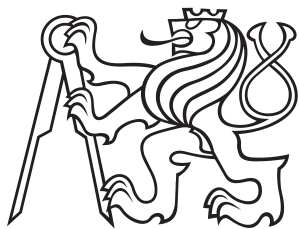


Bakalářská práce



České  
vysoké  
učení technické  
v Praze

**F3**

Fakulta elektrotechnická  
Kybernetika a Robotika

## Testování robotické jednotky pro měření hluku

**Jan Jakub Baroň**

Vedoucí: Ing. Tomáš Drábek  
Obor: Kybernetika a Robotika  
Květen 2021



## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Baroň** Jméno: **Jan Jakub** Osobní číslo: **483421**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávající katedra/ústav: **Katedra měření**  
Studijní program: **Kybernetika a robotika**

## II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

**Testování robotické jednotky pro měření hluku v interiéru**

Název bakalářské práce anglicky:

**Testing of a robotic unit for measuring indoor noise**

Pokyny pro vypracování:

Seznamte se s metodikou měření hluku v interiéru.  
Vytvořte rešerši v oblasti měření hluku v interiéru.  
Seznamte se s robotickou jednotkou a jejím softwarovým vybavením.  
Implementujte extrakci půdorysu z vytvořené mapy.  
Implementujte komunikaci mezi měřicím přístrojem a robotickou jednotkou.  
Otestujte software robotické jednotky v reálných podmínkách a software případně upravte.  
Vytvořte návod pro obsluhu robotické jednotky.

Seznam doporučené literatury:

- [1] ČSN EN 61672-1 ed. 2 Elektroakustika - Zvukoměry - Část 1: Technické požadavky
- [2] ČSN EN 61672-2 ed. 2 Elektroakustika - Zvukoměry - Část 2: Typové zkoušky
- [3] ČSN ISO 1996-1 Akustika - Popis, měření a hodnocení hluku prostředí - Část 1: Základní veličiny a postupy pro hodnocení (2017)
- [4] ČSN ISO 1996-2 Akustika - Popis, měření a hodnocení hluku prostředí - Část 2: Určování hladin akustického tlaku (2018)

Jméno a pracoviště vedoucí(ho) bakalářské práce:

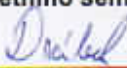
**Ing. Tomáš Drábek, katedra měření FEL**

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

Datum zadání bakalářské práce: **21.01.2021**

Termín odevzdání bakalářské práce: \_\_\_\_\_

Platnost zadání bakalářské práce:  
**do konce letního semestru 2021/2022**

  
Ing. Tomáš Drábek  
podpis vedoucí(ho) práce

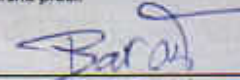
  
podpis vedoucí(ho) ústavu/katetry

  
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.

**5.5.2021**  
Datum převzetí zadání

  
Podpis studenta



## Poděkování

Děkuji Ing. Tomáši Drábkovi za vedení mé práce, za podporu a pomoc při jejím zpracování. Děkuji také panu Ing. Vladimíru Petříkovi, Ph.D., Bc. Petru Kůrkovi a Filipu Richterovi za přínosné konzultace.

## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a uvedl jsem veškerou použitou literaturu.

V Praze, 21. května 2021

## Abstrakt

Tato bakalářská práce rozšiřuje experimentální metodiku měření krátkodobého pravidelného a dlouhodobého stacionárního hluku. V kapitole 2 je zpracována rešerše na téma měření hluku. Kapitola 3 popisuje hardware mobilní robotické jednotky použité v měřicím procesu. Navazuje část se softwarovou implementací, ve které jsou popsány postupy implementace metod použitých v řetězci měření.

Těžištěm práce bylo nasazení mobilní robotické jednotky do reálných podmínek (viz kapitola 5), kde bylo zjištěno několik jejích nedostatků. V další kapitole jsou popsány kroky, nutné pro provedení měřicího procesu. V závěru práce jsou uvedeny náměty pro zlepšení dosavadní robotické jednotky a pro úplnou automatizaci měřicího procesu.

**Klíčová slova:** měření, mobilní robotická jednotka, krátkodobý pravidelný hluk, dlouhodobý stacionární hluk, mapování

**Vedoucí:** Ing. Tomáš Drábek  
Katedra měření,  
FEL ČVUT Praha.

## Abstract

The goal of this bachelor's thesis is to extend an experimental measurement method for measuring both short-term periodic noise and long-term stationary noise. Chapter 2 is a recherche into the noise measurement process and it also defines the main goals of this work. Chapter 3 describes the hardware of the mobile robotic unit used for the measurement. The next chapter details the software methods used in the measuring process and its implementation.

The main focus of this thesis lays in chapter 5 where the mobile robotic unit is tested in the real world conditions to find out whether it is capable of achieving its task. The next chapter is a step-by-step guide describing the measurement procedure. The conclusion summarizes future improvements of the robotic unit as well as improvements of the measurement process.

**Keywords:** measurement, mobile robotic unit, short-term periodic noise, long-term stationary noise, mapping

**Title translation:** Testing of a robotic unit for measuring indoor noise.

## Obsah

<b>1 Úvod</b>	<b>1</b>	4.2.1 Ramer–Douglas–Peucker algoritmus . . . . .	18
<b>2 Současný stav měření</b>	<b>3</b>	4.3 Implementace komunikace hlukoměru s Arduinem a řídicím počítačem . . . . .	19
2.1 Zavedený postup měření hluku . .	7	<b>5 Testování robotické jednotky v reálných podmínkách</b>	<b>21</b>
2.2 Experimentální měření hluku . . . .	8	5.1 Testy aproximační metody . . . . .	21
<b>3 Hardware</b>	<b>9</b>	5.2 Testování robotické jednotky pro měření hluku . . . . .	22
3.1 Rám robotické jednotky . . . . .	9	5.2.1 Testované místnosti . . . . .	22
3.2 Pohon robotické jednotky . . . . .	11	5.2.2 Problémy s kobercovým povrchem místnosti . . . . .	23
3.3 Řídicí deska Arduino UNO . . . . .	11	5.2.3 Problém implementace parametrů <i>anti_dead_zones</i> . . . . .	24
3.4 Měřicí senzory robotické jednotky	11	5.2.4 Řešení problému kolizí robotické jednotky se stěnami . . .	25
3.4.1 Laserový dálkoměr Hokuyo UTM–30LX . . . . .	12	5.2.5 Řešení problému s rozpadem mapy . . . . .	26
3.4.2 Hlukoměr Uni-T UT351 . . . . .	12	5.2.6 Testy přesnosti najíždění do cílového bodu . . . . .	27
<b>4 Softwarová implementace</b>	<b>15</b>	5.3 Finální test v místnosti 202a . . .	27
4.1 Extrakce půdorysu mapy . . . . .	15		
4.2 Matematická aproximace půdorysu mapy . . . . .	17		

5.3.1 Algoritmus dlouhodobého stacionárního hluku . . . . .	28	6.3.1 Spuštění jednotlivých uzlů ROS . . . . .	35
5.3.2 Algoritmus krátkodobého pravidelného hluku . . . . .	29	6.3.2 Uložení mapy a spuštění výpočetních algoritmů . . . . .	35
<b>6 Návod k obsluze</b>	<b>31</b>	<b>7 Závěr</b>	<b>39</b>
6.1 Popis řídicího softwaru robotické jednotky . . . . .	31	<b>A Literatura</b>	<b>41</b>
6.2 Kroky před samotným spuštěním robotické jednotky . . . . .	32	<b>B Zadání práce</b>	<b>45</b>
6.2.1 Hardwarová příprava robotické jednotky . . . . .	32		
6.2.2 Softwarová příprava robotické jednotky . . . . .	32		
6.2.3 Knihovny potřebné k běhu algoritmů . . . . .	33		
6.2.4 Nahrávání řídicích algoritmů do desek Arduin UNO . . . . .	33		
6.2.5 Identifikace portů měřicích senzorů robotické jednotky . . . . .	34		
6.3 Spuštění samotného měřicího procesu . . . . .	35		



## Obrázky

1.1 Ukázka rekonstrukce měřených dat laserovým dálkoměrem z chodby rodinného domu za pomoci Rviz Cartographer. ....	2
3.1 Schéma součástí robotické jednotky .....	9
3.2 Napájecí část robotické jednotky	10
3.3 Řídicí část robotické jednotky ..	10
3.4 Laserový dálkoměr Hokuyo UTM-30LX a polystyrenový nárazník .....	11
3.5 Hlukoměr Uni-T UT351 .....	12
3.6 Boční panel hlukoměru .....	13
4.1 Rekonstrukce mapy pokoje rodinného domu .....	16
4.2 Rekonstrukce mapy chodby rodinného domu .....	17
4.3 Pseudokód Ramer-Douglas-Peucker algoritmu .....	18
4.4 Použití Ramer-Douglas-Peucker algoritmu pro různá $\epsilon$ .....	19
5.1 Rekonstrukce matematické aproximace mapy chodby rodinného domu pro různé koeficienty $\epsilon$ .....	21
5.2 Rekonstrukce matematické aproximace mapy chodby rodinného domu pro $\epsilon = 3$ .....	22
5.3 Rekonstrukce matematické aproximace mapy školních tříd pro koeficient $\epsilon = 3$ .....	22
5.4 Ukázka <i>safe zones</i> .....	23
5.5 Robotická jednotka v místnosti 202b .....	24
5.6 Kolize robotické jednotky se stěnou .....	26
5.7 Rozpad mapy .....	26
5.8 První dvě iterace algoritmu pro určení dlouhodobého stacionárního hluku 202a .....	28
5.9 Dokončená mapa kontrolních bodů algoritmu dlouhodobého stacionárního hluku místnosti 202a	29

5.10 První dvě iterace algoritmu pro určení krátkodobého pravidelného hluku místnosti 202a .....	29
5.11 Dokončená mapa kontrolních bodů algoritmu krátkodobého pravidelného hluku místnosti 202a	30
6.1 Ovládací lišta Arduino IDE ....	34
6.2 Start mapování robotické jednotky .....	36

## Tabulky

6.1 Seznam knihoven .....	33
---------------------------	----

# Kapitola 1

## Úvod

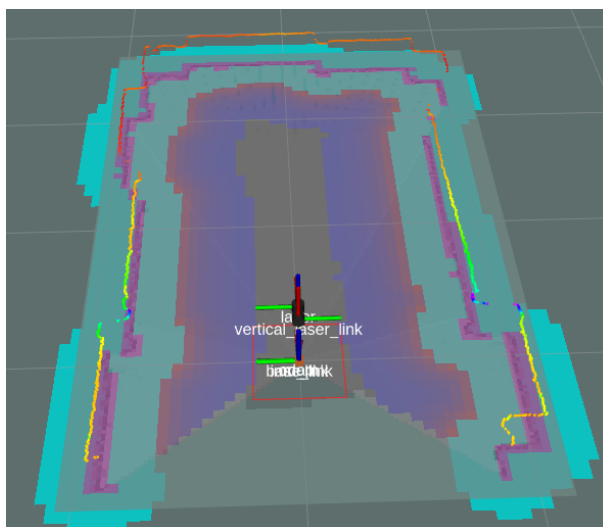
Někdy si to ani neuvědomujeme, ale hluku jsme vystaveni velkou část našeho života, ať už je to hluk od aut u silnice nebo naše oblíbená hudba nebo jen rozhovory lidí kolem nás. Tyto všechny každodenní aspekty ovlivňují hladinu hluku a také naše zdraví.

Hodnocení hluku je určeno mezinárodní normou [1], její česká verze je rozdělena do několika vládních nařízení, pro tuto práci budou nejvíce důležité [2], [3] a [4]. Za dodržování těchto nařízení je zodpovědná Česká hygienická stanice, která je také zodpovědná za správné měření hluku podle normy [5]. Postup měření pro dané prostředí je podrobně popsán v metodickém návodu ministerstva zdravotnictví [6]. Dále jsou stanoveny určité limity hluku pro jednotlivé typy prostor, ty jsou popsány v zákonech [7] a [8]. Kvůli těmto omezením vyvstává potřeba hluk měřit a měřené a zpracované výsledky porovnat se zavedenými limity.

I přes to, že metod pro měření hluku je velká řada, většina vědeckých publikací zabývajících se problematikou měření hluku řeší spíše zpracování výsledků měření pro zvolenou měřicí metodu.

Má studie je v tomto ohledu rozdílná, rozšiřuje totiž experimentální automatizovanou metodu určování kontrolních bodů pro měření hluku v interiéru místnosti [9]. Tato nová metoda určuje kontrolní body pro dva účely: krátkodobý pravidelný hluk a dlouhodobý stacionární hluk. Také navazuje na práci pana Kůrky [10], ve které je popsána použitá upravená robotická jednotka a algoritmus pro najíždění do stanovených pozic v místnosti.

Cílem mé práce je extrakce půdorysu místnosti (zobrazen na obrázku 1.1) pomocí laserového dálkoměru umístěného na robotické jednotce, nastavení správné komunikace mezi jednotlivými algoritmy měřicího procesu, vytvoření komunikace mezi robotickou jednotkou a měřicím přístrojem a testování celého měřicího procesu. Zároveň je mým úkolem napsat návod k obsluze robotické jednotky pro její další využití.



**Obrázek 1.1:** Ukázka rekonstrukce měřených dat laserovým dálkoměrem z chodby rodinného domu za pomoci Rviz Cartographer. Fialová barva značí stěny místnosti, světle modrá zónu, do které se robot nesmí dostat (z důvodu svých rozměrů). Tmavě modrá má význam zóny vhodné pro otáčení robota a nakonec šedá – kompletně volný prostor pro pohyb. Spodní souřadnicový systém značí origin robota, jeho startovní pozici. Horní souřadnicový systém značí pozici laserového dálkoměru. Značení je stejné pro všechny obrázky zobrazující vizualizaci uzlu Cartographer.

## Kapitola 2

### Současný stav měření

Měření hluku je velmi široká vědní oblast, avšak je možné ji pomyslně rozdělit dle lokace do dvou velkých kategorií, a to měření hluku v interiérech a exteriérech. Velkou část venkovního hluku tvoří zvuky dopravních prostředků. Lze najít celá řada článků o měření hluku od silnic a dálnic, železnic, přístavů či letišť.

Prvním příkladem je článek [5] autora Ahmada Jamraha zabývající se měřením hluku způsobeným dopravními prostředky ve městě Amman. Měření probíhalo na 28 místech dvakrát denně, a to brzy ráno a ve večerních hodinách, tedy v době, kdy lidé nejvíce jezdí do práce a z práce. Účelem bylo zachytit hlukovou špičku. Výsledky měření byly následovné. V ranních hodinách byla naměřena hladina hluku mezi 46 dB(A) a 81 dB(A) a večer mezi 58 dB(A) a 71 dB(A). Pro většinu měřících lokací byl tedy přesažen limit 62 dB(A).

Článek [11] je příkladem publikace zabývající se měřením hluku provozu aut. Byla použita tzv. měřicí metoda  $L_{den}$  a  $L_{night}$ . Zkratka *DEN* značí Day-Evening-Night, tedy celodenní měření ve třech blocích a  $L_{night}$  se zaměřuje pouze na měření nočního hluku. Pro měření byla vybrána velmi kontrastní oblast po stránce hlukové zátěže. Centrem této oblasti je totiž kampus dublinské univerzity, tedy dopravně velmi tiché místo. Výsledky byly zakresleny do hlukové mapy, ze které je tento kontrast velmi zřejmý. Data pro tuto hlukovou mapu byla získána mimo jiné od dopravního monitorovacího systému města Dublinu, který poskytoval data o typu projíždějících aut každou hodinu na křižovatkách dané oblasti. Výsledkem studie bylo nařazení provozu na centrálních trasách Dublinu a zavedení určitých dopravních omezení.

Studie probíhající v Brazílském Sao Paulu [12] zkoumala nejen zdejší hladinu hluku silničního provozu, ale hlavně faktor, jak moc ji tamější obyvatelé vnímají a nakolik jim vadí. Tento faktor byl zjišťován za pomoci dotazníků, které byly získány od 225 rezidentů dané oblasti. Dotazníky obsahovaly otázky ano/ne, popřípadě jak moc dotazovaným vadí daný typ hluku při dané aktivitě a to na stupnici jemně/středně/silně. Samotné měření hluku probíhalo v 75 kontrolních bodech hustě rozsetých po ulicích zkoumané městské části



redukci hluku na letištích. Analýza je prováděna na 248 evropských letištích. Letiště byla vyhodnocována na základě několika parametrů jako například počet ranvejí, míra letištního provozu, či vzdálenost nejbližšího města. Bylo zjištěno, že všechny tyto faktory mají vliv na protihluková opatření. Tedy pokud jsou tyto parametry vyšší, navyšují se i různá protihluková opatření. Pro pochopení důležitosti okolních vlivů na měření hluku v interiéru uvádím studii [16], která se zabývá vlivem otevřenosti oken na vnitřní hladinu hluku v místnosti. Autor článku provedl 102 měření v různých švýcarských apartmánech s cílem zjistit, jak moc ovlivňují otevřená, zavřená a přitopená okna hluk uvnitř místnosti, přicházející od silnice před apartmány. Na základě těchto měření byl vytvořen lineární regresní model, ze kterého byl vypočten medián rozdílu venkovního a vnitřního hluku. Rozdíly byly pro otevřená okna 10 dB(A), pro přitopená 16 dB(A) a pro zavřená 28 dB(A).

Hlukoměrů je celá řada a stále se vytvářejí nové jak ukazuje například bakalářská práce pana Kubíčka [17]. Ta zpracovává měření zvuku jakožto vědní disciplínu z obecné stránky. V druhé části práce se autor snaží vytvořit vlastní hlukoměr. Hlukoměr, který se mu podařilo sestavit slouží jen k orientačním účelům, jelikož funguje správně jen pro některé frekvence.

O měření hluku v budovách je možné najít spoustu dalších studií a vědeckých článků, tak například v publikaci [18], kde se autoři zabývají měřením hluku v továrně. Měření započali stanovením největších zdrojů hluku. Tyto zdroje poté zanesli do modelu továrny a studovali jejich dopad na sebe samé a na zaměstnance továrny. Tento přístup k měření hluku nese jméno RAP-ONE (Room Acoustics Prediction and Occupational Noise Exposure) a je jedním z velmi používaných přístupů měření. Pro kritická místa (místa s nejvyšší hladinou naměřeného hluku) v továrně pak byla zavedena časová omezení pobytu v těchto zónách, popřípadě nutnost použití protihlukových sluchátek. O trochu jiném měření průmyslového odvětví se píše v bakalářské práci [19], která se zabývá měřením hluku energetických strojů. V této práci je mimo samotnou metodu uvedeno jaké normy by měly stroje splňovat, a jak jejich hluk správně měřit. Například byla měřena hlučnost napájecího čerpadla. Zjistilo se, že hodnoty akustického tlaku garantované výrobcem nejsou překročeny a stroj tedy splňuje všechny požadavky co se týče emisní hlučnosti.

Další měření provedl pan Macháček ve své bakalářské práci [20]. Jeho práce zprvu pojednává o negativních faktorech hluku na lidské zdraví. V druhé části své práce provádí měření hluku na studenských kolejích Univerzity Karlovy a naměřené výsledky porovnává s platnými normami a limity v ČR. Měření

probíhalo v nočních hodinách od 22:00 do 6:00. Byla měřena tzv. ekvivalentní trvalá hladina hluku. Výsledkem měření bylo vždy jedno číslo, a to z toho důvodu, že použitý hlukoměr je integrační a výpočet, který provádí, je z velkého množství měřených hladin. Jednotlivé odpočty jsou v časových intervalech kratších než 1 s po celou zvolenou dobu měření. Limity hluku (35 dB(A)) byly ve všech budovách překračovány.

Jiná studie [21] se zaměřuje na dlouhodobé měření hluku po dobu 8 dnů. Měření probíhalo v 44 domácnostech s dětmi docházejícími na základní školu ve Francii. Byla použita metoda TLAD (time–location–activity diary), která funguje na principu zapisování jednotlivých výsledků měření dle doby, umístění a aktivity rodiny. Měření probíhalo v různých místnostech domu. Nejvyšší zastoupení hluku bylo vždy naměřeno v ložnici a poté v obývacím pokoji.

V nejmodernějších studiích se používají nové metody měření, jakými jsou například zapojení mobilních zařízení do procesu měření nebo třeba využití robotických jednotek či humanoidních robotů. Jednu z nových metod představuje například článek [22], ve kterém jsou pro měření použity chytré mobilní telefony. Tento článek se zabývá zpracováním naměřených dat do hlukových map. Zajímavostí je, že data jsou v této studii získávána od veřejnosti, a to díky měření za pomoci volně přístupné mobilní aplikace NoiseCapture. Takto změřená data jsou uložena s GPS souřadnicemi telefonu do databáze měření dané oblasti. Takto naměřená data jsou poté validována experty a je z nich vypočtena hluková mapa dané oblasti. Tímto přístupem metoda přispívá i k obecnému vzdělání lidí v oblasti měření hluku.

Maximální usnadnění práce přináší použití robota. To učinili ve studii [23], kde využili robotickou jednotku pro měření v kancelářích rozdělené do třech fází. V první fázi robot změřil teplotu, vlhkost, hluk a osvětlení dané kanceláře. Toto měření bylo provedeno ve 20ti bodech v místnosti. Následně v druhé fázi provedl dotazník s pracovníkem dané kanceláře. V poslední fázi robot vyhodnocoval získaná data. Na základě tohoto vyhodnocení poté pracovníkovi kanceláře sdělil doporučení, kterými by se měl řídit, pro větší komfort v jeho pracovně.

Dalším způsobem získávání dat může být použití robotické jednotky pro zmapování dané oblasti. Tato metoda je používána hlavně v nových studiích a cílí na ještě větší usnadnění lidské práce při měření hluku. V článku [24] se autoři zabývají problematikou šumu při měření zvuku. Pro redukci šumu představují dvě metody. První metoda je velmi rozšířená a pojednává o vytváření nejrůznějších filtrů pro odstranění určitých informací ze zvukových



vln. Druhou představovanou metodou, kterou se snaží tato studie rozvinout, je prosté přesunutí hlukoměru (právě za pomoci robotické jednotky) na lepší místo pro měření. Z tohoto důvodu využívají tvorbu hlukových map dané oblasti a následné přesunutí robotické jednotky na nejméně hlučná místa.

Více o metodě použití filtrů je možné zjistit například v publikaci [25]. Filtrů existuje mnoho typů, například váhové filtry. Jelikož lidské ucho vnímá jednotlivé zvukové frekvence s různou citlivostí, je přirozené měřit hluk s použitím váhových filtrů, které vyrovnávají tyto rozdíly zvukových frekvencí. Existují čtyři typy váhových filtrů a dělí se podle hlasitosti, vzestupně A, B, C, D. Filtry typu A, B, C mají stanovené hlasitostní pásmo použití, filtr typu D nemá přesné zařazení, je používán pro velmi vysokou hladinu hluku a je využíván například v letectví. Nejpoužívanější jsou filtry A a C, které jsou velmi často součástí běžně dostupných hlukoměrů.

Má studie na rozdíl od výše zmíněných článků neslouží primárně k získání a zpracování dat o měření hluku, ale rozšiřuje měřicí metodu popsanou v článku [9]. K měření je využita robotická jednotka, která má automatizovat celý proces měření a tedy cílí na maximální usnadnění lidské práce při měření hluku. Tato robotická jednotka byla upravena a popsána v bakalářské práci pana Kůrky [10].

Mým přínosem do projektu je softwarová implementace vytvoření polygonu (půdorysu) měřené místnosti. Začlenění tohoto půdorysu do měřicí metody popsané v práci [9]. A dále samotné měření hluku ve vypočtených kontrolních bodech a otestování celého měřicího procesu v reálných místnostech.

## 2.1 Zavedený postup měření hluku

Postup dosavadní měřicí metody je následující. Nejprve je třeba získat půdorys měřené místnosti a to za pomoci dálkoměru a nebo vyčíst tyto rozměry ze stavebního plánu místnosti. Měření probíhá v kontrolních bodech, kdy vzájemná vzdálenost dvou bodů nesmí být menší než 0,7 m. Jeden z měřicích bodů je vždy situován do rohu místnosti. Dalším omezením, vyplývajícím z technické normy [5], je nutnost umisťovat kontrolní body v minimální vzdálenosti 0,5 m od stěny a 1 m od všech zdrojů hluku v místnosti. Mezi tyto zdroje patří například okna, dveře nebo vývody klimatizace.

V průběhu měření musí být zavřena okna i dveře místnosti. K určování hodnot je použit hlukoměr popsaný v kapitole 3. Měření je prováděno v jednotlivých kontrolních bodech s hlukoměrem namířeným směrem ke zdroji hluku, pokud je znám, nebo vertikálně vzhůru, pokud není známo umístění zdroje hluku.

Měření probíhá ve výšce mezi 1,2 m až 1,5 m kolmo od podlahy místnosti.

## ■ 2.2 Experimentální měření hluku

Experimentální postup měření hluku za pomoci robotické jednotky lze shrnout následovně. Z dat LIDARU software robotické jednotky vypočte polygon, který je dále použit pro měření. V dalším kroku software vypočte kontrolní body pro měření hluku po místnosti pro dva uvedené účely: krátkodobý pravidelný hluk a dlouhodobý stacionární hluk.

Dlouhodobý stacionární hluk je emitován nehybnými stroji, například domácími spotřebiči. Z tohoto důvodu je kladen důraz na rozmístění kontrolních bodů co nejbližší u sebe tak, aby se jich do místnosti vešlo co nejvíce. Díky tomuto rozmístění lze pak snadněji zjistit, kde je intenzita hluku nejvyšší a tedy určit směr, kde se tyto spotřebiče nacházejí.

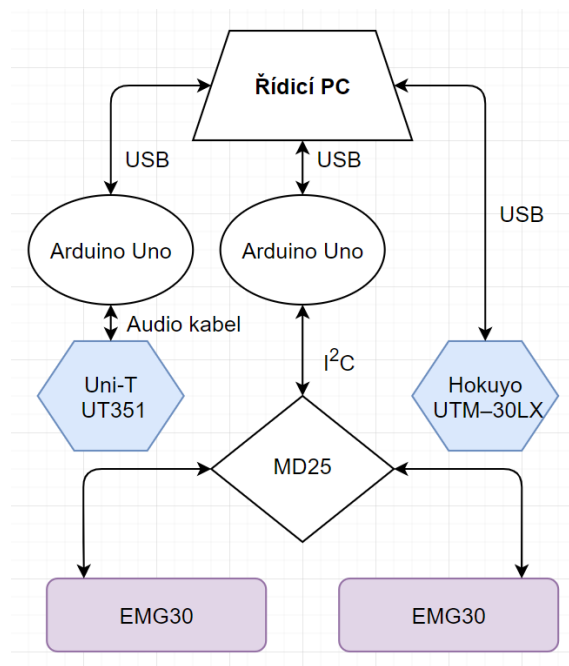
Na druhou stranu krátkodobý pravidelný hluk emitují pohybující se stroje. Rozmístění kontrolních bodů se provádí tak, že každý nově přidávaný kontrolní bod pokrývá co největší plochu měřicí oblasti. Jsou tedy umísťovány do nejvzdálenějšího bodu od již umístěných kontrolních bodů.

Měření bude probíhat stejně jako je popsáno v článku [9] a bude splňovat všechny parametry popsané v kapitole 2.1. Po určení kontrolních bodů pro jeden z uvedených účelů bude robotická platforma najíždět do vypočtených bodů a započne zde samotné měření hluku. Výsledky spolu s polohou daného měřeného bodu budou uloženy a mohou sloužit k dalšímu zpracování.

## Kapitola 3

### Hardware

Robotická jednotka využitá v mé práci vychází z diplomové práce pana Ing. Drábka [26]. O její rozšíření se zasloužil Bc. Kůrka ve své bakalářské práci [10]. V této kapitole shrnu jednotlivé součástky, ze kterých je jednotka sestavena. Nový měřicí přístroj – hlukoměr, který je na jednotku přidán v rámci mé práce, více popíši. Schéma celého propojení jednotlivých prvků robotické jednotky je pro přehlednost zaneseno v obrázku 3.1.

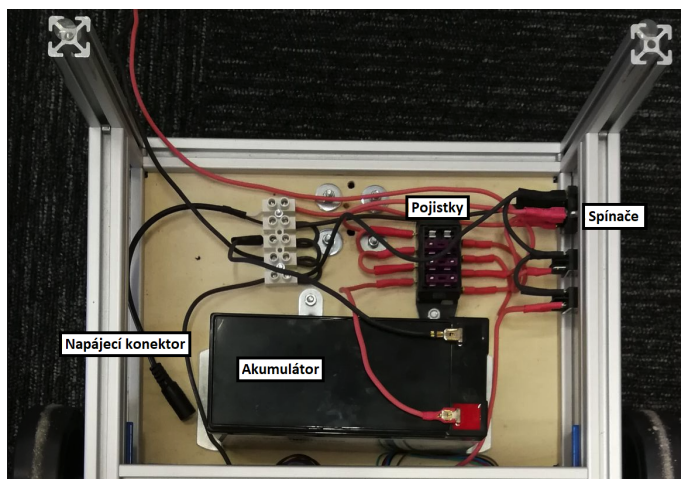


**Obrázek 3.1:** Schéma zapojení řídicích jednotek, motorů a senzorů robotické jednotky. Fialově jsou vyznačeny motory, modře senzory a bíle řídicí jednotky. Popis šipek značí typ sběrnice přenosu mezi danými zařízeními.

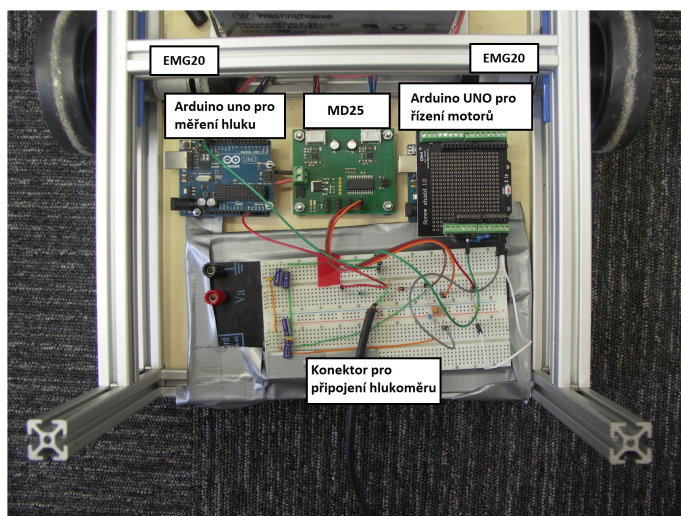
### 3.1 Rám robotické jednotky

Rám robota je sestaven z hliníkové konstrukce pro zajištění bezpečí ostatních komponent proti nárazu, ty jsou vkládány dovnitř konstrukce. Spodní patro konstrukce s napájecími součástkami je zobrazeno na obrázku 3.2. Je zde

vidět akumulátor, spínače, pojistky a napájecí konektor robotické jednotky. Na obrázku 3.3 jsou motory a jejich řídicí jednotka, desky Arduino a napájecí pole pro propojení těchto součástí.

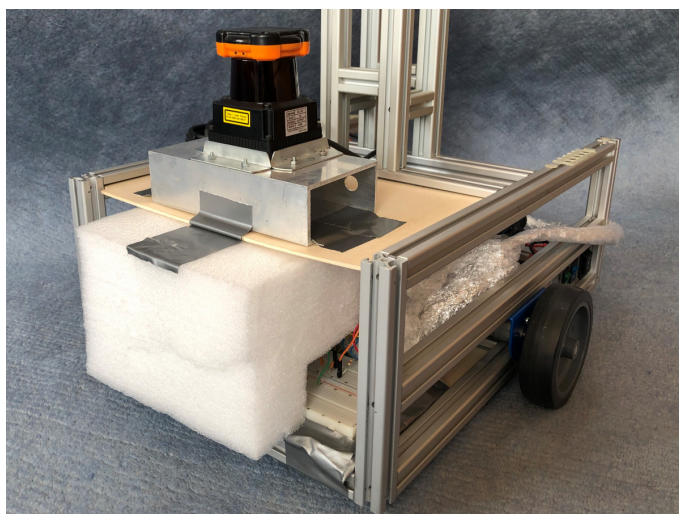


**Obrázek 3.2:** Konstrukce robotické jednotky a její napájecí část



**Obrázek 3.3:** Konstrukce robotické jednotky a její řídicí část

Na konstrukci je dále přidán polystyrenový blok, přesahující zepředu přes okraj konstrukce, pro utlumení nárazů při kolizích viz obrázek 3.4. Tento nárazník slouží zároveň jako podpora pro desku v horní části hliníkové konstrukce, na které je umístěn laserový dálkoměr, ten popíši dále v této kapitole.



**Obrázek 3.4:** Laserový dálkoměr Hokuyo UTM-30LX umístěný na horní podestě chladiče a polystyrenový nárazník/podpěra robotické jednotky pro zmírnění nárazů [10]

## 3.2 Pohon robotické jednotky

K pohonu robotické jednotky slouží dva motory EMG30 [27], které jsou ovládány za pomoci řídicí jednotky MD25 [28]. Tato jednotka je spojena nízkorychlostí sběrnici I<sup>2</sup>C s hlavní řídicí jednotkou Arduino-Uno [29]. Díky enkodérům (dvojici halových sond) na každém z motorů je možné zjišťovat polohu natočení motorů. Informace z těchto enkodérů jsou klíčové k navigaci robota. Napájení těchto motorů zajišťuje akumulátor NPG8-12, 12 V, 8 Ah, Shimastu [30].

## 3.3 Řídicí deska Arduino UNO

Pro propojení hlukoměru s robotickou jednotkou jsem využil další vývojové desky Arduino UNO [29]. Deska slouží k čtení změny stejnosměrného napětí na jejím analogovém pinu A3. Z této změny napětí díky svému převodníku vypočítá hodnotu hluku okolí změřenou hlukoměrem. Tuto hodnotu v dB dále komunikuje s řídicím počítačem za pomoci jednoduché sériové komunikace.

## 3.4 Měřicí senzory robotické jednotky

Na robotické jednotce jsou dva hlavní senzory. Jedním je laserový dálkoměr Hokuyo UTM-30LX [31] a druhým hlukoměr Uni-T UT351 [32], o který rozšiřuji robotickou jednotku.

### 3.4.1 Laserový dálkoměr Hokuyo UTM-30LX

LIDAR je umístěn do přední části robotické jednotky a je upevněn na kovovém chladiči. Jeho maximální dosah činí 30 m v měřícím úhlu 270°. Dokáže sbírat vzorky rychlostí 25 ms/vzorek s rozlišením 0,25°/vzorek. Je napájen 12 V akumulátorem robotické jednotky. Laserový dálkoměr je přesnější v detekci bližších objektů, jeho přesnost klesá se vzdáleností k objektu. Do předmětu vzdáleného 10 m je přesnost  $\pm 30$  mm. Pro vzdálenější objekty je tato přesnost  $\pm 50$  mm.

### 3.4.2 Hlukoměr Uni-T UT351

Jedná se o hlukoměr od čínské společnosti Uni Trend zobrazen na obrázku 3.5. Jsou k dispozici dva měřicí režimy přepínatelné pomocí tlačítka A/C.



Obrázek 3.5: Hlukoměr Uni-T UT351

Přepnutím mezi těmito módy se aktivují filtry typu A popřípadě C [25]. Dále je nastavitelná vzorkovací frekvence pro dva typy měření: krátkodobý pravidelný hluk (frekvence 125 ms) a dlouhodobý stacionární hluk (frekvence 1 s). Měřicí rozsah hlukoměru je 30-150 dB. K napájení je možné využít dvě možnosti. První možností je napájet hlukoměr z externího zdroje a druhou za pomoci baterií. Pro účely testování jsem hlukoměr napájel za pomoci baterií. Hlukoměr je propojen s deskou Arduina za pomoci upraveného kabelu. Ten bylo třeba modifikovat tak, aby šel jeho jeden konec připojit do nepájivého pole na obrázku 3.3 a druhý do zdířky DC výstupu hlukoměru viz obrázek 3.6. Tento propojovací kabel jsem vyrobil z audio kabelu připájením konektoru jack 3,5 mm na jeden z jeho konců.

Hlukoměr tedy komunikuje s deskou Arduina, potažmo s řídicím počítačem pomocí změn stejnosměrného napětí. Tato změna má rozlišení 10 mV/dB. Tedy hodnota odečtená na desce Arduina je třeba vynásobit konstantou 100, pro získání hodnoty dB okolního hluku.



**Obrázek 3.6:** Boční panel pro konektory hlukoměru UNI-T UT351. Zleva, kalibrační zdířka, výstup střídavého napětí, výstup stejnosměrného napětí, externí napájení hlukoměru.





## Kapitola 4

### Softwarová implementace

Software robotické jednotky je realizován pomocí ROS (Robotic Operating System) [33]. Jedná se o freewarovou implementaci souboru softwarových rámců. Celý tento systém funguje na principu uzlů (nodes).

Hlavním uzlem je ROScore – řídicí uzel, který je zodpovědný za předávání a zpracování dat ostatních uzlů. K tomuto řídicímu uzlu je možné přidávat další, ať už jsou to knihovny od poskytovatelů ROSu nebo samostatně doimplementované uzly. Nové uzly je možné psát v kterémkoliv z programovacích jazyků C, C++, C#, Python, Java, dokonce každý uzel může být napsaný v jiném.

Celý systém tedy funguje na bázi sestavení jednotlivých základních kamenů (uzlů) v komplexní systém, za pomocí kterého je robotická jednotka ovládána. Ve své implementaci využívám uzlů, které jsou popsány v práci [10] a zároveň přidávám své vlastní.

#### 4.1 Extrakce půdorysu mapy

Pro extrakci půdorysu místnosti jsem využil knihovnu ROSu *map\_server* [34]. Tato knihovna využívá k sestavení mapy data z uzlu *urg\_node*, který čte surová data z laserového dálkoměru a převádí je do prostředí ROS. Přesný popis uzlu *urg\_node* a dalších si lze přečíst v práci [10]. Pomocí této knihovny lze data z *urg\_node* přečíst a uložit do dvou formátů: YAML (Yet Another Markup Language) a PGM (Portable Grey Map).

V YAML souboru jsou popsána meta-data mapy, těmi jsou:

- *image* – cesta k datovému souboru mapy PGM
- *resolution* – měřítko pixelů v metrech tedy kolik metrů má jeden pixel
- *origin* – pixel dolního levého rohu mapy
- *pixel occupancy threshold* – pixel s pravděpodobností vyšší než je tato hodnota je vyhodnocen jako plný (okupovaný)

- pixel free threshold – pixel s pravděpodobností menší než je tato je vyhodnocen jako prázdný (neokupovaný)

V PGM souboru poté najdeme přímo hodnoty pro jednotlivé pixely. Formát PGM je jeden z nezákladnějších obrazových formátů, obsahuje totiž pouze hodnoty úrovně šedi každého pixelu. Dále je z této knihovny využít metodu *map\_saver* za pomoci kterého je možné tyto soubory uložit do počítače. Při zadávání tohoto příkazu lze volitelně nastavit pixel occupancy a pixel free threshold. Mapy uložené pomocí této metody jsou zobrazeny na obrázcích 4.1 a 4.2.



**Obrázek 4.1:** Vlevo mapa pokoje rodinného domu s nábytkem a předměty na podlaze a kolem stěn získaná pomocí knihovny *map\_server*. Černá barva značí stěnu, bílá volný prostor a šedá prostor mimo měřenou místnost. Tato mapa byla vygenerována jako jedna z prvních a sloužila k seznámení se s prostředím *map\_serveru*. Přibližná velikost pokoje je  $6\text{ m} \times 4,5\text{ m}$ . Vpravo reálná fotografie místnosti focena z levého dolního rohu.

Na obrázku 4.1 je vidět nedokončenost mapy hned v několika místech. V dolní a levé části mapy je tento šum způsoben předměty na zemi místnosti a tedy nemožnost laserového dálkoměru změřit vzdálenost až ke stěnám. V horní části můžeme vidět závěs a vstup na balkón, kde je vidět jak dálkoměr prosvítil skleněné dveře i závěs a tudíž namodeloval i část balkónu za dveřmi. V pravé části místnosti jsou pak vidět pozůstatky kotoučů činek ležících na zemi pokoje.

Další dvě měření jsem provedl v totožné místnosti chodby. Jedno s otevřeným prostorem schodiště a v druhém s uzavřeným přístupem ke schodišti. Na obrázcích 4.2 je vidět důležitost uzavřenosti místnosti. Dálkoměr díky odrazům měřících paprsků při nedouzavřenosti místnosti vidí velmi daleko za prostory měřené místnosti. Na levém obrázku 4.2 je vidět artefakt spodní místnosti

pod schodištěm a tečkovaně i zábradlí schodiště.



**Obrázek 4.2:** Mapy prázdné chodby rodinného domu získané pomocí knihovny *map\_server*. Černá barva značí stěnu, bílá volný prostor a šedá prostor mimo měřenou místnost. Přibližná velikost chodby je  $5\text{ m} \times 2\text{ m}$ . Vlevo otevřený přístup ke schodišti, vpravo uzavřený.

Na obrázku 4.2 vpravo je vidět místnost bez předmětu a s uzavřenými dveřmi, tedy jsou nastaveny takové podmínky, jaké mají být pro měření hluku. Výsledek stále není dostačující. Můžeme si všimnout nedotažeností některých stěn, které v navigaci robotické jednotky mohou způsobovat chyby. Zároveň je povrch celé mapy zbytečně členitý což způsobuje delší prodlevy při navigaci a výpočtu kontrolních bodů. Tento problém jsem řešil pomocí matematické aproximace takovýchto map.

## 4.2 Matematická aproximace půdorysu mapy

Nedokonalosti exportované mapy pomocí *map\_saveru* byly třeba odstranit, jelikož na znalosti půdorysu měřené místnosti stojí algoritmus pro výpočet měřících bodů. Pokud by půdorys nebyl správně změřen, mohlo by dojít k nepřesnému, či špatnému stanovení měřících bodů. Z tohoto důvodu jsem implementoval software pro extrakci a následnou aproximaci uložené mapy z *map\_saveru*.

Software pro aproximaci je založený na funkcích knihovny *OpenCV* [35]. V této knihovně můžeme najít celou řadu funkcí zaměřujících se především na počítačové vidění a zpracování obrazu. Využil jsem modulu *cv2*, který umožňuje aproximaci kontur vstupního obrázku [36], kde jako vstup použiji extrahovanou mapu místnosti. Aproximace jednotlivých kontur na vstupním obraze je vypočítána pomocí matematického algoritmu nesoucí název

Ramer–Douglas–Peucker algoritmus [37].

### 4.2.1 Ramer–Douglas–Peucker algoritmus

Tento algoritmus jsem zvolil ze dvou důvodů. Jedním z nich je zjemnění měřicí chyby dálkoměru (nekonzistentnost měřené vzdálenosti k objektu), druhým je pak snížení počtu bodů, kterým je popsána mapovaná místnost. To má za následek zjednodušení a zrychlení algoritmu pro výpočet kontrolních bodů. Obecně je algoritmus hojně využíván k účelům kartografie.

Myšlenkou tohoto algoritmu je z dané křivky zachovat takové body pro tvorbu nové křivky, aniž by odchylka vzdáleností obou křivek nepřesáhla určitou mez. Tato odchylka je měřena jakožto Hausdorffova vzdálenost [38].

Algoritmus dostane na vstupu set bodů, jejichž spojnice reprezentuje vstupní křivku. Dále je třeba určit  $\epsilon$ , které zobrazuje míru tolerance odchylky vzdálenosti výstupní křivky od vstupní. Zároveň musí platit podmínka  $\epsilon > 0$ . Na obrázku 4.4 je vidět aproximace pro hodnotu  $\epsilon = 10$  (vlevo) a pro hodnotu  $\epsilon = 1$  (vpravo).

Algoritmus je zachycen pomocí pseudokódu na obrázku 4.3. Jako první algo-

```
function DouglasPeucker(PointList[], epsilon)
  // Find the point with the maximum distance
  dmax = 0
  index = 0
  end = length(PointList)
  for i = 2 to (end - 1) {
    d = perpendicularDistance(PointList[i], Line(PointList[1], PointList[end]))
    if (d > dmax) {
      index = i
      dmax = d
    }
  }

  ResultList[] = empty;

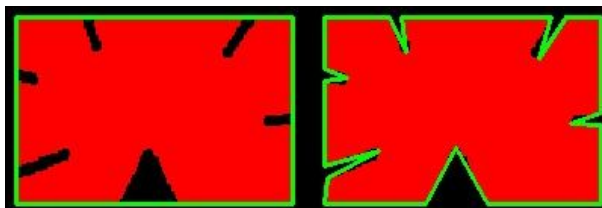
  // If max distance is greater than epsilon, recursively simplify
  if (dmax > epsilon) {
    // Recursive call
    recResults1[] = DouglasPeucker(PointList[1..index], epsilon)
    recResults2[] = DouglasPeucker(PointList[index..end], epsilon)

    // Build the result list
    ResultList[] = {recResults1[1..length(recResults1) - 1],
                    recResults2[1..length(recResults2)]}
  } else {
    ResultList[] = {PointList[1], PointList[end]}
  }
  // Return the result
  return ResultList[]
end
```

**Obrázek 4.3:** Pseudokód zobrazující implementaci Ramer-Douglas-Peucker algoritmu [37]

ritmus označí začáteční a koncový bod dané křivky k ponechání pro křivku novou. Dále algoritmus pracuje rekurzivně. Vybere bod, který je v poloviční vzdálenosti mezi počátečním a koncovým bodem. Pro přehlednost tomuto

bodů řekněme  $M$ , počátečnímu  $S$  a koncovému  $E$ .



**Obrázek 4.4:** Ilustrace aproximace pomocí Ramer–Douglas–Peucker metody pro hodnoty  $\epsilon = 10$  vlevo a pro hodnotu  $\epsilon = 1$  vpravo [36]

Dále je třeba zjistit zda mezi body  $S$  a  $M$  leží další bod. Pokud ano, tak algoritmus vypočte vzdálenost tohoto bodu od pomyslné křivky mezi body  $S$  a  $M$ . Pokud je tato vzdálenost menší než zadané  $\epsilon$ , může být vnitřní bod vynechán. Algoritmus končí po vyčerpání všech bodů dané původní křivky. Výsledkem je nový set bodů, reprezentující aproximovanou křivku, v nejhorším případě stejnou jako byla křivka vstupní.

### 4.3 Implementace komunikace hlukoměru s Arduinem a řídicím počítačem

Jedná se o jednoduchou sériovou komunikaci mezi algoritmem Arduina (soubor *arduino\_hluk.ino*) a programem pro postupné najíždění do měřících bodů (soubor *multiple.py*) řídicího počítače. Komunikace je realizována pomocí knihovny *pySerial* [39], ta slouží k přenosu zprávy za pomoci jednotlivých bitů skrze USB port řídicího počítače.

V kódu Arduina najdeme nekonečnou while sekvenci, čekající na příkaz z Python kódu řídicího počítače ke startu měření. Ten přichází v moment, kdy robotická jednotka dokončí přesun do měřícího bodu. Tuto skutečnost oznamuje Arduino zasláním bitu  $1$ . Poté co Arduino přijme tento bit, změří hodnotu  $X$  za pomoci interní funkce *analogRead(A3)*. Do argumentu funkce vkládám  $A3$  jakožto pin, ke kterému je připojen výstup hlukoměru.

$X$  ještě přímo nereprezentuje napětí na pinu  $A3$ . Arduino má k dispozici převodník s  $2^{10}$  hodnot s rozlišovací schopností  $5\text{ mV}$  na dílek hodnoty  $X$ . Napětí na pinu  $A3$  se tedy vypočítá podle rovnice 4.1.

$$U_{A3} = \frac{X \cdot 5}{2^{10}}, \quad (4.1)$$

kde  $U_{A3}$  značí napětí na pinu  $A3$  a  $X$  je hodnota získaná pomocí funkce *analogRead(A3)*.

Takto získané napětí  $U_{A3}$  dále přepočítáme na hodnotu dB okolního hluku za pomocí vztahu 4.2.

$$I = U_{A3} \cdot 100, \quad (4.2)$$

kde  $I$  značí intenzitu okolního hluku v dB a  $U_{A3}$  je napětí na pinu A3. To je násobené hodnotou 100, jelikož hlukoměr šifruje intenzitu okolního hluku do napětí v poměru 10 mV/dB.

Hodnota  $I$  je odeslána přes sériovou komunikaci, kde ji Python kód uloží k danému měřenému bodu. Zároveň se jednotka rozpohybuje směrem k dalšímu měřicímu bodu a celý proces se opakuje. Po dokončení celé měřicí sekvence jsou jednotlivé hladiny hluku vypsány spolu s měřicími body na obrazovce terminálu.

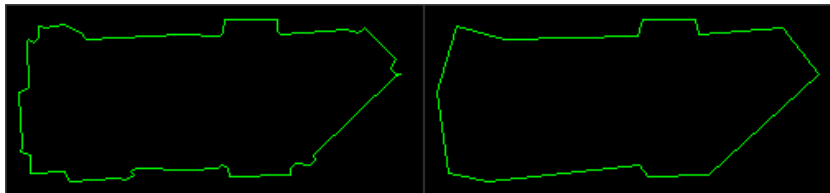
## Kapitola 5

### Testování robotické jednotky v reálných podmínkách

V této kapitole popíši, jak se chová má implementace v reálných podmínkách. Tedy zda robotická jednotka správně vytváří polygony testovaných prostor, zda z nich správně počítá měřící body, naviguje se do nich a provádí v nich měření. Dále také popíši problémy, které jsem musel řešit při nasazení robota do reálného prostředí a jak jsem tyto problémy řešil.

#### 5.1 Testy aproximační metody

První parametr, který byl třeba určit za pomoci reálných testů, bylo určení vhodného koeficientu  $\epsilon$  pro aproximační metodu. Nechal jsem aproximační algoritmus zkoušet různé hodnoty  $\epsilon$  pro různé místnosti. Zjistil jsem, že pro málo členité místnosti je rozptyl volby vhodného  $\epsilon$  veliký. Přešel jsem tedy k vykreslování aproximovaných půdorysů členitějších místností. Například na obrázcích 5.1 můžeme vidět, jak působí změna tohoto koeficientu na místnost chodby s mnoha vchody do vedlejších místností. Jak je

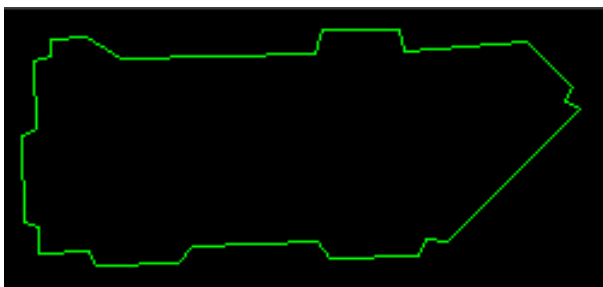


**Obrázek 5.1:** Rekonstrukce mapy chodby s rozměry  $2\text{ m} \times 5\text{ m}$  získaná pomocí aproximačního algoritmu pro koeficienty  $\epsilon = 1$  (vlevo) a  $\epsilon = 6$  (vpravo). Zelená křivka zobrazuje spojnicí bodů tvořící hranici místnosti (body možno chápat jako rohové body polygonu místnosti). Značení je stejné pro všechny obrázky zobrazující matematickou aproximaci.

patrné z obrázku 5.1 zvyšující se  $\epsilon$  má za následek větší vyhlazení nerovností půdorysu místnosti. Pokud ovšem koeficient zvolím příliš velký (na obrázku 5.1  $\epsilon = 6$ ) celá místnost se deformuje.

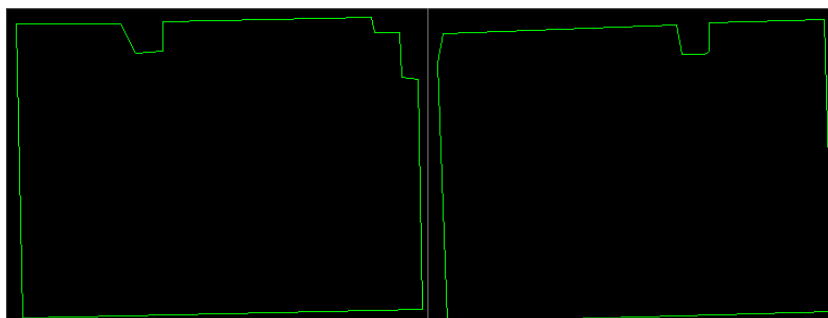
Hledal jsem proto takové  $\epsilon$ , které nedeformuje hlavní rysy místnosti a zároveň co nejvíce vyhlazuje její půdorys. Pro místnost na obrázcích 5.1 je ideální

hodnotou  $\epsilon = 3$ , viz obrázek 5.2. Tato hodnota se osvědčila i pro další měřené



**Obrázek 5.2:** Rekonstrukce mapy chodby s rozměry 2 m  $\times$  5 m pomocí aproximačního algoritmu pro  $\epsilon = 3$ .

místnosti, jak je vidět na obrázku 5.3. To způsobuje fakt, že ostatní měřené místnosti byly méně členité než místnost chodby. Pokud by bylo třeba měřit v budoucnu více členité místnosti, je žádoucí zvýšit koeficient aproximační metody. To je možné provést v kódu *contour\_approximation.py* ve složce *Plocha/codes*.



**Obrázek 5.3:** Rekonstrukce mapy školních tříd 202b (vlevo) a 202a (vpravo) s rozměry 5,5 m  $\times$  7 m získaných pomocí aproximačního algoritmu pro koeficient  $\epsilon = 3$ .

## 5.2 Testování robotické jednotky pro měření hluku

V této sekci popíši testy najíždění do cílových bodů, výpočty těchto bodů a nastavení testovaných parametrů pro odstranění některých problémů.

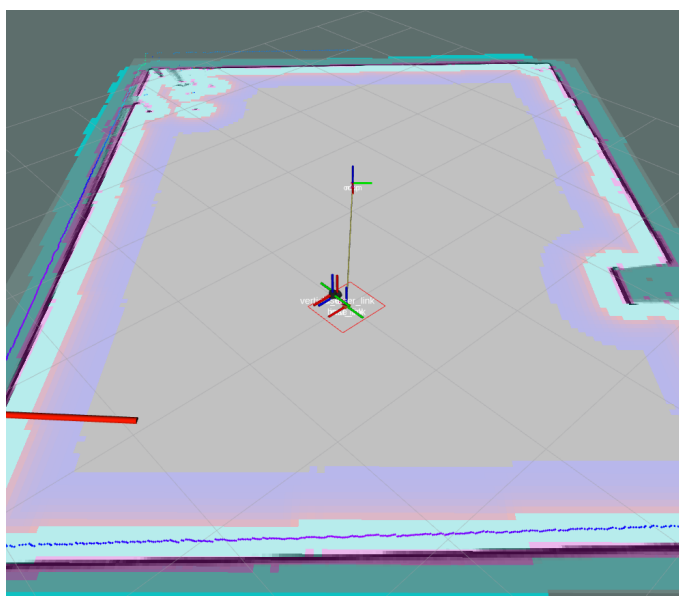
### 5.2.1 Testované místnosti

Zprvu jsem robotickou jednotku testoval v domácím prostředí. Jak je popsáno v kapitole 4, testy probíhaly v jednom z pokojů mého domu a také na chodbě s přístupem ke schodišti.

Bohužel domácí prostředí nebylo pro testy dostatečné a to z důvodu velké



členitosti místností, nábytku (který by neměl být v místnosti v době testů) a obecně malým rozměrům prostor. Ze začátku to ale byla jediná volba vzhledem k situaci s Covid-19 a obtížemi testovat ve školních podmínkách. Testování domácích místností tedy sloužilo hlavně k seznámení se s robotickou jednotkou, jejím ovládním, spouštěním měřicího procesu a k testům aproximační metody. Zároveň jsem upravil uzel Cartographer a nastavil jeho parametry tak, aby zobrazoval tzv. *safe zone* parametr, tedy vizualizace, jak daleko se může robotická jednotka přiblížit ke stěnám či překážkám v místnosti, viz obrázek 5.4.



**Obrázek 5.4:** Vizualizace měřené místnosti získaná pomocí Cartographeru – odstín modré barvy značí parametr *safe zone*.

Později mi bylo umožněno začít testovat v prostorech univerzity. Zde jsem postupně testoval ve třech místnostech. Testy jsem začal v místnosti s kobercovým povrchem, ten se ale ukázal jako nevhodný. Proto jsem se přesunul do hlavních dvou testovacích místností T2:A4-202a a T2:A4-202b (dále jen 202a a 202b), které mají povrch z linolea. Na obrázku 5.5 je zobrazena robotická jednotka v při testování v místnosti 202b.

## 5.2.2 Problémy s kobercovým povrchem místnosti

Při testování v místnosti s kobercem jsem narazil na častý problém. Ten se projevoval zastavením robotické jednotky při nízkém nastavení hodnoty rychlosti motorů. To bylo patrně z důvodu nedostatečné energie pro překonání tření kol robotické jednotky s kobercem. S tímto problémem se potýkal již



**Obrázek 5.5:** Robotická jednotka focena v průběhu měření v místnosti 202b

pan Kůrka ve své práci [10].

Tento problém je v ní řešen dvěma parametry *anti\_dead\_zones*. Jeden slouží pro jízdu dopředu (*anti\_dead\_zones\_linear\_static*) druhý pak pro otáčivý pohyb (*anti\_dead\_zones\_yaw\_static*). Konstanty těchto parametrů jsou použity k vynásobení se zadanou rychlostí motorů ještě před tím, než jsou hodnoty odeslány do desky Arduina potažmo do motorů samotných. Toto se děje jen za podmínky, kdy robotická jednotka stojí na místě (to značí slovo *static* v názvu těchto parametrů) před příkazem pohybu motorů.

Dalším problémem většího tření povrchu koberce je potřeba hnát do motorů větší proud pro rozpohybování robotické jednotky. Je tedy možné, že díky této skutečnosti se motor porouchal.

Při sledování hodnot vypisovaných deskou Arduina, nastavující rychlost motorů, jsem si všiml, že i přes nastavení vyšší hodnoty rychlosti pro levý motor než pro pravý, robotická jednotka stále zatačí na levou stranu. Robotická jednotka také nejezdila tak plynule jako dříve. Po výměně levého motoru začala jednotka správně najíždět do kontrolních bodů.

### ■ 5.2.3 Problém implementace parametrů *anti\_dead\_zones*

Po výše popsané zkušenosti, jsem se rozhodl robotickou jednotku testovat na povrchu linolea v místnosti 202b. Tato místnost byla i velmi prostorná a málo členitá, tedy ideální pro testovací účely. Před zahájením jednotlivých jízd jsem vždy odebral všechnen nábytek z místnosti, abych zajistil správné

podmínky pro měření hluku [5].

Při dalších zkouškách se ukázalo řešení pomocí parametrů *anti\_dead\_zones* jako nevhodné. Kvůli implementaci těchto zón bylo totiž třeba provádět výpočet pro měřicí body až pro vzdálenost 1,5 m od stěn a ne 0,5 m jak je uvedeno v normě [5].

Algoritmus totiž vypočítává pozici měřících bodů už na úplném okraji, tedy v zadané nejbližší vzdálenosti ke stěnám. Robotická jednotka ovšem potřebuje více manévrovacího prostoru pro správné najetí do cílového bodu nežli 0,5 m a to právě z důvodu zavedení *anti\_dead\_zones*.

Vynásobením rychlostí motorů se totiž překoná tření s podlahou, ale zároveň se jednotka zrychlí natolik, že je třeba ji nějakou dobu brzdit. Tato doba může způsobit najetí platformy do *safe zóny* (zóny do které robot nesmí najet, jelikož hrozí kolize s předměty či stěnami).

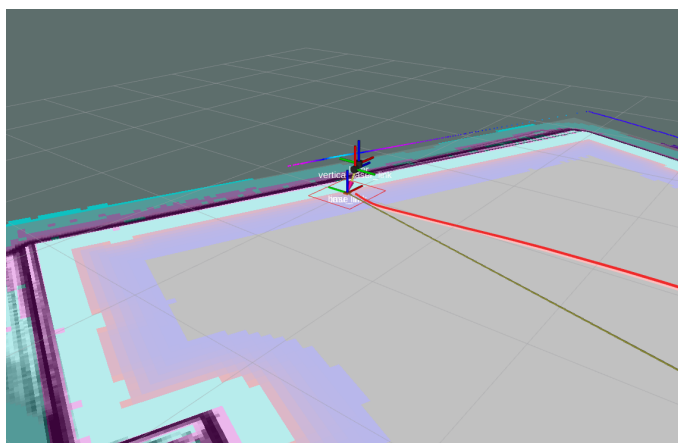
#### 5.2.4 Řešení problému kolizí robotické jednotky se stěnami

V algoritmech pro výpočet měřících bodů jsem tedy upravil hodnotu proměnné *distance* určující nejbližší možnou vzdálenost umístění kontrolního bodu ke stěně na 2 m. Tímto se vyřešil problém, kdy robotická jednotka zůstala stát moc blízko stěny a nemohla vypočítat novou trajektorii k cílovému bodu.

Na obrázku 5.6 je zachycený moment, kdy se robotická jednotka dostala do kolize se stěnou. Algoritmus pro najíždění do následujícího cílového bodu sice vypočítá trajektorii, ovšem ta nejde vypočítat tak, aby neprocházela skrze *safe zónu*. Do ní ovšem robotická jednotka nemůže najet, jelikož se již celá nachází v *safe zóně* a tudíž je uvězněná na místě.

Dalšími testy bylo zjištěno, že robotické jednotce k manévrum do kontrolních bodů na okraji místnosti stačí 1,5 m. Testy jsem provedl i pro hodnotu *distance* 1 m, ale tato vzdálenost byla pořád nedostatečná pro manévrovací prostor robotické jednotky. Stávalo se (i když méně často než pro hodnotu 0,5 m), že se robotická jednotka dostala moc blízko ke stěně (do prostoru *safe zóny*).

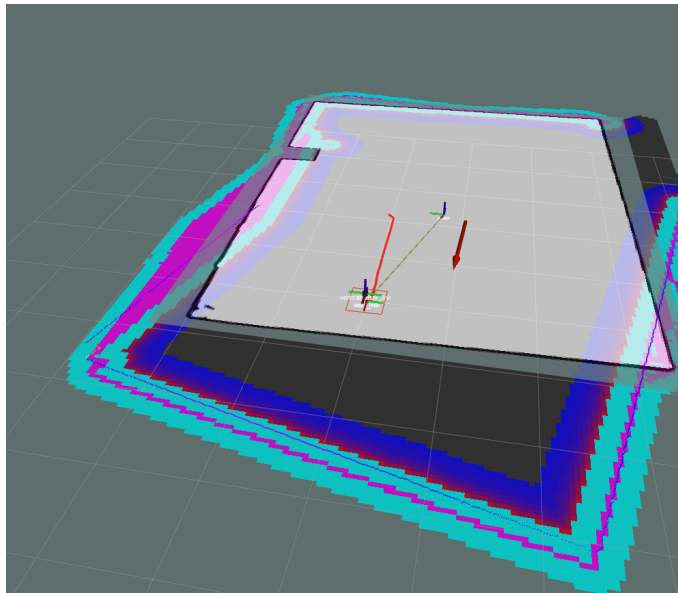
Tento problém by se dal v budoucnu vyřešit implementací plynulého rozjíždění pro překonání tření namísto řešení pomocí parametrů *anti\_dead\_zones*. Či dokonce implementací algoritmu, který by v závislosti na blízkosti k překážce volil parametr pro zrychlení motorů ze statické pozice.



**Obrázek 5.6:** Vizualizace měřené místnosti získaná pomocí Cartographeru. Souřadnicový systém v popředí obrázku zobrazuje momentální polohu robotické jednotky. Červená křivka zobrazuje vypočítanou trajektorii k zadanému cílovému bodu.

### ■ 5.2.5 Řešení problému s rozpadem mapy

Dalším v řadě problémů který se mi podařilo vyřešit je rozpad mapy Cartographeru. Rozpad je zachycený na obrázku 5.7. K tomuto problému docházelo z důvodů protáčení se kol na hřídeli upevněné v motoru.



**Obrázek 5.7:** Vizualizace měřené místnosti získaná pomocí Cartographeru. Ukázka rozpadu mapy Cartographeru po protočení hřídele kol.

Z tohoto důvodu odometrie robotické jednotky vykreslila vypočítanou pozici robotické jednotky po vykonání otáčivého pohybu. Toto vykreslení ale neod-

povídalo realitě, jelikož se robotická jednotka z důvodu prokluzu nepohnula a tedy data z LIDARu vykreslili prostor místnosti nepozměněn.

S takto rozpadlou mapou měřené místnosti bohužel nelze pokračovat v měřícím procesu, jelikož hrozí srážka se stěnou či jinou překážkou. Pokud tedy nastane tato situace je třeba celé měření opakovat. Tomuto problému lze předejít, pokud se zkontroluje dotažení šroubů obou kol k jejich hřídelím před samotným měřením.

Po zjištění příčiny tohoto problému jsem přidal ke každému kolu ještě jeden šroub držící kola k jejich hřídelím. Po této úpravě a občasnému dotažení se již tento problém neopakoval.

### ■ 5.2.6 Testy přesnosti najíždění do cílového bodu

Dalšími parametry podrobenými testování byly tzv. *goal\_tolerance* parametry. Jedná se o dva parametry *xy\_goal\_tolerance* a *yaw\_goal\_tolerance*, představující míru tolerance odchylky vzdálenosti potažmo natočení od cílového bodu v metrech. Cílem bylo snížit tyto parametry co nejvíce, aby byla zajištěna co nejvyšší přesnost měření v kontrolních bodech.

Snížením těchto parametrů jsem získával lepší přesnost nájezdu do cílových bodů za cenu zvýšení doby celého měřícího procesu. Pro nízké hodnoty totiž musela robotická jednotka najíždět do bodů opakovaně, než se jí podařilo dosáhnout požadované přesnosti.

Testy byly prováděny na obou algoritmech pro měření hluku, tedy krátkodobý pravidelný hluk a dlouhodobý stacionární hluk. Byly provedeny testy pro hodnoty tolerancí 0,2 m, 0,1 m a 0,05 m. Po zvýšení přesnosti na 0,05 m již bylo zapotřebí nespočet pokusů pro najetí robotické jednotky do každého z měřících bodů, díky čemuž se doba celého měření několikanásobně zvýšila. Pro další testy proto byla volena přesnost 0,1 m.

## ■ 5.3 Finální test v místnosti 202a

Po vyřešení všech výše zmíněných problémů jsem se přesunul do místnosti 202a pro provedení finálních testů robotické jednotky. Parametry pro finální test byly na základě výše zmíněného zvoleny jako:

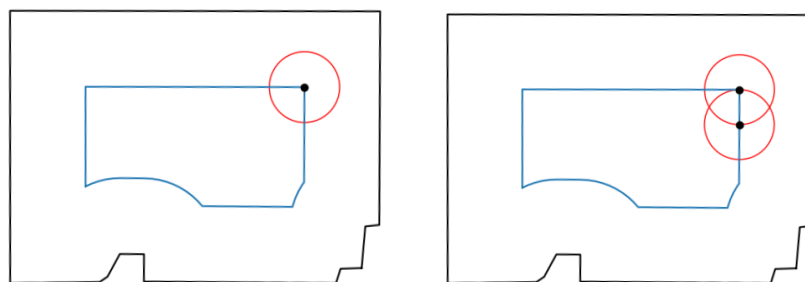
- Nejbližší vzdálenost umístění kontrolního bodu od stěny – 1,5 m
- Tolerance přesnosti najetí do kontrolního bodu – 0,1 m
- Parametr *anti\_dead\_zones\_linear\_static* – 4,1

- Parametr *anti\_dead\_zones\_yaw\_static* – 6,1

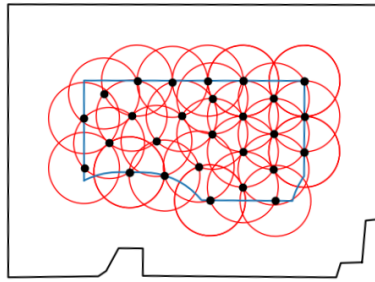
Test byl proveden pro oba algoritmy jak pro měření krátkodobého pravidelného hluku (soubor *optimalizace\_2.py*) tak pro měření dlouhodobého stacionárního hluku (soubor *maxPoints.py*). Robotická jednotka postupně projela všechny měřicí body daného algoritmu a poté se úspěšně vrátila do původní pozice s uloženými výsledky měření. Její pozice po ukončení měřicího procesu se od startovní lišila o 6 cm pro oba algoritmy.

### ■ 5.3.1 Algoritmus dlouhodobého stacionárního hluku

Algoritmus měření dlouhodobého stacionárního hluku je napsán v souboru *maxPoints.py*. Postup tohoto algoritmu začne výběrem jednoho rohového bodu dané místnosti. Kolem kterého vytvoří kružnici, na kterou umístí další kontrolní bod tak, aby se do místnosti vešlo co nejvíce kontrolních bodů. Postup algoritmu je zobrazený na obrázku 5.8, kde jsou vidět první dva kroky algoritmu. Dále na obrázku 5.9 jsou vidět všechny kontrolní body tohoto algoritmu pro místnost 202a.



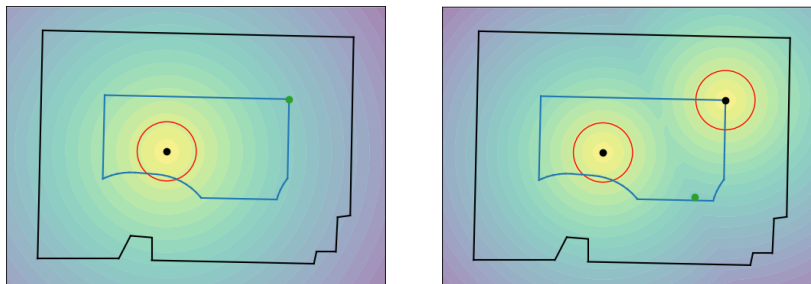
**Obrázek 5.8:** První dvě iterace algoritmu pro určení dlouhodobého stacionárního hluku místnosti 202a s rozměry 7,5 m × 5,5 m. Černá křivka – stěnu místnosti. Modrá křivka – oblast do které je možné umístit kontrolní body. Černé tečky – kontrolní body. Červené kružnice – vzdálenost, která musí být dodržena mezi jednotlivými kontrolními body. Značení je stejné pro následující vizualizaci tohoto algoritmu.



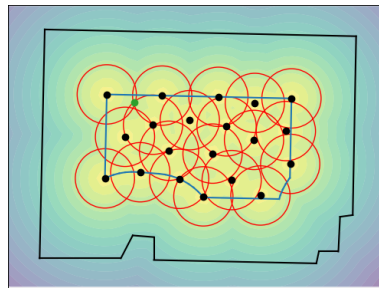
**Obrázek 5.9:** Dokončená mapa kontrolních bodů algoritmu dlouhodobého stacionárního hluku místnosti 202a s rozměry 7,5 m × 5,5 m.

### 5.3.2 Algoritmus krátkodobého pravidelného hluku

Algoritmus měření krátkodobého pravidelného hluku je napsán v souboru *optimalizace\_2.py*. Postup tohoto algoritmu začne umístěním prvního kontrolního bodu do startovní pozice robotické jednotky. Každý další bod umístí do nejvzdálenějšího možného bodu v místnosti od již určených bodů. Tímto cílí na maximální možnou plochu pokrytou v minimálním počtu kontrolních bodů. První kroky algoritmu jsou znázorněny na obrázku 5.10. Celá mapa místnosti 202a s vypočtenými kontrolními body tohoto algoritmu je na obrázku 5.11.



**Obrázek 5.10:** První dvě iterace algoritmu pro určení krátkodobého pravidelného hluku místnosti 202a s rozměry 7,5 m × 5,5 m. Černá křivka – stěnu místnosti. Modrá křivka – oblast do které je možné umístit kontrolní body. Černé tečky – kontrolní body. Červené kružnice – vzdálenost, která musí být dodržena mezi jednotlivými kontrolními body. Zelená tečka – umístění příštího bodu algoritmu. Značení je stejné pro následující vizualizaci tohoto algoritmu.



**Obrázek 5.11:** Dokončená mapa kontrolních bodů algoritmu krátkodobého pravidelného hluku místnosti 202a s rozměry 7,5 m × 5,5 m.



# Kapitola 6

## Návod k obsluze

V této kapitole shrnu všechny nutné kroky ke spuštění výpočetního algoritmu robotické jednotky. Zároveň také uvedu, jaké kroky musí být provedeny před samotným měřením za pomoci robotické jednotky. Součástí kapitoly je popis potřebných knihoven a jednotlivé příkazy pro spuštění měření. Dále je zde popsáno také umístění a funkčnost jednotlivých kódů měřících a výpočetních algoritmů i s jejich hlavními proměnnými parametry.

### 6.1 Popis řídicího softwaru robotické jednotky

Na řídicím počítači jsou všechny algoritmy rozděleny do tří základních složek. Zde uvedu cesty k těmto složkám: *BAKALARKA/PC/main*, dále *catkin\_ws* a *Plocha/codes*.

V adresáři *main* je umístěn soubor *my\_robot\_node.py* [10]. Tento soubor je zodpovědný za komunikaci s motory robotické jednotky. Lze zde nastavit parametry *anti\_dead\_zones*, které slouží k zvýšení rychlosti motoru ze statické pozice robotické jednotky pro překonání třecích sil s povrchem, na kterém se jednotka nachází. Je možné tuto hodnotu ponechat nižší pro hladké povrchy, pro hrubé je třeba hodnoty zvýšit.

V adresáři *catkin\_ws* je možné nalézt uzly ROSu: *move\_base*, Cartographer a *map\_saver*. Uzel *move\_base* je zodpovědný za výpočet trajektorie k cílovému bodu. V souboru *base\_local\_planner\_params.yaml* tohoto uzlu je možné nastavit toleranci vzdálenosti dosažení cílového bodu v metrech, parametr *xy\_goal\_tolerance*. Uzel Cartographer slouží k vizualizaci trajektorie robotické jednotky. Uzel *map\_saver* slouží k uložení mapy zachycené uzlem Cartographer.

V poslední složce *codes* je hned několik algoritmů. Je zde například soubor *contour\_approximation.py*, který je zodpovědný za matematickou aproximaci uložené mapy. V tomto souboru lze upravovat parametr  $\epsilon$  metody.

Dále se zde nachází algoritmy pro výpočet měřících bodů: *maxPoints.py* (sloužící pro měření dlouhodobého stacionárního hluku) a *optimalizace\_2.py* (sloužící pro měření krátkodobého pravidelného hluku). V obou těchto sou-



*maxPoints.py*). Ten využívá Pythonu verze 3 a výše. Pro instalaci verze 2 je třeba zadat do konzole příkaz: **`sudo apt install python2`**. Dále pro verzi 3 zadáme příkaz: **`sudo apt install python3.8`**.

### 6.2.3 Knihovny potřebné k běhu algoritmů

V tabulce 6.1 jsou uvedeny všechny knihovny, které je zapotřebí nainstalovat na řídicí počítač za účelem spuštění měřícího procesu. Zároveň jsou zde uvedené verze daných knihoven s příkazem k jejich instalaci.

**Tabulka 6.1:** Seznam knihoven potřebných pro správné fungování výpočetních algoritmů včetně návodu na jejich instalaci.

Název knihovny	Verze	Příkaz pro instalaci
<b>cycler</b>	0.10.0	<code>pip install Cycler==0.10.0</code>
<b>distro</b>	1.5.0	<code>pip install distro==1.5.0</code>
<b>future</b>	0.18.2	<code>pip install future==0.18.2</code>
<b>kiwisolver</b>	1.1.0	<code>pip install kiwisolver==1.1.0</code>
<b>matplotlib</b>	2.2.5	<code>pip install matplotlib==2.2.5</code>
<b>numpy</b>	1.16.6	<code>pip install numpy==1.16.6</code>
<b>opencv-python</b>	4.2.0.32	<code>pip install opencv-python==4.2.0.32</code>
<b>packaging</b>	20.9	<code>pip install packaging==20.9</code>
<b>pyarsing</b>	2.4.7	<code>pip install pyarsing==2.4.7</code>
<b>pyserial</b>	3.4	<code>pip install pyserial==3.4</code>
<b>python-dateutil</b>	2.8.1	<code>pip install python-dateutil==2.8.1</code>
<b>pytz</b>	2021.1	<code>pip install pytz==2021.1</code>
<b>PyYAML</b>	5.4.1	<code>pip install PyYAML==5.4.1</code>
<b>scikit-build</b>	0.11.1	<code>pip install scikit-build==0.11.1</code>
<b>scipy</b>	1.2.3	<code>pip install scipy==1.2.3</code>
<b>setuptools</b>	44.1.1	<code>pip install setuptools==44.1.1</code>
<b>Shapely</b>	1.7.1	<code>pip install Shapely==1.7.1</code>
<b>six</b>	1.15.0	<code>pip install six==1.15.0</code>
<b>subprocess32</b>	3.5.4	<code>pip install subprocess32==3.5.4</code>
<b>tqdm</b>	4.60.0	<code>pip install tqdm==4.60.0</code>
<b>tripy</b>	1.0.0	<code>pip install tripy==1.0.0</code>
<b>wheel</b>	0.36.2	<code>pip install wheel==0.36.2</code>

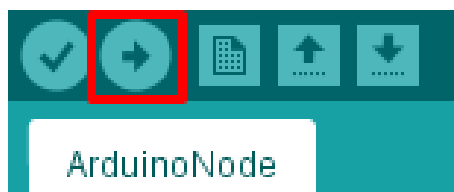
### 6.2.4 Nahrávání řídicích algoritmů do desek Arduino UNO

Desky Arduino disponují pamětí pro jeden řídicí algoritmus. Zůstává v nich uložený poslední nahraný kód. Kroky této sekce tedy není třeba vykonávat před každým měřením, ale jen za účelem určité změny řídicích algoritmů desek.

Na řídicím počítači se algoritmy pro Arduina nachází ve složce *Plocha/Arduino*.

Je zde kód *ArduinoNode.ino* sloužící pro Arduino řídicí motory jednotky. Dále tu najdeme soubor *arduino\_hluk.ino*, který slouží jakožto řídicí algoritmus pro desku Arduina komunikující s hlukoměrem.

Po jejich případné úpravě je třeba spustit aplikaci Arduino IDE, která je již nainstalovaná na řídicím počítači. Dále je třeba připojit Arduino k řídicímu počítači pomocí USB kabelu. V aplikaci pak po provedení vyžadované úpravy kódu kliknout na tlačítko nahrát zvýrazněné na obrázku 6.1



**Obrázek 6.1:** Ovládací lišta aplikace Arduino IDE s červeně označeným tlačítkem pro nahrání kódu do desky.

### 6.2.5 Identifikace portů měřících senzorů robotické jednotky

Před spuštěním algoritmů desek Arduina či uzlu *urg\_node* je třeba identifikovat port pro danou desku či měřící přístroj. Pro identifikaci portů měřících senzorů robotické jednotky je třeba postupovat dle následujících kroků:

1. Zapnout robotickou jednotku pomocí spínačů umístěných na její boční straně. Všechny tři spínače přepneme do polohy I, zapnuto.
2. Položit řídicí počítač na robotickou jednotku a připojit černý USB kabel pro sběr dat z laserového dálkoměru.
3. Otevřít terminál a zadat příkaz: ***sudo chmod a+rw /dev/ttyACM0***. Tento příkaz je nutné potvrdit heslem administrátora řídicího počítače.
4. Připojit stříbrně obalené USB komunikující s Arduinem Unem pro řízení motorů.
5. Zadat příkaz: ***sudo chmod a+rw /dev/ttyACM1***.
6. Připojit černé USB komunikující s Arduinem Unem pro sběr dat z hlukoměru.
7. Zadat příkaz: ***sudo chmod a+rw /dev/ttyACM2***.

## 6.3 Spuštění samotného měřicího procesu

Nejprve je třeba spustit jednotlivé uzly ROS, poté je třeba zmapovat měřenou místnost, uložit mapu a nakonec spustit aproximační, výpočetní a najížděcí algoritmus.

### 6.3.1 Spuštění jednotlivých uzlů ROS

Uzly ROS se spustí za pomoci následujících kroků:

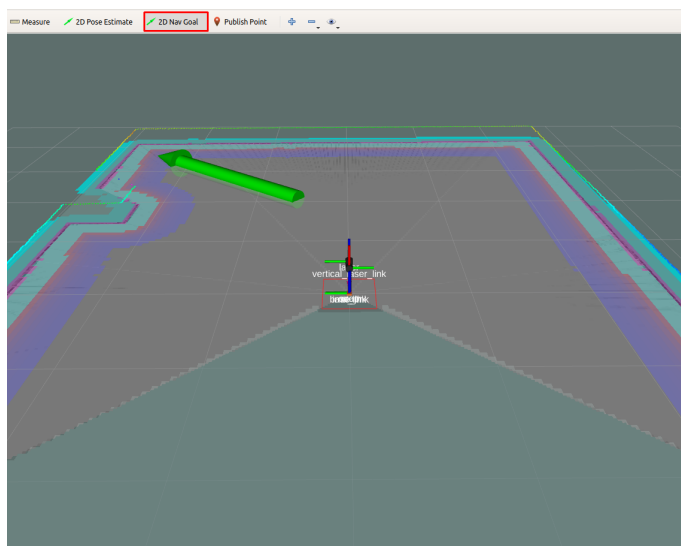
1. Spustit hlavní uzel ROS pomocí příkazu:  
*roscore.*
2. Spustit uzel zodpovědný za předávání dat mezi laserovým dálkoměrem a řídicím počítačem pomocí příkazu:  
*roslaunch urg\_node urg\_node.*
3. Navigovat se do složky uzlu *move base* pomocí příkazu:  
*cd catkin\_ws/move\_base.*
4. Spustit uzel pro plánování cesty robotické jednotky pomocí příkazu:  
*roslaunch my\_move\_base.launch.*
5. Navigovat se do složky uzlu *my robot node* pomocí příkazu:  
*cd BAKALARKA/PC/main.*
6. Spustit uzel *my\_robot\_node*, který je zodpovědný za komunikaci s Arduinem pomocí příkazu:  
*python my\_robot\_node.py.*
7. Navigovat se do složky uzlu *Cartographer* pomocí příkazu:  
*cd catkin\_ws/cartographer.*
8. Položit robotickou jednotku do středu měřené místnosti.
9. Spustit uzel *Cartographer*, který vizualizuje mapu a plánovanou trasu robotické jednotky pomocí příkazu:  
*roslaunch my\_backpack\_2d.launch*

### 6.3.2 Uložení mapy a spuštění výpočetních algoritmů

Nyní je třeba zmapovat měřenou místnost tak, aby bylo možné uložit mapu místnosti, která je potřebná pro další výpočty. Tedy nařídit robotické jednotce

najet do několika bodů po místnosti v závislosti na jejím půdorysu tak, aby byla získána data z laserového dálkoměru pro celou místnost.

Příkaz pro najetí do zvoleného bodu je předán robotické jednotce pomocí tlačítka *2D navigation goal*, které se nachází v horní liště rozhraní viz obrázek 6.2. Po stisknutí tlačítka je třeba kliknout do prostoru místnosti pokud je



**Obrázek 6.2:** Vizualizace měřené místnosti před procesem jejího mapování získaná díky uzlu Cartographer. V horní liště je červeně zvýrazněno tlačítko pro najetí do zvoleného bodu. Zelená šipka zobrazující výsledný směr natočení robotické jednotky.

stisk přidržen, je možné vybrat i natočení robotické jednotky ve zvoleném cílovém bodě.

Poté co je dokončeno mapování měřené místnosti je třeba uložit mapu Cartographeru a spustit výpočetní a najížděcí algoritmus. To je provedeno následujícími kroky:

1. Navigovat se do složky s kódy pro výpočet bodů pomocí příkazu:  
*cd Plocha/codes.*
2. Uložit mapu v této složce za pomoci příkazu:  
*roslun map\_server map\_saver -f mymap.*
3. Spustit výpočet aproximace uložené mapy pomocí příkazu:  
*python contour\_approximation.*
4. Spustit výpočet kontrolních bodů pro aproximovanou mapu pomocí jednoho z příkazů:  
*python optimalizace\_2.py, python maxPoints.py.*

5. Spustit algoritmus najíždění robotické jednotky do vypočtených bodů pomocí příkazu:  
*python multiple.py.*
6. Počkat, než robotická jednotka doměří místnost a vrátí se zpět na původní pozici.

Experimentální postup měření hluku za pomoci robotické jednotky lze shrnout následovně. Z dat LIDARU software robotické jednotky vypočte polygon, který je dále použit pro měření. V dalším kroku software vypočte kontrolní body pro měření hluku po místnosti pro dva uvedené účely – krátkodobý pravidelný hluk a dlouhodobý stacionární hluk.





# Kapitola 7

## Závěr

Cílem mé práce bylo rozšířit robotickou jednotku jak po softwarové tak hardwarové stránce, aby byla schopná měřit krátkodobý intenzivní hluk a dlouhodobý stacionární hluk v interiéru. Celý proces měření je rozdělen do několika kroků. Prvním krokem je zmapování měřené místnosti. Následuje výpočet kontrolních bodů pro daný účel z extrahované mapy půdorysu místnosti. Nakonec robotická jednotka v sekvenci najede do všech měřících bodů, kde odečte hodnotu z hlukoměru a uloží ji k danému bodu. K dosažení tohoto cíle bylo třeba seznámit se s robotickou jednotkou, jejími algoritmy, vymyslet jejich propojení a doimplementovat chybějící části.

Prvním krokem v mé práci bylo seznámení se s metodikou měření hluku. To jsem učinil vytvořením rešerše na toto téma viz kapitola 2. Kde jsem také popsal celý proces měření.

V kapitole 4 jsem se zabýval extrakcí půdorysu mapy měřené místnosti. Ta je realizována pomocí ROS knihovny *map\_server* a dále mou implementací aproximační metody. Algoritmus této metody bere jako vstup uloženou mapu a za pomoci Ramer–Douglas–Peucker metody ji zjednoduší pro účely výpočtu měřících bodů.

Dále bylo třeba přidat na robotickou jednotku hlukoměr pro snímání intenzity hladiny okolního hluku. Ten jsem propojil s deskou Arduina UNO za pomoci modifikovaného audio kabelu, toto je zpracováno v kapitole 3. Dále bylo třeba napsat algoritmus sériové komunikace mezi deskou a řídicím počítačem (viz kapitola 4). Pomocí této komunikace vývojová deska posílá do řídicího počítače data o hladině hluku v dB, které vypočte na základě změny napětí na jeho analogovém pinu.

Těžištěm mé práce bylo testování robotické jednotky v reálných místnostech a testování její funkčnosti. Zprvu se objevilo několik problémů, díky kterým robotická jednotka nefungovala správně. Postupem času se mi ale podařilo všechny tyto nesnáze eliminovat a připravit tak jednotku na finální testování. Cenou za řešení těchto překážek ovšem bylo posunutí výpočetních bodů ze vzdálenosti 0,5 m od stěny (jak udává norma [5]) na hodnotu 1,5 m.

Finální test, popsáný v kapitole 5, proběhl bez komplikací pro oba algoritmy: krátkodobý intenzivní hluk a dlouhodobý stacionární hluk. Robotická jednotka správně změřila půdorys místnosti, který následně uložila pro další zpracování. Algoritmus vypočetl aproximaci a určil všechny kontrolní body v měřicí oblasti. Robotická platforma postupně najížděla do všech určených bodů a v nich prováděla měření hladiny hluku. Tento test ověřil správnou funkčnost implementace a splnil hlavní cíl mé práce.

V závěrečné části mé práce jsem vytvořil návod pro další uživatele robotické jednotky. V návodu je popsán průběh samotného měření a nutná údržba robotické jednotky. Dále jsem do kapitoly 6 zapracoval také popis jednotlivých řídicích algoritmů s vysvětlením jejich základních parametrů.

Závěrem uvedu náměty pro budoucí práci na robotické jednotce. Největším problémem robotické jednotky bylo násobení hodnoty rychlosti motorů při rozjezdu. Lepším řešením by bylo do budoucna implementovat plynulou rozjíždku robotické jednotky. Dalším vylepšením, které by vedlo k úplné automatizaci procesu je napsat algoritmus pro automatické mapování místnosti.

# Příloha A

## Literatura

- [1] International Organization for Standardization. 1996-2:2017 acoustics - description, measurement and assessment of environmental noise - part 2: Determination of sound pressure levels, 2017.
- [2] Vláda ČR. „nařízení vlády č. 217/2016 sb.“, 2016. Dostupné z: <https://www.zakonyprolidi.cz/cs/2016-217/zneni-20160730>.
- [3] Vláda ČR. „nařízení vlády č. 9/2002 sb.“, 2002. Dostupné z: <https://www.zakonyprolidi.cz/cs/2002-9>.
- [4] Vláda ČR. „vyhláška č. 523/2006 sb.“, 2006. Dostupné z: <https://www.zakonyprolidi.cz/cs/2006-523>.
- [5] Czech Standards Institute. ČSN 1996-1: Acoustics - Description, measurement and assessment of environmental noise - Part 1: Basic quantities and assessment procedures. *Czech Standards Institute*, 2017.
- [6] Ministerstvo zdravotnictví-hlavní hygienik České Republiky. Metodický návod pro měření a hodnocení hluku v mimopracovním prostředí [online]. Praha: 2001, 11. 12. 2001 [cit. 2021-04-05]. č. j. Technical report, HEM-300-11.12. 01-34065. Dostupný z [www:< http://www.nrl.cz/metodika . . .](http://www.nrl.cz/metodika)
- [7] Vláda ČR. „zákon č. 258/2000 sb.“, 2000. Dostupné z: <https://www.zakonyprolidi.cz/cs/2000-258>.
- [8] Vláda ČR. „nařízení vlády č. 272/2011 sb.“, 2011. Dostupné z: <https://www.zakonyprolidi.cz/cs/2011-272>.
- [9] Tomáš Drábek, Vladimír Petřík, and Jan Holub. Automatic control points computation for the acoustic noise level audits. In *17th IMEKO TC 10 and EUROLAB Virtual Conference: “Global Trends in Testing, Diagnostics & Inspection for 2030”*, pages 374–379. IMEKO TC10, 2020.



- [20] Lukáš Macháček. Měření a hodnocení hlukové zátěže na studentských kolejích. Bakalářská práce, Univerzita Karlova. 1. lékařská fakulta, 2008.
- [21] Sophie Pujol, Marc Berthillier, Jérôme Defrance, Joseph Lardies, J-P Levain, Rémy Petit, Hélène Houot, and Frédéric Mauny. Indoor noise exposure at home: a field study in the family of urban schoolchildren. *Indoor air*, 24(5):511–520, 2014.
- [22] G Graziuso, M Grimaldi, S Mancini, J Quartieri, and C Guarnaccia. Crowdsourcing data for the elaboration of noise maps: a methodological proposal. In *Journal of Physics: Conference Series*, volume 1603, page 012030. IOP Publishing, 2020.
- [23] Marina Bonomolo, Patrizia Ribino, Carmelo Lodato, and Gianpaolo Vitale. Post occupancy evaluation and environmental parameters monitoring by a humanoid robot. In *2019 IEEE International Conference on Environment and Electrical Engineering and 2019 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe)*, pages 1–6. IEEE, 2019.
- [24] Eric Martinson and Ronald C Arkin. Noise maps for acoustically sensitive navigation. In *Mobile Robots XVII*, volume 5609, pages 50–60. International Society for Optics and Photonics, 2004.
- [25] Petr Jedlička. Noise meter and sound detector. Bakalářská práce, Vysoké učení technické v Brně, 2017.
- [26] Drábek Tomáš. Robotická jednotka pro měření osvětlenosti v interiérech v rovnoměrné optimalizované síti kontrolních bodů. Diplomová práce, České vysoké učení technické v Praze. Výpočetní a informační centrum., 2014.
- [27] „EMG30 – Mounting bracket and wheel specification“ [online]. Dostupné z: <https://www.robot-electronics.co.uk/htm/emg30.htm>.
- [28] „MD25 - Dual 12volt 2.8amp H Bridge Motor Drive“ [online]. Dostupné z: <https://www.robot-electronics.co.uk/htm/md25tech.htm>.
- [29] „Arduino UNO.“ [online]. Dostupné z: <https://store.arduino.cc/arduino-uno-rev3>.

- [30] „Olověný akumulátor shimastu npg8-12, 12v 8ah“ [online]. Dostupné z: <https://www.emerx.cz/oloveny-akumulator-shimastu-npg8-12-12v-8ah.html>.
- [31] LTD. Hokuyo automatic CO. „Hokuyo UTM-30LX Scanning Laser Rangefinder“ [online]. Dostupné z: <https://www.hokuyo-aut.jp/search/single.php?serial=169>.
- [32] Uni-Trend Technologies. „UT351 Sound Level Meter“ [online]. Dostupné z: [https://www.uni-trend.com/meters/html/product/Environmental/Environmental\\_Tester/UT350/UT351.html](https://www.uni-trend.com/meters/html/product/Environmental/Environmental_Tester/UT350/UT351.html).
- [33] „Robot Operating System.“ [online]. Dostupné z: <http://wiki.ros.org/Documentation>.
- [34] „map\_server.“ [online]. Dostupné z: [http://wiki.ros.org/map\\_server](http://wiki.ros.org/map_server).
- [35] „OpenCV“ [online]. Dostupné z: <https://docs.opencv.org/master/index.html>.
- [36] „cv2 – Contour approximation“ [online]. Dostupné z: [https://docs.opencv.org/master/dd/d49/tutorial\\_py\\_contour\\_features.html](https://docs.opencv.org/master/dd/d49/tutorial_py_contour_features.html).
- [37] Non-parametric Ramer-Douglas-Peucker. „Ramer–Douglas–Peucker algorithm. 1972.
- [38] „Hausdorffova vzdálenost“ [online]. Dostupné z: [https://worddisk.com/wiki/Hausdorff\\_distance/](https://worddisk.com/wiki/Hausdorff_distance/).
- [39] „pySerial“ [online]. Dostupné z: <https://pyserial.readthedocs.io/en/latest/pyserial.html>.