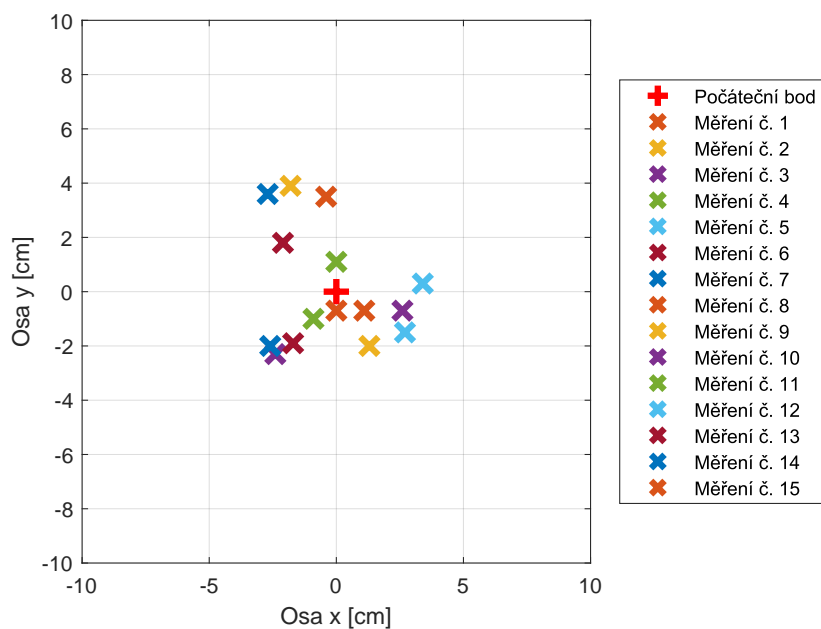
Výsledkem měření je tabulka 7.2. Pro lepší názornost jsou data z tabulky 7.1 vynesena do dvou grafů, viz obrázek 7.6 a 7.7.

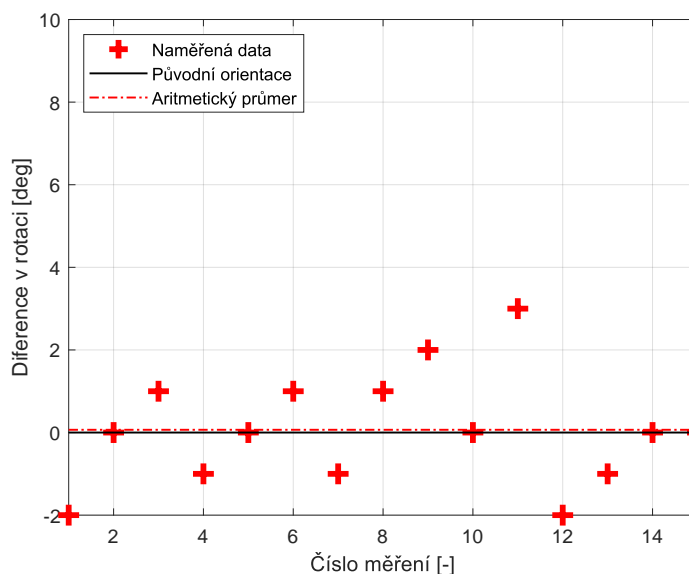
Je zřejmé, že oproti prvnímu měření došlo k znatelnému zlepšení přesnosti, a to ve všech směrech. Nejmenší přímá vzdálenost mezi počátkem a měřeným bodem byla 0,7 cm, největší 4,5 cm a v průměru okolo 2,68 cm.

Systematická chyba rotace (viz obrázek 7.7) se při druhém měření již neobjevila. Její příčinou bylo nejspíš špatné nastavení DWA planneru.

Tabulka 7.2: Diference dojezdu do počátku vůči skutečnému počátečnímu bodu

Diference na ose x [cm]	Diference na ose y [cm]	Diference rot z [deg]
-0.4	3.5	-2
-1.8	3.9	0
-2.4	-2.3	1
-0.9	-1	-1
3.4	0.3	0
-1.7	-1.9	1
-2.6	-2	-1
1.1	-0.7	1
1.3	-2	2
2.6	-0.7	0
0.0	1.1	3
2.7	-1.5	-2
-2.1	1.8	-1
-2.7	3.6	0
0.0	-0.7	0

**Obrázek 7.6:** Diference dojezdu do počátku – osy x,y



Obrázek 7.7: Diference dojezdu do počátku – rotace

7.2 Zátěžový test stability

Před nasazením robotické jednotky do skutečných testů měření osvětlenosti, byl provede zátěžový test. Cílem bylo prověřit:

- stabilitu hardwaru (hlavně vývojové desky Arduino Uno)
- stabilitu softwaru
- správné nastavení parametrů SLAM
- kontinuální funkci po delší dobu

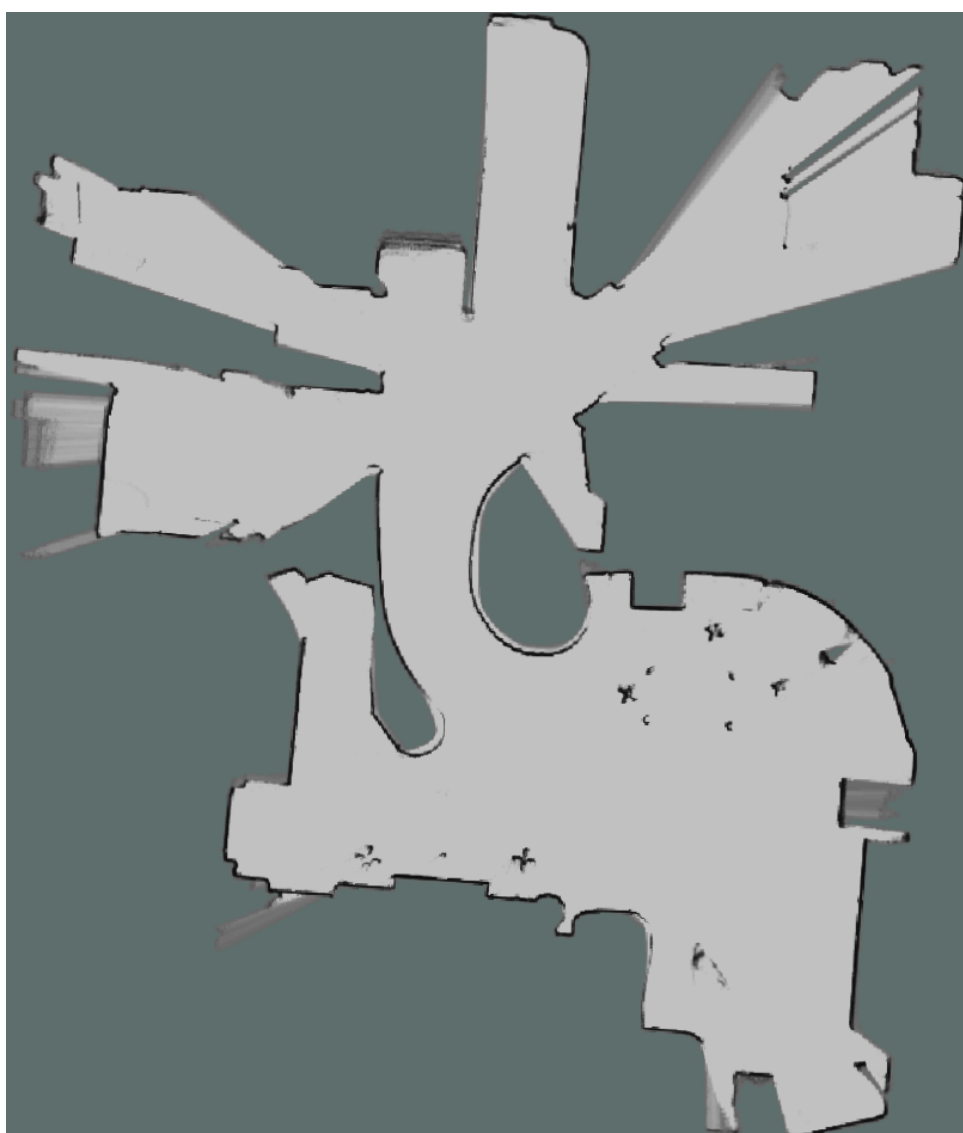
Jako zátěžové prostředí byl vybrán interiér s netradičním půdorysem, o několika místnostech, jež byly navzájem oddělené. Hlavní pozornost byla směřována na vytvoření správné mapy prostředí (nesmělo dojít k jejímu rozpadnutí) a schopnost lokalizace – pohybu robotické jednotky. Cílové body mapování byly zadávány manuálně pomocí GUI v Rviz [43].

Při prvních předběžných testech se vyskytl problém. Vždy po čase 20 – 40 minut došlo k výraznému rozpadnutí mapy. Tento fakt byl přisuzován nesprávnému nastavení parametrů SLAM, nebo jakémukoliv jinému nastavení softwaru. Jak se později ukázalo, problém byl hardwarového charakteru (viz kapitola *Robotická jednotka – Ochrana vývojové desky Arduino Uno proti poklesům napětí*).

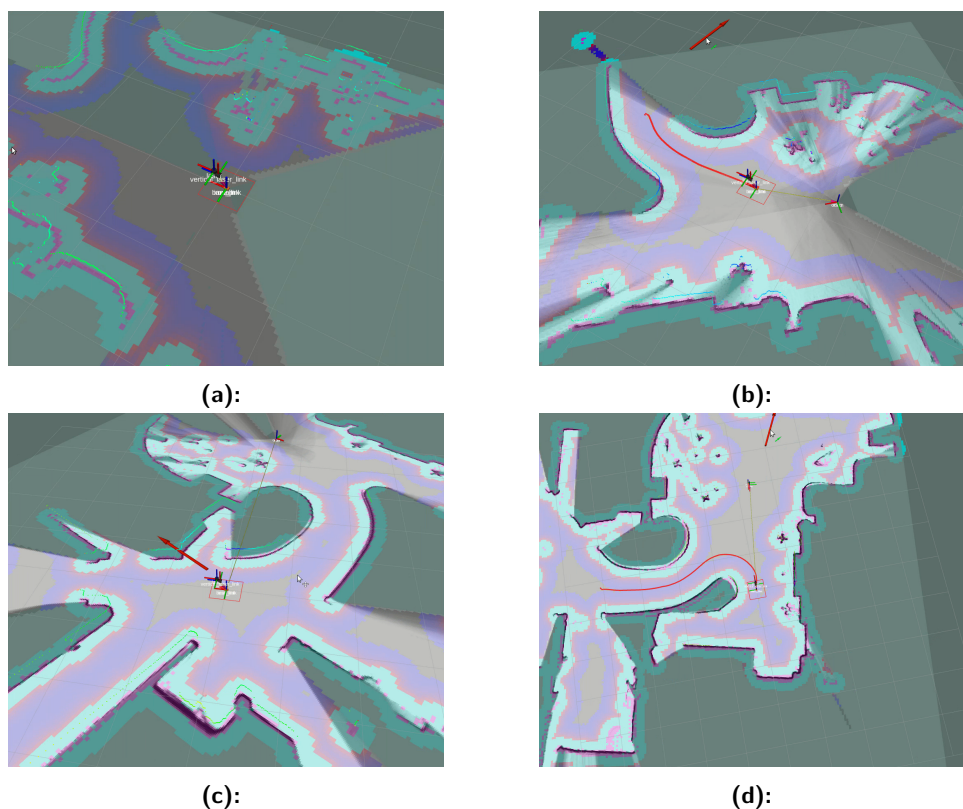
Po odstranění problému bylo konečně možné přistoupit k testům. Průběh kompletního mapování netradičního interiéru je znázorněn na obrázku 7.9.

Jak můžeme vidět, mapované prostředí je pro zátěžový test SLAMu skutečně ideální. Jeho nejasná struktura a rozsáhlost zamezuje jakékoliv viditelnosti mezi jednotlivými částmi. Na obrázku 7.9 dále můžeme vidět označené překážky, naplánovanou trajektorii a stopu robotické jednotky (červený obdélník). Výsledná mapa je znázorněna na obrázku 7.8.

Během testu nedošlo k jakémukoliv selhání robotické jednotky. Hardware i software (hlavně SLAM) splnily svoji funkci. Výsledná mapa s rozlišením jednotlivých buněk $2 \times 2 \text{ cm}^2$ dobře odpovídala skutečnosti. Výsledkem testu byla mapa s relativně vysokým rozlišením a jistota, že robotická jednotka může být bezproblémově použita pro další intenzivní testování.



Obrázek 7.8: Zátěžový test – výsledná mapa



Obrázek 7.9: Zátěžový test – průběh mapování

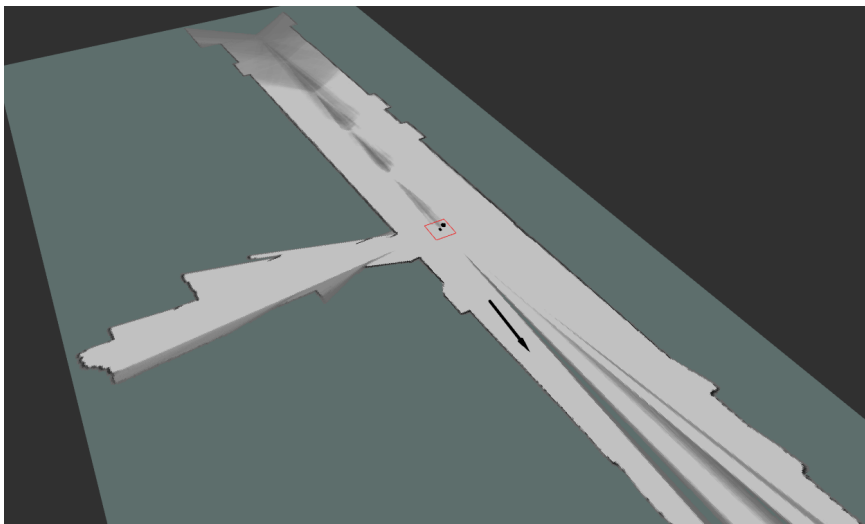
7.3 Série kontrolních bodů na přímce

Po předchozích experimentech bylo možné přistoupit k prvnímu základnímu simulovanému měření osvětlenosti. Robotické jednotce byla zadána série pěti bodů ve směru osy x . Vzdálenost mezi jednotlivými body byla 2 metry a orientace robotické jednotky byla souhlasná s osou x . Po nájezdu do kontrolního bodu setrvala robotická jednotka v klidu (5 sekund), než se vydala do bodu dalšího. Test byl uskutečněn na chodbě FEL ČVUT v Praze.

Výsledná mapa vytvořená robotickou jednotkou je vidět na obrázku 7.10. Linie táhnoucí se po diagonále je samotná chodba. Malý boční výběžek byl průhled do náhodně otevřených bočních dveří. Jak můžeme vidět, mapa v pravém dolním rohu je špatně vykreslená. To je z toho důvodu, že na konci chodby bylo skleněné okno, které neumožnilo lidar správně čtení vzdálenosti. Pravý horní roh mapy také není dobře vykreslený. To je zase z toho důvodu, že se tato část mapy nacházela ve slepém úhlu lidarů a robotická jednotka držela svou orientaci ve směru osy x . Zvláštní nezmapované části chodby jsou způsobené nájezdem do kontrolních bodů. Mobilní platforma po nájezdu do kontrolního bodu začala lehce rotovat, aby srovnala požadovanou orientaci s osou x . Tím dojde k zmapování malého okolí robota, ale nedojde k zmapování slepého úhlu.

Robotická jednotka fungovala dle očekávání. Problém však nastal v tom,

že se zadní kolečko neustále zachytávalo do spár mezi dlaždicemi. Mobilní platforma tak nebyla schopná najet do kontrolního bodu s požadovanou přesností. Situaci řešila neustálým opakováním nájezdu do kontrolního bodu. Po snížení nároku na přesnost došlo k vyřešení problému s několikanásobným nájezdem. Tento experiment ukázal, že na jiných než rovných površích není možné dosáhnout stejně dobrých výsledků nájezdu, jako v předchozích experimentech.



Obrázek 7.10: Měření na chodbě FEL ČVUT v Praze

7.4 Závěrečné testování

Závěrečné testování bylo provedeno na FEL ČVUT v Praze v učebně T2:A4–230a. Důraz byl kladen na to, aby místnost co nejvíce odpovídala běžným testovaným místnostem. Měření mělo následující průběh:

1. robotické jednotce bylo umožněno plné zmapování místnosti
2. robotické jednotce byla zadána sekvence 52 bodů, do kterých se musela dopravit a poté provést simulované měření

Mobilní platforma neprováděla skutečné měření osvětlenosti. Cílem bylo prokázání schopnosti lokalizace, mapování a následného nájezdu do 52 měřených bodů, které byly rozmístěny v pravidelné pravoúhlé síti. V každém z měřených bodů robotická jednotka setrvala deset sekund.

Na obrázku 7.11 můžeme vidět testovanou místnost spolu s robotickou jednotkou, která v okamžiku focení prováděla simulované měření.

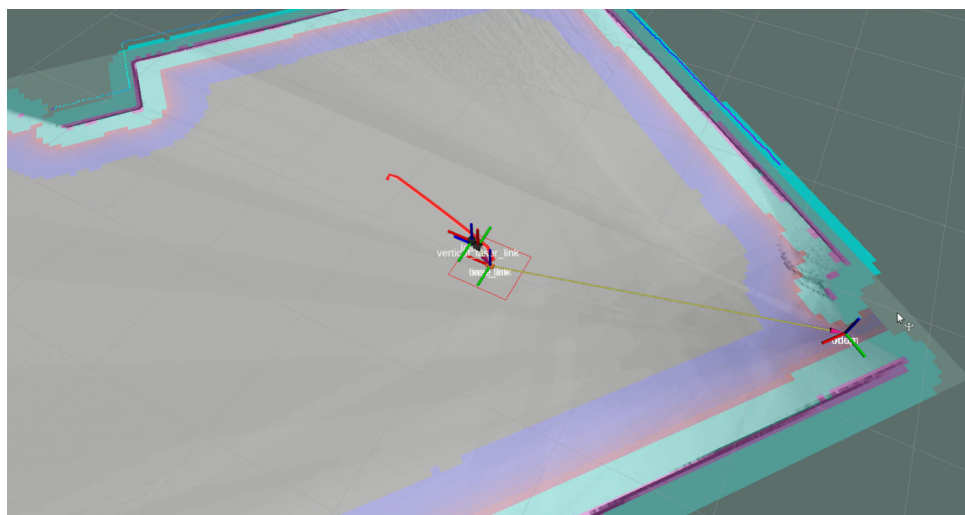
Na obrázcích 7.12 a 7.13 je ukázán pohled z palubního počítače mobilní platformy. Obrázek 7.12 zachycuje počáteční mapování měřené místnosti. Obrázek 7.13 dále ukazuje samotné měření ve formě najíždění do kontrolních bodů.

7. Experimenty

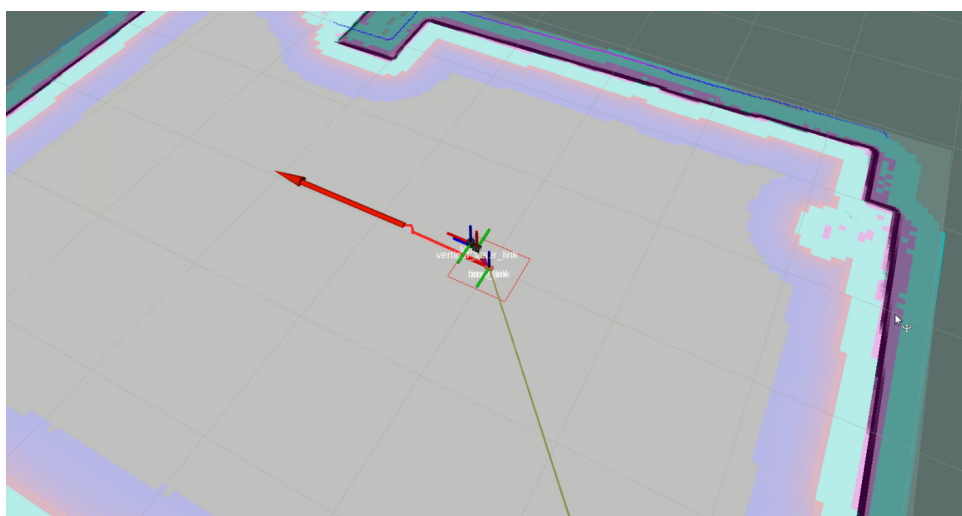
Tímto experimentem bylo dokázáno, že robotickou jednotku je možné použít pro skutečná měření osvětlenosti v běžných podmínkách.



Obrázek 7.11: Závěrečné testování – měřená místnost



Obrázek 7.12: Závěrečné testování – mapování



Obrázek 7.13: Závěrečné testování – nájezd do kontrolních bodů

7.5 Shrnutí

Série experimentů prezentovaná v této kapitole otestovala robotickou jednotku několika způsoby.

1. Byl testován nájezd do kontrolního bodu v různých místnostech, kdy na rovné hladké podlaze bylo dosaženo průměrné difference mezi požadovanou a skutečnou pozicí okolo 2,68 cm.
2. Nájezd do kontrolního bodu byl testován na několika různých typech povrchu. Nejlepších výsledků dosahovala robotická jednotka na rovných hladkých podlahách. I v případě nerovných povrchů (koberce, dlaždičky atd.) byla robotická jednotka schopna fungovat bez větších problémů (nedocházelo k rozpadání mapy). Při vysokých požadavcích na přesnost však musela robotická jednotka do některých bodů najíždět opakovaně. Opakovanému nájezdu do kontrolních bodů se dalo zabránit snížením požadavků na přesnost nájezdu.
3. Mobilní platforma byla podrobena zátěžovým testům, které ověřily její stabilitu.
4. Bylo ověřeno, že v případě rozsáhlých prostor nedochází k rozpadům mapy.
5. Robotická jednotka vykonávala simulovaná měření, kdy v měřené místnosti najížděla do série 52 kontrolních bodů, rozmístěných v pravidelné pravoúhlé síti.

Úkolem této práce bylo vytvoření softwaru pro robotickou jednotku, umožňující lokalizaci a mapování, s důrazem na nájezd do kontrolního bodu.

To se sérií výše popsaných experimentů podařilo potvrdit. V budoucnu by bylo vhodné určit přesnost nájezdu do kontrolního bodu nejenom experimentálně. Vzhledem k rozsahu práce a okolnostem (koronavirová karanténa) již nebyl čas se touto problematikou dále zabývat.

Kapitola 8

Závěr

Cílem této bakalářské práce bylo vytvoření a naprogramování softwaru pro robotickou jednotku, která měla za úkol měření osvětlenosti v interiérech v síti kontrolních bodů. V úvodu práce byla provedena rešerše v oblasti měření osvětlenosti (kapitola 2.), používaných senzorů pro mobilní robotiku v interiéru (kapitola 3.) a lokalizace–SLAM (kapitola 5.). Rešerše se dále zaměřovala na jednotlivé části, ze kterých se robotická jednotka skládá (kapitola 4.).

Praktická část spočívala ve vytvoření softwaru robotické jednotky (kapitola 6.). Software byl zaměřen na mapování, lokalizaci mobilní platformy a najíždění do předem zadaných kontrolních bodů. V průběhu práce na robotické jednotce se objevilo několik hardwarových a softwarových problémů, např. nestabilita vývojové desky, chyby v softwarových knihovnách. I přes nepříznivé podmínky se všechny problémy podařilo v domácích podmínkách vyřešit.

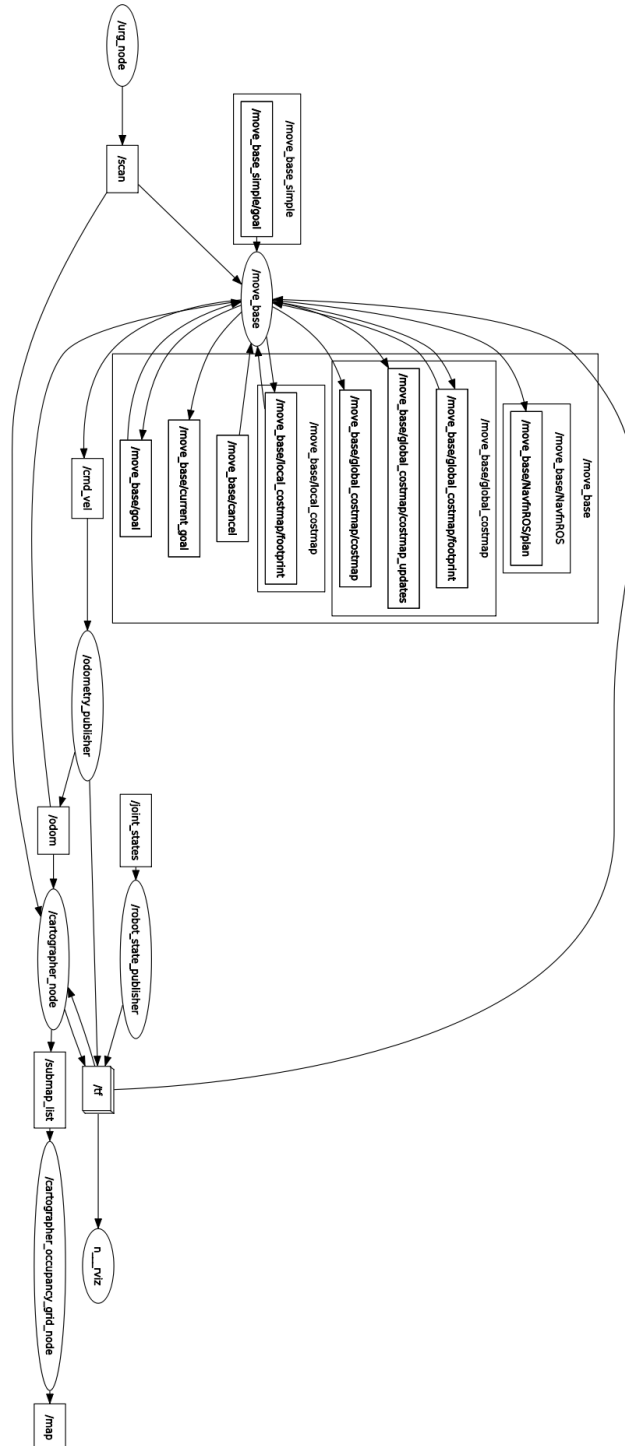
V závěrečné experimentální části (kapitola 7.) byla robotická jednotka testována v několika různých místnostech s různými povrchy. Dále byly testy zaměřeny na nájezdy do kontrolních bodů, jak na dopředný směr, tak na rotaci. Při postupném doladování robotické jednotky byl experiment nájezdu do kontrolního bodu opakován se zřejmým zlepšením. Zátěžový test dále ukázal, že mobilní platforma je schopna fungovat spolehlivě delší dobu, a že při mapování rozsáhlejších interiérů nedochází k rozpadnutí mapy. Na závěr bylo přistoupeno ke komplexnímu testu. Robotická jednotka nejdříve zmapovala prázdnou testovanou místnost. Poté ji byla zadána sekvence kontrolních bodů, do kterých musela dopravit svůj střed. V těchto kontrolních bodech bylo prováděno simulované měření osvětlenosti testované místnosti. Tím bylo potvrzeno, že robotická jednotka je použitelná pro skutečná měření osvětlenosti v interiérech.

V navazující práci by bylo vhodné rozvinout ovládání motorů, aby bylo dosaženo lepší plynulosti pohybu. To by mohlo být uskutečněno přepsáním DWA planeru, který neumožňuje rotaci na místě. Dále by bylo potřeba uskutečnit rozsáhlejší testy a určit přesnost nájezdu robotické jednotky do kontrolních bodů. V neposlední řadě je také možnost dalšího rozšiřování měřených veličin a zaměřit se na lepší řešení zadního opěrného kolečka, které hlavně na nerovném povrchu snižovalo přesnost nájezdu do kontrolního bodu.



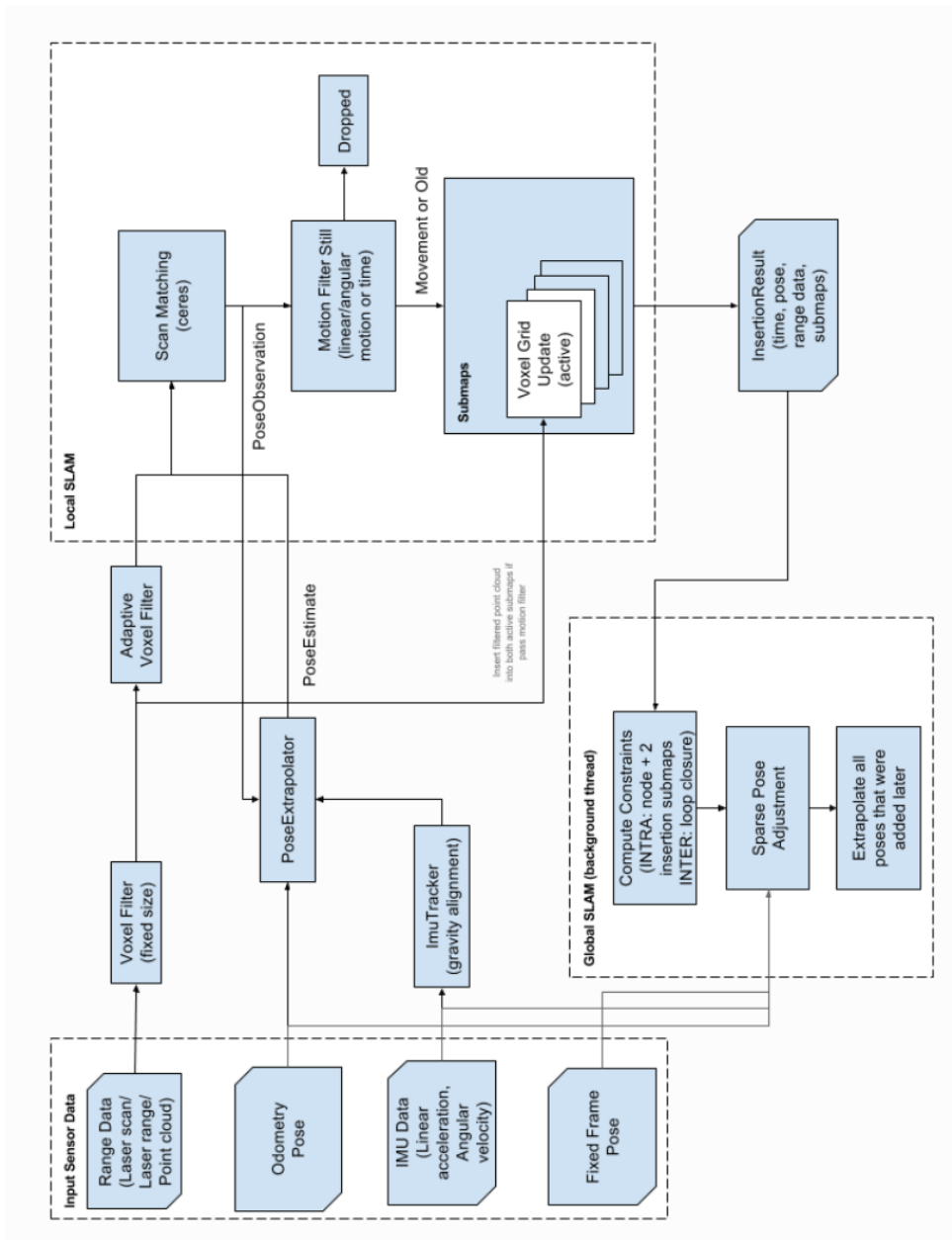
Přílohy

8.1 Příloha A



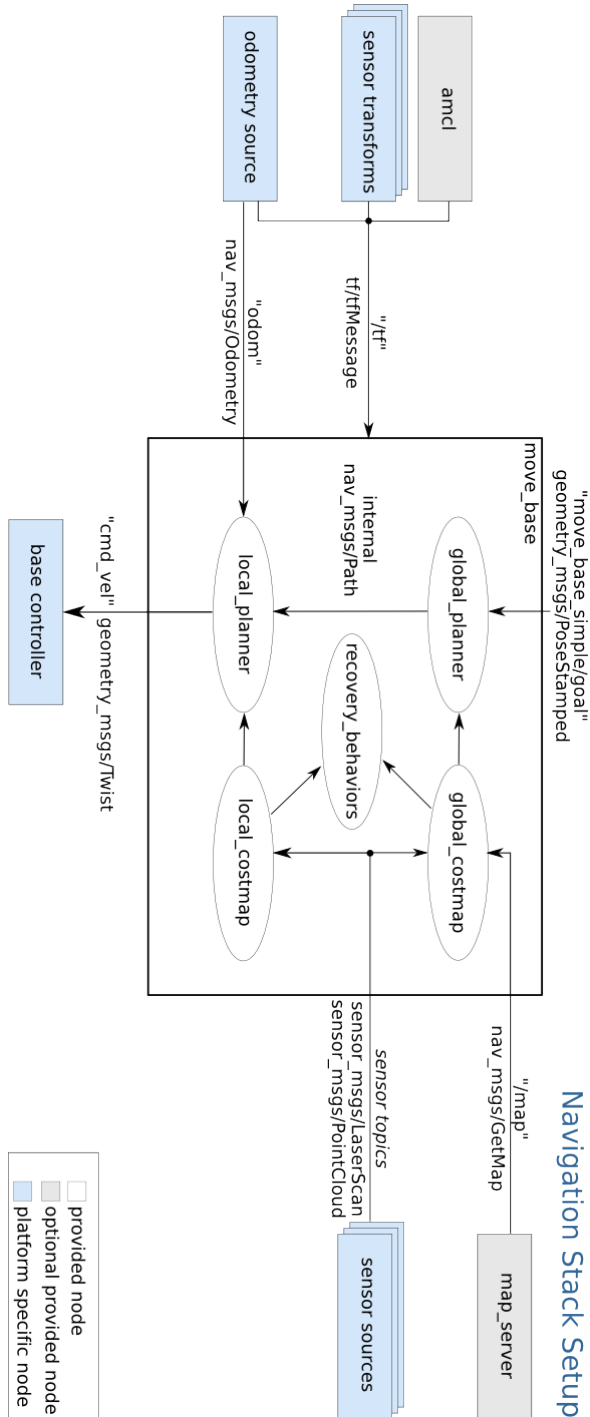
Obrázek 8.1: Kompletní propojení jednotlivých uzlů v ROS

8.2 Příloha B



Obrázek 8.2: Přehled Cartographeru [5]

8.3 Příloha C



Obrázek 8.3: Přehled Move base [6]

8.4 Příloha D

```

1 #-->>left wheel - travel distance
2 dl = -2 * pi * wheelradius * delta_L / TPR #Ticks Per Rotation
3 #-->>right wheel - travel distance
4 dr = -2 * pi * wheelradius * delta_R / TPR #delta_R from /odom
5 dc = (dl + dr) / 2
6 #-->>time
7 dt = (current_time - last_time).to_sec()
8 dth = (dr - dl) / wheeltrack
9 #-->>translation
10 if dr == dl:
11 #-->>new contribution on the x,y-axis
12 dx = dl * cos(th)
13 dy = dl * sin(th)
14 #-->>rotation
15 else:
16 radius = dc / dth
17 iccX = x - radius * sin(th)
18 iccY = y + radius * cos(th)
19 #-->>new contribution on the x,y-axis
20 dx = cos(dth) * (x - iccX) - sin(dth) * (y - iccY) + iccX - x
21 dy = sin(dth) * (x - iccX) + cos(dth) * (y - iccY) + iccY - y
22 #-->> x,y-axis plus dx,dy contribution
23 x += dx
24 y += dy
25 #-->>pose
26 th = (th + dth) % (2 * pi)

```

8.5 Příloha E

Tabulka 8.1: Pravý motor – koberec

Interní rychlost [–]	Měření I [IMP/5s]	Měření II [IMP/5s]	Měření III [IMP/5s]
-50	2352	2350	2346
-45	2110	2106	2104
-40	1850	1852	1848
-35	1622	1610	1618
-30	1352	1356	1356
-25	1188	1188	1188
-20	951	948	951
-15	698	697	698
-10	446	446	446
-5	209	208	208
0	0	0	0
5	-206	-206	-208
10	-447	-443	-444
15	-697	-697	-698
20	-949	-948	-944

25	-1186	-1180	-1182
30	-1361	-1358	-1360
35	-1612	-1618	-1621
40	-1840	-1848	-1846
45	-2081	-2085	-2085
50	-2332	-2328	-2336

Tabulka 8.2: Právý motor – chodba

Interní rychlost [–]	Měření I [IMP/5s]	Měření II [IMP/5s]	Měření III [IMP/5s]
-50	2357	2362	2351
-45	2117	2118	2116
-40	1862	1874	1855
-35	1626	1626	1623
-30	1370	1365	1370
-25	1195	1189	1193
-20	953	952	948
-15	706	704	700
-10	456	448	451
-5	215	214	214
0	0	0	0
5	-212	-214	-214
10	-453	-449	-450
15	-703	-702	-699
20	-950	-954	-954
25	-1190	-1199	-1196
30	-1374	-1375	-1378
35	-1628	-1633	-1622
40	-1867	-1855	-1870
45	-2104	-2097	-2102
50	-2348	-2348	-2334

Tabulka 8.3: Právý motor – lino

Interní rychlost [–]	Měření I [IMP/5s]	Měření II [IMP/5s]	Měření III [IMP/5s]
-50	2356	2356	2345
-45	2107	2111	2117
-40	1864	1860	1860
-35	1628	1618	1626
-30	1361	1364	1366

-25	1190	1193	1192
-20	953	952	952
-15	700	707	700
-10	458	462	458
-5	216	214	210
0	0	0	0
5	-210	-212	-209
10	-457	-456	-450
15	-700	-702	-704
20	-954	-955	-956
25	-1194	-1195	-1188
30	-1378	-1373	-1376
35	-1619	-1630	-1634
40	-1864	-1858	-1866
45	-2108	-2096	-2105
50	-2349	-2333	-2350

Tabulka 8.4: Levý motor – koberec

Interní rychlost [–]	Měření I [IMP/5s]	Měření II [IMP/5s]	Měření III [IMP/5s]
-50	2353	2352	2342
-45	2108	2105	2103
-40	1848	1855	1851
-35	1619	1612	1618
-30	1351	1356	1356
-25	1186	1187	1187
-20	950	947	946
-15	697	693	699
-10	444	444	445
-5	207	208	206
0	0	0	0
5	-204	-204	-204
10	-445	-443	-450
15	-696	-697	-694
20	-950	-948	-946
25	-1184	-1180	-1183
30	-1360	-1357	-1357
35	-1608	-1611	-1620
40	-1834	-1852	-1852
45	-2091	-2086	-2091
50	-2332	-2322	-2335

Tabulka 8.5: Levý motor – chodba

Interní rychlost [–]	Měření I [IMP/5s]	Měření II [IMP/5s]	Měření III [IMP/5s]
-50	2351	2359	2358
-45	2115	2114	2125
-40	1861	1872	1866
-35	1627	1623	1626
-30	1377	1366	1368
-25	1192	1190	1196
-20	953	954	946
-15	699	710	707
-10	454	449	448
-5	220	212	209
0	0	0	0
5	-208	-212	-214
10	-452	-453	-452
15	-704	-702	-700
20	-957	-955	-953
25	-1192	-1197	-1194
30	-1375	-1378	-1368
35	-1627	-1633	-1627
40	-1860	-1862	-1857
45	-2099	-2094	-2102
50	-2332	-2336	-2339

Tabulka 8.6: Levý motor – lino

Interní rychlost [–]	Měření I [IMP/5s]	Měření II [IMP/5s]	Měření III [IMP/5s]
-50	2345	2352	2355
-45	2108	2110	2107
-40	1860	1865	1862
-35	1627	1620	1624
-30	1361	1367	1364
-25	1192	1191	1192
-20	951	951	954
-15	704	701	702
-10	451	453	455
-5	215	215	214
0	0	0	0

5	-210	-210	-210
10	-454	-452	-452
15	-701	-701	-702
20	-953	-955	-955
25	-1191	-1194	-1190
30	-1377	-1367	-1369
35	-1621	-1631	-1629
40	-1859	-1860	-1863
45	-2109	-2095	-2104
50	-2342	-2328	-2343



Literatura

- [1] C. S. Institute, “Čsn 36 0011-1: Lighting measurement in interiors. part 1: General regulations,” *Czech Standards Institute*, 2005.
- [2] C. S. Institute, “Čsn 36 0011-3: Lighting measurement in interiors. part 3: Artificial lighting measurement,” *Czech Standards Institute*, 2005.
- [3] T. Drábek, “Robotická jednotka pro měření osvětlenosti v interiérech v rovnoměrně optimalizované síti kontrolních bodů,” Master’s thesis, České vysoké učení technické v Praze, 2014.
- [4] M. Filipenko and I. Afanasyev, “Comparison of various slam systems for mobile robot in an indoor environment,” in *2018 International Conference on Intelligent Systems (IS)*, pp. 400–407, IEEE, 2018.
- [5] “Cartographer.” <https://google-cartographer.readthedocs.io/en/latest/>. Datum: 2020-04-21.
- [6] “Move base.” http://wiki.ros.org/move_base. Datum: 2020-04-22.
- [7] H. A. C. LTD, “Scanning laser range finder utm-30x/ln specification,” *Technical description*, 2012.
- [8] “EMG30.” <https://www.robot-electronics.co.uk/htm/emg30.htm>. Datum: 2020-03-22.
- [9] D. Rotman, “How technology is destroying jobs,” *Technology Review*, vol. 16, no. 4, pp. 28–35, 2013.
- [10] M. Ford, *Rise of the Robots: Technology and the Threat of a Jobless Future*. Basic Books, 2015.
- [11] C. J. Bay, T. J. Terrill, *et al.*, “Autonomous robotic building energy audits: Demonstrated capabilities and open challenges,” *ASHRAE Transactions*, vol. 123, p. 3, 2017.
- [12] T. Bělaška, “Inteligentní osvětlení průmyslové haly,” 2014.
- [13] C. S. Institute, “Čsn en 13032-1: Light and lighting measurement and presentation of photometric data of lamps and luminaires. part 1: Measurement and file format,” *Czech Standards Institute*, 2005.

- [14] N. Pfeifer and C. Briese, “Laser scanning—principles and applications,” in *GeoSiberia 2007-International Exhibition and Scientific Congress*, Citeseer, 2007.
- [15] P. Jensfelt, *Approaches to mobile robot localization in indoor environments*. PhD thesis, KTH, 2001.
- [16] A. Elfes, “Sonar-based real-world mapping and navigation,” *IEEE Journal on Robotics and Automation*, vol. 3, no. 3, pp. 249–265, 1987.
- [17] G. Benet, F. Blanes, J. E. Simó, and P. Pérez, “Using infrared sensors for distance measurement in mobile robots,” *Robotics and autonomous systems*, vol. 40, no. 4, pp. 255–266, 2002.
- [18] G. Jang, S. Lee, and I. Kweon, “Color landmark based self-localization for indoor mobile robots,” in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, vol. 1, pp. 1037–1042, IEEE, 2002.
- [19] Z. Tian, X. Fang, M. Zhou, and L. Li, “Smartphone-based indoor integrated wifi/mems positioning algorithm in a multi-floor environment,” *Micromachines*, vol. 6, no. 3, pp. 347–363, 2015.
- [20] N. Ganganath and H. Leung, “Mobile robot localization using odometry and kinect sensor,” in *2012 IEEE International Conference on Emerging Signal Processing Applications*, pp. 91–94, IEEE, 2012.
- [21] J. Palacin, I. Valganon, and R. Pernia, “The optical mouse for indoor mobile robot odometry measurement,” *Sensors and Actuators A: Physical*, vol. 126, no. 1, pp. 141–147, 2006.
- [22] A. N. Raghavan, H. Ananthapadmanaban, M. S. Sivamurugan, and B. Ravindran, “Accurate mobile robot localization in indoor environments using bluetooth,” in *2010 IEEE international conference on robotics and automation*, pp. 4391–4396, IEEE, 2010.
- [23] J. Biswas and M. Veloso, “Wifi localization and navigation for autonomous indoor mobile robots,” in *2010 IEEE international conference on robotics and automation*, pp. 4379–4384, IEEE, 2010.
- [24] J. Jung, S.-M. Lee, and H. Myung, “Indoor mobile robot localization and mapping based on ambient magnetic fields and aiding radio sources,” *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 7, pp. 1922–1934, 2014.
- [25] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, “Rgb-d mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments,” *The International Journal of Robotics Research*, vol. 31, no. 5, pp. 647–663, 2012.

- [26] “Kinect.” <https://developer.microsoft.com/cs-cz/windows/kinect/>. Datum: 2020-04-19.
- [27] M. Drumheller, “Mobile robot localization using sonar,” *IEEE transactions on pattern analysis and machine intelligence*, no. 2, pp. 325–332, 1987.
- [28] “MD25.” <https://www.robot-electronics.co.uk/htm/md25tech.htm>. Datum: 2020-03-22.
- [29] “ARDUINO UNO.” <https://store.arduino.cc/arduino-uno-rev3>. Datum: 2020-03-22.
- [30] L. Pfozter, J. Oberlaender, A. Roennau, and R. Dillmann, “Development and calibration of karola, a compact, high-resolution 3d laser scanner,” in *2014 IEEE International Symposium on Safety, Security, and Rescue Robotics (2014)*, pp. 1–6, IEEE, 2014.
- [31] “Arduino-MD25-Examples.” https://www.robot-electronics.co.uk/htm/arduino_examples.htm#MD25%20RD02%20Motor%20Controller. Datum: 2020-04-13.
- [32] D. R. Hofstadter *et al.*, *Gödel, Escher, Bach: an eternal golden braid*, vol. 20. Basic books New York, 1979.
- [33] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, “Robust monte carlo localization for mobile robots,” *Artificial intelligence*, vol. 128, no. 1-2, pp. 99–141, 2001.
- [34] H. Zhang, L. Zhang, and J. Dai, “Landmark-based localization for indoor mobile robots with stereo vision,” in *2012 Second International Conference on Intelligent System Design and Engineering Application*, pp. 700–702, IEEE, 2012.
- [35] X. Hou and T. Arslan, “Monte carlo localization algorithm for indoor positioning using bluetooth low energy devices,” in *2017 International Conference on Localization and GNSS (ICL-GNSS)*, pp. 1–6, IEEE, 2017.
- [36] N. Dinh-Van, F. Nashashibi, N. Thanh-Huong, and E. Castelli, “Indoor intelligent vehicle localization using wifi received signal strength indicator,” in *2017 IEEE MTT-S international conference on microwaves for intelligent mobility (ICMIM)*, pp. 33–36, IEEE, 2017.
- [37] N. Bulusu, J. Heidemann, D. Estrin, *et al.*, “Gps-less low-cost outdoor localization for very small devices,” *IEEE personal communications*, vol. 7, no. 5, pp. 28–34, 2000.
- [38] S. Nirjon, J. Liu, G. DeJean, B. Priyantha, Y. Jin, and T. Hart, “Coin-gps: indoor localization from direct gps receiving,” in *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*, pp. 301–314, ACM, 2014.

- [39] “ROS.” <http://wiki.ros.org/Documentation>. Datum: 2020-04-15.
- [40] “Atmega328.” <https://www.microchip.com/wwwproducts/en/ATmega328>. Datum: 2020-04-15.
- [41] “Urg node.” http://wiki.ros.org/urg_node. Datum: 2020-04-22.
- [42] “SetGoal.” <https://hotblackrobotics.github.io/en/blog/2018/01/29/action-client-py/>. Datum: 2020-04-22.
- [43] “Rviz.” <http://wiki.ros.org/rviz>. Datum: 2020-04-24.
- [44] M. Perlmutter and S. Breit, “The future of the mems inertial sensor performance, design and manufacturing,” in *2016 DGON Inertial Sensors and Systems (ISS)*, pp. 1–12, IEEE, 2016.
- [45] “Navigation Stack.” <http://wiki.ros.org/navigation>. Datum: 2020-04-13.
- [46] “Navigation stack tutorial.” <http://wiki.ros.org/navigation/Tutorials/RobotSetup>. Datum: 2020-04-13.
- [47] “Wire.h.” <https://www.arduino.cc/en/reference/wire>. Datum: 2020-04-13.
- [48] “Wire.h problems.” <https://github.com/arduino/Arduino/issues/1476>. Datum: 2020-04-13.