



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV AUTOMATIZACE A INFORMATIKY

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

LOKALIZACE POMOCÍ LASEROVÉHO DÁLKOMĚRU SICK

LOCALIZATION BY SICK RANGE-FINDER

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JAN ZHÁŇAL

VEDOUCÍ PRÁCE
SUPERVISOR

ING. TOMÁŠ MARADA, PH.D.

BRNO 2010

ZADÁNÍ ZÁVĚREČNÉ PRÁCE

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav automatizace a informatiky

Akademický rok: 2009/2010

ZADÁNÍ BAKALÁRSKÉ PRÁCE

student: Jan Zháňal

který studuje v **bakalářském studijním programu**

obor: **Aplikovaná informatika a řízení (3902R001)**

Ředitel ústavu Vám v souladu se zákonem c.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Lokalizace pomocí laserového dálkoměru SICK

v anglickém jazyce:

Localization by SICK range-finder

Stručná charakteristika problematiky úkolu:

Cílem této bakalářské práce je seznámit se s používanými metodami lokalizace, zvolit vhodnou metodu pro použití s laserovým dálkoměrem SICK LMS-291 a funkčnost této metody demonstrovat.

Cíle bakalářské práce:

1. Seznamte se s metodami lokalizace mobilních robotu na internetu.
2. Zvolte metodu lokalizace, která bude vhodná pro použití s laserovým dálkoměrem SICK LMS-291.
3. funkčnost metody demonstруйте.

Seznam odborné literatury:

[1] <http://www.doc.ic.ac.uk/~ajd/Robotics/RoboticsResources/SLAMTutorial1.pdf>

[2] <http://www.doc.ic.ac.uk/~ajd/Robotics/RoboticsResources/SLAMTutorial2.pdf>

[3] <http://www.robotika.sk/mains.php>

Vedoucí bakalářské práce: Ing. Tomáš Marada, Ph.D.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2009/2010.

V Brně, dne 23.11.2009

L.S.

Ing. Jan Roupec, Ph.D.
Ředitel ústavu

prof. RNDr. Miroslav Doupovec, CSc.
Děkan fakulty

LICENČNÍ SMLOUVA

ABSTRAKT

K základním vlastnostem mobilního robotu patří schopnost pohybu a lokalizace v okolním prostředí. Tato práce se zabývá lokalizací i mapováním okolí jedním robotem, také označovaná jako SLAM (Simultaneous Localization and Mapping), který bude využívat laserového dálkoměru SICK. Budou probrány úlohy související s pohybem robotů např. odometrie, mapy prostředí a jejich tvorba. Také bude probráno využití Kalmanova filtru v robotice. Na závěr práce bude zhotoven model lokalizace a mapování pomocí metody SLAM vytvořený v programu Matlab.

ABSTRACT

Motion and localization are basic attributes of mobile robots in the environment. In this thesis we focus on localization and also mapping environment by one robot (named as well SLAM) which will use laser range-finder SICK. There will be stated a lot about activities related to robot movements' i.e. odometry, maps and their making. There is Kalman Filter and his use in this document. In the end there will be made model of the SLAM method made by Matlab program.

KLÍČOVÁ SLOVA

Lokalizace, SICK, SLAM

KEYWORDS

Localization, SICK, SLAM

PODĚKOVÁNÍ

Touto cestou bych chtěl poděkovat za připomínky panu Ing. Tomáši Maradovi, Ph.D. a také za e-mailovou konzultaci při tvorbě simulace panu Williamu Greenovi z Filadelfie, USA.

OBSAH:

Zadání závěrečné práce	3
Licenční smlouva	5
Abstrakt.....	7
Klíčová slova	7
Poděkování.....	9
1. Úvod.....	13
2. Metody lokalizace mobilních robotů	15
2.1 Dead reckoning.....	15
2.1.1 Ackermanovo řízení.....	15
2.1.2 Diferenciální podvozek.....	17
2.1.3 Všesměrový podvozek.....	18
2.1.4 Odometrie	18
2.2 Aktivní majáky	19
2.2.1 GPS (Global Positioning System).....	19
2.2.2 Stacionární majáky	20
2.3 Druhy lokalizace.....	21
2.3.1 Kalmanův filtr.....	21
2.3.2 Monte Carlo	22
2.3.3 Markovova lokalizace.....	24
3. Využití dálkoměru SICK LMS-291	27
3.1 Laserový dálkoměr SICK LMS-291.....	27
3.2 Princip laserového měření	28
4. Metody tvorby map	31
4.1 Modely prostředí.....	31
4.2 Mapování	32
4.2.1 Historie.....	32
4.2.2 Základní úlohy při mapování	32
4.3 Metody mapování	34
4.3.1 SLAM (Simultaneous Localization and Mapping).....	35
4.3.2 Kalmanovy filtry	35
4.3.3 Algoritmus očekávané maximalizace (Expectation Maximization – EM)	36
4.3.4 Hybridní přístup	36
4.3.5 Mřížky obsazenosti	37
4.3.6 Objektové mapy	39
4.3.7 Mapování dynamického prostředí	39
4.3.8 Geometrický přístup	40
5. Model SLAMu	43
5.1 Úvod	43
5.2 Výsledky získané z modelu	43
5.3 Zhodnocení možnosti využití	44
6. Závěr.....	45
Literatura	47

1. ÚVOD

Robot bývá obvykle elektromechanický stroj, který je schopen „vnímat“ svoje okolí a toto okolí ovlivňovat. Mobilní robot je takový robot, který se dokáže pohybovat v prostoru a může mít rozdílnou míru autonomie – od teleoperované (řízené dálkově) až po plně autonomní. Druhá zmíněná je v dnešní velmi překotně vyvíjenou oblastí robotiky a snaží se dosáhnout toho, aby se robot mohl pohybovat v prostředí a případně vykonávat činnosti bez zásahu lidské obsluhy.

Tato práce se zabývá možnostmi lokalizace a mapování robotů za pomoci laserového dálkoměru SICK LMS 291.

Zde představené metody se používají, nebo byly používány v robotice. Pro využití některých stačí jednoduchý automat, jiné mohou být velmi náročné i pro dnešní výkonné počítače. Což bývá jedním z problémů – málokterý robot si s sebou může vézt obří zdroj, nebo rozměrný a na energii náročný výpočetní systém.

V úvodních kapitolách jde o shrnutí jednotlivých metod lokalizace, později půjde i o mapování, jelikož uvažovaný robot by měl být schopen si své prostředí prozkoumat.

2. METODY LOKALIZACE MOBILNÍCH ROBOTŮ

Lokalizací robota v okolí se myslí jeho „povědomí“ o tom, kde je. Člověk, který přijde na známé místo, vyvolá ze správného místa své paměti mapu či plánec oné např. budovy. Robot většinou pro svůj pohyb také potřebuje mapu, řekněme jí mapa světa – jeho světa, kde se bude pohybovat, aby se dokázal orientovat a vyhýbat se překážkám. Poté již stačí jen správně určit svou pozici. Což zatím není tak jednoduché, jak by se mohlo zdát (jedním pohledem si přece člověk změří vzdálenost a zároveň i další vlastnosti okolí), protože zatím nikdo nevymyslel žádnou stejně dokonalou technickou pomůcku, jako je lidské oko, musí si robot pomoci po svém. Problémy při tom mohou způsobit i zdánlivé maličkosti – lidská noha, na nesprávném místě, kluzká podlaha, nebo třeba stěhování nábytku. V této kapitole budou probrány jednotlivé metody lokalizace.

2.1 Dead reckoning

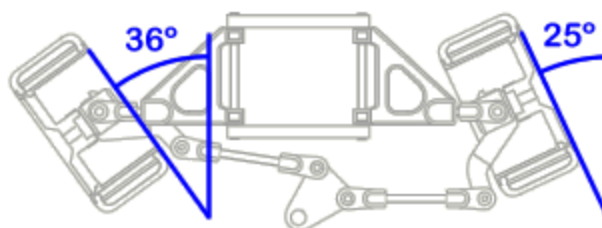
Je nejjednodušší metodou lokalizace a je definována takto „Matematická procedura pro určování současné pozice objektu pomocí postupného přičítání dráhy díky známému kurzu a rychlosti v čase“. Jednoznačnou nevýhodou je, že při výpočtu aktuální pozice robota může docházet k chybám, které se navíc kumulují. Proto se zde také nepoužívá termín *určit*, ale *odhadnout* pozici.

Tato metoda je i přes svoji relativně vysokou chybovost hojně využívána, ale většinou jako součást složitější lokalizaci, kdy jsou její informace skládány společně s dalšími senzory.

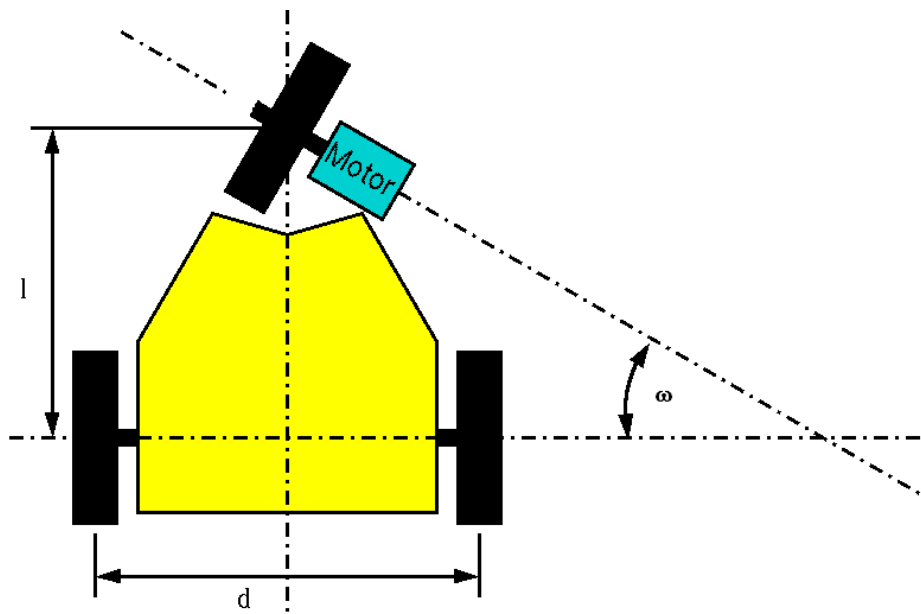
Základní implementace této metody se nazývá *odometrie*[4]. Což je proces, který popisuje transformaci dat poskytnutých enkodéry na změnu pozice a orientace robota[5]. Pro výpočty za pomoci odometrie je nutné znát geometrii robota, přesněji toho, jakého jsou schopni pohybu. Dělí se do těchto kategorií.

2.1.1 Ackermanovo řízení

V závislosti na natočení říditelného kolečka se tříkolka pohybuje buď po přímce (to když je kolečko rovnoběžně s osou robota) anebo po kružnici. Pro zjištění změny pozice je třeba dvou údajů – orientace řídicího kolečka a jeho ujetá vzdálenost. Necht' je referenčním bodem střed nápravy pevných kol. Předpokládáme-li nenulové natočení řídicího kola, a tedy pohyb po kružnici, bude nás pravděpodobně zajímat střed a poloměr této kružnice. Jak je vidět z obr. 2., střed kružnice bude ležet na ose pevné nápravy (vlevo, či vpravo podle směru natočení)[5].



Obr. 1 Ackermanovo řízení.



Obr. 2 Model pro výpočet změny polohy.

Samotné Ackermanovo řízení obr. 1. se používá v případě řízení podvozku se dvěma řídicími koly - v zatáčce je nutno natočit každé z nich jinak. Využívá se u větších robotů. V případě, že vnitřní kolo opisuje správnou – menší kružnici než kolo vnější, můžeme pro výpočet použít stejné vztahy jako pro tříkolku z obr. 2.

Vztahy:

Poloměr R je roven:

$$R = l / \tan(\omega),$$

kde l je vzdálenost náprav a ω úhel natočení.

Po této kružnici robot urazí úhel θ , který bude odpovídat i změně jeho orientace:

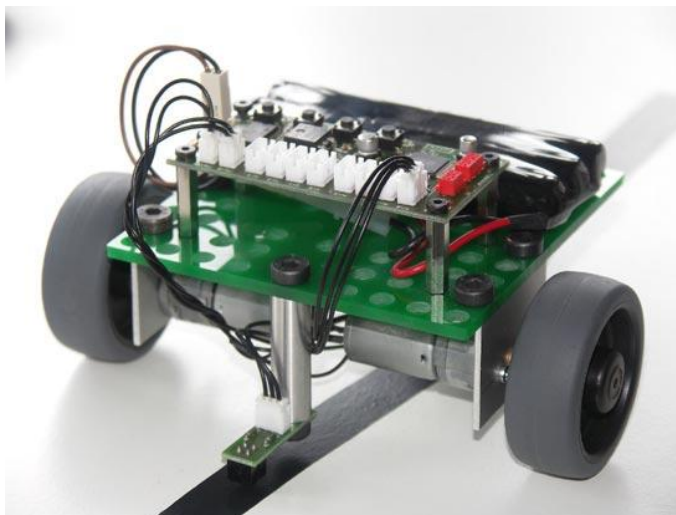
$$\theta = dF \cdot \cos(\omega) / R$$

Změna pozice d_x : $d_x = R - R \cdot \cos(\theta)$

Změna pozice d_y : $d_y = R \cdot \sin(\theta)$

2.1.2 Diferenciální podvozek

Nejpoužívanější pro svou jednoduchost, obsahuje dvě kola, která bývají poháněna malým stejnosměrným elektromotorem nebo krokovým motorem jako na obr. 3. Pokud se obě kola točí stejným směrem stejně rychle, jede robot rovně. Pokud by se točily obráceně, bude se točit na místě.



Obr. 3 Diferenciální podvozek.

Pokud je rozdíl rychlostí mezi nápravami nenulový, ale obě nápravy se točí na stejnou stranu, pohybuje se i tento robot po kružnici. Vzdálenost středu této kružnice je dána poměrem obou rychlostí (b je rozchod a v_L a v_R jsou rychlosti):

$$R = \frac{1}{2} b \frac{v_L + v_R}{v_L - v_R}$$

Pokud levé kolečko ujede vzdálenost d_L a pravé d_R změní se orientace o úhel θ :

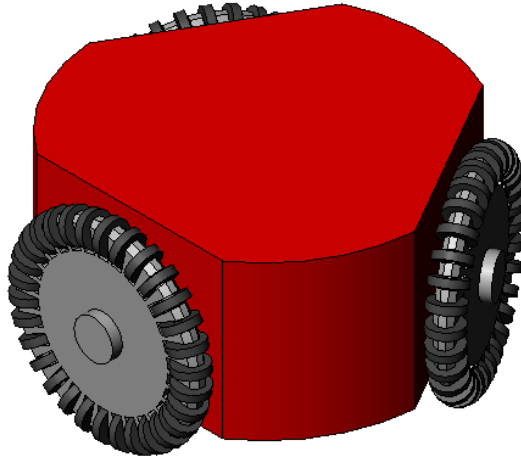
$$\theta = \frac{d_L - d_R}{B}$$

Celková ujetá vzdálenost d (počítaná pro střed poháněné osy) je pak:

$$d = \frac{d_L + d_R}{2} .$$

2.1.3 Všesměrový podvozek

Všesměroví roboti se mohou pohybovat v daném okamžiku všemi směry. V praxi se toho dosahuje speciálními kolečky, která povolí i pohyb podélný s osou kola. Minimální počet kol jsou potom 3 a všechna musí být hnací.



Obr. 4 Všesměrový podvozek (malá kolečka umožní pohyb po ose velkého kola).

Informace potřebné k výpočtu změny pozice zahrnují jednotlivé enkodéry na kolech d_1 , d_2 , d_3 a vzdálenost kol od středu robota b .

$$d_1 = d_x + \theta b$$

$$d_2 = -0.5 d_x + 0.867 d_y + \theta b$$

$$d_3 = -0.5 d_x - 0.867 d_y + \theta b$$

2.1.4 Odometrie

Odometrie je proces, který popisuje transformaci dat poskytnutých *enkodéry* na změnu pozice a orientace robota. Její základní myšlenkou je integrace informací, které s časem přibývají. Každý údaj z *enkodéru* obsahuje malou zanedbatelnou chybu. Chyba je sama o sobě malá a na první pohled nemá výrazný vliv na výsledek. Problém ovšem nastává při větším počtu měření. Informace získané z *enkodéru* se akumulují. Stejně tak se akumulují i chyby, které jsou obsaženy v každém jednom měření.

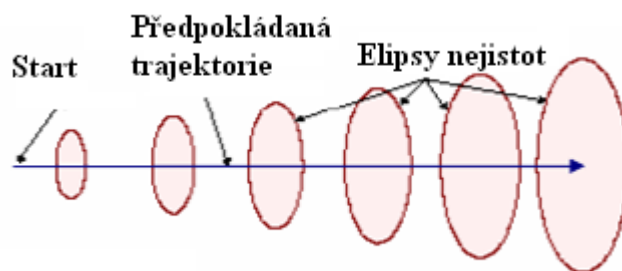
Systematické chyby:

- nerovnoměrné průměry poháněných kol
- skutečný rozvor kol se liší od nominálního
- nesouosost kol
- konečné rozlišení *enkodéru*

Nesystematické (náhodné) chyby:

- pohyb po nerovném povrchu
- pohyb přes neočekávané předměty na podlaze
- prokluz kol zapříčiněný různými vlivy (kluzký povrch, rychlé zatočení atd.)

Kvůli těmto chybám, se často využívá tzv. výpočtová navigace, (Dead Reckoning). Kdy je vypočtená pozice obklopena chybovou elipsou – popisuje aktuální nejistotu polohy robotu a s větší uraženou dráhou se zvětšuje, viz obr. 6.



Obr. 5 Elipsy nejistot výpočtové navigace.[6]

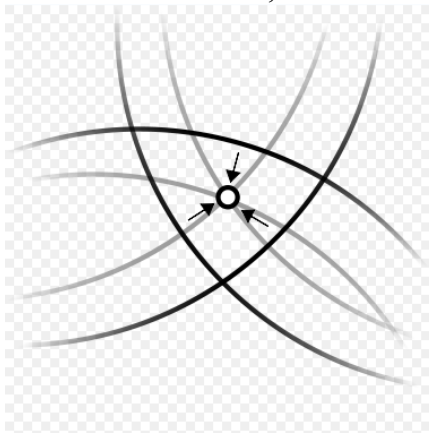
2.2 Aktivní majáky

Další možnou metodou je lokalizace využívající aktivních prvků (majáků). Což je i lokalizace prostřednictvím GPS, která využívá družic rozmístěných na oběžné dráze. Ve vnitřních prostorech se využívá majáků vysílajících ultrazvukové nebo světelné signály. Poloha robotu je určena vzhledem k jednotlivým majákům a není třeba mít mapu prostředí.[7]

2.2.1 GPS (Global Positioning System)

Dnes hojně rozšířený, byl vyvinut americkou armádou. Využívá systému 32 družic. Nevýhodou GPS modulů je nemožnost použití v uzavřených prostorech – žádný signál, nebo např. v listnatém lese, kde dochází ke zkreslení signálu družic a tím pádem k výpočtu špatné polohy. Zjednodušeně ho lze popsat jako *družicový radiový dálkoměrný systém*:

- *Dálkoměrný (trilaterární) systém* je takový, kdy se poloha nějakého objektu určuje ze vzdáleností od bodů se známou polohou. Např. v krajině lze určit polohu pomocí mapy a dalekohledu, který umí změřit vzdálenost od pozorovaného objektu. Změříme vzdálenost ke dvěma význačným objektům a kružítkem na mapě nakreslíme kolem každého objektu kružnici o změřeném poloměru. Zjišťovaná poloha je v jednom z průsečíků obou kružnic, viz obr. 6.



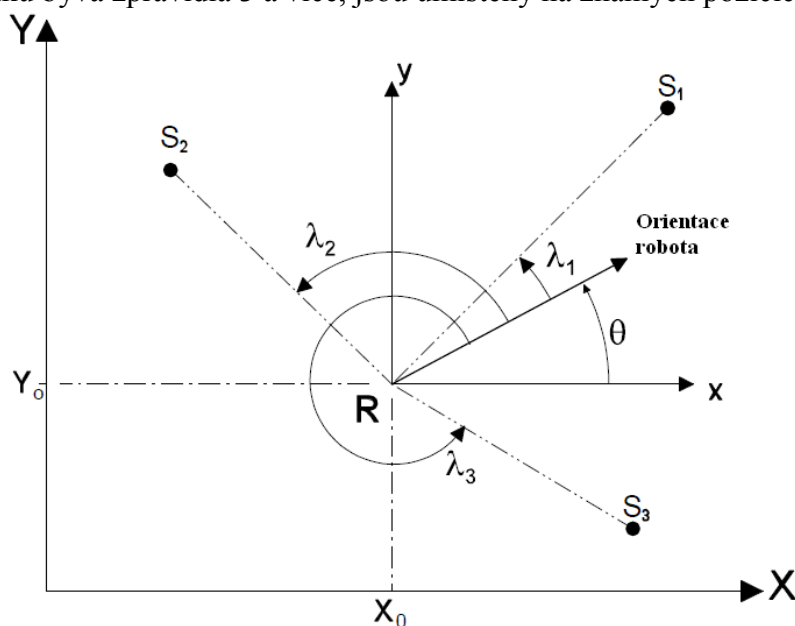
Obr. 6 Průsečík kružnic – vzdáleností od „majáků“.

- *Rádiový* systém pro měření určitého parametru využívá rádiových vln. „Rádiový dálkoměrný“ systém k měření vzdálenosti využívá rádiových vln takto: Do bodu se známou polohou je umístěn vysílač, který vysílá rádiové vlny s časovými značkami. V bodě, jehož poloha se měří, umístíme přijímač, který porovnává časové značky se svými „hodinami“. Tím je možno změřit *zpoždění*, tj. jak dlouho trvalo rádiové vlně, než k přijímači dorazila. Protože se rádiové vlny pohybují známou rychlostí, stačí pro výpočet požadované vzdálenosti vynásobit změřené zpoždění touto rychlostí.
- *Družicový* je systém označován proto, že body se známou polohou jsou družice obíhající Zemi. Aby bylo možno určit polohu družic, musí být v jejich vysílání nejen časové značky, ale i parametry dráhy dané družice.

2.2.2 Stacionární majáky

Využívá se uvnitř budov, její podstata je ale velmi podobná GPS. V místnosti musí být umístěny majáky vysílající zvukové, ultrazvukové, světelné nebo rádiové signály, ze kterých poté robot počítá svoji polohu. Předností tohoto řešení je relativně vysoká přesnost, rychlost a jednoduchost výpočtu pozice robota. Základním omezením této metody je dosah a zkreslení signálů z majáků. Také tímto snížíme autonomii robota – je odkázán na signály z okolí. [7]

Trilaterace - vyjadřuje pozici vozidla na základě měření vzdálenosti od známých majáků. Majáků bývá zpravidla 3 a více, jsou umístěny na známých pozicích (i GPS).



Obr. 7 Triangulace – rotující senzor na robotovi měří 3 úhly mezi vozidlem a 3 majáky

Triangulace - také využívá 3 a více majáků, se známou pozicí. Robot počítá svoji polohu pomocí měření úhlů (obr. 7.), ze kterých přichází jednotlivé signály.

2.3 Druhy lokalizace

Lokalizaci lze rozdělit na 2 druhy podle toho, jakým způsobem odhadují pozici robotu.

Lokální metody (sledovací) se zaměřují na kompenzaci chyb odometrie. Hlavní podmínkou pro použití této metody je, že musíme znát počáteční polohu robotu. Hlavní nevýhodou bývá, že si většinou nedokáže poradit s náhodnou změnou pozice (např.: z důvodu kolize s okolím, ručním posunutím apod.). Tato lokalizace většinou probíhá ve 4 fázích:

- přemístění se
- odhadnutí nové polohy na základě údajů z enkodérů a znalosti předchozí pozice
- provedení senzorických měření
- porovnání vypočtené pozice s modelem světa a korekce polohy [11]

Globální metody jsou používány tam, kde robot předem nezná svou polohu. Což se může stát např. po jeho startu. Měly by být účinnější a přesnější než lokální metody.[11]

Tyto metody používají výsledky senzorického měření okolí a porovnávají je s modelem (mapou prostředí). Tím dochází k eliminaci chyb *odometrie*. Naprostá většina známých a aplikovaných metod je založena na teorii pravděpodobnosti. Víme, že každé senzorické měření obsahuje chybu. Tato chyba by měla mít *gaussovské rozložení*. Na základě této skutečnosti byly uvedeny různé metody. Metoda *Monte Carlo*, jejíž charakteristickou vlastností je způsob reprezentace hustoty pravděpodobnosti popisující odhad pozice robota, nebo *Kalmanův filtr*, který je založen na principu odhadu plovoucího průměru. Další je tzv. *Markovova lokalizace*. Jejíž hlavní myšlenkou je výpočet pravděpodobnostního rozdělení na všech možných pozicích v prostředí.

2.3.1 Kalmanův filtr

Zjednodušeně by se dalo říci, že Kalmanův filtr je „vylepšený“ odhad plovoucího průměru. Toto vylepšení spočívá v rozdělení algoritmu do dvou kroků:

- předpověď nového stavu
- korekci integrací nového měření

V případě odhadu plovoucího průměru, je predikce stavu pro následující krok velmi jednoduchá – předpokládalo se, že se nemění. V tuto chvíli je kromě toho třeba si pamatovat, jak moc našemu odhadu nevěříme => S každou předpovědí se musí zvýšit nedůvěryhodnost odhadu[13].

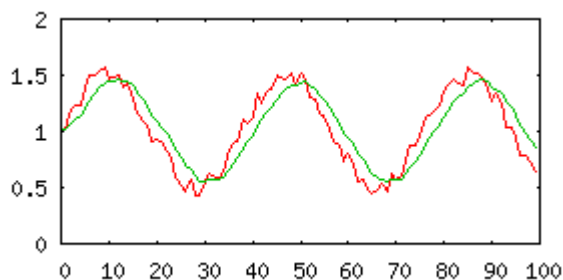
$$P = P + Q$$

Korekci nově predikovaného stavu provedeme pomocí nového měření:

$$\text{avg} = \text{avg} + K \cdot (y[n] - \text{avg}),$$

kde K je určitá váha. Kalmanův filtr definuje K , Kalmanovo zesílení (*Kalman gain*) jako:

$$K = P / (P + R)$$



Obr. 8 Kalmanův filtr odpovídající $k=4$. [13] Zeleně odhad, červeně skutečnost.

kde R je míra „nedůvěryhodnosti“ měření (nebo také množství šumu). Kalmanovo zesílení je vždy menší než 1. Pokud by R bylo rovno 0, znamená to, že měření má naši 100%ní důvěru, a proto aplikujeme celou korekci. Jinak aplikujeme vždy pouze její část, která je dána poměrem nedůvěryhodnosti odhadu stavu a měření. Po korekci vlastního stavu bychom rádi opravili i jeho aktuální míru nedůvěryhodnosti. Tato korekce je definována jako:

$$P = (1 - K) * P$$

Tento vztah lze interpretovat tak, že pokud jsme použili jednu desetinu z korekce stavu, zmenšila se nedůvěryhodnost našeho odhadu na devět desetin. Pokud zvolíme $P=0,9$, $Q = 0,1$ a $R = 9$, degeneruje náš příklad na odhad plovoucího průměru pro délku okna 10. [13]

Zahrnutí informací relativního charakteru odpovídá predikčnímu kroku, zahrnutí absolutních měření naopak kroku korekčnímu.

Nicméně, musíme splnit určité předpoklady, které plynou z charakteru výpočtů probíhajících „uvnitř“ filtru. Filtr předpokládá, že pravděpodobnostní hustota reprezentující neurčitost stavu může být charakterizována střední hodnotou a rozptylem. Stejný předpoklad je kladen i na uvažovaná měření.

Pokud bychom chtěli využít měření, které tento požadavek nesplňuje, Kalmanův filtr použít nemůžeme. [12]

2.3.2 Monte Carlo

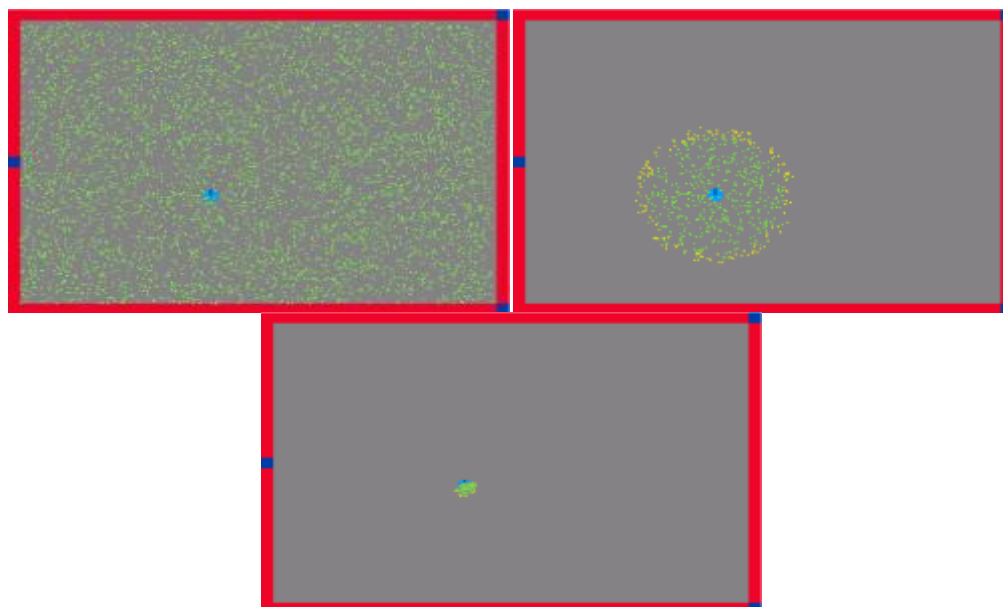
Lokalizace Monte Carlo používá metody, které byly vyvinuty v sedmdesátých letech a v poslední době se opět začaly používat například pro sledování cíle, statistiku či strojové vidění. Lokalizační metoda Monte Carlo je částicový filtr, ve kterém je hustota pravděpodobnosti reprezentována pomocí sady vzorků, které jsou s její pomocí náhodně vytvořeny. Za použití reprezentace založené na vzorcích získáváme lokalizační metodu, která dokáže reprezentovat libovolné statistické rozdělení. Tím pádem je metoda schopna lokalizovat robot globálně. Metoda Monte Carlo je schopna provést lokalizaci robotu bez toho, že by znala počáteční pozici, což umožňuje robotu lokalizovat se po nárazu do překážky, či jiného i pohybujícího se objektu. [14]

Při lokalizaci nás nejvíce zajímá aktuální odhad stavu robotu. Stavový vektor sestává z pozice a natočení robotu. Funkce hustoty pravděpodobnosti se používá k reprezentaci veškerých znalostí, které o stavu máme, a na jejím základě jsme schopni odhadnout aktuální pozici. K lokalizaci musíme hustotu vypočítat rekurzivně. K tomu dochází ve dvou fázích:

Fáze predikce: V první fázi používáme model pohybu k odhadu aktuální pozice robotu. V našem případě je odhad proveden na základě odometrie. Odometrie je způsob výpočtu pozice na základě informací o otáčení hnacích kol.

Fáze aktualizace: Ve druhé fázi používáme model měření k integraci informací ze senzorů kvůli korekci odhadu pozice.

Jak probíhá lokalizace metodou Monte Carlo je znázorněna na následujících obrázcích. Při prvním průchodu je inicializována tím, že vytvoří 5000 vzorků reprezentujících hustotu pravděpodobnosti.



Obr. 9 Inicializace 5000 vzorků, vedle globální lokalizace po prvním průchodu programem. Dole odhad pozice po 15 průchodech.

Výhody MCL dle autorů publikace zmíněné pod číslem 15:

- schopnost reprezentovat *multimodální* hustoty pravděpodobnosti z čehož vyplývá schopnost globální lokalizace robota.
- drasticky redukuje potřebnou paměť v porovnání s Markovovou lokalizací a zároveň dokáže integrovat naměřené hodnoty s nezanedbatelně vyšší frekvencí
- je přesnější než Markovova lokalizace s pevnou mřížkou – vzorky nejsou diskretizovány
- je jednoduchá na implementaci

Algoritmus

Má dvě fáze nebo tři fáze: *predikce*, *korekce* a používá se i *převzorkování*.

Výsledkem *predikce* (např.: posun vzorků na základě informací z odometrie) je nová sada vzorků reprezentujících upravenou hustotu pravděpodobnosti na základě informací o změně pozice. Využití těchto informací by mohlo spočívat v prostém posunu každého vzorku o příslušnou hodnotu. Výsledkem by byla v podstatě totožná hustota pravděpodobnosti jako ta výchozí, pouze příslušně posunutá.[12]

Nicméně, takto by situace vypadala pouze v případě, že bychom si informací o změně pozice byli stoprocentně jisti. V situaci, kdy i tato informace je svým způsobem nejistá, potřebujeme toto zohlednit. Potřebujeme tedy nějaký ekvivalent sčítání rozptylů, ovšem pro reprezentaci hustoty pravděpodobnosti pomocí vzorků. Toho docílíme tak, že neposuneme všechny vzorky o stejnou hodnotu. Ke každému z posunů přidáme navíc nějaký náhodný šum, jehož míra bude odpovídat míře nedůvěry, kterou k této informaci chováme. Tím získáme hustotu pravděpodobnosti, která bude svým způsobem trochu rovnoměrnější.[12]

V tuto chvíli je třeba *korekce*. To spočívá v modifikaci vah jednotlivých vzorků. Necht' l , p je vzorek. Potom $p \leftarrow \alpha P(s | l)$, kde s je výsledek senzorického měření a α je normalizační konstanta.

Každý je tudíž ohodnocen podle toho, jak získané měření odpovídá předpokladu pro příslušnou pozici hodnoceného vzorku. Např.: zvýšíme váhu všem takovým vzorkům, které se v mapě nachází na pozici, kde je pod robotem barva shodná s barvou naměřenou senzory a naopak snížíme, kde se neshoduje.[12]

Tyto fáze se opakují pro každé měření, které využíváme k odhadu pozice. Neustálým opakováním korekce, ale časem dojde k tomu, že většina vzorků bude mít zanedbatelné váhy a několik málo naopak váhy obrovské. Vzorky s malou vahou odpovídají pozicím, kde se robot pravděpodobně nevyskytuje, a proto jsou pro nás zbytečné. Naopak pozice s vzorky s vysokou vahou jsou více pravděpodobné. Proto se často využívá ještě fáze třetí – *převzorkování*. Účelem převzorkování je v podstatě „zahodit“ vzorky s hodně malou vahou a vzorky s vahou velkou naopak rozdělit na vzorků několik. Je vhodné, aby se hustota pravděpodobnosti tímto krokem změnila co nejméně.[12]

2.3.3 Markovova lokalizace

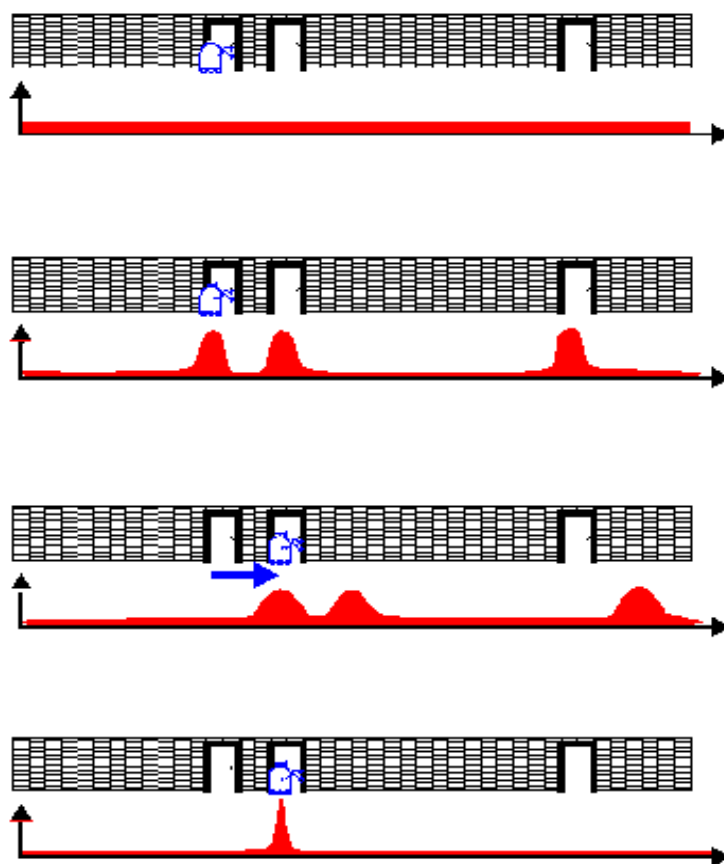
Je to pravděpodobnostní algoritmus. Raději než udržovat jednu hypotézu o tom, kde ve „světě“ může robot být, Markovova lokalizace udržuje pravděpodobnostní rozdělení nad celým prostorem hypotéz.

Jednoduché vysvětlení si můžeme předvést podle obr. 10. Pro jednoduchost budeme uvažovat 1-D prostor, takže robot bude pohybovat pouze v horizontální rovině (nebude moci zatočit).

Nyní budeme předpokládat umístění robota kdekoli v prostředí bez znalosti toho, kde je. Markovova lokalizace reprezentuje tento stav nejistoty stejným rozložením přes všechny pozice, jak je vidět v první části obr. 10.

Dále použije robot své senzory a zjistí, že se nachází u dveří. Markovova lokalizace mění důvěru – zvýšením pravděpodobnosti pro místa u dveří a snížením všude jinde (druhý diagram). Je vidět, že výsledek rozdělení je pro globální lokalizaci multimodální a dále, že místa vedle dveří stále mají nenulovou pravděpodobnost což je způsobeno šumem na senzorech. Také je jasně vidět, že z toho, že je robot u dveří, ještě nemůže usoudit, u kterých. Proto musí dojít k posunu robota, změní se tím i pravděpodobnost pro celé rozložení – počítáme se ztrátou informací – šum ze senzorů. Rozložení je nyní hladší a nejasnější jak je vidět na diagramu 3.

Jasně rozhodnutí může přijít až robot za pomoci senzorů zjistí, že se znovu nachází u dveří. Což ho jednoznačně ujistí o jeho aktuální pozici. V tuto chvíli je veškerá pravděpodobnost blízko jednoho místa. Robot si je „jistý“ svojí pozicí.[16]



Obr. 10[16] Markovova lokalizace v 1-D prostoru, červeně je vyznačen stav rozložení nejistot pozice robotu, modře samotný robot a modrá šipka znázorní jeho pohyb.

Základní zápis

Poněkud formálnější zápis toho, co jsme prošli výše, může znít takto:

Pokud l je třídimenzionální proměnná, pro kterou platí, že: $l = (x, y, \theta)$, kde x, y jsou osy Kartézského systému a θ je směr pohybu, l_t je poloha robota v čase t , L_t je náhodná proměnná, která vyjadřuje pozici robota, $\text{Bel}(L_t)$ je pozice, kterou robot předpokládá jako místo svého výskytu – pravděpodobnostní rozložení na prostoru pozic.

$$\text{Bel}(L_t = l)$$

je pravděpodobnost, že robot přidělí do rozložení, že jeho pozice v čase t je l . Což se upravuje podle odometrie a dalších použitých senzorů.

$$d_i = \{a_i - \text{odometrie}, s_i - \text{další senzory}\}$$

Potom náš výsledek je:

$$P(L_t = l \mid d) = P(L_t = l \mid d_0, \dots, d_t)$$

3. VYUŽITÍ DÁLKOMĚRU SICK LMS-291

3.1 Laserový dálkoměr SICK LMS-291

Německá společnost SICK je jedním z hlavních výrobců průmyslových senzorů. Především pak na optická řešení v automatizaci, bezpečnostní systémy, automatické identifikace a měření.

SICK LMS 291 je extrémně přesný a rychlý dálkoměr určený pro průmyslové prostředí. Jedná se o bezkontaktní měřicí systém, který skenuje svoje okolí ve dvou dimenzích (tzv. laserový radar - „LIDAR“). Na rozdíl od některých jiných skenovacích systémů ke své funkci nevyžaduje žádné speciální reflexní nebo poziční značky.[8]



Obr. 11 SICK LMS 291.

Výstup z LMS je implementován sériovým rozhraním a dostaneme sadu hodnot. V závislosti na využití je můžeme:

- zobrazit na PC pomocí dodávaného softwaru „LMSIBS“, nebo pomocí vlastní aplikace, např. pro detekci polohy objektů a jejich velikosti.
- Získávat a vyhodnocovat v reálném čase na počítači. Integrované vyhodnocovací rutiny také dovolují využít LMS jako senzoru. Specifikace nejdůležitějších parametrů LMS 291 je uvedena v tabulce.[8]

SICK LMS 291	
Maximální dosah	80m
Zorné pole	100° / 180°
Úhlové rozlišení	0.25° / 0.5° / 1°
Doba odezvy	13 ... 53 ms
Frekvence skenování	75 Hz
Rozlišení	10 mm
Systematická chyba	+/- 35 mm
Datové rozhraní	RS 232, RS 422
Přenosová rychlost	9,6 / 19,2 / 38,4 / 500 kBaud
Napájecí napětí	24 V DC +/- 15%
Příkon	20W
Hmotnost	4,5 kg
Rozměry	156 x 155 x 210 mm

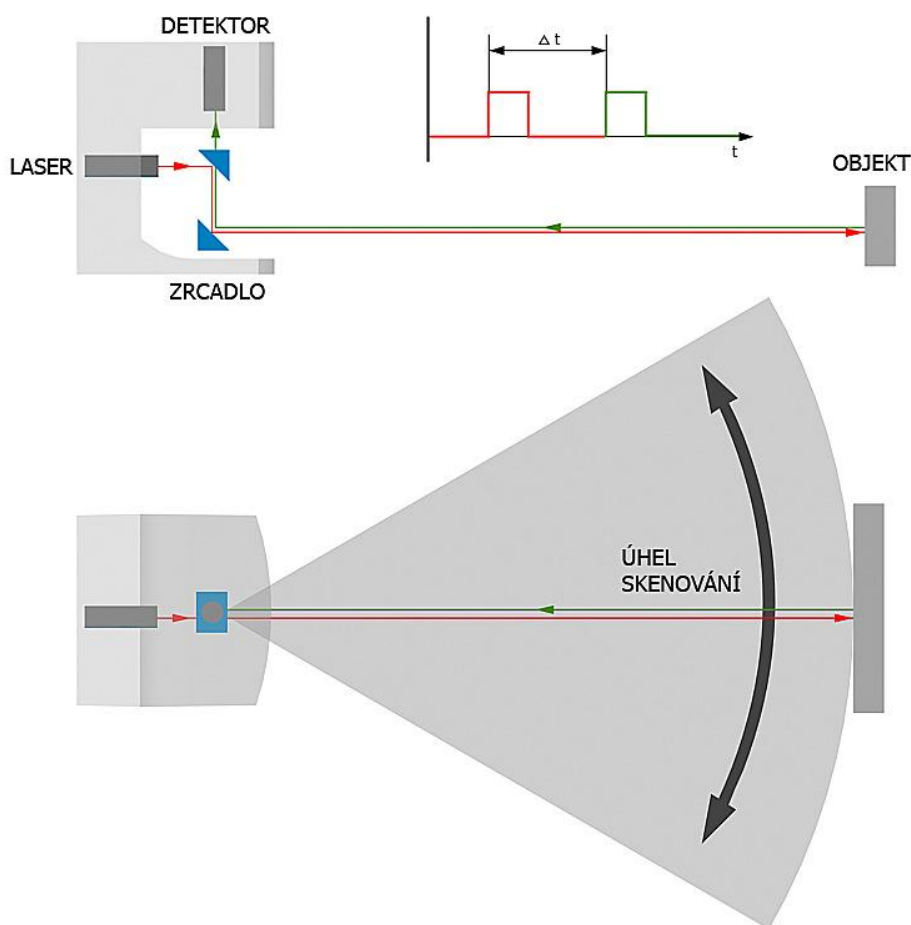
Tab. 1.: Technická specifikace LMS 291. [9]

3.2 Princip laserového měření

Základním principem bezkontaktního měření vzdálenosti je vyslání signálu (radiového, ultrazvukového nebo optického) na povrch objektu a zpracování signálu odraženého. Použití laseru je lepší než ostatní typy signálu z následujících důvodů:

- Laser poskytuje koncentrovaný paprsek s velmi malým rozptylem a odchylkami. Rádiové a ultrazvukové vlny nemohou takto malého rozptylu dosáhnout.
- Laserový paprsek má velkou intenzitu a díky malému rozptylem si tuto intenzitu zachovává na velké vzdálenosti.
- Monochromaticnost laserového paprsku vede ke snazšímu zpracování signálu. Laserový paprsek je produkován laserovou diodou a příslušnou optikou. Odražený signál je detekován pomocí lavinové fotodiody. Následně jsou použity vhodné členy pro zesílení, zpracování a filtraci signálu. Jedním z nejčastějších matematických principů měření vzdálenosti je měření doby letu laserového pulzu. Princip této metody je jednoduchý. Laserový impulz je vyslán do okolí a následně je změřen čas, který paprsku zabere cestu k cíli, odražení a cestu zpět.

Schéma 2D skeneru je zobrazeno na obr. 12. Laserový paprsek znázorněný šipkami je vychýlen rotujícím zrcadlem. Tímto způsobem může 2D skener poskytnout body, které leží v půlkruhové rovině odklonu paprsku. Obrys cílového objektu je získán jako posloupnost přijímaných impulzů.[11]



Obr. 12 Princip laserového měření.[10]

Dosah skeneru je závislý především na reflektivitě skenovaných objektů a na síle vysílaného paprsku. LMS používá přijatou energii k vyhodnocení vzdálenosti a porovnává ji s vnitřními referencemi. Hodnoty přijaté energie (reflektivity) jsou počítány pro vlnovou délku 905 nm.

Reflektivita je schopnost objektu odrážet světlo, kterou nelze snadno zaznamenat. Naměřené hodnoty energetické úrovně z objektu se odvíjejí od reflektivity objektu, ale nejsou stejně jako jeho absolutní reflektivita.

Tmavý testovací objekt v blízkosti skeneru může mít stejnou energetickou hodnotu, jako světlý objekt ve větší vzdálenosti. Hodnoty reflektivity mohou pomoci popsat strukturní přechody materiálu ve stejné vzdálenosti. Lze ji tak využít např. pro zjišťování přechodu z bílé do černé ve stejné vzdálenosti měření.

4. METODY TVORBY MAP

Až do této chvíle bylo v souvislosti s pohybem robota uváděno, že zná své okolí, že využívá nějakou „mapu světa“. Takovou mapu může tvůrce robota samozřejmě přímo implementovat, jenže potom je robot omezen pouze prostředím, které zná. Navíc bude mít velké problémy se lokalizovat, jestliže nějak zásadně změníme jeho okolí například postavení nábytku v místnosti. Proto se tato kapitola bude zabývat tvorbou mapy.

4.1 Modely prostředí

Při tvorbě mapy je třeba řešit některé problémy - reprezentace modelu prostředí, propojení senzorických dat, lokalizace robota v prostoru a naplňování trajektorie pohybu. Druh použitého modelu prostředí závisí na typu prostředí a na tom, jakou úlohu má robot vykonávat. Například pro pohyb robota ve venkovním terénu mimo cesty není vhodné používat dvojrozměrnou planární mapu. Model prostředí musí respektovat nepřesnost senzorických dat i nepřesnost určení polohy robota. Podle toho jak je model abstraktní, lze mapy prostředí rozdělit:

- senzorické - obsahují pouze data ze senzoru
- geometrické - mapu tvoří 2D nebo 3D geometrické objekty vypočtené ze senzorických dat
- mapy s lokálními vztahy - obsahují i informace o funkčních, strukturálních nebo sémantických relacích mezi geometrickými objekty, které spolu sousedí,
- topologické - obsahují informace o vzájemných vztazích mezi objekty a jejich polohou v měřítku celé mapy,
- sémantické - obsahují funkční označení prvků mapy.

Jedním z prvních řešených problému v mobilní robotice je mapování uvnitř budov. K tomu lze využít dvojrozměrné mřížky obsazenosti a pravděpodobnostní mřížky, ve kterých je dvojrozměrný prostor rozdělen na jednotlivé buňky, které obsahují informaci o obsazenosti, respektive o míře pravděpodobnosti obsazení buňky překážkou. Vzhledem k tomu, že budovy bývají strukturované a skládají se převážně ze vzájemně kolmých rovin, lze je s výhodou zastoupit soustavou úseček. To sice vyžaduje více výpočtů nad senzorickými daty, ale snižuje paměťové nároky pro uložení výsledné mapy.

K práci s neurčitostí jsou používány Kalmanovy filtry. Při mapování vnějších prostředí se začaly používat 2,5-D mřížky, kde jednotlivé buňky obsahují navíc informaci o sklonu povrchu. K mapování trojrozměrného světa se dále používají shluky bodů, 3-D mřížky a 3-D sítě. Ve 3-D mapách je třeba zpracovávat mnohem větší objemy dat, proto nutné upravovat senzorická data do jednodušších objektů. Většina používaných modelu předpokládá statické prostředí, takže pro zahrnutí pohybujících se objektů musíme předchozí modely rozšířit o časový rozměr.

4.2 Mapování

4.2.1 Historie

Vzhledem k tomu, že v oblasti mapování již 20 – 30 let probíhá intenzivní vývoj, existují spolehlivé metody pro mapování statického, strukturovaného prostředí, které ale nesmí být příliš rozlehlé.[17]

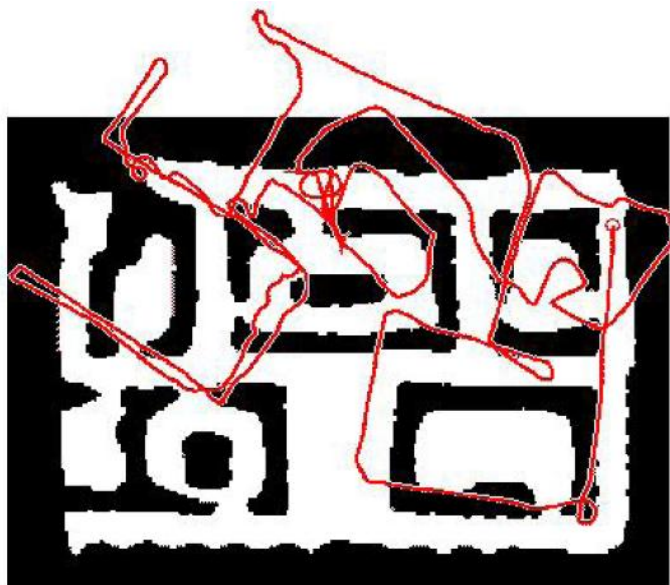
Jedním z prvních řešení byl Elfesův a Moravcův algoritmus mapování využívající mřížku obsazenosti. Tento přístup byl poté využit v mnoha dalších robotických systémech. Metrické mapy byly uvedeny Chatilou a Laumondem, kteří využívali mnohoúhelníky k popisu geometrie prostředí. Topologické mapy využívali při mapování Mataric, Kuipers. Mezi metrickými a topologickými mapami nikdy nebyla jasná hranice, protože téměř všechny přístupy stavějí na geometrických informacích. V praxi bývají metrické mapy podrobnější než topologické, což sice vyžaduje více výpočetních zdrojů, ale umožňuje to řešit některé další problémy.

V devadesátých letech začaly v mapování převažovat pravděpodobnostní techniky. Smith, Self a Cheesman uvedli ve svých článcích mocnou statistickou metodu pro současné řešení mapování a s ním souvisejícího problému lokalizace robotu ve vytvářené mapě. Od té doby je mapování označováno jako SLAM (Simultaneous Localization And Mapping) nebo CLM (Concurrent Mapping and Localization). Pravděpodobnostní přístupy se dají rozdělit na tři skupiny:

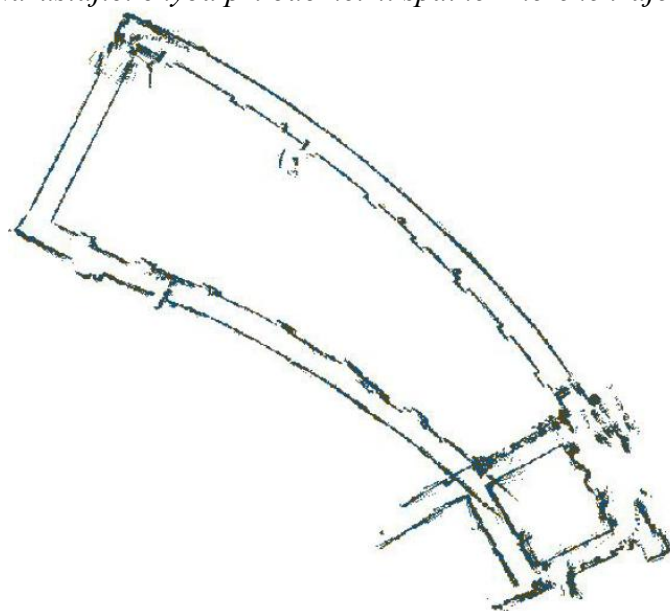
- využití Kalmanových filtrů k odhadu mapy a pozice robotu.
- Dempster expectation maximization algorithm, řeší, zda data nasnímaná senzory na různých pozicích odpovídají stejnému fyzikálnímu objektu v reálném světě.
- hledání objektů, které mohou odpovídat Například zdem, dveřím, nábytku, nebo pohybujícím se objektům.

4.2.2 Základní úlohy při mapování

Základem mapování by mělo být to, že robot zná svoji polohu. Což může zjistit za pomoci metod popsanych z lokalizace. Samozřejmě při průzkumu prostředí těžko může znát svoji absolutní pozici v daném prostoru, když vlastně ani neví, jak daný prostor vypadá. Je proto třeba využít všech prostředků pro to, aby si uchoval pojem o tom, kde se nachází alespoň relativně. Bohužel i zde zasahují do hry již zmíněné nepříznivé vlivy – šumy. Ať už přichází chyba z odometrie při pohybu robota nebo jeho dalších přístrojů např. kamery, laserové nebo infračervené dálkoměry, sonary, radary, doteková čidla, kompas a systém GPS. Vše by bylo jednodušší, kdyby se robot nemusel pohybovat, bohužel, za překážky zatím vidět neumíme, takže budeme muset zůstat u pohybu. Na obr. 13 je vidět jak se postupně zvětšuje chyba z odometrie.



Obr. 13 Narůstající chyba při odometrii špatně změřené trajektorii.[17]



Obr. 14 Velký problém při tvorbě mapy kruhového prostředí – robot s chybou v odometrii se snaží zmapovat velkou kruhovou chodbu. Navázání shody začátku a konce mapování bývá jedním z nejnáročnějších problémů při mapování.[17]

Na obr. 14 je vidět, jak robot díky šumu z odometrie „narovná“ kruhovou chodbu a pak se mu nepodaří ji správně napojit. Protože nemá správné „povědomí“ o své pozici.[17] Z obr. 14 také vyplývá jeden z největších úkolů pro robota – napojit správně začátek a konec své cesty a tím pádem i mapy.

Samozřejmě musíme brát v potaz, že se většina prostředí s časem mění. Téměř největším problémem mohou být dveře – původně otevřené a při druhém průjezdu zavřené takže robot bude stát před otázkou, jestli jsou dveře jen zavřené, nebo je někde úplně jinde.

Jak bylo výše zmíněno základem mapování je tudíž zcela a naprosto závislé na lokalizaci, tudíž je ji třeba robotu implementovat v kvalitní podobě. Zároveň je v dnešní době mapování považováno za největší a nejtěžší úkol v robotice.

4.3 Metody mapování

Pravděpodobně všechny metody mapování uvedené v dostupné literatuře mají jedno společné – jsou založeny na pravděpodobnosti. Všechny využívají pravděpodobnostního modelu robota a jeho prostředí a všechny závisí na dedukci pozice pro přenesení dat ze senzorů do mapy.

Pravděpodobnostní algoritmy totiž modelují různé zdroje šumů a jejich vliv na měření. Základem každého funkčního algoritmu je Bayesovo pravidlo:

$$p(x | d) = \eta p(d | x) p(x) \quad [17]$$

V tab. 2 můžeme vidět porovnání nejdůležitějších používaných metod, které přímo vychází z tohoto pravidla a také jejich silné a slabé stránky.

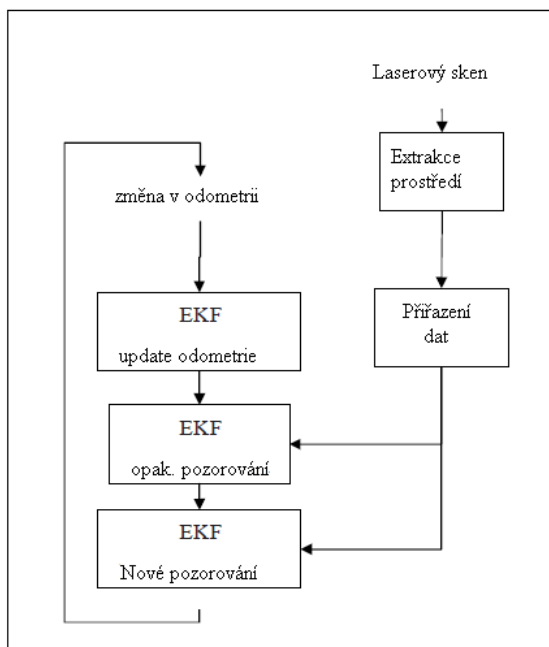
	Kalman	Lu/Milios	EM	Inkrement	Hybrid	Multiplanár. mapy	Dogma
Reprezentace	Pozice mezníků	Bodové překážky	Bodové překážky	Pozice mezníků/ Mřížkové mapy	Bodové překážky	Objekty a polygony	Mřížky obsazenosti
Nejistota	Pozdější pozy a mapa	Pozdější pozy a mapa	Maximálně pravděpodobná mapa	Maximálně pravděpodobná mapa	Maximálně pravděpodobná mapa	Maximálně pravděpodobná mapa	Pozdější mapa
Konvergence	silná	ne	slabá	ne	ne	slabá	slabá
Lokální minimum	ne	ano	ano	ano	ano	ano	ano
Inkrementální	ano	ne	ne	ano	ano	ne	ne
Šum senzorů	Gausovský	Gausovský	jakýkoli	jakýkoli	jakýkoli	Gausovský	jakýkoli
Schopnost mapovat cykly	ano	ne	ano	ne	Ne vnořené	n/a	n/a
Dimenzionalita mapy	$\sim 10^3$	Bez limitu	Bez limitu	Bez limitu	Bez limitu	Bez limitu	Bez limitu
Shoda	ne	ano	ano	ano	ano	ano	ano
Zvládá hrubé data	ne	ano	ano	ano	ano	ano	ano

Tab. 2 Porovnání hlavních mapovacích algoritmů.

Dále bude proveden výběr z některých již používaných metod mapování:

4.3.1 SLAM (Simultaneous Localization and Mapping)

Současná lokalizace a mapování je v robotice považováno nikoliv za metodu ale za problém, který je třeba řešit. Jednoduché schéma tohoto problému je vidět na obr. 15.



Obr. 15 SLAM.

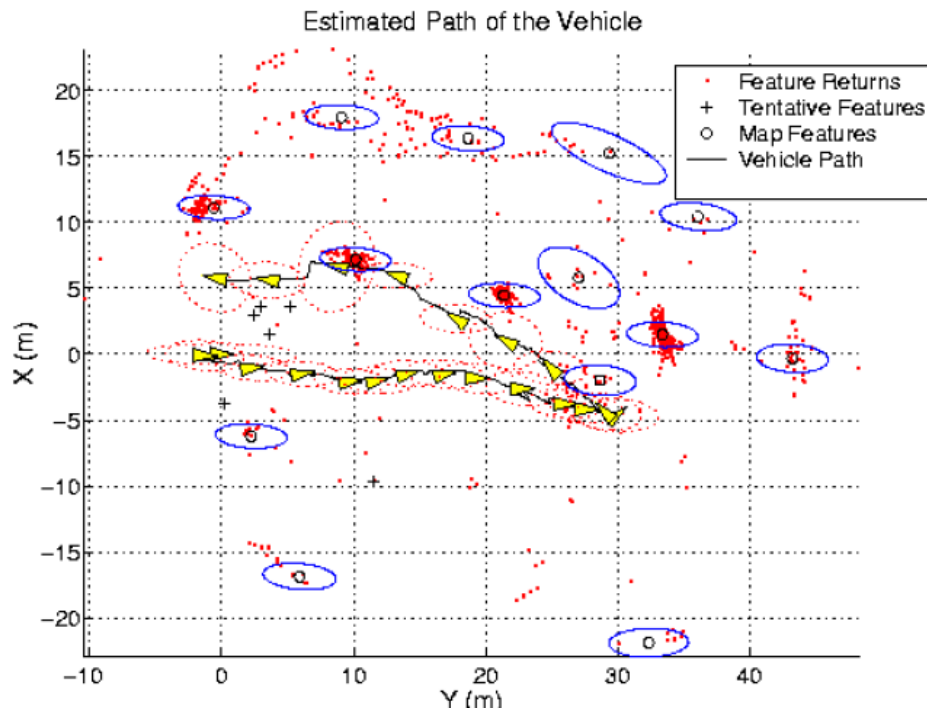
Řešením tohoto problému obsahuje nalezení odpovídající reprezentace pro pozorující (mapující) a pohybový (lokalizující se) model. Toto řešení vede k využití tzv. Extended Kalman Filter (Rozšířeného Kalmanova filtru).[2]

4.3.2 Kalmanovy filtry

Mapování pomocí Kalmanových filtrů závisí na třech základních předpokladech:

- příští postavení modelu musí být lineární s přidaným Gaussovským šumem
- stejná charakteristika musí platit i pro model pozorovací
- počáteční nejistota musí mít Gaussovo rozložení

Příkladem Kalmanova filtru může být podvodní vozidlo na Obr. 16(ze zdroje[17]). Toto vozidlo využívá pro detekci překážek sonaru. Na obrázku vidíme mapu sestávající ze 14 rysů. Elipsy okolo nich udávají jejich nejistotu. Toto měření udává pouze relativní pozici robotu vůči překážkám.



Obr. 16 Kalmanův filtr[17]

4.3.3 Algoritmus očekávané maximalizace (Expectation Maximization – EM)

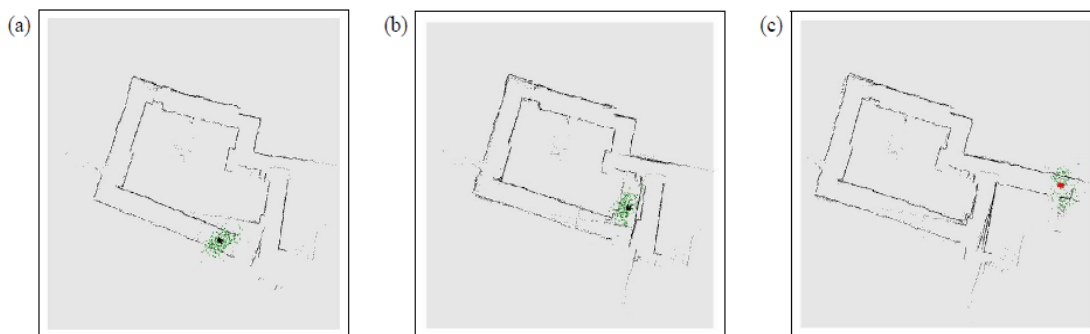
Je to relativně moderní alternativa Kalmanova filtru. Zásadní výhodou proti KF, je že jednoznačně řeší problém shody (Correspondence problém) zmiňovaný na začátku kapitoly. Je to díky tomu, že po dojetí jsou vypočítány hypotézy, kde vlastně robot mohl být a tudíž nemůže vzniknout problém shody.[17]

4.3.4 Hybridní přístup

Literatura okolo robotiky zná příklady hybridního řešení na základě EM a KF. Anglicky nazývaná *incremental maximum likelihood method*. Základní myšlenkou je inkrementálně budovat jednu mapu jakmile přijdou data ze senzorů, ale bez ukládání jakékoli nejistoty. Podstata tkví v tom, že si ukládá vzorky svých pozic a po dokončení kola tyto vzorky použije pro své znovu-zlokalizování. Bohužel tento algoritmus má mnohé nevýhody, z nichž ta první může být opravdu zásadní:

- rozhodnutí o změně mapy zpět v čase je přesně to, co, pokud je špatné, může vést k absolutnímu selhání
- není to real-time algoritmus, vzhledem k tomu, že k opravě kola dochází právě po jednom kole, a tudíž závisí na jeho délce

Nicméně, v praxi to vypadá, že tato metoda může fungovat i v real-time, pokud je užívána uvnitř budov kancelářského typu.[17]



Obr. 17 Hybridní přístup – udržuje poslední odhady robotových pozic, při uzavírání kola použije získané vzorky pro relokaci robota a opravení mapy. Získáno z [17]

4.3.5 Mřížky obsazenosti

Mřížky obsazenosti jsou základní reprezentací prostředí mobilního robota. Jejich základní nevýhodou je jejich paměťová náročnost, což vede i k časové náročnosti jejich zpracování.

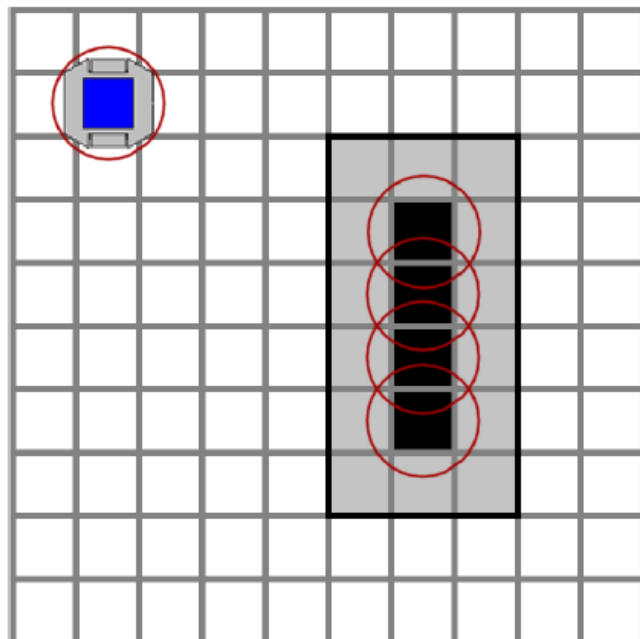
Využití mřížky obsazenosti dle autora S. Thuna[17] se odlišuje od standardních přístupů založených na mřížkách obsazenosti, které rozkládají dvojrozměrné pole buněk na jednorozměrné s tím, že obsazenost každé buňky je odhadována jednotlivě, z čehož vyplývá nekonzistentnost map. Tento algoritmus řeší problém mapování nad celou mřížkou obsazenosti se zachováním všech závislostí mezi sousedními buňkami. Autor ho označuje názvem *Occupancy Grid Maps With Forward Sensor Model*. Tímto algoritmem je dosahováno větší konzistence mapy. Nevýhoda této metody je větší citlivost na změny v okolním prostředí, což zamezuje úspěšnému použití v dynamických prostředích. Další nevýhoda je častější dotazování na informace jednotlivých buněk, což vylučuje použití tohoto algoritmu v reálném čase. I přes nevýhody tento algoritmus generuje o mnoho přesnější a konzistentnější mapy než standardní přístupy k mapování založené na mřížce obsazenosti. Výsledek této metody je patrný z obr. 18.

Nafukování překážek

Pokud má být vytvářená mapa použita pro navigaci robota, je potřeba ji nejprve upravit

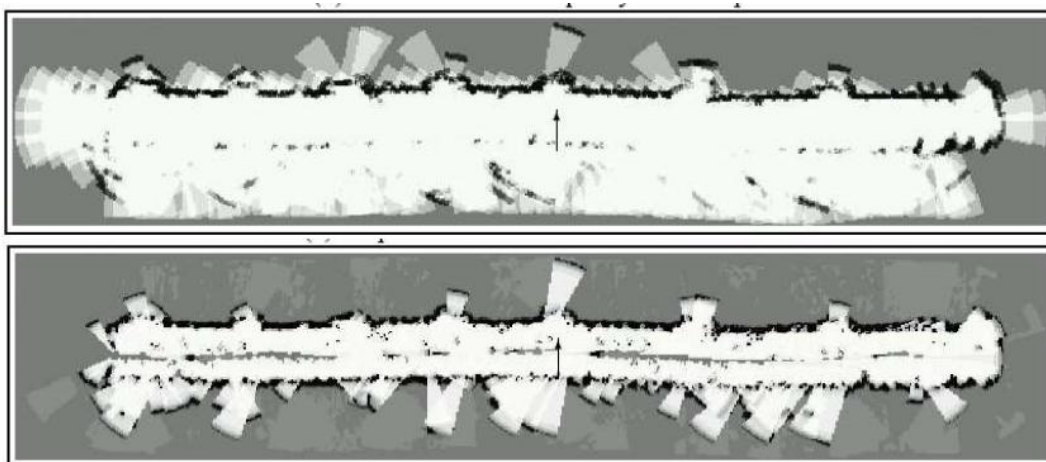
tak, aby navigační algoritmy respektovaly skutečné rozměry robota. Dále popsané algoritmy počítají s nulovými rozměry robota, u kterého je poloha robota v prostředí určena pouze jediným bodem. Vzhledem k tomu, že robot má nenulové rozměry, je třeba mapu upravit tak, aby se i při tomto zjednodušení nemohlo stát, že by robot narazil do překážky. Toho je dosaženo použitím techniky „nafukování překážek“, kdy jsou okraje překážek rozšířeny v každém směru o požadovanou vzdálenost. V tomto případě je kolem každé buňky představující překážku vykreslen kruh. Každá buňka, do které kruh zasahuje je označena jako překážka. Tento postup je zobrazen na obrázku 18.

Po uvedené úpravě je docíleno toho, že se robot nikdy nedostane k překážce tak blízko, aby do ní narazil. Nejbližší body k překážce, na kterých se může vyskytovat střed robota leží na černé hranici kolem nafouknuté překážky.



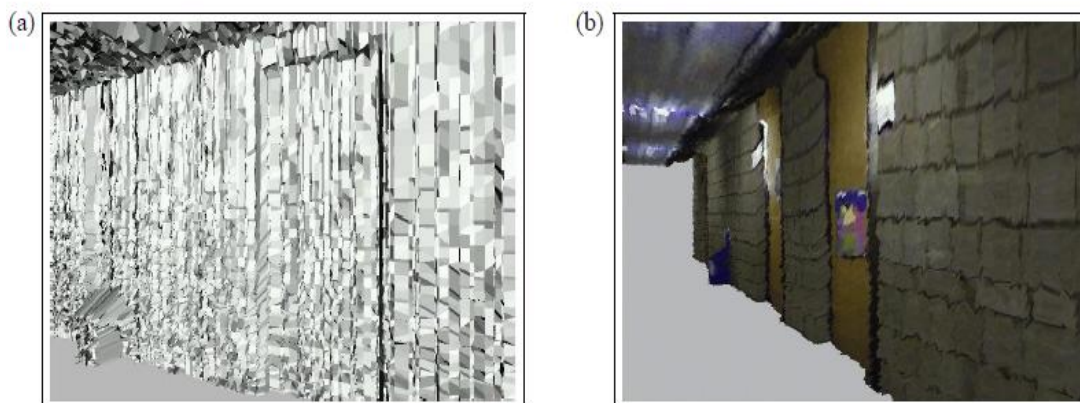
Obr. 18: Princip nafukování překážek.

Využití mřížky obsazenosti dle autora S. Thuna[17] se odlišuje od standardních přístupů založených na mřížkách obsazenosti, které rozkládají dvojrozměrné pole buněk na jednorozměrné s tím, že obsazenost každé buňky je odhadována jednotlivě, z čehož vyplývá nekonzistentnost map. Tento algoritmus řeší problém mapování nad celou mřížkou obsazenosti se zachováním všech závislostí mezi sousedícími buňkami. Autor ho označuje názvem *Occupancy Grid Maps With Forward Sensor Model*. Tímto algoritmem je dosahováno větší konzistence mapy. Nevýhoda této metody je větší citlivost na změny v okolním prostředí, což zamezuje úspěšnému použití v dynamických prostředích. Další nevýhoda je častější dotazování na informace jednotlivých buněk, což vylučuje použití tohoto algoritmu v reálném čase. I přes nevýhody tento algoritmus generuje o mnoho přesnější a konzistentnější mapy než standardní přístupy k mapování založené na mřížce obsazenosti. Výsledek této metody je patrný z obr. 19.



Obr. 19 Nahoře mapa chodby standardním mapováním založeným na mřížce obsazenosti, dole speciální algoritmu S. Thuna. Je vidět, že výsledek je výrazně přesnější.[17]

4.3.6 Objektové mapy



Obr. 20 Objektové mapy.[17]

V publikaci [17] popisují algoritmus pro tvorbu 3D map za pomoci laserových senzorů a kamery. Výsledná mapa je složena z polygonů, které mají různé zabarvení. Tvorba mapy probíhá v reálném čase a její vizualizace je plně interaktivní.

Trojrozměrná interpretace prostředí má několik neoddiskutovatelných výhod:

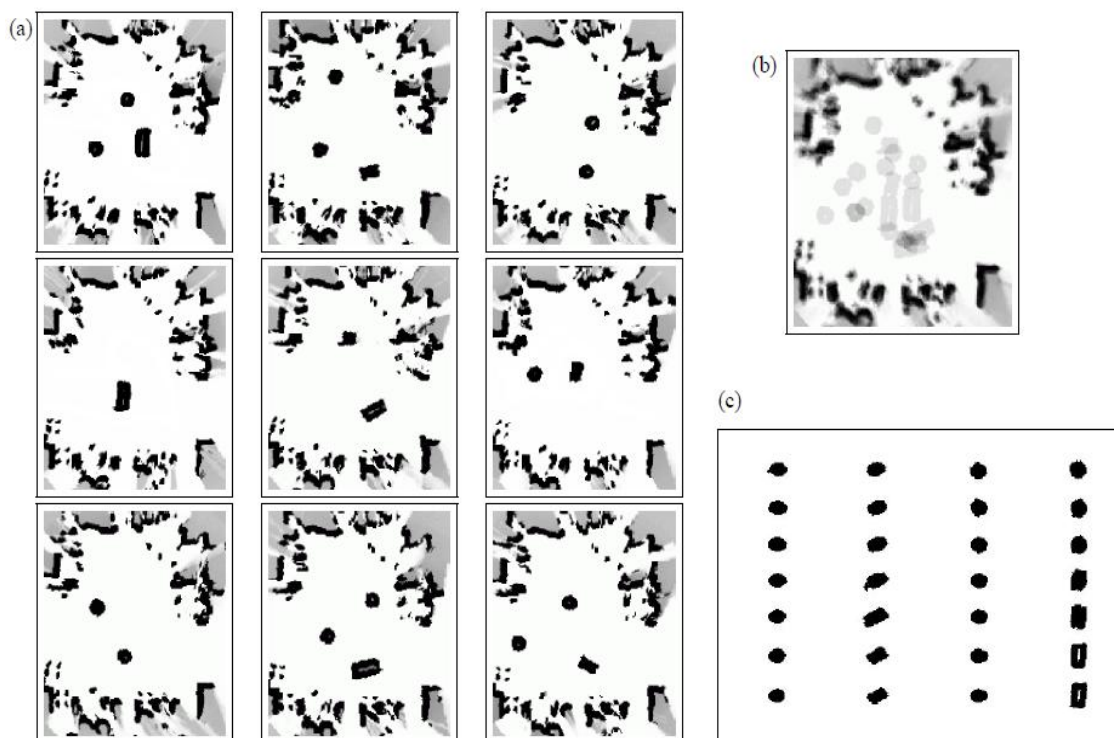
- Popis prostředí je přesnější a přináší možnost rozlišit i podobné místnosti, které by se pomocí 2D reprezentace jevily jako stejné.
- 3D mapa je vhodnější pro potřeby lidí jako jsou například záchranáři, kteří potřebují komplexní popis vnitřních prostorů.

Rovné plochy v okolním prostředí jsou mapovány pomocí obdélníkových ploch. V případě, že se v okolním prostředí vyskytuje nerovinný objekt, je tento objekt reprezentován pomocí malých polygonů, které věrně napodobí jeho reálný tvar. Výsledná mapa je zobrazována ve formátu VRML (virtual reality markup language).

Výsledky této metody mapování jsou zobrazeny na obrázku 20.

4.3.7 Mapování dynamického prostředí

Jak již bylo řečeno, většina předchozích algoritmů vůbec nedokáže fungovat, pokud je použijeme v dynamickém prostředí, což je ovšem naprostá většina našeho světa, proto byly některé z nich modifikovány pro funkčnost. Modifikace znamená, že spolupracují, ale jen pro určité změny prostředí. Například rozšíření mřížek obsazenosti, které dokáže detekovat často se měnící pozice, jako jsou dveře a ve chvíli, kdy je zjistí, během každodenní navigace, rychle určí jejich momentální stav.



Obr. 21 Dynamické prostředí.[17]

Na obrázku 21, pod bodem a) je vidět 9 mřížek obsazenosti a v nich měnící se prostředí místnosti. b) znázorňuje mapu vygenerovanou statickou mřížkou obsazenosti. c) ukazuje 4 objekty identifikované v průběhu mapování za pomoci speciálního algoritmu nazvaného Dogma.[17] V zásadě až s identifikací objektů, které mění svoji pozici, může metoda pomýšlet na úspěšné uplatnění, když robot bude schopen v kanceláři poznat například krabice nebo nábytek, který byl posunut.

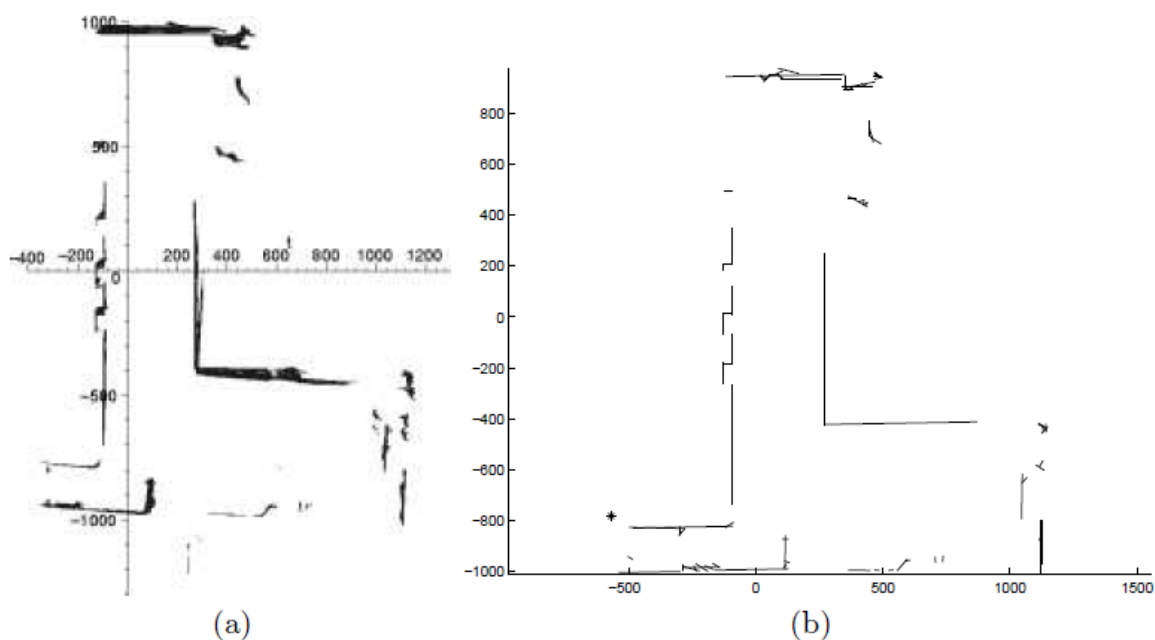
4.3.8 Geometrický přístup

Na rozdíl od metod lokalizace robotu, o kterých bylo prezentováno množství metod založených na různých principech, je situace v oblasti získávání geometrického popisu prostředí poněkud jednodušší. To je dáno zejména tím, že senzory umožňující detekovat překážky s dostatečnou přesností (zejména laserové dálkoměry), se staly dostupnými až v posledním desetiletí. Navíc pro řadu robotických úloh a aplikací není geometrický popis prostředí nutný a proto se pro jejich řešení používají jiné modely. Geometrická reprezentace má však nesporné výhody. Topologické modely totiž neobsahují informace o tvaru, velikosti ani poloze překážek, což eliminuje jejich použití při lokalizaci a při detailním plánování trajektorie. Pravděpodobností mřížky sice informace o obsazenosti prostoru překážkami poskytují, ale jejich nevýhodou je velká paměťová náročnost. Geometrické mapy navíc mají formu srozumitelnou člověku.[19]

Jak již bylo řečeno, obecně se geometrickým popisem rozumí reprezentace prostředí pomocí geometrických primitiv, jako jsou například úsečky, polygony, eliptické oblouky či spliny. Vzhledem k tomu, že práce se složitějšími útvary je výpočetně náročnější a lze je vhodně aproximovat úsečkami, budeme se v dalším textu zabývat pouze modely skládajícími se z úseček, případně polygonů. Dále rovněž předpokládáme, že je v každém okamžiku známa přesná pozice senzoru (ať již některou lokalizační metodou či jinak), kterým se provádí měření a že tedy tato měření nejsou zatížena chybou polohy. Hlavní proud v oblasti mapování založeném na

zpracování dat ze senzorů měřících vzdálenost spočívá v sekvenčním zpracování jednotlivých scanů a jejich postupném přidávání do budované mapy. Základní kostru algoritmu lze popsat v následujících krocích:

1. Inicializuj mapu M podle apriorní znalosti prostředí. Pokud apriorní znalost není, pak M bude prázdná.
2. Získej aktuální sensorický scan.
3. Aproximuj tento scan (respektive jeho vyjádření v kartézské soustavě souřadnic)množinou úseček. Tyto úsečky tak reprezentují části hranic objektu.
4. Na základě definovaného korespondenčního kritéria se nalezne množina odpovídajících si dvojic úseček
5. Korespondenční kritérium je voleno tak, aby jej splňovaly dvojice úseček odpovídající jedné hraně reálného objektu.
6. Úsečky mapy, které by mely být senzorem detekovatelné a k nimž nebyla nalezena žádná odpovídající úsečka aktuálního *scanu*, odstraň z mapy.
7. Každou úsečku z mapy, pro kterou byla nalezena korespondující úsečka aktuálního *scanu*, uprav s ohledem na tuto korespondující úsečku.
8. Kroky 2 až 7 opakuj do té doby, dokud jsou dostupná data ze senzoru.



Obr. 22 (a) Globální mapa vytvořená pomocí laserového scanneru. Mřížka je sestavena z 144400 buněk. (b) Zde je zobrazena odpovídající mapa složená z 50 úseček.(100 koncových bodu). Mapa byla vytvořena pomocí algoritmu Map Merging Algorithm. Obě mapy ukazují výsledek měření stejného senzoru. Převzato z [19].

Problém geometrického mapování komplikuje několik aspektů. Kvůli nepřesnosti senzoru dostaneme různá měření stejné scény a to dokonce i tehdy, pokud se pozice senzoru nemění. Situace je o to složitější, že se robot pohybuje a v různých okamžicích vidí jinou část scény. Jedna konkrétní hrana překážky tak může být měřena z různých pozic, přičemž v každé pozici je detekována jiná její část. To se projeví tak, že pro jednu hranu získáme několik úseček, jejichž délka, pozice a natočení je různá. Na kvalitu výsledné mapy tak má největší vliv nalezení takových úseček, které odpovídají jedné hraně překážky a jejich následné zpracování (sloučení) tak, aby tato hrana byla reprezentována právě jednou výslednou úsečkou. Protože se mapování provádí

sekvenčně *scan* po *scanu*, stačí v každé iteraci nalézt korespondence mezi úsečkami aktuálně zpracovávaného *scanu* a dosud vybudovanou mapou a na základě těchto korespondencí mapu změnit, což se provádí v krocích 4 až 7.

Dvě úsečky jsou prohlášeny za sobě odpovídající, pokud jsou skoro paralelní a pokud jejich vzdálenost není příliš velká, přičemž to, jak moc mají být úsečky paralelní a blízké, je dáno pevně stanovenými parametry. [19]

5. MODEL SLAMU

5.1 Úvod

Model, který má za úkol dokázat funkčnost mnou zvolené metody – SLAMu, je napsán v aplikaci Matlab a vychází ze základu, který na svých stránkách [20] uveřejnil W. Green. Tento základ obsahoval několik částí:

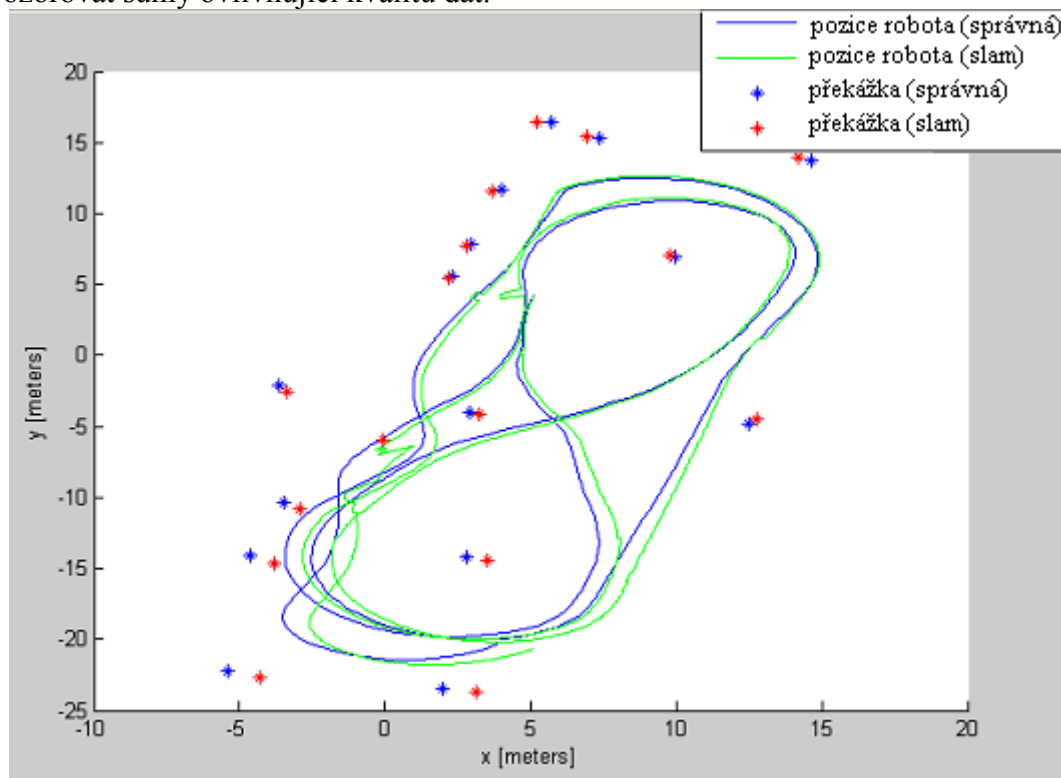
- skutečná data změřená pomocí dálkoměru SICK
- data o robotovi použitém při snímání
- data z gps zpracovaná pro využití při odhadu pozice robota
- aplikaci, která vykreslí pozici překážek (skutečnou i odhadnutou), a také cestu robota (skutečnou i odhadnutou)

Do tohoto základu bylo nutno dodat:

- model robotu
- model dálkoměru
- zpracování dat
 - použití Kalmanova filtru pro odhad pozice
 - výpočet pozice překážek

5.2 Výsledky získané z modelu

Na obrázku 22 je vidět zpracovaná data a jejich vykreslení. Je zde také možno pozorovat šumy ovlivňující kvalitu dat.



Obr. 22 Trasa robota s vykreslením objektů, tak jak jsou ve skutečnosti i jak je „vidí“ robot.

Vzhledem k tomu, že jsem si data pouze zapůjčil z [20], řekl bych, že můj model celkem věrně (průměrná odchylka pozice robotu je 0,5m) a průměrná odchylka polohy překážek je asi 0,3m) počítá pozici robotu, že data z gps jsou relativně přesná. Také poloha bližších objektů, je v tomto modelu relativně dobrá. Vzdálenější překážky jsou nicméně zobrazeny s trochu větší chybou, kterou bych ale připočetl na vrub laserovému dálkoměru.

5.3 Zhodnocení možnosti využití

Vlastní využití metody založené na gps pro lokalizaci a laserového dálkoměru je omezené samotnou metodou gps, pro užití uvnitř budov, či pro robota, jehož pozici potřebujeme počítat s větší přesností, by bylo nutné použít data z odometrie.

Jinak byl model, který je ale podložený reálnými daty funkční a z tohoto hlediska si myslím, že je i použitelný v reálném prostředí. Samotné zpracování odometrie by nedalo příliš velkou práci a proto si myslím, že je tento systém použitelný i v reálném světě tam, kde gps nevyhoví.

6. ZÁVĚR

Dle zadání jsem prostudoval metody lokalizace a mapování. Jejich souhrn je uveden v kapitolách 2 a 4. Jsou zde uvedeny základní principy pohybu robotů jako odometrie, filtrování dat a jejich využití při zpracování. Dále jsem se zabýval druhy map okolí robota, jejich tvorbou a využitím při lokalizaci.

Na závěr jsem si vybral a za pomoci simulačního prostředí Matlab, dat a části zapůjčeného programu zrealizoval model metody SLAM, upravené pro použití s laserovým dálkoměrem SICK. Výsledkem je zobrazení dráhy robota a překážek v okolí a porovnání tohoto se skutečností.

LITERATURA

- [1] BURNHILL, Darren. Ackerman Steering Principle, 2001, 2009
Dostupné z:
http://www.rctek.com/technical/handling/ackerman_steering_principle.html
- [2] DURRANT-WHYTE Hugh, BAILEY Tim. Simultaneous Localization and Mapping: Part I, 2006, 10 stran
Dostupné z:
<http://www.doc.ic.ac.uk/~ajd/Robotics/RoboticsResources/SLAMTutorial1.pdf>
- [3] DURRANT-WHYTE Hugh, BAILEY Tim. Simultaneous Localization and Mapping: Part II, 2006, 10 stran
Dostupné z:
<http://www.doc.ic.ac.uk/~ajd/Robotics/RoboticsResources/SLAMTutorial2.pdf>
- [4] ŠOLC, František; Žalud Luděk. Robotika, vydáno Brno VUTIUM, 2002, 61 stran
- [5] WINKLER, Zbyněk. Odometrie, 2005
Dostupné z WWW:
<http://robotika.cz/guide/odometry/cs>
- [6] Odometry and Dead Reckoning, 2006
Dostupné z:
http://www.rst.e-technik.uni-dortmund.de/cms/Medienpool/Downloads/Lehre/Vorlesungen/Robotics_Theory/Odometry.pdf
- [7] BORENSTEIN J., EVERETT H. R., FENG L. Where am I? Sensors and Methods for Mobile Robot Positioning, vydáno University of Michigan, 1996
Dostupné z:
<http://www-personal.umich.edu/~johannb/shared/pos96rep.pdf>
- [8] Firemní materiály přístroje SICK LMS 291
Dostupné z:
<http://www.sick.cz/cz/produkty/00/02/indoor/cs.html>
- [9] SICK LMS Technical Description. SICK AG Germany, 2003.
Dostupné z:
<https://mysick.com/partnerPortal/eCat.aspx?go=DataSheet&Cat=Row&At=Fa&Cult=English&ProductID=9174&Category=Produktfinder>
- [10] Bc. FRITZ Tomáš. Aplikace s 3D laserovým dálkoměrem SICK. Brno, 2009, 63 stran. Diplomová práce na fakultě strojního inženýrství VUT na Ústavu automatizace a informatiky. Vedoucí práce Ing. Tomáš Marada, Ph.D.
- [11] BJELKA Jan. Návrh a realizace metody mapování okolí pro mobilní robot. Brno, 2007, 68 stran. Diplomová práce na fakultě strojního inženýrství VUT na Ústavu automatizace a informatiky. Vedoucí práce Ing. Stanislav Věchet, Ph.D.

- [12] WINKLER, Zbyněk. Lokalizace, 2003
Dostupné z:
<http://robotika.cz/guide /localization/cs>
- [13] WINKLER, Zbyněk. Měření rychlosti, 2005
Dostupné z:
<http://robotika.cz/guide /filtering/cs>
- [14] METROPOLIS, Nicholas. The beginning of the Monte Carlo method, Los Alamos Science, 1987
Dostupné z:
<http://library.lanl.gov/cgi-bin/getfile?00326866.pdf>
- [15] F. DELLAERT, D. FOX, W. BURGARD, S. THRUN. Monte Carlo Localization for Mobile Robots, 1999, 7 stran
Dostupné z:
http://www.ri.cmu.edu/pub_files/pub1/dellaert_frank_1999_2/dellaert_frank_1999_2.pdf
- [16] D. FOX, W. BURGARD, S. THRUN. Markov Localization for Mobile Robots in Dynamic Environments, 1999,
Dostupné z:
<http://www.cs.cmu.edu/afs/cs/project/jair/pub/volume11/fox99a-html/jair-localize.html>
- [17] THRUN, S.: Robotic Mapping: A Survey. 2002, 31 stran
Dostupné z:
<http://robots.stanford.edu/papers/thrun.mapping-tr.pdf>
- [18] Mgr. KULICH, Miroslav. Lokalizace a tvorba modelu prostředí v inteligentní robotice. Dizertační práce. 2003, 112 stran
Dostupné z:
<http://labe.felk.cvut.cz/~kulich/diser-kulich.pdf>
- [19] LAKAEMPER, Rolf; LONGIN, Jan Latecki; XINYU, Sun; WOLTER, Diedrich. Geometric Robot Mapping. Philadelphia, 2005, 12 stran
Dostupné z:
<http://www.cis.temple.edu/~latecki/Papers/dgciRobot05.pdf>
- [20] GREEN, William. Simultaneous Localization and Mapping (SLAM), 2006
Dostupné z:
<http://prism2.mem.drexel.edu/~billgreen/slam/slam.html>