

Bakalářská práce



České  
vysoké  
učení technické  
v Praze

**F3**

Fakulta elektrotechnická  
Katedra kybernetiky

## Mapování terénu pozemním robotem

**Jiří Koktan**

Vedoucí: Ing. Jan Chudoba  
Obor: Kybernetika a robotika  
Studijní program: Robotika  
Květen 2017



## Poděkování

Chtěl bych poděkovat vedoucímu práce Ing. Janu Chudobovi za podporu a ochotu až do samého konce.

## Prohlášení

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

V Praze, dne .....

.....  
podpis autora práce

## Abstrakt

Tato práce se zabývá mapováním a uchováváním informací o okolním terénu pomocí pozemního robotu. Práce nejdříve pojednává o používaných způsobech pro pozemní mapování autonomními roboty. Dále se zabývá programovou částí, v které je vytvořena v systému IMRnav metoda pro mapování terénu pozemním robotem. Tato metoda je následně podrobena testování v simulačním prostředí MORSE. Implementace metody je realizována v jazyce C++. Dále projekt směřuje k testování na reálných robotech ve venkovním prostředí, kde robot podle naměřených informací o okolním terénu bude vybírat nejvhodnější trajektorii pohybu.

**Klíčová slova:** pozemní robot, mapování, 3D mapa

**Vedoucí:** Ing. Jan Chudoba

## Abstract

This thesis focuses on terrain mapping and information classification using ground mobile robot. The thesis presents the world methods used for terrain mapping using autonomous robots. In the next part there is a terrain mapping method created using ground robot in the IMRnav system. This method is then subjected to testing in the MORSE simulation environment. The implementation of the method is realized in C++. In addition, the project is aimed at real-world robot testing, where the robot will choose the most appropriate trajectory of motion based on terrain information.

**Keywords:** ground robot, terrain mapping

**Title translation:** Robot terrain mapping

## Obsah

<b>1 Úvod</b>	<b>1</b>		
1.1 Senzory	2		
1.2 Mapování	3		
<b>2 Metody</b>	<b>5</b>		
2.1 Uskutečněné projekty	5		
2.1.1 Autonomous Terrain Mapping and Classification Using Hidden Markov	5		
2.1.2 An Efficient Extension to Elevation Maps for Outdoor Terrain Mapping and Loop Closing	6		
2.1.3 Stanley: The Robot that Won the DARPA Grand Challenge	7		
2.1.4 A Campaign in Autonomous Mine Mapping	8		
2.1.5 Shrnutí projektů	9		
2.2 Vlastní metoda	10		
<b>3 Implementace</b>	<b>13</b>		
3.1 Simulátor	13		
3.1.1 Senzory simulátoru	13		
3.1.2 Modely simulátoru	14		
3.1.3 Prostředí simulátoru	15		
3.1.4 Běh simulátoru	16		
3.2 Implementace kodu	16		
3.2.1 Zpracování laserových dat	18		
3.2.2 Odometrie	21		
3.2.3 Mapa	23		
<b>4 Experiment</b>	<b>25</b>		
4.1 Experiment mapování terénu	25		
4.1.1 Popis experimentů	25		
4.1.2 Výsledky experimentu	26		
4.2 Shrnutí experimentů	31		
<b>5 Závěr</b>	<b>33</b>		
<b>Literatura</b>	<b>35</b>		
<b>Zadání práce</b>	<b>39</b>		

## Obrázky

1.1 Senzorická mapa budovy reprezentovaná dvourozměrnou mřížkou obsazenosti[19] . . . . .	3	4.1 První část experimentu . . . . .	27
1.2 Příklady topologických map . . . . .	4	4.2 Druhá část experimentu . . . . .	28
2.1 Porovnání prostředí robotu a vytvořené mapy [12] . . . . .	6	4.3 Třetí část experimentu . . . . .	29
2.2 Výsledky projektu Mapování venkovního terénu [14] . . . . .	7	4.4 Čtvrtá část experimentu . . . . .	30
2.3 Ukázky z projektu DAPRA [15] . . . . .	8		
2.4 Ukázky z projektu Mine mapping [16] . . . . .	9		
3.1 Sensory simulátoru MORSE . . . . .	14		
3.2 Robotické modely simulátoru MORSE . . . . .	15		
3.3 Prostředí simulátoru MORSE . . . . .	16		
3.4 Struktura systému IMRnav [27] . . . . .	17		
3.5 Struktura modulu realizující metodu mapování . . . . .	18		
3.6 Zobrazení veličin pro výpočet . . . . .	19		
3.7 Znázornění pohybu diferenciálního vozidla a měřené veličiny [22] . . . . .	22		

## Tabulky

4.1 Parametry laserového senzoru . .	26
--------------------------------------	----







# Kapitola 1

## Úvod

V dnešní době robotického rozvoje je snaha usnadnit těžkou a opakovanou práci člověku pomocí robotických pomocníků. V průmyslovém odvětví se většinou jedná o dopravníky, manipulátory a mechanická ramena, která jsou často součástí montážních linek. Tito robotičtí pomocníci jsou řízeni jednoduchými algoritmy přesně definujícími jejich činnost a možnosti jejich rozhodování jsou značně omezeny na několik stavů.

Dále se může jednat o autonomní roboty, které je možné potkat na každém kroku. Může se jednat o vysávací robotické pomocníky, robotické sekačky na trávu nebo třeba o nezávislé robotické průvodce v muzeích. Dále se může jednat o roboty pro mapování nebezpečných nebo špatně přístupných prostor a dokonce automaticky řízená auta.

Autonomní roboty jsou schopny dlouhodobé práce bez zásahu člověka, samostatného pohybu v pracovním prostředí a vyvarování se kritickým a nebezpečným situacím. Aby byly tyto podmínky dodrženy, je nutné shromáždit dostatečné množství informací o pracovním okolí robotu. Analýza pracovního okolí a mapování prostoru je jedním ze základních prvků mobilní robotiky. Veškerá autonomní činnost robotu je závislá na znalosti přesné polohy robotu a vhodné reprezentaci pracovního prostředí. Úkolem robotu může být změna stavu prostředí např. manipulací s předměty nebo pouhý bezkolizní pohyb v daném prostoru.

Tyto autonomní roboty získávají údaje o svém okolí a své poloze výhradně pomocí senzorických systémů. Data získaná těmito systémy popisující skutečný stav prostředí jsou transformována na vnitřní reprezentaci okolí neboli

mapu. Kvalitní mapa je základem pro autonomní řízení robota. Konfigurace senzorového systému a typ reprezentace mapy je specifický pro jednotlivé metody mapování, které budou uvedeny v další kapitole.

Tato práce se bude zabývat návrhem a implementací metody mapování okolního terénu mobilním robotem. Přesněji tvorbou 3D mapy pro bezkolizní autonomní navigaci robota v okolním prostředí za použití rozmítaných laserových dálkoměrů a inerciálních senzorů. Metoda bude realizována jako kód v jazyce C++ a následně otestována v simulačním prostředí.

## 1.1 Senzory

K získávání povědomí o svém okolí je robot vybaven různými senzory, z jejichž dat získává důležité informace. Senzory jsou děleny na vnitřní/ vnější a aktivní/pasivní [2].

Vnitřní senzory měří výhradně hodnoty vnitřního systému, tedy robota. Například to jsou senzory pro snímání rychlosti motoru, zatížení koleček, úhly ramen, napětí na akumulátoru. Vnější senzory dávají informace z pracovního prostředí robota. Například to jsou senzory pro snímání vzdálenosti, obrazu, zvuku, světla. Aktivní senzory interagují s okolním prostředím a měří reakce prostředí. Například to jsou laserové a ultrasonické senzory. Pasivní senzory měří pouze hodnoty vstupující do senzoru z okolního prostředí. Například to jsou kamery, mikrofony, teploměry.

Pro mapování okolí je nejvhodnější používat senzory schopné detekovat překážky a nerovnosti. K tomu jsou vhodné senzory ultrazvukové a laserové. Laserové senzory vzdálenosti jsou velice přesné, vzdálenost několika desítek metrů určí na milimetry. Jejich hlavní nevýhodou je bohužel vysoká cena. Ultrazvukové senzory jsou sice levnější, ale přesnost i dosah jsou výrazně nižší než u laserových senzorů a navíc je měření zatíženo šumem. Dále je možné použít obrazové snímače, kamery, které poskytují velké množství informací. V této práci je zadáno použít laserových dálkoměrů.

Pro lokalizaci robota je vhodný GPS systém, který je dostatečně přesný (v rádech metrů). Jeho hlavní nevýhodou je možnost absence signálu, k tomu dochází především v budovách, podzemí a zalesněných oblastech. Dále je ještě možné použít například triangulaci, kde dochází k odhadu pozice podle vzdálenosti od orientačních bodů. Velice vhodná je také odometrie, kdy je měřeno otáčení koleček a z toho následně dopočítávána pozice robota, ta

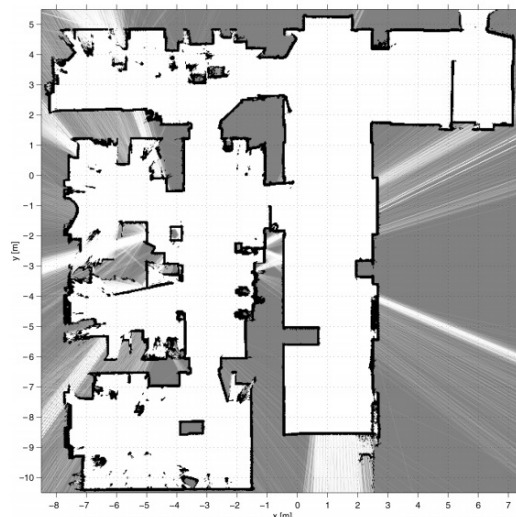
se bohužel může lišit od skutečné. Podobný problém má inerciální navigace, která spoléhá na integraci podélných a úhlových zrychlení.

## 1.2 Mapování

Dalším důležitým krokem je mapování. Mapování je metoda ukládání důležitých informací ze sensorů do vhodně reprezentovaného systému. Mapa může být reprezentována různými způsoby, například Sensorická mapa, Topologická mapa a Geometrická mapa [19].

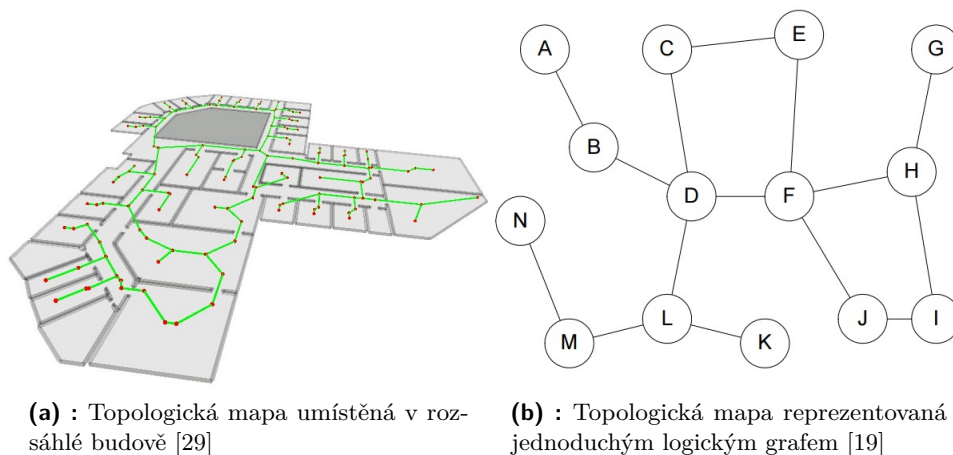
Senzorická mapa obsahuje přímá data ze sensorů ukládaná do mřížky. Pracovní prostor je rozdělen do jednotlivých buněk, jejichž hodnota určuje informaci o terénu. Velikostí mřížky je určen rozměr a rozlišení vytvořené mapy. Sensorické mapy jsou velice oblíbené díky jejich jednoduchosti, snadnému přístupu a práci s uloženými daty.

Příklad sensorické mapy ve formě mřížky obsazenosti z projektu [19] je na obrázku 1.1. Světlé plochy v mřížce značí volný prostor, černé oblasti definují hranice překážek a šedý prostor je neznámá oblast. Světlé paprskové průniky směrem k okrajům mřížky jsou způsobeny chybnými měřeními, které byly během tvorby mřížky nahrazeny nekonečnou vzdáleností. Tato měření odpovídají situaci, kdy laserovým scannerem nebyl detekován návrat vyslaného paprsku [20].



**Obrázek 1.1:** Sensorická mapa budovy reprezentovaná dvourozměrnou mřížkou obsazenosti[19]

Topologická mapa reprezentuje pracovní prostředí robotu jako logický graf. Jednotlivá místa (místnosti, pozice) jsou znázorněna jako uzly grafu a mohou obsahovat informace o daném místě. Tato jednotlivá místa jsou spojena cestami, které popisují způsob pohybu mezi jednotlivými uzly. Na obrázku 1.2 jsou ukázkové příklady topologických map.



**Obrázek 1.2:** Příklady topologických map

Geometrická mapa je složena pomocí geometrických primitiv, neboli úseček a polygonů. Jednotlivá tato primitiva jsou spojována do samostatných objektů. Tato reprezentace mapy je vhodná při zadávání známého prostředí robotu, například elektronické plány budov. Nevýhodou této mapy bývá její problematická tvorba, úprava a přiřazení sensorových dat.

Pro reprezentaci mapy obecně platí následující vztahy[2]:

- Přesnost a kvalita mapy musí odpovídat přesnosti, s kterou robot musí postupovat, aby dosáhl požadovaných výsledků.
- Přesnost a kvalita mapy by měla být stejná jako přesnost a typ naměřených dat ze senzorů robotu.
- Složitá reprezentace mapy má přímý dopad na složitost výpočtů a funkcí souvisejících s mapou.

## Kapitola 2

### Metody

#### 2.1 Uskutečněné projekty

V této části jsou vybrány některé z nastudovaných projektů zabývající se problematikou mapování pozemního terénu pomocí autonomních mobilních robotů.

##### 2.1.1 Autonomous Terrain Mapping and Classification Using Hidden Markov

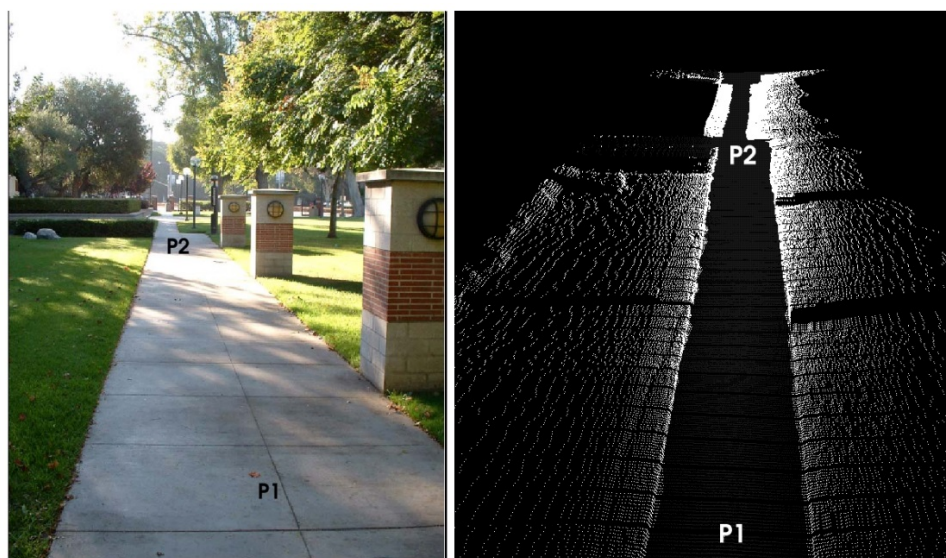
Uvedený projekt [12], který proběhl na Univerzitě ve Washingtonu v roce 2005, má podobnou senzorovou konfiguraci jako tato práce. Autonomní mapování terénu je realizováno pomocí laserového dálkoměru umístěného v přední části robotu s náklonem 45 stupňů. Jedoucí robot vytváří mrak(cloud) bodů, z kterých je sestavena 3D mapa.

Pozice robotu je určena pomocí odometrie a IMU(Inertial Measurement Unit) senzoru. Samotná odometrie při větší dráze zavádí příliš velkou chybu, ta se ve venkovním prostředí eliminuje pomocí dodatečných senzorů polohy, například GPS.

Vlastní 3D mapa je reprezentována mřížkou a následně je každá buňka

klasifikována na sjízdnu nebo nesjízdnu. K eliminaci chyb a nepřesností v mapě způsobených šumem senzorů je využíván HMM (Hidden Markov Modeling) [13], velice silný statistický algoritmus.

Alogritmus byl testován na dvou mobilních platformách, byl to Pioneer AT a Segway RMP. Výsledky provedených experimentů jsou vyobrazeny na obrázku 2.1. Robot při testování v městském parku namapoval pěší chodník a okolní terén.

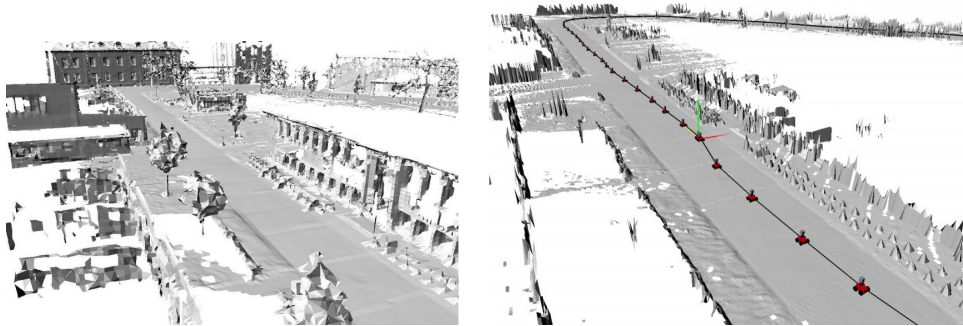


Obrázek 2.1: Porovnání prostředí robotu a vytvořené mapy [12]

### ■ 2.1.2 An Efficient Extension to Elevation Maps for Outdoor Terrain Mapping and Loop Closing

Roku 2007 v Německu na Freiburgské univerzitě se uskutečnil experiment mapování terénu [14]. Ve výsledku byl robot schopen vytvořit rozsáhlou 3D mapu s velikostí několika obytných bloků a obrazem i vícepatrových budov, viz obrázek 2.2.

Jako robotický model byl použit Pioneer II AT vybavený nakláněcím laserovým dálkoměrem SICK LMS. Laser se během jízdy robotu naklání a měří vzdálenosti objektů v různých výškách. Jednotlivá naměřená data jsou skládána do 3D mapy. Tato mapa je složitě softwarově upravována do polynomiální reprezentace celého okolí. Pozice robotu je určována pomocí odometrie a laserových dálkoměrů.



(a) : Výsledná mapa vytvořená v obytné části města

(b) : Vytvořená mapa cesty s jednotlivými pozicemi robotu

**Obrázek 2.2:** Výsledky projektu Mapování venkovního terénu [14]

Protože skenování probíhá na velkém území a robot často uzavírá smyčky, je použit algoritmus SLAM. Algoritmus určí pomocí odometrie a 3D skenů okolí, jestli se na této pozici nacházel a smyčka je opravdu uzavřena. Jednotlivé skeny okolí jsou softwarově porovnávány a je pravděpodobnostně vypočítávána jejich duplicita.

### ■ 2.1.3 Stanley: The Robot that Won the DARPA Grand Challenge

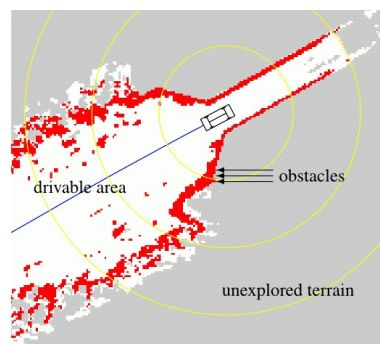
Dalším zajímavým projektem [15] je vítěz soutěže DARPA(Defense Advanced Research Projects Agency) Grand Challenge 2005, při které musel autonomní robot urazit náročnou trasu pouští dlouhou 200 km.

Jedná se o automobil VW Touareg na obrázku 2.3a, který je vybaven velkým množstvím senzorů. Jsou to například dva radary s dosahem 200 m, podle kterých se určuje terén ve větší vzdálenosti. Dále je vybaven barevnou kamerou umístěnou na střeše auta, která sleduje cestu před autem. Vedle kamery je umístěno 5 SICK laserových dálkoměrů pod různým náklonem měřících terén přibližně 25 m před robotem. Poloha je určována pomocí GPS, IMU senzoru a odometrie.

Data z kamery, laseru a radaru jsou ukládána do tří samostatných map, které jsou po úpravě Kalmanovým filtrem spojeny do jedné 2D mapy. Laserová mapa, viz obrázek 2.3b, je reprezentována dvourozměrnou mřížkou, kde každá buňka je označena jako sjízdná, nesjízdná nebo neznámá. Autoři uvádějí, že i malá chyba v pozici robotu, způsobí zásadní chybu při klasifikaci terénu, např. i rovná silnice je hodnocena jako nesjízdná.



(a) : Automobil VW Touareg použitý jako testovací robotický model



(b) : Senzorická mapa reprezentovaná dvourozměrnou mřížkou s ohodnoceným terénem okolí

**Obrázek 2.3:** Ukázky z projektu DAPRA [15]

Pomocí všech senzorů a propracovaného autonomního řízení robot překonal celou trasu a po necelých 7 hodinách dorazil do cíle.

### 2.1.4 A Campaign in Autonomous Mine Mapping

Jeden z ukázkových příkladů využití autonomních robotů je mapování opuštěných dolů. Uskutečnilo se několik podobných projektů [24], [25], [3], kdy autonomní robot mapuje prostředí opuštěných dolů a jeskyní nebezpečných pro člověka. Zde [16] se tým z americké univerzity v Pittsburghu pustil do průzkumu opuštěného uhelného dolu.

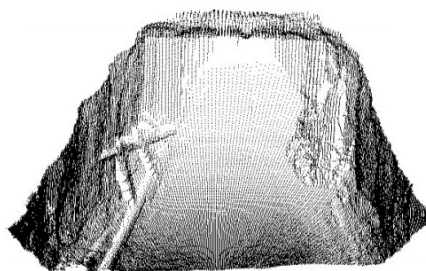
Čtyřkolový téměř tunový robot na obrázku 2.4a s odpruženou nápravou pro dobrou prostupnost těžkým terénem je vybaven řadou senzorů. Hlavním z nich jsou laserové dálkoměry SICK LMS-200 umístěné v přední i zadní části robotu. Senzor se vertikálně naklání v rozmezí 60 stupňů pro generaci 3D skenů a je hlavním zdrojem informací o okolním prostředí. Dále je robot vybaven několika senzory pro snímání plynů a noční kamerou.

Ze získaných 3D skenů, obrázek 2.4b, je následně vytvářena mřížková černobílá 2D mapa vhodná pro autonomní řízení robotu, kde každá buňka mapy podle odstínu určuje průchodnost. Světlá buňka je průchodnější než tmavá. Velikost jedné buňky je 10 cm krychlových a v případě nedostupné hodnoty pro některou buňku je její hodnota určena ze sousedních buněk. To se často stává při měření vodní hladiny, tyto buňky jsou označovány a řídicí algoritmus se tomuto faktu přizpůsobí.





**(a)** : Robot uzpůsobený pro pohyb v náročném terénu důlních šachet



**(b)** : Naskenovaná důlní šachta pomocí laserového dálkoměru

**Obrázek 2.4:** Ukázky z projektu Mine mapping [16]

Pozice robotu je určována pomocí odometrie a laserových senzorů. Samotný průzkumný algoritmus robotu je možné prezentovat jako logický graf příkazů a postupů při různých situacích, jako například restart mise, nebezpečí, porucha a nouzový únik.

### 2.1.5 Shrnutí projektů

Všechny výše zmíněné projekty mají se zadáním této práce téměř totožnou konfiguraci sensorového systému, tedy laserový dálkoměr a inerciální senzor. Přehled o okolí je vždy vytvářen jedním nebo více laserovými dálkoměry, které jsou buď pevně umístěny na konstrukci robotu nebo se pohybují. Ze získaných údajů je vždy vytvářena mřížková mapa poskytující přehled o okolním terénu. Robot je autonomně řízen pomocí této mřížkové mapy, která určuje průjezdnost jednotlivými částmi terénu. K určení pozice robotu je vždy používána odometrie a další senzory, jako GPS a IMU.

Nejvíce podobným a inspirujícím projektem z výše zmíněných je první projekt [12], který velice podrobně mapuje okolní terén robotu. Robot je vybaven pouze dvěma senzory přesně shodnými se zadáním této práce a nevyužívá žádné další. Algoritmus i testovací robotický model je navržen na dobře prostupný a nenáročný terén, jako například městské prostředí nebo pěší park. Výsledkem skenování je detailní 3D mapa okolního terénu, ze které je určována dostupnost jednotlivých jejích částí. Takto ohodnocená mapa je plně vhodná pro autonomní řízení robotu.

## 2.2 Vlastní metoda

Prvním kritériem pro metodu mapování je konfigurace senzorů robotu, jejich počet, umístění a rozsah. Rozmítané laserové dálkoměry mohou být umístěny na robotu různě. Pokud by robot byl vybaven více dálkoměry, které by snímaly velkou oblast nebo by se částečně překrývaly, bylo by povědomí o okolí větší než pouze s jedním senzorem. Jediný laserový dálkoměr dokáže získat dostatečné množství dat o okolí a pokud je navíc senzor vertikálně nebo horizontálně nakláněn získá mnohem více informací než staticky umístěné senzory. V této práci bude dostačující jeden rozmítaný laserový dálkoměr umístěný v přední části robotu, vertikálně nakloněný pod úhlem 45 stupňů. Senzor bude snímat terén před robotem a za jízdy generovat 3D mrak bodů, identicky jako ve výše zmíněné práci [12].

Dalším kritériem je způsob zjišťování pozice robotu. Musí být neustále dostupná informace z jakých souřadnic a při jakém natočení robot právě provedl měření dálkoměry. Díky svojí přesnosti by byl vhodný GPS systém, ale momentálně nebude uvažován pro nepoužitelnost v budovách a dalších oblastech. Často využívanou metou je díky své jednoduché funkčnosti odometrie, to je důvodem jejího použití v této práci. Dopočítávání pozice robotu podle otáčení koleček je ve větším prostoru nespolehlivé, z toho důvodu je odometrie většinou kombinována s dalšími senzory polohy. Tím může být například inerciální navigační systém, který pomocí gyroskopů a akcelerometrů dokáže přispět k určení pozice i polohy robotu bez externích zdrojů. Další přípustnou metodou je triangulace neboli zjišťování pozice podle orientačních bodů.

Samotná odometrie je dostatečně přesná pouze v malém okolí výchozího bodu. To je ideální pro vytvoření lokální mapy nejbližšího prostředí. Není ale možné vytvářet větší mapové úseky bez závažnějších chyb a nepřesností, protože přesnost odometrie výrazně klesá s ujetou vzdáleností robotu. Tato lokální mapa okolí robotu bude vytvářena z nejnovějších dat a ta budou následně opět nahrazena novějšími. Robot bude mít neustále povědomí pouze o svém aktuálním okolí a nebude mít žádné informace o vzdálenějších dříve navštívených místech. Tedy bude řešena pouze aktuální průjezdnost nejbližšího okolí, pro které bude odometrie dostatečně přesná. Robot bude navíc vybaven inklinometrem, senzorem pro měření úhlu náklonu robotického modelu, aby bylo možné mapovat okolní prostředí i v nerovném terénu.

Všechna nově získaná data budou ukládána do mřížkové mapy. Mřížková mapa je jedna z nejvhodnějších reprezentací pro řízení autonomním algoritmem. Ukládání laserových dat do mřížky je velice jednoduché a přehledné. Každá buňka mřížky bude obsahovat výšku terénu naskenované části okolí

a podle její hodnoty bude určena jako průjezdná nebo neprůjezdná. Tedy pokud naměřená výška buňky bude mimo rozmezí nastavené konstantou, bude buňka klasifikována jako *neprůjezdná*. Jízdou robotu bude získáváno větší povědomí o okolí. Postupně tak vznikne 3D mapa okolního prostředí, která bude vhodná pro bezpečnou a bezkolizní autonomní navigaci robotu. Velikost a rozlišení mapy bude rozebrána v dalších kapitolách.



## Kapitola 3

### Implementace

#### 3.1 Simulátor

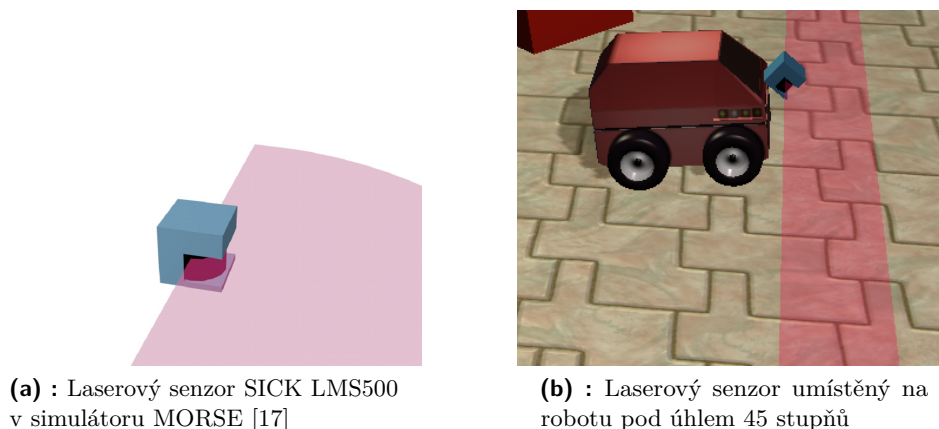
Prvním krokem implementace bylo zprovoznění simulačního prostředí a napojení se systémem IMRnav. Bylo zvoleno simulační prostředí MORSE (Modular OpenRobots Simulation Engine) a to z důvodu ověřené kompatibility s IMRnav, v kterém implementace probíhá.

Simulátor MORSE je opensource robotický generický simulátor, který je vhodný pro autonomní robotické systémy. Jedná se o 3D simulaci, která je vytvářena z Python skriptu a ovládána příkazovým řádkem. Tento program je podporován pouze systémy Linux, doporučené hardwarové požadavky jsou procesor Intel i5 a 4GB operační paměti RAM.

##### 3.1.1 Senzory simulátoru

Simulátor má připravené některé standardní senzory, jako například video kameru, hloubkovou kameru, laserový skener, infračervený senzor, GPS, akcelerometr, gyroskop, senzor baterie, barometr, odometrii, dotykový senzor, anemometr[17]. Laserový senzor je vyobrazený na obrázku 3.1a.

Každý z těchto senzorů má své nastavitelné parametry. U laserových



**Obrázek 3.1:** Sensory simulátoru MORSE

senzorů, které jsou pro tuto práci nejdůležitější, to je dosah, rozlišení, rozsah, úhel, počet a odsazení rovin. Každý senzor má nastavitelnou viditelnost při simulaci a umístění na robotickém modelu.

V této práci je využit senzor SICK LMS500. Tento rozmítaný laserový dálkoměr je umístěn v přední části robotu viz obrázek 3.1b sklopený do země pod úhlem 45 stupňů, jako tomu bylo ve zmíněné práci [12]. Konfigurace senzoru je taková, aby se co nejvíce blížila vlastnostem reálného senzoru a zároveň vyhovovala testování. Laserový dálkoměr měří v jedné rovině s šířkou záběru 180 stupňů a úhel mezi jednotlivými paprsky je půl stupně. Maximální dosah senzoru je nastaven na 5 metrů, tato hodnota je dostatečná pro dobrý přehled robotu o nejbližším okolí.

### 3.1.2 Modely simulátoru

MORSE dále obsahuje předpřipravené roboty a robotické modely. Jsou zde například robot ATRV, robotický model patrobot, pioneer3dx, model automobilu Hummer, model gyroptéry, ponorka a další [17].

Stejně jako u jednotlivých senzorů i jednotlivé robotické modely umožňují nastavení celé řady parametrů. Základní parametry jsou počáteční poloha a natočení. Další složitější nastavení jsou: působení gravitace, hmotnost, tření, kolize, pružnost, pohon a pozemní pohyb.

V této práci je použit robotický model typu ATRV [18], který je zobrazen na obrázku 3.2a. Jedná se o čtyřkolový robot s podobným typem pohybu jako

tank. To znamená, že robot je schopný otáčet se na místě a zatáčí pomocí přibrzdování kol na jedné straně. Reálný robotický model na obrázku 3.2b je vybaven několika lištami pro umístění senzorů.



**(a)** : Testovací robotický model ATRV v simulátoru MORSE [17]



**(b)** : Reálný robotický model ATRV vybavený laserovým dálkoměrem [18]

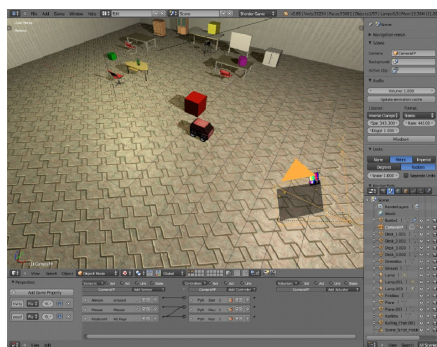
**Obrázek 3.2:** Robotické modely simulátoru MORSE

### ■ 3.1.3 Prostředí simulátoru

Simulátor umožňuje vytváření vlastních testovacích prostředí, ovšem lze využít prostředí předdefinovaná. Ta jsou zastoupena v několika variantách, například prostředí budovy, přírody a vodní plochy.

Každé prostředí má mnoho nastavitelných parametrů jako několik typů kamer, jejich polohu a natočení. Dále nastavení času, gravitace, světla, pozadí, počasí, mlhy, fyziky, frekvence vykreslování a výpisů parametrů. Navíc je v prostředí možno editovat jednotlivé prvky, objekty a terén pomocí Blender GUI. Objekty se posouvají, rotují, mění svoji velikost a další fyzikální vlastnosti.

V této práci je využito kancelářské prostředí budovy, na obrázku 3.3a, robot na rovném terénu tohoto prostředí bude mapovat okolí. Prostředí na obrázku 3.3b je venkovní prostor s velkou členitostí terénu, na kterém robot kromě samotných překážek musí umět pracovat i s nerovnostmi povrchu.



(a) : Blender GUI pro kancelářské prostředí simulace



(b) : Testovací prostředí outdoor s mobilním robotem

Obrázek 3.3: Prostředí simulátoru MORSE

#### 3.1.4 Běh simulátoru

Robotický model i sledovací kamera jsou v 3D simulaci ovládány klávesnicí. MORSE využívá Blender Game Engine, jedná se o základ počítačových her. Simulátor umožňuje multi-texturní renderování, stínování a vykreslování světelných paprsků.

## 3.2 Implementace kodu

Samotná implementace mapovací metody je realizována v systému IMRnav, jako modul obsahující veškerý kód. Tento systém podporuje komunikaci se zvoleným simulátorem MORSE a reálnými robotickými modely.

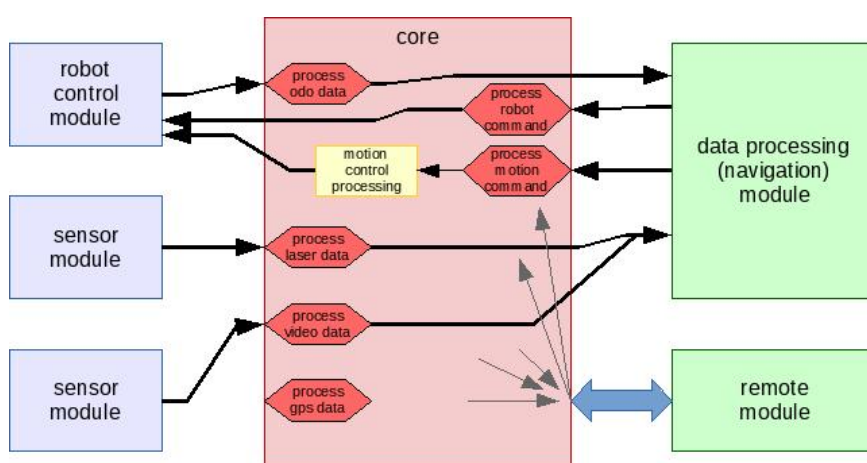
### Systém IMRnav

Struktura systému IMRnav, patrná z obrázku 3.4, je navržena jako modulární. Hlavním úkolem systému je propojení modulů tak, aby si mohli posílat data, příkazy a události. Moduly je možné rozdělit na robotické moduly, sensorické moduly, komunikační moduly a navigační moduly. Moduly mají jednotné vstupní rozhraní zahrnující funkce pro posílání sensorických dat, posílání změny konfigurace systému a posílání provozních hodnot. Všechny moduly jsou propojeny přes centrální prvek "core", který přijímá data od jednotlivých modulů a přeposílá je ostatním modulům, které daná data požadují. Hlavní



funkcí jádra je udržovat informaci o připojených modulech, předávat data mezi moduly a spravovat globální konfiguraci systému.

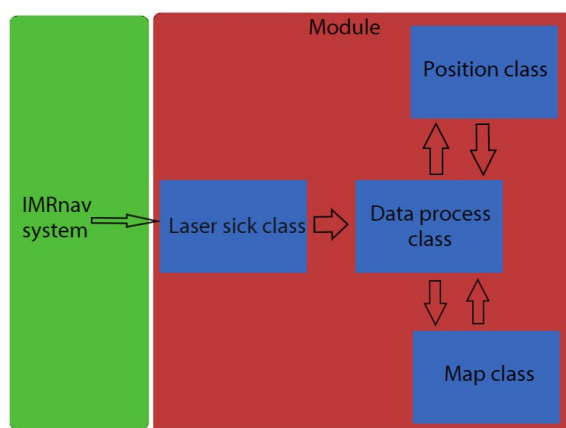
Rozhraní pro řízení robotu je reprezentováno samostatným modulem, který je odvozen od třídy CRobotInterface. Řízením je zde myšleno řízení motorů zajišťujících pohyb robotu. Rozhraní navíc poskytuje data ze základních senzorů jako například data z inkrementálních snímače polohy (odometrie). Může také poskytovat další senzorská data, pokud je např. rozhraní příslušných senzorů integrováno do rozhraní robotu<sup>1</sup> [27].



Obrázek 3.4: Struktura systému IMRnav [27]

V tomto systému je vytvořen modul, jehož struktura je vyobrazena na obrázku 3.5, zpracovávající data přijatá ze simulátoru. Jedná se o výstupní data z laserových dálkoměrů a odometrie. Protože laserová data mohou být přijímána rychleji než je možné je zpracovat a zapsat, je nutné jejich ukládání do vyrovnávací paměti (bufferu). Laserová data jsou kopírována do tohoto bufferu běžícím v jiném vlákne, aby nebyl zdržován běh celé aplikace. Z bufferu jsou hodnoty čteny v jiné třídě a následně zpracovávány a uloženy do mapy. Pokud jsou přijata nová laserová data dříve než jsou data uložená v bufferu čtena, tak nová data přepisují buffer a původní jsou ztracena. Hodnoty příchozí odometrie jsou přeposlány do třídy pro výpočet polohy bez nutnosti meziukládání, protože zpracování odometrie je rychlé.

<sup>1</sup>Data z libovolných senzorů však mohou být poskytována i samostatnými moduly.



Obrázek 3.5: Struktura modulu realizující metodu mapování

### 3.2.1 Zpracování laserových dat

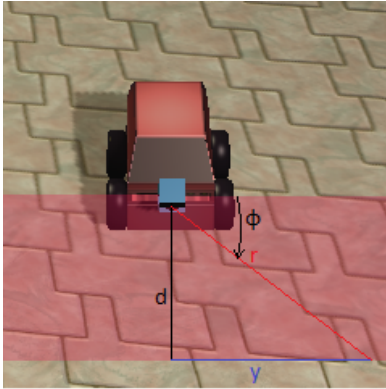
Z příchozích laserových dat je vypočítávána výška terénu relativně k úrovni povrchu na kterém se robot nachází. Jak už bylo zmíněno výše, senzor je umístěn v přední části robotického modelu ve výšce 50 centimetrů a je nakloněn 45 stupňů směrem do země. Tento senzor snímá v jedné rovině s šířkou záběru 180 stupňů, úhel mezi jednotlivými paprsky je půl stupně a jeho dosah je 5 metrů.

Data přečtená z bufferu jsou umístěna v poli typu *double*. Pole má délku závislou na nastavení senzoru, pro výše definované nastavení rozsahu a úhlu mezi paprsky je pole dlouhé 360 prvků. Hodnota každého prvku je naměřená délka jednoho paprsku od senzoru k překážce, pokud není naměřena překážka je v buňce hodnota maximálního dosahu senzoru, v tomto případě 5 metrů. Vzdálenosti jsou udávány v metrech. S laserovými daty jsou dále přijímány dodatečné informace, jako časový krok a nastavení laserového senzoru.

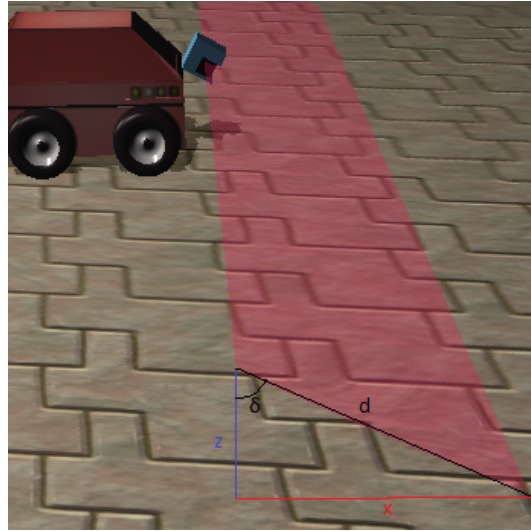
### Teorie výpočtu

Ve třídě pro zpracování dat je dalším krokem transformace délek naměřených z laserového dálkoměru na informace o překážkách. To je uskutečněno jednoduchým výpočtem, pomocí vzdálenosti paprsku a úhlem, pod kterým vystupuje ze senzoru, jsou vypočítány souřadnice překážky  $x_p$ ,  $y_p$ ,  $z_p$ , tedy bodu detekce překážky.

Na obrázku 3.6a jsou zobrazeny veličiny výpočtu. Ze známých hodnot



(a) : Názorné zobrazení jednotlivých veličin na reálném robotu



(b) : Názorné zobrazení jednotlivých veličin na reálném robotu

**Obrázek 3.6:** Zobrazení veličin pro výpočet

$r$  značících délku paprsku a  $\phi$  značící úhel výstupu paprsku z laseru je vypočítána pomocí rovnice 3.1 a 3.2 délka úsečky  $d$  a souřadnice překážky  $y_p$ . Následně jsou z parametru  $d$  a úhlu náklonu senzoru  $\delta$ , na obrázku 3.6b, podle rovnice 3.3 a 3.4 vypočítány souřadnice překážky  $x_p$  a  $z_p$ . Tyto souřadnice polohy překážky se vztahují pouze k soustavě robotu, pro zápis do globální mapy je nutná jejich opětovná transformace.

$$d = r \sin(\phi) \quad (3.1)$$

$$y_p = r \cos(\phi) \quad (3.2)$$

$$x_p = d \cos(\delta) \quad (3.3)$$

$$z_p = d \sin(\delta) \quad (3.4)$$

Body překážky  $x_p, y_p$  z lokálních souřadnic robotu musí být transformovány do globálních souřadnic mapy, které jsou definované mřížkou mapy. Souřadnici  $z_p$  není nutné přepočítávat, protože je počítáno s nulovou elevací robotického modelu. Transformaci provedeme pomocí rovnice 3.5, kde  $B_M$  je hledaný rozšířený vektor globálních souřadnic mapy,  $B_R$  je známý rozšířený vektor lokálních souřadnic příslušného bodu a  $T$  je transformační matice, která má tvar 3.6.

$$B_M = T \cdot B_R \quad (3.5)$$

$$T = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & x \\ \sin(\alpha) & \cos(\alpha) & y \\ 0 & 0 & 1 \end{pmatrix} \quad (3.6)$$

Po roznásobení vychází dvě rovnice 3.7 a 3.8, kde  $[x_m, y_m]$  jsou globální souřadnice mapy,  $[x_p, y_p]$  jsou lokální souřadnice robotu, a  $[x_o, y_o]$  jsou souřadnice posunu z odometrie a úhel  $\alpha$  udává otočení robotu kolem osy z.

$$x_m = x_p \cos(\alpha) - y_p \sin(\alpha) + x_o \quad (3.7)$$

$$y_m = x_p \sin(\alpha) + y_p \cos(\alpha) + y_o \quad (3.8)$$

### ■ Kod

Výše zmíněný výpočet a následné zapsání překážky do mapy v jazyce C++ je vidět ve výpisu 3.1. Proměnná `laser_resolution` definuje úhel mezi jednotlivými paprsky laserového senzoru, `laser_range` definuje jeho dosah, `laser_rotation_ry` definuje jeho náklon, `laser_translation_z` definuje výšku umístění senzoru a `cell_width` je konstanta pro různé velikosti buněk mapy.

Další kód je uzavřen v cyklu, v každé iteraci jsou vypočítávány hodnoty pro jeden paprsek. Na řádku 10 je podmínka zjišťující, jestli laserový senzor s tímto paprskem opravdu něco naměřil a nevrací pouze hodnotu svého maximálního dosahu nebo jinou nesprávnou hodnotu. Na řádcích 15 až 18 jsou rovnice 3.1, 3.2, 3.3 a 3.4. Na řádku 20 a 23 jsou rovnice 3.7, 3.8 a na řádku 26 je vypočítána výška překážky.

Na řádku 28 je podmínka ošetřující, zda laserový paprsek nemá stejné souřadnice v mapě jako ten předchozí. Jakmile je tato podmínka nepravdivá, jsou naměřené hodnoty zapsány do mapy viz. řádek 39. V případě pravdivosti podmínky se přistupuje k dalšímu prověřování v podmínce na řádku 31, kde je určováno, jestli je hodnota hodnota přínosná a má být zapsána do mapy.

**Listing 3.1:** Zpracování laserových dat do mapy.

```

1 laser_resolution =M_PI/360 ;
2 laser_range = 5.0;
3 laser_rotation_ry= M_PI/4 ;
4 laser_translation_z=0.6;

```

```

5  cell_width =0.1;
6
7  last_map_x=-1;
8  last_map_y=-1;
9
10 for( unsigned int ray=0; ray < laserData.ranges.size(); ray++)
11 {
12
13 if(laser_range>laserData.ranges[ray] && laserData.ranges[ray] > 0)
14 {
15     double d=laserData.ranges[ray]*sin(ray*laser_resolution);
16     y_length= laserData.ranges[ray]*cos(ray*laser_resolution);
17     x_length= d*cos(laser_rotation_ry);
18     z_length=d*sin(laser_rotation_ry);
19
20     map_x= round((my_odo.position.x
21     + x_length *cos(my_odo.position.heading)
22     -y_length *sin(my_odo.position.heading))/cell_width);
23     map_y= round((my_odo.position.y
24     + y_length *cos(my_odo.position.heading)
25     + x_length *sin(my_odo.position.heading))/cell_width);
26     terrain= laser_translation_z-z_length;
27
28     if((map_x == last_map_x) && (map_y == last_map_y))
29     {
30         // same position
31         if(std::isnan(my_map.getCell(map_x,map_y))||
32         (my_map.getCell(map_x,map_y) < terrain) )
33         {
34             my_map.setCellValue(map_x,map_y, terrain);
35         }
36     }
37     else
38     {
39         my_map.setCellValue(map_x,map_y, terrain);
40     }
41 }
42 }
43
44 last_map_x=map_x;
45 last_map_y=map_y;
46
47 }

```

### 3.2.2 Odometrie

Jako základní senzorický systém pro určování polohy je odometrie, která umožňuje měřit relativní změnu pozice mezi dvěma časovými okamžiky. Odometrický polohovací systém převádí naměřenou rotaci kol mobilního robotu na jeho aktuální polohu a orientaci. Rotace kol může být zjišťována přímo z akčních zásahů na motory, nebo z optických, magnetických, či jiných

obdobných čidel snímající přímo jejich otáčení. Použití enkodérů je zpravidla konstrukčně jednodušší.

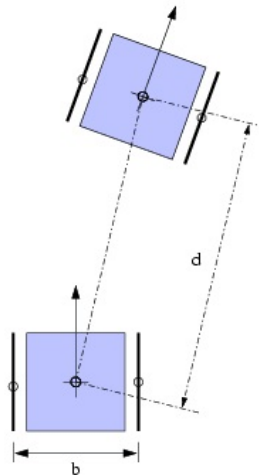
Měření odometrie je zatíženo přídatnou nesystematickou měřicí chybou v případě, že pohonná kola robotu se protáčí vlivem povrchu nebo blokáce těla robotu. Absolutní poloha se obecně získá integrací jednotlivých změn pozice:

$$x_1 = x_0 + \int_{t_0}^{t_1} \frac{dx}{dt} dt \quad (3.9)$$

kde  $\frac{dx}{dt}$  je změna pozice neboli rychlost,  $x_0$  je pozice v čase  $t_0$  a  $x_1$  je pozice v čase  $t_1$ .

Vlastní poloha robotu je určována na základě znalosti kinematiky daného mobilního robotu, velikosti kola a počtu impulsů na jednu otáčku. Robotický model ATVR vybraný pro tuto práci je vozidlo diferenciálního typu. Další typy robotických modelů jsou například auto a všesměrové roboty.[20]

Základním rysem diferenciálního modelu jsou dvě nezávisle poháněná kola nebo nápravy. Pokud se obě točí stejně rychle stejným směrem, robot jede rovně. Pokud se naopak točí opačným směrem, točí se celý robot na místě kolem středu mezi nápravami. To je také nejčastější referenční bod. Popis vozidla je na obrázku 3.7.



**Obrázek 3.7:** Znázornění pohybu diferenciálního vozidla a měřené veličiny [22]

Natočení robotu závisí pouze na rozdílu celkové ujeté vzdálenosti pravého a levého kola a nikoliv na průběhu jednotlivých změn.[22] [23] .

Výpočet pozice a směru rovnice je uskutečněno pomocí následujících rovnic

$$\varphi_t = \varphi_{t-1} + \frac{\Delta S_l - \Delta S_r}{b} \quad (3.10)$$

$$x_t = x_{t-1} + \frac{\Delta S_l - \Delta S_r}{2} * \cos(\varphi_t) \quad (3.11)$$

$$y_t = y_{t-1} + \frac{\Delta S_l - \Delta S_r}{2} * \sin(\varphi_t) \quad (3.12)$$

kde  $b$  je rozchod kol,  $[x_t, y_t, \varphi_t]$  a  $[x_{t-1}, y_{t-1}, \varphi_{t-1}]$  jsou souřadnice robotu v kartézské soustavě souřadnic a jeho natočení v čase  $t$  a  $t-1$ .  $\Delta S_l$  a  $\Delta S_r$  jsou vzdálenosti ujeté pravým resp. levým kolem v časovém intervalu  $\langle t; t-1 \rangle$ .

Hlavními daty vstupujícími do třídy vypočítávající relativní polohu robotu resp. odometrii je celková ujetá dráha jednotlivých kol. Dalšími vstupními daty jsou časový krok a rozchod kol robotického modelu. Výpočet je realizován pomocí výše uvedených rovnic 3.10, 3.11 a 3.12. Hodnoty posunu  $x, y$  jsou udávány v metrech a úhel  $\varphi$  je vypočítáván v radiánech a je udáván v kladném směru od osy  $x$  k ose  $y$ .

Měření otočení kol je vždy zatíženo chybou. Tato chyba je většinou sama o sobě zanedbatelná, ale při větší dráze může mít zásadní vliv. Čím více změn pozice robotu je měřeno, tím více se chyba akumuluje a tím více se vypočtená relativní pozice liší od skutečné [20][22].

### ■ 3.2.3 Mapa

Mřížková mapa je jedna z nejnižších možných reprezentací prostředí. Jedná se pouze o vhodně reprezentovaná a uchovaná sensorická data. Tato reprezentace je velice vhodná pro rychlý přístup k datům pro předcházení a řešení kolizí, kdy je potřeba rychlých reakcí. Uplatní se tedy často na úrovni autonomního řízení pohybu. Nevýhodou mřížek obsazenosti je skutečnost, že paměťová náročnost na udržování mřížek obsazenosti roste s velikostí prostředí  $O(n^2)$  a ne s jeho složitostí. S vysokou paměťovou náročností roste též výpočetní čas pro operace s těmito datovými strukturami, což může být pro aplikace pracující v rozsáhlejších prostorech závažný nedostatek [20].

Všechna data ze senzorů jsou ukládána do této mapy reprezentované dvou-rozměrnou mřížkou. Jedná se o zapsání pozic překážek, jejichž výpočet byl ukázán v předešlé kapitole. Každá buňka reprezentuje část reálného světa a její hodnota přítomnost překážky. Velikost mřížky automaticky definuje velikost mapy a přesnost s jakou je možno počítat.

Pokud se rozšíří hodnoty v mřížce z *překážka/volno* na více hodnot, může být reprezentována pravděpodobnost přítomnosti překážky. Takto se vytvoří pravděpodobnostní mřížka obsazenosti (occupancy grid), která umožňuje poměrně přímočarou integraci dat z několika různých sensorových systémů [26].

### ■ Implementovaná metoda

Samotná mapa je reprezentována dvourozměrným polem typu *double*, rozměr pole určující velikost a rozlišení mapy je definován při vytvoření třídy. Hodnota jednotlivých buněk pole určuje výšku terénu v příslušné oblasti, tato hodnota je udávána v metrech. Pokud je buňka stále neznámá nebo se nepodařilo naměřit výšku terénu má hodnotu *NaN*. Dále je vytvořena dvourozměrná mřížka definující průjezdnost jednotlivých výškových oblastí jako *průjezdná/neprůjezdná*. Klasifikace těchto buněk je určována z výškové mapy, podle vztahu 3.13, kde *terrain* označuje naměřenou výšku terénu, *min* a *max* jsou konstanty určující záporné resp. kladné převýšení.

$$\min < \text{terrain} < \max \quad (3.13)$$

Pokud platí tato podmínka je buňka klasifikována jako *průjezdná*, v opačném případě je buňka klasifikována jako *neprůjezdná*.

Jakmile je nastaven příliš velký rozměr buněk, potom více laserových paprsků může naměřit více hodnot v oblasti pouze jedné buňky. V takovém případě je příslušné oblasti přidělena hodnota nejvýše naměřeného terénu z dotyčných paprsků, toto ošetření předchází nepravdivým hodnotám v buňkách. V případě, že dojde k chybnému měření buňky mapy, která má přidělenou hodnotu z předešlých měření, je ponechána původní hodnota buňky.

Jakmile robot vyjede z definované mapy, je nejvzdálenější část mapy zapomenuta a je vytvořen nový prostor pro zápis nových dat. Tedy dvourozměrné pole se posune požadovaným směrem o desetinu velikosti mapy a buňky na druhém kraji jsou přepsány.



## Kapitola 4

### Experiment

#### 4.1 Experiment mapování terénu

Navržená metoda pro mapování terénu pozemním robotem byla otestována v simulátoru MORSE. Experimentálně byla zjišťována funkčnost metody pro jednotlivé případy a stavy prostředí.

Výsledná mapa je exportována jako dvourozměrná matice do textového souboru a při každém dalším skenu je tento soubor přepsán nejnovější verzí mapy. Data ze souboru jsou vykreslena pomocí programu Matlab 2016a a to funkcí *bar3()*, která vytváří třírozměrný graf z dvourozměrné matice. Protože tato funkce vykresluje hodnotu nulovou stejně jako hodnotu *NaN*, jsou naměřené výšky terénu uměle zvýšeny o 0.1 metru. Hodnoty buněk pro které se nepodařilo naměřit překážku jsou uměle navýšeny o hodnotu -0.1 metru. Všechny tyto změny byly provedeny pro zlepšení grafické čitelnosti mapy.

##### 4.1.1 Popis experimentů

Experimenty byly prováděny v prostoru virtuální kancelářské budovy. V průběhu experimentu musely být dodrženy předpoklady pro správné fungování metody, jako např. rovná podložka, po které se robot pohybuje, a statické

překážky v prostoru.

První část experimentu se zabývá detekcí jednoduchých překážek a její správnou reprezentací v mapě vůči okolí. V druhé části experimentu bude namapována větší oblast s mnoha překážkami, kde robot bude projíždět více směry. V předposlední, třetí části experimentu bude plošně mapováno území o velkém rozsahu. Konečná čtvrtá část experimentu se zabývá mapováním nerovného terénu v přírodě, zároveň je splněna podmínka pohybu robotu na rovném povrchu.

### ■ Parametry při testování

Nastavení laserového dálkoměru SICK LMS 500 bylo popsáno již výše a je znovu shrnuto v tabulce 4.1.

Úhel záběru	180°
Úhlové rozlišení	0.5°
Maximální detekovatelná vzdálenost	5 m

**Tabulka 4.1:** Parametry laserového senzoru

Velikost mapy je nastavena maximálně na 400 metrů čtverečních, tato velikost je hraniční pro přehledné vykreslení pomocí zvoleného nástroje Matlab 2016a. Velikost buňky je pro experiment zvolena jako jeden decimetr čtvereční, toto rozlišení dostatečně vystihuje podobu prostředí, ale je odzkoušen i přesnější rozměr.

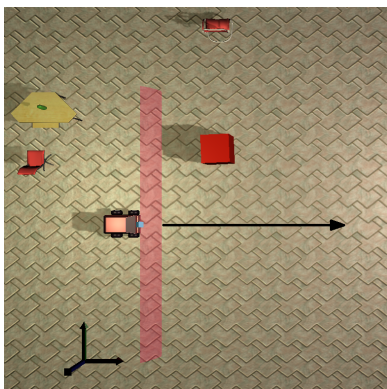
### ■ 4.1.2 Výsledky experimentu

#### ■ První část

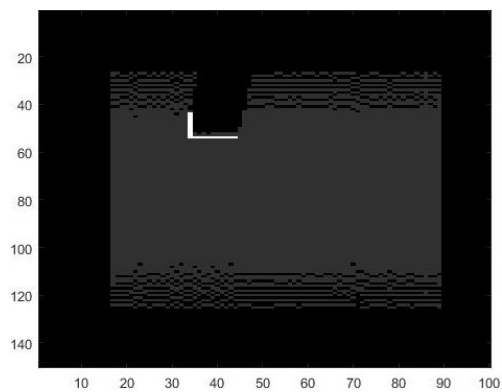
Výsledky první části experimentu jsou na obrázku 4.1, při kterém byl namapován okolní terén robotu s překážkou v podobě červené krychle. Robot se z

pozice na obrázku 4.1a přesune dopředu ve směru šipky, tedy po ose X přibližně o 7 metrů. Robot je inicializován na souřadnicích mapy  $[x,y]=[100,75]$  a její velikost je 10x15 metrů.

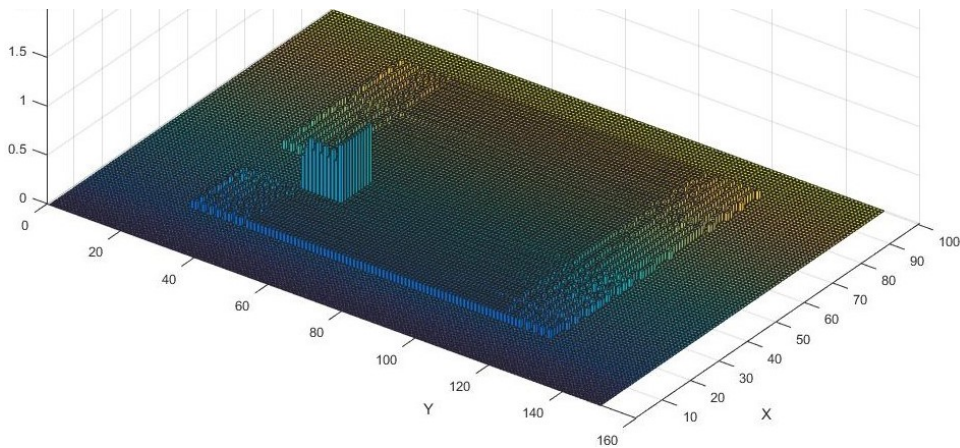
Na obrázku 4.1b a 4.1c je graficky znázorněna vytvořená mapa, ze které je patrná překážka na jinak rovném povrchu. Okolní terén se pohybuje kolem uměle navýšené hodnoty 0.1 metru s odchylkou  $\pm 3$  milimetry, která je způsobena jízdou robotu. Buňky s nulovou hodnotou jsou neznámá místa, kde neproběhlo měření.



(a) : Počáteční pozice robotu v první části experimentu s vyznačenou trajektorií



(b) : Černé buňky jsou neznámé, šedé jsou sjízdné a bílé nesjízdné



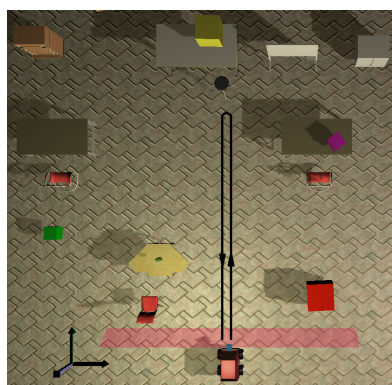
(c) : Grafický výstup mapy vytvořené v první části experimentu

**Obrázek 4.1:** První část experimentu

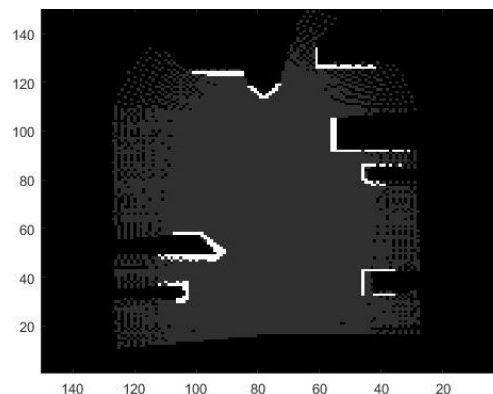
## Druhá část

Výsledky druhého experimentu jsou na obrázku 4.2, kde bylo mapováno na mapu o velikosti 225 metrů čtverečních a v prostoru byl rozmístěn kancelářský nábytek, viz obrázek 4.2a. Robot z výchozí pozice na obrázku 4.2a popojede směrem dopředu, tedy po ose Y přibližně 10 metrů a otočí se o 180 stupňů a ujede stejnou trasu zpět. Robot je inicializován na souřadnicích mapy  $[x,y]=[100,10]$ .

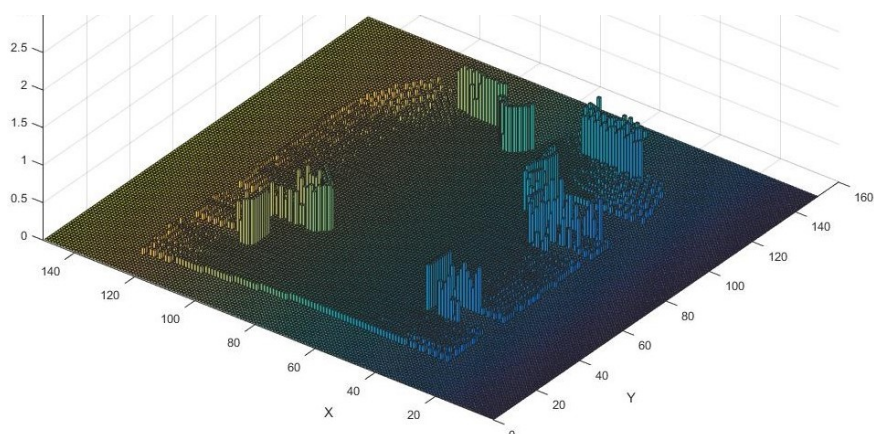
Při této trajektorii byla vytvořena mapa, jejíž grafický výstup je na obrázku 4.2b a 4.2c. Robot namapoval všechny okolní překážky z obou směrů podle osy Y. Díky tomu jde poměrně dobře určit trojúhelníkový tvar stolu v levé části a červené krabice v pravé části. Nábytek v zadní části nelze přesně podle mapy identifikovat, ale jeho rozměry jsou jasně zachyceny.



(a) : Počáteční pozice robotu v druhé části experimentu s vyznačenou trajektorií



(b) : Černé buňky jsou neznámé, šedé jsou sjížděné a bílé *nesjížděné*



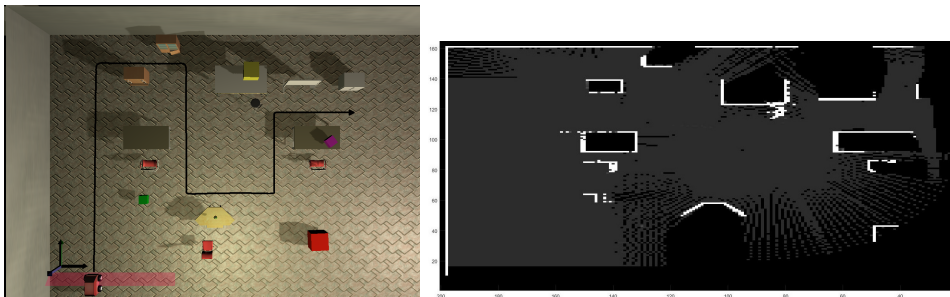
(c) : Grafický výstup mapy vytvořené v druhé části experimentu

**Obrázek 4.2:** Druhá část experimentu

### ■ Třetí část

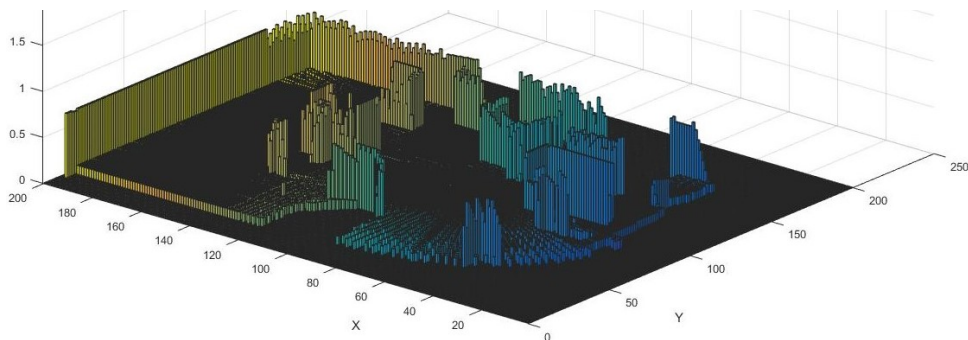
V této části experimentu bylo provedeno mapování většího okolí robotu. Proto byla nastavena větší mapa, konkrétně 400 metrů čtvereční, větší rozměr mapy už není bohužel možné přehledně graficky vyjádřit. Startovní pozice robotu je na obrázku 4.3a, z ní robot jede po vyznačené trajektorii a mapuje překážky. Robot je v mapě inicializován na pozici  $[x,y]=[170,10]$

V mapě na obrázku 4.3b a 4.3c jsou zřetelné některé objekty. Velice dobře patrné jsou zdi, skříně a stoly. Robot v tomto experimentu úspěšně namapoval velkou část místnosti a zachytil většinu překážek. Jejich výška je ale vždy velice podobná, protože robot dokáže měřit pouze do výše jednoho metru.



(a) : Počáteční pozice robotu v třetí části experimentu s vyznačenou trajektorií

(b) : Černé buňky jsou neznámé, šedé jsou sjízdné a bílé nesjízdné



(c) : Grafický výstup mapy vytvořené v třetí části experimentu

**Obrázek 4.3:** Třetí část experimentu

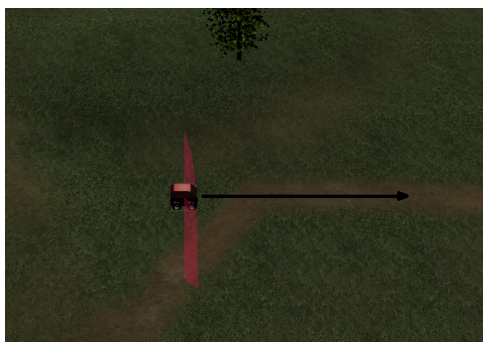
### ■ Čtvrtá část

V této části experimentu byl mapován nerovný terén v lesní oblasti. Pro pohyb robotu byla vybrána trajektorie přes relativně rovnou cestu ve venkovním

#### 4. Experiment

simulačním prostředí viz. obrázek 4.4a. Mapa byla nastavena na rozměř 10x20 metrů. Robot je v mapě inicializován na pozici  $[x,y]=[50,30]$ .

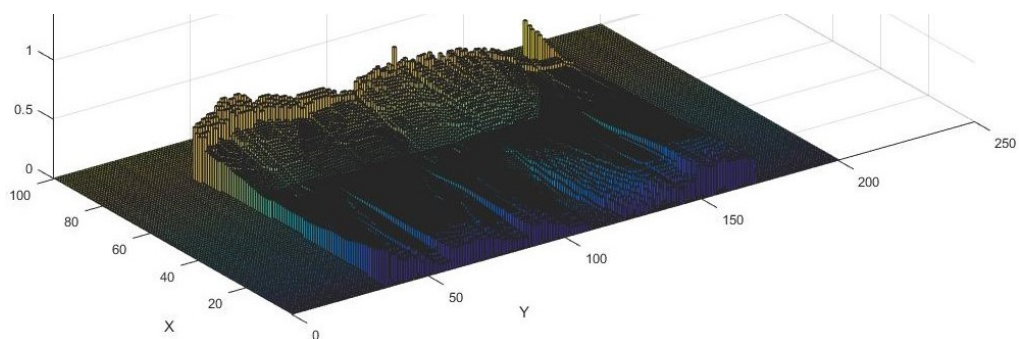
Vytvořená mapa na obrázku 4.4b a 4.4c přibližně zachycuje tvar okolního prostředí. Velice dobře zřetelný je kopec vlevo od robotického modelu, který se táhne podél celé cesty. Na pravé straně jsou naskenovány drobné nerovnosti povrchu. Protože zvolená cesta není úplně rovná a algoritmus nekompensuje náklon robotu, tak veškeré měření je zatíženo chybou.



(a) : Počáteční pozice robotu v třetí části experimentu s vyznačenou trajektorií



(b) : Černé buňky jsou neznámé, šedé jsou *sjízdné* a bílé *nesjízdné*



(c) : Grafický výstup mapy vytvořené v čtvrté části experimentu

**Obrázek 4.4:** Čtvrtá část experimentu

## 4.2 Shrnutí experimentů

Výsledkem všech experimentů byla vytvořená mapa okolí robotu s grafickým výstupem, ze kterého je patrná výška jednotlivých oblastí terénu a překážek. Dalším výstupem je oklasifikovaná mapa určená pro autonomní řízení, kde jsou jednotlivé buňky hodnoceny jako *sjízdné* nebo *nesjízdné*.

Robot v testech naskenoval dostatečně pro autonomní řízení jak profil terénu tak významné překážky. Při příliš rychlém natáčení robotu docházelo k nerovnoměrnému skenování a velké množství buněk nebylo definováno nebo jejich výška neodpovídala skutečnosti. Při jízdě po relativně rovné cestě dokázal robot namapovat členitost okolního terénu.

Při experimentu byl potvrzen předpoklad závislosti rovného povrchu pod robotu pro správné mapování okolního prostředí. Bez inerciálního senzoru je robot schopný mapovat pouze při jízdě po nenakloněné rovině. Nerovnosti terénu mohou způsobit detekci falešných překážek při náklonu robotu. Metoda je tedy vhodná zejména do prostředí budov.

Slabinou mapovací metody jsou prudké výškové propady terénu. Pokud robot na takový propad narazí a laserový senzor nenaměří žádnou hodnotu, zůstává tato oblast jako neznámá. K tomuto stavu může například dojít, jakmile robot dojedez na počátek klesajícího schodiště nebo prudkého sjezdu. V tu chvíli vzniká těsně před robotem slepé místo, které nedokáže klasifikovat. Jedná se o principiální nedostatek metody pro skenování šikmým laserem.





## Kapitola 5

### Závěr

V této práci byly prostudovány některé používané metody pro mapování terénu s využitím mobilních autonomních robotů. Vybrané vědecké články se zabývají vytvářením třírozměrných map sloužících pro autonomní řízení samotného robotu. Tyto roboty získávaly povědomí o svém okolí pomocí laserových senzorů a pozici pomocí odometrie a dalších senzorů.

Následně byl vytvořen modul v systému IMRnav pro zpracování a ukládání dat z laserového dálkoměru a odometrie. Aktuální pozice robotu je zjišťována odometrií a s kombinací laserových dat je vytvářena mapa. Tato třírozměrná mapa reprezentuje naskenované okolí robotu pomocí výškového ohodnocení překážek a následné klasifikace jako průjezdné nebo neprůjezdné.

Tato reprezentace okolního prostředí je vhodná pro autonomní řízení robotu s použitím základních algoritmů pro plánování na mřížce. Například to mohou být potenciálová pole [28], grafové algoritmy prohledávání do šířky, Dijkstra, A\*, B\*. Mapa by také mohla být upravena do podoby pravděpodobnostní mřížky obsazenosti, tím získáme jistotu s jakou je daná buňka obsazená nebo volná .

Metoda mapování byla testována v simulačním prostředí. Ve všech experimentech robot vytvořil mapu reprezentující jeho okolí, kde byla pozice a výška překážky zaznamenána s chybou odpovídající rozlišení mřížky. V první části experimentu byla testována základní detekce překážky. V druhé části experimentu bylo prověřeno skládání skenů překážek z více směrů. V třetí části byla otestována schopnost robotu získat přehled ve větším prostředí s velkým množstvím překážek a složitější trajektorií. V poslední části experi-

mentu robot projížděl po rovné cestě v lesním prostředí a mapoval nerovnosti okolní krajiny.

Vytvořená mapa, která je hlavním výstupem metody, je vhodná pro autonomní řízení robotu. Nebyl použit inerciální senzor, který by umožňoval měřit náklon robotu. Z toho důvodu je možné vytvořenou metodu mapování použít v případě, že se robot nachází na rovném podkladě a není nakloněn. Algoritmus je vhodný pro použití do prostoru budov a rovných plochých prostranství. Pro složitější terén bude nutné uvažovat i náklon robotu.



## Literatura

- [1] Sebastian Thrun, Dieter Fox, Wolfram Burgard, *Probabilistics robotics*, 1999-2000
- [2] Roland Siegwart, Illah R. Nourbakhsh, *Autonomous Mobile Robots*, 2004
- [3] D. Ferguson, A. Morris, D. Hahnel, C. Baker, Z. Omohundro, C. Reverte S. Thayer, W. Whittaker, W. Whittaker, W. Burgard, S. Thrun *An Autonomous Robotic System for Mapping Abandoned Mines*
- [4] Jean-Francois Lalonde, Nicolas Vandapel , Daniel F. Huber, and Martial Hebert *Natural Terrain Classification using Three-Dimensional Ladar Data for Ground Robot Mobility* , June 7, 2006
- [5] I. S. Kweon, T. Kanade *High resolution terrain map from multiple sensor data* ,
- [6] Cang Ye, Johann Borenstein *A Method for Mobile Robot Navigation on Rough Terrain* , 2004
- [7] Michael Montemerlo, Sebastian Thrun, Daphne Koller, Ben Wegbreit *FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem* , 2002
- [8] Dirk Hahnel, Wolfram Burgard, Dieter Fox, Sebastian Thrun *An Efficient FastSLAM Algorithm for Generating Maps of Large-Scale Cyclic Environments from Raw Laser Range Measurements* , 2001
- [9] Zbyněk Winkler *Simultaneous Localization and Mapping* , 2003-12-08
- [10] Georg Klein, David Murray *Parallel Tracking and Mapping for Small AR Workspaces*, 2007

- [11] Petr Neduchal *Návrh a testování metod vizuální simultánní lokalizace a mapování*, 2013
- [12] Denis F. Wolf, Gaurav S. Sukhatme, Dieter Fox, Wolfram Burgard *Autonomous Terrain Mapping and Classification Using Hidden Markov Models* , 2005
- [13] Lawrence R. Rabiner *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition* , 1989
- [14] Patrick Pfaff, Rudolph Triebel, Wolfram Burgard *An Efficient Extension to Elevation Maps for Outdoor Terrain Mapping and Loop Closing*, 2007
- [15] Sebastian Thrun, Mike Montemerlo, Hendrik Dahlkamp, David Stavens *Stanley: The Robot that Won the DARPA Grand Challenge*, 2005
- [16] C. Baker, A. Morris, D. Ferguson, S. Thayer, C. Whittaker, Z. Omohundro, C. Reverte, W. Whittaker, D Hahnel and S. Thrun *A Campaign in Autonomous Mine Mapping*, 2004
- [17] [http://www.openrobots.org/morse/doc/1.2/components\\_library.html](http://www.openrobots.org/morse/doc/1.2/components_library.html)
- [18] <http://rescue.isr.ist.utl.pt/atrujr.php>, <https://web.eecs.umich.edu/~kuipers/research/ssh/robot-gallery.html>
- [19] Štěpán, P. *Vnitřní reprezentace prostředí pro autonomní mobilní roboty*, Disertační práce. ČVUT, Elektrotechnická fakulta, katedra kybernetiky. Praha 2001
- [20] Ing. Roman Mázl, *Lokalizace pro autonomní systémy*, Disertační práce. ČVUT, Elektrotechnická fakulta, katedra kybernetiky. Praha 2007
- [21] Ing. Tomáš Neužil, *Průběžná lokalizace a mapování pomocí mobilního robotu* , 2008
- [22] Jan Bjelka, *Návrh a realizace metody mapování okolí pro mobilní robot* , 2007
- [23] Zbyněk Winkler, *Odometrie*, <https://robotika.cz/guide/odometry/cs>, 2005-12-05
- [24] Andreas Nuchter, Hartmut Surmann, Kai Lingemann, Joachim Hertzberg *6D SLAM with an Application in Autonomous Mine Mapping*, 2004
- [25] C. Baker, A. Morris, D. Ferguson, S. Thayer, C. Whittaker, Z. Omohundro, C. Reverte, W. Whittaker, D Hahnel and S. Thrun *6D SLAM with an Application in Autonomous Mine Mapping*, 2003
- [26] Zbyněk Winkler, *Senzorická mapa*, <https://robotika.cz/guide/gridplan/cs>, 2003-12-03

- [27] Jan Chudoba, *IMRnav documentation*, 2016
- [28] Martin Novotný, *Robustní postupy řízení formací robotu*, 2017
- [29] <http://www.sfbtr8.uni-bremen.de/project/r3/research.html>
- [30] Bc. Lukáš Hess, *Návrh dvoukolového autonomního robota*, 2013



## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

**Student:** Jiří K o k t a n

**Studijní program:** Kybernetika a robotika (bakalářský)

**Obor:** Robotika

**Název tématu:** Mapování terénu pozemním robotem

### Pokyny pro vypracování:

Cílem práce je návrh a implementace metody mapování okolního terénu robotem vybaveným rozmítanými laserovými dálkoměry a inerciálními senzory (inklinometr, akcelerometr, gyroskop) tak, aby vytvořená mapa prostředí sloužila pro bezpečnou a bezkolizní autonomní navigaci robotu v prostředí.

Provedte rešerši prací zabývajících se podobnou tematikou a prostudujte metody pro stavbu 3D map se zaměřením na 3D mřížky obsazenosti prostoru.

Metodu implementujte jako kód v jazyce C/C++. Navrhněte vhodnou konfiguraci senzorů na robotu tak, aby optimálně pokrývaly okolní prostředí. Metodu otestujte v robotickém simulátoru, nebo na robotu ve venkovním prostředí.

### Seznam odborné literatury:

- [1] Roland Siegwart, Illah Reza Nourbakhsh and Davide Scaramuzza: Introduction to Autonomous Mobile Robots, MIT Press, 2011.
- [2] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, W. Burgard: OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees, Autonomous Robots, 2013.
- [3] Filip Jareš: Exploration of an Unknown 3D environment, Diploma thesis, CVUT - FEL, Prague, 2014.

**Vedoucí bakalářské práce:** Ing. Jan Chudoba

**Platnost zadání:** do konce letního semestru 2017/2018

L.S.

prof. Dr. Ing. Jan Kybic  
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.  
děkan

V Praze dne 24. 1. 2017