

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

LOKÁLNÍ NAVIGACE AUTONOMNÍHO MOBILNÍHO ROBOTA

DIPLOMOVÁ PRÁCE

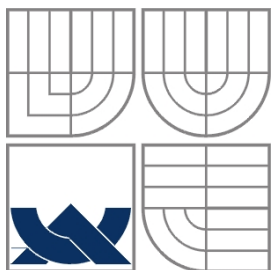
MASTER'S THESIS

AUTOR PRÁCE

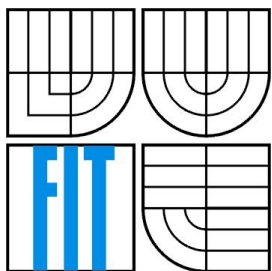
AUTHOR

Bc. DAVID HERMAN

BRNO 2010



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

LOKÁLNÍ NAVIGACE AUTONOMNÍHO MOBILNÍHO ROBOTA

LOCAL NAVIGATION OF AN AUTONOMOUS MOBILE ROBOT

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. David Herman

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. Filip Orság, Ph.D.

BRNO 2010

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav inteligentních systémů

Akademický rok 2009/2010

Zadání diplomové práce

Řešitel: **Herman David, Bc.**

Obor: Inteligentní systémy

Téma: **Lokální navigace autonomního mobilního robota**
Local Navigation of an Autonomous Mobile Robot

Kategorie: Zpracování obrazu

Pokyny:

1. Prostudujte metody zpracování obrazových dat a lokální navigace robota pomocí optického snímače a dálkoměru.
2. Vytvořte matematický model pro agregaci dat ze senzorů a navrhnete vlastní algoritmus pro lokální navigaci. Důležitá je detekce cesty a překážek, robustnost a aplikovatelnost výsledného řešení v praxi.
3. Navržené řešení implementujte, otestujte na reálných datech.
4. Otestujte funkčnost navigace v reálných podmínkách definovaných pravidly soutěže Robotour.
5. Shrňte a zhodnoťte dosažené výsledky. Posudte, zda je výsledné řešení dostačující pro autonomní pohyb mobilního robota v daném terénu.

Literatura:

- M. Sonka, V. Hlavac, R. Boyle: *Image Processing, Analysis, and Machine Vision*. Cengage-Engineering, 2007.
- Bradski, G., Kaehler, A.: *Learning OpenCV: Computer Vision with the OpenCV Library*. 1st edn. O'Reilly Media, Inc., 2008.
- Gonzalez, R. C., Woods, R. E.: *Digital Image Processing*. 2nd Edition, Prentice Hall, 2002.

Při obhajobě semestrální části diplomového projektu je požadováno:

- Body 1 a 2 zadání.

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci ročníkového a semestrálního projektu (30 až 40% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Orság Filip, Ing., Ph.D.,** UITS FIT VUT

Datum zadání: 21. září 2009

Datum odevzdání: 26. května 2010

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav inteligentních systémů
612 66 Brno, Božetěchova 2

doc. Dr. Ing. Petr Hanáček
vedoucí ústavu

Abstrakt

Cílem této diplomové práce je navrhnout koncept pro navigaci autonomního mobilního robota v parkovém prostředí. Konkrétně navrhnout metody lokalizace a detekce cesty s využitím dostupného senzorického systému, uvést matematický model pro fúzi těchto údajů a zvolenou vnitřní reprezentaci, a to s ohledem na budoucí plánovací a řídicí algoritmy.

Abstract

This paper deals with the topic of design of a navigation system for an autonomous mobile robot in a park-like environment. Precisely, designing methods for road detection using available sensoric system, designing a mathematical model for fusion of these data, and suggesting a representation of an environment suitable for planning and local navigation.

Klíčová slova

klasifikace terénu, detekce cesty z obrazu, detekce cesty pomocí laserového dálkoměru, řídicí architektura robota, metody lokalizace, Kalmanova filtrace, Unscented Kalman filtr, mřížka obsazenosti, monokulární obrazový systém, laserový dálkoměr, fúze senzorických měření

Keywords

terrain classification, road detection, autonomous navigation, real-time mobile robot control, computer vision, sensor fusion, monocular vision, lidar, Kalman filtration, Unscented Kalman filter, grid-based fusion, localization

Citace

Herman, D. Lokální navigace autonomního mobilního robota, diplomová práce, Brno, FIT VUT v Brně, 2010.

Lokální navigace autonomního mobilního robota

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Filipa Orsága, Ph.D. a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

.....
David Herman
1. května 2010

Poděkování

Děkuji vedoucímu diplomové práce Ing. Filipu Orságovi, Ph.D. za příkladnou metodickou, pedagogickou a odbornou pomoc a za další cenné rady při zpracování mé diplomové práce. Mé díky patří rovněž Ing. Tomáši Ondráčkovi, Ph.D. za konstruktivní připomínky a nápady, Ing. Janu Najvárkovi, Ph.D. za svěží myšlenky a vůbec celému týmu roboauto za to, že nám to více jezdí, než-li stojí.

© David Herman, 2010

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah.....	1
1 Úvod.....	2
1.1 Historie autonomních vozidel.....	3
1.2 Robotické soutěže.....	4
1.3 Cíle práce.....	7
1.4 Přehled relevantních prací.....	7
1.5 Přehled kapitol.....	8
2 Experimentální platforma.....	10
2.1 Senzorický systém.....	10
2.2 Mechanická konstrukce.....	14
2.3 Souřadné systémy.....	15
2.4 Kinematický model.....	15
2.5 Shrnutí.....	17
3 Navigační architektura.....	18
3.1 Typy architektur.....	18
3.2 Lokalizace.....	20
3.3 Reprezentace světa.....	23
3.4 Konceptuální model.....	26
3.5 Shrnutí.....	27
4 Detekce cesty.....	28
4.1 Detekce cesty laserovým dálkoměrem.....	28
4.2 Detekce cesty pomocí jedné kamery.....	32
4.3 Shrnutí.....	35
5 Lokální navigace.....	36
5.1 Vnitřní reprezentace prostředí.....	36
5.2 Estimace pozice.....	38
5.3 Shrnutí.....	46
6 Globální navigace.....	47
6.1 Vnitřní reprezentace prostředí.....	47
6.2 Estimace pozice.....	48
6.3 Shrnutí.....	54
7 Programová realizace.....	55
7.1 Implementace.....	55
7.2 Uživatelské rozhraní.....	59
7.3 Formát záznamu.....	60
7.4 Shrnutí.....	61
8 Závěr.....	62
Literatura.....	64
Seznam příloh.....	69
Příloha A Softwarová architektura.....	70
Příloha B Metoda plánování pohybu.....	71
Příloha C Diplom za 2. místo v Robotour 09.....	72

1 Úvod

Robotika je moderní multidisciplinární obor, který zasahuje do mnoha oblastí metod řízení, plánování, rozhodování nebo rozpoznávání. Ruku v ruce s rozvojem mechatroniky, mechaniky, elektroniky, umělé inteligence, počítačového vidění a dalších, zaznamenává i obor robotiky v posledních letech dynamický rozvoj. Výzkum v robotice se soustřeďuje především na základní otázky inteligence, respektive na zvýšení samostatnosti robotů. Hlavní obtíží těchto výzkumů je právě obrovská složitost kognitivních procesů a komplexnost prostředí, v nichž se roboti pohybují.

S autonomními mobilními roboty, tedy roboty, jež jsou schopni se pomocí vlastních prostředků pohybovat a tento pohyb plánovat, se setkáváme čím dál častěji. Již dnes vidíme aplikace autonomních robotů v mnoha různých situacích, například ve vojenských či záchranných misích. Mobilní roboti se také uplatňují jako podpůrné systémy v logistice atp., avšak stále se jedná o jednoduché umělé systémy se zárodky komplexního kognitivního řízení.

V posledních letech je v civilní sféře velice populární výzkum autonomních mobilních systémů sloužících pro transport osob nebo zboží, tzv. bezpilotních vozidel. V minulosti byl bezpilotní automobil v očích laické veřejnosti považován za science fiction, avšak vývoj od roku 1994, kdy poprvé vozidlo VaMP urazilo více než 1 000 km na dálniční komunikaci s minimálním množstvím lidské intervence do řízení, naznačuje, že se bude jednat o jeden z nejlukrativnějších technických problémů během příštích dvaceti let.

Úspěšné dokončení vývoje těchto systémů bude zcela jistě znamenat revoluci v dopravní sféře. Autonomní vozidla nejsou pouze o bezpečnější a pohodlnější dopravě, ale o zcela nových možnostech v budování dopravní infrastruktury jednak na lokální (městské), jednak na globální úrovni (meziměstské). Již dnes existuje koncepce hromadné dopravy, jež je efektivnější obdobou taxi služby v ekonomických i ekologických aspektech (více viz [1]).

Avšak v popředí budoucí automobilové revoluce nelze vidět jen dopravní hlediska. Autonomní vozidla mohou být průlomové i v ekologických otázkách. Bepilotní systémy mají potenciál zásadně urychlit snížení závislosti na fosilních palivech a přispět tak významnou mírou k redukci znečištění životního prostředí, a to ať už v podobě rozšíření elektromobilů, nebo několikanásobným zvýšením efektivity transportu.

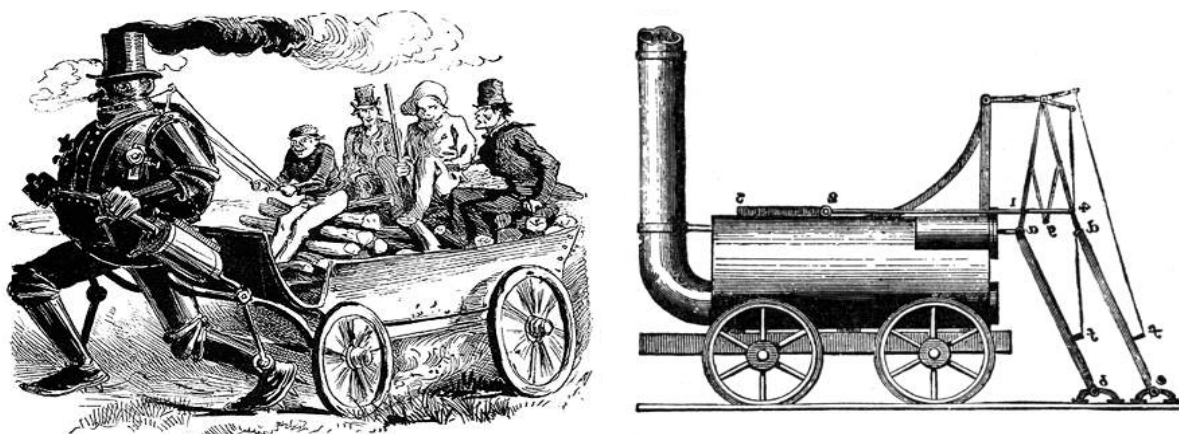
Obdobné systémy lze snadno uplatnit i v plně automatizovaném zemědělství, lodní přepravě, letectví atd., kde už jsou značné pokroky. Nicméně ať bude konkrétní aplikace mobilního robota jakákoliv, existuje alespoň jeden společný problém všech těchto strojů a tím je navigace.

Navigace autonomních mobilních robotů se koncentruje do schopnosti přesunu robota z aktuální pozice na pozici cílovou. Přesto se pod tímto procesem skrývá velice složitá sekvence podúloh, počínaje sběrem dat ze senzorů, tvorbou modelu okolního prostředí, lokalizací, plánováním a konče samotnou realizací ustanoveného plánu. Nadto je třeba, aby robot uměl řešit nečekané situace, kolize s překážkami apod., neboť reálné prostředí, v němž se robot pohybuje, představuje vysoce dynamický systém.

1.1 Historie autonomních vozidel

Průmyslová revoluce 18. století započatá vynálezem parního stroje vedla ke zrodu společnosti, v níž lidská práce mohla být s mnohem větší produktivitou vykonávána stroji. Industrializace se stala

externalizací manuálních funkcí člověka a technologická úroveň již dovoľovala realizaci prvních motorických strojů sloužících pro transport. Například William Brunton v roce 1813 navrhl konstrukci mechanického čtyřkolového stroje, který byl poháněn složitým systémem nohou, neboť se tehdy věřilo, že adheze otáčejících se kol vůči kolejnici nebude dostatečně velká. Stroj se tak odstrkoval při pohybu vpřed, podobně jako lyžař [2]. Raketový nástup železnice pak podobné bizarní konstrukce velice rychle odsunul na pokraj zájmu tehdejší vědecké společnosti.



Obrázek 1: Vlevo představa bezpilotního vozidla jakožto vlečky tažené parním automatem v roce 1893. Vpravo dopravní prostředek navržený Williamem Bruntonem roku 1813. Převzato z [2].

Historie autonomních vozidel, tak jak je chápeme dnes, se však datuje až od roku 1977, kdy se v japonské laboratoři Tsukuba podařilo při rychlosti 30km/h trasovat bílé značky pomocí speciálně navrženého hardwaru [3, 4].

V roce 1980 robotické vozidlo Mercedes-Benz, vyvinuté Ernestem Dickmannsem a jeho týmem z univerzity v Mnichově, řízené kamerovým systémem, dosáhlo rychlosti 100 km/h na ulici bez silničního provozu. Po tomto úspěchu následně Evropská komise zahájila financování projektu o automobilních vozidlech - Euro EUREKA Prometheus (1987 – 1995). Rovněž v roce 1980 ve Spojených státech amerických nadace DARPA-AVL (Autonomous Land Vehicle) poprvé úspěšně trasovala cestu s využitím laserového dálkoměru a počítačového vidění.

Psal se rok 1987, když HRL Laboratories demonstrovaly první autonomní navigaci použitelnou mimo silniční síť. Vozidlo ujelo 600 metrů při rychlosti 3 km/h po náročném terénu s příkrými svahy, velkými kameny a vegetací.

Roku 1994 vozidla VaMP a Vita-2 urazila více než 1 000 kilometrů na dálničních cestách při rychlosti 130 km/h s občasnými lidskými zásahy. Demonstrována byla autonomnost řízení v jízdním pruhu, změna jízdního pruhu při předjíždění ostatních vozidel a jízda v koloně.

V roce 1995 Ernestem Dickmanns vykonal cestu s bezpilotním vozidlem Mercedes-Benz dlouhou 1 600 km z Mnichova do Kodaně. Auto bylo ovládáno pomocí počítačového vidění a dosáhlo nejvyšší rychlosti 175 km/h na německé dálnici. Střední délka tratě mezi lidskými zásahy činila 9 km. Téhož roku se na univerzitě Carnegie Mellon podařilo dosáhnout z 98,2 % bezpilotní jízdy, avšak testovaný vůz byl jen z části řízen autonomně. Umělé neuronové sítě ovládaly volant vozidla, plyn a brzdy byly řízeny člověkem.

Alberto Broggia z univerzity Parma odstartoval v r. 1996 projekt ARGO, jehož výsledkem bylo upravené auto Lancia Thema schopné trasovat bílé silniční pruhy na dálničních komunikacích.

Vyvrcholením projektu se pak stala šestidenní cesta, při níž se ujelo 2 000 km na dálnicích v severní Itálii. Průměrná rychlost vozidla činila 90 km/h a 90 % jízdního času bylo auto řízeno bez lidské intervence. Senzorický systém robota se skládal z levných černobílých videokamer použitých pro stereoskopické vidění.

V roce 2002 byly vyhlášeny závody DARPA Grand Challenge. V prvních dvou ročnících musela autonomní vozidla soutěžit na nerovném a nebezpečném terénu v neobydlených příměstských oblastech. Třetí soutěž se orientovala na městské prostředí.

Roku 2008 General Motors uvedl, že se začne s testováním bezpilotních vozidel do roku 2015 s cílem tato vozidla dostat na silniční síť již od roku 2018.

1.2 Robotické soutěže

Dnes existuje bezpočet soutěží téměř ve všech kategoriích autonomních robotických systémů, jež přitahují pozornost jak laické, tak i odborné veřejnosti a jsou proto ideální příležitostí k propagaci a k výměně praktických zkušeností. Hnacím motorem těchto soutěží není zpravidla samotná výherní cena, ale možnost změřit své síly s jinými týmy, seznámit se s lidmi, kteří mají podobné zájmy a konečně také v případě úspěchu vejít do povědomí této komunity a získat další finanční i lidské zdroje.

Ačkoliv se někdy soutěž v očích diváků může jevit jako pouhé hraní, vytvoření robota schopného účastnit se soutěže, byť jen amatérské kategorie, vyžaduje rozsáhlé teoretické znalosti v oblasti elektrotechniky, mechatroniky, mechaniky a informatiky. Klade na soutěžní skupinu nemalé požadavky, např. na schopnost práce v týmu, v organizačních a komunikačních dovednostech. Vedle toho je třeba zmínit finanční nároky na stavbu hardwarové platformy. Profesionální podvozky pro malé roboty stojí desetitisíce korun. Stejně tak ani senzorický systém robota není levnou záležitostí a snadno se i u malých robotů vyšplhá na částky převyšující možnosti nejen jednotlivců, ale též větších týmů. Proto bývá finanční a technické zázemí často prvotním faktorem implikující možný úspěch.

Soutěže vytvářejí umělé konkurenční prostředí a snadno se stávají motivací k usilovnějšímu vývoji a větší snaze. Mezi nejznámější soutěže autonomních mobilních vozidel patří americká soutěž DARPA Grand Challenge. V Evropě se od roku 2006 koná každoročně soutěž bez vítěze, tedy spíše předváděcí akce civilní a armádní robotiky, ELROB. V České republice je velice populární soutěž Robotour, která se v roce 2009 stala soutěží mezinárodní.

1.2.1 DARPA Grand Challenge

DARPA Grand Challenge [5] je závod pořádaný od roku 2004 americkou vojenskou organizací DARPA „Defense Advanced Research Projects Agency“. Oficiálním cílem soutěže je podpora technické inovace pozemních bojových vozidel, proto se vozidla musejí umět pohybovat nejen po existující silniční síti, ale také po volném prostranství. Dále musí být schopna autonomní navigace z jednoho místa na druhé, a to i v rozsáhlých otevřených oblastech, kde není k dispozici dostatečně přesná mapa, zabraňovat kolizím, dynamicky přehodnotit naplánovanou trasu, pakliže je blokována atd. Hodnocena je rovněž rychlost zvládnutí úkolu. K tomu jsou požadavky kladeny též na mechanickou konstrukci robotů. Maximální váha robota je 30 000 liber, horizontální vzdálenost mezi středem předního a zadního kola musí být nejméně 72 palců a vozidlo by nemělo být větší než 9

stop na šířku a 12 stop na výšku (více viz [6]). Díky extrémní náročnosti a komplexnosti soutěže se jí proto účastní týmy disponující špičkovou technikou a zpravidla i s vysokoškolským zázemím. Odměnou vítězi je suma 1 mil. USD (r. 2004), respektive 2 mil. USD (r. 2005 a r. 2007).

Dvě hodiny před startem roboti dostanou CD-ROM s mapou v digitální podobě ve formátu RNDP (*Route Network Definition File*, [7]) a soubor s cíli ve formátu MDF (*Mission Data File*). Trasa je definována seznamem GPS souřadnic (tzv. *waypoints*) s poloměrem definujícím kruh, kudy robot musí projet. Zároveň je omezena šířka koridoru mezi dvěma *waypoints*. Nadto každý z těchto cílových uzlů má časový limit, do něhož jej robot musí navštívit. Robotické vozidlo, které opustí stanovený koridor nebo nedodrží časový limit, ze soutěže automaticky vypadává.

I přes tyto tvrdé podmínky zaznamenala soutěž již v r. 2005 prvního vítěze. Byl jím robot Stanley od Stanford Racing týmu tvořeného studenty, doktorandy a profesory Stanfordovy univerzity, spolu se zaměstnanci VW ERL „Volkswagen Electronics Research Laboratory“, kterému se podařilo projet trať dlouhou 132 milí v časovém limitu deseti hodin. Úspěšné v roce 2005 byly rovněž další čtyři týmy (Sandstorm, Hlglander, TerraMax a Kat-5). Roku 2007 se stal vítězem Tartan Racing (Carnegie Mellon University, General Motors Corporation, Caterpillar, Continental a další) s robotem Boss. Na druhém místě se opět umístil tým Stanford Racing.

Zajímavé je srovnání robotických vozidel soutěžících v tomto závodě s projektem VaMP z roku 1990, který byl částečně financován z 1 miliardy dolarů na projekt EUREKA Prometheus [4, 8]. VaMP měl být schopný autonomní činnosti v městském prostředí, měl se umět vypořádat s pohyblivými překážkami, být schopen předjíždět pomalejší vozidla atp. Maximální rychlost VaMP činila 180 km/h. Vozidla DARPA jsou limitována 50 km/h. V roce 1995 VaMP dokázal překonat bez lidské intervence 158 km na dánské dálnici (DARPA 212 km v r. 2005). Vozidlo VaMP se pohybovalo převážně na přímých dálničních komunikacích, kdežto v soutěži DARPA se jezdí po rozmanitých stezkách, včetně příkrých horských tras. Detekce sjízdného povrchu VaMP je postavena na kamerovém systému a radaru. V lokalizaci vozidla není uplatněna GPS. Naopak vozidla pro DARPA závod jsou silně závislá na globálním navigačním systému už jen díky formátu zadané trasy. Senzorický systém těchto vozidel typicky zahrnuje několik laserových dálkoměrů, barevný kamerový systém, radary a interní navigační systém postavený na akcelerometrech, gyroskopech, odometrii atd. Výpočetní výkon VaMP počítače je tisíckrát menší na dolar než u vozidel účastnících se DARPA závodu.



Obrázek 2: DARPA Grand Challenge. Zleva vítězné robotické vozidlo r. 2005, robot Stanley, vpravo robotické vozidlo HIGLANDER, které rovněž dorazilo přes technické potíže do cíle. Převzato z [5].

Možná právě díky počátečnímu debaklu v očích diváků v roce 2004, kdy cena uspořádání soutěže několikanásobně převyšovala výhru jednoho miliónu dolarů a finanční náročnost dílčích projektů na přípravu vozidel dovolující účast rovněž, se tato soutěž stala výbornou motivací a velkou výzvou pro perfektně financované týmy disponující nejmodernější technikou a špičkovým technickým zázemím. V současnosti tento závod představuje nejprestižnější soutěž v oblasti autonomních, roboticky řízených vozidel s obrovskou mediální podporou a peněžními zdroji.

1.2.2 ELROB

ELROB (The European Robotic Trial) je více přehlídkou robotů, než-li soutěží. Pravděpodobně se jednalo o reakci Evropy na americkou soutěž DARPA Grand Challenge, avšak filosofie ELROBu staví na jiných základech. ELROB si klade za cíl být spíše předváděcí akcí, jež umožní srovnání a výměnu zkušeností v oblasti mobilní robotiky. Je orientován na porovnání bezpilotních systémů v reálném prostředí, které nemusí být nutně pozemní. Dovoluje posouzení současných technologií a strategií uplatněných při řešení dílčích problémů, soustředí se na krátkodobě realizovatelné robotické systémy a umožňuje navázání kontaktů mezi výzkumnými pracovníky a průmyslem.

Roboti se prezentují ve dvou kategoriích, a to armádní (M-ELROB) a civilní (C-ELROB), jež se každý rok střídají. První akce se konala v Německu roku 2006 a byla zaměřena na vojenské autonomní mobilní systémy. Snahou ELROBu je rovněž vytvořit katalog evropských týmů zabývajících se mobilní robotikou. K dnešnímu dni tento katalog obsahuje 30 týmů převážně z Německa a Finska (více viz [9]).

1.2.3 Robotour

Robotour outdoor delivery challenge je původem česká soutěž autonomních robotů orientovaných do parkových prostředí. Soutěž se koná každoročně na podzim, a to již od roku 2006. Každý ročník narůstá obtížnost soutěže nejen v podobě ztěžujících se pravidel, ale i v konkurenci. Od roku 2009 se soutěž stala mezinárodní díky účasti italských a slovenských týmů.

Koncept soutěže vychází z myšlenky postavit robotické vozidlo pro přepravu osob nebo zboží schopné jízdy v reálném silničním provozu. Každým rokem se doplňují pravidla a kladou se větší nároky na samostatnost robotů, a to tak, aby restrikce co nejvíce postihovaly skutečné scénáře a aplikovatelnost řešení v praxi. Například roku 2009 nebyla dopředu známa trasa, kudy roboti pojedou. Robotům byla trasa zadána minutu před startem jako sekvence segmentů v RNDF mapě (podobně jako v DARPA Grand Challenge). Roboti se museli umět vypořádat jednak se statickými překážkami, jednak s pohybujícími se objekty. Byla kladena omezení na mechanickou konstrukci robota, a to definováním minimální velikosti a hmotnosti nákladu, který robot musí zvládnout do cíle přepravit. Rovněž i doba na splnění úkolu byla ohraničena časovým limitem.

Příští soutěž (r. 2010, Bratislava) již není povoleno předchozí mapování prostředí pomocí vlastních systémů. Roboti budou startovat pro zvětšení divácké atraktivity z jednoho místa a soutěž tak dostane nádech velice populárních motorických závodů. Na další roky se plánuje realizace tzv. autonomní robotí taxi služby, kde úlohou robotů nebude pouze za co nejkratší čas projet určitou trasu, ale taktéž přemístit „zákazníka“ dle jeho vlastních požadavků (opět možnost interakce s diváky). To zahrnuje soutěžení robotů o lukrativní „zákazníky“, a tedy výrazně stupňující se požadavky na kognitivní i senzorický systém robota (více viz [10]).

Právě stupňující požadavky znemožňují začlenění nových soutěžních týmů z řad amatérů a začínajících robotiků, neboť i zde platí, že nejen um soutěžících, ale i finanční a technické zázemí jsou nutnou podmínkou k vítězství. Robotour se tak ruku v ruce s časem stává elitní soutěží pro týmy z řad vysokoškolských institucí, případně soukromých firem, které mají dostatek finančních prostředků a času pro realizaci konkurenceschopné hardwarové platformy a kognitivního systému robota. Z toho důvodu vznikají nové „odnože“ této soutěže, jež svými nároky dovolí začínajícím týmům a nadšencům změřit si své síly a vyměnit vlastní zkušenosti s konstrukcí byť jen jednodušších systémů bez pokročilých lokalizačních a řídicích algoritmů (např. Robotem rovně aneb autíčka v parku [11]).



Obrázek 3: Robotour 2009. Zleva vítězný robot LEE (ČVUT), robotický vláček Loko s doprovodem (radioklub Písek, 3. místo) a robot Karlík (Roboauto, 2. místo). Převzato z [10].

1.3 Cíle práce

Cílem této diplomové práce je navrhnout koncept pro navigaci mobilního robota, konkrétně robotického čtyřkolového modelu vozidla jménem *Karlík* (projekt *Roboauto*, více viz [12]), v parkovém prostředí s důrazem na použitelnost řešení v reálných podmínkách definovaných pravidly soutěže Robotour [10]. Jmenovitě:

- navrhnout metody detekce cesty z dat laserového dálkoměru a jedné barevné kamery v parkovém prostředí a sestavit matematické modely těchto senzorů umožňující agregaci výsledků detekčních algoritmů,
- navrhnout metodu lokální lokalizace robota z dostupných senzorů pro interní navigaci (rotační enkodér, kompas a inerciální jednotka),
- nalézt vhodnou vnitřní reprezentaci okolního prostředí pro budoucí plánovací a řídicí algoritmy robota, která by reflektovala požadavek na agregaci senzorických vjemů,
- vybudovat algoritmus pro globální lokalizaci robota s využitím GPS senzoru, mapového podkladu ve formátu RNDF a lokálního pozičního estimátoru,
- ověřit použitelnost všech uvedených algoritmů v praxi na reálné experimentální platformě a diskutovat další možnosti zlepšení, případně směr budoucího vývoje.

1.4 Přehled relevantních prací

Tato podkapitola obsahuje přehled relevantních prací, které svým rozsahem řešené problematiky pokrývají nebo přesahují cíle této práce. Dále se v textu nacházejí odkazy na po většinu vědecké články, jež řeší dílčí úlohy, doplňují informace k problematice či nabízejí jiné přístupy.

Disertační práce M. A. S. Vázqueze - *Vision-based Road Transport-system for Unmanned Operation on Urban-like Scenarios* [13] popisuje úplný koncept globální navigace autonomních automobilů v částečně známém prostředí, jako jsou průmyslové oblasti apod. Rozpoznávání cesty je postaveno pouze na obrazové informaci s využitím HSV barevného modelu a aproximaci sjízdného povrchu polynomem druhého řádu. V práci je rovněž řešen problém stínů, detekce jiných vozidel, řízení apod.

Navigaci autonomních robotů pomocí vidění a laserového dálkoměru se zabývá v celé své komplexnosti rovněž disertační práce J. Ch. Andersena – *Mobile Robot Navigation* [14]. K segmentaci cesty z obrazu využívá kombinace detekce rohů a barevné analýzy. Lidarová data jsou vyhodnocována na základě několika spočtených příznaků v rámci jednoho skenu.

Diplomová práce Ch. R. Greca – *Real-time forward urban environment perception for an autonomous ground vehicle using computer vision and Lidar* [15] se zabývá, podobně jako [14], analýzou obrazových a lidarových dat, avšak data z laserového dálkoměru jsou aplikována pouze při detekci překážek. Rozpoznávání cesty je situováno do městského prostředí a uplatňuje apriorní znalosti o ohraničení a šířce cesty.

Kalmanův filtr je stochastický estimátor stavu a patří mezi klasické případy využití metody adaptivní filtrace. Nejčastěji se tato technika uplatňuje při integraci dat z několika informačních zdrojů pro dosažení lepšího odhadu aktuálního stavu systému. Diplomová práce *Robot Localization and Kalman Filters* [16] od Rudyho Negenborna shrnuje rozsáhlou problematiku využití metod Kalmanovy filtrace v oblasti mobilní robotiky pro estimaci pozice robota. Bohužel, v jinak velice komplexní práci, není zahrnut poslední trend této oblasti, a to *Unscented Kalman Filter*.

Hodnotná je rovněž disertační práce Petra Štěpána – *Vnitřní reprezentace prostředí pro autonomní mobilní roboty* [17], jež se zabývá výzkumem metod souvisejících se získáváním znalostí o okolním prostředí, jejich sdružováním do vnitřní reprezentace a způsoby generování plánů činnosti robota. Součástí práce je i srovnání matematických modelů sonarů, laserových hloubkoměrů a tvorba mřížky obsazenosti pomocí barevné kamery.

1.5 Přehled kapitol

Druhá kapitola popisuje experimentální platformu *Karlik*, na které jsou testovány algoritmy řízení, metody detekce sjízdného povrchu atd. Rozvedena je mechanická konstrukce robota, jeho kinematický model a senzorický subsystém.

John J. Leonard a Hugh F. Durrant-Whyte roku 1990 sumarizovali hlavní problémy navigace mobilních robotů do tří otázek: „Kde jsem?“, „Kam se chci dostat?“ a „Jak toho dosáhnu?“. Třetí kapitola nazvaná *Navigační architektura* se přehledově věnuje klíčovým prvkům kognitivního systému řešící tyto otázky. Jmenovitě řídicí architekturu robota, možnostmi lokalizace a vnitřní reprezentaci okolního světa (zejména v kontextu bezpilotních vozidel).

Kapitola *Detekce cesty* se zabývá detekcí sjízdného povrchu z laserového dálkoměru a barevné monokulární kamery. V úvodu jsou uvedeny obecné požadavky na detekční algoritmy a následně jsou formálně popsány vlastní metody detekce sjízdného povrchu pro každý senzor samostatně.

Navigace tvoří souhrnný pojem zastřešující postupy lokalizace a plánování cesty. Pátá kapitola je zasvěcena podkategorii tohoto pojmu, konkrétně lokální navigaci. Popisuje vnitřní reprezentaci okolního prostředí založenou na věrohodnostní mřížce obsazenosti. Základním předpokladem pro tvorbu této mapy je mít pevný souřadný systém robota a možnost jeho lokalizace v tomto

souřadném systému. K určení aktuální lokální polohy robota se uplatňuje interní lokalizační systém založený na Kalmanově filtraci.

Šestá kapitola pojednává o globální navigaci robota, respektive jeho lokalizaci v globální mapě ve formátu RNDF s využitím GPS senzoru, mapových porovnávacích algoritmů a lokální poziční estimace. GPS informace je do aktuální polohy agregována pomocí fuzzy logiky, výsledek agregace se následně zpětně konfrontuje s dodanou mapou.

Předposlední kapitola se zabývá velmi stručně programovou realizací. Implementováno bylo jádro deklarující *thread-safe* komunikaci mezi objekty pro zajištění snadného paralelního běhu jednotlivých modulů detekce, lokalizace atd. Dále byla realizována mezivrstva mezi senzory a detektory zaručující časovou konzistenci zpracovaných dat. Samozřejmostí jsou vizualizační nástroje, komunikační prostředky obstarávající spojení skrze TCP/IP protokol s dalšími moduly robota aj.

Poslední kapitola sumarizuje představené řešení a prezentuje úspěchy, kterých bylo s vybudovanou navigační architekturou dosaženo na robotických soutěžích.

2 Experimentální platforma

Když roku 1920 Josef Čapek odpovídal Karlu Čapkovi na otázku, jak pojmenovat umělé dělníky v nově vznikající hře R.U.R., vycházel při své odpovědi z něčeho hmotného, mechanického, co dokáže zasahovat do okolního prostředí a být tímto prostředím zpětně ovlivňováno (více o původu slova robot v [18]). Robot je tedy stroj, má fyzickou realizaci, která mu interakci s okolním světem dovoluje.

Hardware autonomních mobilních robotů se dotýká široké oblasti vědních disciplín počínaje mechanikou, mechatronikou, automatizací, řízením, senzorickými systémy atp. Důležitým aspektem při návrhu fyzické podoby mobilního robota je charakter povrchu terénu, po kterém se robot bude pohybovat. V případě málo členitého povrchu mající rovinný charakter jsou vhodná kolová a pásová pohybová ústrojí. Naopak ve velmi členitém terénu s výškovými překážkami se uplatňují kráčejíci, nebo plazící se roboti.

Tato kapitola je stručným úvodem do problematiky nízkoúrovňového řízení mobilního robota automobilového typu ve spojitosti s experimentální platformou *Karlík* a jeho senzorickým systémem. Následující čtyři podkapitoly postupně popisují senzorický subsystém, mechanickou konstrukci, souřadné systémy a kinematický model *Karlíka*.

2.1 Senzorický systém

Jedním z nejdůležitějších úkonů všech autonomních systému je sběr informací o svém okolí a jejich vyhodnocení. Zařízení, jímž robot vnímá okolní svět, se nazývá senzor a skupina těchto zařízení senzorický systém (subsystém). Tento subsystém lze dále ještě dělit na vlastní senzory a jejich obslužná zařízení a na část realizující komunikaci, tj. obvody rozhraní a sběrnice vytvořené z různých přenosových médií.

Informace ze senzorů mohou sloužit k monitorování interakce s prostředím, k měření parametrů pro uzavírání zpětných vazeb, k lokalizaci předmětů včetně sebe sama, k tvorbě vlastního modelu okolního světa, k introspekci, k detekování překážek a chybových stavů, atp. Pro tyto účely existuje široká škála senzorů používaná v mobilní robotice, od jednoduchých čidel indikující teplotu vnitřní elektroniky, nebo stav baterie, až po sofistikované senzory poskytující obrovské množství informací o vnějším prostředí (více viz [19]).

Mezi základní parametry senzoru patří jeho citlivost, limit detekce, plný rozsah, dynamický rozsah, statická přenosová funkce, linearita, hystereze, rozlišení, reprodukovatelnost, přechodová charakteristika, frekvenční analýza, doba odezvy, spotřeba, spolehlivost atd. Volba vhodných senzorů, či jejich kombinace, ke snímání dané veličiny je netriviální úloha, při níž musí být zvaženo mnoho faktorů zohledňujících nejen vlastnosti senzorů, ale také mechanickou konstrukci robota, pracovní prostředí, možnost vzájemné interakce, výpočetní nároky, energetické nároky aj.

Jedna z hlavních kategorizací senzorů dělí senzory na pasivní a aktivní. Pasivní senzor pouze kvantizuje energii vstupující do čidla senzoru. Příkladem toho typu senzoru je mikrofón, či CCD nebo CMOS kamera. Aktivní senzor emituje záření do prostředí a měří velikost reakce prostředí na vyzářenou energii. Typickými představiteli těchto senzorů jsou laserové dálkoměry, radary a ultrazvuky.

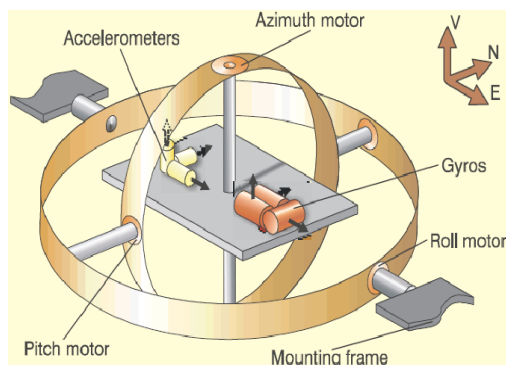
2.1.1 Rotační optický enkodér

Rotační optický enkodér je interní inkrementální senzor pro měření rychlosti otáček. Název inkrementální je vzat z principu činnosti založené na otáčivém mezikruží s pravidelně se střídajícími průhlednými a neprůhlednými ryskami, jež při rotaci přerušují emitované světlo LED diody. Změna množství světla dopadajícího na fototranzistor, který se nachází zpravidla na opačné straně mezikruží v ose LED diody, způsobuje generování kvazi-sinusového signálu, jenž je dále komparátorem převeden na obdelníkový průběh. Rozlišení směru otáčení se řeší přidáním druhého segmentu s ryskami posunutými vůči ryskům prvního segmentu. Přidanému segmentu náleží další fototranzistor snímající fázově posunutý světelný tok (kvadrurní výstup). Více viz [20].

Realizace rotačního optického enkodéru na robotu má podobu počítačové myši umístěné na samostatném kolečku pod podvozkem robota. Výhodou tohoto řešení je, že veškeré kódování a dekódování se provádí přímo v elektronice myši, která již poskytuje informaci o počtu pulzů enkodéru obou os s ohledem na znaménko. Elektronika myši komunikuje s PC pouze jako mluvčí (žádná data nepřijímá). Odebíraný proud bývá typicky do 10 mA.

2.1.2 Inerciální jednotka

Inerciální měřicí jednotka (angl. *Inertial Measurement Unit*, IMU) je elektronické zařízení pro měření úhlové rychlosti a akceleračních sil pomocí kombinace akcelerometrů a gyroskopů, respektive inerciálních senzorů [21, 22]. Jelikož gyroskopy snímají úhlovou rychlost, tak se hodnota natočení v určité ose získává integrací dle času u příslušného gyra. Původně gyroskopy pracovaly na principu zachování rotační energie, avšak většina moderních mikroelektromechanických (MEMS¹) gyr ke své činnosti využívá principu Coriolisovy síly.



Obrázek 4: Závěsový IMU [22].

skutečná motorická (pohybová) zrychlení mající vliv na trajektorii objektu² (nedochází k rozložení gravitačního zrychlení do tří složek, obr. 4).

Tímto způsobem koncipovaná inerciální jednotka má vysokou přesnost při inerciální navigaci a bývá tradičně používána v kosmických lodích, letadlech a řízených střelách. Nicméně díky komplikovanosti vnitřního uspořádání závěsů a s tím spojenými složitými kalibračními procedurami včetně časově náročného testování se od počátku sedmdesátých let vyvíjela tzv. „strapdown“ technologie. „Strapdown“ je alternativní a jednodušší uspořádání bez systému závěsů. Snímače

1 Podstatnou výhodou MEMS technologie je nízká spotřeba a menší rozměry.

2 Vyjma akcelerometru měřícího zrychlení na svislé ose (vliv gravitačního zrychlení).

zrychlení jsou pevně spojeny s konstrukcí přístroje a gyroskopy v něm nevystupují jako stabilizátory, nýbrž jako senzory pro měření otáčení tělesa v prostoru.

Nevýhody „strapdown“ technologie plynou z pohybu akcelerometrů v prostoru a s tím souvisejícím ovlivňováním jejich měření gravitačním zrychlením, což vede k vyšší chybovosti. Naopak výhodou je snazší mechanické provedení a lepší cenová dostupnost.

Robot je vybaven inerciální jednotkou od firmy Oprox stavící na čipu ADIS16362. Tento čip obsahuje tříosý gyroskop spolu s tříosým akcelerometrem. Kalibrace senzorů je provedena při výrobě (citlivost, zarovnání, práh a lineární akcelerace). Vylepšené *SPI* rozhraní spolu s registrovou architekturou čipu dovoluje rychlý sběr dat. Jedná se o kompaktní modul (23x23x23 mm) schopný pracovat při běžných venkovních teplotách.

2.1.3 Kompas

Směr vozidla patří mezi signifikantní parametry v navigaci, které mají významný vliv na akumulaci chyby. Z tohoto důvodu jsou senzory, jež poskytují absolutní měření o směru velmi důležité v otázkách navigace autonomních robotů. Magnetický kompas, jehož funkcionalita vychází z existence magnetického pole Země, se k těmto senzorům řadí.

Bohužel magnetické pole Země je deformováno elektricky nabitými částicemi ze Slunce, které proudí v horních vrstvách atmosféry a způsobují malá kolísání. Některá z nich jsou pravidelná (den/noc), některá občasná (magnetická bouře). Dalším faktorem výrazně ovlivňujícím přesnost měření je narušení magnetického pole silovým vedením, případně většími ocelovými strukturami v blízkosti robota (i robotem samotným – elektromagnetické motory). Z toho důvodu je velmi obtížné použít magnetické kompas pro navigaci v interiéru. Na základě způsobu snímání magnetického pole dělíme elektronické kompas na [23]:

- mechanické magnetické kompas,
- fluxgate kompas,
- hall-effect kompas,
- magnetorezistivní kompas,
- magnetoindukční kompas,
- magnetoelastické kompas.

Senzorický systém robota Karlík zahrnuje kompas CMPS 03 speciálně navržený k navigačním úkolům v oblasti robotiky, jenž obsahuje senzor magnetického pole Philips KMZ51. Informace o azimutu je poskytována dvěma způsoby, jednak formou PWM signálu, jednak pomocí *I2C* rozhraní. Kompas ukrývá dvojici vzájemně kolmých senzorů magnetického pole založených na magnetorezistivním jevu. Přesnost měření se uvádí mezi třemi a čtyřmi stupni s rozlišitelností desetiný stupně. Podrobný popis kompasu lze nalézt v elektronické podobě v [24].

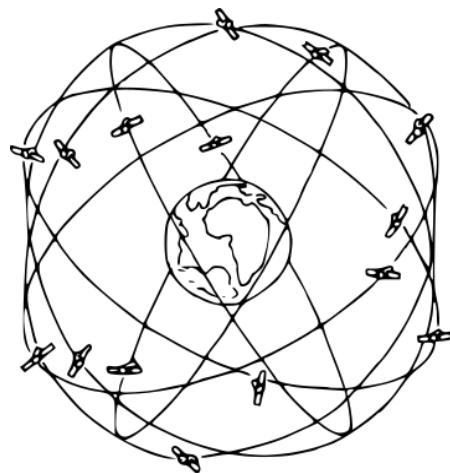
2.1.4 GPS

GPS spadá do třídy globálních polohovacích systémů postavených na satelitní navigaci [25]. Pozice přijímače se počítá z poloh satelitů pomocí určení časového zpoždění signálu, který satelity vysílají. Aby bylo možné určit polohu ve třech rozměrech a provádět korekci vnitřních hodin přijímače, tak musí být signál dostupný nejméně ze čtyř různých satelitů. V roce 2000 byl odstraněn umělý nízkofrekvenční šum zvaný *Selective Availability* (SA), jenž měl být pojistkou proti zneužití systému

GPS teroristickými skupinami, což znamenalo další zpřesnění určování polohy běžných auto-navigačních systémů (10 metrů, 95 %).

Při potřebě přesnosti do jednoho metru je nutné zvolit některou z tzv. diferenčních technik. Diferenční GPS (angl. *Differential GPS* – DGPS) poskytuje diferenční korekce pro opravu určování polohy pohyblivými přijímači, kterými lze eliminovat celou řadu chyb měření. Podmínkou pro uplatnění korekcí je, aby oba přijímače používaly ve svém výpočtu stejnou množinu satelitů. Bohužel DGPS nedokáže ze své podstaty postihnout místní chyby způsobené odrazy, útlumem atp., které jsou v parkovém prostředí díky vegetaci zcela běžné, a také nepostihuje vlastní šum přijímače. Z tohoto důvodu je přínos DGPS pro globální lokalizaci vůči klasickým GPS přijímačům v těchto prostředích spíše teoretickou záležitostí.

Na robotu je umístěna klasická USB GPS s čipem SIRF3. Zařízení podporuje EGNOS a WAAS, což deklaruje přesnost okolo pěti metrů v otevřených oblastech (30 metrů, pokud EGNOS/WAAS není dostupné).



Obrázek 5: Kosmický segment systému GPS (dráhy satelitů).

2.1.5 Laserový dálkoměr

Laserový dálkoměr (*Light Detection And Ranging*, LIDAR³) se řadí do kategorie aktivních bezkontaktních senzorů založených na principu měření doby letu vyslaného signálu (angl. *time of flight*). Výhoda použití laserového paprsku plyne z vysoké koncentrace energie s velmi malým rozptylem (nedochází k dramatickému snižování intenzity) a z monochromatickosti, jež obecně implikuje snazší zpracování signálu.

Nejčastěji emituje paprsek laserová dioda s příslušnou optickou soustavou⁴. Odražený paprsek je následně detekován pomocí lavinové fotodiody, která převádí světelnou vlnu zpět na elektrický proud. Dosah skenování závisí především na reflexivitě skenovaných objektů a na intenzitě vyslaného paprsku. Za ideálních podmínek je tato technologie schopna detekovat tmavé objekty s reflexivitou deset procent na vzdálenost třiceti metrů [26]. Odezva měření se odvíjí od jejich počtu a způsobu vyhodnocení signálu, přesto však lze konstatovat, že laserový dálkoměr pracuje asi tisíckrát rychleji, než sonarové čidlo [17]. Principiální nevýhodou 2D laserových dálkoměrů je rozmítání paprsku v jedné rovině, a tedy snímání relativně malé části z prostředí. Navíc s rostoucí vzdáleností roste vzdálenost mezi vyslanými paprsky a snižuje se tak absolutní rozlišení senzoru.

Robot je vybaven laserovým dálkoměrem od firmy SICK, konkrétně typem LMS 221 s rozhraním RS232. Tato řada laserových dálkoměrů je optimalizována pro měření vzdálenosti. Vyhodnocením pouze prvního impulzu (dodatečné impulzy na cestě jsou ignorovány) jsou zaručeny eliminace rušení způsobená odlesky a fakt, že pokud dojde k detekci objektu, leží tento objekt skutečně v cestě paprsku.

3 Ve vojenském se často používá akronym LADAR (*LAser Detection And Ranging*).

4 Vlnová délka paprsku činí 1064 nm.

Přesnost měření LMS 221 je typicky ± 35 mm a maximální dosah senzoru činí 80 m. Senzor dovoluje volit mezi dvěma velikostmi pozorovacího úhlu, a to buď 180° , nebo 100° , a mezi rozlišovací přesností 0.25, 0.5 a 1 stupeň (více viz [27]).

Mobilní robot používá laserový dálkoměr v režimu, kdy pozorovací úhel má hodnotu 100° a rozlišovací schopnost činí 1° . LIDAR je namontován na robota ve výšce 75 cm pod fixním úhlem 9° (implikuje snímání povrchu). Umístění snímače bylo zvoleno s ohledem na citlivost snímání, dosah a požadavek na minimální velikost detekovatelné překážky.

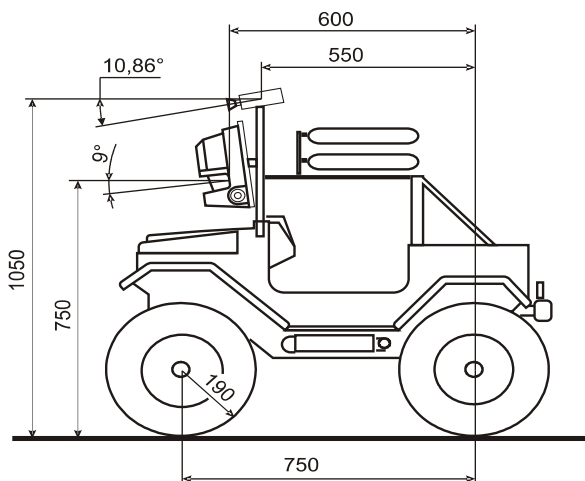
2.1.6 Kamera

Mezi pasivní optické snímače robota patří barevná CCD kamera SONY FCB-IX11AP připojená k počítači přes *FireWire* rozhraní, bohužel bez širokoúhlého objektivu. Hlavním prvkem každého čidla CCD snímače je Schottkyho dioda, jež převádí světelnou energii na elektrickou a ukládá ji do kapacitoru. Vyčítání této informace funguje na principu posuvného registru - každý kondenzátor předává svůj nahromaděný náboj sousednímu kondenzátoru. Výstup posledního kondenzátoru je zesílen a převeden pomocí A/D převodníku do digitální podoby. Mezi přednosti CCD technologie patří malý šum a velká citlivost. Nedostatky jsou dány technologií vyčítání (vzájemné ovlivňování sousedních pixelů, nemožnost adresovat samotné pixely) a malým rozsahem intenzit.

Kamera poskytuje snímky o rozlišení 384×288 pixelů a nachází se nad LIDAREm ve výšce 105 cm s pevným úhlem náklonu 10.86° do země. Náklon kamery je zvolen tak, aby se horizont nacházel těsně nad horní částí snímaného obrazu (v případě plochého terénu). Důvod tkví v redukci špatné saturace způsobené příliš světlou oblohou. Automatické vyvažování bílé dovoluje adaptaci na změnu barevného podání (teploty barev) při měnících se venkovních podmínkách.

2.2 Mechanická konstrukce

Jako základ pro vytvoření pohyblivého vozidla je použito dětské vozítko PégPerego (obr. č. 6) z tvrzeného plastu, jež má dva elektrické motory, hřebenové řízení a nosnost 50 Kg. Pro bezpečné uchycení senzorů byl přidán kovový rám z ohýbaného plechu, jenž je pevně spojen s kosterou robota. K rámu je připevněna přihrádka pro uchycení notebooků (počítá se maximálně se třemi). Kvůli přemístění těžiště byly přesunuty baterie z přední části auta do části zadní.



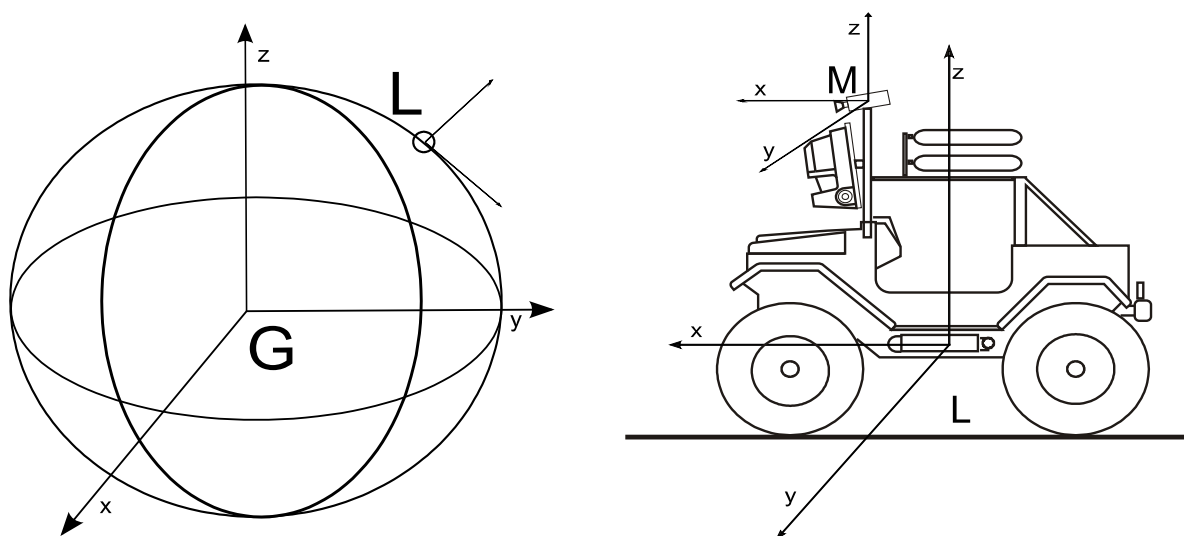
Obrázek 6: Robotické čtyřkolové vozidlo jménem Karlík (projekt Roboauto).

Dále bylo nahrazeno hřebenové řízení servomotorem, které je přímo připojené na táhla kol. Snaha o potlačení otřesů vedla k výměně originálních plastových kol za kola gumová a také k odpružení podvozku. Robot má oddělené napájení pro silovou část (motory + řízení) a pro elektroniku. Na obě části jsou shodně použity 12 V baterie Akuma Comfort 44 Ah, které předepisují průměrně až dvou hodinové ježdění při váze auta 40 Kg a rychlosti přibližně 2.5 km/h.

2.3 Souřadné systémy

K popisu polohy tělesa v prostoru slouží souřadný systém. Nejčastěji se používá známý kartézský souřadný systém zavedený francouzským filosofem René Descartem, kde jsou všechny souřadnicové osy tvořeny přímkami (přímočará soustava rovnic). Z pohledu modelování jsou důležité tři souřadné systémy, a to souřadný systém spjatý s měřicím zařízením, lokální a globální souřadný systém robota.

Veličiny měřené senzory lze chápat jako průběhy dynamických veličin vyjádřené v souřadném systému daného měřicího zařízení [21]. Avšak pro další zpracování (fúzi, rekonstrukci, atp.) je třeba znát průběhy těchto veličin v souřadném systému robota spojeném s neodpruženou hmotou a umístěným v jeho těžišti, tzv. lokálním souřadném systému. Globální souřadný systém se uplatňuje při definování celkové konfigurace, ve které se robot nalézá. Typicky se jedná o pozici robota a jeho natočení. Pakliže nebude uvedeno v textu jinak, tak bude implicitně předpokládán pravotočivý souřadný systém. Přehled uvažovaných souřadných systémů ilustruje obrázek č. 7.



Obrázek 7: Přehled uvažovaných souřadných systémů. *G* – globální souřadný systém, *L* – lokální souřadný systém, *M* – souřadný systém spjatý s měřicím zařízením (senzorem).

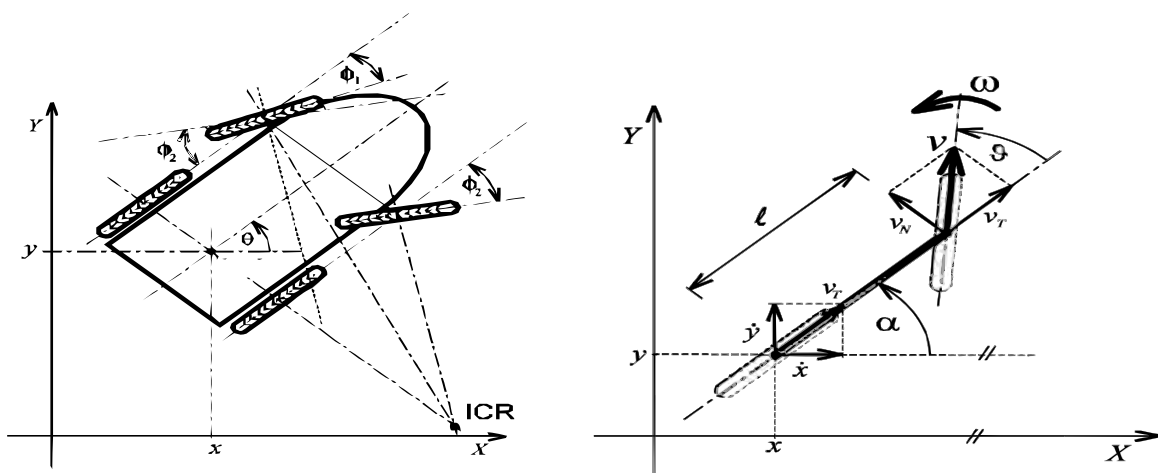
2.4 Kinematický model

Kinematika je oblast fyziky zabývající se geometrií pohybu, a proto v těchto modelech není uvažována hmotnost těles, setrvačnost, tření a ostatní konsekvence vyvolané působením sil, jež jsou nedílnou složkou modelů dynamických [28]. Díky této skutečnosti je kinematický model z hlediska reality značně zjednodušený a při jeho použití musí být splněn předpoklad, že otáčení kol vyvolá požadovaný pohybový efekt. Jinými slovy je třeba upravit dynamické vlastnosti robota tak, aby se

kinematický model stal dostatečně přesným, například použitím výkonných pohonů a omezením maximální rychlosti. Avšak toto řešení není vždy u reálných systému vhodné. Stejného efektu lze dosáhnout rozšířením kinematického modelu o integrátory na vstupech. Integrátory pak zaručují, že průběhy vstupních signálů budou mít spojitě derivace až do řádu n , kde n je počet přidávaných integrátorů.

Tvorba kinematického modelu zahrnuje volbu referenčního bodu a stavových proměnných, definování vstupních (řídících) proměnných a sestavení příslušných pohybových rovnic. Referenční bod reprezentuje pozici celého robota a jeho umístění implikuje tvar rovnic modelu i jeho vlastnosti. Stavovými proměnnými rozumíme minimální sadu veličin, které jednoznačně určují konfiguraci, v níž se systém nachází. V případě mobilních robotů jde typicky o pozici referenčního bodu a orientaci⁵ (u podvozku automobilového typu se ještě přidává informace o stavu celého podvozku, neboli informace o natočení nápravy s řízenými koly). Vstupní proměnné by měly odpovídat řídicím vstupům reálného robota, ačkoliv od tohoto pravidla je někdy vhodné ustoupit.

Automobilový podvozek sestává z jedné nápravy fixní (zadní) a z jedné s řízenými koly (přední). Protože osy kol jedné nápravy splývají, lze použít zjednodušený model při zachování kinematických vlastností, a to model pseudo-bicyklu. Ztráta informace o šířce robota je výhodná, neboť se není nutné zabývat přizpůsobením úhlu natočení jednotlivých kol v řízené nápravě, respektive rozdílem rychlostí kol v hnané nápravě.



Obrázek 8: Vlevo ilustrační obrázek Ackermannova řízení (problematika okamžitého středu otáčení). Vpravo zjednodušený model automobilu – bicykl. Převzato z [29].

Kinematicky zjednodušený model automobilu (bicykl) je vyobrazen na obr. 8. Rychlost pohybu v předního kola je rozložena na složku translační v_T a normálovou v_N . V případě podvozku ze zadním náhonem (robot *Karlík*) odpovídá translační rychlost v_T řídicímu vstupu v a platí:

$$\begin{aligned} \dot{x} &= v \cos \alpha \\ \dot{y} &= v \sin \alpha \end{aligned} \quad (1)$$

Protože $v_N = v \tan(\theta)$, pak:

$$\dot{\theta} = \frac{v_N}{l} = \frac{v \tan \theta}{l} \quad (2)$$

⁵ Složky vektoru pozice ve 3D prostoru tvoří souřadnice polohy (x , y , z) a úhly natočení tělesa kolem podélné, příčné a svislé osy (*roll*, *pitch*, *yaw*).

Dále platí, že $\omega = \dot{\vartheta}$ a zjednodušený model automobilu se zadním pohonem má konečný tvar:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\alpha} \\ \dot{\vartheta} \end{bmatrix} = \begin{bmatrix} \cos \alpha & 0 \\ \sin \alpha & 0 \\ \tan \vartheta / l & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (3)$$

2.5 Shrnutí

Na řízení robotů lze pohlížet ze dvou hledisek, a to jako na nízkoúrovňové řízení a vysokoúrovňové řízení. Cílem této kapitoly bylo představení mechanické konstrukce experimentální platformy *Karlík*, včetně sensorického subsystému, a stručný úvod do problematiky nízkoúrovňového řízení tohoto mobilního robota automobilového typu. Zároveň byla ilustrována uvažovaná trojice souřadných systémů pro zpracování sensorických měření a řízení robota, konkrétně globální a lokální souřadný systém a souřadný systém měřicího zařízení.

3 Navigační architektura

Navigace je pojem, který se váže k jakémukoliv pohybujícímu se tělesu. Představuje souhrnný název pro postupy a techniky, jimiž je možné kdekoliv na Zemi stanovit svoji polohu a nalézt optimální cestu do cíle vzhledem k zadaným kritériím. Termín pochází z latinského slova *navis* znamenající loď, jenž se později metonymicky přenesl na zjišťování polohy, směru a volbu trasy [29].

Navigace zastřešuje v mobilní robotice obecné otázky, které v roce 1991 sumarizoval Leonard a Durrant-Whyte na tři, a to na „Kde jsem?“ („*Where am I?*“, lokalizace robota), „Kam chci jít?“ („*Where am I going?*“, rozpoznání cíle) a „Jak se tam mohu dostat?“ („*How should I get there?*“, plánování cesty). Řešení těchto otázek zasahuje do volby metod určování pozice, vnitřní reprezentace okolí, fúze senzorických dat, plánování, řídicí architektury atd., samozřejmě v kontextu řešené úlohy a s ohledem na prostředí, v němž se robot pohybuje.

Úroveň řízení autonomního robota implikuje možnosti přístupu k navigaci. Bezpilotní vozidla se řadí svou úrovní řízení do kategorie kognitivních systémů, neboť se jedná o systémy schopné samostatné interakce s reálným prostředím za účelem splnění stanoveného cíle. Autonomně řízené vozidlo tedy ztělesňuje kybernetický systém rozpoznávající své okolí, tvořící jeho vnitřní reprezentaci, řešící nepředvídané události, formulující si vlastní podcíle, samostatně vykonávající své plány atp.

Existence metod pro plánování cesty, lokalizaci, vyhýbání se překážkám, percepční interpretaci atd. je výsledkem dekompozice komplexního procesu navigace kognitivního robota. Výzkum řídicích architektur proto v principu znamená studování designu návrhu softwarových modulů a jejich součinnosti za účelem autonomní navigace.

Avšak robotický systém nebude zkoumán v celé své obecnosti. Náplní této kapitoly je stručná rešeršní studie na témata týkající se tvorby, reprezentace a interpretace mapy okolního světa, metod lokalizace a typů řídicích architektur, která poslouží jako terminologické a teoretické podhoubí závěrečné podkapitoly - popisu konceptuálního modelu robota *Karlik*.

3.1 Typy architektur

V raných dobách teorie mobilních robotů se bádání soustředilo více na získávání a interpretaci reálných dat, než na samotnou efektivitu a limity architektur, jež je řídily. Kupříkladu první navigační systémy robotů se skládaly z pevné sekvence volání jednotlivých modulů, přičemž výstup jednoho modulu byl vstupem následujícího. Další vývoj robotiky přinesl zdokonalování tehdejších architektur řídicích systémů, ale i koncepčně nové postoje založené na moderních přístupech v umělé inteligenci. Výsledkem tohoto vývoje je existence mnoha softwarových architektur, jež lze rozdělit do pěti skupin (dle [17]), a to na:

- architektury vycházející z funkční dekompozice,
- reaktivní systémy,
- plánování na vyšší úrovni,
- neuronové sítě a genetické algoritmy,
- hybridní systémy.

3.1.1 Funkční dekompozice

Funkční dekompozice reflektuje klasický přístup dekompozice shora dolů, která se uplatňuje v řadě jiných aplikací, nejenom v oblasti robotiky (např. modulární návrh programu). Při návrhu řídicího systému se postupuje dělením úlohy na jednodušší podúlohy, jež řeší samostatné moduly. Základní kategorizace dle funkcionality dělí moduly na modul vnímání, modul modelování, modul prostředí, modul plánování, modul realizace plánu a modul řízení pohybu (zjednodušeně též na modul snímání, modul plánování a modul akce).

Vzájemné provázání modulů lze řešit několika způsoby. Nejjednodušší způsob, vyskytující se u prvních experimentálních robotů, sekvenčně propojuje moduly do cyklu snímání – plánování – akce. Bohužel toto provázání má základní nedostatky v pevně daném toku dat, kdy chyba jednoho modulu se dále distribuuje do modulů následných. Problematická je rovněž rychlost reakce systému, neboť ta je limitována součtem časových zpoždění všech modulů zpracovávajících vstupní informaci.

Jiná varianta funkční dekompozice využívá hierarchického zapojení modulů. Hierarchický systém sestává z více vrstev, kde každá vrstva reprezentuje jeden cyklus snímání – plánování – akce. Vrstvy se od sebe liší mírou abstrakce, a tedy časem potřebným pro vykonání řídicího cyklu.

Posledním přístupem v řešení koordinace modulů je použití tabule. Tabule představuje komunikační kanál, respektive místo, s jehož pomocí si všechny moduly vyměňují informace (princip ne nepodobný nástěnce v programovacím jazyce Linda).

3.1.2 Reaktivní systémy

Naopak reaktivní architektura se řadí mezi příklady dekompozice zdola nahoru. Tento typ architektury byl inspirován chováním primitivních zvířat, kde se předpokládá, že akce na podněty vycházejí z modelů chování a nejsou výsledkem procesu porozumění a vyšších myšlenkových pochodů. U reaktivních systémů tudíž jednotlivé moduly přímo ovládají senzorický a motorický subsystém robota a prostředí není buď vůbec, nebo jen velmi hrubě modelováno⁶. Paradigmatem této architektury je proto řetězec snímání – akce.

Výhoda reaktivních systémů spočívá v nepřímé specifikaci cílů. Tyto systémy byly vytvořeny pro řešení jednodušších úloh, kde se vyžaduje vysoká robustnost a necitlivost na nepředpokládané změny prostředí. Nevýhoda plyne z neexistence modelu okolí, a tedy z poten-ciální hrozby uvážnutí v lokálním extrému řídicích funkcí.

3.1.3 Plánování na vyšší úrovni

Reaktivní architektura, stejně jako architektura vycházející z funkční dekompozice reflektuje procedurální reprezentaci znalostí v umělé inteligenci (procedurální přístup). Jinou možností jsou architektury založené na deklarativní reprezentaci znalostí o světě. Jedná se proto o logický (symbolický) přístup k umělé inteligenci, jenž je podnícen oblastmi psychologie a logiky. Do této skupiny spadá např. plánování akcí pomocí algoritmu STRIPS, systémy na bázi jazyka PROLOG atp.

3.1.4 Neuronové sítě a genetické algoritmy

U úloh, které vyžadují akceptaci všudypřítomné nepřesnosti reálného světa, vysokou robustnost, běh v reálném čase atd. se uplatňují metody z oblasti soft-computingu. Jedním z postojů k umělé

⁶ Vychází se z předpokladu, že prostředí samo o sobě je svým nejlepším modelem.

inteligenci, jenž spadá do této kategorie, je přístup založený na neuronových sítích. Jako příklad úspěšné aplikace neuronových sítí v navigaci autonomního robota může sloužit již zmíněné řízení bezpilotního vozidla ALV z roku 1993. Systém byl ovládán neuronovou sítí typu vícevrstvý perceptron s adaptivní dynamikou zpětného šíření chyby, jejíž 30x32 neuronů vstupní vrstvy bylo přímo napojeno na výstup z kamery. Sít' obsahovala pouze jednu skrytou vrstvu se čtyřmi neurony a výstupní vrstvu o třiceti neuronech. Výstup neuronové sítě byl promítnut do úhlu natočení přední nápravy.

Problém tzv. neuronového přístupu spočívá v nutnosti existence vhodné trénovací sady dat a v nepředvídatelnosti chování této architektury, neboť je velmi obtížné interpretovat hodnoty vah sítě jako pravidla definující závislost mezi vstupem a výstupem, a tím ověřit a deklarovat bezpečnost celého systému.

Své uplatnění našly v robotice i genetické algoritmy, a to jak na nízké úrovni (např. v učení pohybu robota), tak i na vyšších úrovních pro plánování akcí a řešení rozmanitých optimalizačních úloh. Pozitivní vlastností genetických algoritmů je, že vždy poskytnou nějaké řešení v omezeném čase, byť nemusí být optimální, a také, že lze relativně snadno charakterizovat hledané řešení pomocí fitness funkce.

3.1.5 Hybridní systémy

V praxi sestává řídicí systém robota většinou z kombinace předchozích přístupů. Hybridní systémy jsou opět děleny do vrstev, avšak v jednotlivých vrstvách mohou být aplikovány odlišné typy architektur. Hybridní systém se snaží vhodnou kombinací dílčích přístupů eliminovat nejpálčivější problémy v řízení mobilních robotů.

3.2 Lokalizace

Lokalizace robota je vnímána jako jeden z fundamentálních problémů inteligentní robotiky, jež hledá odpověď na otázku „Kde jsem?“. Lokalizovat mobilního robota znamená stanovit pozici robota relativně k dodané mapě prostředí. Proto lze problém lokalizace formulovat i jako problém ustanovení korespondující transformace koordinát mezi souřadným systémem robota a souřadným systémem mapy [30].

3.2.1 Taxonomie lokalizační problematiky

Ne každý lokalizační problém má stejnou obtížnost. Pod pojmem lokalizace se ukrývá řada instancí tohoto problému s různým stupněm složitosti [30]. V případě trasování pozice (angl. *position tracking*, nebo *local localization*) zná robot počáteční lokaci. Cílem lokalizace je zpřesňovat pozici během vlastního pohybu robota, která se vlivem nepřesností (nerovnost terénu, vůle v řízení atp.) zhoršuje. Metody řešící trasování pozice mají přívlastek lokální, nebo sledovací a patří k těm jednodušším (eliminují pouze chyby měření ujeté vzdálenosti).

Globální lokalizace (angl. *wake-up robot*, nebo *global positioning*) je náročnější než lokální lokalizace, jelikož není známa iniciační pozice robota a robot se tedy musí umět lokalizovat od píky. Nadto řešení globální lokalizace může někdy vést k práci s vícero počátečními možnostmi polohy robota. Metody pro nalezení globální polohy se nazývají globální techniky.

Jeden z nejtěžších případů určení polohy robota v prostoru ztělesňuje problém tzv. „únosu“ (angl. *kidnapped robot problem*). V této situaci robot na počátku přesně zná svoji polohu, ale následně dojde k jeho nevědomému přesunu na jiné místo. Robot si proto stále „myslí“, že se nachází v původní lokaci, ačkoliv se fyzicky nalézá již ve zcela jiném prostředí. Přestože se problém jeví jako čistě akademický, lze i tak snadno najít možné příčiny, jejichž následky budou odpovídat svým charakterem situaci „únosu“, např. pád robota, chyba apod.

Faktor, který značně komplikuje řešení všech uvedených instancí lokalizačního problému, představuje dynamika prostředí robota. Většina metod určování pozice se koncipuje do statických prostředí, kde robot tvoří jediný objekt schopný měnit jeho stav. Naproti tomu dynamický svět obsahuje další aktivní elementy. Lokalizace v prostředí, které se mění i bez intervence robota, patří k signifikantně náročnějším úlohám.

3.2.2 Dostupné informace

V procesu estimace pozice má robot k dispozici dva druhy informace, a to apriorní informace dostupné z externího zdroje a informace získané z vlastního pohybu prostředím, tzv. navigační informace. Apriorní znalosti upotřebitelné v lokalizaci charakterizují prostředí, v němž se robot pohybuje a jsou časově invariantní, typicky jde o mapy okolí. Do kategorie navigačních informací spadají nejen měření senzorického subsystému, ale i informace ze samotného řízení robota, respektive z monitorování činnosti řídicího subsystému.

3.2.3 Měření pozice

Většina řešení měření pozice může být přímo rozdělena do dvou hlavních skupin. První skupinu tvoří metody pro relativní měření pozice, známé též pod názvem lokální, a druhou skupinu metody pro absolutní měření pozice (ekvivalentně označovány jako metody globální měření pozice). Protože se obě skupiny vzájemně doplňují, tak bývá v praxi často využívána jejich kombinace [19, 31].

Relativní měření pozice

Původně byl tento proces odhadu aktuální pozice, známý rovněž jako *dead reckoning*, založen pouze na dřívější poloze, uplynulém čase, rychlosti a směru pohybu. Zřejmá nevýhoda relativního měření se skrývá v charakteristickém integračním výpočtu, jenž vede k časové kumulaci chyby. V robotických aplikacích je relativní měření pozice postaveno buď na odometrii, nebo na inerciální navigaci.

Slovo odometrie pochází z řečtiny a vzniklo konkatencí řeckých slov pro cestování (*hodos*) a měření (*metron*). Odometrie pracuje na principu integrace inkrementální informace dle času a bývá široce aplikována právě kvůli vynikající krátkodobé přesnosti, nízké ceně a vysoké vzorkovací frekvenci. Nevýhoda odometrie spočívá ve zmíněné časové kumulující se chybě, a proto musí být kombinována s další nezávislou metodou, jež periodicky redukuje narůstající nepřesnost.

Inerciální navigace staví na vysoce citlivých gyroskopech a akcelerometrech pro měření rychlosti rotace a zrychlení robota. Podobně jako u odometrie se i zde pozice estimuje integrací získaných měření dle času. Výhodou je, že systém není závislý na vnějšku a jedná se tudíž skutečně o interní navigaci. Nicméně díky integračnímu procesu, který má navíc v případě výpočtu pozice dvojitou podobu, a velkému počtu vlivů, jež mohou negativně ovlivnit měření (teplota, mechanické

provedení atp.), má jen krátkodobou přesnost. Mezi nepříjemné aspekty se řadí i cena přesných inerciálních jednotek.

Absolutní měření pozice

Absolutní měření pozice poskytuje informaci o lokaci robota nezávisle na předchozích stavech, neboli poloha není počítána integračními postupy, ale je přímo získána měřením. Důsledkem toho je, že chyba estimace nemůže neomezeně růst jako při relativním měření pozice. Absolutní měření pozice se kategorizuje do metod vybudovaných na orientačních bodech (angl. *landmark based*) a do metod postavených na mapách (angl. *map based*).

U technik absolutní měření pozice vycházející z detekce orientačních bodů rozlišujeme ještě typy orientačních bodů, a to na pasivní a aktivní. Aktivní orientační body, nazývané rovněž jako majáky (angl. *beacons*), samočinně emitují do prostředí lokalizační informaci a nabývají podoby např. satelitů, nebo radiových majáků. S využitím triangulace, respektive trilaterace⁷ lze pak determinovat polohu robota přijímajícího vysílané informace. Pasivní orientační body aktivně nevyzařují do prostředí žádnou informaci a robot musí tyto body sám činně vyhledávat pro odhad své pozice. Pasivní orientační body se dále dělí na umělé a přírodní dle toho, zda se v prostředí vyskytují přirozeně, nebo do něj byly účelně zaneseny.

Jinou skupinou metod spadajících do kategorie absolutní měření pozice jsou metody založené na mapách. Tyto přístupy uplatňují signifikantní geometrické znaky prostředí ke kalkulaci polohy robota. Princip spočívá ve tvorbě lokálního modelu okolí a následném porovnání tohoto modelu s globální mapou uloženou v paměti robota. Tvorba lokálního modelu bývá zpravidla výpočetně náročná a je rovněž úzce spjata s potenciálem senzorického subsystému. Proto se často úloha zjednodušuje na porovnání tvaru ujeté trajektorie, nebo na nalezení se v mapě při znalosti přibližné globální polohy.

3.2.4 Lokalizační techniky

Současný trend v této oblasti jde směrem fúze relativní a absolutní měření pozice. Snahou je poskytnout přesnější estimaci polohy robota vybudovanou na odlišných typech senzorů, jejichž charakteristiky chyb se vzájemně doplňují. V této souvislosti lze proto problém lokalizace chápat i jako problém fúze senzorických měření⁸ za účelem získání přesnějších výsledků polohy (typicky u kontinuální lokalizace – *position tracking*).

Existence chyb a šum senzorů je povšechný faktor, který komplikuje odhad pozice, neboť agregace nejisté informace povede ke značně nepřesným výsledkům. Z tohoto důvodu je třeba při estimaci polohy kalkulovat s chybami měření a vytvořit tak více realistický model fyzikálního světa. Pravděpodobnostní vzhled do problematiky lokalizace dovoluje přirozený popis nejistoty reálného světa. Navíc lokalizační techniky založené na pravděpodobnostní teorii⁹ poskytují aparát k fúzi senzorických měření *in situ*, respektive rozšíření o integraci dat z více zdrojů není obtížné

7 Trilaterace uplatňuje pouze vzdálenostní informaci (např. GPS). Triangulace využívá vztah mezi vzájemnou orientací robotu a polohy majáků [20].

8 Pochopitelně nemusí jít o přímá měření. Fúze může probíhat i u předzpracovaných dat, avšak musí být zajištěna korektnost agregačního modelu.

9 Prakticky většina používaných přístupů je založena na Bayesově inferenci a má podobu rekurzivních filtrů. Více viz [30, 32].

(angl. *multi-sensor multi-temporal data fusion*, viz [32]). Mezi nejpropíranější metody lokalizace v posledních letech patří varianty Kalmanovy filtrace a Monte Carlo lokalizace.

Kalmanova filtrace

Nejčastěji používanou metodou pro fúzi senzorických měření v mobilních robotických aplikacích se stal Kalmanův filtr popsáný roku 1960 Rudolfem Kalmanem. Je-li systém definován lineárním modelem a chyby systému i chyby měření mohou být modelovány jako bílý Gaussův šum, pak Kalmanův filtr poskytuje statisticky optimální odhad pro agregaci dat ze senzorů [31, 33].

Základní podoba Kalmanova filtru (LKF) je tudíž limitována lineárním předpokladem. Avšak komplexní systémy zpravidla lineární nebývají. Nelinearita může být asociována buď s modelem prostředí nebo s modelem měření. Za předpokladu, že model měření i prostředí lze matematicky formulovat pomocí parciálních derivací nelineárních funkcí dle vnitřních stavů systému, tak může být aplikován tzv. Rozšířený Kalmanův filtr (*Extended Kalman Filter*, EKF). Alternativu k EKF představuje *Unscented Kalman Filter*¹⁰ (UKF), jenž na rozdíl od analytického řešení propagace náhodných proměnných s Gaussovým rozdělením skrze nelinearitu systému v EKF, vychází z myšlenky deterministického vzorkování za použití tzv. sigma bodů (více viz [34, 35, 36], nebo v kapitole *Lokální navigace*).

Monte Carlo lokalizace

Metoda Monte Carlo lokalizace, MCL, patří k moderním lokalizačním technikám spadajících do kategorie částicových filtrů. Klíčová myšlenka MCL se skrývá v popisu hustoty pravděpodobnosti množinou vzorků (u KF se jedná o aproximaci v parametrické formě). Jinak řečeno, MCL jednoduše reprezentuje pozici robota „náhodně“ vytvořenou kolekcí váhovaných částic, které aproximují požadovanou distribuci. Díky tomu nejsou kladena žádná omezení na tvar hustoty pravděpodobnosti a lze tedy bez obtíží obsáhnout i multimodální hustoty pravděpodobnosti (mající více než jeden vrchol), což bývá přínosné při globální lokalizaci robota (více viz [30, 37]).

3.3 Reprezentace světa

V oblasti mapování prostředí v posledních třiceti letech probíhá velmi aktivní výzkum, jelikož existence modelu fyzického okolí je klíčovým prvkem autonomních mobilních systémů. Prakticky jakýkoliv řídicí systém na vyšší úrovni řízení staví na vnitřní reprezentaci prostředí. Mapa může být zadána z vnějšku, nebo si ji robot autonomně buduje během svého pohybu. V drtivé většině případů nastává druhá varianta, protože mapy ve vhodném formátu nebývají k dispozici a z principu nemohou odpovídat skutečnosti (dynamické prostředí).

V současné době existují robustní metody pro mapování statických prostředí, jež jsou navíc strukturovaná a velikostně omezená. Tyto metody jsou zpravidla vybudovány na pravděpodobnostních přístupech. Mapování dynamického prostředí ve velkém měřítku se stále ještě řadí k dnešnímu dni k otevřeným problémům [38].

Nejen šum a chyby senzorů znesnadňují proces tvorby mapy. Komplikace způsobuje i omezený dosah senzorů a jejich necitlivost vůči některým povrchům objektů. Dále v okamžicích

¹⁰ Název filtru reflektuje techniku vzorkování známou jako *unscented transform* (transformace charakteristických bodů).

snímání dat je třeba znát polohu senzorů, a tedy i polohu robota, která bývá zpravidla největším zdrojem chybovosti (např. kumulovaná chyba v odometrii). Mnoho existujících mapovacích algoritmů je proto velmi složitých.

3.3.1 Typy map

V robotice se uplatňují různé druhy map. Volba konkrétního modelu okolního prostředí robota souvisí s úlohou, kterou má robot řešit. Jinou strukturu budou mít modely prostředí pro exaktní plánování ve srovnání s modely pro generování potenciálových polí atp. Základní dělení tyto mapy klasifikuje na mapy senzorické, mapy geometrické, mapy topologické (někdy též nazývané jako relační) a mapy symbolické [39].

Senzorické mapy mohou uchovávat přímá měření ze senzorů, nebo poskytovat prostor pro první agregaci měření z jednotlivých senzorů. Jedná se tudíž spíše o vhodně strukturovaná data. Senzorické mapy, podobně jako geometrické mapy, jsou význačné pevným souřadným systémem. Typickým představitelem těchto map je mřížka obsazenosti.

Velmi efektivní reprezentaci prostředí deklarují geometrické mapy, jež popisují objekty pomocí geometrických primitiv v pevném kartézském souřadném systému. Mezi tato geometrická primitiva patří úsečky, polygony, oblouky atd. Geometrické mapy díky tomu poskytují určitou míru abstrakce abstrahující model od přílišných detailů, což vede k nižším paměťovým nárokům. Tvorba těchto map je výpočetně náročnější, nežli tvorba senzorické mapy, jelikož je třeba nalézt vhodné geometrické primitivum popisující daný tvar. Geometrické mapy, které v sobě zahrnují informaci o propojení sousedních objektů, nebo obsahují dekompozici volného prostoru na konvexní části, lze uplatnit v řadě plánovacích algoritmů.

Naproti tomu topologické mapy uchovávají pouze informaci o relaci mezi objekty, o propojení jednotlivých prostor, případně jiné informace (jména objektů atp.). Topologické mapy mají proto formu grafů, kde uzly ztělesňují pozorované objekty a spojnice představují vztahy mezi nimi. Výhodou těchto map je, že dovolují integraci informací z rozsáhlých ploch bez enormních paměťových nároků a přirozeně zachycují podstatné vlastnosti pro plánování činnosti. Na tvorbu této mapy nemá vliv neurčitost polohy mobilního robota, avšak nastává tu problém s rozpoznáváním a identifikací význačných vlastností, které by měly být při autonomním budování do mapy zaznamenány.

Přirozený jazyk je lákavou možností jak ovládat robota. Komunikace robota s operátorem je doménou symbolických map. Tyto mapy obsahují informace, které nejsou zjistitelné senzorickým subsystémem robota, ačkoliv jsou nezbytné pro zadávání příkazů robotovi pomocí přirozeného jazyka. Implementace symbolické mapy se zpravidla opírá o jazyk predikátové logiky (snadné navázání na algoritmy analýzy přirozené řeči).

3.3.2 Mřížka obsazenosti

Mřížka obsazenosti představuje výchozí abstraktní model řady počítačových algoritmů z oblasti pravděpodobnostní robotiky, které řeší problém generování map ze zašumělých a nejistých senzorických měření¹¹. Základní idea mřížky obsazenosti zakládá na modelování okolního světa jako

¹¹ Jelikož v základní podobě mřížky obsazenosti nepředpokládají, že vkládané hodnoty mohou pocházet z více typů senzorů, omezme se implicitně na případ jednoho senzoru a více měření. Fúze senzorických vjemů z více zdrojů bude popsána dále v konkrétní aplikaci (kap. *Lokální navigace*).

stejněměrného prostorového pole buněk, kde hodnota buňky reflektuje míru obsazenosti, či volnosti jednotlivých částí mapovaného prostoru.

Mřížky obsazenosti poprvé představil v roce 1989 Albert Elfes [40]. Od té doby si získaly oblibu jako velmi stabilní prostředí pro datovou fúzi (angl. *grid-based fusion*) měření ze senzorů. Rovněž poskytují jednotný přístup k řešení rozmanitých problémů v prostorové percepci a navigaci mobilních robotů. V praxi se uplatnily tři základní podoby, jak reprezentovat informaci o obsazenosti v mřížce [17], a to pravděpodobnostní přístup, věrohodnostní přístup a fuzzy přístup.

Pravděpodobnostní mřížka obsazenosti modeluje okolí dvoudimenzionálním polem buněk¹², kde hodnota každé buňky definuje pravděpodobnost s jakou je namapovaný segment v reálném prostředí obsazen překážkou. Pravděpodobností se zde rozumí poměr počtu světů, které odpovídají dosavadním měřením při obsazení zkoumané buňky, ku počtu všech světů, jež odpovídají dosavadním měřením. Formálně, nechť $s(a) = O$ značí stav *obsazeno* buňky a a nechť $s(a) = V$ značí stav *volno* buňky a , potom pravděpodobnost obsazení buňky a za podmínky měření r je dána formulí:

$$P(s(a)=O|r) = \frac{\sum_{m \in M \wedge s(m_a)=O} P(m|r)}{\sum_{m \in M} P(m|r)}, \quad (4)$$

kde M je množina všech možných světů, m_a označuje buňku odpovídající buňce a ve světě m a $P(m|r)$ pravděpodobnost světa m při provedení měření r .

Věrohodnostní přístup staví na Dempster-Shaferovy teorii, která tvoří alternativu k popisu nejistých událostí podobně jako teorie fuzzy množin Lotfiho Zadeha. Pakliže potřebujeme rozdělit buňky mřížky na obsazené a volné, tak základní množinu stavů tvoří dvouprvková množina $\Omega = \{V, O\}$. Množina všech případů se pak definuje jako $\Lambda = 2^\Omega = \{\emptyset, V, O, \Omega\}$ (v souladu s D - S teorií, viz [41]). Míra důvěry v obsazenost jedné buňky se spočte dle:

$$\begin{aligned} Bel(V) &= \sum_{A \in \Lambda, A \subseteq V} p(A) = p(V) \\ Bel(O) &= \sum_{A \in \Lambda, A \subseteq O} p(A) = p(O) \end{aligned}, \quad (5)$$

přičemž p je pravděpodobnostní přiřazení $p: \Lambda \rightarrow \langle 0, 1 \rangle$.

Fuzzy logika nachází uplatnění v aplikacích, kde je třeba postihnout dva odlišné typy neurčitosti. Jedná se o vágnost (nejasnost, mnohoznačnost) a o „nedostatek důkazů“ (není jasné, je-li prvek členem konkrétní ostré množiny, angl. *lack of evidence*). Při tvorbě mapy okolí se pracuje s druhou skupinou neurčitosti, neboť lze tvorbu mapy formulovat jako rozhodovací problém zdali daná buňka v oblasti zájmu je obsazena, nebo nikoliv. Proto fuzzy přístup vychází z definice dvou nezávislých fuzzy množin Of a Vf , kde množina Of zahrnuje buňky, které jsou obsazeny, komplementárně množina Vf buňky, jež jsou neobsazeny. Funkce příslušnosti následně reflektují modely senzorů. Způsob agregace měření odpovídá konkrétní implementaci fundamentálních množinových fuzzy operací (průnik, sjednocení, doplněk, atd.). Více viz [42].

¹² Techniky tvorby mřížky obsazenosti lze zobecnit i pro 3D reprezentaci, avšak dojde k signifikantnímu nárůstu výpočetní složitosti. Nejen proto budeme dále v textu uvažovat pouze dvourozměrné modely.

3.4 Konceptuální model

Autonomní řízení musí být spolehlivé, robustní vůči chybám a musí běžet v reálném čase. Navržený konceptuální model pokrývá tyto požadavky pomocí distribuované softwarové architektury. Architektura vychází z paralelního řetězeného zpracování, a proto komunikace mezi moduly odlišné vrstvy je seřazená a bufferovaná. Vstupní data ze senzorů jsou postupně transformována do interních modelů, plánů činnosti a nakonec konkrétních pohybových úkonů. Paralelní řetězení datového toku minimalizuje dobu zpracování, respektive čas mezi získáním dat ze senzorů a odpovídajícím kontrolním akcím robota. Koncept softwarové architektury sestává ze třinácti modulů běžících souběžně (obr. č. 9), jež se dělí do čtyř vrstev:

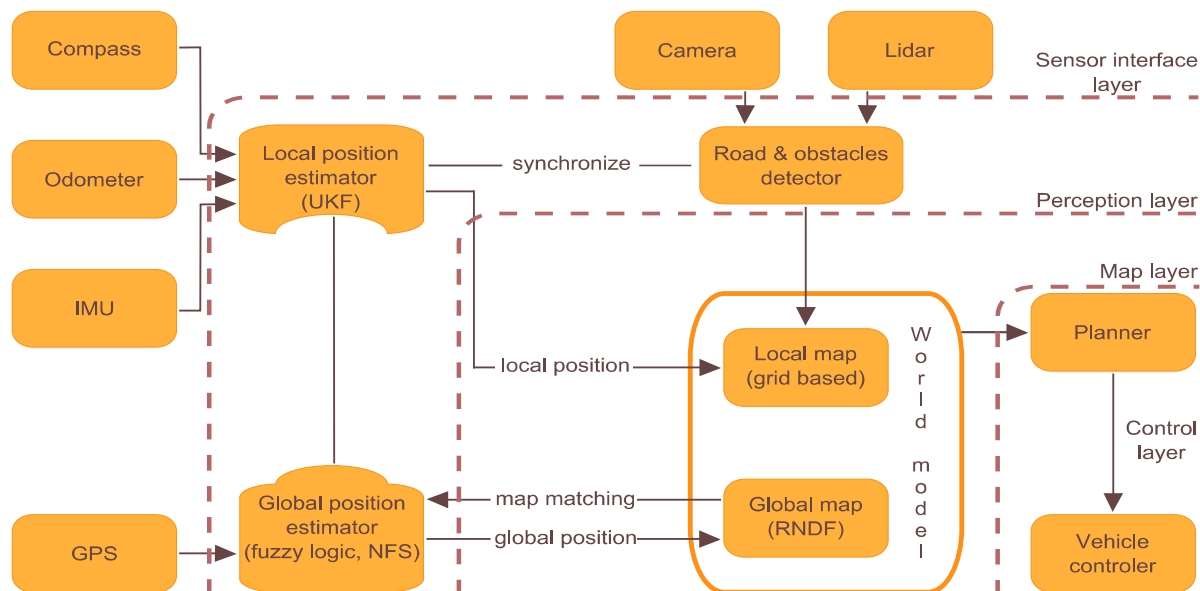
- Senzorická vrstva zahrnuje moduly rozhraní k dílčím senzorům, které provádí nízkoúrovňové předzpracování dat (např. odstranění chyby způsobené unášením u gyr) a data časově orazítkuje. Frekvence příchodu dat z LIDARU je 15 Hz, z IMU 20 Hz, z rotačního enkodéru 25 Hz, z kamery 4 Hz a z GPS senzoru 1 Hz.
- Percepční vrstva transformuje měření do interních datových struktur, respektive vykonává vysokoúrovňové zpracování dat. Primárními moduly této vrstvy jsou moduly poziční estimace, která je dvojího typu – lokální a globální. Lokální estimátor se datově (nikoliv časově) synchronizuje s moduly detekce sjízdného povrchu z dat laserového dálkoměru a kamery, které pracují konkurenčně¹³.
- Mapovací vrstva se skládá ze dvou vnitřních modelů světa, jednak pro lokální navigaci, jednak pro globální navigaci. Lokální model je budován mapováním výsledků z modulů detekce sjízdného povrchu dle polohy získané z lokálního odhadu pozice do vnitřní reprezentace založené na mřížce obsazenosti. Globální mapa není autonomně tvořena, ale je do systému dodána z vnějšku ve formátu RNDF.
- Kontrolní vrstva se stará o plánování a řízení robota jak na lokální úrovni (vyhýbání se překážkám, udržení robota na cestě), tak i na globální úrovni (rozhodování na křižovatkách, naplánování trasy do cíle).

Kognitivní systém se dvěma vnitřními modely světa dle měřítka navigace představuje účinnou architekturu řešící celou řadu stěžejních problémů v autonomních mobilních systémech pohybujících se v rozsáhlých nestatických prostředích. Mezi hlavní obtíže unimodálních interních modelů patří otázka lokalizace robota a souběžná tvorba mapy okolí (alias SLAM), jež je nezbytná pro vyšší úroveň řízení. Tvorba interního modelu vyžaduje stabilní a přesnou estimaci polohy, jinak by docházelo ke špatnému mapování výsledků detekce přes sebe a tedy k zanášení umělých překážek do vnitřní reprezentace. Předpoklad prostorové spojitosti odhadu pozice je díky charakteristice chyb některých absolutních senzorů (např. GPS) splnitelný jen s obtížemi (de facto pouze rozprostřením náhlé poziční změny v čase). Navíc další nesnáze vznikají ve volbě vhodné datové struktury, která by poskytovala dostatečný detail a možnost fúze senzorických vjemů, a to za podmínky nízké paměťové náročnosti a vysoké rychlosti plánovacích algoritmů operujících nad tímto modelem.

Nejen z těchto důvodů je vhodné využít přirozené dekompozice navigace na lokální a globální úroveň, přičemž lokální úroveň navigace je míněna navigace v nejbližším okolí robota. To jest vyhýbání se překážkám, jízda po cestě a detekce křižovatek. Naopak globální navigace řeší otázky

¹³ V obrázku č. 9 jsou zahrnuty do jednoho funkčního bloku, ačkoliv se ve skutečnosti jedná o dva paralelně běžící moduly, pro každý senzor samostatně.

týkající se globálního záměru, například správné odbočení, nebo nalezení posloupností navazujících segmentů vedoucích do cílové pozice na dodané globální mapě. Jednotlivé aspekty související s dvouúrovňovou navigací robota jsou popsány z důvodu větší přehlednosti a obsáhlosti textů v následujících kapitolách samostatně, a to včetně lokalizačních technik a vnitřního modelu prostředí.



Obrázek 9: Konceptuální model robota Karlik.

3.5 Shrnutí

Obsahem této kapitoly byla velmi stručná rešeršní studie problematiky vysokoúrovňového řízení mobilního robota. Byl uveden historický sled vývoje řídicích architektur a jejich stručná charakteristika, dále byly nastíněny základní principy a techniky v otázkách lokalizace robota a konečně možnosti ve vnitřní reprezentaci okolního prostředí. Závěr kapitoly se věnuje vlastnímu návrhu paralelně zřetězené navigační architektury s dvouúrovňovou reprezentací světa pro řízení mobilního robota *Karlik*, která se snaží postihnout nejpalčivější otázky těchto autonomních systémů.

4 Detekce cesty

Jedním z majoritních úkolů v řízení autonomních mobilních robotů v předem neznámém prostředí je schopnost detekce sjízdného povrchu. Proměnnost a variabilita okolí, v němž se robot pohybuje, činí tento úkol značně obtížným. Nadto dynamičnost prostředí nemusí být nutně chápána v kontextu s viditelnými zásahy do snímané scény, neboť například díky nepřesnosti senzorů, šumu a jiným vlivům bude i zcela statická scéna vnímána robotem vždy trochu jinak. Povaha úlohy vyžaduje:

- robustnost algoritmů vůči proměnným podmínkám venkovního prostředí,
- dostatečnou přesnost a rozumnou míru jistoty rozpoznávacích metod,
- interpretaci sensorových dat v reálném čase.

Detekce a trasování cesty je netriviální úlohou, kterou lze řešit mnoha strategiemi. Dílčí strategie se velmi liší především ve vstupních předpokladech o prostředí, v typu použitých senzorů (aktivní/pasivní), v jejich počtu, v samotné fúzi sensorových dat, výpočetní náročnosti, modelu okolí atd. Současný trend této oblasti jde směrem multi-sensorových systémů, zejména velice častá je kombinace CCD-kamer a laserových dálkoměrů.

4.1 Detekce cesty laserovým dálkoměrem

Laserové dálkoměry jsou s úspěchem používány jak pro detekci překážek, tak i detekci sjízdného povrchu a patří mezi velice často používané senzory v mobilní robotice. Jejich výhodou je nezávislost na denním osvětlení, relativně vysoká přesnost a u novějších zařízení i rozlišovací schopnost. Jako nevýhodu uvedme vyšší pořizovací cenu oproti CCD-kamerám, závislost na venkovním počasí atd.

Dnešní pozornost se zaměřuje zejména na 3D laserové dálkoměry, které se stávají díky své klesající ceně dostupnějšími. Metoda pro hledání cesty, respektive metoda pro budování trojrozměrné vnitřní reprezentace pomocí 3D dálkoměru (rotující/kmitající 2D laserový dálkoměr) byla představena v [43]. Rovněž často se lze setkat s algoritmy agregující data z více 2D laserových dálkoměrů současně, například viz [44]. Mnohé přístupy také uplatňují kombinaci Kalmanova filtru a klotoidního modelu cesty pro estimaci [45, 46, 47].

4.1.1 Předpoklady

Díky nepřesnostem v měření, jež jsou navíc umocněny otřesy robota při pohybu, je velice obtížné rekonstruovat 3D model okolí. Lze však předpokládat, s ohledem na rychlost sejmutí snímku (13,32 ms u LMS 221 při 1°), že naměřené hodnoty v jednom skenu jsou vůči sobě časově invariantní, neboli že jsou postihnuty stejnou chybou v měření způsobenou otřesem atd. Proto je jednodušší klasifikovat data bez plné zpětné vazby ve 3D prostoru. Dále zavedme několik tvrzení, o která se opírá detekční algoritmus, případně která značně ulehčují výsledné řešení, a to že:

- cesta má vysokou míru „rovinatosti“ a hodnoty z -souřadnice bodů cesty se blíží nule,
- cesta má nulový nebo velmi malý náklon vůči xy -rovině robota,
- cesta je nejméně tak široká jako je šířka robota,
- na počátku běhu se robot nachází na cestě,

- pozice souřadného systému robota se mění vůči souřadnému systému prostředí jen v xy -rovině¹⁴.

4.1.2 Invalidní data

V případě, že LIDAR není schopen detekovat povrch, vrací maximální vzdálenost. Tato situace nastává například na vodní hladině, kdy dochází k lomu světla a ne k jeho odrazu zpět do senzoru. Podobně se může chovat i mokrá vozovka. Množina invalidních dat je definována jako:

$$M_I = \{i \mid d_i \geq d_{inf} \forall i \in [0..100]\} \quad (6)$$

kde d_{inf} označuje maximální akceptovatelnou vzdálenost pro validní data ($d_{inf} = 30$ m).

4.1.3 Surová data

Počet dat zahrnutých v jednom skenu a pozorovací úhel stanovuje aktivní provozní mód LIDARu¹⁵. Každý sken obsahuje pole naměřených vzdáleností $d_i, i \in [0..100]$, kde hodnota indexu i implikuje úhel $\theta_i = \text{deg2rad}(i - 50)$, ve kterém byl údaj naměřen. Získané hodnoty mohou být transformovány do kartézského souřadného systému robota (matematický model LIDARu), pakliže známe výšku umístění senzoru h_L a úhel Θ_L , který svírá s rovinou, podle vztahu (7).

$$P_i = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = \begin{bmatrix} d_i \cos(\Theta_L) \cos(\theta_i) \\ d_i \sin(\theta_i) \\ h_L - d_i \sin(\Theta_L) \cos(\theta_i) \end{bmatrix} \quad (7)$$

4.1.4 Výpočet vhodných příznaků

Sjízdňý povrch je význačný svojí rovinností a malým sklonem. Zároveň musí ležet v odpovídající vzdálenosti dané umístěním senzoru, neboli jeho výška se musí blížit k nule v souřadném systému robota.

Příznak výšky

Výpočet příznaku výšky je triviální, odpovídá souřadnici z_i zkoumaného bodu.

$$\text{height}(P_i) = [0, 0, 1] \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} \quad (8)$$

Příznak hrboлатosti a sklonu

Pro výpočet příznaku hrboлатosti a sklonu nejprve stanovme okolí bodu P_i jako množinu bodů O_{pi} ¹⁶ na oboru hodnot přirozených čísel indexu k , pro kterou platí:

$$O_{pi} = \{k \mid k \in \langle i-2, i+2 \rangle \wedge k \notin M_I\}, |O_{pi}| \geq 3 \quad (9)$$

¹⁴ Zjednodušuje transformaci do kartézského souřadného systému prostředí.

¹⁵ Dále předpokládáme, že LIDAR běží v módu, kdy je pozorovací úhel 100° a rozlišovací schopnost 1° .

¹⁶ Velikost okolí byla stanovena empiricky.

Lineární regresí bodů z množiny O_{pi} v xy -rovině, například pomocí metody nejmenších čtverců (LMS), zjistíme koeficienty a a c rovnice přímky $x = ay + c$. Příznak hrbatosti bodu P_i pak vypočteme jako odmocninu součtu druhých mocnin odchylek od lineární regrese okolí (10) a sklon jako absolutní hodnotu lineárního koeficientu a rovnice přímky (11).

$$roughness(P_i) = \sqrt{\sum_{k \in O_{P_i}} (x_k - a_i y_k - b_i)^2} \quad (10)$$

$$slope(P_i) = |a_i| \quad (11)$$

4.1.5 Sjízdné a nesjízdné segmenty

S využitím uvedených příznaků definujeme sjízdný segment $traversable_{nm}$ jako množinu sousedících bodů P_i , pro než platí:

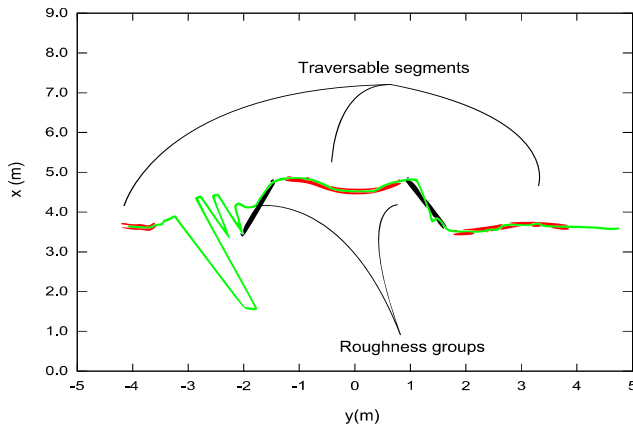
$$traversable_{nm} = \left\{ P_i \left| \begin{array}{l} \forall i \in \langle n, m \rangle. \\ roughness(P_i) < R_{sup} \\ \wedge |height(P_i)| < H_{sup} \\ \wedge slope(P_i) < S_{sup} \end{array} \right. \wedge |y_n - y_m| \geq width_{car} \right\} \quad (12)$$

kde R_{sup} je konstanta maximální povolené hrbatosti okolí bodu sjízdného segmentu ($R_{sup} = 0,7$), H_{sup} představuje maximální absolutní hodnotu výšky bodu sjízdného segmentu ($H_{sup} = 20$ cm), S_{sup} maximální sklon ($S_{sup} = 0,7$) a $width_{car}$ značí minimální šířku sjízdného segmentu v y -ose ($width_{car} = 70$ cm).

Nesjízdný segment $untraversable_{nm}$ je opět množinou sousedících bodů P_i , tentokrát však bez omezení na velikost segmentu v ose y , tedy:

$$untraversable_{nm} = \left\{ P_i \left| \begin{array}{l} \forall i \in \langle n, m \rangle. \\ roughness(P_i) > \alpha R_{sup} \\ \vee |height(P_i)| > \beta H_{sup} \\ \vee slope(P_i) > \gamma S_{sup} \end{array} \right. \right\} \quad (13)$$

kde α , β a γ jsou vhodně zvolené konstanty ($\alpha = 2,3$; $\beta = 1,3$; $\gamma = 2,0$).



Obrázek 10: Zleva ukázka detekce sjízdných segmentů, promítnutí klasifikovaných bodů do obrazu z kamery.

Poznamenejme, že veškeré konstanty uvedené v této kapitole byly zvoleny empiricky tak, aby bylo dosaženo co nejpřesnějších výsledků detekce při zachování dostatečné generality pro různé povrchové úpravy cest. Obrázek č.10 ilustruje výsledek procesu detekce sjízdných a nesjízdných segmentů.

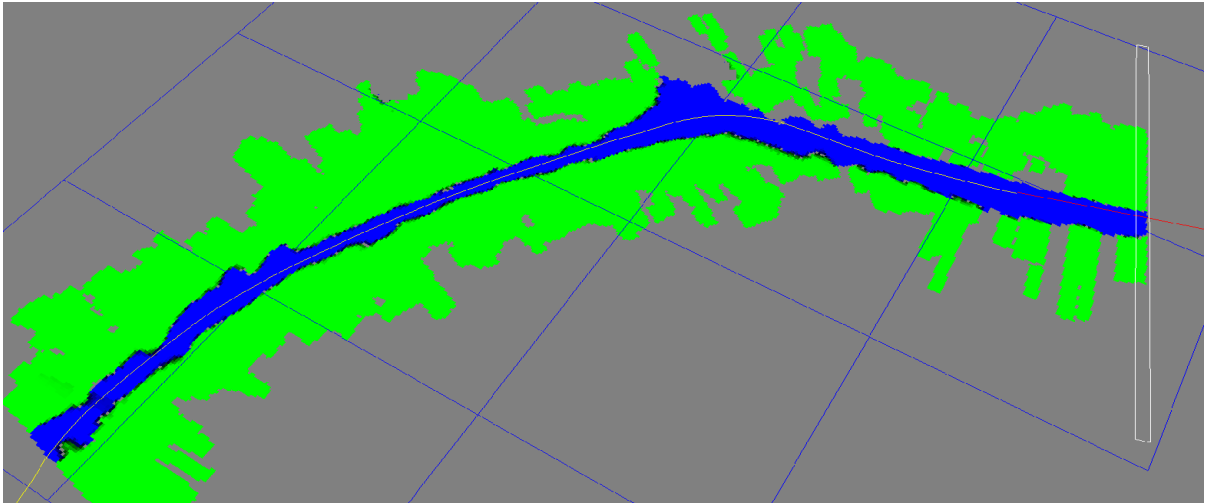
4.1.6 Detekce cesty

Ke zvýšení úspěšnosti detekce cesty a odstranění odlehlých sjízdných segmentů z předchozího kroku, je cesta chápána jako prostorově navazující sekvence sjízdných segmentů, jež jsou dostupné z aktuální či dřívější pozice robota.

Více formálně můžeme cestu popsat následovně. Necht' množina *traversableSegments* obsahuje všechny sjízdné segmenty v aktuálním snímku a necht' množina *roadSegments(t - 1)* značí segmenty cesty v čase *t - 1*, pak množina segmentů cesty v čase *t* je dána:

$$roadSegments(t) = \left\{ traversable_{nm} \left| \begin{array}{l} traversable_{nm} \in traversableSegments \\ \wedge \exists roadSeg \in roadSegments(t-1). \\ overlap(traversable_{nm}, roadSeg, \delta) \end{array} \right. \right\}, \quad (14)$$

kde funkce *overlap(segment, segment, δ)* vrací hodnotu *true*, pokud mezi segmenty existuje prostorové překrytí na ose *y* větší než δ , jinak vrací *false*. Hodnota δ byla stanovena a posteriori na 30 cm. Jestliže množina segmentů cesty neobsahuje žádný prvek, potom se jako segment cesty z množiny sjízdných segmentů volí právě ten segment, jehož střed je nejbližší počátku lokálního souřadného systému robota.



Obrázek 11: Ukázka detekce cesty. Modrá barva značí cestu, zelená nesjízdný povrch a šedá neznámou oblast.

Obrázek č. 11 vyobrazuje výsledek detekce cesty z lidarových dat. Pro správné mapování úseků cesty do lokální mapy je navíc využito údajů z pozičního estimátoru robota. Šířka segmentu cesty byla stanovena tak, aby se detekované segmenty v jednotlivých krocích částečně překrývaly a docházelo k implicitnímu vyhlazování (30 cm).

4.2 Detekce cesty pomocí jedné kamery

Variabilita prostředí, proměnné osvětlení, šum, měkké a tvrdé stíny, ztráta informace o hloubce, závislost na kontextu a křížovatky jsou jen krátkým výčtem obtíží v netriviálním segmentačním problému, kterým je rozpoznávání cesty z obrazových dat. Speciálně nepredikovatelně proměnné světelné podmínky ve venkovním prostředí a ztráta třetího rozměru činí tento úkol extrémně obtížným.

Existuje nespočetně různých řešení založených na shlukových metodách (*mean-shift* algoritmus, [48]), metodách regionálního růstu (invariantní model vůči stínům, [49]), metodách vycházejících z detekce hran (aktivní kontury, [50]), hybridních metodách atd. Většina těchto přístupů využívá apriorních znalostí, mezi něž patří šířka cesty, přibližná poloha a tvar (aproximace polynomem, [13]), texturní a barevné vlastnosti (neuronové sítě, [51]), planární charakter cesty, vodící čáry u strukturovaných cest ([52]) apod.

4.2.1 Model kamery

Pro zpracování obrazu, agregaci senzorových dat, musí být známa „skutečná“ poloha bodů na snímku v reálném světě, neboli musí existovat matematický model, který popisuje transformaci 3D systému na 2D systém. Modelu definující tuto transformaci se říká model kamery.

Mezi nejznámější a nejjednodušší model kamery patří dírkový model kamery (*pinhole camera model*), jenž využívá perspektivní projekce. Existují různé úpravy tohoto modelu postihující charakteristiku snímání obrazu konkrétním typem snímače, např. *CCD camera model*. Ve své základní podobě je dírkový model kamery matematicky definován rovnicí:

$$m' = K * [R|t] * M' , \text{ specifickěji:} \quad (15)$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} , \quad (16)$$

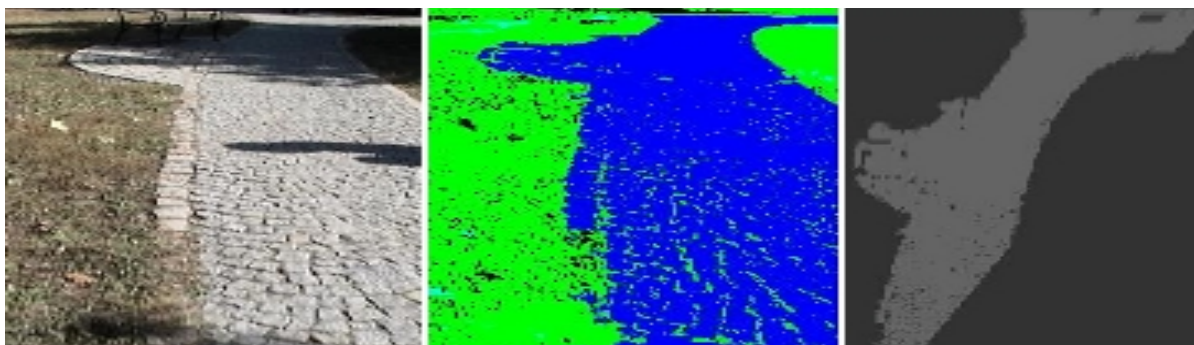
kde (X, Y, Z) jsou světové souřadnice bodu ve 3D prostoru, (x, y) jsou souřadnice projekce bodu v pixelech, K reprezentuje kalibrační matici (vnitřní parametry kamery), R je maticí rotace a t vektor posunutí (vnější parametry kamery). Dále c_x, c_y jsou souřadnice středu obrazu a f_x, f_y představují ohniskové vzdálenosti v pixelech.

Do vnitřních parametrů kamery lze rovněž zařadit i koeficienty postihující deformace způsobené zakřivením optických čoček snímačů. Základní deformací je radiální zakřivení evokované tvarem čočky, jež bývá modelováno Taylorovým rozvojem. Více o této problematice, včetně metod kalibrace, lze nalézt například v [53].

4.2.2 Segmentační algoritmus

S ohledem na rychlostní požadavky, proměnné světelné podmínky, měkké a tvrdé stíny byl navrhnout relativně triviální segmentační alg. postavený na metodě regionálního růstu a lokální klasifikaci bodů. I přes svoji jednoduchost vykazuje tento přístup vynikající výsledky, neboť je optimalizován do konkrétního prostředí a není výpočetně náročný. Zároveň postihuje nedostatek jinak vysoce spolehlivé detekce cesty z lidarových dat, a to nesprávnou klasifikaci posekaných travnatých ploch.

Proces segmentace lze rozdělit do tří kroků. V prvním kroku se provede lokální klasifikace bodů na základě barevné informace a aplikují se morfologické operace na odstranění „defektů“. Ve druhém kroku se uplatní proces narůstání oblastí dle agregačního kritéria pro segmentaci cesty. Nakonec se detekují nesjízdné segmenty představující zatravněné oblasti pomocí integrálního obrazu. Obrázek č. 12 ilustruje výsledek segmentačního procesu na snímku s ostrými stíny, a to včetně převodu do ptačího pohledu.



Obrázek 12: Výsledek segmentace cesty na snímku s tvrdými stíny a jeho transformace do ptačího pohledu.

Předpoklady

Opět zavedme několik předpokladů, podobně jako ve zpracování dat z laserového dálkoměru, nad nimiž je vybudována metoda rozpoznávání cesty z obrazu, a to že:

- kamera snímá povrch cesty, kterou lze aproximovat rovinou,
- cesta je prostorově souvislá množina bodů s charakteristickou barvou,
- robot se nachází na cestě a relativně blízko středu cesty.

Tvrzení, že povrch snímáný kamerou má rovinný charakter, zjednodušuje matematický model kamery a *de facto* dovoluje ustanovit jednoznačné mapování mezi skutečným prostředím a jeho abstraktním modelem (2D mapa) pomocí homogení inverzní transformace vůči modelu.

Lokální klasifikace

Lokální klasifikace probíhá s využitím poměrů jednotlivých barevných složek nad RGB barevným modelem a rozděluje body do tří tříd, a to na body, které mohou být potenciálně cestou, na body, jež označují travnatou plochu a na body nesjízdných úseků. Konkrétní kritéria jsou definována vztahem¹⁷:

$$class(x, y) = \begin{cases} 0, & \begin{cases} r < 35 \wedge g < 35 \wedge b < 35, \\ r > 240 \wedge g > 240 \wedge b > 240, \\ g > b \wedge r > b \wedge b < 130, \\ g/r < 0,7 \end{cases} \\ 1, & g > b \wedge g > r \wedge g/b > 1,1 \\ 2, & otherwise \end{cases} \quad (17)$$

¹⁷ Konstanty byly zjištěny experimentálně.

přičemž 0 značí nesjízdný bod, 1 bod trávy a 2 potenciální bod cesty. Nad výsledky klasifikace se provede morfologická operace *CLOSE* s jádrem obdélníkového tvaru o velikosti 3x7 pixelů pro eliminaci drobných souvislých chyb, které by mohly vést ke zcela chybným závěrům v dalším kroku (body typu 0 a 1 mají při této operaci stejnou sémantiku).

Segmentace cesty

Segmentační algoritmus narůstání oblastí potřebuje ustanovit inicializační body, kterým se říká semínka. Pozice semínek může být na základě předpokladu o rozložení cesty v obraze stanovena fixně do středu spodní části obrazu. Počet semínek by měl být adekvátní profilu cesty, neboť při malém množství se může stát, že všechna spadnou do nesjízdných ploch. AgregáčnÍ kritérium semínkového algoritmu je stanoveno takto:

$$road(x, y) = \begin{cases} true, & \begin{cases} class(x, y) = 2 \wedge \exists other\ pixel(x_n, y_n) in\ 8-connected \\ neighborhood, where\ class(x_n, y_n) = 2 \\ \wedge diss((x_n, y_n), (x, y)) < 0,2 \end{cases} \\ false, & otherwise \end{cases} \quad (18)$$

kde $diss(p_1, p_2)$ je funkce postihující distanci v poměru barevných složek bodů a je definována jako:

$$diss(p_1, p_2) = \max \left(\frac{|r_1/g_1 - r_2/g_2|}{|r_1/b_1 - r_2/b_2|} \right) \quad (19)$$

Segmentace travnatých ploch

Travnatá plocha se vyznačuje nemalým plošným obsahem prostorově navazujících bodů s majoritním podílem zelené barvy, tedy bodů, které byly označeny v lokální klasifikaci body trávy. Podmínka plošného obsahu filtruje šum (například spadlé listí na cestě) a její splnění je možné velice snadno a efektivně ověřovat nad datovou strukturou, které se říká integrální obrázek.

Integrální obrázek má stejné rozměry se zdrojovým obrázkem a obsahuje hodnoty, jež jsou spočteny jako součet všech hodnot bodů uvnitř obdélníku ohraničeného levým horním rohem obrázku a aktuálním bodem. Tato struktura tak umožňuje získat sumu pixelů v kterékoliv obdélníkové oblasti pouze čtyřmi přístupy do paměti.

Samotná detekce nesjízdných ploch probíhá nad upraveným výstupem z algoritmu lokální klasifikace, který se dále transformuje do ptačího pohledu, aby se odstranily problémy s nelinearitou perspektivní projekce. Úprava výstupu spočívá v přeznačení potenciálních bodů cesty na body nesjízdné. Výpočet nad integrálním obrázkem pak bude reflektovat počet bodů trávy na zadané ploše. Klasifikační funkce je dána tímto předpisem:

$$grass(x, y) = \begin{cases} true, & \left(\frac{I(x+\Delta, y+\Delta) - I(x+\Delta, y-\Delta)}{-I(x-\Delta, y+\Delta) + I(x-\Delta, y-\Delta)} \right) > \alpha \\ false, & otherwise \end{cases} \quad (20)$$

kde I označuje integrální obrázek, Δ implikuje velikost čtvercového okolí ($\Delta = 2$) a α představuje minimální počet bodů trávy uvnitř oblasti ($\alpha = 20$).

4.3 Shrnutí

Detekce sjízdného povrchu, respektive cest v parkovém prostředí se řadí k nejnáročnějším pasážím kognitivního systému autonomního robota, přičemž robustnost, přesnost a spolehlivost zvolených klasifikačních metod implikuje úspěšnost celé platformy. Bez dostatečné úrovně rozpoznání scény a jistoty rozpoznávacích metod je delší pohyb v dynamickém prostředí takřka nemyslitelný.

V této kapitole byly představeny dvě konkurenční metody pro detekci sjízdného povrchu, a to pomocí barevné kamery a laserového dálkoměru. Detekční metoda uplatňující data z laserového dálkoměru má vysokou spolehlivost v případě detekce překážek či nesjízdných ploch a je nezávislá na denním osvětlení (možná jízda v noci). Naopak klasifikace obrazových dat přináší relativně bezpečné rozpoznání travnatých oblastí a tím kompenzuje možná selhání detekce z lidarových dat, kde jsou právě zatravněné plochy od úseků cesty špatně rozlišitelné. Obrovskou překážku spolehlivé segmentace cesty z barevných snímků kamery představují měkké a tvrdé stíny, rozmanitost světelných podmínek a s tím související barevné podání (např. listy stromů se chovají jako barevné filtry).

5 Lokální navigace

Pod pojmem lokální navigace se skrývá problematika řešení odhadu relativní pozice robota vůči okolním objektům a způsob interakce s blízkým prostředím, jmenovitě detekce cesty, zabránění kolizím při jízdě, rozpoznání křižovatek atp. Dekompozice navigace dle jejího měřítka na lokální a globální úroveň zjednodušuje volbu vhodné vnitřní reprezentace a výběr metody pro estimaci pozice robota.

Cílem této kapitoly je uvést zvolený model vnitřní reprezentace prostředí včetně metod fúze výsledků detekčních algoritmů sjízdného povrchu, jenž bude svým detailem a paměťovou náročností vhodný pro budoucí řídicí a plánovací algoritmy, a dále představit lokalizační metodu založenou na Kalmanově filtraci. Avšak při popisu Kalmanovy filtrace se předpokládá alespoň zběžná znalost problematiky toho stavového estimátoru (viz např. [34, 54, 55]). Podkapitola *Estimace pozice* v obecné rovině shrnuje základní aspekty nejznámějších variant filtru, tj. LKF (*Linear Kalman Filter*) a EKF (*Extended Kalman Filter*), a nakonec podrobněji rozebírá tzv. *Unscented Kalman Filtr (UKF)*, který byl v odhadu polohy robota uplatněn.

5.1 Vnitřní reprezentace prostředí

Nutnou podmínkou k řízení robota na vyšší úrovni kognitivních procesů je dostatečně detailní model okolního prostředí, často nazývaný mapa. Ruční tvorba map představuje zdoluhavý proces, navíc mapy vhodné pro člověka nemusí být použitelné pro řídicí systém robota. V dynamických prostředích je nevhodnost dopředné tvorby mapy vysokého detailu, ať už robotem nebo člověkem, znatelnější, jelikož získaná mapa bude mít z principu krátkodobou platnost. Proto vnitřní model okolí si robot buduje během běhu samostatně, a to s využitím dostupného senzorického systému a navržených detekčních metod sjízdného povrchu. S ohledem na úroveň potřebného detailu, možnost agregace výsledků detekčních algoritmů sjízdného povrchu, vysokou stabilitu a dostatečnou přesnost byla zvolena mřížka obsazenosti jako abstraktní model vnitřní reprezentace prostředí pro lokální navigaci.

Implementovaná mřížka obsazenosti má podobu dvourozměrného pole buněk, do kterého se mapují výsledky z detekčních algoritmů sjízdného povrchu. Hodnoty buněk vyjadřují míru jistoty ve sjízdnost daného úseku (tzv. věrohodnostní přístup). Dvourozměrná aproximace reálného světa vychází z předpokladu o rovinném charakteru cesty a je přirozeným důsledkem možnosti senzorického subsystému robota. Nadto by případný trojrozměrný model signifikantně zvýšil složitost algoritmů operujících nad datovou strukturou a kladl výrazně vyšší nároky nejen na přesnost odhadu polohy robota, ale i na samotné předzpracování a interpretaci senzorických měření. Klíčovým aspektem při tvorbě mapy obsazenosti je metodika slučování dat senzorických měření, respektive detekčních algoritmů sjízdného povrchu.

5.1.1 Slučování dat

Kvalita informace získaná ze senzorů nejen jednoho typu, ale i různých typů, se může velmi lišit a není snadné říci, která data jsou přesnější a jaká interpretace je spolehlivější. Proto se prakticky vždy využívá možnosti agregace informace z více zdrojů, přičemž konečná interpretace postihuje charakteristiku chyb detekčních metod (senzorů).

Existuje více způsobů integrace dat na různé úrovni zpracování, např. fúze nízké úrovně (angl. *low level fusion*), fúze na vyšší úrovni (strukturální), asociační fúze (angl. *track-based fusion*), víceúrovňová fúze (angl. *multi-level fusion*) atd., viz [56]. Slučování dat využívající modelu prostředí, konkrétně mřížku obsazenosti se řadí do kategorie slučování založené na mřížce, angl. *grid-based fusion*. V základní podobě mřížka obsazenosti předpokládá pouze jeden zdroj vstupních informací, avšak lze ji snadno rozšířit i pro více typů zdrojů informací za splnění podmínky porovnatelnosti měř modelů jednotlivých senzorů [17]. Navzdory tradičním přístupům stavějícím na Bayesově teorii nebo teorii fuzzy množin bylo experimentálně zjištěno a demonstrováno, že alternativní jednoduché metody mohou poskytovat obdobné výsledky s nižší výpočetní náročností a nabídnout tak bezpečnou agregaci dat v reálném čase [57]. Ačkoliv samotné slučování může mít vícero úrovní a podob, závěrečná fúze vždy zohledňuje aplikaci robota, tj. v případě robotického vozidla bezpečnou detekci překážek a nesjízdných segmentů (angl. *collision avoidance*).

Ve shodě s teorií mřížek obsazenosti musí ke každému sémantickému významu hodnot věrohodnosti existovat separátní model okolí, což nejen značně zvyšuje paměťovou náročnost, ale i činí složitějšími samotné algoritmy pro tvorbu modelu a plánování pohybu. V situacích, kdy je navíc tato informace sémanticky výlučná (např. překážka versus sjízdňý povrch), avšak je ji vhodné modelovat odděleně, lze bez ztráty obecnosti v rámci jedné mřížky reprezentovat dvojí informaci a rozlišovat ji znaménkem hodnoty, přičemž nulová hodnota bude značit neznalost daného segmentu. Dále definujeme význam hodnoty buňky v souvislosti řízení robotického vozidla. Kladná hodnota indikuje sjízdňý úsek (větší hodnota reflektuje vyšší míru věrohodnosti informace). Komplementárně tomu bude u záporných hodnot, jež označují překážku.

Aby bylo možné postihnout počet měření zasahujících do konkrétní buňky při výpočtu hodnoty, je buňka implementována jako struktura obsahující čítač zásahů detekčních algoritmů do této oblasti, čítač počtu detekcí cesty a čítač počtu detekcí překážek, a to pro každý detektor separátně. Výpočet hodnoty buňky definuje rovnice:

$$cell(i) = \begin{cases} -1, \exists j \in detectors, cell(i).obstacle_j > 0 \\ \sum_{j \in detectors} trust(i, j) \cdot \delta_j, otherwise \end{cases} \quad (21)$$

kde $obstacle_j$ značí počet detekcí překážek detektoru j buňky i . δ_j má význam váhy a platí, že:

$$\sum_{j \in detectors} \delta_j = 1 \quad (22)$$

Funkce $trust(i, j)$ vrací věrohodnost konkrétního detektoru j s ohledem na počet detekcí cesty $road$ a počet zásahů detekcí $hits$ buňky i a je dána předpisem:

$$trust(i, j) = \frac{cell(i).road_j}{cell(i).hits_j} \quad (23)$$

Neboli buňka nabývá hodnoty mínus jedna, pakliže existuje alespoň jediný detektor, který detekoval překážku, v opačném případě se počítá vážená suma z jednotlivých detekcí detektorů. U detekčních metod se tudíž očekává jejich absolutní spolehlivost v rozpoznání nesjízdných úseků.

5.1.2 Dynamické prostředí

Postihnout dynamiku prostředí na úrovni detekce a sledování pohyblivých objektů, včetně následné predikce jejich trajektorie pro plánování svého pohybu, se řadí k složitějším úlohám, jejichž řešení má

výrazný dopad jak na úroveň rozpoznávání okolní scény pomocí dostupného senzorického subsystému, tak i na samotnou vnitřní reprezentaci okolí. Proto bylo zavedeno značné zjednodušení nabízející částečné východisko z této nesnáze, jež však navíc přináší výhody v otázce přesnosti estimace polohy pro tvorbu lokální mapy, a to model zapomínání projetych úseků.

Zapomínání částí mapy světa, které robot již navštívil, respektive hodnot, jež nejsou časově aktuální, má řadu pozitivních dopadů i v ostatních aspektech lokální navigace. Například díky zapomínání výsledků detekce detekčních algoritmů není kladen požadavek na vysokou přesnost estimace polohy robota pro tvorbu lokální mapy, neboť akumulovaná chyba se nestává kritickou. Dále zapomínání přirozeně ohraničuje paměťové a výpočetní nároky tvorby mřížky obsazenosti a tím pádem i plánovacích algoritmů operujících nad touto datovou strukturou. Nevýhodou je samotný přístup, kdy při vysoké dynamice prostředí může dojít ke znemožnění pohybu nebo k opomenutí některých odboček. Nicméně i přes tyto handicapy se jedná o relativně spolehlivé a účinné řešení pro parková prostředí. Konkrétně jsou zapomínány a z mřížky obsazenosti odstraňovány údaje starší než třicet vteřin¹⁸.

5.2 Estimace pozice

Díky neexistenci přímé vazby mezi globální a lokální polohou¹⁹ robota nevzniká komplikace v lokální navigaci s prvotním určením pozice na začátku běhu, neboť tato pozice může být stanovena libovolně. Estimaci polohy lze tudíž chápat jako nejjednodušší lokalizační problém, a to trasování pozice (lokální lokalizaci). Korektní mapování výsledků detekce sjízdného povrchu do interní mapy na lokální poziční estimátor klade tyto požadavky:

- kontinualitu změn odhadu pozice a jejich pohybovou příčinnost,
- vysokou přesnost na krátké vzdálenosti (relativní přesnost),
- schopnost zpracovat redundantní a asynchronní vstupy,
- dostatečnou frekvenci výpočtu,
- filtrování odlehklých hodnot,
- maximální stabilitu.

Jako vhodná zdrojová data pro estimaci se z těchto důvodů jeví výstupy senzorů přímo snímající změnu polohy a polohu robota, jež mají vysokou citlivost, přesnost a vzorkovací frekvenci a de facto představují spojitě měřící systémy. Zároveň jejich kombinace musí umožnit, ať už přímo nebo nepřímě, robota prostorově lokalizovat po celou délku jeho běhu prostředím. Proto byla zvolena filtrovaná agregace dat z inerciální jednotky, optického rotačního enkodéru a kompasu pomocí Kalmanova filtru.

5.2.1 Kalmanova filtrace

Kalmanův filtr je rekurzivní filtr, který odhaduje stavy dynamického systému dle zašumělých měření. Filtr má mnoho variant pro spojitý i diskrétní čas, pro lineární a nelineární systémy a velice rychle se ujal v navigační oblasti jako řešení k dříve nepoddajným problémům. Praktická aplikace filtru

¹⁸ Jednoduše jsou dekrementovány jednotlivé čítače při každém dalším příchodu měření po inicializační fázi.

¹⁹ Polohou nebo pozicí robota rozumíme stavový vektor určující jeho konfiguraci v prostoru lokální mapy.

vychází ze schopnosti estimovat stav systému (v našem případě stav robota) na základě měření, jež jsou [54]:

- nekompletní (mající vztah ne ke všem proměnným systému),
- neperiodicky dostupná,
- nepřímo kvantizující,
- nepřesná.

Podstata adaptivní Kalmanovy filtrace spočívá v modelování systému jakožto dynamického systému, jenž je buzen šumem a jehož vstupy jsou rovněž zašumělé, pomocí modelu procesu a modelu měření. Myšlenka vychází ze základního principu estimační teorie, respektive z ideje, že pokud známe statistickou charakteristiku šumu, tak lze zkonstruovat optimální estimátor, a to i bez znalosti samotných chyb.

V případě diskrétního času je model procesu definován následně. Necht' x_k označuje stav systému v čase k a u_k a z_k jsou kontrolní vstupy a pozorování v čase k . Dále předpokládejme, že dynamický systém se vyvíjí ve shodě s přechodovou funkcí stavu g , pak stavový model²⁰ má tvar:

$$x_k = g(u_{k-1}, x_{k-1}) + \varepsilon_k, \quad (24)$$

přičemž ε_k označuje procesní šum s nulovou střední hodnotou a Gaussovým rozdělením a kovarianční maticí Q_k , neboli $\varepsilon_k \sim N(0, Q_k)$. Obdobně v případě měřicího modelu necht' pozorování m_k je funkcí h aktuálního stavu s přidaným šumem s Gaussovým rozdělením a kovarianční maticí R_k v čase k , neboli:

$$m_k = h(x_k) + \delta_k. \quad (25)$$

Kalmanova filtrace cílí na časovou propagaci odhadu stavu x a kovarianční matici chyb tohoto odhadu Σ počítáním časového a datového kroku. V časovém kroku se predikuje budoucí stav x a jeho kovarianční matice Σ na základě aktuálních hodnot. Datový krok znamená filtraci a aplikuje se, pakliže jsou k dispozici nová měření. Během procesu filtrace lze měření agregovat. Právě možnost fúze informací z více zdrojů ke zlepšení přesnosti odhadu je jedna ze stěžejních vlastností tohoto estimátoru.

Jestliže model systému i model měření jsou lineární a platí-li všechna kritéria pro estimaci pomocí metody minimalizace středně kvadratické hodnoty chyby (MMSE, viz [58]), pak může být použit LKF. Bohužel u většiny reálných modelů bývají funkce g a h nelineární, a proto aplikace LKF je buď obtížná nebo zcela nevhodná.

Pravděpodobně mezi nejznámější estimátory nelineárních systémů patří EKF. EKF jednoduše linearizuje nelinearitu v okolí aktuálního stavu, jde proto o lokální metodu. Algoritmus EKF se v důsledku od LKF liší jen v popisu modelu měření a modelu procesu pomocí parciálních derivací nelineárních funkcí dle vnitřního stavu systému, tj. jakobiány, jež představují reprezentaci nejlepší lineární aproximace k diferencovatelné funkci poblíž zkoumaného bodu. Přesto EKF má několik stinných stránek [59]:

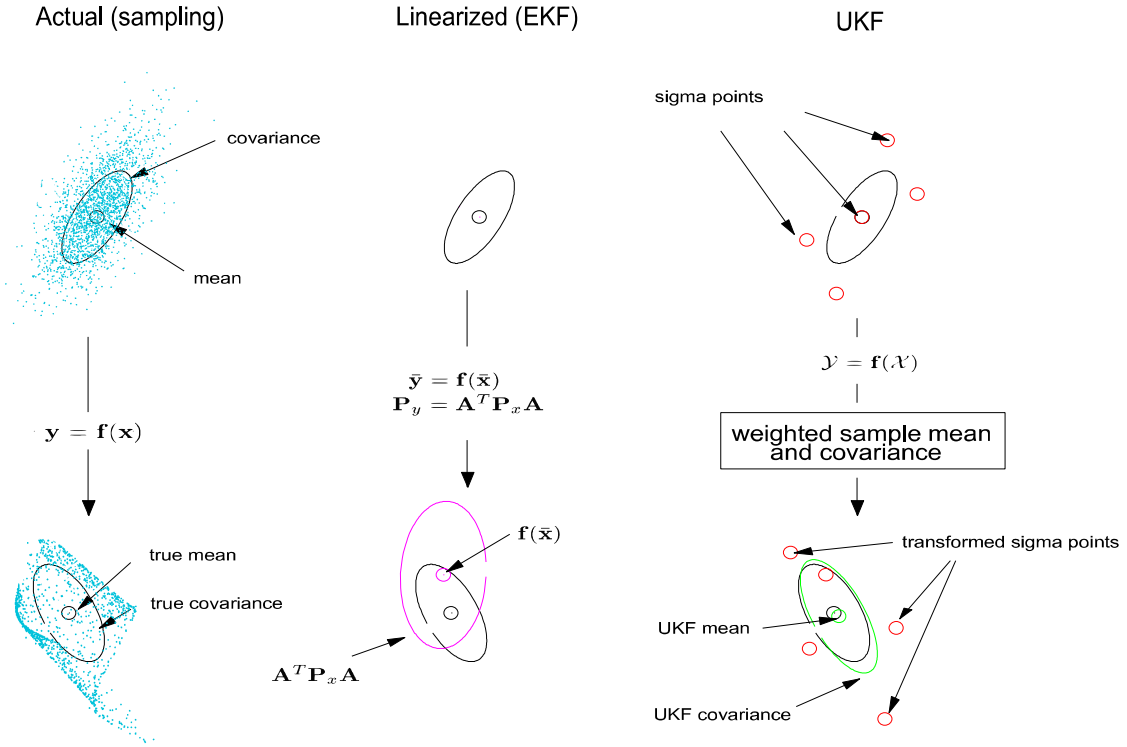
- linearizace může způsobit vysokou nestabilitu filtru, jestliže předpoklad o lokální linearitě není naplněn (odhady mají pouze lokální platnost),
- jakobián musí být odvoditelný, avšak jsou případy, kdy tomu tak není (např. skokově lineární systémy),

²⁰ Model procesu a stavový model znamená totéž. Podobně pozorovací model je akronymem k modelu měření.

- odvození jakobiánu nebývá ve většině aplikací snadné a často vede k implementačním obtížím (lidský faktor).

Unscented Kalman Filter

Unscented Kalman Filter (UKF) byl navržen v roce 1997 jako alternativa k EKF bez derivací. Fundamentální difference mezi EKF a UKF vyplývá ve způsobu propagace náhodných proměnných s Gaussovým rozdělením (*Gaussian random variables*, GRV) skrze nelineární systém. V případě EKF jsou GRV propagována analyticky linearizací systému. UKF staví na tzv. transformaci charakteristických bodů (angl. *unscented transform*).



Obrázek 13: Srovnání propagace střední hodnoty v EKF a UKF. Obrázek převzat z [60].

Transformace charakteristických bodů označuje novou metodu pro výpočet statistických charakteristik náhodné proměnné, jež podstupuje nelineární transformaci. Metoda je založena na intuici, že lze snadněji aproximovat Gaussovo rozdělení, než-li aproximovat libovolnou nelineární funkci [61]. Předpokládejme n -dimensionální náhodnou proměnnou x s Gaussovým rozdělením se střední hodnotou μ^{21} a kovarianční maticí Σ . Množina transformačních bodů (tzv. sigma bodů) je vybrána tak, aby jejich střední hodnota a kovarianční matice odpovídala μ , respektive Σ . Na první pohled se transformace jeví jako obdoba Monte Carlo metody, avšak signifikantní difference spočívá právě ve způsobu volby sigma bodů, který je u této transformace striktně deterministický. Konkrétně se $2n + 1$ sigma bodů $\chi^{[i]}$ s příslušnými váhami vybere podle následujícího pravidla (viz [33, 62]):

$$\begin{aligned} \chi^{[0]} &= \mu \\ \chi^{[i]} &= \mu + (\sqrt{(n+\lambda)\Sigma})_i \quad i \in \langle 1, \dots, n \rangle \\ \chi^{[i]} &= \mu - (\sqrt{(n+\lambda)\Sigma})_{i-n} \quad i \in \langle n+1, \dots, 2n \rangle \end{aligned} \quad (26)$$

21 Míněno jako marginální střední hodnota (tj. pro vícerozměrné veličiny).

přičemž $(\sqrt{(n+\lambda)\Sigma_x})_i$ označuje i -tý sloupec odmocniny matice a λ tzv. škálovací parametr. Škálovací parametr se zpravidla definuje jako $\lambda = \alpha^2(n+\kappa) - n$, kde α určuje, jak dalece budou sigma body rozptýleny od střední hodnoty μ a obvykle nabývá malého pozitivního čísla a κ představuje sekundární škálovací parametr, jenž se nejčastěji nastavuje na hodnotu nula. Vypočtené sigma body jsou transformovány skrze potenciálně nelineární funkci f , aby se zjistil vliv transformace na tvar Gaussova rozdělení:

$$\Psi^{[i]} = f(\chi^{[i]}) \quad (27)$$

Střední hodnota $\hat{\mu}$ a kovariance výsledku $\hat{\Sigma}$ transformovaného Gaussova rozdělení se extrahuje z přenesených sigma bodů dle:

$$\begin{aligned} \hat{\mu} &= \sum_{i=0}^{2n} w_m^{[i]} \Psi^{[i]} \\ \hat{\Sigma} &= \sum_{i=0}^{2n} w_c^{[i]} (\Psi^{[i]} - \hat{\mu})(\Psi^{[i]} - \hat{\mu})^T \end{aligned} \quad (28)$$

kde váhy $w_m^{[i]}$ a $w_c^{[i]}$ jsou nastaveny vhodně. Způsob výpočtu vah není jednotný, některé implementace kupříkladu nerozlišují váhu pro kovarianční matici $w_c^{[i]}$ a pro střední hodnotu $w_m^{[i]}$ středového sigma bodu (viz např. [63]), nicméně nejčastěji se počítají váhy následně:

$$\begin{aligned} w_m^{[0]} &= \lambda / (n + \lambda) \\ w_c^{[0]} &= \lambda / (n + \lambda) + (1 - \alpha^2 + \beta) \\ w_m^{[i]} &= w_c^{[i]} = 1 / (2(n + \lambda)) \end{aligned} \quad (29)$$

přičemž β slouží ke včlenění dřívější znalosti o distribuci x (pro Gaussovo rozdělení platí, že $\beta = 2$).

Samotná procedura Kalmanovy filtrace má dvě formy, a to rozšířenou a nerozšířenou. V rozšířeném UKF je stav a kovarianční matice rozšířena o komponenty šumu měření a procesního šumu. Rozšířený UKF dává lepší výsledky estimace v případě neaditivního šumu, nicméně za cenu vyšší výpočetní náročnosti související s nárůstem počtu stavů (velikosti vektoru; viz [36, 59]). Avšak dále předpokládáme klasickou, respektive nerozšířenou podobu UKF, jež za podmínky aditivního šumu poskytuje stejnou přesnost.

Časový krok UKF odpovídá transformaci charakteristických bodů, kde množina těchto bodů se spočte z dřívější známé střední hodnoty stavu μ_{k-1} ²² a jeho kovarianční matice Σ_{k-1} s využitím (26). Sigma body jsou následně propagovány skrze potenciálně nelineární přechodovou funkci stavu g . Z transformovaných charakteristických bodů se ustanoví nová střední hodnota stavu μ_k a jeho kovarianční matice Σ_k v souladu s (28), avšak ke kovarianční matici stavu Σ_k se ještě připočte kovarianční matice Q_k (procesní šum; zpravidla se nemění), respektive:

$$\begin{aligned} 1. \quad & \chi = (\mu_{k-1} \quad \mu_{k-1} + (\sqrt{(n+\lambda)\Sigma_{k-1}}) \quad \mu_{k-1} - (\sqrt{(n+\lambda)\Sigma_{k-1}})) \\ 2. \quad & \Psi^{[i]} = g(\chi^{[i]}), i \in \langle 0, \dots, 2n \rangle \\ 3. \quad & \mu_k = \sum_{i=0}^{2n} w_m^{[i]} \Psi^{[i]} \\ 4. \quad & \Sigma_k = \sum_{i=0}^{2n} w_c^{[i]} (\Psi^{[i]} - \mu_k)(\Psi^{[i]} - \mu_k)^T + Q_k \end{aligned} \quad (30)$$

²² Střední hodnota stavu v čase k je de facto odhad stavu v čase k .

V datovém kroku UKF se množina sigma bodů odvozených od stavového vektoru μ_{k-1} a jeho kovarianční matice Σ_{k-1} propaguje tentokrát skrze funkci měření h . Predikované měření z_k a kovarianční matice S_k jsou opět získány obdobně jako u časového kroku, přičemž ke kovarianční matici se přičte kovarianční matice R_k při měření m_k (šum měření; mění se spolu s aktuálním senzorem). Následuje výpočet vzájemné (křížové) kovarianční matice stavu a měření $\Sigma_k^{x,z}$, Kalmanova zisku K_k , nové střední hodnoty μ_k a kovariance Σ_k :

$$\begin{aligned}
1. \quad & \chi = (\mu_{k-1} \quad \mu_{k-1} + (\sqrt{(n+\lambda)\Sigma_{k-1}}) \quad \mu_{k-1} - (\sqrt{(n+\lambda)\Sigma_{k-1}})) \\
2. \quad & \Psi^{[i]} = h(\chi^{[i]}), i \in \langle 0, \dots, 2n \rangle \\
3. \quad & z_k = \sum_{i=0}^{2n} w_m^{[i]} \Psi^{[i]} \\
4. \quad & S_k = \sum_{i=0}^{2n} w_c^{[i]} (\Psi^{[i]} - z_k)(\Psi^{[i]} - z_k)^T + R_k \\
5. \quad & \Sigma_k^{x,z} = \sum_{i=0}^{2n} w_c^{[i]} (\chi^{[i]} - \mu_{k-1})(\Psi^{[i]} - z_k)^T \\
6. \quad & K_k = \Sigma_k^{x,z} S_k^{-1} \\
7. \quad & \mu_k = \mu_{k-1} + K_k (m_k - z_k) \\
8. \quad & \Sigma_k = \Sigma_{k-1} - K_k S_k K_k^T
\end{aligned} \tag{31}$$

5.2.2 Návrh filtru

Vnitřní popis systému vychází z popisu dynamického vývoje stavu systému. Podmínkou řešitelnosti úloh filtrace, predikce a interpolace, formulovaných pomocí vnitřního popisu systému, je možnost estimovat stav systému na základě pozorování vstupů a výstupů [64]. Systém tedy musí být pozorovatelný, aby mohla být aplikována Kalmanova filtrace. Specifičtěji, pro návrh filtru založeném nejen na nerozšířeném UKF je zapotřebí úplný popis systému, tj. znalost transformačních matic v rovnici měření a v rovnici stavu, stavového šumu a šumu měření, počátečních podmínek popsanych Gaussovým rozdělením se známou střední hodnotou a kovarianční maticí atd.

Model procesu

Model procesu popisuje nominální pohyb robota, neboli predikuje pohyb bez kalkulace stavových chyb a řízení. Diskrétní pohybový model robota *Karlík* se opírá o omezený pohybový model pro automobilový podvozek uvedený Brandtem a Gardner roku 1998 (viz [65]), jenž byl vybudován na těchto předpokladech:

- robot se pohybuje pouze ve směru x -osy ve smyslu jeho lokálního souřadného systému,
- robot se pohybuje po dokonale plochem povrchu.

Jinak řečeno, podmínky implikují, že robotické vozidlo nebude konat žádný pohyb v příčném směru a podél normály sjízdného povrchu. To má za následek výrazné zjednodušení pohybového modelu a v konečném důsledku i redukci pozičních chyb. Nicméně díky dostupné informaci o náklonu robota je přínosné tuto informaci do modelu procesu zahrnout, proto model procesu má

podobu nelineárního dynamického modelu se sedmiprvkovým stavovým vektorem popsaného rovnici v maticovém tvaru:

$$\begin{bmatrix} x \\ y \\ v_f \\ a_f \\ \theta \\ \psi \\ \dot{\psi} \end{bmatrix}_k = \begin{bmatrix} 1 & 0 & \cos(\theta)\sin(\psi)dt & 1/2\cos(\theta)\sin(\psi)dt^2 & 0 & 0 & 0 \\ 0 & 1 & \cos(\theta)\cos(\psi)dt & 1/2\cos(\theta)\cos(\psi)dt^2 & 0 & 0 & 0 \\ 0 & 0 & 1 & dt & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & dt \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ v_f \\ a_f \\ \theta \\ \psi \\ \dot{\psi} \end{bmatrix}_{k-1} \quad (32)$$

Dvojice x a y určuje polohu robota v souřadném systému lokální mapy, v_f rychlost ve směru jízdy, a_f zrychlení ve směru jízdy, θ náklon, ψ stáčení a $\dot{\psi}$ rychlost stáčení vozidla.

Model měření

Estimace polohy je založena na agregaci dat z rotačního enkodéru, inerciální jednotky a kompasu. Kompas představuje absolutní senzor pro měření směru vozidla a stabilizuje tak odhad stáčení počítaný inerciální jednotkou integrací úhlové rychlosti z gyroskopu. Podobně rotační enkodér zamezuje neohrazenému růstu rychlosti dle akcelerometru v IMU, ke kterému snadno dojde kvůli integraci a nepřesnostem měření. Samotné transformační matice měření dílčích senzorů jsou za předpokladu budování modelu v lokálním dvourozměrném souřadném systému, v jehož počátku se senzory nacházejí a který je umístěn v těžišti robota, triviální. Dokonce díky zvolenému stavovému vektoru v modelu procesu se jedná o lineární transformace. Rovnice (33) popisuje model měření pro přehlednost kompletně, ačkoliv data jsou zpracovávána asynchronně.

$$\begin{bmatrix} v_{enc} \\ a_{imu} \\ \theta_{imu} \\ \psi_{comp} \\ \dot{\psi}_{imu} \end{bmatrix}_k = \begin{bmatrix} v_f \\ a_f \\ \theta \\ \psi \\ \dot{\psi} \end{bmatrix}_k, \quad h(x_k) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T \cdot x_k \quad (33)$$

Model chyb

Dle teorie Kalmanovy filtrace by se měl odhad hodnoty každého prvku kovarianční matice co nejvíce blížit skutečné míře vzájemné závislosti dvou náhodných veličin. Nicméně dosáhnout takovýchto hodnot je prakticky nemožné, jelikož některé šумы mohou být korelovány či mají systematický charakter, jejichž model není znám. Proto dvojice kovariančních matic Q_k a R_k bývá diagonální, a postihuje tudíž pouze varianci (34). Prvky mimo diagonálu se u kovarianční matice odhadu stavu Σ_k vyvíjejí přirozeně při běhu filtru.

$$\begin{aligned} Q_k &= \text{diag} \left[\sigma_x^2 \quad \sigma_y^2 \quad \sigma_{v_f}^2 \quad \sigma_{a_f}^2 \quad \sigma_\theta^2 \quad \sigma_\psi^2 \quad \sigma_{\dot{\psi}}^2 \right] \\ R_k &= \text{diag} \left[\sigma_{enc}^2 \quad \sigma_{imu, acc}^2 \quad \sigma_{imu, \theta}^2 \quad \sigma_{comp}^2 \quad \sigma_{imu, \psi}^2 \right] \end{aligned} \quad (34)$$

Model nejistoty stavu modeluje nepřesnost vznikající při běhu systému ve smyčce časového kroku. Kovarianční matice Q_k se může měnit pro každý časový krok dle aktuální konfigurace systému

a postihnout tak nepřesnosti způsobené např. nerovným terénem, zanedbáním kontrolních vstupů, nepřesným modelem atp.

Model nejistoty měření je kritickým pro korektnost estimace, protože jakmile dojde byť jen ke krátkému výpadku dat ze senzorů, filtr velice rychle diverguje. Kovarianční matice R_k zpravidla postihuje práh a nestabilní citlivost senzorů, jejich opomenutou dynamiku, šum, vibrace atd.

Konkrétní konstantní hodnoty kovariančních matic Q a R byly stanoveny experimentálně dle několika testových běhů (35). Hodnoty byly optimalizovány na agregační model tak, aby systém jako celek vykazoval vysokou přesnost odhadu pozice, a nemusejí proto reflektovat skutečnost. Inicializační hodnota kovarianční matice stavu Σ_0 má podobu jednotkové matice.

$$\begin{aligned} Q &= \text{diag}[0,001 \ 0,001 \ 0,001 \ 0,001 \ 0,001 \ 0,001 \ 0,001] \\ R &= \text{diag}[0,001 \ 0,7 \ 0,1 \ 4,5 \ 0,001] \end{aligned} \quad (35)$$

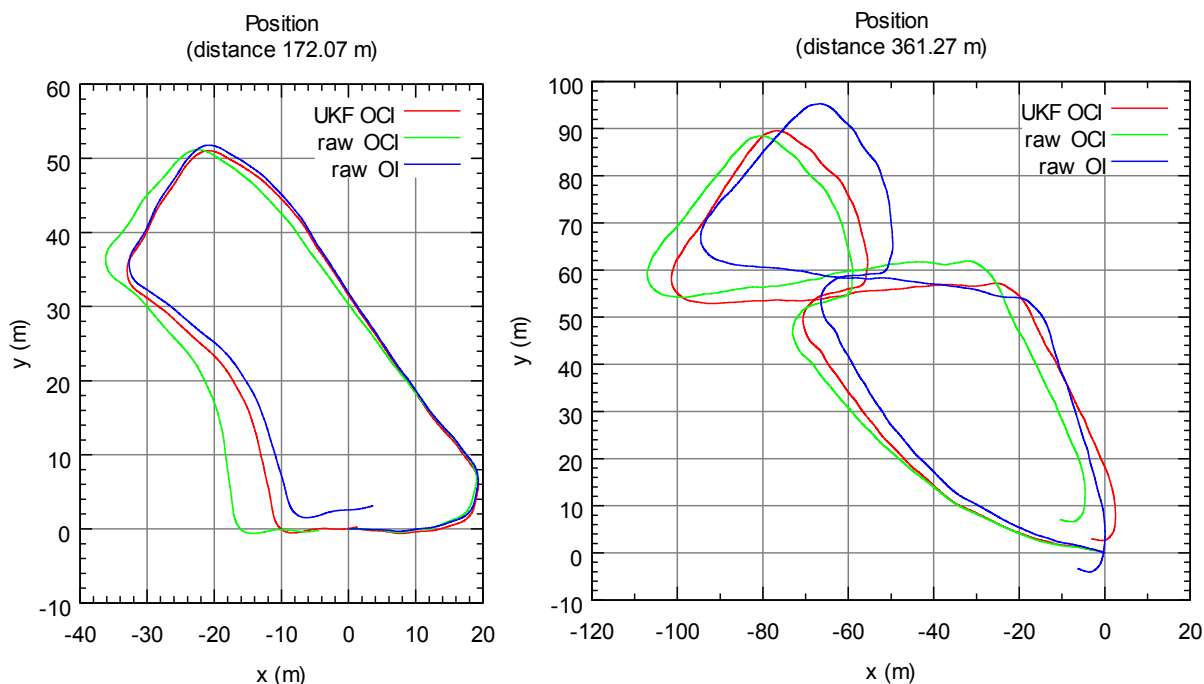
Aby bylo lépe postihnuto unášení gyroskopu, drift akcelerometru atp., je vždy při zastavení vozidla vypočtena průměrná hodnota z hodnot změřených při stání, pomocí již se další získaná měření dorovnávají tak, aby byla splněna podmínka nulové střední hodnoty šumu daného měření, a to pouhým posunem (odečtením). Právě podmínka stání vozidla je pro výpočet daných posunů klíčová, neboť zaručuje, že měřené hodnoty nemají pohybovou příčinnost.

5.2.3 Experimentální výsledky

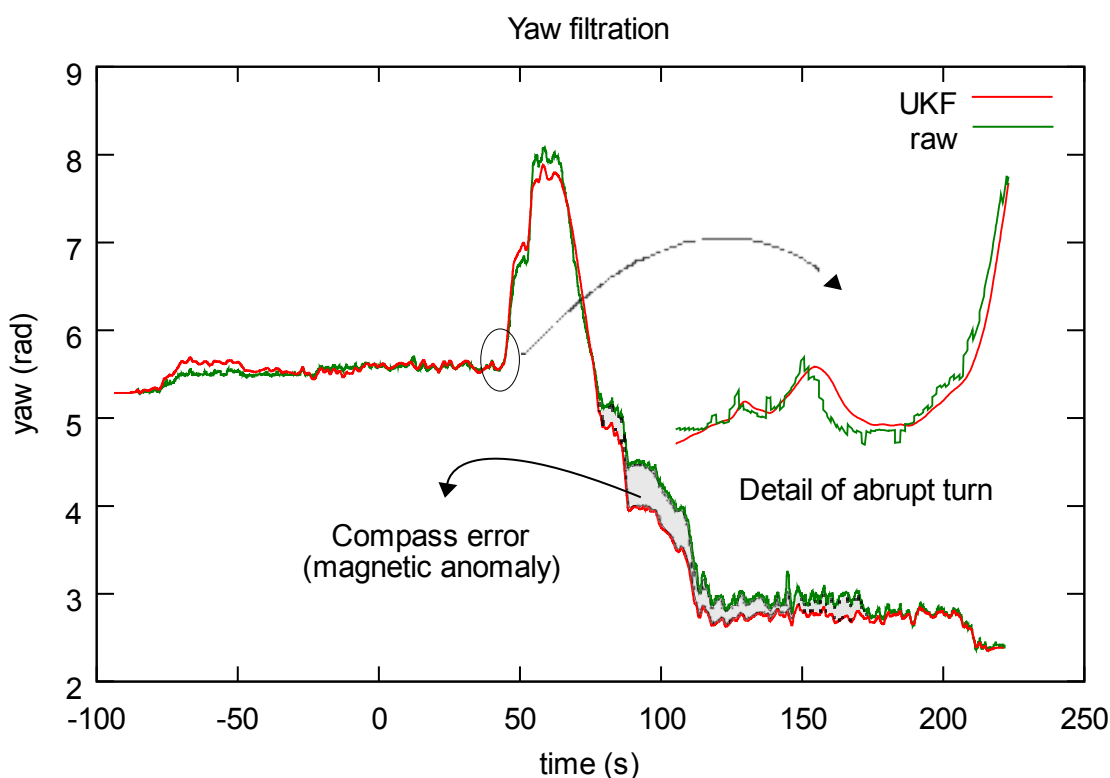
Pro otestování dopadu UKF filtrace na přesnost estimace polohy byla uskutečněna řada testovacích jízd v parku na Božetěchově ulici Brně. Na jejich základě byly následně upraveny parametry filtru, respektive kovarianční matice procesu a měření, aby se dosáhlo znatelně lepších výsledků odhadu pozice. Bohužel neznalost referenční trajektorie znemožňuje kalkulaci chyby v jednotlivých časových krocích filtrace během jednoho běhu, a lze tudíž vycházet jen ze startovní a koncové pozice. Proto bylo provedeno vícero měření při různých trajektoriích robota, aby výsledky byly co nejvíce objektivní.

Obrázek č. 14 ukazuje na dvou trajektoriích o různém průběhu a délce přínos Kalmanovy filtrace v estimaci polohy robota. Mimo to je ilustrována přesnost odhadu s použitím a bez použití kompasu v agregačním modelu během celé jízdy (na počátku běhu musí být vždy uplatněn pro ustanovení prvotního stáčení). Experimenty rovněž ukázaly, že měření rychlosti integrací hodnot z akcelerometrů této cenové kategorie při relativně nízké frekvenci snímání atd. je bez stabilizačního prvku v podobě absolutního senzoru během několika málo vteřin chybné. Přesnost bez Kalmanovy filtrace se pohybuje průměrně kolem 4 %, naproti tomu filtrace s UKF dosahuje přesnosti v odhadu polohy do 1,33 %, což je více než dostatečné k mapování výsledků detekčních algoritmů cesty do lokální mapy s ohledem na její detail.

Na obrázku č. 15 je vyobrazen výsledek filtrace stáčení. Zároveň byl ilustrován i malý vliv kompasu na estimaci úhlu natočení, což se pozitivně projeví při krátkodobých magnetických anomáliích, které tak nemají viditelný efekt na výsledek estimace. Avšak absolutní hodnota stáčení z kompasu slouží i jako stabilizační prvek ohraničující maximální možnou chybu estimace úhlu stáčení, a proto kombinace této dvojice senzoru se ukázala jako velmi přínosná a vzájemně se doplňující.



Obrázek 14: Ukázka zlepšení přesnosti estimace pozice při filtraci pomocí UKF. O značí rotační optický enkodér, C kompas, I inerciální jednotku. Křivka raw ilustruje výsledek bez filtrace, UKF pak s filtrací. Na obrázku vlevo je přesnost s UKF 1,07 %, na obrázku vpravo 1,56 %. Průměrná přesnost z více jízd činí 1,33 %.



Obrázek 15: Ukázka filtrace stáčení. Díky vysoké míře „nedůvěry“ v hodnoty kompasu se chyby způsobené krátkodobými magnetickými anomáliemi příliš neprojeví. Zároveň však kompas slouží jako stabilizační prvek pro inerciální navigaci.

5.3 Shrnutí

Obsah této kapitoly byl věnován široké problematice lokální navigace robota. Úvodní část popisovala zvolenou metodu fúze výsledků detekčních algoritmů za účelem autonomní tvorby lokální mapy okolí, která je nezbytně nutnou složkou každého kognitivního systému. Avšak potřeba mapovat své okolí se neodráží jen ve volbě vhodné vnitřní reprezentace, respektive metody agregace výsledků detekčních algoritmů, ale též i ve způsobu lokalizace robota v souřadném systému mapy. Aby estimace polohy robota byla co nejpřesnější, jsou získaná měření slučována a filtrována pomocí jednoho z nejnovějších přírůstků do rodiny Kalmanových filtrů, a to Unscented Kalman filtrem. Fúze a filtrace pomocí UKF, jejíž lze aplikovat na nelineární systémy, přinesla v některých případech až čtyřnásobné zpřesnění určení polohy robota.

6 Globální navigace

Globální navigace má neméně důležitou roli v navigační architektuře robota a podobně jako lokální navigace řeší problém lokalizace a hledání cesty, avšak v jiném měřítku, a tedy i na zcela jiné úrovni abstrakce. Signifikantní rozdíl mezi lokální navigací a globální navigací je v absenci mapy prostředí, kde v lokálním měřítku tvorba vlastní mapy představuje de facto nutný počín, avšak na globální úrovni mapy bez přílišného detailu zpravidla existují. Navíc u autonomních mobilních vozidel je dostupnost těchto map vynucena, neboť tvoří podkladový materiál pro vymezení a definování jízdní trasy.

Míra abstrakce spjatá s měřítkem globální navigace předepisuje typy map, jež se ve vnitřních modelech uplatňují, tedy geometrické a topologické mapy. Právě kvalitativní geometrie, neboli geometrie bez velikostí, se jeví jako přirozený a efektivní model pro řešení některých otázek v globální navigaci robotů, tj. hledání cesty a rozhodovacích procesů. Samotná reprezentace těchto map formou grafů dovoluje aplikaci celé řady grafových algoritmů²³. Mapy mají také nižší nároky na paměť a větší robustnost proti chybám, jelikož metrická informace není jedinou a mnohdy ani prioritní informací.

Tato kapitola se zabývá stručným popisem vnitřní reprezentace světa pro globální úroveň navigace a metodami lokalizace robota ve zvolené vnitřní reprezentaci s využitím technik nazývaných angl. *map matching*. Estimace polohy uplatňuje jak data z GPS senzoru, tak i lokálního pozičního estimátoru, jež slučuje pomocí fuzzy logiky.

6.1 Vnitřní reprezentace prostředí

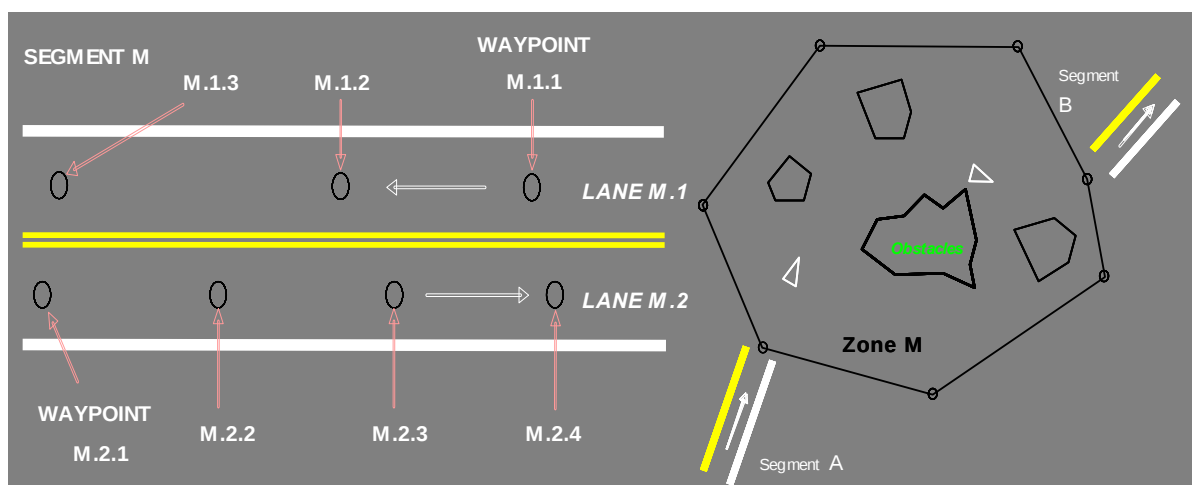
Vnitřní reprezentaci prostředí pro globální úroveň navigace představuje geometrická mapa, která se do systému dodává z vnějšku v podobě souboru ve formátu RNDP (*Route Network Definition File*, [7]). Jak již bylo nastíněno, geometrickým popisem se rozumí reprezentace prostředí pomocí geometrických primitiv jako jsou např. úsečky, kružnice, polygony, eliptické oblouky atd. Avšak práce se složitějšími útvary klade vyšší nároky nejen na výpočetní výkon, ale i na samotné tvůrce těchto map. Málokdy bývá vstupní mapa natolik podrobná (míněno v kontextu robotických soutěží), aby poskytovala informaci například o šířce vozovky, o parkovacích zónách atp. Proto se vnitřní reprezentace abstrahuje ještě více, a to až do podoby úseček, které ztělesňují úseky cesty. Samotné úsečky jsou definovány body cesty v souřadném systému UTM (Univerzální Transverzální Mercatorův systém souřadnic, [66]). Stěžejní výhodou tohoto globálního souřadného systému je možnost spočítat vzdálenost dvou bodů pomocí Pythagorovy věty, pakliže oba body leží ve stejné zóně, což je oprávněný předpoklad vzhledem k měřítku globální navigace uvažovaného mobilního robota.

²³ Geometrická mapa v té nejprimitivnější podobě má k topologické mapě velmi blízko a de facto z pohledu implementace se spíše jedná o triviální topologickou mapu s metrickými atributy. Proto se v textu dopouštíme drobné nepřesnosti při společné charakteristice těchto vnitřních modelů.

6.1.1 Mapová data

Vnitřní reprezentace prostředí se ustanoví načtením vstupního souboru ve formátu RNDF. Tato data mají podobu textového souboru s jednoduchou, avšak komplexní strukturou deklarující snadné parsování. Data z RNDF popisují přístupné silniční úseky a poskytují informace o šířce jízdního pruhu, o parkovacích místech, o silničním značení atd. Nicméně detailní informace nebývají často pro parková prostředí k dispozici (např. mapy pro soutěž Robotour) a mapa obsahuje pouze informaci o sjízdňích segmentech bez větších podrobností.

RNDF dělí prostor na segmenty. V každém segmentu se může nacházet několik cest. Cesta je determinována úsečkami, které definují tzv. body cesty (angl. *waypoints*). Body cesty mají zadány souřadnice ve formátu WGS84, jenž určuje lokaci pomocí dvojice čísel, konkrétně zeměpisnou šířkou a zeměpisnou délkou. Při načítání mapy do vnitřního modelu proto musí docházet k transformaci těchto souřadnic do zvolené vnitřní reprezentace v souřadnicovém systému UTM (více v [67]).



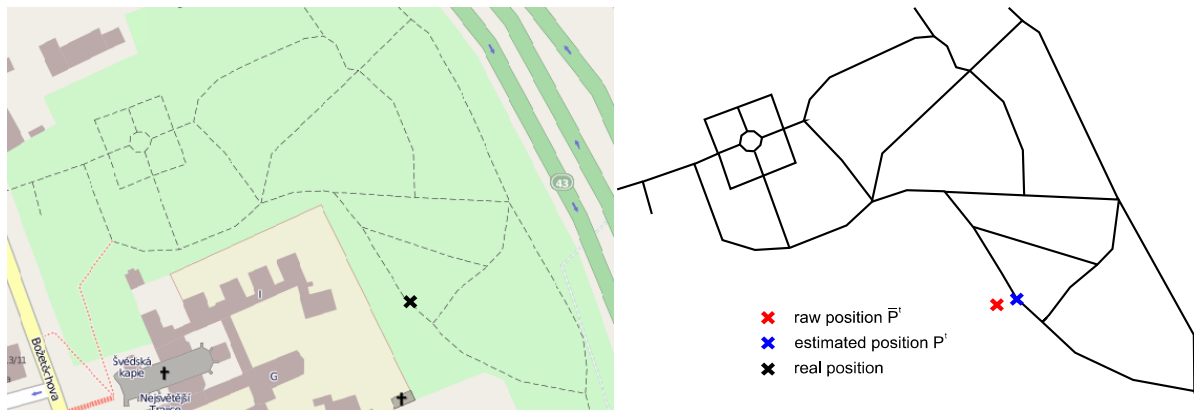
Obrázek 16: Ukázka RNDF formátu specifikace cesty pomocí tzv. bodů cesty. Převzato z [7].

6.2 Estimace pozice

Globální poziční systém GPS se stává stále oblíbenější v autonomních navigačních systémech (*Autonomous Navigation System*, ANS) a významně ulehčuje problém absolutní lokalizace mobilního robota. Avšak cílem globální lokalizace při globální navigaci mobilního robota není určení absolutně přesné pozice vzhledem ke zvolenému globálnímu souřadnému systému, ale jeho lokalizace na dodané globální mapě, která není zcela bezchybná (neúplnost, geometrické nepřesnosti).

Více formálně specifikujme problém estimace odhadu pozice jako sledování pohybu vozidla na konečné množině cest $\bar{N}$. Dále buď aktuální pozice v čase t označena jako $\bar{P}^t$ a odhad pozice v čase t buď označen jako P^t . Cílem je vybrat cestu $n \in \bar{N}$ zahrnující $\bar{P}^t$ v čase t . Bohužel $\bar{N}$ není známo přesně, ale ve shodě s vnitřním modelem existuje pouze síťová reprezentace N sestávající z množiny křivek $A \in \mathbb{R}^2$. Necht', opět ve shodě s vnitřním modelem, je každá křivka po částech lineární, a tedy může být charakterizována konečnou sekvencí bodů $(A_0, A_1, \dots, A_n)$ nazývaných body cesty. Podstatou problému, který se v angl. terminologii označuje jako *map-matching problem*, je porovnání aktuální pozice $\bar{P}^t$ se všemi $A \in N$ a následné vybrání té křivky, jež nejlépe koresponduje s aktuální pozicí, a odhadnutí pozice P^t na této křivce (viz obr. č. 17, [68]). Není proto překvapením,

že se jedná o podkategorii obecnější úlohy mobilních robotů, a to ustanovení jednoznačné korespondence mezi aktuální lokální mapou a uloženou globální mapou za účelem lokalizace mobilního robota.



Obrázek 17: Map-matching problem, park Božetěchova.

6.2.1 Existující techniky

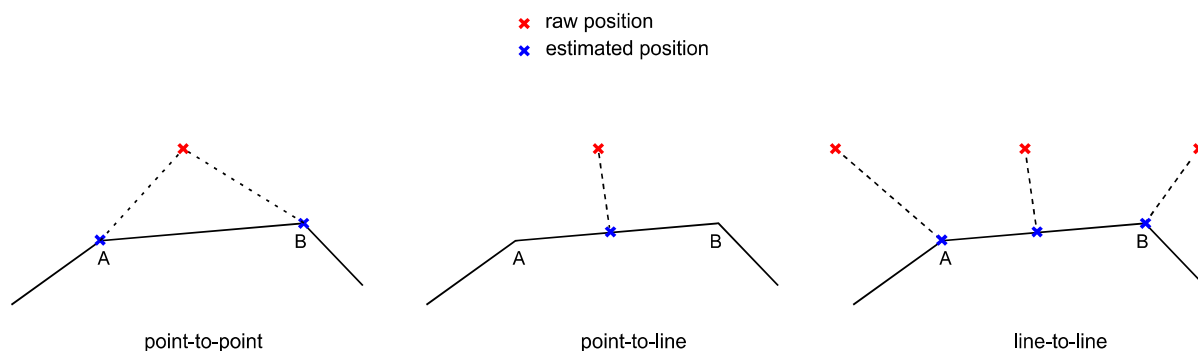
Map-matching techniky musí být nejen přesné, ale i výpočetně efektivní. První algoritmy byly více zaměřeny na přesnost a konzistentnost estimace pozice nežli na výpočetní efektivitu, neboť počet zkoumaných možností byl stále ještě dostatečně malý (uvažovaly se nerozsáhlé oblasti). Proto většina recenzovaných algoritmů míní výkonnosti algoritmu procento správně identifikovaných segmentů, nikoliv výpočetní efektivitu algoritmu [69].

Existence GPS a jeho rostoucí popularita v otázkách lokalizace dopravních prostředků, kde měřítko navigace nabývá zcela jiných rozměrů, vnesla požadavek na nízkou výpočetní složitost aplikovaných metod. Nadto je potřeba výpočetní nenáročnosti umocněna u síťových map vysokého rozlišení, které jsou nezbytné pro přesnou identifikaci zvolené trasy.

Nicméně v kontextu uvažovaného prostředí, v němž se robot pohybuje, bude hlavní metrikou výkonnosti metody její přesnost a ne její výpočetní náročnost, kterou lze navíc reálně zlepšit celou řadou indexačních algoritmů či jiných technik (problém hledání nejbližšího souseda NNS; inspirace v prostorových databázích, GIS software apod.). Dle přístupu mohou být *map-matching* techniky klasifikovány do čtyř základních kategorií, a to na přístupy těžící z metrických informací, přístupy postavené na topologické informaci, pravděpodobnostní přístupy a jiné pokročilé metody [70].

6.2.1.1 Přístupy založené na geometrické informaci

Mezi elementární techniky patřící do kategorie geometrických procedur, jež neuplatňují informaci o propojení segmentů, se řadí tzv. *point-to-point* metoda hledající nejbližší uzel sítě k aktuální pozici robota. Drobné vylepšení přináší technika kalkulující se vzdáleností od úseku cesty definované dvěma uzly označovaná jako *point-to-line*, nebo *point-to-curve*. Přirozené rozšíření algoritmu na celou množinu pozic se skrývá pod názvem *line-to-line*, respektive *curve-to-curve*, kdy se porovnává každá pozice z množiny jednotlivě pomocí *point-to-line* metody, přičemž se zvolí takový segment mapy, k němuž mají všechny pozice z množiny pozic nejbližší. Hlavním nedostatkem všech těchto geometrických přístupů je jejich ignorace časové posloupnosti příchoďů GPS, stejně jako opomenutí informací o propojení uzlů v síti. Díky tomu jsou tyto metody velmi citlivé na odlehle hodnoty, může u nich snadno docházet k oscilacím odhadu pozice mezi dvěma segmenty apod.



Obrázek 18: Ilustrace procedur založených na geometrické informaci (point-to-point, point-to-line, line-to-line).

6.2.1.2 Přístupy založené na topologické informaci

V kontrastu s geometrickým přístupem berou topologické metody v úvahu nejen metrickou informaci, tj. vzdálenost mezi GPS body a elementy mapy, ale i znalost o propojení elementů sítě a časovou sekvenci příchodů GPS dat. Většina těchto metod pracuje ve dvou krocích. Během inicializační fáze se najde počáteční uzel nebo úsečka cesty dle geometrické procedury. Následně se trasa odvíjí vybíráním takových úseků, u nichž existuje topologická vazba na předchozí stav. Konkrétní volba úseku z kandidátní množiny může záviset na mnoha příznacích. Nejčastěji se uplatňuje velikost vektoru definovaného GPS bodem²⁴ a bodem vzniklým kolmým průmětem GPS bodu do úseku cesty, respektive minimum ze vzdáleností ku koncovému či počátečnímu uzlu úseku cesty, pakliže se kolmý průmět nachází vně úsečky.

6.2.1.3 Pravděpodobnostní přístupy

Pravděpodobnostní algoritmy vyžadují definici eliptických nebo čtvercových oblastí „důvěry“ okolo fixní pozice získané z GPS senzoru. Tento přístup poprvé uplatnil a uvedl Honey r. 1989 u relativní měření pozice (americký patent č. 4,796,191, [71]). V roce 1997 se Zhao zabíral možnostmi aplikace této metodologie i pro data získaná z GPS senzoru a ukázal, že velikost oblasti lze odvodit ze střední kvadratické fluktuace chyby GPS měření. Oblasti chyb jsou pak navrstveny do mapy a identifikují svým překrytím segmenty vozovky, po kterých se vozidlo pravděpodobně pohybuje. Jestliže oblast chyby obsahuje větší počet kandidátních segmentů, je výsledný segment zvolen dle dalších kritérií, mezi než patří např. Euklidovská vzdálenost, konektivita, úhlová vzdálenost aj. Zajímavou modifikaci uvedl r. 2004 Ochieng [72], který v prezentované metodě uplatňuje tzv. eliptickou chybu pouze v případě, kdy vozidlo projíždí skrz křižovatku. Oblast chyby se tudíž nevytváří kolem fixních pozic, pakliže se vozidlo pohybuje po jediné cestě (lince), a dochází k významnému snížení výpočetních nároků spjatých s tvorbou eliptických tvarů.

6.2.1.4 Pokročilé techniky

Do kategorie pokročilých technik lze zařadit různé metody vybudované na hybridních přístupech, Kalmanově filtraci, Dempster-Shafer teorii, částicových filtrech, fuzzy logice aj. Typickou vlastností těchto metod je využití více zdrojů k přesnější estimaci polohy robota, řešení agregace získané

²⁴ GPS bodem je míněn bod s GPS koordináty převedenými do UTM souřadného systému. Dále v textu již implicitně předpokládáme transformaci GPS koordinát do souřadného systému globální mapy.

informace, filtrace odlehklých hodnot atd. Některé algoritmy jsou založeny i na geometrické analýze vozovky, aktuální dynamice jízdy vozidla atp.

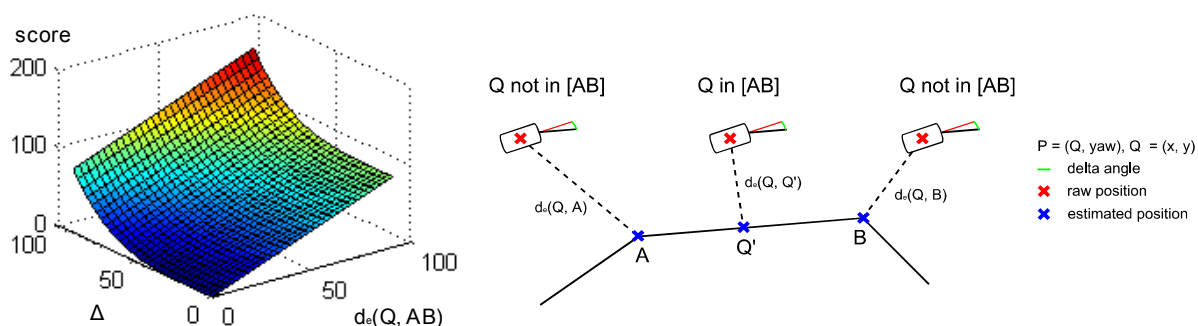
6.2.2 Navržená map-matching metoda

Mezi základní omezení pohybu vozidla prostorem patří skutečnost, že vozidlo se smí vyskytovat v průběhu jízdy pouze na silnici. Při podmínce úplné a korektní mapy silniční síť pak musí existovat „jednoznačné“ zobrazení aktuální polohy robota do dodané mapy. Dalším omezením nechť je podmínka planimetrického pohybu vozidla ve shodě s dvourozměrnými metrickými informacemi, jež jsou zaneseny v mapě.

Navržená lokalizační metoda slučuje poziční informaci z více zdrojů pomocí fuzzy logiky a využívá metrických i topologických informací obsažených v mapě. Spadá tudíž do kategorie hybridních či pokročilých lokalizačních technik. Následující kapitoly postupně popisují prostorovou konfiguraci robota, metodu ohodnocení úseku cesty vůči aktuální konfiguraci, slučování poziční informace z GPS a lokálního pozičního estimátoru a jednotlivé fáze estimace polohy pomocí stavového automatu.

Prostorová konfigurace robota

S ohledem na počet dimenzí uvažovaného prostoru a znalost o stáčení vozidla (kompas) definujeme globální prostorovou konfiguraci jako dvouprvkový vektor $P=(Q, yaw)$, kde $Q=(x, y)$ určuje pozici vozidla v UTM a yaw jeho úhel stáčení. Obrázek č. 19 ilustruje tři různé konfigurace.



Obrázek 19: Vlevo grafické znázornění hodnotící funkce úseku cesty pro konstantní váhu $\omega = 30$. Vpravo tři možné konfigurace pozice robota vůči úseku cesty.

Ohodnocení úseku cesty

Mějme množinu cest N , kde každá cesta $A \in N$ je po částech lineární a je definována konečnou sekvencí bodů $(A_0, A_1, \dots, A_n)$, přičemž $A_i = (x, y)$. Nechť P značí odhadovanou konfiguraci vozidla dle měření ze senzorů $P=(Q, yaw)$. Nejprve stanovme distanci mezi pozicí (bodem) Q a úsekem cesty $AB \in \{(A_i, A_{i+1}) | A_i, A_{i+1} \in A\}$, a to jako:

$$d(Q, AB) = \begin{cases} d_e(Q, Q'), & \text{pokud } Q' \in [AB] \\ \min\{d_e(Q, A), d_e(Q, B)\}, & \text{jinak} \end{cases}, \quad (36)$$

kde $d_e(P_1, P_2)$ je Euklidovská vzdálenost mezi dvěma body a Q' označuje bod vzniklý kolmým průmětem do přímky určené úsečkou AB .

Dále buď Δ absolutní úhlová distance ve stupních mezi stáčením vozidla a směřováním úseku cesty AB . Pak celkové skóre úseku cesty se spočte rovnicí:

$$score(P, AB) = d(Q, AB) + \left(\frac{\Delta}{\omega}\right)^4, \quad (37)$$

přičemž ω má význam váhy a představuje maximální úhlovou vzdálenost, od které dochází k významnému postihu úseku vůči aktuální konfiguraci vozidla (viz obr. č. 19).

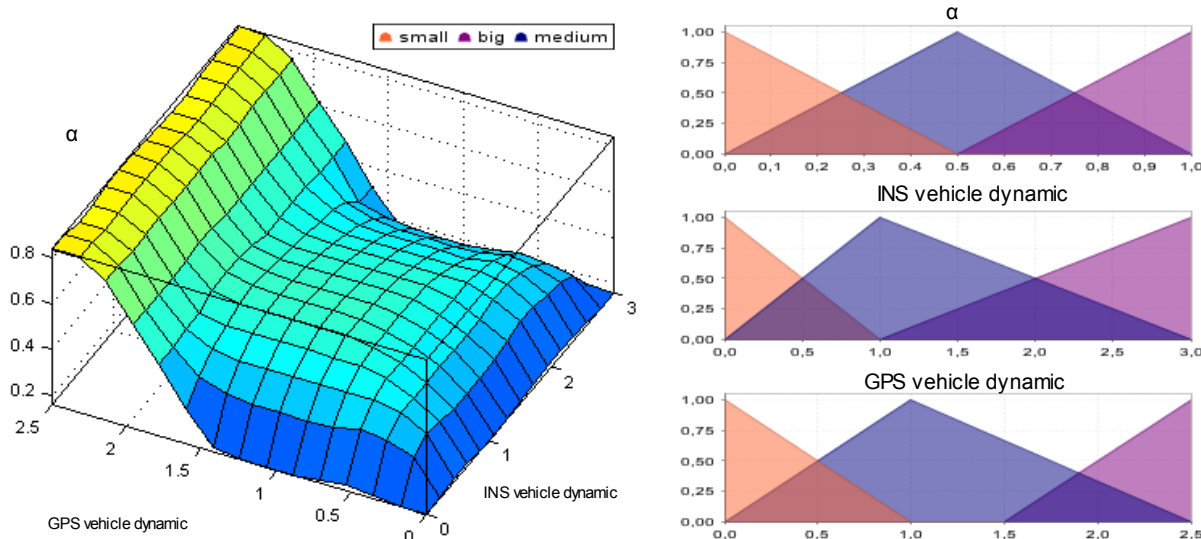
Agregace poziční informace

Kombinace globálního polohovacího systému a interního navigačního systému (INS, tj. lokální estimátor pozice) má doplňující se vlastnosti. GPS poskytuje globální poziční informaci s nízkou frekvencí, avšak bez integrační chyby. Právě nízkou frekvencí a občasnou nedostupností dat lze kompenzovat INS, jenž je nezávislý na vnějších signálech a představuje spojitý měřicí systém.

Informace o pozici vozidla z GPS senzoru není do globální konfigurace integrována přímo, ale jako vážená velikost změny vůči konfiguraci v předchozím kroku. Naopak data z INS jsou agregována bez dalších úprav, protože se jedná o data již filtrovaná Kalmanovým filtrem v lokálním estimátoru pozice. Podobně jako u GPS se i u INS připočítává velikost změny (38).

$$\begin{array}{ll} \text{INS} & \text{GPS} \\ x_{k+1} = x_k + \Delta x_{INS} & x_{k+1} = x_k + \alpha(x_{GPS} - x_k) \\ y_{k+1} = y_k + \Delta y_{INS} & y_{k+1} = y_k + \alpha(y_{GPS} - y_k) \\ yaw_{k+1} = yaw_{INS} & \alpha \in \langle 0, 1 \rangle \end{array} \quad (38)$$

Konstantní velikost váhy α pro GPS může způsobit selhání estimace, případně vést k méně přesným výsledkům (tzv. problém s prahovými hodnotami [70]). Naproti tomu vhodná volba α dle aktuální situace přirozeně povede ke zlepšení přesnosti odhadu pozice i k jeho vyšší spolehlivosti. Bohužel exaktní kvantitativní popis a analytické vyjádření pozorovaných závislostí mezi zvolenými veličinami není triviální a vyžaduje proto užití takového inferenčního systému, který si poradí s vágností definovaných pojmů a poskytne prostředky pro přirozený popis požadovaného chování.

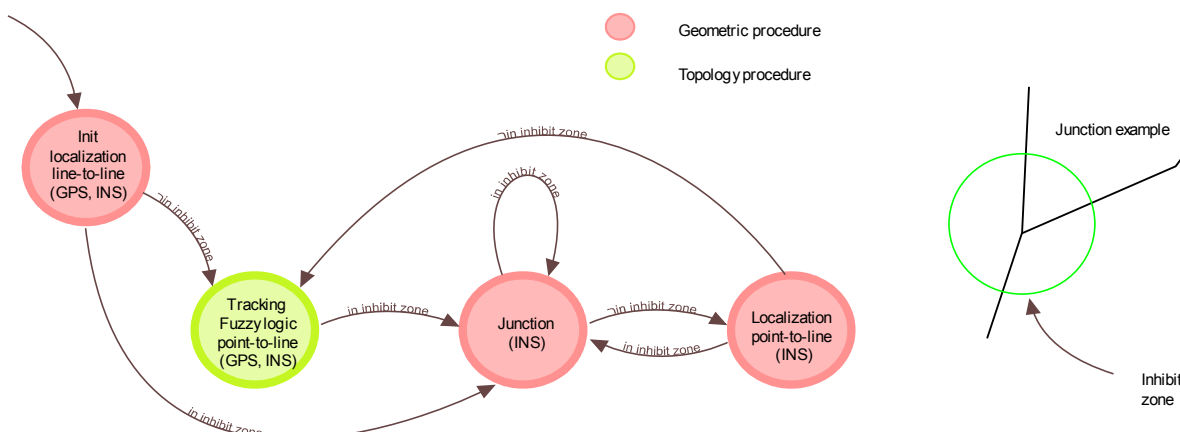


Obrázek 20: Vlevo fuzzy plocha pro výpočet váhového koeficientu, vpravo definice fuzzy proměnných pomocí trojúhelníkových funkcí příslušnosti.

Mezi formální systémy, jež disponují zmíněnými vlastnostmi, se řadí fuzzy logika. Fuzzy logika umožňuje popis závislosti váhy α na ostatních veličinách ve strukturované a snadno srozumitelné podobě pomocí inferenčních pravidel. Na základě pozorování byly definovány dvě vstupní fuzzy proměnné, a to dynamika vozidla dle INS (*INS vehicle dynamic*) a dynamika vozidla dle GPS (*GPS vehicle dynamic*). Dynamika vozidla ve skutečnosti vyjadřuje ujetou vzdálenost od posledního příchodu GPS dat podle INS, tj. *INS vehicle dynamic*, a vzdálenost mezi polohou danou nově příchozími GPS souřadnicemi promítnutými do mapy pomocí *point-to-line* techniky a dříve estimovanou polohou při příchodu předešlých GPS dat, tj. *GPS vehicle dynamic*. Fuzzy regulátor vychází z Mandaniho modelu a v defuzzyfikaci se uplatňuje metoda těžiště (angl. *centroid of area*). Konkrétní podobu fuzzy regulátoru pro výpočet váhy α ilustruje obrázek č. 20.

Lokalizační fáze

K hlavním obtížím *map-matching* metod patří jejich inicializační fáze, řešení křižovatek (zejména tvaru Y) a nekvalitní mapové podklady ve smyslu přesnosti zaznamenaných informací. Navržená *map-matching* metoda podává dobré výsledky a stabilní estimaci (nedochází k oscilacím mezi úseky cesty), má vysokou úspěšnost ve styčných oblastech a velkou toleranci jak k nepřesnostem měření, tak i k podkladovému materiálu. Celkový koncept, respektive grafické znázornění logiky a definic přechodů mezi jednotlivými stavy lokalizační metody zobrazuje obr. č. 21.



Obrázek 21: Vlevo stavový popis navržené *map-matching* metody. Vpravo ilustrace inhibiční zóny.

Inicializační fáze

Inicializační procedura tvoří klíčový prvek pro následné správné trasování pohybu vozidla. Cílem inicializační fáze je lokalizovat vozidlo do mapy bez předchozích informací o jeho pohybu a poloze, tj. problém globální lokalizace. Aby se minimalizovala možnost vzniku chyby, uplatňuje se v procesu estimace vícero²⁵ měření polohy z absolutních senzorů (kompas a GPS). Na získanou množinu změřených konfigurací M se aplikuje modifikovaná geometrická metoda *line-to-line*. Konkrétně se vybere takový úsek cesty, jenž má nejmenší hodnotu skóre (39). Nakonec proběhne namapování vozidla do vybraného úseku AB (kolmý průmět, hodnota stáčení není měněna) a estimovaná poloha na vozovce je prohlášena za inicializační konfiguraci.

$$AB^* = \underset{AB \in E}{\operatorname{argmin}} \sum_{P \in M} \operatorname{score}(P, AB), \text{ kde } E = \cup_{A \in N} A \quad (39)$$

²⁵ Experimentálně bylo stanoveno $|M| = 3$, tj. tři měření dle GPS a kompasu.

Fáze sledování

Fáze sledování využívá topologické informace pro lokalizaci vozidla. Specifičtěji, jakmile je vozidlo přimknuto k cestě dle inicializační nebo lokalizační fáze, dovoluje se již pouze změna v rámci úseků cesty, nikoliv mezi cestami samotnými. Přičemž mapování do úseků cesty se provádí jen pro korekci pozice po započítání dat z GPS, jež může způsobit větší odskok, a to dle:

$$AB^* = \operatorname{argmin}_{AB \in A} \operatorname{score}(P, AB) \quad , \quad (40)$$

kde P má význam konfigurace po agregaci dat a A označuje silnici, po níž se vozidlo pohybuje. Koeficient hodnotící funkce ω byl experimentálně stanoven na hodnotu $\omega = 45$, neboli větší zřetel je ve fázi sledování kladen na Euklidovskou vzdálenost nežli na směrový rozdíl. Aktuální konfigurace se následně získá kolmým průmětem do vybraného úseku AB bez změny změřeného stáčení vozidla. Velikost odchylky od úseku cesty způsobené agregací dat INS při frekvenci příchodů GPS dat (nepředpokládají se dlouhodobé výpadky GPS senzoru) implicitně vhodně omezují jízdní limity vozidla a není tudíž účelné spouštět proces lokalizování se v mapě s tak velkou frekvencí.

Inhibiční zóna

Křižovatky představují oblasti, kde se přesnost a stabilita odhadu globální pozice stává směrodatným kritériem pro správné vyhodnocení situace, korektní odbočení vozidla a následnou zpětnou lokalizaci do patřičného úseku cesty. Aby bylo dosaženo uvedených kritérií, byly kolem koncových uzlů cest vytvořeny tzv. inhibiční zóny (viz obr. č. 21).

Inhibiční zóna je kruhová oblast s definovaným poloměrem²⁶, v níž se zcela potlačuje vliv GPS senzoru na estimovanou pozici, a tedy i proces mapování do mapy. Proto jedinou agregovanou informací jsou data z INS a systém tak přejímá vlastnosti lokálního pozičního estimátoru, tj. stabilitu, prostorovou kontinuitu estimované pozice aj. Inhibiční oblast tudíž poskytuje prostor pro projevení vlastního manévrování vozidla a značně zpřesňuje řešení zpětné lokalizace po opuštění křižovatky.

Fáze lokalizace

Fáze lokalizace nastává po vyjetí vozidla z inhibiční zóny a jejím cílem je opětovně určit prostorovou konfiguraci vozidla v dodané mapě. Výběr cesty a úseku cesty, na kterém se vozidlo nachází, se děje opět pomocí (39), avšak uvažuje se pouze poslední měření ($|M| = 1$). Primární vliv na ohodnocení úseku má ve fázi lokalizace úhlový rozdíl mezi stáčením vozidla a směrováním cesty, neboť estimovaná pozice nemusí být dostatečně přesná (je kupříkladu zatížena chybou již při samotném vstupu vozidla do inhibiční zóny). Podle provedených experimentů byla hodnota koeficientu ω stanovena na $\omega = 25$.

6.3 Shrnutí

Cílem této kapitoly bylo představit používané fundamentální techniky globální lokalizace robota na dodané mapě. Dále uvést zvolenou vnitřní reprezentaci okolního světa a navrženou metodu globální estimace pozice, jež se umí vypořádat jak s nepřesnými mapovými podklady, tak i s nepřesnostmi samotných senzorů měření absolutní pozice robota (GPS, INS).

²⁶ Poloměr byl s ohledem na charakter chyby GPS senzoru odhadnut na šest metrů.

7 Programová realizace

Kapitola *Programová realizace* pojednává velmi stručně o implementovaném programu pojmenovaném *Videomap*. Název programu plyne z historického vývoje navigační architektury (viz. příloha A), kdy hlavní funkcionalitou *Videomapy* byla pouze detekce cesty z obrazových dat. Dnes je *Videomap* vícevláknovou aplikací řešící lokalizaci robota na globální i lokální úrovni, tvořící vnitřní model okolí pro lokální navigaci, zpracovávající data z laserového dálkoměru a kamery pro detekci sjízdného povrchu atd.

7.1 Implementace

Aplikace *Videomap* je implementována v jazyce C++ s využitím multiplatformního toolkitu *Qt* a multiplatformní knihovny pro zpracování obrazu *OpenCV*. V lineární algebře (práce s maticemi a vektory) bylo využito šablonové knihovny s otevřeným zdrojovým kódem *Eigen* ve verzi 2.0.12. Zpracování dat fuzzy logikou staví na objektové knihovně optimalizované pro aplikace s kritickými požadavky na výkon, a to na *Free Fuzzy Logic Library (FFLL)*, která dovoluje definovat inferenční pravidla a fuzzy proměnné dle standardu IEC 61131-7 (*FCL, Fuzzy Control Language*). Součástí aplikace je rovněž rozhraní pro zpracování dat pomocí programového prostředí a skriptovacího jazyka určeného k vědeckotechnickým výpočtům *Matlab*²⁷.

Programová realizace má rozsáhlou strukturu a sestává téměř ze 180 objektů, jejichž funkcionalita je definována přibližně na 30 000 řádcích programového kódu. V samotném objektovém návrhu byla uplatněna řada návrhových vzorů, například jedináček, šablonové metody, iterátory, adaptér, továrny, pozorovatel atd. Avšak rozdělení programu do nezávislých sekcí, které poskytuje objektové programování, nemusí být vždy dostatečné. Často je třeba program rozdělit i do oddělených, samostatně běžících podúkolů, kdy při uplatnění *multithreadingu* je každý z těchto podúkolů řízen výkonným vláknem.

Jedním z nejpersvědčivějších důvodů pro užití paralelního programování se jeví možnost zpracování dat, jež jsou svázána s určitým zdrojem nebo událostí tak, aby nedocházelo ke zdržování zbytku programu. Vlákenný model poskytuje tedy způsob, jak vytvářet transparentně škálovatelné programy a zvýšit tím průchodnost celého systému, což je kritická výhoda právě v kontextu řídicí architektury robota. Nejen z těchto důvodů byla aplikace *Videomap* koncipována jako vícevláknová.

Samotný popis užitých programových technik a podrobný rozbor návrhu aplikace by byl neúměrně rozsáhlý, proto se tato podkapitola zabývá realizací pouze v přehledové formě, přičemž podrobněji uvedeny jsou jen vybrané partie řešení.

7.1.1 Jádro

Jádro plní účel bazového rozhraní a jeho hlavní rolí je vytvořit společnou mezivrstvu pro bezpečnou komunikaci mezi objekty žijícími v různých vláknech (angl. *thread-safe objects*) a systémově řešit obecné požadavky. Mezi ně patří například deklarace univerzálního rozhraní pro přístup k datovým

²⁷ Přeložená aplikace kvůli licenčním podmínkám je této funkcionality zbavena (závislost na placených knihovnách), avšak zdrojové kódy pro přenos dat z aplikace do prostředí *Matlabu* jsou na příloženém datovém nosiči k dispozici.

složkám objektu, generování jedinečných identifikátorů objektů, schopnost transformace objektu do stromové struktury, jež je následně využita při generování XML atd.

Míra transparentnosti paralelního modelu, kdy na jedné straně paralelizujeme provádění cyklů (typická aplikace *OpenMP*) a na straně druhé máme k dispozici skutečně paralelní jádro umožňující každému objektu žít v samostatném vlákne starajícím se o smyčku událostí *in situ*, má signifikantní dopad na celkový návrh aplikace. Jádro *Videomaps* se řadí k druhému extrému a snaží se nabídnout zcela průhledné paralelní podhoubí, které dovoluje za běhu aplikace škálovat stupeň souběžného zpracování. Při této úrovni jsou o to více palčivé otázky týkající se sdílení omezených zdrojů, komunikace mezi vlákny, respektive objekty (sdílená paměť, zasílání zpráv), jejich spolupráce, zajištění jejich existence a jejich korektní dealokace atd., neboť návrh jádra musí danou problematiku řešit v co nejobecnější rovině.

Když standardizační komise pro C++ definovala první standard, byl mechanismus paralelního programování záměrně vynechán, neboť existovalo mnoho rozdílných přístupů, jak nástroje pro paralelní programování implementovat. Proto paralelně programovat v C++ v současnosti znamená sáhnout po některé z dostupných knihoven. Avšak toolkit *Qt* poskytuje vláknovou podporu ve formě platformě nezávislé třídy *QThread* a širokou škálu synchronizačních prvků (třídy *QWriteLocker*, *QWaitCondition*, *QSemaphore* aj.).

Mezi zajímavé problémy, jež jsou zvláště nápadné ve zpracování senzorických měření²⁸, patří právě přístup k distribuci získaných údajů do jednotlivých vláken. Protože se povětšinou jedná o datově objemné struktury (např. obrazová data), je nevhodné obcházet sdílení přístupu jejich kopírováním. Navíc samotný obsah není měněn, ale čten a transformován do jiné struktury. Senzorická měření mají z principu dočasnou platnost a nemá tudíž velký smysl uchovávat tato data v surové formě po celou dobu běhu robota (pomineme-li navíc paměťovou náročnost) a je tak třeba řešit způsob jejich dealokace. Při předávání klasických ukazatelů na sdílená data vyvstávají dvě podotázky - kdy data dealokovat a které vlákno má dealokaci provést.

Jádro pro korektní dealokaci zdrojů obsahuje implementaci tzv. bezpečných ukazatelů, jež se předávají pomocí mechanismu signálů a slotů v toolkitu *Qt* (předávání dat pomocí signálů a slotů je vláknově bezpečné, pakliže se konekce ustanoví s atributem *Qt::BlockingQueuedConnection* nebo *Qt::QueuedConnection*). Bezpečný ukazatel má podobu objektu obsahujícího ukazatel na sdílená data, tj. objekt podděněný z bazové třídy *SharedData* zahrnující čítač počtu referencí s atomickými operacemi dekrementace, inkrementace a testu na nulu, respektive kombinace těchto operací. Vytvoření nového bezpečného ukazatele na sdílený objekt pak vyvolá automatickou inkrementaci počítadla referencí v odkazovaném sdíleném objektu. Naopak volání destruktoru bezpečného ukazatele způsobí atomickou operaci dekrementace čítače referencí v odkazovaném objektu a testu tohoto čítače na nulu. Pakliže je čítač referencí po dekrementaci roven nule, volá se destruktory i odkazovaného sdíleného objektu.

7.1.2 Senzorická vrstva

Senzorická vrstva představuje generické rozhraní pro práci se senzory a měřenými daty. Abstrahuje přímou závislost na konkrétním datovém zdroji a emuluje jednotlivé senzory. Programová emulace senzorů dává nejen prostor pro nízkoúrovňové předzpracování měřených dat nebo jejich rektifikaci,

²⁸ Konkretizace je zavedena záměrně. Nicméně na úrovni jádra se jedná o generický problém, jenž je v obecné rovině rovněž řešen.

ale i možnost uživatelské korekce např. přes grafické rozhraní (okno vlastností), a to centrálně s dopadem na celý zbytek navigačního systému. Díky navrženému rozhraní, které staví na volání členských metod s dynamickou identifikací objektů za běhu, mohou být data měření transparentně získána buď přes TCP/IP protokol (reálný běh, viz příloha A), nebo z načteného souboru (záznam ve formátu XML). Zvolený koncept poskytuje snadnou rozšiřitelnost o další vstupní datové formáty, např. binární formát dat.

Lokální poziční estimátor lze chápat jako virtuální senzor měřící aktuální lokální prostorovou konfiguraci robota. Agregace a filtrace dat postavená na Unscented Kalman filtru znamenala jeho implementaci, neboť v současné době neexistuje žádná dostupná knihovna v C/C++, která by jej nabídla v generické formě (knihovny/toolkity jako *OpenCV*, *MRPT*, *BFL* nabízejí nejvýše EKF). Největším úskalím při implementaci UKF je výpočet odmocniny z matice, kterou lze realizovat třemi technikami, a to QR dekompozicí, Choleského dekompozicí a efektní metodou nejmenších čtverců (více viz [60]). Realizovaná generická implementace UKF (využití šablon v C++ a knihovny pro lineární algebru *Eigen*) počítá odmocninu z matice právě jako Choleského dekompozici.

Obdobně globální poziční estimátor představuje de facto virtuální senzor udávající polohu na dodané globální mapě. Fuzzy logika pro fúzi dat z GPS senzoru a z INS je definována v samostatném textovém souboru dle standardu IEC 61131-7. Samotná fuzzy logika nebyla implementována, ale používá se již existující knihovna *FLL*, jež umí tento soubor načíst a ustanovit dle něj fuzzy regulátor.

Požadavek na vyhodnocení dat v reálném čase se odráží rovněž v architektuře distribuující měření dále ke zpracování. Používaný operační systém Windows 7 není operačním systémem reálného času (angl. *Real-Time Operating System*, RTOS). Z toho pramení celá řada nemalých latencí souvisejících s přepínáním vláken, reakcemi na události aj. Důsledkem nepředvídatelnosti délky reakcí na externí události může být kupříkladu kumulace měřených hodnot a pak jejich dávkové odeslání ke zpracování, což bývá častý jev. Příchod dat z jednotlivých senzorů je tak silně asynchronní a nedeterministický. Další nesnáz vyvstává u časově náročného zpracování dat, kdy doba nutná pro vyhodnocení překročí průměrnou „periodu“ jejich příchodu. Zde může snadno nastat situace, že systém bude zpracovávat stará a již nezajímavá či dokonce neplatná data. Navíc z pohledu odběratele těchto dat je žádoucí, aby komunikační kanál poskytoval možnost časové synchronizace dat z více senzorů a odběratel tedy obdržel tzv. „balíček“ časově souvztažných měření.

Implementované komunikační rozhraní poskytuje uvedenou funkcionalitu. Příjemce se může připojit na každý senzor samostatně a definovat, k jakému senzoru se mají ostatní požadovaná data automaticky synchronizovat. Vnitřní logika pak zajistí, že komunikační kanál bude poskytovat časově aktuální a vůči sobě časově souvztažná data.

7.1.3 Detekční vrstva

Detekční vrstva sestává ze dvou komponent, a to detektoru cesty z obrazových dat a detektoru cesty v datech z laserového dálkoměru. Detektory přijímají skrze komunikační rozhraní nejen snímky z kamery, respektive skeny z laserového dálkoměru, ale také časově souvztažnou informaci o prostorové konfiguraci robota v lokální mapě. Ta je následně důležitá při mapování výsledku detekce do vnitřního modelu.

Při zpracování obrazu se hojně volají funkce z *OpenCV*, nicméně byla implementována celá řada dalších algoritmů pracujících nad obrazovou datovou strukturou, neboť se ukázalo, že některé algoritmy je vhodné mírně upravit či vytvořit nové (např. metoda regionálního růstu s více

7.1.4 Vnitřní reprezentace

Vnitřní model prostředí robota má dvojí úroveň dle měřítka navigace, přičemž u lokální navigace se uplatňuje mřížka obsazenosti, naopak u globální navigace geometrická mapa. Mřížka obsazenosti je implementovaná kvůli vyšší rychlosti mapování výsledků detekčních algoritmů jako pole čtvercových segmentů (segment de facto představuje tzv. *bounding box*). Každý segment obsahuje 10 000 buněk a jedna buňka pokrývá čtvercovou plochu o rozměru 10x10 cm.

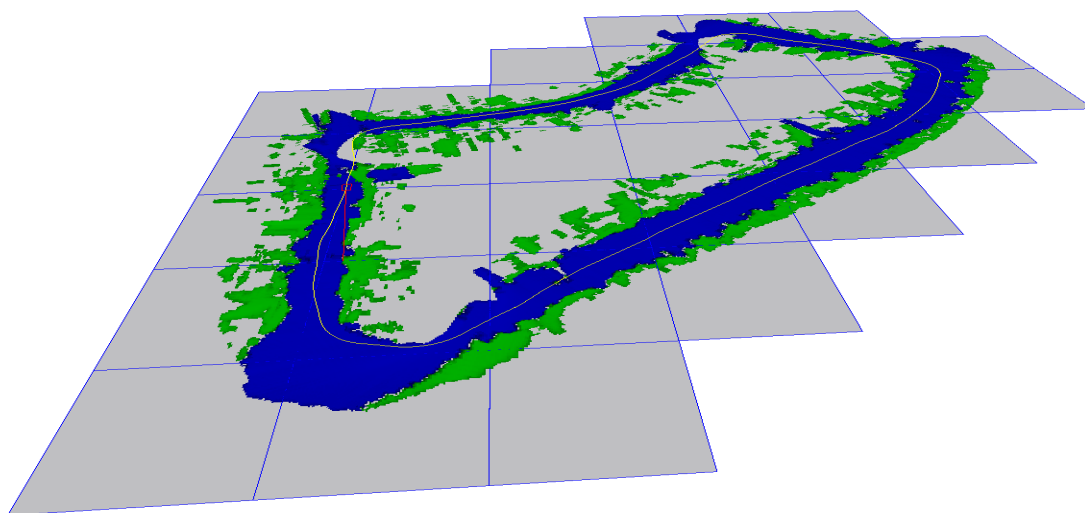
Geometrická mapa má formu neorientovaného nesouvislého grafu, kde uzly obsahují atributy o jejich umístění v UTM souřadném systému. Samotná implementace reflektuje přirozenou hierarchii danou RNDF formátem, tedy sestává ze segmentů, cest, úseků cest a bodů cesty. Pro načtení mapy z RNDF byl implementován vlastní parser tohoto formátu.

7.2 Uživatelské rozhraní

Videomap má koncept SDI aplikace. To znamená, že vždy umožňuje práci pouze s jedním projektovým souborem²⁹. Ve snaze nabídnout co nejprívětivější a intuitivní ovládání je uživatelské rozhraní *Videomapy* kompozicí několika samostatných účelových oken a panelů, přičemž celkový vzhled a rozmístění ovládacích prvků kopíruje zažitý standardy rozsáhlejších aplikací.

7.2.1 Lokální a globální mapa

Filozofie uživatelského rozhraní vychází z pojetí globální navigace a lokální navigace jako dvou graficky nezávislých celků, proto každé navigaci přísluší samotné grafické okno. Lokální mapa vyobrazuje interní reprezentaci pomocí knihovny *OpenGL*. Vzhledem k obrovskému množství buněk je vykreslování urychlováno průběžnou tvorbou tzv. display listů. Okno umožňuje různé pohledy na vyobrazená data (tažením myši při držení pravého tlačítka). Stisk tlačítka ALT aktivizuje automatické trasování pohybujícího se auta.



Obrázek 23: Vizualizace interní reprezentace pro lokální navigaci. Park Božetěchova, jaro 2010.

²⁹ Projektový soubor chápeme jako souhrnný pojem pro skupinu souborů popisujících jeden běh robota reálným prostředím, tj. soubor se záznamem, odpovídající soubor s videem, konfigurační soubor interních a externích parametrů senzorů a soubor globální mapy v RNDF.

Globální mapa sestává z grafických primitiv knihovny *Qt* pro dvourozměrnou grafiku, přičemž inhibiční oblasti kolem uzlů jsou znázorněny zelenými kružnicemi. Textové pole kopírující pohyb robota obsahuje informaci o aktuální rychlosti, jeho pozici v UTM atd. Globální vyobrazení dovoluje rovněž aktivaci trasování, podobně jako okno pro lokální mapu (tlačítko nástrojového panelu).

7.2.2 Přístup k datovým složkám objektů

Pro přístup k objektům a k datovým složkám objektů slouží dvojice oken *Tree view* a *Property view*, jež jsou obě implicitně dokovány do levé části aplikace. Okno *Tree view* vyobrazuje objektovou stromovou strukturu programu a umožňuje spolu s oknem *Property view* u selektovaného objektu snadný přístup k vnitřním datovým složkám a jejich editaci. Interně se pro ustanovení obsahu obou oken využívá obecného rozhraní pro přístup k datovým složkám objektu a jeho následníkům, které je deklarováno jádrem (poděděním od třídy *BObject*).

7.3 Formát záznamu

Záznam z jízdy robota je uložen v samopopisném formátu XML. Soubor obsahuje surová data ze senzorů v pořadí, v jakém byla získána, a konfiguraci senzorů. Navíc každý záznam má časové razítko pořízení. Výjimku tvoří obrazová informace, která je uložena v samostatném souboru ve formě videa. XML soubor pak uchovává pouze klíčové indexy indikující čas pořízení snímku a umístění snímku ve video souboru³⁰.

```
<RCRecord>
  <SCCompassFrame timeStamp="476704644" alpha="98" />
  <SCImuFrame timeStamp="476704644" accX="-0.02" accY="-0.01"
    accZ="-0.01" gyrX="-0.24" gyrY="-0.3" gyrZ="0.01" roll="0.6"
    pitch="0" yaw="222.9" />
  <SCLidarFrame timeStamp="476704691" mode="0" lidarAngle="0"
    lidarMountingHeight="0" data="702, 680, 656, 651, 484, 437,
    432, 624, 631, 617, 621, 611, 599, 592, 586, 581, 575, 563,
    557, 551, 546, 536, 538, 533, 527, 526, 517, 517, 518, 513,
    512, 506, 503, 498, 496, 490, 495, 493, 488, 486, 477, 481,
    480, 479, 476, 476, 475, 475, 475, 474, 473, 475, 474, 474,
    475, 476, 477, 478, 479, 480, 482, 482, 484, 486, 489, 492,
    495, 500, 504, 511, 513, 511, 500, 494, 497, 499, 462, 506,
    506, 549, 554, 577, 583, 614, 619, 671, 679, 655, 737, 764,
    782, 803, 841, 854, 879, 903, 929, 958, 977, 1003, 1031" />
  <SCImgFrame timeStamp="476704706" index="45" />
</RCRecord>
```

Obrázek 24: Ukázka záznamu v XML formátu.

³⁰ Reálně se používá binární formát dat, jež načítá účelová aplikace SensorPad a distribuuje je pomocí TCP/IP protokolu dalším programům. Aby se odstranila závislost na jiné aplikaci a zároveň se nabídla možnost jednoduchého nahlédnutí do záznamu, byl zvolen pro potřeby této práce právě formát XML.

7.4 Shrnutí

Tato kapitola se ve vší stručnosti zabírala podstatnými částmi relativně rozsáhlé programové realizace (30 000 řádků programového kódu, 180 objektů). Popsáno bylo užívané programové vybavení, charakterizovány fundamentální řešené problémy plynoucí z požadavku na zpracování dat v reálném čase, hlavní prvky uživatelského rozhraní aplikace a formát záznamu senzorických měření.

8 Závěr

Robotika je širokou oblastí, která v sobě skrývá různé vědní disciplíny. Náplní této diplomové práce bylo značně rozsáhlé téma, a to navigace robotického autonomního vozidla s využitím laserového dálkoměru a jedné barevné kamery pro detekci sjízdného povrchu v parkovém prostředí. Navržená navigační architektura se snaží řešit nejpálčivější problémy mobilních robotických systémů, které plynou z nutnosti vyhodnotit senzorická měření v reálném čase.

Přirozená dekompozice problému navigace na lokální a globální úroveň se v představené řídicí architektuře odráží v dvouúrovňové vnitřní reprezentaci okolního světa, pro každé měřítko samostatně. Z toho plyne dvojí prostorová konfigurace robota, s níž musí být počítáno v řídicích algoritmech. Avšak v daném případě si prostorové konfigurace „nekonkurují“, jako je tomu u jiných lokalizačních technik (kupř. metoda Monte Carlo lokalizace), ale naopak se doplňují. Navíc ideál absolutně přesné a jediné estimace polohy robota skrze všechna měřítka navigace si nezádá s charakteristikami chyb měření dostupných senzorů a s požadavky na časový průběh této estimace pro autonomní mapování okolí robota, neboť zde např. malá nestabilita stáčení díky agregaci GPS informace může zapříčinit špatné namapování výsledků do lokální mapy a zneplatnit tak budovaný vnitřní model.

V lokální navigaci, kde vlastní tvorba mapy vyžaduje maximálně stabilní odhad a kontinuální prostorovou změnu udávané pozice v čase, se využívá fúze redundantních a asynchronních senzorických měření, a to pomocí Kalmanovy filtrace. Z důvodu nelinearity pohybového modelu robota byl zvolen Unscented Kalman filtr, jenž je založen na myšlence, že snadněji lze aproximovat Gaussovo rozdělení, nežli potenciálně nelineární funkci. Experimenty ukázaly až čtyřnásobné zlepšení přesnosti estimace polohy při použití tohoto stavového estimátoru.

Jakožto vnitřní model poskytující přirozené prostředí pro fúzi výsledků detekčních algoritmů sjízdného povrchu byla zvolena mřížka obsazenosti, která je charakteristická pevnou souřadnou soustavou. Bohužel mapování rozsáhlých dynamických oblastí patří v mobilní robotice stále ještě k otevřeným otázkám. Nicméně úspěšně (s ohledem na uvažovaná parková prostředí) lze tuto nesnáz řešit tzv. technikou zapomínání, při níž robot postupně úmyslně ztrácí informaci o místech, která dříve navštívil. Tento přístup má navíc řadu pozitivních konsekvencí, kupříkladu ohraničuje paměťové nároky vnitřního modelu i čas potřebný k jeho analýze a „zcela“ odbourává požadavek na absolutní přesnost lokální estimace při zavedení předpokladu o dopředném pohybu.

Detekce sjízdného povrchu se děje na základě zpracování obrazové informace a dat z laserového dálkoměru, který je netradičně využit nejen k detekci překážek. Vyhodnocení měření každého senzoru probíhá nezávisle na sobě, a nedochází tudíž k nízkourovňové fúzi před samotným zpracováním. Kombinace této dvojice senzorů, respektive navržené detekční algoritmy, se ukázaly jako velice robustní řešení pro parková prostředí v letním období.

Globální navigace je značně zjednodušena existencí GPS a dodanou geometrickou mapou, jež slouží jako podkladový materiál pro definování jízdní trasy. Pojem lokalizace zde má tudíž význam nalezení se v mapě, neboli určení aktuálního segmentu cesty, po němž se vozidlo pohybuje. Navržená globální lokalizační metoda kombinuje jak metrickou, tak i topologickou informaci zanesenou do mapy. Spolehlivost lokalizace je navíc umocněna sloučením poziční informace z interního navigačního systému (lokální estimátor polohy) s pozicí získanou z GPS senzoru, a to pomocí fuzzy logiky.

Pouze běh robota v reálném prostředí může zaručit správnost a použitelnost uvedených metod, neboť samotná počítačová simulace není schopna postihnout všechny aspekty, které mohou ve skutečném okolí robota nastat. Proto byly všechny uvedené algoritmy ověřeny s úspěchem v praxi na reálném robotu jménem *Karlik* realizovaném pod záštitou projektu *Roboauto*, jenž obsadil 2. místo na mezinárodní soutěži Robotour 2009 (diplom viz příloha C). Od roku 2009 začal vývoj nové robotické platformy jménem *Quido*, která má stejný senzorický subsystém jako robot *Karlik*, avšak v kompaktnější formě, a proto mohla být řídicí architektura z *Karlika* nasazena takřka beze změny. Vývoj *Quida* byl ukončen v květnu roku 2010, kdy se téhož měsíce účastnil ryze české soutěže Robotem rovně aneb Autíčka v parku pořádané v Písku, v níž bezkonkurenčně obsadil první místo. Tým *Roboauto* propagující FIT VUT se tak zařadil ke špičce mezi týmy v České republice, jejichž cílem je vytvořit autonomně řízené vozidlo.

Literatura

- [1] *Guardian.co.uk* [online]. 2005-12-29 [cit. 2010-05-22]. Robot car: streets ahead in cities of the future. Dostupné z WWW: <<http://www.guardian.co.uk/uk/2005/dec/29/transport.highereducation>>
- [2] *Wikipedia* [online]. 2010-05-14 [cit. 2010-05-22]. Steam Horse locomotive. Dostupné z WWW: <http://en.wikipedia.org/wiki/Steam_Horse_locomotive>.
- [3] *Homepage for Prof. Schmidhuber* [online]. 2009-11-09 [cit. 2010-05-22]. Prof. Schmidhuber's highlights of robot car history. Dostupné z WWW: <<http://www.idsia.ch/~juergen/robotcars.html>>.
- [4] *Driverless car* [online]. 2010-05-21 [cit. 2010-05-22]. Wikipedia. Dostupné z WWW: <http://en.wikipedia.org/wiki/Driverless_car>.
- [5] *Darpa urban challenge* [online]. 2010-05-22 [cit. 2010-05-22]. Dostupné z WWW: <<http://www.darpa.mil/grandchallenge/index.asp>>.
- [6] *Urban Challenge Rules 2007* [online]. [s.l.] : [s.n.], 2007-09-27 [cit. 2010-05-22]. Dostupné z WWW: <http://www.darpa.mil/grandchallenge/docs/Urban_Challenge_Rules_102707.pdf>.
- [7] *Route Network Definition File (RNDF) and Mission Data File (MDF) Formats* [online]. [s.l.] : [s.n.], 2007-03-14 [cit. 2010-05-22]. Dostupné z WWW: <http://www.darpa.mil/GRAND-CHALLENGE/docs/RNDF_MDF_Formats_031407.pdf>.
- [8] *Wikipedia* [online]. 2010-05-20 [cit. 2010-05-22]. EUREKA Prometheus Project. Dostupné z WWW: <http://en.wikipedia.org/wiki/EUREKA_Prometheus_Project>.
- [9] *Elrob Website* [online]. 2010-05-22 [cit. 2010-05-22]. Dostupné z WWW: <<http://www.el-rob.org/imprint.html>>.
- [10] *Robotika.cz* [online]. 2010-05-22 [cit. 2010-05-22]. Robotour. Dostupné z WWW: <<http://robotika.cz/competitions/robotour/cs>>.
- [11] *Kufr.cz* [online]. 2010-05-22 [cit. 2010-05-22]. Pozor soutěž ROBOTEM ROVNĚ je již za dveřmi. Dostupné z WWW: <<http://www.kufr.cz/view.php?navezclanku=pozor-soutez-robotem-rovne-je-jiz-za-dvermi&cislocclanku=2010050002>>.
- [12] *Roboauto* [online]. 2009-04-10 [cit. 2010-05-22]. Dostupné z WWW: <www.roboauto.cz>.
- [13] SOTELO, M. A. *VIRTUOUS: VIsion-based Road Transport-system for Unmanned Operation on Urban-like Scenarios* [online]. [s.l.], 2004. 97 s. Dizertační práce. Departamento de Electrónica of the UAH. Dostupné z WWW: <<http://www.robosafe.com/repositorio/documentos/VIRTUOUS-PhDThesis.pdf>>.
- [14] ANDERSEN, J. Ch. *Mobile Robot Navigation* [online]. [s.l.], 2006. 212 s. Dizertační práce. Technical University of Denmark. Dostupné z WWW: <<http://orbit.dtu.dk/getResource?recordId=193297&objectId=1&versionId=1>>.
- [15] GRECO, Ch. R. *Real-time forward urban environment perception for an autonomous ground vehicle using computer vision and lidar* [online]. [s.l.], 2008. 106 s. Diplomová práce. Brigham Young University. Dostupné z WWW: <<http://contentdm.lib.byu.edu/ETD/image/etd2314.pdf>>.
- [16] NEGENBORN, R. *Robot Localization and Kalman Filters* [online]. [s.l.], 2003. 156 s. Diplomová práce. Utrecht university. Dostupné z WWW: <http://www.negenborn.net/kal_loc/thesis.pdf>.
- [17] ŠTĚPÁN, P. *Vnitřní reprezentace prostředí pro autonomní mobilní roboty* [online]. [s.l.], 2001. 126 s. Dizertační práce. ČVUT FEL. Dostupné z WWW: <<http://labe.felk.cvut.cz/~stepan/disertace.zip>>.
- [18] Karel Čapek [online]. 2008-06-10 [cit. 2010-05-22]. O původu slova robot. Dostupné z WWW: <<http://capek.misto.cz/robot.html>>.
- [19] EVERETT, H. R. . *Sensors for mobile robots: theory and application*. [s.l.] : [s.n.], 1995 . 528 s. ISBN 1-56881-048-2.
- [20] ORSÁG, F. *Studijní opora do předmětu ROB*. [s.l.] : [s.n.], 2006. 131 s.
- [21] NOVÁK, J. *Snímání a zpracování údajů lokalizace dopravního prostředku*. [s.l.], 2006. 65 s. Dizertační práce. FSI VUT v Brně.

- [22] KING, A.D. *Inertial Navigation – Forty Years of Evolution* [online]. [s.l.] : [s.n.], 1998 [cit. 2010-05-22]. Dostupné z WWW: <http://www.imar-navigation.de/download/inertial_navigation_introduction.pdf>.
- [23] JANOŠEK, M. *Magnetometr s miniaturními senzory fluxgate*. Praha, 2007. 85 s. Diplomová práce. ČVUT FEL.
- [24] *Robot Electronics* [online]. 2009-03-13 [cit. 2010-05-22]. CMPS03 - Compass Module. Dostupné z WWW: <<http://www.robot-electronics.co.uk/htm/cms3tech.htm>>.
- [25] RAPANT, P. *Družicové polohové systémy*. Ostrava : [s.n.], 2001. 202 s. Dostupné z WWW: <http://gis.vsb.cz/dokumenty/dns-gps/at_download/file>. ISBN 80-248-0124-8.
- [26] FRITZ, T. *APLIKACE S 3D LASEROVÝM DÁLKOMĚREM SICK*. [s.l.], 2009. 63 s. Diplomová práce. FSI VUT v Brně. Dostupné z WWW: <http://uai.fme.vutbr.cz/szz/2009/DP_Fritz.pdf>.
- [27] *Technical information LMS 220/221* [online]. [s.l.] : [s.n.], 2003-06-11 [cit. 2010-05-22]. Dostupné z WWW: <<http://www.sick-automation.ru/images/File/pdf/LMS%20220-221%20Tech%20Info.pdf>>.
- [28] GREPL, R. *Mechatronika - vybrané problémy* [online]. FSI, VUT v Brně : [s.n.], 2007 [cit. 2010-05-22]. Dostupné z WWW: <http://www.umat.fme.vutbr.cz/mechlab/publikace/Mechatronika_vybrane_problemy_2008.pdf>. ISBN 978-80-214-3804-0.
- [29] *Wikipedia* [online]. 2006-07-24 [cit. 2010-05-22]. Navigace. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/Navigace>>.
- [30] THRUN, S.; BURGARD, W.; FOX, D. *Probabilistic Robotics*. [s.l.] : MIT Press, 2006. 647 s. ISBN 0-262-20162-3.
- [31] BORENSTEIN, J.; EVERETT, H. R.; FENG, L. *Where Am I?* [online]. [s.l.] : [s.n.], 1996 [cit. 2010-05-22]. Dostupné z WWW: <<http://www.personal.umich.edu/~johannb/shared/pos96-rep.pdf>>.
- [32] MITCHELL, H. B. *Multi-Sensor Data Fusion: An Introduction* . [s.l.] : Springer, 2007. 282 s. ISBN 978-3540714637.
- [33] TEREJANU, G. A. . *Unscented Kalman Filter Tutorial* [online]. [s.l.] : [s.n.], 2009-05-18 [cit. 2010-05-22]. Dostupné z WWW: <www.acsu.buffalo.edu/~terejanu/files/tutorialUKF.pdf>.
- [34] WELCH, G.; BISHOP, G. *An Introduction to the Kalman Filter* [online]. [s.l.] : [s.n.], 2006 [cit. 2010-05-22]. Dostupné z WWW: <http://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf>.
- [35] TAMAS, L., et al. *Robot Motion and Control* [online]. [s.l.] : [s.n.], 2009 [cit. 2010-05-22]. Position Estimation Techniques for Mobile Robots, s. . Dostupné z WWW: <<http://www.springerlink.com/content/54505p250p48kqh2/>>.
- [36] JOSE, J. M. *Performance comparison of Extended and Unscented Kalman Filter implementation in INS-GPS integration* [online]. [s.l.], 2009. 78 s. Diplomová práce. Lulea University of Technology. Dostupné z WWW: <<http://epubl.ltu.se/1653-0187/2009/095/LTU-PB-EX-09095-SE.pdf>>.
- [37] THRUN, S., et al. *Robust Monte Carlo Localization for Mobile Robots*. [s.l.] : [s.n.], 2001. 49 s.
- [38] THRUN, S. *Robotic Mapping* [online]. [s.l.] : [s.n.], 2002 [cit. 2010-05-22]. Dostupné z WWW: <<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.9.7791>>.
- [39] KULICH, M. *Lokalizace a tvorba modelu prostředí v inteligentní robotice* [online]. [s.l.], 2003. 112 s. Dizertační práce. ČVUT FEL. Dostupné z WWW: <<http://labe.felk.cvut.cz/~kulich/diser-kulich.pdf>>.
- [40] ELFES, A. *Using Occupancy Grids for Mobile Robot Perception and Navigation*, Carnegie Mellon University [online]. Carnegie Mellon University : [s.n.], 1989 [cit. 2010-05-22]. Dostupné z WWW: <http://www.cs.cmu.edu/~biorobotics/motion/www/papers/sbp_papers/integrated4/elfes_occup_grids.pdf>.
- [41] *Wikipedia* [online]. 2009-06-10 [cit. 2010-05-22]. Dempster-Shafer theory. Dostupné z WWW: <http://en.wikipedia.org/wiki/Dempster%E2%80%93Shafer_theory>.
- [42] ORIOLO, G.; ULIVI, G.; VENDITTELLI, M. *Fuzzy Maps: A New Tool for Mobile Robot Perception and Planning* [online]. [s.l.] : [s.n.], 1997 [cit. 2010-05-22]. Dostupné z WWW: <www.dis.uniroma1.it/~labrob/pub/papers/JRS97.ps.gz>.

- [43] MONTEMERLO, M.; THRUN, S. *A multi-resolution pyramid for outdoor robot terrain perception* [online]. Nineteenth National Conference on Artificial Intelligence. [s.l.] : [s.n.], 2004 [cit. 2010-05-22]. Dostupné z WWW: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.79.8979&rep=rep1&type=pdf>>.
- [44] PETERSON, K.; ZIGLAR, J.; RYBSKI, P. E. *Fast Feature Detection and Stochastic Parameter Estimation of Road Shape using Multiple LIDAR* [online]. Carnegie Mellon University : [s.n.], 2008 [cit. 2010-05-22]. Dostupné z WWW: <http://www.ri.cmu.edu/pub_files/2008/9/peterson_kevin_2008_1.pdf>.
- [45] KIRCHNER, A.; HEINRICH, Th. *Model based detection of road boundaries with a laser scanner* [online]. IEEE International Conference on Intelligent Vehicles. [s.l.] : [s.n.], 1998 [cit. 2010-05-22]. Dostupné z WWW: <<http://citeseerx.ist.psu.edu/viewdoc/download?sessionid=D2226A64B8266B41E34122CACB0DAEF6?doi=10.1.1.32.9888&rep=rep1&type=pdf>>.
- [46] CREMEAN, L. B.; MURRAY, R. M. *Model-Based Estimation of Off-Highway Road Geometry using Single-Axis LADAR and Inertial Sensing* [online]. 2006 International Conference on Robotics and Automation. [s.l.] : [s.n.], 2006 [cit. 2010-05-22]. Dostupné z WWW: <http://www.cds.caltech.edu/~murray/papers/2005u_cm06-icra.html>.
- [47] CRAMER, H.; WANIELIK, G. *Road border detection and tracking in non cooperative areas with a laser radar system* [online]. Chemnitz University of Technology : [s.n.], 2002 [cit. 2010-05-22]. Dostupné z WWW: <http://www.tu-chemnitz.de/etit/nt/home/images/stories/forschungsbereiche/Detektion/cr_grs2002.pdf>.
- [48] LIN, L.; ZHOU, W. *Advanced Intelligent Computing Theories and Applications. With Aspects of Theoretical and Methodological Issues* [online]. [s.l.] : Springer Berlin / Heidelberg, 2007 [cit. 2010-05-22]. A Robust and Adaptive Road Following Algorithm for Video Image Sequence, s. . Dostupné z WWW: <<http://www.springerlink.com/content/3664615108501663/>>. ISBN 978-3-540-74170-1.
- [49] ÁLVAREZ, J. M.; LÓPEZ, A. M.; BALDRICH, R. *Pattern Recognition and Image Analysis* [online]. [s.l.] : Springer Berlin / Heidelberg, 2007 [cit. 2010-05-22]. Shadow Resistant Road Segmentation from a Mobile Monocular System , s. . Dostupné z WWW: <<http://www.springerlink.com/content/m2376t08j772wtr7/>>. ISBN 978-3-540-72848-1.
- [50] CERVANTES, G. A.; DEVY, M. ; HERNÁNDEZ, A. M. *Lane Extraction and Tracking for Robot Navigation in Agricultural Applications* [online]. Proceedings of ICAR 2003 The 11th International Conference on Advanced Robotics. [s.l.] : [s.n.], 2003 [cit. 2010-05-22]. Dostupné z WWW: <<http://www.uv.mx/anmarin/papers/ICAR03-816.pdf>>.
- [51] FOEDISCH, M. *Adaptive Real-Time Road Detection Using Neural Networks* [online]. in Proc. 7th Int. Conf. on Intelligent Transportation Systems, Washington D.C . [s.l.] : [s.n.], 2004 [cit. 2010-05-22]. Dostupné z WWW: <<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.-64.3651>>.
- [52] AUFRERE, R.; CHAPUIS, R.; CHAUSSE, F. A model-driven approach for real-time road recognition. *Machine Vision and Applications* [online]. 2001, Volume 13, Number 2, [cit. 2010-05-22]. Dostupný z WWW: <<http://www.springerlink.com/content/pbgr6bcvbg64yuh/>>.
- [53] BRADSKI, G.; KAEHLER, A. *Learning OpenCV*. [s.l.] : O, 2008. 576 s. ISBN 978-0-596-15620-6.
- [54] KELLY, A. *A 3D State Space Formulation of a Navigation Kalman Filter for Autonomous Vehicles* [online]. Pittsburgh, PA 15213 : The Robotics Institute Carnegie Mellon University, 1994, 2006 [cit. 2010-05-22]. Dostupné z WWW: <http://www.frc.ri.cmu.edu/~alonzo/pubs/reports/kalman_V2.pdf>.
- [55] HAVLENA, V.; ŠTECHA, J. *Moderní teorie řízení*. [s.l.] : FEL ČVUT, 2000. 304 s. Dostupné z WWW: <http://support.dce.felk.cvut.cz/e-kurzy/file.php/15/prednasky/mtr_old.pdf>. ISBN 80-01-02095-9.
- [56] PARK, S. B., et al. *ProFusion2 – Sensor Data Fusion for Multiple Active Safety Applications* [online]. 13th World Congress & Exhibition on Intelligent Transport Systems and Services. ExCel London, United Kingdom : [s.n.], 2006 [cit. 2010-05-22]. Dostupné z WWW:

- <http://www.prevent-ip.org/download/Events/20061008-12_ITS_WC_London/TS103/Paper%201711.pdf>.
- [57] MONTEIRO, T., et al. Robust mobile robot map building using sonar and vision: Evolving the navigation abilities of the ApriAlpha home robot. *Nippon Kikai Gakkai Robotikusu* [online]. 2005, 2005, [cit. 2010-05-22]. Dostupný z WWW: <http://www-personal.acfr.usyd.edu.au/sildomar/files/Monteiro_ROBOMECH_2005.pdf>.
 - [58] PECH, J. *Statistické odhady parametrů* [online]. [s.l.] : [s.n.], 2001 [cit. 2010-05-22]. Dostupné z WWW: <<http://noel.feld.cvut.cz/vyu/tss/stat.pdf>>.
 - [59] HARTIKAINEN, J.; SÄRKKÄ, S. *Optimal filtering with Kalman filters and smoothers – a Manual for Matlab toolbox EKF/UKF* [online]. [s.l.] : [s.n.], 2008 [cit. 2010-05-22]. Dostupné z WWW: <www.lce.hut.fi/research/mm/ekfukf/documentation.pdf>.
 - [60] MERWE, R.; WAN, E. A. *THE SQUARE-ROOT UNSCENTED KALMAN FILTER FOR STATE AND PARAMETER-ESTIMATION* [online]. Oregon Graduate Institute of Science and Technology : [s.n.], 2001 [cit. 2010-05-22]. Dostupné z WWW: <<http://speech.bme.ogi.edu/publications/ps/merwe01a.pdf>>.
 - [61] JULIER, S. J.; UHLMANN, J. K. *Unscented Filtering and Nonlinear Estimation* [online]. PROCEEDINGS OF THE IEEE, VOL. 92, NO. 3., [s.l.] : [s.n.], 2004 [cit. 2010-05-22]. Dostupné z WWW: <www.gatsby.ucl.ac.uk/~byron/nlds/julier04.pdf>.
 - [62] CHOW, S. M.; FERRER, E.; NESSELROADE, J. R. *An Unscented Kalman Filter Approach to the Estimation of Nonlinear Dynamical Systems Models* [online]. [s.l.] : [s.n.], 2007 [cit. 2010-05-22]. Dostupné z WWW: <http://psychology.ucdavis.edu/labs/ferrer/pubs/mbr_inpress.pdf>.
 - [63] CUNHA, F. *The Unscented Kalman Filter Code* [online]. University of Pennsylvania : [s.n.], 2009 [cit. 2010-05-22]. Dostupné z WWW: <http://athens.src.uchicago.edu/jenni/flavio/Code/mnukf_01-08-2009_fc.pdf>.
 - [64] ŠTECHA, J.; HAVLENA, V. *Teorie dynamických systémů* [online]. [s.l.] : FEL ČVUT, 2005 [cit. 2010-05-22]. Dostupné z WWW: <<http://dce.felk.cvut.cz/tds/ots.pdf>>.
 - [65] WANG, J. H.; GAO, Y. MULTI-SENSOR DATA FUSION FOR LAND VEHICLE ATTITUDE ESTIMATION USING A FUZZY EXPERT SYSTEM. In *Data Science Journal* [online]. [s.l.] : [s.n.], 2005 [cit. 2010-05-22]. Dostupné z WWW: <http://journals.eecs.qub.ac.uk/codata/journal/contents/4_05/4_05pdfs/DS427.pdf>.
 - [66] *Wikipedia* [online]. 2007-04-18, 2010-02-22 [cit. 2010-05-22]. UTM. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/UTM>>.
 - [67] *GPSy* [online]. 2002-08-28 [cit. 2010-05-22]. Converting Latitude/Longitude to Universal Transverse Mercator (UTM). Dostupné z WWW: <<http://www.gpsy.com/gpsinfo/geotoutm/>>.
 - [68] BERNSTEIN, D.; KORNHAUSER, A. *An Introduction to Map Matching for Personal Navigation Assistants* [online]. Princeton University : [s.n.], 1996 [cit. 2010-05-22]. Dostupné z WWW: <<http://www.njtide.org/reports/mapmatchintro.pdf>>.
 - [69] SCHUESSLER, N.; AXHAUSEN, K. W. *Map-matching of GPS traces on high-resolution navigation networks using the Multiple Hypothesis Technique (MHT)* [online]. Zurich : ETH Zurich, 2009 [cit. 2010-05-23]. Dostupné z WWW: <<http://www.ivt.ethz.ch/vpl/publications/reports/ab568.pdf>>.
 - [70] QUDDUS, M. A., et al. *Current map-matching algorithms for transport applications: State-of-the art and future research directions* [online]. [s.l.] : [s.n.], 2006 [cit. 2010-05-23]. Dostupné z WWW: <linkinghub.elsevier.com/retrieve/pii/S0968090X07000265>.
 - [71] HONEY, S. K. . *Vehicle Navigational System and method : United States Patent number 4.796.191* [online]. [s.l.] : [s.n.], 1989 [cit. 2010-05-23]. Dostupné z WWW: <http://ieeeghn.org/wiki/images/0/0f/Vehicle_navigational_system_and_method.pdf>.
 - [72] OCHIENG, W. Y.; QUDDUS, M. A.; NOLAND, R. B. Map-Matching in Complex Urban Road Networks. In *Brazilian Journal of Cartography* [online]. [s.l.] : [s.n.], 2003 [cit. 2010-05-23]. Dostupné z WWW: <<http://www.cts.cv.ic.ac.uk/html/ResearchActivities/publicationDetails.asp?PublicationID=373>>.

*„In theory, there is no difference between theory and practice.
In practice, there is.“
Jan L.A. Van de Snepscheut*

Seznam příloh

Příloha A. Softwarová architektura.

Příloha B. Metoda plánování pohybu.

Příloha C. Diplom za 2. místo v soutěži Robotour 2009.

Příloha D. DVD obsahující elektronickou verzi technické zprávy, demonstrační video, spustitelnou aplikaci včetně připravených záznamů a zdrojové soubory programu.

Příloha A Softwarová architektura

Cílem této přílohy je uvést skutečně používanou aplikační architekturu robota, tj. rozdělení úloh do jednotlivých nepřímo závislých programů, které spolu komunikují skrze protokol TCP/IP. Ačkoliv je nyní komunikace mezi programy stále ještě jednosměrná, nemusí tomu tak nutně být. Do budoucna se počítá s využitím zpětné vazby od programu tvořícího lokální mapu k samotným detektorům atd.

Role jednotlivých aplikací

Dále je stručně charakterizována úloha jednotlivých aplikací schématu v systému (viz obr. 25). Poznamenejme, že uvedená funkcionalita není striktní, neboť může být vhodné některá data využívat souběžně i v jiných modulech než bylo původně dáno. Jedná se proto spíše o prvotní vzhled, jenž se odráží v názvosloví a reflektuje primární cíle samotných aplikací.

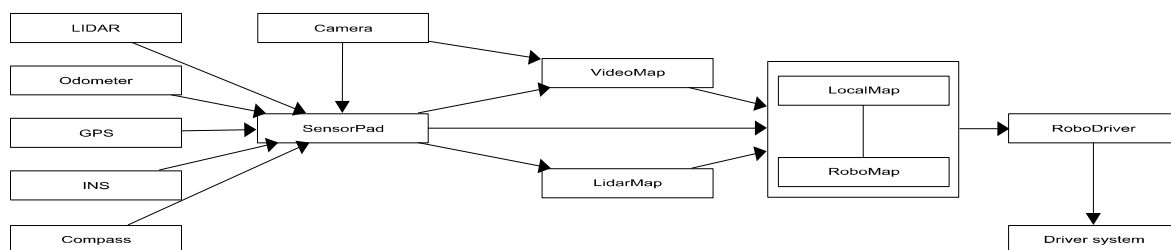
SensorPad je aplikace, která představuje kolektor senzorických měření. Zajišťuje připojení senzorů, získávání dat ze senzorů a indikuje aktuální stav senzorů robota (jednoduchá diagnostika). Dále dovoluje data zaznamenávat a zpětně je přehrávat. Zároveň slouží jako distributor dat přes protokol TCP/IP dalším programům, jež si registrují, která data chtějí přijímat³¹.

LidarMap odebírá data z laserového dálkoměru a detekuje nad nimi sjízdný a nesjízdný povrch. Výsledky detekce pak předává přes protokol TCP/IP aplikaci tvořící lokální mapu *LocalMap*. Součástí datového balíčku je rovněž informace o poloze robota ku přijatému skenu pro správné pozicování výsledku detekce do lokální mapy.

VideoMap, podobně jako *LidarMap*, slouží pro detekci cesty, tentokrát však z obrazových dat. Kvůli rychlosti je obraz z kamery přenášen do aplikace přímo. *SensorPadu* jsou zpětně posílány pouze indexy snímků s časovými razítky (tvorba záznamu). Výsledek detekce je stejně jako v předchozím případě předán aplikaci *LocalMap* skrze síťovou komunikaci.

LocalMap a *RoboMap* představují kognitivní složku řízení robota. Úkolem *LocalMapu* je sestavovat lokální mapu robota na základě přijímaných dat z detektorů a provádět lokální řízení, tedy i promptně reagovat na nečekané situace. *RoboMap* zodpovídá za dosažení globálního cíle. Interní reprezentace prostředí modeluje globální mapu, nad níž probíhá proces plánování s využitím údajů z GPS a dat získaných z RNDF souboru.

RoboDriver zajišťuje spojení s výkonovou elektronikou vozidla, přijímá požadavky na řízení, které následně realizuje. Jedná se o prioritní systém, který umí přijímat pokyny z více zdrojů a upřednostnit kritický požadavek (např. zabránění srážce má vyšší prioritu než samotné inteligentní řízení).



Obrázek 25: Softwarová architektura.

31 Samotný popis komunikačního protokolu by vydal na několik stránek, proto není součástí této práce.

Příloha B Metoda plánování pohybu

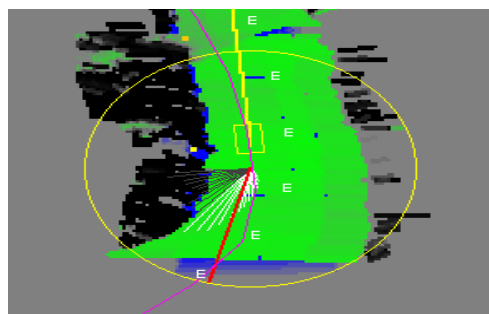
Ačkoliv postihnout problematiku plánování pohybu (vykonání plánu) nebylo cílem této práce, byla by škoda neucelit celý koncept navigační architektury a neuvést taktéž metodu plánování pohybu, jež byla použita na soutěži Robotour 2009.

Metody plánování pohybu, nebo možná lépe metody hledání cesty ve vnitřní reprezentaci, jsou spjaty s volbou vnitřního modelu okolí robota. Mezi algoritmy, které nabízí možnost uplatnit věrohodností informaci uloženou v rastrové reprezentaci *in situ* patří například metoda dynamického programování, metoda potenciálových polí (s drobnou úpravou), paprskový algoritmus aj.

Paprsková metoda

Tento algoritmus je relativně jednoduchou metodou hledání cesty nad rastrovou mapou, která před robota umisťuje ve vhodné vzdálenosti úhlově omezený bodový zdroj tzv. paprsků. Každý paprsek definuje směr možného pohybu robota a jeho ohodnocení značí výhodnost pohybu tímto směrem. Implicitním cílem paprskové metody je proto dopředný pohyb robota.

Mezi hlavní výhody patří nízká výpočetní náročnost a vysoká škálovatelnost. Nevýhodu představuje samotný tvar paprsku (přímkový tvar), neboť jednak nereflektuje manévrovací schopnosti robota a jeho prostorovou složitost, a jednak nemusí v některých situacích na mapě nalézt optimální řešení. Nedostatky lze odstranit například souborem tzv. manévru robota a stanovením šířky zkoumané trajektorie. Nicméně dále uvažujme metodu ve svém původním znění a pro zjednodušení navíc zavedme předpoklad, že cesta nemá konce (nebudeme řešit problém zastavení ve snaze zprůhlednit formální zápis).



Obrázek 26: Paprsková metoda.

Úspěšnost této metody tkví ve stanovení způsobu ohodnocení dílčích paprsků a dále ve výběru nejlepšího kandidáta. Funkci ohodnocení paprsku délky h a úhlu α formálně definujeme jako:

$$fitness(h, \alpha) = \begin{cases} 0, \exists cell(i) < 0, i \in trajectory(h, \alpha) \\ \sum_{i \in trajectory(h, \alpha)} cell(i), otherwise \end{cases} \quad (41)$$

kde $trajectory(h, \alpha)$ je množina indexů mřížky obsazenosti v trajektorii paprsku z bodového zdroje. Optimální směr pohybu je dán maximalizací zisku dle:

$$\alpha^* = argmax_{\alpha} \left(\begin{cases} \sum_{j=-1}^{j=1} fitness(h, \alpha_{i+j}), \neg \exists fitness(h, \alpha_{i+j}) \leq 0 \\ 0, otherwise \end{cases} \right) \quad (42)$$

Obrázek č. 26 ilustruje paprskovou metodu detekce cesty na rastrové mapě. Navíc je zde zvýhodněna levá strana natočením úhlově omezeného bodového zdroje paprsků dle globálního cíle (korektní odbočení na křižovatce). Paprsek zvýrazněný červenou barvou signalizuje zvolený směr pohybu robota.

Příloha C Diplom za 2. místo v Robotour 09

