

Bakalářská práce

Navigační systém mobilního robotu ve venkovním prostředí

Martin Stejskal



Vedoucí práce: Ing. Jan Chudoba

České vysoké učení technické v Praze
Fakulta elektrotechnická, Katedra kybernetiky

Praha 2014

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: Martin Stejskal
Studijní program: Kybernetika a robotika (bakalářský)
Obor: Robotika
Název tématu: Navigační systém mobilního robotu ve venkovním prostředí

Pokyny pro vypracování:

1. Prostudujte současné metody používané pro navigaci a lokalizaci mobilních robotů v obecném venkovním prostředí.
2. Navrhněte navigační systém pro mobilní robot, jehož úkolem bude navigace na zadanou pozici v mapě při respektování omezení prostředí.
3. Definujte nutné senzorní vybavení robotu.
4. Zkonstruujte model robotu a navrženou navigační metodu implementujte.
5. Funkci navržené metody demonstруйте při experimentech v reálném venkovním prostředí.

Seznam odborné literatury:

- [1] S. Thrun, W. Burgard, D. Fox: Probabilistic Robotics, The MIT Press, Cambridge, Massachusetts 02142, 2005.
- [2] T. Krajník, H. Szücsová, M. Svědihrová, J. Chudoba, a V. Vonásek: Jak to vidí roboti 5. ROBOT REVUE, 8/10(1):31-35, 2010.
- [3] T. Krajník, J. Faigl, V. Vonásek, H. Szücsová, O. Fišer, and L. Přeučil: A Visual Navigation System for RoboTour Competition. In First International Conference on Robotics in Education, Bratislava, volume 1, pages 95-100, Bratislava, 2010. Slovak University of Technology in Bratislava.

Vedoucí bakalářské práce: Ing. Jan Chudoba

Platnost zadání: do konce zimního semestru 2014/2015

L.S.

doc. Dr. Ing. Jan Kybic
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 10. 12. 2013

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

Poděkování

Rád bych poděkoval všem, kteří mi při tvorbě bakalářské práce pomáhali a podporovali mne, hlavně Ing. Janu Chudobovi za ochotu, připomínky a cenné rady. V neposlední řadě pak patří velké díky mým rodičům za jejich trpělivost a dlouholetou podporu.

Abstrakt

Bakalářská práce se zabývá navigací mobilních robotů, kteří se pohybují ve venkovním prostředí. Na začátku práce jsou shrnuty senzory běžně používané při řešení problému navigace robotů. Kromě sensorů jsou dále popsány základní techniky navigace - navigace na základě význačných bodů, podle mapy a vizuální navigace. Na základě toho je v další části uveden návrh vlastního navigačního systému, schopného autonomně vést robot do cílového bodu při současném respektování omezení prostředí. Návrh se opírá především o rozpoznávání cesty pomocí kamery, odometrii, GPS a magnetický kompas. Data ze sensorů jsou zpracována podle pravděpodobnostních modelů a slouží jako vstupní parametry algoritmu lokalizace na mřížce. Během návrhu byl kladen důraz na robustnost a využití běžně dostupných prostředků (mapy apod.). Nakonec jsou shrnuty dosažené experimentální výsledky a vlastnosti navrženého systému.

Klíčová slova

mobilní robot; navigace; lokalizace

Abstract

This bachelor's thesis covers the topic of mobile robots' navigation in outdoor environment. The beginning of the thesis contains description of the sensors which are usually used for solving the navigation problem. Beside the sensors the basic techniques of navigation and their principles are described. In the next part the navigation system is designed. The system is designed to be able to lead the robot to the given destination with respect to the environmental constraints. The design is based on road recognition using camera and on the grid localization. The emphasis was laid to robustness and usage of commonly available resources (e.g. maps). At the end of the work the experimental results and features of the system are resumed.

Keywords

mobile robot; navigation; localization

Obsah

1. Úvod	1
2. Senzory	2
2.1. Enkodéry	2
2.2. Akcelerometry	3
2.3. Gyroskopy	4
2.4. Ultrazvukové senzory vzdálenosti	4
2.5. Magnetický kompas	4
2.6. GPS - Global Positioning System	5
3. Metody navigace mobilních robotů	6
3.1. Relativní měření polohy	6
3.1.1. Odometrie	6
3.1.2. Inerciální navigace	7
3.2. Navigace založená na význačných bodech (Landmark Navigation)	7
3.3. Navigace na základě mapy (Map-based Navigation)	7
3.4. Vizualní navigace	8
4. Použitý robot	9
4.1. Hardware	9
4.1.1. Řízení pohonů	9
4.1.2. Senzorová síť	10
Ultrazvukové senzory vzdálenosti	10
Kompas	11
GPS přijímač	11
4.1.3. Řídící počítač	11
4.2. Komunikační rozhraní mezi robotem a počítačem	12
4.2.1. Fyzická vrstva	12
4.2.2. Linková vrstva	12
5. Návrh navigačního systému	13
5.1. Mapa	13
5.2. Rozpoznávání cesty z obrazu	14
5.3. Lokalizace	16
5.3.1. Měření přesnosti GPS	16
5.3.2. Lokalizace na mřížce	17
5.3.3. Pravděpodobnostní modely pohybu a senzorů	18
Pravděpodobnostní model odometrie	19
Pravděpodobnostní model GPS	20
5.4. Plánování cesty	22
5.5. Proces navigace	23
6. Implementace	25
6.1. Popis funkce programu	25
7. Experimentální výsledky	26
7.1. Simulace lokalizace na umělých datech	26

7.2. Testování lokalizace na skutečném robotu	26
7.2.1. Pokus s jednou očekávanou možnou hranou	27
7.2.2. Pokus s více možnostmi na malém prostoru	27
7.2.3. Vliv velké nepřesnosti GPS	29
7.3. Test celého navigačního systému	29
8. Závěr	32
Literatura	33
Přílohy	
A. Obsah přiloženého CD	35
B. Experimentální výsledky lokalizace na mřížce	36
B.1. Simulace s uměle vytvořenými daty	36
B.2. Pokus s jednou očekávanou možností	39
B.3. Pokus s více možnostmi na malém prostoru	42
B.4. Vliv velké nepřesnosti GPS	45

Seznam obrázků

1.	Inkrementální a absolutní optický enkodér [12]	3
2.	Struktura akcelerometru a ukázka MEMS [12]	3
3.	Použitý robot: 1 - řídicí počítač, 2 - kamera, 3 - ultrazvukové dálkoměry, 4 - enkodéry, 5 - kompas, 6 - GPS přijímač	9
4.	Zapojení senzorové sítě	10
5.	Ultrazvukový dálkoměr SRF02 a jeho citlivost udávaná výrobcem (převzato z [9])	11
6.	Kompas CMPS09 a jeho zapojení v režimu I ² C (převzato z [8])	11
7.	Zapojení rozhraní mezi počítačem a robotem	12
8.	Ukázka vykreslené mapy Vyšehradu	15
9.	Odchylka skutečné a předpokládané polohy vlivem nepřesné odometrie a kompasu	19
10.	Normální rozdělení $N(0, 1)$ odchylky odometrie	20
11.	Ukázky histogramů a odhadnutých logaritmicko-normálních rozdělení v různých prostředích	21
12.	Logaritmicko-normální rozdělení chyby GPS LN(1, 0, 506)	22
13.	Vnitřní struktura navigačního systému a komunikace s hardwarem	25
14.	Rozložení generovaných dat v mapě	27
15.	Volba místa pro pokus s jednou očekávanou možnou hranou	28
16.	Volba místa pro pokus s více možnostmi na malém prostoru	28
17.	Pokus s chybným výsledkem lokalizace	29
18.	Mapa Vyšehradských sadů, vyznačena delší trasa	30
19.	Mapa Vyšehradských sadů, vyznačena kratší trasa	30
20.	Kompletní přehled vývoje pravděpodobnosti simulace lokalizace z kapitoly 7.1	38
21.	Rozložení pravděpodobností při lokalizace na mřížce, příklad 4	41
22.	Rozložení pravděpodobností při lokalizace na mřížce, příklad 3	44
23.	Rozložení pravděpodobností při lokalizace na mřížce, příklad 2 - chybné určení polohy	47

Seznam tabulek

1.	Formát zpráv na sběrnici sériové linky	12
2.	Výsledky měření přesnosti GPS	17
3.	Měření odchylky odometrie od skutečně ujeté vzdálenosti	19
4.	Struktura přiloženého CD	35

1. Úvod

Navigaci můžeme popsat jako sledování a řízení pohybu s cílem dostat se z počátečního místa do cílového s ohledem na omezení okolního prostředí. Zahrnuje interakci s prostředím a plánování cesty. Aby se robot mohl pohybovat do určeného místa, musí znát svou aktuální polohu. V případě, že bychom počáteční polohu robotu zadali, mohl by se robot ihned navigovat do cíle. Pokud by ale musel svou počáteční polohu zjistit sám, předcházela by samotné navigaci ještě lokalizace. Lokalizace robotu není jednoduchým problémem. Většinou totiž nemáme k dispozici buď souřadný systém, ve kterém bychom určili polohu (např. v případě, že by robot používal topologickou mapu, viz. kapitola 3.3), nebo nejsme schopni dostatečně přesně zjistit polohu kvůli chybě senzorů (např. GPS). Pokud nemůžeme zajistit znalost přesné polohy, je navigace složitější, protože musí s chybou lokalizace počítat.

Navigační systém by měl také zahrnovat interakci s okolím (např. detekce a vyhýbání se překážkám, rozpoznání cesty) a plánování cesty. To znamená, že navigace musí respektovat omezení prostředí. Z výše uvedených důvodů se jedná o velmi komplexní problém. Proto obecně není snadné implementovat jeho řešení.

V případě robotů, kteří se pohybují ve vnitřním omezeném prostředí (budova, kancelář, atd.), je úloha lokalizace a navigace o něco jednodušší v tom smyslu, že máme k dispozici menší prostor, ve kterém můžeme například najít význačné body a při lokalizaci se podle nich řídit (landmark navigation). Zároveň můžeme předpokládat, že se vyznačené body nebudou s časem příliš měnit (posouvat, ubývat či přibývat další). Jinou možností je vybraný omezený prostor, ve kterém se má robot pohybovat, vybavit aktivními majáky a lokalizaci provádět triangulací nebo trilaterací. Vnitřní prostředí tedy nabízí více možností, jak určit polohu robotu a dále jej navigovat.

Na rozdíl od omezeného prostoru a vnitřního prostředí je navigace robotů ve venkovním prostředí složitější. Mohli bychom samozřejmě využít například metodu význačných bodů. Zmapovat však větší území, tak aby bylo možné robot jednoznačně lokalizovat, by si vyžádalo velkou databázi a především velké množství času potřebného pro zmapování. Mimo to se venkovní prostředí může rychle měnit a databáze by se tak musela často aktualizovat. Tomu se dá předejít použitím mapy cest. V takovém případě ale nemáme k dispozici dostatečně přesnou metodu, abychom jednoznačně určili polohu robotu.

Cílem této práce je proto navrhnout a implementovat vhodný navigační systém, který bude schopen pracovat na většině cest (parkové cesty, městské ulice, silnice) bez potřeby vlastního mapování, navigačních majáků či jiné úpravy prostředí. Z toho důvodu byl při návrhu kladen důraz na využití takových mapových podkladů, které jsou běžně dostupné a nevyžadují si speciální úpravu závislou na platformě robotu.

2. Senzory

Abychom mohli určit polohu robotu, detekovat překážku nebo určit směr jízdy, potřebujeme získat informaci o stavu okolí. K těmto účelům slouží robotu senzory. Vhodné senzory musíme zvolit pro konkrétní robot podle konstrukce a především podle jeho účelu a prostředí, ve kterém se bude pohybovat.

Vzhledem k velkému množství různých senzorů je můžeme rozdělit například podle měřené veličiny. Z hlediska lokalizace robotu, měření jeho polohy a pohybu, se nabízí jiný možný způsob dělení. Podle [10] a [11] se dají senzory rozdělit na základě zpracování výsledků měření a významu informace, jaký data o poloze či pohybu robotu nesou. První skupinou jsou senzory využívané při relativním měření polohy (tzv. dead-reckoning). Tyto senzory udávají změnu polohy nebo pohybu vzhledem k předchozímu stavu. Podrobněji je tento přístup popsán v kapitole 3.1. Druhou skupinou jsou pak senzory používané k absolutnímu měření polohy, jejichž výsledek měření je vztažen k nějaké vnější referenci (např. pro kompas je to severní magnetický pól).

V další části této kapitoly bude následovat stručný přehled senzorů, které se v robotice běžně používají bez ohledu na dříve uvedená dělení.

2.1. Enkodéry

Enkodéry jsou senzory polohy, případně natočení. Při vhodném vyhodnocování dat můžeme pomocí nich měřit i rychlost. Podle měřené veličiny rozdělujeme enkodéry na optické, magnetické, induktivní nebo kapacitní. Vedle toho se dělí enkodéry podle způsobu vyhodnocování na inkrementální a absolutní.

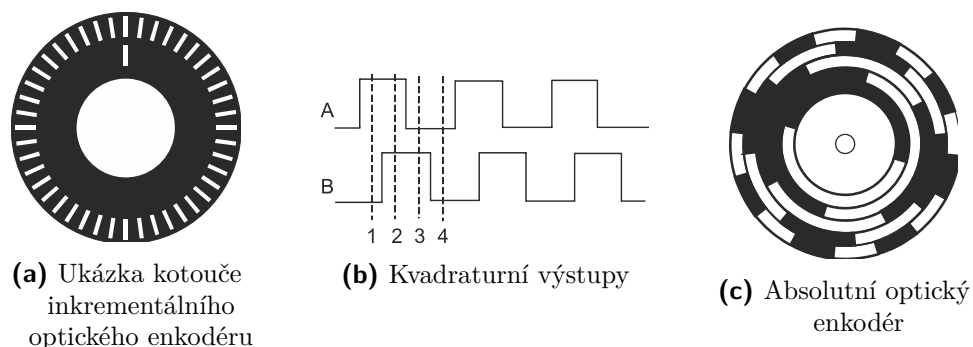
Princip funkce inkrementálního enkodéru bude popsán na optickém enkodéru. Inkrementální optický enkodér se skládá z děrovaného kotouče (viz obrázek 1a), který rotuje mezi zdrojem světla (vysílací LED dioda) a přijímačem (fototranzistor, resp. fotodioda). Vzniká tak optická závora, která generuje pulzy v závislosti na střídání děr a neprůhledných ploch mezi přijímačem a vysílačem. Velikost děr určuje rozlišení enkodéru.

Vedle děrovaného kolečka se používá i kolečko, kde se střídají reflexní a matné plochy, přičemž reflexní plochy mají funkci děr a matné plochy mají funkci neprůhledných částí. Přijímač a vysílač jsou v tomto případě na stejné straně. Na kolečkách bývá navíc pod hlavní řadou děr jedna díra pro synchronizaci jedné otočky. Počet otoček enkodéru spočítáme ze vztahu

$$rot = \frac{n}{N_{PR}}, \quad (1)$$

kde rot je počet otoček, n jsou napočítané pulzy a N_{PR} je počet pulzů na jednu otočku. V případě, že enkodér připevníme například na kolo robotu, můžeme spočítat přímo ujetou vzdálenost v $[m]$, pokud bychom počet pulzů n dělili počtem pulzů na 1 m.

Obecně inkrementální enkodéry se vždy skládají z rotujícího kotouče, jehož obvod je rozdělen dobře detekovatelným prvkem na několik desítek až stovek částí. U optických enkodérů je to díra, u magnetických pak magnet. U kotouče je připevněn senzor, který



Obr. 1. Inkrementální a absolutní optický enkodér [12]

podle otáčení snímá jednotlivé dílky na obvodu kotouče. Podle počtu dílků, které kolem senzoru prošly se na základě známého rozlišení určí o kolik se kolo natočilo.

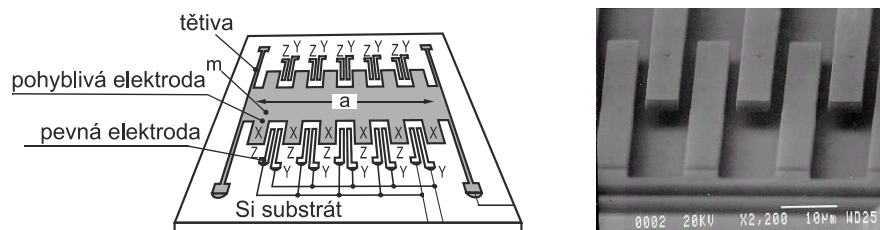
Jednoduchý inkrementální enkodér s jedním snímačem je schopen detekovat změnu a tím i vzdálenost, ale už není schopen podat informaci o směru rotace. Abychom mohli určit směr, jsou zapotřebí kvadraturní výstupy. Princip spočívá ve vhodném umístění dvou přijímačů a vysílačů. Na výstupu enkodéru dostáváme dva obdélníkové signály, jejichž vzájemný fázový posuv odpovídá směru rotace (obrázek 1b). Vzdálenost vyhodnocujeme opět podle napočítaných pulzů.

Absolutní enkodéry na rozdíl od inkrementálních neudávají změnu, ale jsou schopny měřit absolutní natočení rotujícího kotouče. Informace o poloze musí být vhodně zakódována a podle toho také musí být uzpůsobený počet senzorů. Příklad absolutního optického enkodéru je na obrázku 1c. Poloha je zde zakódována kombinací bílých a černých ploch v jednotlivých drahách Grayovým kódem.

V robotice mají enkodéry širokou oblast uplatnění - od měření natočení ramena průmyslového robotu až po odometrii v mobilní robotice. Odometrie je blíže popsána v kapitole 3.1.1.

2.2. Akcelerometry

Akcelerometr je senzor sloužící k měření zrychlení. Nejčastěji se v běžné praxi setkáváme s mikromechanickými kapacitními akcelerometry založenými na struktuře MEMS (Micro Electro-Mechanical System). Základem MEMS akcelerometrů je destička z křemíkového substrátu s pevnými elektrodami. Na ní je pomocí pružných tětív připevněna pohyblivá elektroda, která je současně pohyblivou hmotností a může se pohybovat pouze v jedné ose. Pevné a pohyblivé elektrody tvoří dva do sebe zaklesnuté „hřebeny“ a podle jejich vzájemné kapacity se měří zrychlení ve směru. Z toho vyplývá, že pokud bychom chtěli vyrobit akcelerometr schopný měřit zrychlení ve třech osách (x , y , z), potřebovali bychom tři takovéto struktury. Ukázka jedné struktury elektrod je na obrázku 2.



Obr. 2. Struktura akcelerometru a ukázka MEMS [12]

2.3. Gyroskopy

Gyroskopy nám umožňují měřit úhlové zrychlení. Pracují na základě různých principů, např. mechanický gyroskop s rotující hmotou, gyroskop s vibrujícími prstenci založený na struktuře MEMS nebo optický vláknový, který využívá funkce Sagnacova interferometru. V robotice se gyroskopy používají k inerciální navigaci, která je popsána v kapitole 3.1.2.

Kromě inerciální navigace dokáže gyroskop kompenzovat jeden z nedostatků odometrie. Během pohybu robotu totiž může dojít ke chvilkovým změnám směru pohybu, které odometrie není schopna nijak zaznamenat. Tyto drobné změny mohou ve výsledku ale způsobit velkou odchylku vypočtené polohy od skutečné. Proto je třeba je okamžitě rozpoznat, což nám umožňuje právě gyroskop. Poté co změny zaznamenáme, můžeme na ně vhodně zareagovat, tzn. zohledníme při výpočtech polohy změněný směr pohybu.

2.4. Ultrazvukové senzory vzdálenosti

Měření vzdálenosti je založeno na principu doby letu (Time Of Flight) ultrazvukového signálu. V čase $t_0 = 0$ vyšle snímač ultrazvukový signál a současně začne měřit čas. Měření končí ve chvíli, kdy se zpět k senzoru vrátí odražený signál. Z naměřené doby t se určí vzdálenost d ze vztahu

$$d = \frac{v_s t}{2}, \quad (2)$$

kde v_s je rychlost šíření zvuku ve vzduchu. Ve vztahu se vzdálenost vypočtená z doby letu a rychlosti v_s dělí dvěma, protože se signál šíří k překážce a po odrazu zpět k senzoru.

Ultrazvukové dálkoměry jsou, díky své nízké pořizovací ceně, jednoduchému ovládání a relativně dobré přesnosti (v řádech mm až cm) častou součástí mobilních robotů. Moduly běžně dostupné na trhu mívají implementovanou lokální inteligenci potřebnou pro zpracování dat. Jsou tak schopny komunikovat po některé ze standardních sběrnic (I²C, SPI, UART) a můžeme z nich rovnou číst výslednou vzdálenost překážky bez potřeby dalších výpočtů.

Jejich nevýhodou ve venkovních prostředích je špatná detekce nejednotlivých objektů jako například křoví, vysoká tráva apod. V tomto případě se vyslaný signál rozptýlí na různé strany a zpět k senzoru dorazí pouze slabý signál. Další nevýhodou je horší detekce tenkých objektů (např. nohy lavičky nebo některé sloupky veřejného osvětlení) nebo šikmých ploch, od kterých se signál neodrazí zpět k senzoru.

2.5. Magnetický kompas

Kompas je senzor, který určuje azimut relativně vzhledem k severnímu pólu v intervalu $\langle 0^\circ; 360^\circ \rangle$. Základem magnetických kompasů používaných v elektronice je magnetometr. Magnetometr je zařízení umožňující měřit velikost nebo směr magnetického pole. Podle fyzikálního jevu, jež magnetometr využívá k měření se rozlišuje několik typů magnetometrů - mechanický, fluxgate, magnetorezistivní, magnetoelastický nebo magnetometr využívající Hallův efekt. Vedle magnetických kompasů existují tzv. gyrokompany. Jedná se o nemagnetické kompany. Místo měření magnetického pole je základem těchto kompasů gyroskop.

V praxi bývají kompany používané v robotice tvořeny 3-osým magnetometrem zároveň s 3-osým akcelerometrem. Při měření jsme tak schopni kompenzovat náklon senzoru

z vodorovné polohy, který měření ovlivňuje. Problémem, magnetických kompasů je náchylnost na vnější elektromagnetické rušení. Tím bývá silový vodič v zemi nebo pohon robotu v případě použití elektromagnetického motoru.

2.6. GPS - Global Positioning System

GPS označuje globální družicový polohovací systém, který umožňuje kdekoliv na Zemi určit polohu a přesný čas. Kolem celé Země je rozmístěno 24 družic. Změřením vzdálenosti alespoň od 3 z nich jsme schopni zjistit čas a polohu v souřadnicích zeměpisné šířky a délky. Pokud je k dispozici 4. satelit, určíme i nadmořskou výšku. Vzdálenost od družic se měří metodou „time of flight“ (doba letu) rádiového signálu od přijímače k satelitu. Aby bylo možné změřit dobu přesně, je nejprve nutné synchronizovat čas mezi oběma zařízeními. Vyšší přesnosti lze dosáhnout měřením vzdálenosti od více družic.

Původně se jednalo o systém americké armády. Později byl zpřístupněn i civilnímu obyvatelstvu. Do satelitních signálů byla ale pro civilní přijímače v 90. letech záměrně přidána chyba, aby nebylo možné zneužít navigaci například k navádění raket. Tento jev se označoval jako „selective availability“ (dále jen SA). V roce 2000 bylo SA zrušeno. Podle oficiálních stránek projektu GPS [7] se při celodenním měření a zapnuté SA 95 % měřených poloh pohybovalo v okruhu 45 m od skutečné polohy. Po vypnutí se tento okruh zmenšil na 6,3 m. V roce 2011 byla podle testů americké vlády odchylka horizontální polohy již menší než 3 m [7]. I přesto se přesnost GPS s použitím korekčních metod, jako je diferenciální GPS, může dostat až do řádů desítek centimetrů.

Nicméně přesnost, která v dané aplikaci stačí, závisí na rozměrech objektu, který chceme navigovat. V případě navigace lodi na moři odchylka několika málo metrů nehraje příliš velkou roli, jelikož velikost lodi se pohybuje v řádu desítek až stovek, metrů. Na druhé straně chceme-li navigovat malý robot o rozměrech v řádu do jednoho metru na parkových cestičkách, odchylka několika metrů může znamenat chybnou lokalizaci.

GPS je velmi rozšířená - od navigace letadel a lodí přes navigace pro turisty až po využití v mobilní robotice.

Důležitým parametrem GPS přijímačů je doba startu, tzn. doba po kterou přijímač hledá satelity, aby mohl určit pozici. Rozlišuje se mezi třemi druhy startů:

- „cold start“
- „warm start“
- „hot start“

Cold start nastává ve chvíli, kdy si zařízení nepamatuje poslední souřadnice, čas UTC ani polohu satelitů. Zaměření na satelity a určení polohy je proto dlouhé. Částečně tomu předchází warm start, při kterém je předem známá poslední poloha a čas UTC. Zaměření na satelity je pak o něco snazší a zafixování polohy rychlejší. Nejrychlejší je hot start, při kterém jsou známy všechny zmiňované parametry. Pokud se tedy přijímač nachází na stejném nebo blízkém místě poslední vypočítané polohy, může navázat na předchozí výsledky a tak velmi rychle zafixovat polohu.

3. Metody navigace mobilních robotů

V robotice se můžeme setkat s různými přístupy navigace robotů. Hlavní rozdíly mezi jednotlivými přístupy jsou v používaných senzorech a následném zpracování a kombinaci naměřených dat. Velmi často se jednotlivé přístupy prolínají a v praxi se využívají dohromady. Tato kapitola shrnuje tři základní přístupy (podle [10] a [11]) - navigace na základě význačných bodů, na základě mapy a vizuální navigace. Mimo to je uvedeno relativní měření polohy, které se samostatně používá velmi zřídka, ale bývá naopak častou součástí sofistikovanějších přístupů jako doplněk odhadu pozice.

3.1. Relativní měření polohy

Relativní měření polohy (anglicky „*Dead reckoning*“) je jednoduchý a výpočetně nenáročný způsob navigace. Vychází z předchozí známe pozice a naměřených dat o pohybu robotu (např. ujetá vzdálenost nebo rychlost pohybu a čas). Na základě těchto informací lze vytvořit trajektorii pohybu a určit tak novou předpokládanou polohu. Lokalizace čistě na základě relativního měření je v praxi velmi málo používaná, protože postrádá zpětnou vazbu z prostředí a je zatížena kumulativní chybou. Poskytuje pouze krátkodobou přesnost, která s časem postupně klesá.

3.1.1. Odometrie

Jednou z metod relativní lokalizace je odometrie. Pohyb robotu se určí na základě směru a ujeté vzdálenosti (případně rychlosti a času pohybu). V případě, že by se robot pohyboval přímočarým pohybem ve dvourozměrném prostoru, ujel vzdálenost d ve směru θ (úhel mezi osou x a dráhou robotu), vypočítáme nové souřadnice robotu $[x_n, y_n]$ jako

$$x_n = x_0 + d \cos \theta \quad (3)$$

$$y_n = y_0 + d \sin \theta, \quad (4)$$

kde $[x_0, y_0]$ označují předchozí polohu.

Odometrie patří mezi nejrozšířenější techniky využívané v navigaci robotů. Jelikož je většina robotů založená na platformě s koly, ať už se jedná o Ackermannovo, diferenciální nebo všesměrové řízení, vzdálenost ujetou robotem získáme z úhlového natočení motoru či kol. Úhlové natočení měříme pomocí enkodérů (viz kapitola 2.1). Vhodnější je umístění enkodérů na kola, jelikož tím eliminujeme vůle v převodovce mezi motorem a koly.

Největší chyby odometrie bývají způsobeny špatně určeným směrem pohybu. Na začátku malá chyba azimutu robotu může ve výsledku znamenat velké odchylky odhadované a skutečné polohy. Mimoto trpí odometrie dalšími chybami (především kumulativní chybou). Vždy se proto používá v kombinaci s dalšími metodami navigace.

3.1.2. Inerciální navigace

Vedle odometrie je druhou významnou metodou relativního měření polohy inerciální navigace, která využívá akcelerometry a gyroskopy. K odhadu pozice jsou data z akcelerometrů dvakrát integrována. Výhodou metody je, že nepotřebuje žádné externí reference. Oproti odometrii se předejde například i problémům s prokluzováním kol a dalším. Na druhou stranu každá malá chyba v datech, například drift, způsobí časem vlivem dvojité integrace velkou odchylku od skutečné polohy. Proto jsou inerciální senzory pro navigaci v delším časovém úseku nevhodné.

3.2. Navigace založená na význačných bodech (Landmark Navigation)

Význačný bod (tzv. landmark) je takový objekt nebo jeho část, která se vyznačuje specifickými vlastnostmi, jež je robot schopný svými senzory jednoznačně určit (např. barva, tvar). Mimo to mohou body obsahovat dodatečné informace, jako např. čárový kód. Rozlišujeme dva druhy bodů - přirozené a umělé. Zatímco přirozené jsou takové objekty, které již v prostředí byly a původně mají jiný účel než je navigace robotu, umělé byly do prostředí dodány za účelem navigace. Proto mohou umělé body obsahovat dodatečnou informaci, což usnadňuje jejich rozpoznávání.

Obecně mají význačné body, případně jejich kombinace, pevnou a známou polohu. To znamená, že pokud se podle nich má robot navigovat, musí předem znát, jejich charakteristiky a umístění v prostoru. Proto je nutné na začátku vybrat body tak, aby byly dobře odlišitelné a pro robot snadno rozpoznatelné.

Pokud robot na začátku nezná svou polohu alespoň přibližně, musí prohledat celý prostor uložených význačných bodů, aby byl schopen lokalizace, což znamená vyšší režii. V opačném případě, jestliže robot ví, v jaké části prostoru se pohybuje, může se prohledávání redukovat pouze na určitou část.

3.3. Navigace na základě mapy (Map-based Navigation)

Navigace na základě mapy je technika, při které si robot z vhodně zkombinovaných dat ze senzorů vytváří lokální mapu prostředí a pak ji porovnává s předem uloženou globální mapou. Pokud nalezne shodu, může dopočítat svou aktuální polohu, případně i orientaci, v prostoru. V zásadě se tento způsob navigace skládá ze 3 procesů:

- mapování prostředí, vytvoření mapy
- lokalizace
- plánování cesty

Mapování prostředí je proces, při kterém se z vhodně reprezentovaných sensorových dat vytváří mapa prostředí. Způsob, jak bude mapa v paměti uložena, záleží na konkrétní implementaci. Pro navigaci můžeme samozřejmě použít již existující mapu a vynechat tak krok mapování. V případě že je třeba, aby si robot mapu nejprve vytvořil, může mapovat prostředí autonomně nebo za pomoci člověka. Autonomní mapování je netriviální úlohou. Typicky se na začátku předpokládá nulová znalost prostředí. Důležitá je vhodně zvolená strategie pohybu, při které se robot snaží nasbírat co nejvíce užitečných dat v co nejkratším čase. To závisí na počtu a typu dostupných senzorů. Převládá názor ([11]), že jeden typ senzorů nemůže adekvátně zachytit všechny vlastnosti a vzájemné vztahy objektů v prostředí. Proto se většinou používá více typů senzorů tak, aby byl popis prostředí co nejkomplexnější a objekty byly jednoznačně určeny.

Lokalizace (anglicky také *map-matching*) zahrnuje určení polohy robotu v již známé mapě. S využitím senzorů se vytvoří lokální mapa a ta je poté srovnána s uloženou globální mapou. Výsledek je silně závislý na kvalitě vytvořené globální mapy a na použitých senzorech.

Proces plánování cesty můžeme spustit až v případě, kdy je k dispozici mapa a známe aktuální polohu. Cílem plánování je najít takovou cestu (posloupnost akcí), která dovede robot do cíle s ohledem na daná omezení (čas, vzdálenost, použité prostředky atd.).

Mapa obecně představuje určitou konkrétní interpretaci skutečného prostředí v paměti robotu. Popisuje vzájemné prostorové vztahy jednotlivých objektů prostředí nebo jejich absolutní polohu ve zvolené souřadné soustavě. Podle toho také rozlišujeme dva základní druhy map:

- geometrické
- topologické.

Rozdíl mezi nimi je ve způsobu interpretace objektů v prostředí. Geometrické mapy popisují objekty a jejich polohu v absolutních souřadnicích ve zvolené souřadné soustavě. Z toho plyne důležitá vlastnost geometrických map - lokální mapa vytvořená robotem musí být porovnána s globální mapou. Podle toho se nakonec určí poloha. Příkladem může být mřížka obsazenosti nebo mapa reprezentovaná grafem, přičemž souřadnice jednotlivých uzlů jsou souřadnice dvourozměrného prostoru.

Na rozdíl od toho topologické mapy zachycují relativní vztahy mezi jednotlivými objekty bez absolutního umístění v daném souřadném systému. Proto není potřeba lokalizovat robot vzhledem ke globální mapě, jako tomu je u geometrických map. Tento přístup zároveň umožňuje zmapovat větší prostor bez akumulace chyb odometrie, jelikož jsou veškeré vztahy relativní. Výslednou reprezentací topologických map bývá typicky graf, ve kterém uzly představují samotné objekty, jejich polohy a vlastnosti a hrany reprezentují vztahy mezi nimi.

3.4. Vizualní navigace

Vizualní navigace je založena na optických senzorech, nejčastěji na laserovém scanneru nebo kameře. Umožňuje získat z okolí velké množství informací a určit polohu i orientaci robotu. Zpracování dat bývá ale výpočetně náročné, především pak zpracování obrazu z kamery. Využití kamer při řízení robotu je široké - rozpoznávání cesty, překážek, význačných bodů, stereovidění a další.

Často se tento způsob navigace používá v kombinaci s dalšími metodami popsánymi dříve v této kapitole. Příklad navigačního systému, který kombinuje odometrii, vizualní navigaci a navigaci založenou na význačných bodech, je blíže popsán například v [13]. Z obrazu snímaného kamerou jsou rozpoznány význačné body a cesta, odometrie slouží k odhadu ujeté vzdálenosti.

4. Použitý robot

V této kapitole je stručně popsán robot, který byl použit při návrhu a testování lokalizačních a navigačních algoritmů. Jeho fotografie je na obrázku 3 včetně popisu důležitých součástí a senzorů. Rozměry robotu včetně kol jsou 520 mm x 400 mm ($d \times š$), na výšku měří bez sloupku GPS a kompasu 245 mm, se sloupkem je výška 400 mm.

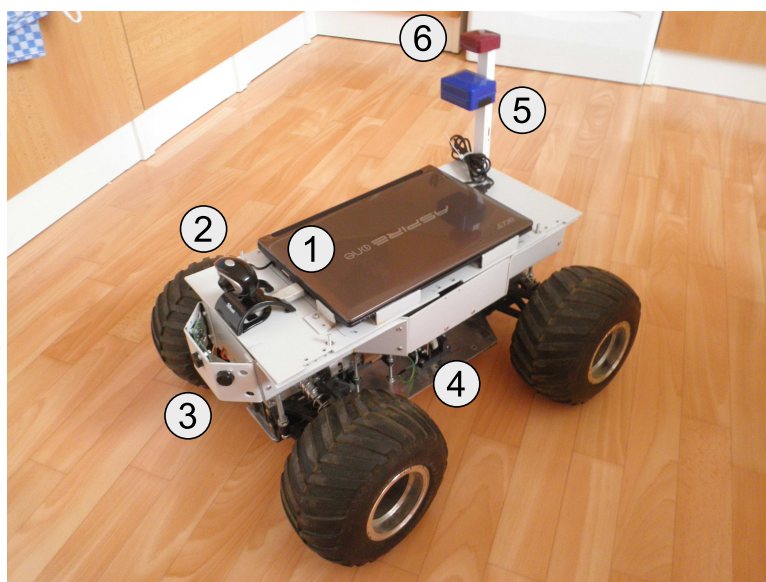
Celá konstrukce je založena na hliníkovém šasi RC modelu terénního auta. Součástí podvozku jsou diferenciály na přední i zadní nápravě. Poháněna jsou všechna čtyři kola. Zároveň má každé kolo nezávislé zavěšení. Tato konfigurace umožňuje robotu zdolávat i obtížnější terén.

Robot je poháněn stejnosměrným komutátorovým motorem s integrovanou převodovkou. Mimo to je výkon motoru ještě převodován na hlavní hnací hřídeli. Maximální rychlost pohybu je proto nižší, avšak o to vyšší je kroutící moment, který se přenáší na kola. Díky tomu je robot schopen zdolávat i větší překážky nebo vézt těžší náklad. Karoserie a elektronika, kromě senzorů, byly navrženy a sestaveny svépomocí. Napájení je zajištěno tříčlánkovým Li-Pol akumulátorem se jmenovitým napětím 11,1 V. Tento typ akumulátoru umožňuje odebírat větší proudy pro napájení motorů.

4.1. Hardware

4.1.1. Řízení pohonů

Řízení hlavního motoru i serva pro zatáčení obstarává mikrokontrolér ATmega128A z rodiny AVR[®] společnosti Atmel[®]. Jedná se o 8 bitový mikrokontrolér s taktovací



Obr. 3. Použitý robot: 1 - řídicí počítač, 2 - kamera, 3 - ultrazvukové dálkoměry, 4 - enkodéry, 5 - kompas, 6 - GPS přijímač

4. Použitý robot

frekvencí až 16MHz. Disponuje čtyřmi čítači/časovači, osmi PWM kanály, A/D převodníkem, komunikačním rozhraním USART, SPI a I²C.

Hlavní motor je řízen opticky odděleným H-můstkem sestaveným z MOSFET tranzistorů. Maximální trvalý proud je omezen na 74 A, pulzně až na 260 A. Výkon motoru reguluje 8-mi bitový čítač s PWM výstupem s frekvencí 10,8 kHz. Natáčení předních kol zajišťuje servo TURNIGY[®] TGY1501 H-TORQUE s kroutícím momentem 17kg/cm. Jedná se o klasické modelářské servo, na jeho vstup je tedy přiveden PWM signál o frekvenci 50 Hz s pulzy dlouhými 1 - 2 ms.

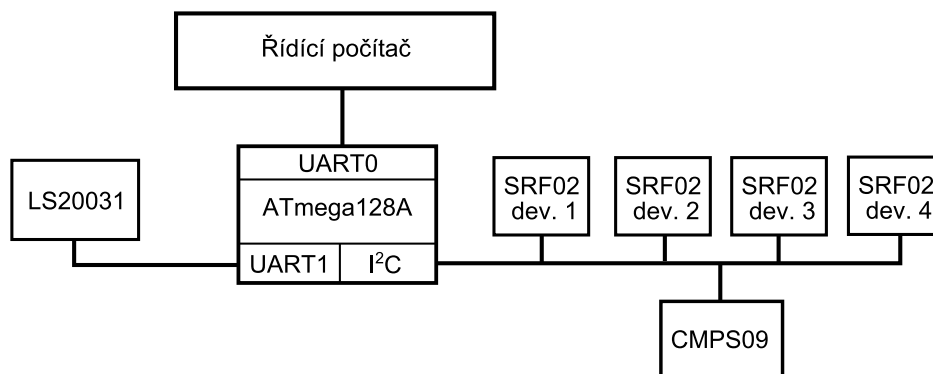
Zpětnou vazbu o pohybu robotu tvoří odometrie. Robot je vybaven jedním optickým inkrementálním enkodérem, který je umístěn na hlavní hnací ose. Kvadrurní výstupy nejsou zapotřebí, jelikož směr pohybu je znám. Po provedení několika měření a následné kalibraci se chyba změřené a skutečně ujeté vzdálenosti dostala v průměru pod 1 % (blíže jsou měření a jeho výsledky popsány v kapitole 5.3.3).

Řízení motorů je navíc doplněno o dvě teplotní čidla - DS1820 a DS18S20. Ty monitorují teplotu akumulátoru a výkonové elektroniky (H-můstku). V obou případech se tak předejde zničením vlivem příliš vysokých nebo naopak nízkých teplot. Užitečné bylo především měření teploty akumulátoru při testování v zimním období.

Připojení k řídicímu počítači a komunikační protokol jsou popsány v kapitole 4.2.

4.1.2. Senzorová síť

Stejně jako u řízení motorů je senzorová síť řízena mikrokontrolérem ATmega128A. Blokové schéma zapojení je na obrázku 4. Popis jednotlivých senzorů je uveden dále.



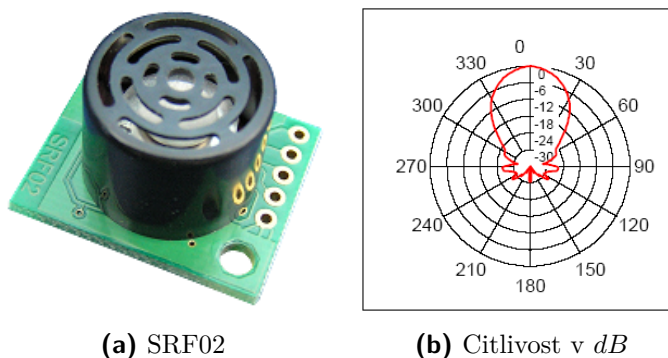
Obr. 4. Zapojení senzorové sítě

Ultrazvukové senzory vzdálenosti

Robot disponuje 4 ultrazvukovými senzory vzdálenosti SRF02 (obrázek 5a). V předu jsou 3, jeden přímo ve směru jízdy a dva pod úhlem 45° vpravo a vlevo, a jeden je umístěn vzadu. Citlivost těchto senzorů je na obrázku 5b. Z grafu citlivosti je vidět, že senzor nemá příliš široký vyzařovací úhel. K detekci překážek bezprostředně před robotem a mírně vlevo a vpravo od přímého směru jízdy jsou proto zapotřebí 3 senzory.

SRF02 je malý ultrazvukový dálkoměr se společným vysílačem a přijímačem. Umožňuje komunikaci po sériové lince v úrovních TTL (UART) nebo po sběrnici I²C. Výsledky měření jsou dostupné v palcích, centimetrech nebo mikrosekundách. Každý modul má svou osmibitovou adresu v rozsahu 224-254 s krokem po dvou (tedy 224, 226, 228, atd.). Lze tak zapojit několik modulů na společnou sběrnici, přitom je jednoznačně adresovat a komunikovat s nimi. V robotu jsou čidla nastavena v režimu I²C.

4. Použitý robot

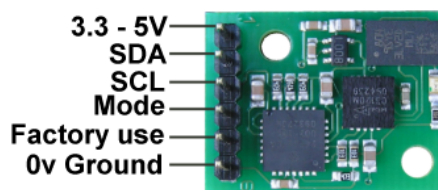


Obr. 5. Ultrazvukový dálkoměr SRF02 a jeho citlivost udávaná výrobcem (převzato z [9])

Kompas

Jako kompas byl použit CMPS09. Jedná se o kompas složený z 3-osého magnetometru a 3-osého akcelerometru. Je proto možné kompenzovat výsledek měření vzhledem k náklonu. Kromě výsledného azimutu jsou dostupné i jednotlivé složky naměřené akcelerometrem a magnetometrem. Obrázek 6 ukazuje kompas a jeho zapojení.

Opět je k dispozici několik komunikačních rozhraní - PWM, sériová linka, I²C. Kompas na robotu je provozován v režimu I²C. Každý modul svou má vlastní adresu. Rozsah adres je 192-206 s krokem po dvou, podobně jako u SRF02. Z toho důvodu a z důvodu úspory vývodů mikrokontroléru je kompas zapojen na stejnou sběrnici jako ultrazvukové dálkoměry.



Obr. 6. Kompas CMPS09 a jeho zapojení v režimu I²C (převzato z [8])

GPS přijímač

Nepostradatelnou součástí při navigaci ve venkovním prostředí je přijímač GPS. Byl zvolen modul LS20031 od společnosti Locosys s integrovanou anténou (bližší specifikace v [16]). Tento GPS přijímač standardně komunikuje po sériové lince v úrovních TTL rychlostí 9600 Bd. Pro potřeby robotu byla rychlost komunikace nastavena na 57600Bd.

Výrobce garantuje obnovení dat minimálně jednou za sekundu, jinak lze souřadnice aktualizovat až desetkrát za sekundu. Cold start trvá méně než 35 s a hot start méně než 2 s (cold start a hot start byly blíže popsány v 2.6).

K určení polohy robotu se používají NMEA věty GGA (Global Positioning System Fixed Data) a RMC (Recommended Minimum Specific GNSS Data), z kterých lze získat všechny potřebné údaje (polohu, čas, počet využívaných satelitů, atd.)

4.1.3. Řídící počítač

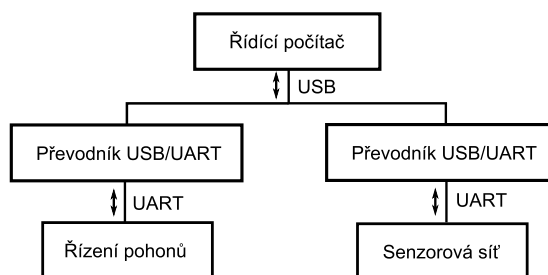
K celkovému řízení robotu (sběr dat ze senzorů, jejich vyhodnocení, zpracování obrazu a další) byl vybrán mini notebook Acer Aspire One D255.

Jde o malý notebook s uhlopříčkou 10,1" a váhou cca 1,25 kg. Má zabudovaný procesor Intel® Atom™s frekvencí 1,66 GHz a operační paměť 1 GB. Svým výkonem tedy plně postačuje potřebám řízení robotu.

4.2. Komunikační rozhraní mezi robotem a počítačem

4.2.1. Fyzická vrstva

Veškerá komunikace s počítačem probíhá přes dva USB porty. Kvůli složitosti protokolu USB je nevhodné aby každý z mikrokontrolérů, který zajišťuje komunikaci s počítačem, tento protokol znal, především kvůli programové paměti. Proto jsou mezi nimi a počítačem převodníky FT232. Tento integrovaný obvod od společnosti FTDI dokáže zajistit převod mezi USB a sériovou linkou UART. UART je mikrokontroléry běžně podporovaným rozhraním a jeho ovládání si proto nevyžaduje velkou softwarovou režii na rozdíl od USB. Z důvodu zrychlení komunikace a zjednodušení softwarového řízení přístupu ke komunikačnímu kanálu z PC mají oba mikrokontroléry svůj vlastní převodník. Blokové schéma zapojení komunikačního rozhraní je na obrázku 7.



Obr. 7. Zapojení rozhraní mezi počítačem a robotem

4.2.2. Linková vrstva

Na sériové lince je přístup k médiu řízen deterministickou metodou master-slave. Roli mastera plní počítač. Na každém ze dvou používaných USB portů je připojen vždy jen jeden slave. Toto zapojení usnadňuje softwarové řešení. Počítač totiž periodicky čte hodnoty senzorů nebo opakovaně zapisuje příkazy řízení motorů v nezávislých vláknech.

Při adresaci se používá adresace uzlů, každá zpráva obsahuje pouze adresu příjemce. Formát zprávy je uveden v tabulce 1. Jako první se vysílá hlavička zprávy, která identifikuje začátek. Tuto funkci plní znak „#“. Za ním následuje adresa příjemce - znak „m“ pro řízení pohonů a znak „s“ pro senzorovou síť. Po adrese následuje příkaz a za ním případně doplňující hodnota. Zpráva je zakončena středníkem. Adresa, příkaz a hodnota jsou od sebe odděleny čárkou. Zpráva pro zapnutí hlavního motoru má tak například tvar „#m,e,1;“. Jednotlivé slave uzly odpovídají na příkazy stejnou zprávou, kde v posledním poli doplní hodnotu podle úspěchu vykonání příkazu. Tzn. pokud master požádá například o zapnutí hlavního motoru jak bylo uvedeno výše, slave odpoví „#m,e,1;“ v případě, že se povedlo motor zapnout. V opačném případě bude mít odpověď tvar „#m,e,0;“

Hlavička	Adresa příjemce	Příkaz	Hodnota	Ukončení
----------	-----------------	--------	---------	----------

Tab. 1. Formát zpráv na sběrnici sériové linky

5. Návrh navigačního systému

Jak již bylo zmíněno v kapitole 3, existuje mnoho metod navigace robotů. Použitý navigační systém je vždy přizpůsoben prostředí, ve kterém se robot pohybuje. Cílem této práce je navrhnout navigační systém schopný řídit robot ve venkovním prostředí. Zároveň však při navigaci musejí být respektována jistá omezení prostředí jako je prostupnost terénem a překážky. Důležitým faktorem při návrhu bylo použití volně dostupných prostředků, které si nevyžadovali žádnou speciální úpravu pro účely navigace. Současně byl kladen důraz na univerzálnost navrženého systému tak, aby byl použitelný i na jiné platformě robotu, než na které byl navrhován. Z těchto důvodů byla zvolena navigace pomocí mapy.

5.1. Mapa

Při výběru mapového podkladu bylo rozhodující, aby byl volně šiřitelný, lehký dostupný, přesný a aktuální. Těmto požadavkům odpovídá projekt OpenStreetMap[©] (dále jen OSM), kde jsou mapy volně ke stažení a jsou šiřitelné za podmínek uvedených v [1]. Jedná se o kvalitně zpracované mapové podklady světa. Standardně jsou mapy dostupné pro online prohlížení ve webovém prohlížeči. Pokud použijeme klienta (při návrhu se osvědčil editor JOSM), lze data stáhnout a používat i offline.

Stažené offline mapy (lze samozřejmě stahovat jen určité úseky) se ukládají ve formátu XML do souboru s příponou *.osm* a obsahují strukturovaný popis mapy. Jednotlivé prvky jsou sdruženy do souvisejících skupin. Nejprve jsou definovány všechny uzly použité v mapě včetně jejich podrobného popisu (tagů), např. názvy ulic, identifikace mostů, řek, pěší cesty atd. Následuje definice silnic, cest, železničních drah, budov a dalších objektů, přičemž jsou využity uzly definované na začátku. Celý soubor tvořený jednoduchou strukturou tak popisuje reálný svět jako mapu jednotlivých objektů a jejich rozmístění.

Tento formát je sice úplný, ale pro potřeby navigace robotu příliš obsáhlý. Při navigaci totiž nepotřebujeme vědět polohu budov ani názvy ulic. Jediné, co nás zajímá, jsou cesty, po kterých může robot jet. Proto je vhodné takto stažené kompletní mapy redukovat do takového formátu, který by nebyl při vyhodnocování příliš rozsáhlý. Přesně to umožňuje formát GPX (GPS Exchange Format; bližší informace o formátu např. v [4]). Opět se jedná o soubor se strukturou XML, tentokrát ale obsahuje pouze definice jednotlivých cest. Na začátku souboru je hlavička, která říká o jaký typ dokumentu se jedná a obsahuje případné doplňující informace o verzi, generátoru GPX souboru a další. Následuje blok metadat a vymezení hranic oblasti, kterou soubor popisuje (tag `<bounds>`). Cesty jsou uvozeny tagy `<trk>` na začátku a `</trk>` na konci. Každá cesta se může skládat z několika segmentů, které jsou uvozeny tagem `<trkseg>` a ukončeny `</trkseg>`. Segmenty označují skupiny bodů, které na sebe logicky navazují. Body cesty jsou uzavřeny v tagu `<trkpt>`. Příklad GPX souboru se dvěma cestami ukazuje kód 5.1.

```
<?xml version='1.0' encoding='UTF-8'?>
<gpx version="1.1" creator="JOSM_GPX_export"
      xmlns="http://www.topografix.com/GPX/1/1"
```

```

xmlns:xsi=" http://www.w3.org/2001/XMLSchema-instance "
xsi:schemaLocation=" http://www.topografix.com/GPX/1/1/">
<metadata>
  <bounds minlat=" 50.0759508 " minlon=" 14.4195134 "
    maxlat=" 50.0775473 " maxlon=" 14.4208539 " />
</metadata>
<trk>
  <trkseg>
    <trkpt lat=" 50.0769054 " lon=" 14.4203578 ">
      <time>2009-10-07T06:55:26Z</time>
    </trkpt>
    <trkpt lat=" 50.0768262 " lon=" 14.4203139 ">
      <time>2012-12-17T21:17:18Z</time>
    </trkpt>
  </trkseg>
</trk>
<trk>
  <trkseg>
    <trkpt lat=" 50.0766071 " lon=" 14.4200448 ">
      <time>2011-02-21T23:22:31Z</time>
    </trkpt>
    <trkpt lat=" 50.0764402 " lon=" 14.4199903 ">
      <time>2011-02-21T23:22:34Z</time>
    </trkpt>
  </trkseg>
</trk>
</gpx>

```

Ukázka kódu 5.1 Ukázka GPX formátu

Jak je z příkladu vidět, GPX formát neobsahuje žádné nadbytečné informace. Popisuje pouze jednotlivé body logicky spojené ve větší celky - cesty. Soubor se tak dobře analyzuje a lehko se z něj získávají potřebné informace. Výsledná mapa vytvořená v paměti po zpracování GPX souboru je reprezentována grafem, kde uzly jsou jednotlivé body a hrany grafu jsou logická spojení mezi body (cesty).

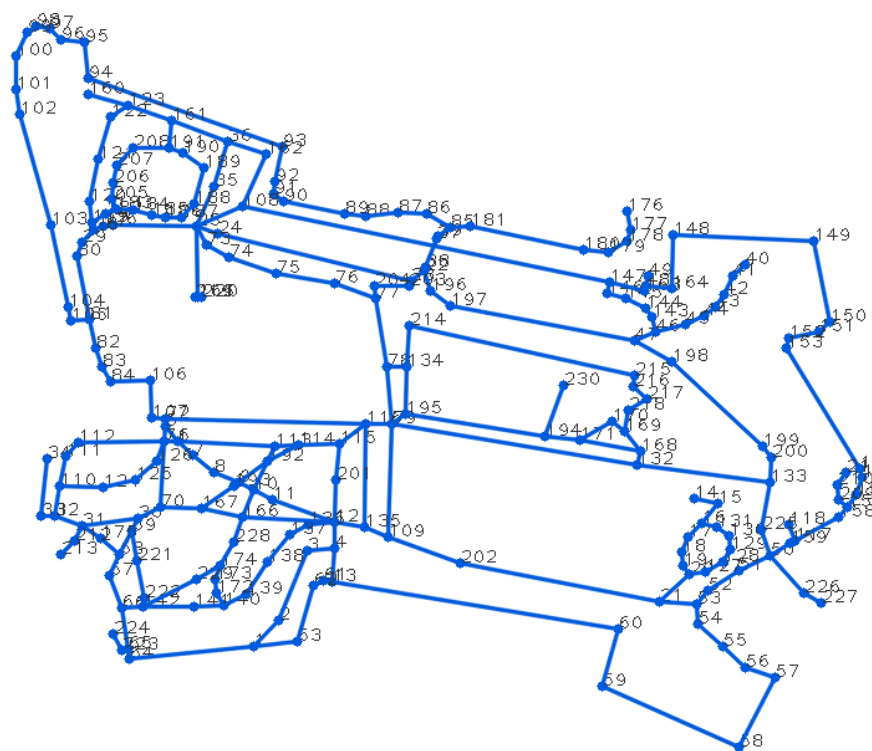
K analýze takto uložené mapy byla použita knihovna libxml2 [2] a její nastavba libxml++ [6] pro jazyk C++. Vykreslování mapy do okna zajišťuje knihovna OpenCV [3]. Ukázka analyzované a vykreslené mapy Vyšehradu je na obrázku 8.

5.2. Rozpoznávání cesty z obrazu

Při jízdě po cestách je nutné, aby robot z cesty nesjížděl a byl schopen určit, zda-li se před ním nachází cesta nebo jestli by dalším pohybem vpřed z cesty sjel. Proto je na robotu umístěna kamera, ze které se přes USB přenáší do počítače aktuální obraz prostředí před robotem.

Algoritmus hledání cesty byl s drobnými úpravami implementován podle [14] a [15]. Algoritmus vyhodnocuje obraz od spodní řádky. Prochází řádku od středu k okrajům obrazu a vyhodnocuje, jestli barva pixelu odpovídá barvě cesty nebo barvě okolí. Řádku prochází do doby než narazí na okraj obrazu nebo na větší počet pixelů s odlišnou barvou než je barva cesty. Podle toho určí v aktuálním řádku šířku a střed cesty a pokračuje

5. Návrh navigačního systému



Obr. 8. Ukázka vykreslené mapy Vyšehradu

na další řádek. Pixely v řádku i jednotlivé řádky prochází s přednastaveným krokem, jelikož procházet každý pixel by bylo výpočetně náročné.

Barvy pixelů se vyhodnocují v barevném systému HSV. Předchází se tak částečně vlivu různého nasvícení stejné barvy, což by ve výsledku mohlo způsobit odlišný výsledek. Vyhodnocení podobnosti barev pixelů se provádí výpočtem barevné vzdálenosti zpracovávaného a naučeného pixelu. Naučeným pixelem se myslí v tomto případě pixel, jehož barva odpovídá barvě cesty. Barevná vzdálenost ΔE^* spočítáme podle vztahu

$$\Delta E^* = \sqrt{(H_e - H_l)^2 + \frac{(S_e - S_l)^2}{4} + \frac{(V_e - V_l)^2}{16}}, \quad (5)$$

kde H_x je odstín, S_x sytost a V_x jas zpracovávaného pixelu. Dolní index e označuje složky zpracovávaného pixelu a dolní index l označuje složky naučeného pixelu. Zároveň ze vztahu vidíme, že složky barvy jsou váhované. Odstín má největší váhu, jas naopak nejmenší. Podle hodnoty ΔE^* a nastaveného prahu se nakonec určí, je-li nový pixel součástí cesty.

Aby hledání cesty nebylo tak výpočetně náročné, je implementováno několik mechanismů snižujících náročnost. Prvním z nich je menší velikost vstupního obrazu. Původní velikost snímků z kamery je 640×480 pixelů, při zpracování je velikost snížena na polovinu bez ztráty kvality výsledku. Druhým mechanismem je zmiňované procházení pixelů s určitým krokem. Osvědčilo se procházení s krokem 4 pro pixely v řádku a s krokem dva pro řádky. Poslední mechanismus je součástí samotného algoritmu vyhodnocování podobnosti pixelů.

Počítat podobnost pixelu obrazu se všemi naučenými pixely by bylo velmi náročné. Proto se na začátku vytvoří pole o velikosti $64 \times 64 \times 64$ reprezentující redukovaný barevný prostor RGB - na každé po sobě jdoucí 4 barvy každé složky připadá jedna buňka pole. V buňkách jsou uloženy hodnoty 0 nebo 1, podle toho, zda daná barva je

nebo není barvou cesty. Přepsáním 0 na 1 se tak systém naučí novou barvu, kterou bude od té doby považovat za barvu cesty. Učení probíhá přes výpočet barevné vzdálenosti popsaný rovnicí 5. S tímto mechanismem je pak počítání barevné podobnosti pixelů realizováno čtením hodnot z pole. To je časově mnohem rychlejší za cenu většího množství spotřebované operační paměti. Navíc se při běhu programu předejde převodu z modelu RGB do HSV, protože evaluace při učení sice probíhá v modelu HSV, výsledek se ale ukládá v modelu RGB.

5.3. Lokalizace

Lokalizace je důležitá a zároveň velmi obtížná část celého procesu navigace. Pokud na počátku procesu dojde k chybnému určení polohy robotu, nemůžeme ho správně navigovat do cíle. Ve venkovním prostředí bez vlastní databáze význačných bodů nebo rozmístěných navigačních majáků je úloha složitější, jelikož zatím není naprosto spolehlivá technika, jak robot lokalizovat. V této práci máme při návrhu navigačního systému k dispozici „pouze“ GPS, kompas a mapu. V případě, že by přesnost GPS byla v řádu nanejvýš několika desítek centimetrů, nebylo by zapotřebí dalších lokalizačních algoritmů. Její přesnost ale silně závisí na podmínkách okolního prostředí (počasí, zastavěná plocha, otevřený nebo zalesněný prostor). Z tohoto důvodu byla před dalším návrhem provedena série měření přesnosti GPS.

5.3.1. Měření přesnosti GPS

Měření přesnosti GPS probíhalo na různých místech vždy třikrát po 100 vzorcích. Prodleva mezi vzorky byla 200 ms a mezi jednotlivými měřeními cca 1 minuta. Místa na kterých měření probíhala byla z hlediska prostředí odlišná - mezi budovami, pod stromy, s částečným zakrytím výhledu stromy, na volném prostranství a na volném prostranství v blízkosti zdroje elektromagnetického rušení (rozvodná stanice). Každé místo zastupovalo prostředí, ve kterém se robot může běžně pohybovat. Kromě souřadnic GPS se současně zaznamenával i azimut.

Souřadnice z každého souboru byly převedeny ze sférické soustavy (za zjednodušujícího předpokladu, že Země budeme považovat za kouli) do kartézské s využitím vztahů (*lat* označuje zeměpisnou šířku a *lon* zeměpisnou délku)

$$x = \cos\left(\frac{\pi}{180} \cdot lat\right) \cdot \cos\left(\frac{\pi}{180} \cdot lon\right) \quad (6)$$

$$y = \cos\left(\frac{\pi}{180} \cdot lat\right) \cdot \sin\left(\frac{\pi}{180} \cdot lon\right) \quad (7)$$

$$z = \sin\left(\frac{\pi}{180} \cdot lat\right). \quad (8)$$

V kartézských souřadnicích je vypočítán střední bod a převeden zpět do sférických souřadnic. Od středního bodu je pak měřena vzdálenost každého bodu v souboru. To nám dá částečný náhled, jak moc jsou data rozptýlena. Vzdálenost d zjistíme z haversin vzorce ([5]):

$$d = 2r \arcsin \left(\sqrt{\sin^2 \left(\frac{\Phi_2 - \Phi_1}{2} \right) + \cos(\Phi_1) \cos(\Phi_2) \sin^2 \left(\frac{\lambda_2 - \lambda_1}{2} \right)} \right). \quad (9)$$

Φ_i označuje zeměpisnou šířku, λ_i zeměpisnou délku a $r \cong 6378,135$ je poloměr Země na rovníku.

Prostředí	Měření	Největší odchylna od středního bodu [m]	Průměrná odchylna od středního bodu [m]	Největší odchylna od střední hodnoty azimutu [°]
Budovy	1	3,74	1,88	0,39
	2	14,52	6,31	0,20
	3	1,98	1,11	0,10
Otevřený prostor	1	0,64	0,48	0,29
	2	0,69	0,45	0,29
	3	4,08	1,01	0,08
Em. rušení	1	0,34	0,14	8,46
	2	0,71	0,21	20,09
	3	4,43	1,27	7,97
Pod stromy	1	1,02	0,45	0,66
	2	0,36	0,17	0,67
	3	0,78	0,43	0,88
Částečné zakrytí stromy	1	0,60	0,49	1,19
	2	8,07	5,85	0,66

Tab. 2. Výsledky měření přesnosti GPS

V tabulce 2 jsou zaznamenány výsledky měření. Z výsledků vyplývá, že až na několik výjimek se průměrná vzdálenost od středního bodu pohybovala nanejvýš do hodnoty cca 1,25 m. Střední bod vypočítaný z naměřených souřadnic se však nikdy neshodoval se skutečným místem měření. Skutečná poloha se od středního bodu lišila vždy do vzdálenosti 10 m. S časem se výsledek měření GPS souřadnic ustaluje a poloha se zpřesňuje. Chyba se ale stále pohybuje v řádu metrů. Vzhledem k velikosti robotu je tato přesnost nedostačující. Proto se při lokalizaci nelze spoléhat pouze na souřadnice GPS. Je potřeba využít i další senzory.

5.3.2. Lokalizace na mřížce

Lokalizační algoritmus v této práci byl navržen podle [17]. Z důvodu jednoduchosti byla vybrána lokalizace na mřížce. Jedná se o jednu ze základních metod. Je založená na Bayesově filtru. Ten v každé iteraci přepočítává rekurzivně pravděpodobnosti pro jednotlivé stavy vektoru \vec{x} ze známých řídicích a naměřených dat. Řídicí data představují informaci o akci provedené robotem. Může se jednat například o výstup odometrie, který také budeme využívat. Řídicí data v čase t budou dále označována jako u_t . Naměřená data jsou informace ze senzorů. Označíme je z_t , což představuje aktuální hodnotu v čase t . V našem případě budeme za z_t považovat naměřené souřadnice GPS.

Algoritmus jedné iterace filtru v pseudokódu vypadá následovně:

```

1: BayesFilter_Iteration( $p(\vec{x}_{t-1}), u_t, z_t$ )
2:   for all  $x_t \in \vec{x}$  do
3:      $\bar{p}(x_t) = \int p(x_t|u_t, x_{t-1})p(x_{t-1})dx$ 
4:      $p(x_t) = \eta p(z_t|x_t)\bar{p}(x_t)$ 
5:   endfor
6:   return  $p(\vec{x}_t)$ 

```

Ukázka kódu 5.2 Iterace Bayesova filtru [17]

5. Návrh navigačního systému

Algoritmus přepočítá pravděpodobnost $p(\vec{x}_t)$ každého stavu z vektoru \vec{x}_t v čase t na základě předchozí pravděpodobnosti $p(\vec{x}_{t-1})$ v čase $t-1$, poslední provedené akce u_t a měřených dat z_t .

Bayesův filtr se skládá ze dvou základních kroků. První krok, řádka 3., vyjadřuje, s jakou pravděpodobností se robot v čase t vyskytuje ve stavu x_t za podmínky, že byla provedena akce u_t a v čase $t-1$ byl robot ve stavu x_{t-1} . Tato podmíněná pravděpodobnost $p(x_t|u_t, x_{t-1})$ je pak násobena pravděpodobností $p(x_{t-1})$, se kterou se robot nacházel v čase $t-1$ ve stavu x_{t-1} . Výpočet probíhá přes všechny stavy x_{t-1} v čase $t-1$, tedy:

$$\bar{p}(x_t) = \int p(x_t|u_t, x_{t-1})p(x_{t-1})dx \quad (10)$$

Druhý krok, řádka 4., započítá do výsledné pravděpodobnosti aktuální měření. Výraz $p(z_t|x_t)$ říká, s jakou pravděpodobností můžeme získat výsledek aktuálního měření z_t za podmínky, že se v čase t robot nachází ve stavu x_t . Výsledná pravděpodobnost stavu x_t je nakonec dána součinem

$$p(x_t) = \eta p(z_t|x_t)\bar{p}(x_t), \quad (11)$$

kde η označuje normalizační konstantu, která zajistí, že se celková pravděpodobnost bude rovnat 1. Jako výsledek je vrácena pravděpodobnost každého ze stavů vektoru \vec{x}_t v čase t .

Uvedený algoritmus představuje pouze jednu iteraci aktualizace pravděpodobnosti. Je proto nutné ho volat opakovaně, aby se pravděpodobnost neustále aktualizovala podle nových parametrů u_t a z_t .

Jak bylo řečeno, lokalizace na mřížce je založena na Bayesově filtru s několika drobnými úpravami. Mřížka odpovídá vektoru \vec{x}_t . Její buňky jsou reprezentovány 2 m úseky cest, které vzniknout rozdělením cest v mapě. Dále filtr diskretizujeme, neurčitý integrál tak přejde na konečnou sumou. Vyjádření pravděpodobnosti $p(x_t|u_t, x_{t-1})$ nahradíme pravděpodobnostním modelem pohybu robotu $p_{move}(x_t, u_t, x_{i,t-1})$ a $p(z_t|x_t)$ nahradíme pravděpodobnostním modelem senzorů $p_{sens}(z_t, x_t, m)$ (m označuje mapu, viz. dále). Oba modely jsou odvozeny dále v kapitole 5.3.3. Poslední změnou oproti původnímu filtru je přidání vstupního argumentu - mapy m . Mapa na vstupu vyjadřuje fakt, že pravděpodobnosti počítáme na základě určitých vztahů mezi u_t , z_t a okolním prostředím robotu. Těmito modifikacemi spojitého Bayesova filtru dostaneme algoritmus lokalizace na mřížce. Výsledný algoritmus v pseudokódu je uveden v ukázce kódu 5.3 (kód popisuje pouze jednu iteraci).

```

1: GridLocalization_Iteration (p( $\vec{x}_{t-1}$ ),  $u_t$ ,  $z_t$ ,  $m$ )
2:   for all  $x_{k,t} \in \vec{x}$  do
3:      $\bar{p}(x_{k,t}) = \sum_i p_{meas}(x_{k,t}, u_t, x_{i,t-1})p(x_{i,t-1})$ 
4:      $p(x_{k,t}) = \eta p_{sens}(z_t, x_{k,t}, m)\bar{p}(x_{k,t})$ 
5:   endfor
6:   return  $p(\vec{x}_t)$ 

```

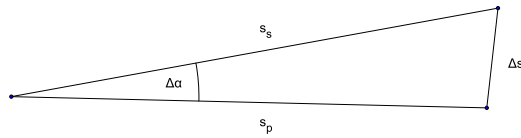
Ukázka kódu 5.3 Iterace algoritmu lokalizace na mřížce [17]

5.3.3. Pravděpodobnostní modely pohybu a senzorů

Při lokalizaci na mřížce jsou potřeba pravděpodobnostní modely pohybu robotu a dat ze senzorů. Ujetou vzdálenost můžeme odhadnout na základě odometrie. Proto na ní

Skutečná vzdálenost [m]	Měřená vzdálenost [m]	Chyba [%]
9,95	9,844	1,06
10,06	10,027	0,33
9,98	9,899	0,81
10,00	9,928	0,72
10,04	9,947	0,92
10,07	10,036	0,34
9,94	9,890	0,50
10,10	10,068	0,31
9,97	9,939	0,31
10,03	9,989	0,41

Tab. 3. Měření odchylky odometrie od skutečně ujeté vzdálenosti



Obr. 9. Odchylka skutečné a předpokládané polohy vlivem nepřesné odometrie a kompasu

také bude pohybový model založen. Model sensorů pak bude vycházet ze souřadnic GPS. V obou případech se pokusíme najít takové rozdělení, které nejlépe postihuje rozložení chyb měření a tedy i míru, s jakou se na získaná data můžeme spolehnout. Modely byly navrženy na základě odvození uvedených v [17].

Pravděpodobnostní model odometrie

Abychom mohli sestavit model odometrie, potřebujeme znát její chybu. Využijeme výsledky pro odchylku azimutu z tabulky 2. Mimo to provedeme měření, při kterém robot necháme opakovaně projet předem danou vzdálenost a na konci každé jízdy srovnáme skutečnou a naměřenou vzdálenost. Pokusy probíhaly na rovné dráze o délce 10 m. Výsledky jsou uvedeny v tabulce 3.

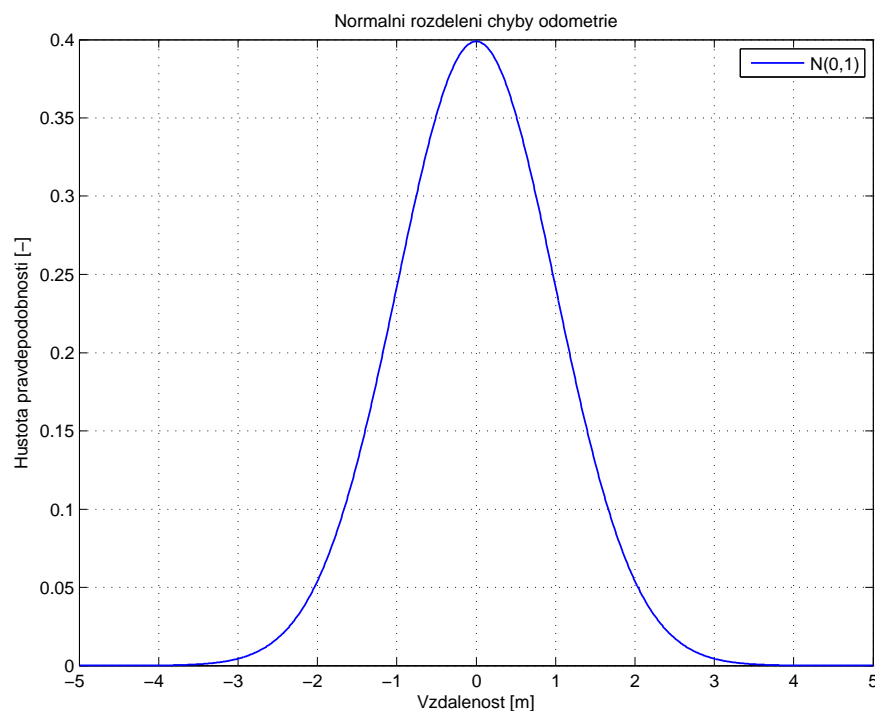
Z tabulek 2 a 3 můžeme vypočítat o kolik se bude lišit předpokládaná poloha, tzn. kde si robot myslí, že je, od skutečné, tzn. kde robot doopravdy je. Výslednou polohu mohou ovlivnit dva faktory - ujetá vzdálenost a azimut. Naměřené hodnoty obou veličin se mohou lišit od skutečných hodnot. Označme s_p jako předpokládanou dráhu, s_s jako skutečnou dráhu, $\Delta\alpha$ jako odchylku azimutu a Δs jako rozdíl konečné předpokládané a skutečné polohy. Situaci znázorňuje obrázek 9. Potom Δs získáme z kosinové věty:

$$\Delta s = \sqrt{s_s^2 + s_p^2 - 2s_s s_p \cos \Delta\alpha} \quad (12)$$

S přihlédnutím k výsledkům v tabulkách 2 a 3 můžeme říct, že odchylka azimutu byla ve většině případů menší než 1° , kromě případů, kdy se v okolí robotu nacházel zdroj elektromagnetického rušení. Podobně pak u odometrie se chyba pohybovala do maximální hodnoty 1 %, což na vzdálenosti 10 m znamená chybu nejvýše 0,1 m a tedy $s_s = 10,1$ m. Pokud tyto hodnoty dosadíme do rovnice 12, dostaneme

$$\Delta s = \sqrt{10,1^2 + 10^2 - 2 \cdot 10,1 \cdot 10 \cdot \cos 1^\circ} [m] = 0,202m, \quad (13)$$

5. Návrh navigačního systému



Obr. 10. Normální rozdění $N(0, 1)$ odchylky odometrie

tzn. při jízdě na vzdálenost 10 m se skutečná poloha robotu bude lišit od předpokládané nanejvýš o 0,202 m. Se zvětšující se dráhou a stejnou procentuální chybou samozřejmě vzdálenost Δs poroste.

Návrh modelu je přizpůsoben algoritmu lokalizace na mřížce. Proto byl zaveden jako pravděpodobnost výskytu robotu v buňce x_{cur} za podmínky, že se robot nacházel v předchozí iteraci v buňce x_{prev} a ujel vzdálenost s ve směru θ . Při výpočtu se z polohy středu buňky x_{prev} vypočte poloha nového bodu p_n ve vzdálenosti s a směru θ . Nakonec se zjistí vzdálenost středu buňky x_{cur} a bodu p_n , podle které se z pravděpodobnostního rozložení vypočítá výsledná pravděpodobnost.

Chybu odometrie můžeme v tomto případě pro zjednodušení zanedbat. Lokalizační algoritmus probíhá přibližně jednou za sekundu a robot za tuto dobu ujede nejvýše 1 m. Odchylka od skutečně ujeté vzdálenosti je tak podle rovnice 12 přibližně 0,02 m.

Jako pravděpodobnostní rozdění modelu bylo zvoleno normální rozdění s hustotou pravděpodobnosti se střední hodnotou $\mu = 0$ a rozptylem $\sigma^2 = 1$ (obrázek 10). Rozptyl odpovídá polovině velikosti buňky, která je nastavena na 2 m. Hustota pravděpodobnosti rozdění $N(0, 1)$ je

$$f_X(u) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(u - \mu)^2}{2\sigma^2}\right). \quad (14)$$

Pravděpodobnostní model GPS

Při návrhu pravděpodobnostního modelu sensorů, tedy v tomto případě modelu GPS, vyjdeme z kapitoly 5.3.1. Rozdění pravděpodobností odhadneme z histogramů vzdálenosti středního bodu od ostatních bodů. Histogramy se pro různá prostředí, kde měření probíhala (viz. kapitola 5.3.1), lišily.

5. Návrh navigačního systému

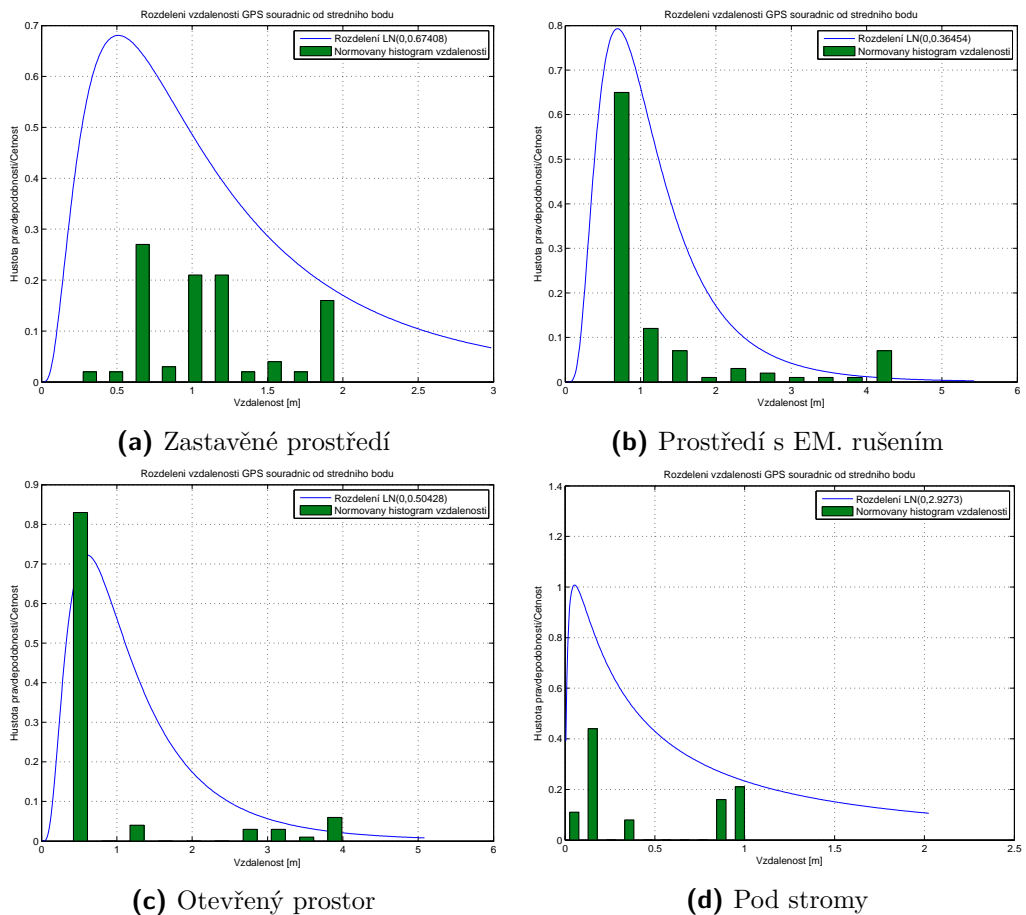
Ze základních spojitých rozdělení odpovídalo histogramům logaritmicke-normální rozdělení $LN(\mu, \sigma)$ se střední hodnotou μ a standardní odchylkou σ . Hustota pravděpodobnosti tohoto rozdělení je

$$f_X(u) = \begin{cases} \frac{1}{u\sigma\sqrt{2\pi}} \exp\left(-\frac{(\ln u - \mu)^2}{2\sigma^2}\right) & \text{pro } u > 0 \\ 0 & \text{jinak} \end{cases} . \quad (15)$$

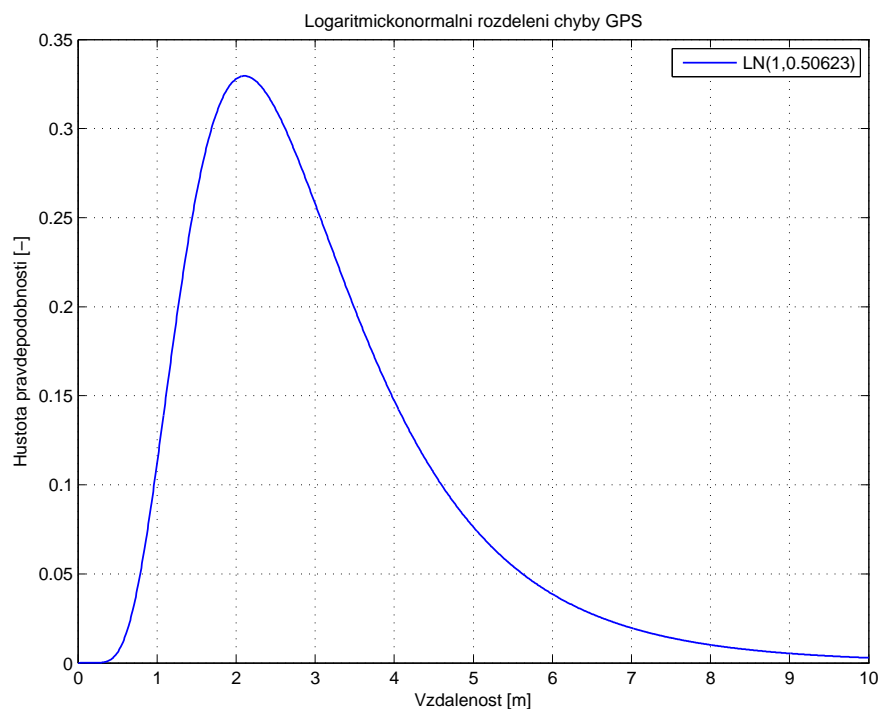
Ukázky, jak rozdělení odpovídalo histogramům, jsou na obrázku 11. Z dílčích rozdělení nakonec můžeme odhadnout parametry výsledného rozdělení tak, aby co nejlépe vystihovalo všechny možné případy. Tedy aby byl pravděpodobnostní model GPS platný ve všech prostředích, v kterých se může robot pohybovat. Jako výchozí hodnoty byly použity hodnoty průměrů střední hodnoty a rozptylu. Po několika experimentech na robotu skutečnosti nejlépe odpovídaly hodnoty

$$\mu = 1, \quad \sigma = 0,506. \quad (16)$$

Hustota pravděpodobnosti výsledného rozdělení je na obrázku 12. Rozdělení odpovídá faktu, že přesné souřadnice změříme pomocí GPS s velmi malou pravděpodobností (založeno na experimentech s konkrétním GPS přijímačem LS20031). Nejčastěji se chyba naměřených souřadnic od skutečné polohy přijímače pohybovala mezi 0,5 m a 4 m.



Obr. 11. Ukázky histogramů a odhadnutých logaritmicke-normálních rozdělení v různých prostředích



Obr. 12. Logaritmicko-normální rozdělení chyby GPS LN(1, 0, 506)

5.4. Plánování cesty

Jak bylo zmíněno v kapitole 5.1, mapa používaná k navigaci je reprezentována grafem, kde jednotlivé uzly jsou body se zeměpisnými souřadnicemi a hrany představují cesty, po kterých může robot projíždět. Plánování cesty od počátečního ke koncovému místu je tedy problémem prohledávání stavového prostoru. Vzhledem k tomu, že máme k dispozici délky jednotlivých cest (ceny hran grafu), použijeme jednu z informovaných metod prohledávání - algoritmus A*. Díky tomu můžeme najít optimální řešení a tím nejkratší cestu do cíle.

Při hledání optimálního řešení používá A* hodnotící funkci

$$f(n) = g(n) + h(n). \quad (17)$$

Kritériem během expandování dalších uzlů je minimální hodnota $f(n)$. Funkce $g(n)$ udává celkovou cenu od počátečního stavu do stavu n a $h(n)$ je heuristická funkce, která je odhadem ceny ze stavu n do cílového stavu. Heuristická funkce musí být přípustná, tzn. musí platit $0 \leq h(n) \leq h^*(n)$, kde $h^*(n)$ je skutečná cena ze stavu n do cíle. Čím více se bude funkce $h(n)$ přibližovat $h^*(n)$, tím rychleji najde A* cestu a tím méně expanduje uzlů. Jako heuristická funkce je použita vzdušná vzdálenost mezi uzlem n a cílovým uzlem. Tím je zajištěna přípustnost, jelikož skutečná vzdálenost do cíle nemůže být nikdy menší než vzdušná vzdálenost mezi uzly (platí $0 \leq h(n) \leq h^*(n)$).

Při plánování cesty může vzniknout situace, při které by se robot musel na začátku cesty otočit do opačného směru. Taková situace nastane ve chvíli, kdy je cesta do cíle kratší přes uzel „za robotem“ (od kterého robot jede). Pokud by byl robot postaven na diferenciálním podvozku, bylo by snadné se otočit na místě. Vzhledem k tomu, že je robot postaven na podvozku s Ackermannovým řízením, bylo nutné otáčení do opačného směru vyřešit.

Aby bylo zaručeno, že robot při otáčení nesjede z cesty, musela by být nainstalována

druhá kamera snímající obraz za robotem. Tento způsob by byl náročný na řízení. Jednodušší je zajistit, aby se plánovací algoritmus vyhnul takové cestě, při které bude nutné robot otočit.

Za počáteční uzel, ze kterého se hledá cesta do cíle, je označen uzel před robotem. Plánování je omezeno tak, že z počátečního uzlu je uzel za robotem nedostupný (vzdálenost do něj je ∞). Z ostatních uzlů je ale dostupný bez omezení za cenu odpovídající skutečné vzdálenosti. Tak je zajištěno, že se robot nebude muset otáčet a zároveň nebude ovlivněno další plánování.

5.5. Proces navigace

Prozatím byly v této kapitole popsána řešení dílčích problémů spojených s navigací. Zde je vhodně sestavíme a vytvoříme navigační systém robotu.

Nejprve je třeba analyzovat mapu, lokalizovat robot a nakonec naplánovat cestu do cíle. V paměti se vytvoří grafová struktura představující mapu. Na základě lokalizace pak dokáže robot určit svou polohu v mapě. Při tom předpokládáme, že na začátku stojí robot přibližně ve směru cesty. Lokalizace probíhá ve dvou krocích. V prvním kroku se zjistí přibližná poloha z GPS souřadnic a nasměrování robotu podle kompasu. Na základě toho se vyloučí všechny cesty takové, které jsou dál od získaných souřadnic, než je nastavený limit. Při zkušebních jízdách se osvědčila hranice 10 m - 30 m v závislosti na prostředí. Zároveň se vyloučí všechny cesty, které vedou jiným směrem. Výstupem prvního kroku je pak seznam několika možných cest, kde se robot může nacházet.

V druhém kroku se tyto cesty rozdělí na jednorozměrnou mřížku s definovanou velikostí buňky (v současné době 2 m; jemnější mřížka dovoluje přesnější výsledky za cenu vyšší rezie). Všem buňkám mřížky je přiřazena stejná pravděpodobnost. Nad takto vytvořenou mřížkou se spustí lokalizace na mřížce z kapitoly 5.3.2. Zároveň s tím se robot rozjede po cestě řízen pouze algoritmem rozpoznání cesty z obrazu. Za jízdy je postupně čtena ujetá vzdálenost a nové souřadnice GPS. Vzdálenost vstupuje do lokalizace na mřížce jako parametr u_t a souřadnice GPS jako z_t .

Důležitou součástí je po každé iteraci algoritmu lokalizace na mřížce pravděpodobnosti normovat. Při lokalizaci se vede současně několik různých hypotéz polohy robotu.

Z různých hypotéz vedených při lokalizaci je potřeba vybrat jednu jako počáteční polohu robotu. Algoritmus takovou hypotézu vybere na základě tří omezení, která musí pravděpodobnost buňky splňovat. První podmínkou je, že pravděpodobnost musí být absolutním maximem na celé mřížce. Druhé a třetí omezení jsou vzájemně propojena - pravděpodobnost musí být o nastavený poměr větší než daný percentil ostatních buněk. Např. při poměru 2 a percentilu 0,9 to znamená, že pravděpodobnost vybrané hypotézy bude dvakrát větší než 90 % ostatních pravděpodobností.

Tento způsob výběru zajišťuje, aby byla vybrána hypotéza až ve chvíli, kdy její pravděpodobnost jednoznačně převyšuje ostatní. Zároveň se tak nevyklučují další lokální maxima pravděpodobnosti (další možné hypotézy, jejichž pravděpodobnost převyšuje ostatní, ale není absolutním maximem). Lokalizace na mřížce je navíc omezena minimálním počtem iterací. K lokalizaci jsou totiž potřeba údaje z odometrie (parametr u_t) a minimální počet iterací zaručí, že robot ujede určitou vzdálenost (její velikost závisí na minimálním počtu iterací), parametr u_t bude nenulový a lokalizace bude přesnější. Pokud by u_t bylo nulové, lokalizace by se opírala pouze o GPS.

Poté, co se určí počáteční poloha robotu, spustí se plánování cesty. Výstupem plánování je seznam uzlů, kterými musí robot projet. V tuto chvíli je robot stále řízen algoritmem rozpoznání cesty až do doby, než se podle GPS nepřiblíží ke křižovatce. Na

křižovatce zvolí robot strategii pohybu podle navazujících uzlů. V případě, že následující uzel má pouze 2 sousední uzly, tedy uzel, z kterého robot přijel, a jeden následující uzel, nejedná se o křižovatku. Robot proto může pokračovat pouze s řízením podle rozpoznání cesty.

V opačném případě, má-li uzel více než dva sousední uzly, musí robot řešit situaci na křižovatce podle následujícího uzlu, do kterého směřuje. Jako první algoritmus zkontroluje, je-li následující cesta nejvíce vlevo nebo vpravo. Pak stačí, aby se po dobu průjezdu křižovatkou robot držel podle rozpoznávání cesty u levého, případně pravého, okraje cesty. U ostatních cest, které nejsou nejvíce vlevo nebo vpravo se snaží robot určit směr další jízdy především podle kompasu. Směrování kompasem s sebou ale nese riziko. Pokud by se robot nacházel v prostoru s elektromagnetickým rušením, data z kompasu by nebyla spolehlivá a robot by se mohl navigovat na špatnou cestu.

Z důvodu možné chybné navigace na křižovatce nebo chybné lokalizace na začátku běží lokalizační algoritmus po celou dobu jízdy robotu. V každém dalším uzlu, do kterého robot dojde (nemusí se nutně jednat o křižovatku), se vytvoří nová mřížka, do které se zkopírují pravděpodobnosti ze staré mřížky, pokud obsahují shodné cesty. Nová mřížka zahrnuje všechny cesty ve větším okolí bez ohledu na jejich azimut na rozdíl od počáteční lokalizace.

Poloha robotu (podle koncových uzlů cesty, po které robot jede) se periodicky po každé iteraci lokalizace srovnává s aktuální mřížkou. Jestliže je zjištěna neshoda, algoritmus znovu naplánuje cestu do cíle a robot pokračuje v jízdě.

Během celé jízdy měří ultrazvukové dálkoměry vzdálenosti před i za robotem, aby se předešlo srážce s překážkou.

6. Implementace

Navigační systém byl implementován pod operačním systémem Linux (distribuce Ubuntu 12.04) v jazyce C++. Vnitřní struktura softwaru a komunikace s hardwarem je znázorněna na obrázku 13. Při návrhu byl kladen důraz na nezávislost jednotlivých částí programu. Je tak zajištěna modularita a při zachování daného rozhraní se dají části snadno nahradit.

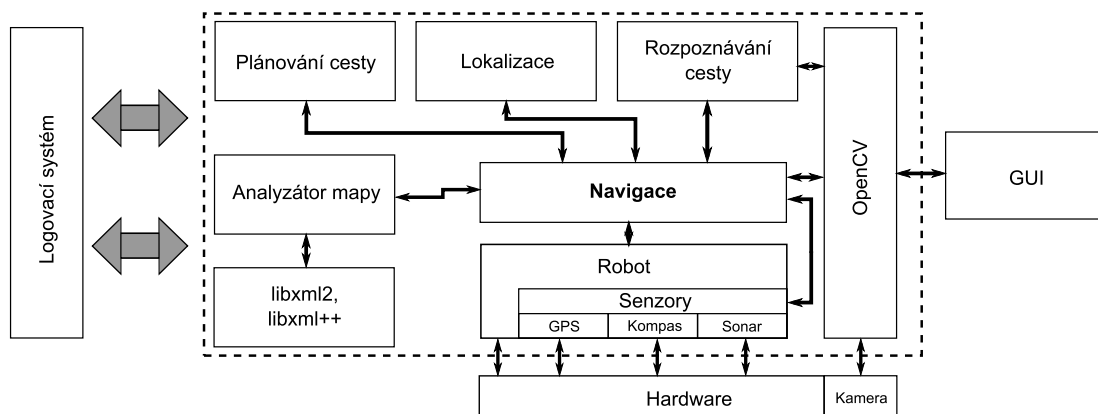
6.1. Popis funkce programu

Po startu programu se inicializuje logovací systém. Jako další se vytvoří instance robotu, kterému je třeba přiřadit konfigurační soubor. Soubor obsahuje informace o komunikačních portech, použité kamere, mapových podkladech, adresách senzorů a souřadnicích cíle. Po inicializaci se vytvořená instance robotu předá třídě navigace.

Třída navigace inicializuje ostatní potřebné součásti - analyzátor mapy, rozpoznání cesty z obrazu a plánování trasy. Kromě inicializace spustí navigace na začátku paralelně další tři vlákna. První vlákno se stará o komunikaci se senzory. Periodicky čte data ze senzorů, aby byly při výpočtech neustále k dispozici aktuální hodnoty.

V druhém vlákne běží algoritmus rozpoznávání cesty. Výsledky rozpoznávání se neaplikují přímo na řízení motorů, ale ukládají se do sdílené proměnné, odkud je může číst hlavní vlákno. Stejně jako u prvního vlákna jsou tak k dispozici aktuální informace o cestě před robotem. Poslední vlákno zajišťuje vykreslování uživatelského prostředí - obraz z kamery s vyznačenou cestou a mapu, do které se zaznamenává poloha robotu a naplánovaná trasa.

Program využívá dvě knihovny třetí strany. První knihovnou je OpenCV (dokumentace knihovny dostupná v [3]). OpenCV je volně šiřitelná a nabízí velké množství funkcí pro zpracování obrazu, tvorbu základního grafického rozhraní i strojové učení. V programu je použita pro snímání obrazu z kamery, jeho předzpracování, zobrazení rozpoznané cesty a k vykreslení mapy. Druhá knihovna, libxml2 a nástavba pro C++ libxml++, zpracovává mapové podklady, které jsou ve formátu XML (viz 5.1).



Obr. 13. Vnitřní struktura navigačního systému a komunikace s hardwarem

7. Experimentální výsledky

Cílem práce bylo vytvořit navigační systém, avšak při testování byl kladen důraz především na vývoj a zlepšování lokalizace. Z toho důvodu jsou popsány především výsledky dosažené při lokalizaci.

Testování lokalizace probíhalo za náročnějších podmínek v zastavěném prostředí mezi panelovými domy, většinou při zatažené obloze. Příjem GPS tak nebyl ideální a na datech se projevovala chyba souřadnic. Proces lokalizace byl nejprve simulován na uměle vytvořených datech a poté vyzkoušen na skutečném robotu.

Na testování lokalizace navazuje experiment, ve kterém byl vyzkoušen celý navigační systém. Robot jel opakovaně po stejné trati. Podle úspěšnosti, s jakou projížděl jednotlivé uzly cesty v souladu s mapou a naplánovanou trasou, byla vyhodnocena spolehlivost navrženého systému.

7.1. Simulace lokalizace na umělých datech

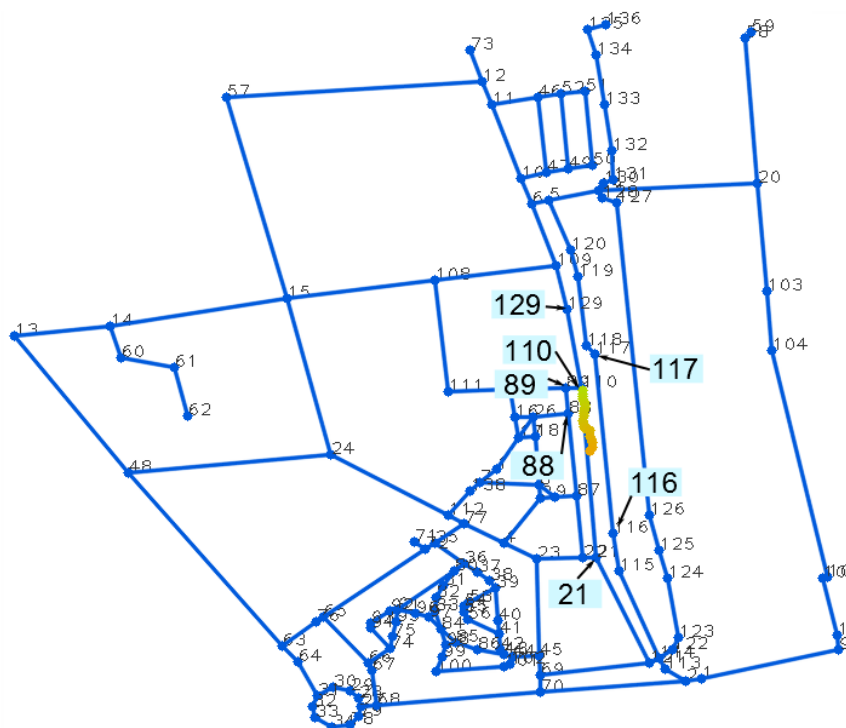
Z důvodu otestování správnosti algoritmu byla první lokalizace vyzkoušena na uměle vygenerovaných datech. Data byla vytvořena postupným „naklikáním“ bodů do mapy a vyčtení jejich souřadnic. První bod byl použit pro první krok lokalizace - určení počátečního seznamu možných cest. Další body simulovaly pohyb robotu. Jak jednotlivé body v mapě ležely ukazuje obrázek 14 (obrázek mapy byl vykreslen navigačním systémem). Do obrázku byly navíc vyznačeny ty uzly mapy, které jsou potřebné pro další popis. Generované body jsou barevně odlišeny podle pohybu (zde podle pořadí, jak byly generovány). Barva přechází ze světle zelené barvy počátečního bodu do okrové barvy koncového bodu. Při simulaci byl povolen také vyšší limit počáteční lokalizace, aby bylo možné vyzkoušet práci nad větší mřížkou.

V prvním kroku označil algoritmus za možné cesty hrany s indexy uzlů 129-110, 21-110, 117-116, 88-89. Ty pak byly rozděleny do mřížky s rovnoměrným rozdělením pravděpodobnosti. Po první iteraci byly výrazné hodnoty pravděpodobnosti u hran 129-110 a 21-110, konkrétně pak u uzlu 110. S každou další iterací klesaly všechny pravděpodobnosti postupně k nule kromě hrany 21-110. Na ní byl patrný pohyb směrem k uzlu 21. Kompletní přehled vývoje pravděpodobnosti buněk mřížky je z důvodu rozsáhlosti v příloze B.1. Pravděpodobnosti jsou znázorněny jako sloupcový graf.

7.2. Testování lokalizace na skutečném robotu

Testy byly provedeny tak, aby vyzkoušely různé situace, do kterých se může robot dostat. Dále budou uvedeny dva úspěšné pokusy a jeden pokus s chybným výsledkem.

Na obrázcích, které ukazují mapu a pohyb robotu, jsou naměřené GPS souřadnice opět vyznačeny zeleným až okrovým kolečkem, podle pohybu robotu. Skutečná počáteční poloha robotu je pak zakreslena červeným bodem a směr jízdy je vyznačen větší šipkou. Zároveň byly označeny uzly mapy, kterých se bude týkat popis.



Obr. 14. Rozložení generovaných dat v mapě

7.2.1. Pokus s jednou očekávanou možnou hranou

Účelem jednoho z testů bylo vyzkoušet chování systému v takové situaci, kde se po prvním kroku lokalizace očekávala jako výsledek pouze jedna hrana. Ostatní hrany by podle očekávání měly vyřazeny kvůli velké odchylce azimutu. Vybraná cesta (hrana 87-9) byla ve svém okolí jediná, která vedla z východu na západ, ostatní cesty vedly ze severu na jih. Situaci je znázorněná na obrázku 15.

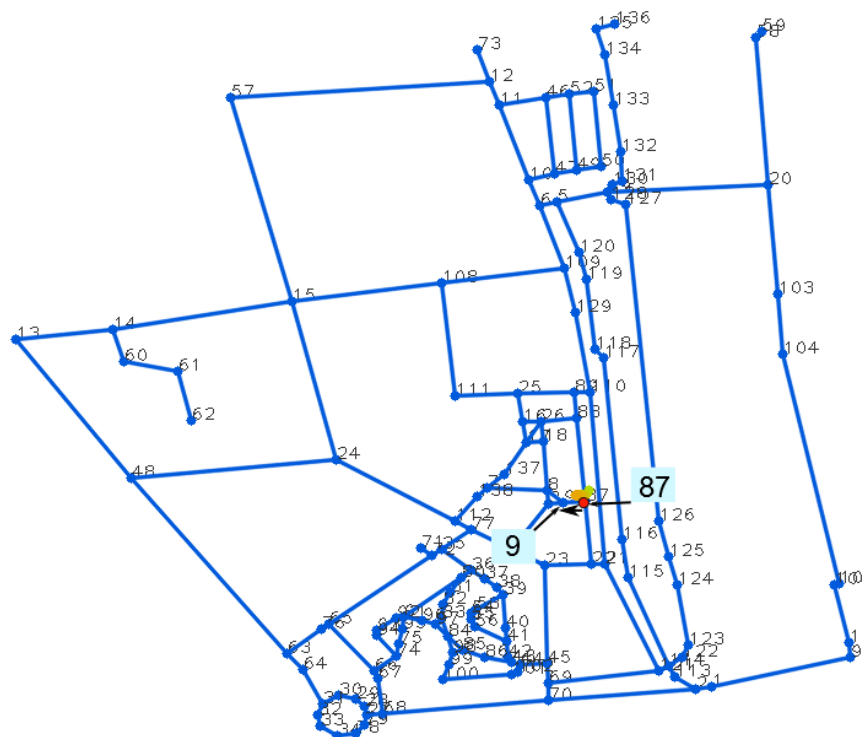
Podle očekávání byla za možnou polohu robotu označena hrana 9-87. Vlivem nepřesných souřadnic byla pravděpodobnost ze začátku blízko uzlu 87. To se však spravilo díky pohybu k uzlu 9. Na průběhu pravděpodobnosti je tento pohyb vidět a poloha byla nakonec určena správně. Podrobné průběhy jsou znázorněny v příloze B.2.

7.2.2. Pokus s více možnostmi na malém prostoru

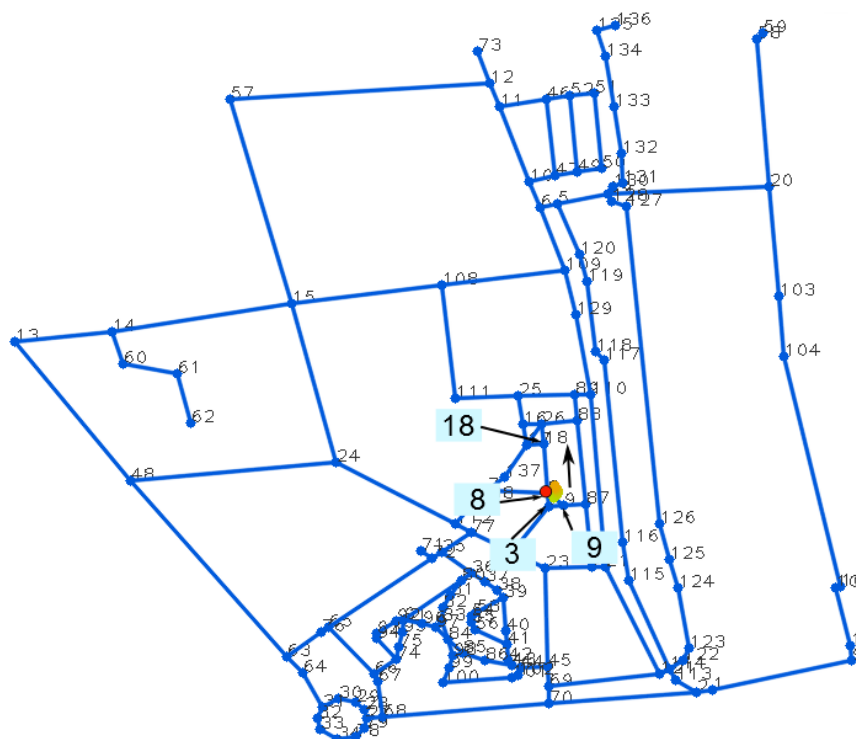
Místo pro další test bylo vybráno na rozdíl od prvního pokusu tak, aby na začátku lokalizace bylo více možností. Jedná se o reálnou obdobu simulace z kapitoly 7.1, kde bylo také více možností. Další požadavek pro výběr místa byl takový, aby se na malém prostoru nacházelo více cest s podobným azimutem. Tomu odpovídala trojúhelníková křižovatka mezi uzly 8, 3 a 9 (viz obrázek 16).

Za možné cesty, na kterých se robot nacházel, byly označeny cesty 9-8, 3-8 a 18-8. Vyšší pravděpodobnost připadla do blízkosti uzlu 8 u všech cest. Přestože získané GPS souřadnice ležely velmi blízko sebe, díky pohybu se pravděpodobnosti hran 9-8 a 3-8 snížily k nule a jako jediná možná cesta zůstala hrana 18-8. Kompletní průběhy pravděpodobností buněk mřížky jsou v příloze B.3.

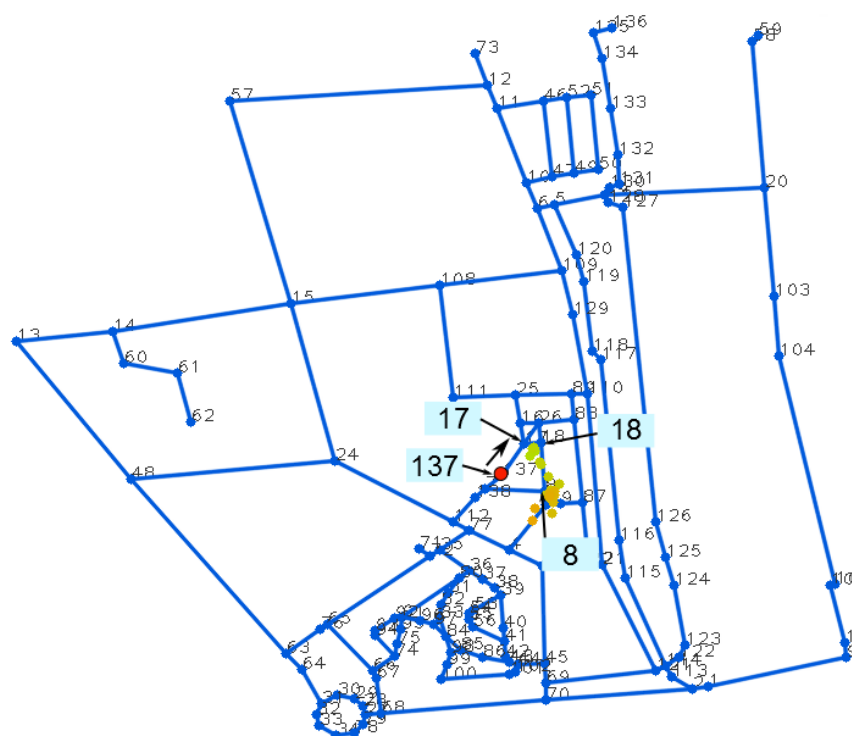
7. Experimentální výsledky



Obr. 15. Volba místa pro pokus s jednou očekávanou možnou hranou



Obr. 16. Volba místa pro pokus s více možnostmi na malém prostoru



Obr. 17. Pokus s chybným výsledkem lokalizace

7.2.3. Vliv velké nepřesnosti GPS

Jako třetí test lokalizace je uveden případ, na kterém bude demonstrován vliv velké nepřesnosti měřených souřadnic GPS. Umístění robotu bylo vybráno náhodně bez jakýchkoliv požadavků na okolní cesty jako v předchozích případech. Počáteční poloha byla zvolena v blízkosti uzlu 137 se směrem jízdy k uzlu 17.

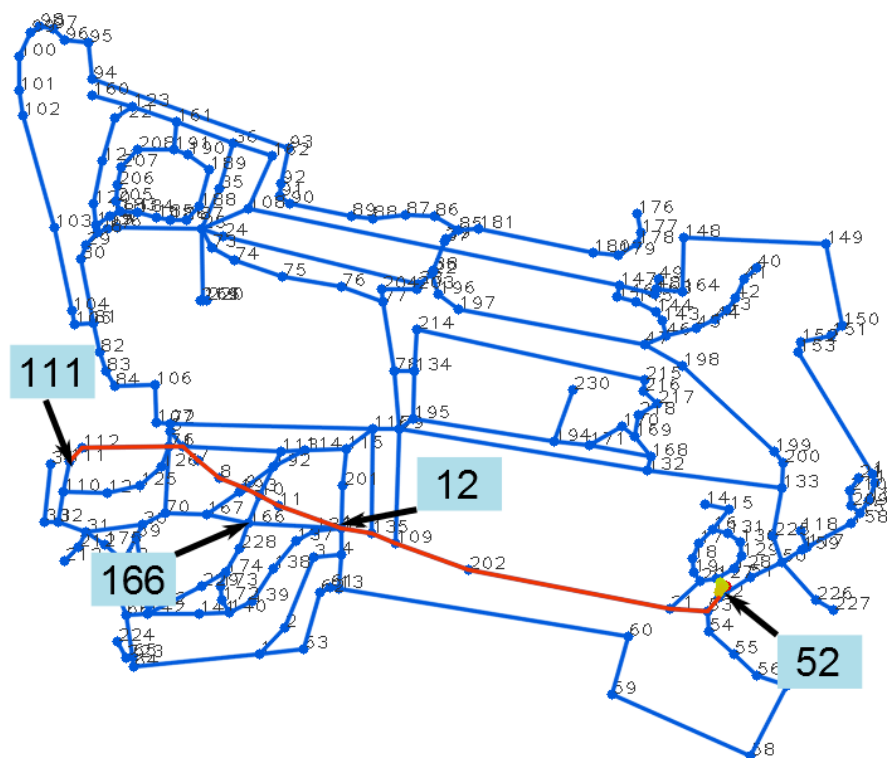
Podle azimutu byly správně vybrány hrany 17-137 a 18-8, avšak vzdálenost hrany 18-8 byla ve skutečnosti větší než nastavený limit. Došlo totiž k velké chybě GPS souřadnic a proto byla cesta zařazena mezi možné polohy robotu. Odchyłka od skutečné polohy byla až 40m. To mělo velký vliv na vyhodnocování pravděpodobnosti buněk mřížky a hrana 17-137 tak byla vyloučena z přípustných hrana. Podle výsledků se robot pohyboval po cestě 18-8, ale ve skutečnosti jel po cestě 17-137. Všechny grafy pravděpodobnosti jsou pro úplnost v příloze B.4.

Tento test ukázal, že vliv přesnosti GPS je stále velký, i přestože se navržený systém nespolehá pouze na ni. Ve velmi nepříznivých podmínkách tak může dojít k chybné lokalizaci a následné navigaci.

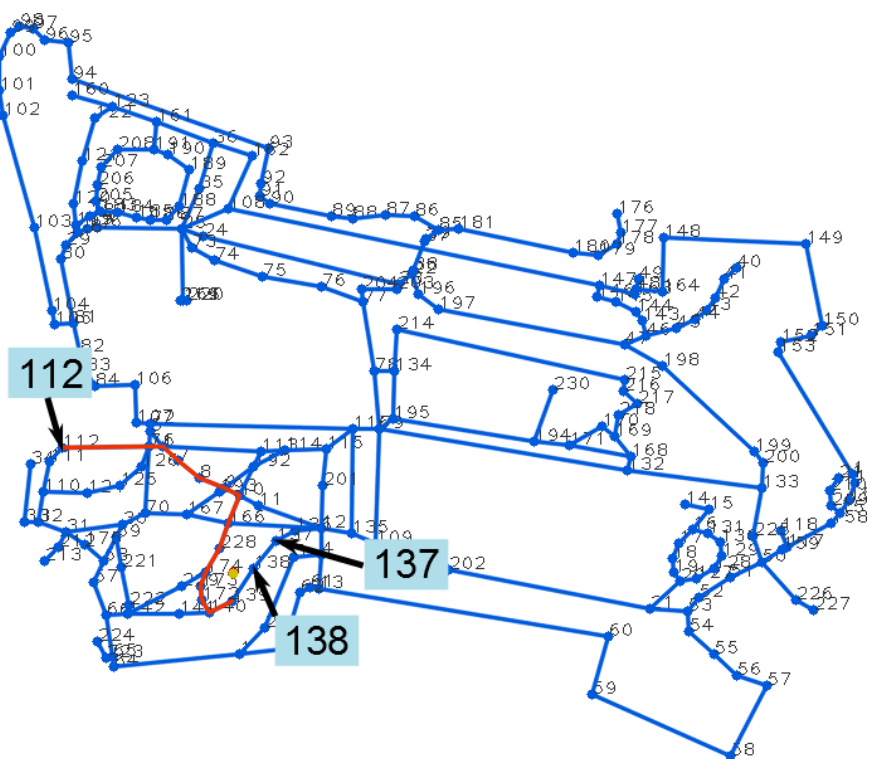
7.3. Test celého navigačního systému

Účelem posledního testu bylo vyzkoušet celý navržený systém a jeho spolehlivost v reálném prostředí. Robot při něm projížděl opakovaně stejné trasy. Dalo se tak předpovídat jeho chování v jednotlivých situacích a v případě odlišných reakcí bylo možné odhalit případnou chybu. Testování probíhalo ve Vyšehradských sadech na dvou různých trasách. Jejich mapa včetně vyznačených cest jízdy robotu a důležitých uzlů jsou na obrázcích 18 a 19 (mapy byly vytvořeny navigačním systémem). Zelené až okrové body zvýrazňují změřené GPS souřadnice při počáteční lokalizaci.

7. Experimentální výsledky



Obr. 18. Mapa Vyšehradských sadů, vyznačena delší trasa



Obr. 19. Mapa Vyšehradských sadů, vyznačena kratší trasa

7. Experimentální výsledky

První cesta (obrázek 18) byla vybrána mezi uzly 52 a 111. Její délka z počáteční polohy do cílového bodu byla cca 350 m a robot ji ujel v průměru za 15 minut. Celkem bylo provedeno šest jízd, tři ve směru od uzlu 52 k uzlu 111 a tři ve směru opačném. Na cestě k uzlu 111 byla jediným problémem křižovatka u uzlu 12, na které se musel robot řídit pouze podle kompasu. Dvakrát ze tří jízd v tomto směru sjel na vedlejší cestu (12-166) a to především protože nestihl včas dostatečně zatočit a algoritmus rozpoznání cesty další zatáčení nedovolil. Dalším důvodem byla malá odchylka azimutů jednotlivých cest, které z křižovatky vedly. Během několika ujetých metrů však díky lokalizaci na mřížce dokázal robot určit svou současnou polohu, naplánoval znovu cestu do cíle a zbytek trasy dojel bez problémů. Opačný směr cesty (od uzlu 111 k uzlu 52) projel robot ve všech třech případech bez větších komplikací.

Druhá vybraná trasa byla kratší, měřila přibližně 190 m (obrázek 19). Robot vyjízďel z úseku mezi uzly 137-138 směrem ke 138, jako cíl byl zvolen uzel 112. V tomto případě byly provedeny tři jízdy vždy ve směru uzlů 138-112. Nejkratší cesta by vedla přes uzel 137. Robot by se ale musel otáčet, což z důvodů popsaných v 5.4 nebylo možné. Proto plánovací algoritmus našel delší cestu, na které se ale robot nemusel otáčet. Jelikož naplánovaná cesta byla dostatečně široká, její barva a barva okolí byly kontrastní a z křižovatek nevycházely více než tři další cesty, projel celou trasu robot bez problémů a vždy dorazil do cíle.

8. Závěr

Cílem práce bylo navrhnout a implementovat navigační systém robotu. Vzhledem ke složitosti byl problém rozložen do několika dílčích problémů a ty byly postupně řešeny. Aby byl návrh řídicího systému robotu možný, byl pro shrnutí uveden přehled běžně používaných senzorů. V návaznosti pak byly stručně popsány základní metody a strategie navigace robotu.

Návrh systému vycházel z popsaných metod navigace a opíral se o popsané senzory. Důležitými kritérii při návrhu systému byly právě jednoduchost a použité prostředky. Proto byl kladen důraz na návrh takového systému, který by využíval pouze běžně dostupné senzory a především by nevyžadoval jakékoliv speciální mapování či složitou úpravu mapových podkladů nebo úpravu prostředí. Důležitým aspektem při implementaci bylo využití volně dostupných knihoven.

Tento cíl byl splněn díky použití map projektu OpenStreetMap[©]. Jejich výhodou je dostupnost, aktuálnost a snadné zpracování v offline režimu ve formě strukturovaného XML souboru. Mapy po vyfiltrování nepotřebných objektů navíc obsahují pouze definice cest, což umožňuje rychlejší zpracování při provozu robotu.

Navržený systém prošel několika testy se zaměřením na lokalizaci. Lokalizace je velmi důležitý a obtížný krok v navigaci, proto jí byla věnována větší pozornost. Mimoto probíhá lokalizace během jízdy opakovaně, kvůli možné chybě v navigaci, proto je potřeba, aby byla co nejpřesnější. Výsledky testů odpovídaly skutečné poloze robotu ať se jednalo o simulaci na uměle vytvořených datech nebo o skutečný pohyb robotu prostředím. Použitá technika lokalizace na mřížce kompenzovala běžné chyby GPS (chyby v řádu jednotek metrů). Velké odchylky GPS ale měly příliš velký dopad na výsledky a docházelo proto k chybné lokalizaci.

Lokalizace a navigace ve venkovním prostředí by se z toho důvodu neměla spoléhat pouze na GPS. Cílem dalšího vývoje jsou proto jiné možnosti lokalizace. Použití jiných senzorů a technik by snížilo význam GPS a problémy s ní spojené by se mohly eliminovat. Další důležitou částí potřebnou pro robustní navigační systém je počítačové vidění a rozpoznání cesty z obrazu. Současný systém nedokáže rozpoznat z obrazu křižovatky. Tato skutečnost je zatím kompenzována odometrií a souřadnicemi GPS, ze kterých je možné určit, že se robot blíží ke křižovatce. Obě v současné době používané techniky jsou však zatíženy chybami, které mohou vést ke špatné navigaci.

Kromě rozpoznání křižovatek by také byla zajímavá implementace přepínání očekávané barvy cesty podle aktuální pozice robotu. Robot by si podle mapy a své pozice načel z paměti soubor, který by obsahoval informace o konkrétní cestě a upravil by tak chování rozpoznání cesty z obrazu. Tento mechanismus by předešel komplikacím, pokud by robot měl například sjet z asfaltové cesty na cestu polní. Pokud bychom obecně definovali třídy cest, např. asfaltová, polní, lesní, parková, nebylo by zapotřebí další mapování. Typ cesty se dá vyčíst z použitých map.

Literatura

- [1] Autorská práva a licence openstreetmap. <http://www.openstreetmap.org/copyright>. [Online], cit. 2013-12-25.
- [2] Dokumentace knihovny libxml. <http://www.xmlsoft.org/>. [Online], cit. 2013-12-25.
- [3] Dokumentace knihovny opencv. <http://docs.opencv.org/>. [Online], cit. 2013-12-25.
- [4] Gpx: The gps exchange format. <http://www.topografix.com/gpx.asp>. [Online], cit. 2014-04-17.
- [5] Movable type scripts: Calculate distance, bearing and more between latitude/longitude points. <http://www.movable-type.co.uk/scripts/latlong.html>. [Online], cit. 2013-12-25.
- [6] Nástavba knihovny libxml pro jazyk c++. <http://libxmlplusplus.sourceforge.net/>. [Online], cit. 2013-12-25.
- [7] Official u.s. government information about the global positioning system (gps) and related topics. <http://www.gps.gov/>. [Online], cit. 2014-04-13.
- [8] Technická specifikace cmeps09. <http://www.robot-electronics.co.uk/html/cmeps09doc.htm>. [Online], cit. 2013-12-25.
- [9] Technická specifikace srf02. <http://www.robot-electronics.co.uk/html/srf02tech.htm>. [Online], cit. 2013-12-25.
- [10] **Borenstein, J., Everett, H.R., Feng, L., Wehe, D.** *Where am I? Sensors and Methods for Mobile Robot Positioning*. University of Michigan, new edition edition, 1996.
- [11] **Borenstein, J., Everett, H.R., Feng, L., Wehe, D.** Mobile robot positioning - sensors and techniques. *Journal of Robotic Systems, Special Issue on Mobile Robots*, 14(4):231–249, 1997.
- [12] **Haasz, V., Ripka, P.** Materiály předmětu a3b38sme, senzory a měření (přednášky), 2013.
- [13] **Krajník, T., Szücssová, H., Faigl, L., Vonásek, V., Fišer, O., Přeučil, L.** A visual navigation system for robotour competition, 2009.
- [14] **Krajník, T., Szücssová, H., Svědihrová, M., Chudoba, J., Vonásek, V.** Jak to vidí roboti iv. *Robot revue: magazín ze světa robotiky*.
- [15] **Krajník, T., Szücssová, H., Svědihrová, M., Chudoba, J., Vonásek, V.** Jak to vidí roboti v. *Robot revue: magazín ze světa robotiky*.

- [16] **LOCOSYS Technology Inc.** Ls20031 (datasheet), 2006.
- [17] **Thrun, S., Burgard, W., Fox, D.** *Probabilistic robotics*. Massachusetts: MIT Press, 2006.

Příloha A.

Obsah příloženého CD

Příložené CD obsahuje elektronickou verzi práce ve formátu PDF, veškerý software (včetně zdrojových kódů) a schemata zapojení elektronických součástí robotu. Software mikrokontrolérů je přiložen ve formě projektů pro vývojové prostředí CodeVision AVR 1.25. Software pro PC je také přiložen jako projekt pro vývojové prostředí Code-Blocks. Schemata zapojení byla vytvořena v programu Eagle 5.11.0. Dále CD obsahuje fotogalerii robotu během vývoje a videa, která zachycují testování rozpoznávání cesty z obrazu. Struktura CD je popsána v tabulce 4.

Soubor, složka	Popis
bp_stejskal.pdf	Elektronická verze práce ve formátu PDF
fotogalerie	Fotogalerie robotu během vývoje
HW	Schemata zapojení elektronických součástí robotu. Obsahuje 3 podsložky - motory , senzory a uart2usb . Složka motory obsahuje schemata zapojení logické části řízení motorů a zapojení výkonového H-můstku. Složka senzory obsahuje schema zapojení desky, která obsluhuje senzory robotu. Zapojení převodníku UART-USB je ve složce uart2usb .
SW	Software robotu včetně zdrojových kódů. V podložce mikrokontrolery jsou uloženy firmwary mikrokontrolérů, které řídí motor a obsluhují senzory. Podsložka PC obsahuje projekt navrženého navigačního systému.
videa	Videa natočená při testování rozpoznávání cesty z obrazu

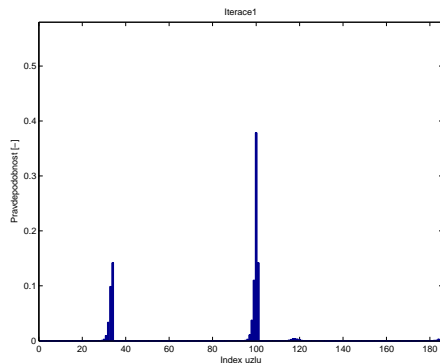
Tab. 4. Struktura příloženého CD

Příloha B.

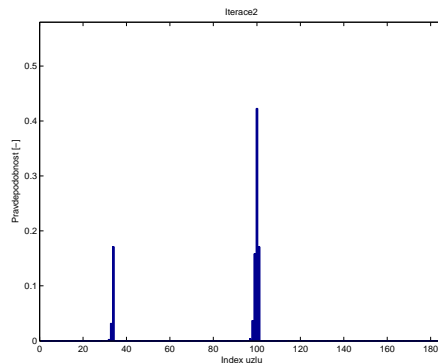
Experimentální výsledky lokalizace na mřížce

B.1. Simulace s uměle vytvořenými daty

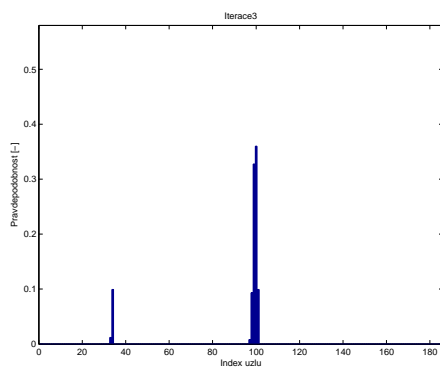
Na obrázcích 20a až 20s jsou průběhy pravděpodobnosti buněk mřížky vytvořené při simulaci v kapitole 7.1. Hraně 129-110 připadají sloupce 1 až 33, hraně 21-110 sloupce 34 až 100, hraně 117-116 sloupce 101 až 173 a hraně 88-89 sloupce 174 až 185.



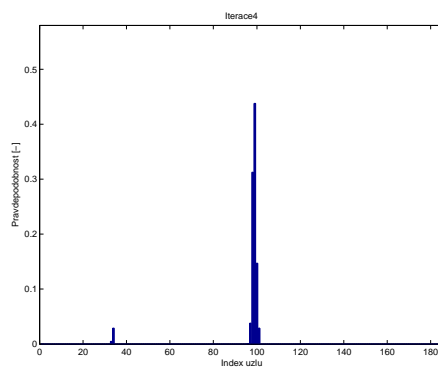
(a) Iterace 1



(b) Iterace 2

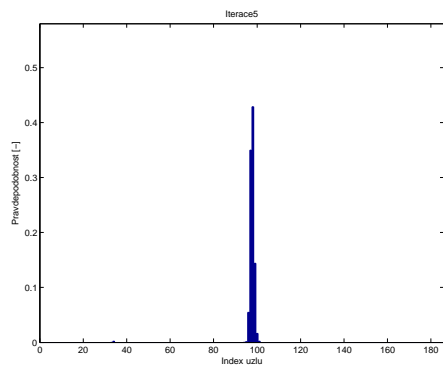


(c) Iterace 3

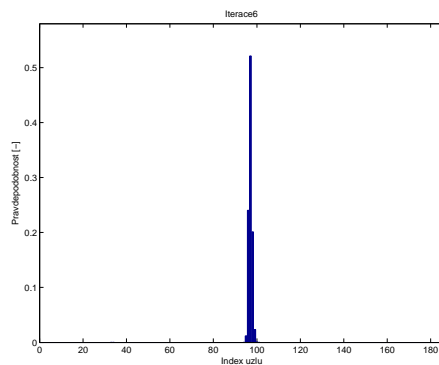


(d) Iterace 4

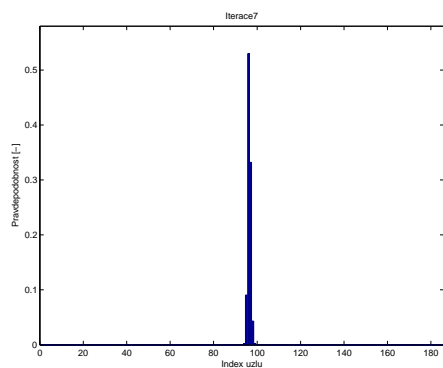
Příloha B. Experimentální výsledky lokalizace na mřížce



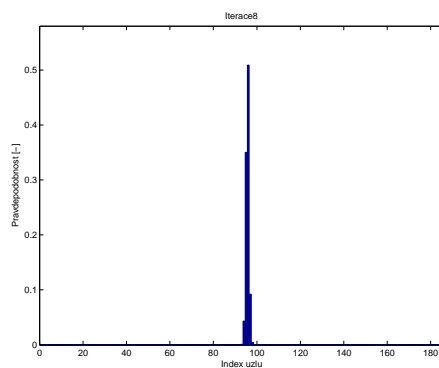
(e) Iterace 5



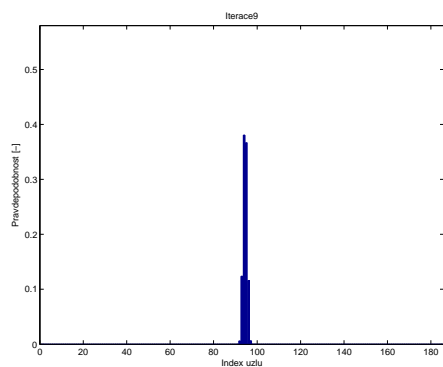
(f) Iterace 6



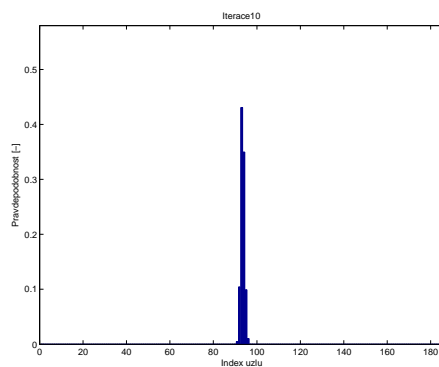
(g) Iterace 7



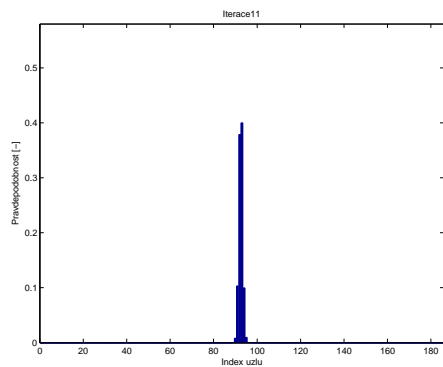
(h) Iterace 8



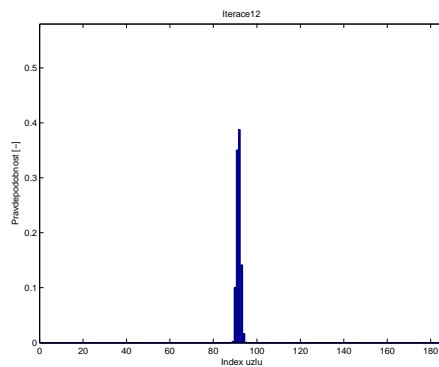
(i) Iterace 9



(j) Iterace 10

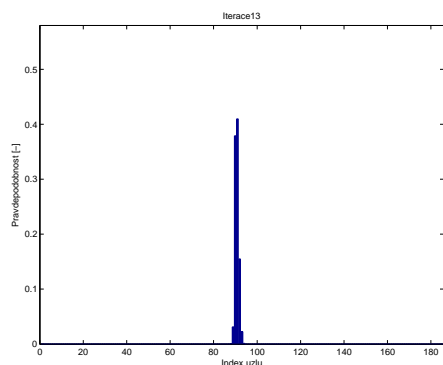


(k) Iterace 11

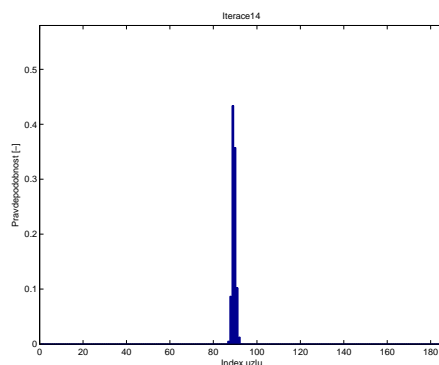


(l) Iterace 12

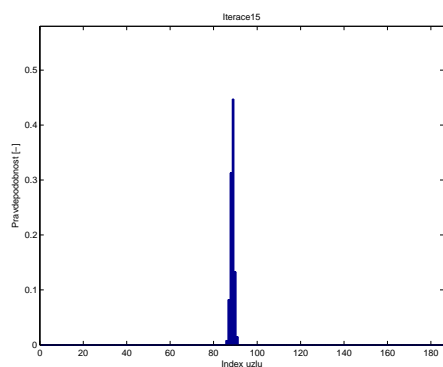
Příloha B. Experimentální výsledky lokalizace na mřížce



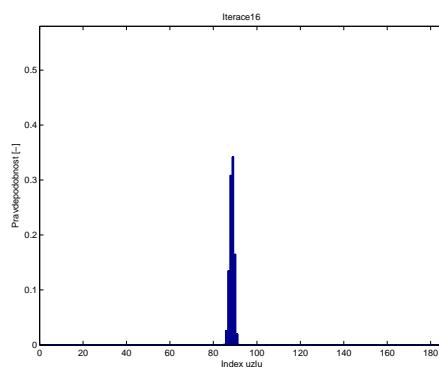
(m) Iterace 13



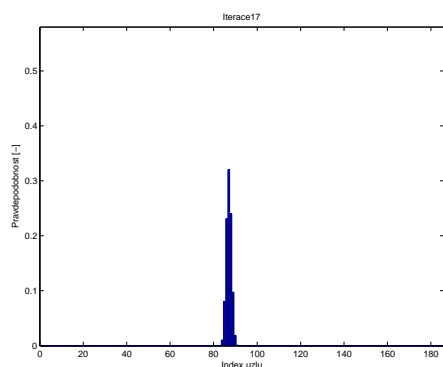
(n) Iterace 14



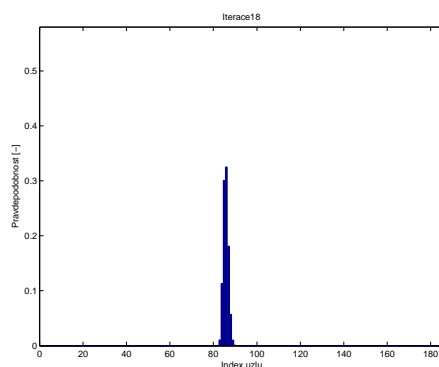
(o) Iterace 15



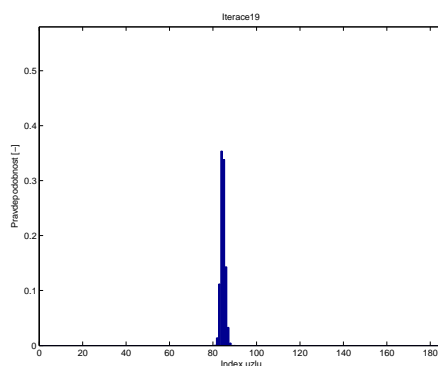
(p) Iterace 16



(q) Iterace 17



(r) Iterace 18

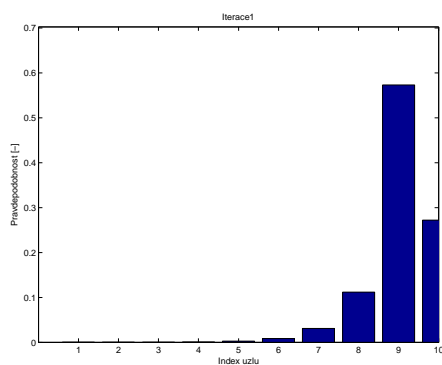


(s) Iterace 19

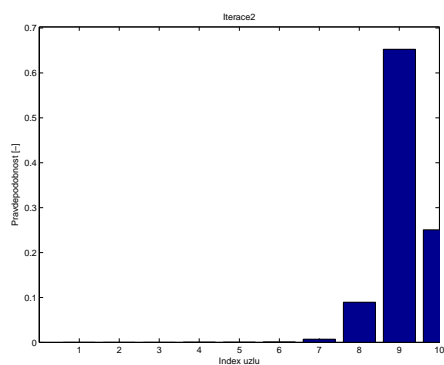
Obr. 20. Kompletní přehled vývoje pravděpodobnosti simulace lokalizace z kapitoly 7.1

B.2. Pokus s jednou očekávanou možností

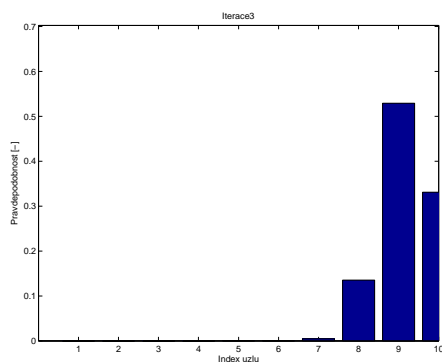
Na obrázcích 21a až 21s jsou průběhy pravděpodobnosti buněk mřížky vytvořené při jednoduchém pokusu s jednou očekávanou možnou hranou v kapitole 7.2.1. Indexy odpovídají buňkám hrany 9-87 v tomto pořadí.



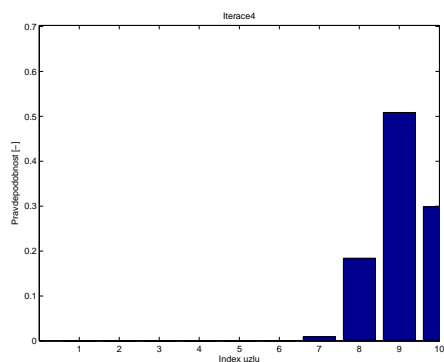
(a) Iterace 1



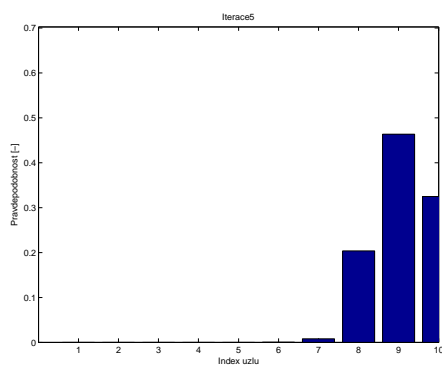
(b) Iterace 2



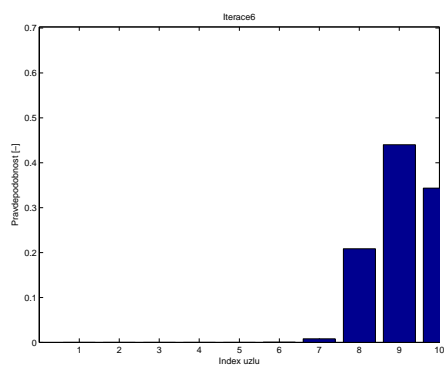
(c) Iterace 3



(d) Iterace 4

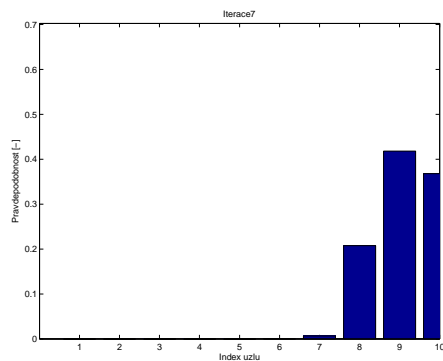


(e) Iterace 5

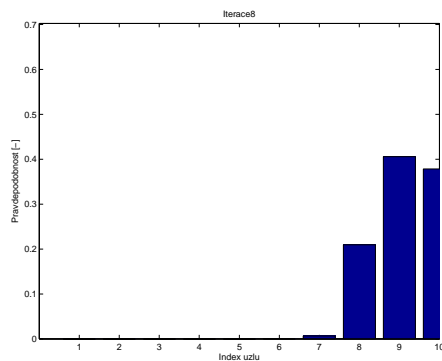


(f) Iterace 6

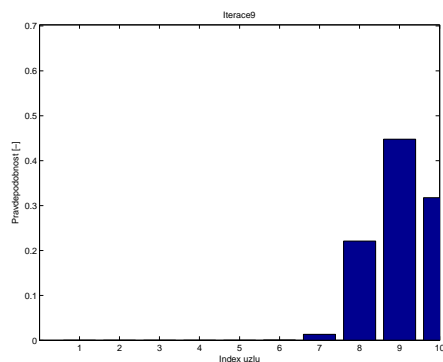
Příloha B. Experimentální výsledky lokalizace na mřížce



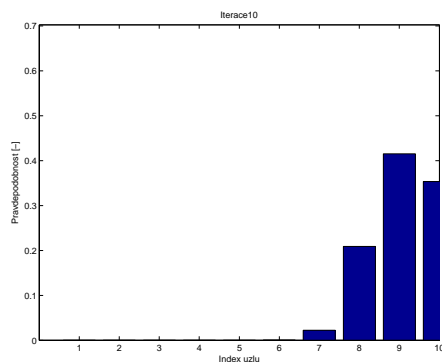
(g) Iterace 7



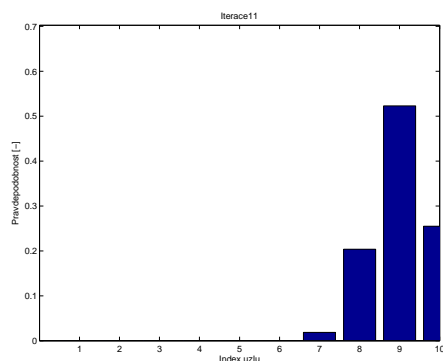
(h) Iterace 8



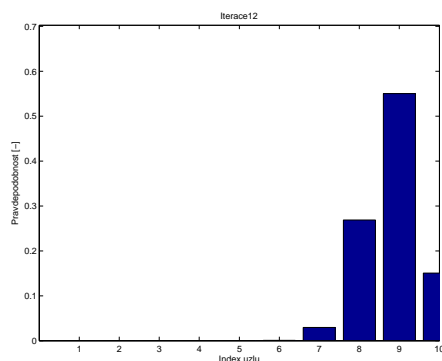
(i) Iterace 9



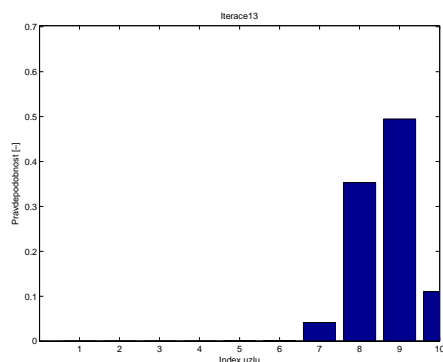
(j) Iterace 10



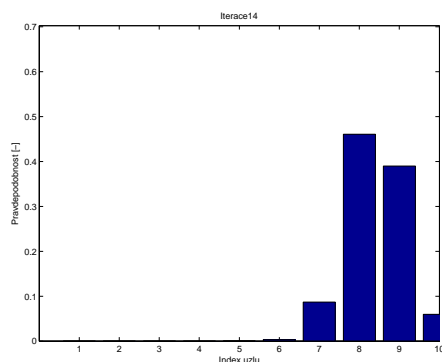
(k) Iterace 11



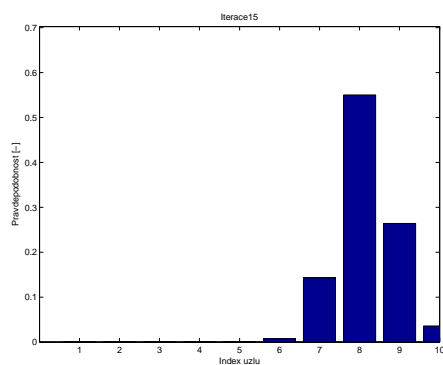
(l) Iterace 12



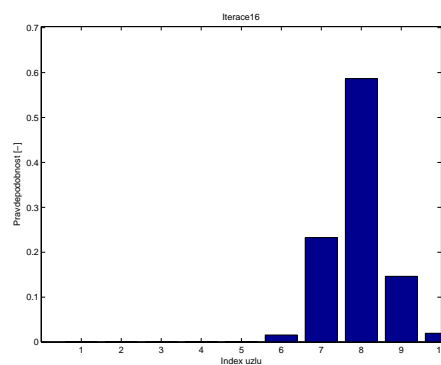
(m) Iterace 13



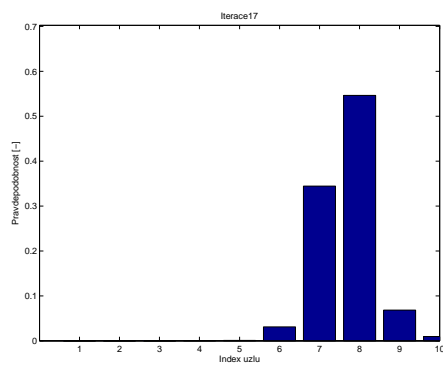
(n) Iterace 14



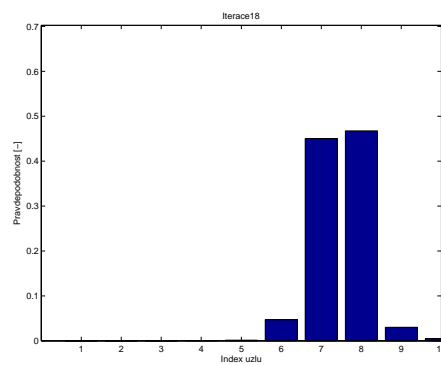
(o) Iterace 15



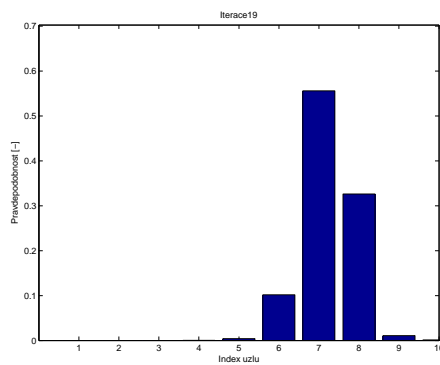
(p) Iterace 16



(q) Iterace 17



(r) Iterace 18

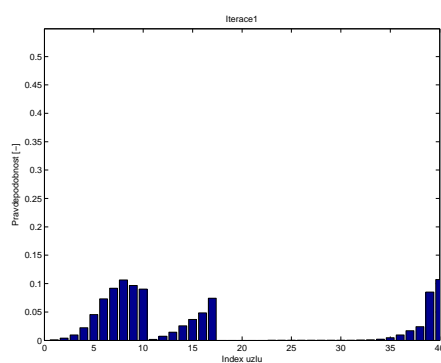


(s) Iterace 19

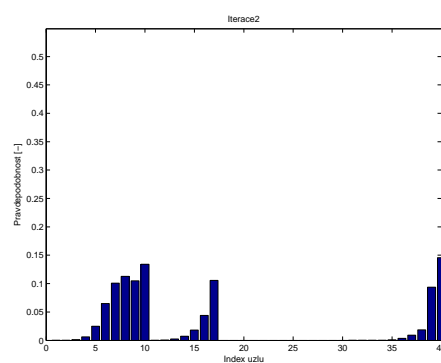
Obr. 21. Rozložení pravděpodobností při lokalizace na mřížce, příklad 4

B.3. Pokus s více možnostmi na malém prostoru

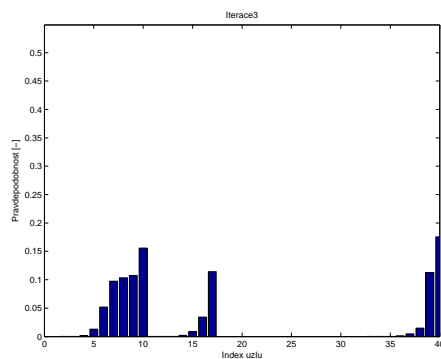
Obrázky 22a až 22s ukazují průběhy pravděpodobnosti buněk mřížky z pokusu uvedeného v kapitole 7.2.2. Indexy 1 až 10 odpovídají buňkám hrany 9-8, indexy 11 až 17 náleží hraně 3-8 a hraně 18-8 připadají indexy 18 až 40 v uvedeném pořadí.



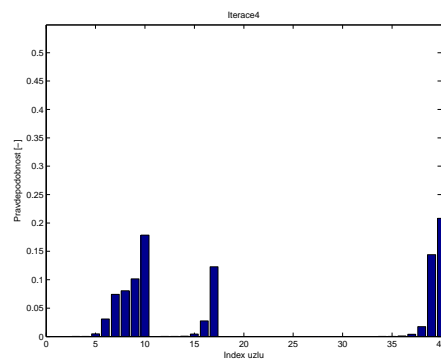
(a) Iterace 1



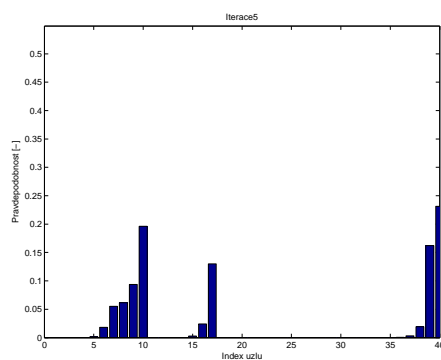
(b) Iterace 2



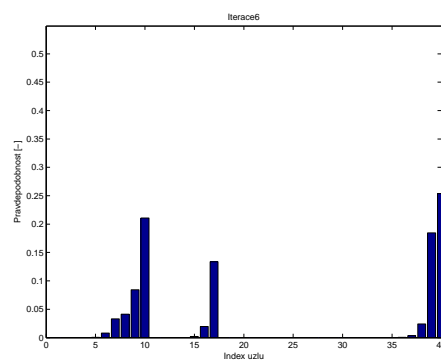
(c) Iterace 3



(d) Iterace 4

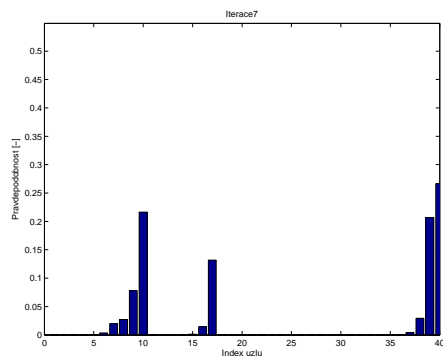


(e) Iterace 5

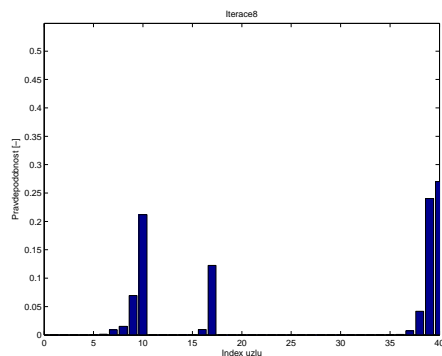


(f) Iterace 6

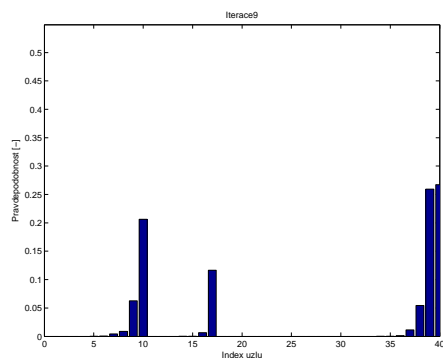
Příloha B. Experimentální výsledky lokalizace na mřížce



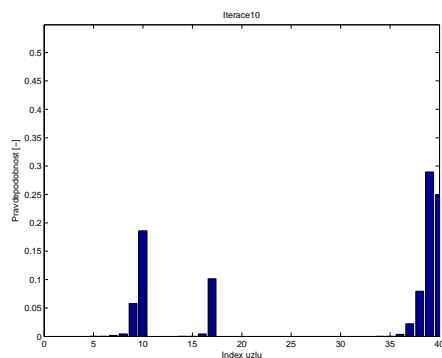
(g) Iterace 7



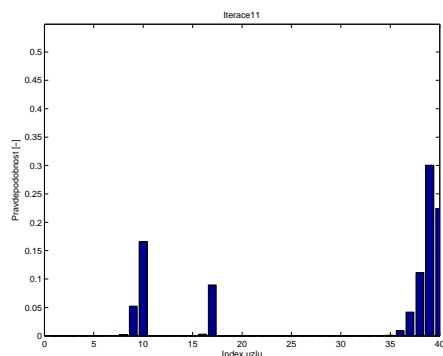
(h) Iterace 8



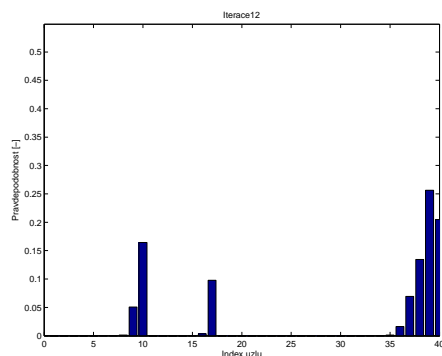
(i) Iterace 9



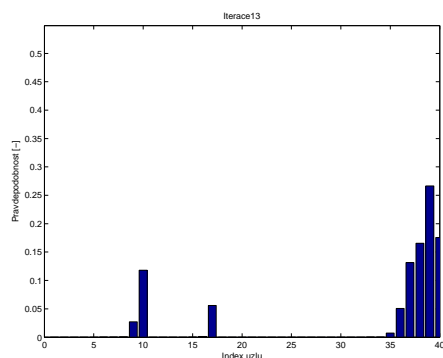
(j) Iterace 10



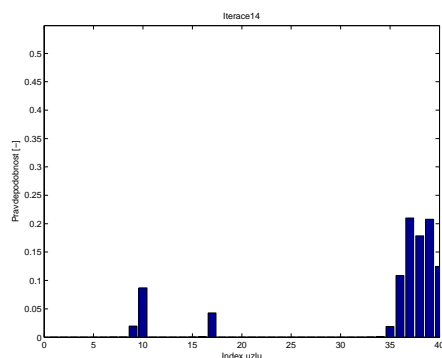
(k) Iterace 11



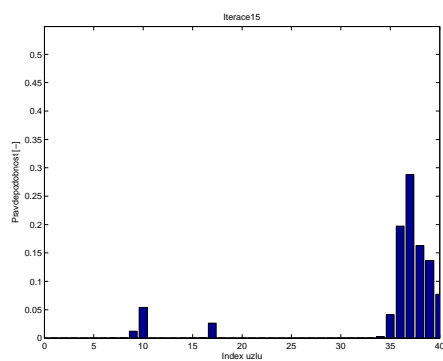
(l) Iterace 12



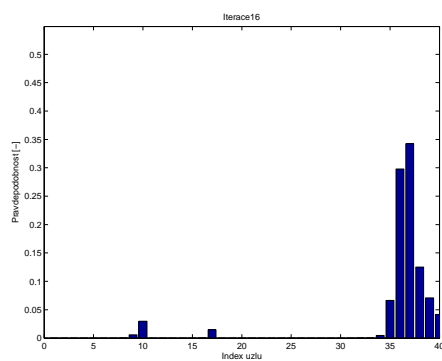
(m) Iterace 13



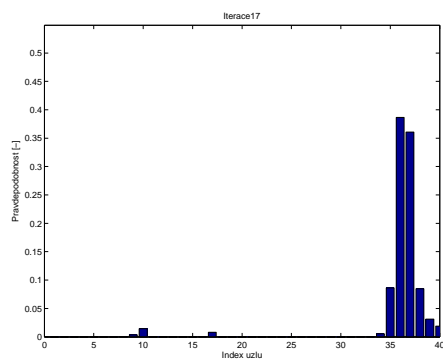
(n) Iterace 14



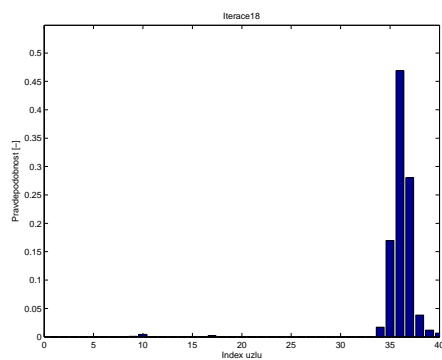
(o) Iterace 15



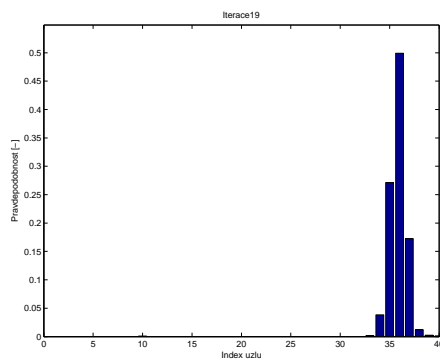
(p) Iterace 16



(q) Iterace 17



(r) Iterace 18

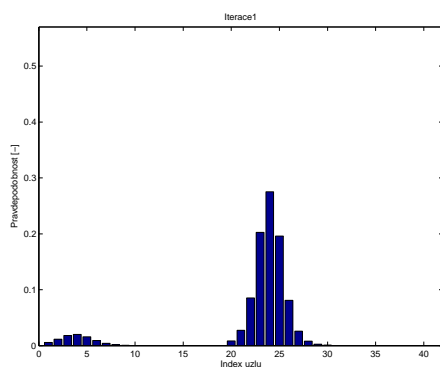


(s) Iterace 19

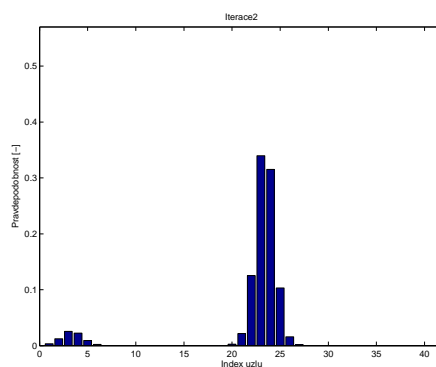
Obr. 22. Rozložení pravděpodobností při lokalizace na mřížce, příklad 3

B.4. Vliv velké nepřesnosti GPS

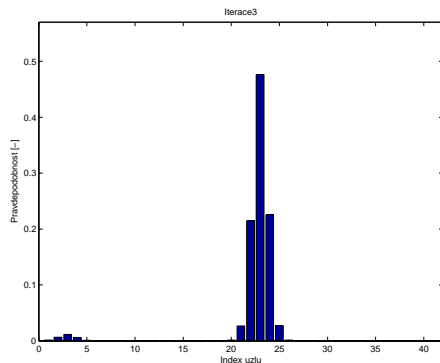
Průběhy pravděpodobnosti buněk mřížky z pokusu uvedeného v kapitole 7.2.3 jsou na obrázcích 23a až 23s. Možné hrany byly 17-137 (indexy v grafech 1 až 19) a 18-8 (indexy od 20 do 42).



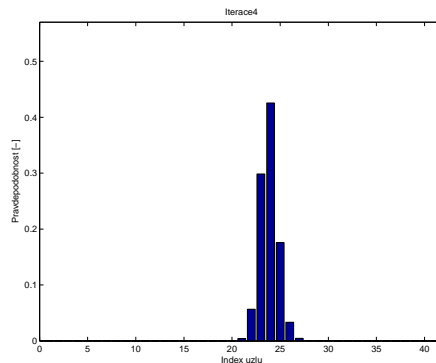
(a) Iterace 1



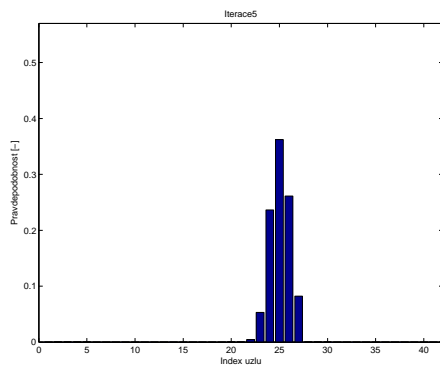
(b) Iterace 2



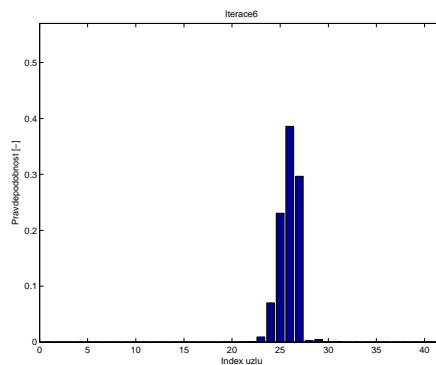
(c) Iterace 3



(d) Iterace 4

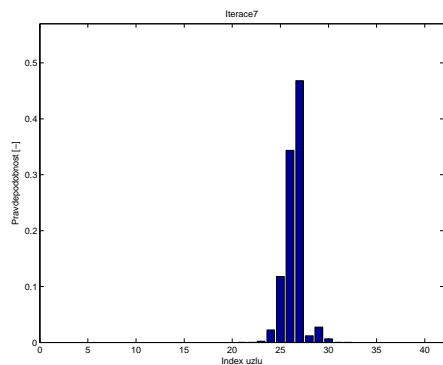


(e) Iterace 5

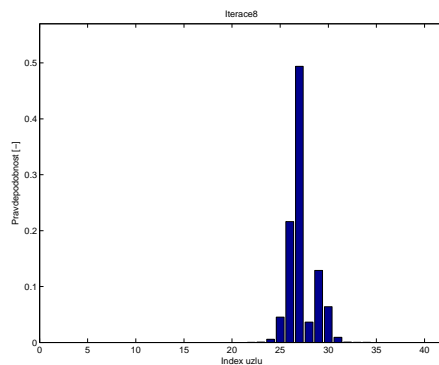


(f) Iterace 6

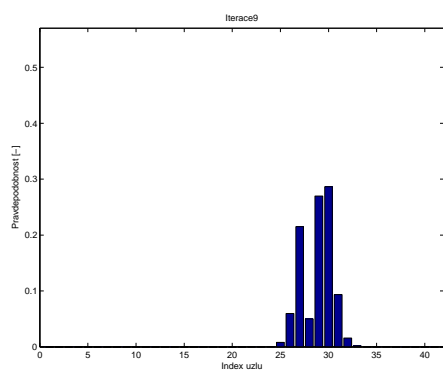
Příloha B. Experimentální výsledky lokalizace na mřížce



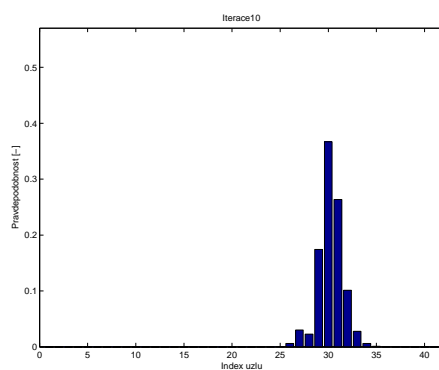
(g) Iterace 7



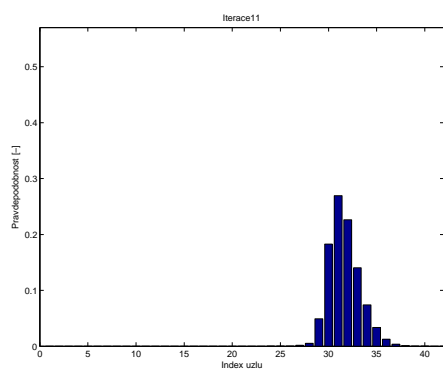
(h) Iterace 8



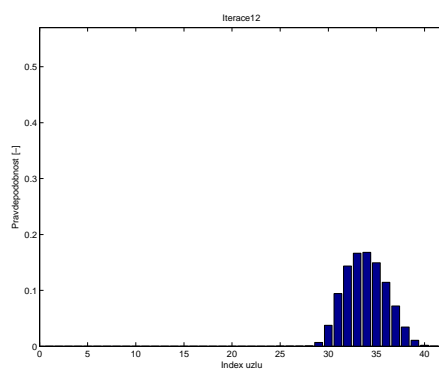
(i) Iterace 9



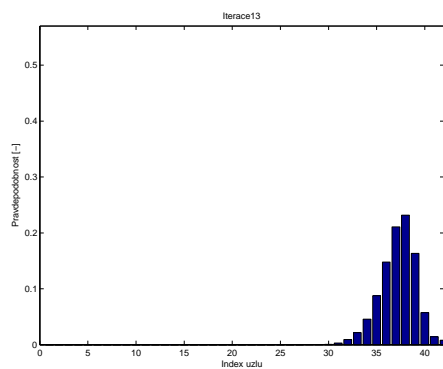
(j) Iterace 10



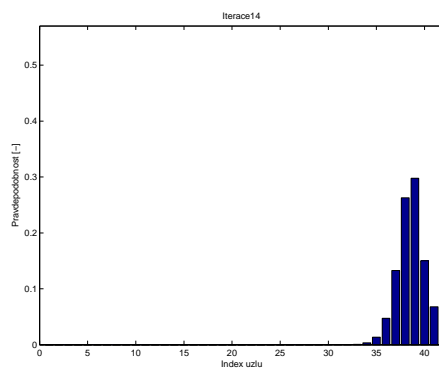
(k) Iterace 11



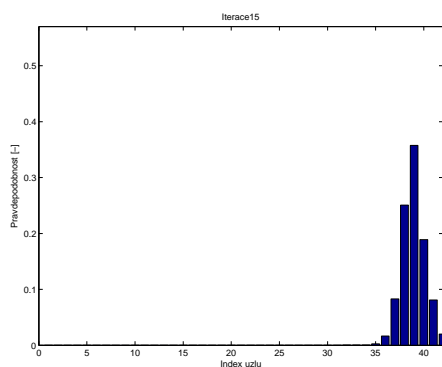
(l) Iterace 12



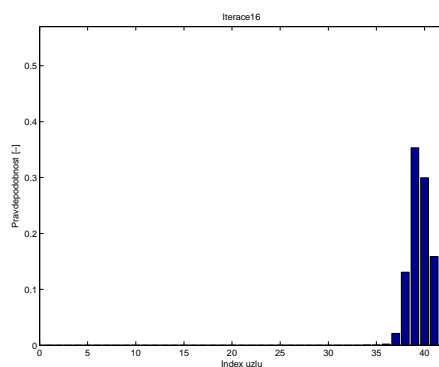
(m) Iterace 13



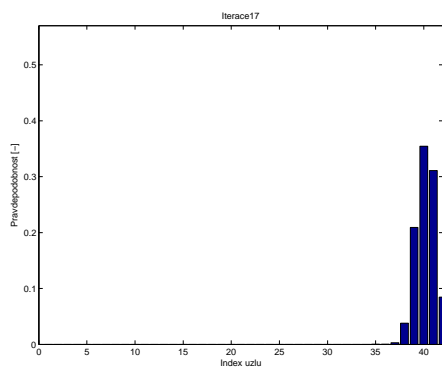
(n) Iterace 14



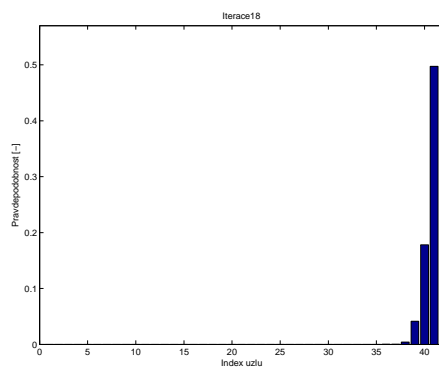
(o) Iterace 15



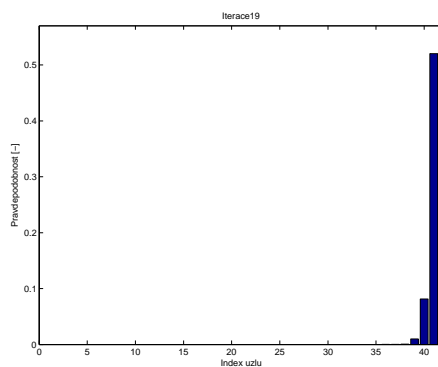
(p) Iterace 16



(q) Iterace 17



(r) Iterace 18



(s) Iterace 19

Obr. 23. Rozložení pravděpodobností při lokalizace na mřížce, příklad 2 - chybné určení polohy