

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická

Diplomová práce



Bc. Vojtěch Šalanský

Kontaktní průzkum terénu pro mobilního robota

Katedra kybernetiky

Vedoucí práce: **Ing. Karel Zimmermann, Ph.D.**

PRAHA 2015

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: Bc. Vojtěch Šalanský
Studijní program: Kybernetika a robotika (magisterský)
Obor: Robotika
Název tématu: Kontaktní průzkum terénu pro mobilního robota

Pokyny pro vypracování:

V Centru strojového vnímání (CMP) na katedře kybernetiky je v rámci Evropského projektu TRADR vyvíjen robot záchranář (<http://cmp.felk.cvut.cz/demos>). Jednou z důležitých funkcionalit je odhad tvaru terénu před robotem v případě, kdy ostatní senzory terén nevidí (např. kvůli hustému kouři, velkému náklonu robota či velké odrazivosti povrchu). Navrhněte a prakticky ověřte algoritmus pro kontaktní průzkum terénu pomocí robotické ruky (Jaco Kinova Arm, <http://kinovarobotics.com>).

1. Navrhněte a naimplementujte algoritmus realizující inverzní kinematickou úlohu v prostředí s překážkami reprezentovanými shluky 3D bodů (point cloud). Využijte balíčky z prostředí ROS (www.ros.org).
2. Naimplementujte funkci, která zjistí výšku a tuhost terénu na daných souřadnicích v souřadném systému roviny robota.
3. Nastudujte algoritmy pro aktivní vidění a 3D rekonstrukci terénu jako je např. [1].
4. Naimplementujte a experimentálně ověřte algoritmus, který prozkoumá chybějící část terénu před robotem. Navržený algoritmus musí zohlednit již známá měření jako jsou náklon robota nebo proudy ve flipperech (např. použitím Gibbsova samplování [2]).
5. Výsledný algoritmus otestujte v hasičském tréninkovém centru pro Urban Search & Rescue mise (přístup zajistí školitel během review meetingu projektu TRADR).

Seznam odborné literatury:

- [1] M. Bjorkman, Y. Bekiroglu, V. Hogman, D. Kragic: Enhancing Visual Perception of Shape through Tactile Glances Intelligent Robots and Systems. IEEE/RSJ International Conference on, 2013.
- [2] C. M. Bishop: Pattern Recognition and Machine Learning (Information Science and Statistics). Secaucus, NJ, USA:Springer-Verlag New York, Inc., 2006.

Vedoucí diplomové práce: Ing. Karel Zimmermann, Ph.D.

Platnost zadání: do konce letního semestru 2015/2016

L.S.

doc. Dr. Ing. Jan Kybic
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 23. 1. 2015

Prohlášení autora práce

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

V Praze dne

.....

Podpis autora práce

Poděkování

Rád bych touto cestou poděkoval mému vedoucímu práce Ing. Karlovi Zimmermannovi, Ph.D. za pomoc při vedení diplomové práce a za věcné připomínky při řešení jednotlivých úloh. Poděkování patří i mé rodině za vytvoření příjemného prostředí pro psaní mé práce a za podporu během celého studia.

Abstrakt

Důležitou funkcionalitou pro mobilní robot je znalost terénu v nejbližším okolí. Tato práce se zabývá návrhem a implementací metody ke zjištění tvaru terénu v okolí mobilního robotu využitím kontaktu robotické ruky Kinova Jaco Arm s terénem. V rámci implementace metody kontaktního průzkumu je v této práci řešeno více jednotlivých úloh, jako je detekce kontaktu robotické ruky s terénem, výběr místa z pracovního prostoru k prozkoumání, odhad výšky v neprozkoumaných místech a tvorba mapy prostředí. Kontakt robotické ruky s terénem je detekován využitím momentů sil v jednotlivých kloubech. Pro volbu místa k prozkoumání byla navržena metoda vybírající místo, u kterého očekáváme, že bude nejvíce užitečné pro odhad okolních bodů. Využitím prozkoumaných míst v terénu a vnitřních senzorů robotu (náklon, proudové odběry motorů) je tvořen model okolního prostředí robotu. Navržená metoda kontaktního průzkumu využívá měřených a odhadnutých výšek terénu pro tvorbu mapy.

Klíčová slova

mobilní robotika, aktivní vnímání, haptická explorace, Gibbsovo vzorkování, propioceptivní mapování, ROS, MoveIt!

Abstract

For mobile robots, it is important to know the surrounding terrain. The goal of this diploma thesis is to provide a design and prototypical implementation of the method for terrain exploration via contact of robotic manipulator Kinova Jaco Arm with the terrain. There are more subgoals to be done within the implementation of contact exploration. This thesis deals with the contact detection of the robotic arm and the terrain, choosing the place from workspace for exploration, and the estimation of unknown places and mapping. The contact of the manipulator with the terrain is detected by using joint torques of actuators. The place that should be explored next is chosen to increase the expected usefulness for future estimation of the other unknown places. The places that are not explored are estimated by using proprioceptive sensors (tilt, currents in actuators). A method that saves and process the map from measured and estimated places is also provided in this thesis.

Keywords

mobile robotics, active perception, haptic exploration, Gibbs sampling, proprioceptive mapping, ROS, MoveIt!

Obsah

1	Úvod	1
2	TRADR projekt	3
2.1	Mobilní robot BlueBotics Absolem	3
2.2	Manipulator Kinova Jaco Arm	4
2.2.1	Pracovní nástroj	5
3	Aktivní vnímání	6
4	Teorie	10
4.1	Základní pojmy kinematiky robotů	10
4.2	Statické síly v manipulátoru	13
4.3	Odometrie	14
4.4	ROS	14
4.5	MoveIt!	15
4.6	DEM	17
4.7	Vektor příznaků	18
4.8	Gibbsovo vzorkování	19
4.9	Výpočet podmíněných pravděpodobností metodou maximální věrohodnosti	21
4.10	Algoritmus pro aktivní vnímání	24
5	Implementace	26
5.1	Rozbor zadaného problému	26
5.2	Robot	28
5.3	Manipulátor	29
5.3.1	Inverzní kinematická úloha v prostředí s překážky	30
5.3.2	Servis pro zjištění výšky terénu	32
5.4	Matlab pipeline	34
5.4.1	Data	35
5.4.2	ROS odběratelé	35

5.4.3	Pseudoservis	37
5.4.4	Aktivní vnímání	38
5.4.5	Gibbsovo vzorkování	40
5.4.6	Proprioceptivní mapování	42
6	Experimenty	45
6.1	Přesnost měření výšky pomocí manipulátoru	45
6.2	Klasifikace tuhosti terénu	46
6.3	Užitečnost algoritmu pro aktivní vnímání	49
6.4	Rekonstrukce DEM	50
6.5	Rekonstrukce mapy	57
7	Shrnutí realizace a budoucí práce	60
8	Závěr	62

Seznam obrázků

1	Mobilní robot využívaný v této práci.	2
2	Postraní schéma robotu [1]	3
3	Konfigurace mobilního robotu. [2]	4
4	Schéma Jaco Arm [3]	5
5	Schémata aktivního vnímání	6
6	Architektura MoveIt! [4]	16
7	Umístění robotu v DEM	18
8	Schéma souboru programů.	27
9	Plánování pohybu využitím "MoveIt!".	31
10	Příklad kartézských sil při prohledávání šesti různých souřadnic v DEM pro tvrdý povrch (nahore) a pro měkký povrch (dole).	33
11	Pracovní prostor manipulátoru v DEM	38
12	Příklad hodnot spočítaných při dvou různých situacích.	39
13	Ukládání pole DEM do mapy.	43
14	Měřené objekty.	46
15	Velikosti změřených sil ve směru z na polích DEM z pracovního prostoru manipulátoru.	48
16	Porovnání průměrné chyby v odhadu výšky v poli DEM implementovaného algoritmu a náhodného generátoru	50
17	Situace, při které byl naměřen vektor příznaků, který je zobrazený na Obrázku 18. Robot je na rovince. Hodnoty proudů i naklonění jsou nepatrné.	51
18	Rekonstrukce DEM a známe hodnoty vektoru příznaků v situaci zobrazené na Obrázku 17.	51
19	Situace, při které byl naměřen vektor příznaků, který je zobrazený na Obrázku 20. Před robotem je schod dolů, který je prozkoumán pomocí jedné akce manipulátoru.	52
20	Rekonstrukce DEM a známe hodnoty vektoru příznaků v situaci zobrazené na Obrázku 19.	52
21	Situace, při které byl naměřen vektor příznaků, který je zobrazený na Obrázku 22. Robot je na rovince a dotýká se předními rameny schodu.	53

22	Rekonstrukce DEM a známe hodnoty vektoru příznaků v situaci zobrazené na Obrázku 21.	53
23	Situace, při které byl naměřen vektor příznaků, který je zobrazený na Obrázku 24. Robot je na rovině, jeho přední ramena se opírají o schod a je prováděna haptická explorace jednoho pole.	54
24	Rekonstrukce DEM a známe hodnoty vektoru příznaků v situaci zobrazené na Obrázku 23.	54
25	Situace, zobrazující nájezd na paletu, bez haptické explorace.	55
26	Rekonstrukce DEM a známe hodnoty vektoru příznaků v situaci zobrazené na Obrázku 25.	55
27	Situace, zobrazující nájezd na paletu, s haptickou explorací.	56
28	Rekonstrukce DEM a známe hodnoty vektoru příznaků v situaci zobrazené na Obrázku 27.	56
29	Skutečné prostředí, které bylo rekonstruováno.	57
30	Chyba rekonstrukce přejetí palety v závislosti na počtu prohledaných polí.	58
31	Rekonstrukce přejetí palety při osmi prohledaných polích.	58

Seznam tabulek

1	Uspořádání DEM ve vektoru příznaků.	19
2	Uspořádání sensorických dat ve vektoru příznaků.	19
3	Uspořádání informace o počtu bodů při tvorbě DEM ve vektoru příznaků.	19
4	Tabulka obsahující odchylky měření výšky od reálných hodnot. Kde z je měřená výška vztažená k podvozku a $z_{v1} = -0,154 m$, $z_{v2} = 0 m$, $z_{v3} = 0,154 m$, $z_{v4} = 0,264 m$ jsou referenční výšky terénu vůči podvozku.	45
5	Měření kontaktu robotického manipulátoru s měkkými objekty.	47
6	Měření kontaktu robotického manipulátoru s tvrdými objekty.	47
7	Obsah CD	65

Seznam algoritmů

1	Gibbsovo vzorkování	21
2	Odběratel vektoru příznaků.	37
3	Gibbsovo vzorkování	41

1 Úvod

Mobilní robotika je obor, který je v současnosti stále široce rozvíjen. Jedná se o doménu robotiky pracující s roboty, které se mohou pohybovat. Pohyb mobilních robotů může být po zemi pomocí pásů, kol, nohou nebo také vzduchem či vodou. Využití mobilních robotů je opravdu široké. Některé mobilní roboty slouží například ve skladech k přesunu zboží (Amazon), jiné slouží k práci na nebezpečných místech (pyrotechnický robot) nebo špatně dostupných místech (explorace Marsu). Úloh, které se mobilní roboty snaží řešit, je tedy mnoho. Od manipulace s objekty, po průzkum terénu. Při průzkumu terénu je využíváno sensorických měření, která jsou uložena do mapy. Častými senzory pro tvorbu mapy jsou laserový dálkoměr, stereo kamera nebo ultrazvuk. Existují však případy, kdy je průzkum terénu řešen jinými způsoby. Příkladem může být kontaktní průzkum terénu.

Při kontaktním průzkumu je ke sběru informací o prostředí využito, jak již název napovídá, kontaktu robotu nebo jeho sensorů s prostředím. Využití kontaktních sensorů má oproti dálkovým sensorům tu výhodu, že je možné zjistit dodatečné informace o zkoumaném terénu nebo objektu, jako je například tuhost nebo struktura. Při porovnání s bezkontaktní tvorbou modelu prostředí je kontaktní průzkum často pomalejší a zisk informací je pouze lokální. Přesto je kontaktního průzkumu využíváno a to například v situacích, kdy ostatní senzory terén nevidí nebo k doplnění informací o zkoumaném objektu.

V rámci evropského projektu TRADR (kapitola 2) je vyvíjen robot záchranář, zobrazený na Obrázku 1. Tento robot by měl ve spolupráci s lidmi pomáhat v záchranných misích při katastrofických scénářích. Pro bezpečný pohyb robotu je důležité znát terén před ním. Může se stát, že terén před robotem není vidět pomocí ostatních sensorů. Taková situace nastává například pokud je před robotem vysoce odrazivý objekt (kaluž), pokud je robot nakloněn tak, že terén pod ním není v zorném úhlu sensorů nebo pokud je celý prostor silně zadýmený. V těchto situacích je k doplnění informace o terénu v okolí robotu užitečný právě kontaktní průzkum. Robot vyvíjený v rámci projektu TRADR má k dispozici pracovní nástroj, kterým je robotický manipulátor Kinova Jaco Arm. Využitím tohoto manipulátoru je možné kontaktně prozkoumat prostředí.

Cílem této diplomové práce byl návrh a implementace algoritmu, který dokáže využitím kontaktního průzkumu, zmapovat prostředí před robotem. K tomu bylo potřeba implementovat program zajišťující inverzní kinematiku a program pro získání výšky v daném místě. Bylo by časově náročné prohledávat všechny neznámé body v okolí robotu. Z tohoto důvodu, jsou k prozkoumání vybírány jen některé body. Pro určení vhodného bodu k prozkoumání je využito metody aktivního vnímání, založené na předpokládané míře užitečnosti jednotlivých bodů. Pro modelování bodů, které nebyly prozkoumány, byla použita metoda Gibbsova vzorkování, která pro odhad výšek neznámého terénu využívá známé výšky terénu v jiných místech a sensorická měření, jako je naklonění robotu nebo proudy

1. ÚVOD

v jednotlivých motorech. Odhadovaná i naměřená data jsou ukládána do mapy, která představuje model prostředí.

Při tvorbě této diplomové práce bylo čerpáno hlavně ze zahraniční literatury. Práce je rozvržena do několika kapitol. V kapitole 2 je popsán projekt TRADR, využívaný mobilní robot a manipulátor. V kapitole 3 jsou uvedeny obecné přístupy k aktivnímu vnímání a konkrétní řešení některých úloh aktivního vnímání ve světě. V kapitole 4 je uveden teoretický základ využívaný při implementaci. V této kapitole jsou popsány základní pojmy kinematiky robotů, knihovny využívané pro implementaci inverzní kinematiky a celkového programu, struktury využívané při implementaci, algoritmus Gibbsova vzorkování a metoda pro aktivní vnímání. Implementace celého programu je popsána v kapitole 5. V kapitole 6 jsou popsány provedené experimenty, dokazující funkčnost programu. Nedostatky implementace a plány na rozšíření algoritmu jsou uvedeny v kapitole 7.



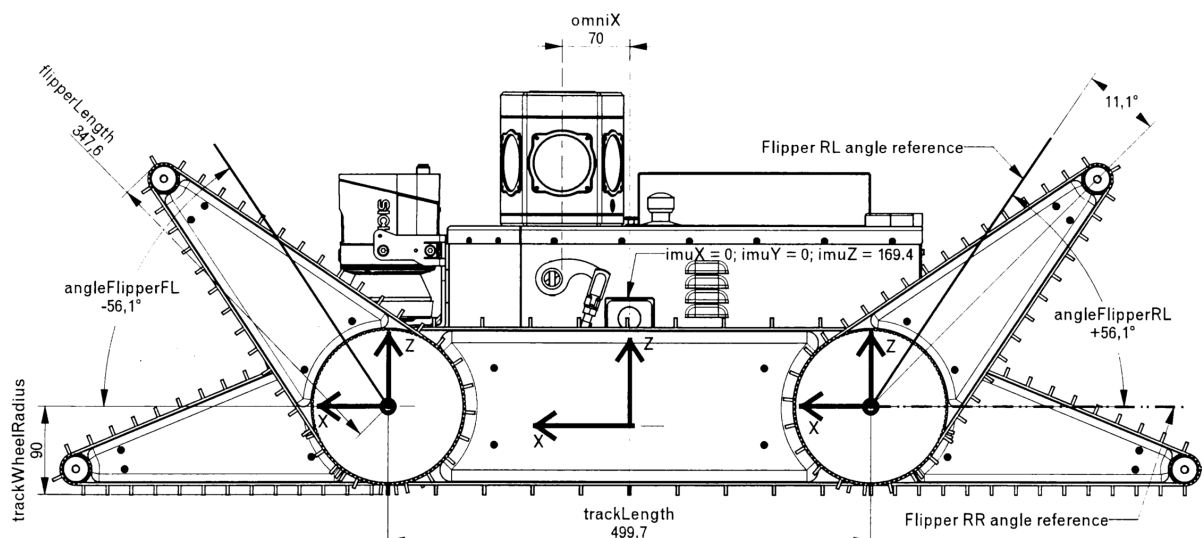
Obrázek 1: Mobilní robot využívaný v této práci.

2 TRADR projekt

TRADR [5, 6] (zkratka anglického Teaming for Robot-Assisted Disaster Response) je integrovaný výzkumný EU FP7 projekt, navazující na výzkum a zkušenosti z projektu NIFTi [1]. Na tomto projektu spolupracuje dvanáct partnerů z šesti různých zemí. Projekt TRADR se soustředí na spolupráci týmů záchranářů tvořených lidmi a roboty různých typů. Projekt se soustředí hlavně na metody spolupráce lidí a robotů. Tyto týmy by měli pomáhat při katastrofických scénářích jako je například zemětřesení. Takovéto scénáře obvykle trvají několik dní. Roboty mají ve spolupráci s lidmi vytvářet mapu prostředí, sbírat vzorky prostředí a vyhledávat oběti katastrof. Projekt TRADR si dává za cíl, aby tým postupně rozvíjel své znalosti o prostředí v oblasti katastrofy. Tým záchranářů tím pak získává informace, jakým způsobem mají pracovat v této lokalitě. Projekt pracuje s více robotickými platformami. V této diplomové práci bylo pracováno pouze s mobilním robotem popsáným níže.

2.1 Mobilní robot BlueBotics Absolem

V projektu TRADR je využíván mobilní robot od společnosti BlueBotics. Tento robot byl vytvořen pro projekt NIFTi a v projektu TRADR došlo zatím jen k menším konstrukčním změnám. Jedná se o mobilní robot s diferenčním typem podvozku opatřeným dvěma hlavními pásy. Kromě těchto hlavních pásů je robot opatřen čtyřmi vedlejšími pásy které jsou umístěné na takzvaných ramenech (flipperech). Tato ramena jsou pohyblivá, což umožňuje robotu zdolávat i složitější překážky, jako jsou například schody. Dále je ro-

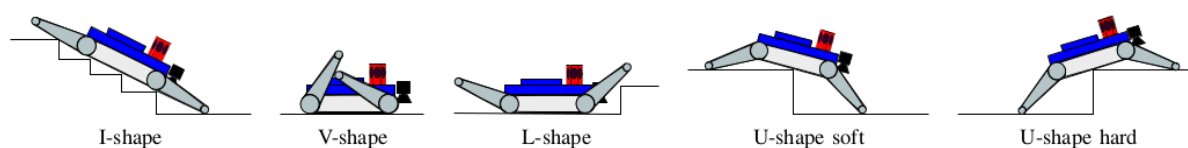


Obrázek 2: Postraní schéma robotu [1]

bot opatřen senzory pro vytváření modelu prostředí. Jedním z nich je rotační laser, který poskytuje 3D scan prostředí. Dále je na robotu umístěna všesměrová kamera, umožňující operátorovi sledovat okolí robotu. Uvnitř robotu je umístěn senzor X-sens MTI-G, který zaznamenává polohu robotu a její změny. Tohoto senzoru je využíváno při tvorbě odometrie (kapitola 4.3). Pro tuto práci je důležitou součástí robotický manipulátor Kinova Jaco Arm 2.2, který byl namontován na tělo mobilního robotu. Robot vydrží na nabitou baterii pracovat více než dvě hodiny a je možná výměna baterie bez vypnutí robotu (takzvaný hot-swap). [6, 5]

Konfigurace ramen mobilního robotu

Jak již bylo napsáno výše, robot BlueBotics Absolom je opatřen čtyřmi rameny (flippery). Využitím těchto ramen je značně zvýšena mobilita robotu. Existuje pět definovaných konfigurací, z nichž je každá využívána pro přejetí jiného typu prostředí. Na Obrázku 3 jsou zobrazeny základní konfigurace ramen mobilního robotu.

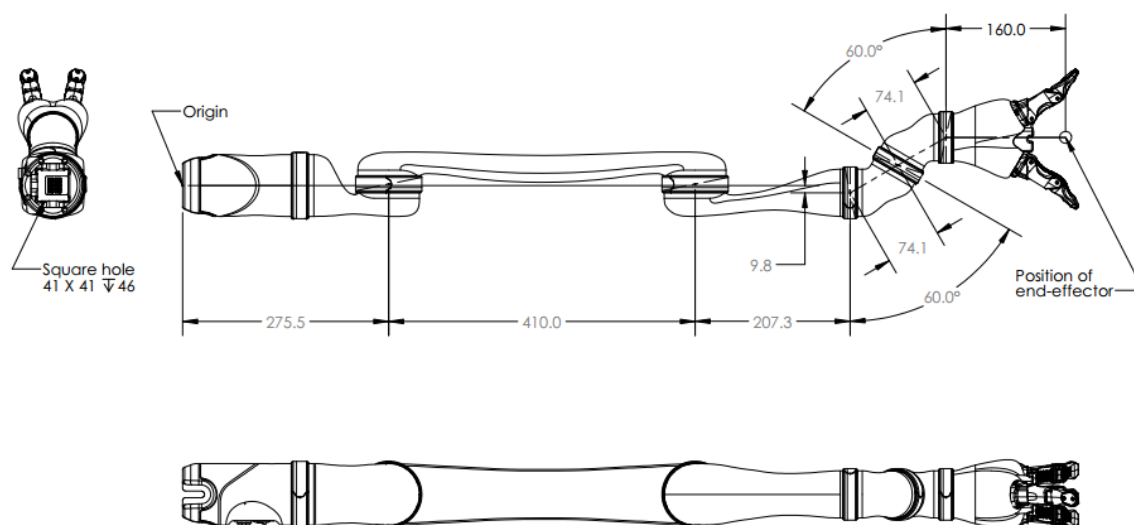


Obrázek 3: Konfigurace mobilního robotu. [2]

2.2 Manipulator Kinova Jaco Arm

Kinova Jaco Arm [3] je robotický manipulátor primárně určený k montáži na elektrické vozíčky pro handicapované lidi. Tato robotická ruka usnadňuje život pohybově postiženým při běžných každodenních činnostech. Těmito činnostmi může být například manipulace s předměty nebo otevírání dveří. Jaco Arm je také využívána v různých robotických projektech, jako je například TRADR. V našem případě, je tato robotická ruka namontována na mobilní robot BlueBotics Absolom, čímž mu přidává další možnosti. Robot může manipulovat s předměty nebo stejně jako u handicapovaných lidí otevírat dveře. V této diplomové práci byla robotická ruka využita pro aktivní průzkum okolí robota. Konkrétně je využitím této ruky zjištěna výška terénu a tuhost v x, y souřadnicích robota.

Jedná se o robotický manipulátor s šesti stupni volnosti. Rameno má šest rotačních kloubů, z nichž čtyři umožňují neomezenou rotaci. Koncový efektor manipulátoru je tvořen třemi prsty. Tyto prsty umožňují úchop a manipulaci s předměty. Dosah této ruky je až 90 cm. Celý manipulátor váží pouhých 5,3 kg což je jeho velkou výhodou. Přes nízkou



Obrázek 4: Schéma Jaco Arm [3]

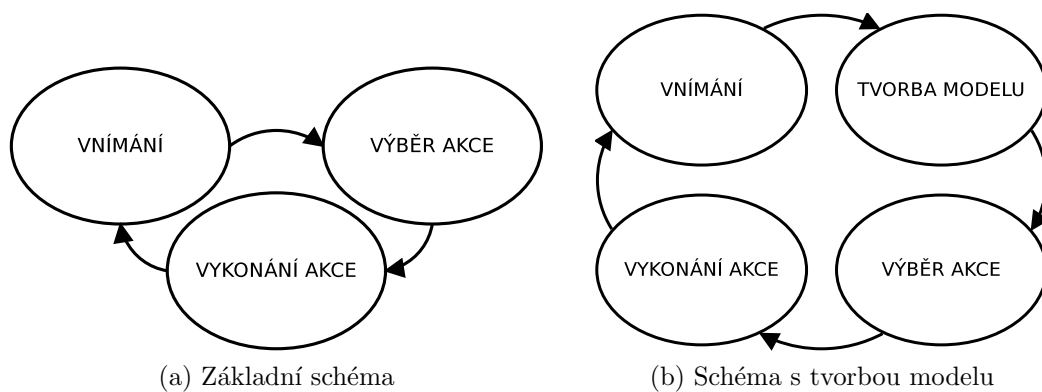
hmotnost dokáže Jaco Arm pohybovat v plné rychlosti předmětem o váze $1,5\text{ kg}$ a s poloviční rychlostí předmětem o váze $2,5\text{ kg}$. Maximální lineární rychlost koncového bodu manipulátoru je 20 cm/s .

2.2.1 Pracovní nástroj

Koncový efektor manipulátoru umožňuje úchop předmětů. Této vlastnosti bylo využito a k prozkoumávání prostředí v okolí robotu se používá pracovní nástroj. Pracovní nástroj je držen v robotické ruce pomocí tří prstů. Tento nástroj je tvořen dřevěnou tyčkou o průměru $0,5\text{ cm}$ a délce 25 cm . Na jednom konci této dřevěné tyčky je připevněn objekt, který je držen koncovým efektozem manipulátoru. Pracovní nástroj má dvě velké výhody. První výhodou je to, že využitím nástroje se zvýší pracovní prostor manipulátoru. Druhou výhodou je to, že při zjišťování výšky terénu pomocí manipulátoru se robotická ruka nedotýká přímo terénu, čímž je chráněna proti poškození. Prsty koncového efektoru nejsou tak pevné jako zbytek manipulátoru a při přímém kontaktu s terénem by mohlo dojít k poškození.

3 Aktivní vnímání

Aktivní vnímání je oblast robotiky, kde pomocí akce (pohyb kamery, pohyb objektu apod.) získáváme nové informace o prostředí nebo o zkoumaném objektu. Tato oblast je zkoumána od devadesátých let minulého století a stále se zdá být dostatečně neprozkoumaným odvětvím. Aktivní vnímání je využíváno například v situacích, kdy má kamera malé zorné pole, při modelování a klasifikaci objektu [7, 8] nebo při mapování [9]. Model aktivního vnímání lze zobrazit pomocí schématu na Obrázku 5a. Je zde vidět, že se skládá ze tří částí. Jednou z těchto částí je senzorická část, která zaznamená pozorování (kamera, dotyk). Druhou částí je výběr akce nebo také rozhodování, tato část je široce zkoumanou oblastí. Třetí částí je vykonání akce nebo také řízení. Většina přístupů aktivního vnímání využívá kromě těchto tří částí navíc i část označenou jako tvorba modelu. Tato část může být nápomocná při výběru akce aktivního vnímání (viz. Obrázek 5b).



Obrázek 5: Schémata aktivního vnímání

Základním cílem úlohy aktivního vnímání je zvolit takovou posloupnost akcí, která pomůže vyřešit danou úlohu. Úlohy aktivního vnímání jsou velmi pestré, což způsobuje to, že je těžké porovnat jednotlivé přístupy aktivního vnímání. Obecně lze rozdělit přístupy aktivního vnímání na tři základní přístupy v závislosti na tom, jakým způsobem je vybírána akce [10]:

- výběr akce na základě užitečnosti,
- výběr akce na základě redukce entropie,
- výběr akce pro nejpravděpodobnější třídu,

První přístup k výběru akce počítá v každém kroku očekávanou užitečnost jednotlivých akcí. Tento přístup určuje užitečnost akce na základě pravděpodobnostních rozdělení pozorování spojených s touto akcí. Vybírá se akce, která má nejvyšší pravděpodobnost toho,

3. AKTIVNÍ VNÍMÁNÍ

že přidá nejvíce informací o modelu. Tato pravděpodobnost je získána z dříve naměřených dat. Přístup výběru akce na základě užitečnosti nepotřebuje žádný trénovací čas, kromě času pro výpočet pravděpodobnostních rozdělání. Druhým přístupem k výběru akce jsou metody, ve kterých je akce vybrána na základě redukce entropie modelu. V tomto přístupu je vytvořena strategie, která podle aktuálního modelu vybere další akci. Vybraná akce je taková, která nejvíce sníží nepřesnost modelu. Může se jednat například o prohledání míst, ve kterých má model největší rozptyl. K učení těchto strategií může být využito například posilovaného učení nebo evolučních algoritmů. Výhoda tohoto přístupu je taková, že umožňuje algoritmům velmi rychlý výběr další akce. Na druhou stranu je potřeba delší trénovací čas. Třetí přístup je využíván hlavně při klasifikaci objektů. Dopředu se spočítají hodnoty jednotlivých akcí. Tyto akce jsou hodnoceny podle snížení nejistoty modelu, za předpokladu, že objekt spadá do nejpravděpodobnější třídy objektů. Tento přístup potřebuje v porovnání s přístupem předchozím nižší trénovací čas. Je ovšem použitelný jen na některé typy úloh.

Mezi aktivní vnímání je možné zahrnout i takzvané haptické vnímání, kde je využito dotyků pro zjištění tvaru, nebo vlastností zkoumaného prvku. S přihlédnutím k zadání této práce, jsou zde popsány některé přístupy zabývající se právě haptickým aktivním vnímáním nebo kombinací haptického a vizuálního aktivního vnímání. Obecně haptické aktivní vnímání, neboli haptická explorace, integruje vstupy ze sensorů (síly, momenty, teplotu a taktilní čidla) ke zjištění informací o objektu, a ke splnění dalších úkolů [9]. Je využívána hlavně v prostředích s částečně nebo úplně sníženou viditelností (například zakouřené nebo zamlžené prostředí). V systémech, které nemají vizuální senzory, nebo k doplnění viděných prvků o informace, jako jsou například teplota, tvrdost, váha, struktura nebo typ materiálu. K haptické exploraci lze přistupovat dvěma způsoby, kterými jsou řízená explorace a neřízená explorace [11]. Neřízená explorace je založena na náhodném prohledávání, zatímco řízená se snaží optimalizovat přínos informace, kterou získáme s každým dotykem. Úloha řízené haptické explorace je stejná jako úloha aktivního vnímání. Cílem je určení místa z pracovního prostoru nebo objektu, které je nejvýhodnější prozkoumat. Při volbě strategie pro výběr pohybu je důležité si uvědomit cenu za pohyb [8]. V některých úlohách je výhodnější prozkoumat spíš místa blízko aktuální polohy ruky, v jiných případech je cena za pohyb nezávislá na aktuální poloze ruky nebo je rozdíl zanedbatelný. V našem případě je cena za pohyb poměrně vysoká a téměř stejná pro všechny body. Ve většině případů je objekt nebo terén během explorace modelován. Model je vytvořen s určitou přesností a pokud je tomuto modelu dostatečně důvěřováno, zkoumaný objekt nebo terén již neprozkoumáváme. V porovnání s vizuálním aktivním vnímáním je problematika haptického aktivního vnímání méně prozkoumaná. Přes to již byla tato úloha v minulosti mnohokrát řešena a bylo využito různých postupů pro určení místa, které po prozkoumání přinese nejvíce informací. Některé z těchto postupů jsou níže popsány. Většina prací, zabývající se

3. AKTIVNÍ VNÍMÁNÍ

tímto tématem, se soustředí hlavně na modelování a klasifikaci objektu. Existuje mnoho prací, které se zabývají haptickou explorací využitím plynulého pohybu po zkoumaném objektu nebo terénu s využitím taktilních senzorů. V těchto případech se jedná o hledání optimální trajektorie poskytující největší množství nových informací.

Jeden z možných přístupů haptického aktivního vnímání ukázal M. Bjorkman a spol. z univerzity KTH [7]. K dispozici mají stereo kameru a manipulátor se třemi prsty, na kterých jsou umístěna taktilní čidla. Jejich cílem je určit, která místa objektu je nejvýhodnější prozkoumat, aby získali co nejpřesnější model objektu, který později mohou klasifikovat a popřípadě uchopit. Stereo kamera poskytuje velké množství informací o tvaru neznámého objektu, ovšem pouze z jedné strany. Dotyky pomocí manipulátoru je využíváno k doplnění informací o neznámých částech objektu. Tento přístup se snaží využít co nejmenší počet akcí. Čím si je robot jistější tvarem objektu, tím méně akcí vykonává. Algoritmus využívá předpokladu, že zkoumaný objekt je uzavřený a je minimálně 1 cm hluboký. Nejprve se pomocí stereo kamery vytvoří část modelu, viditelná kamerou. Dalším krokem je, že pomocí gaussovských procesů získají množinu odhadů tvaru objektu. Jako odhad tvaru objektu využijí střední hodnotu a pomocí rozptylu určují přesnost odhadu. Největší rozptyl mají body nejdál od změřených míst. Pro určení místa, kam má robot sáhnout, je použito místo s maximálním rozptylem. Po změření těchto bodů je objekt znovu vymodelován. Při dotyku objektu s robotem může dojít k nechtěnému posunu objektu. Tento jev řeší spárováním dat ze stereo kamery před dotykem a po dotyku pomocí ICP (Iterative closest point). Tento přístup vykazuje dobré výsledky při modelování neznámých objektů. Nevýhodou ovšem je, že nevyužívá jednoznačné prvky jednotlivých objektů pro jejich klasifikaci.

Přístup, který zvolil Schneider a spol. [12] řeší klasifikaci objektu pomocí haptické explorační. Využívají zde pouze robotický manipulátor, který má za pracovní nástroj dva prsty s taktilními čidly. Objekt je umístěn vždy na stejné místo a robot rozhoduje pouze o výšce, ve které má manipulátor provést měření. Měření z manipulátoru obsahuje dvě matice obsahující data z taktilních senzorů, výšku ve které manipulátor měřil a hodnotu udávající jak byly prsty během měření uzavřeny. Robot se snaží objekt klasifikovat, a to pomocí co nejmenšího počtu dotyků. Na rozdíl od předchozího přístupu se zde objekt nemodeluje. Prsty manipulátoru jsou mnohem menší než je objekt, takže pomocí taktilních čidel je možné detekovat pouze část objektu. Pro klasifikaci využívají metodu zvanou bag-of-features. Tato metoda spočívá v tom, že se vytvoří slovník rysů (feature), pomocí kterých se provádí klasifikace. Volba výšky prstů manipulátoru, ve které se provede další akce, spočívá ve výběru akce, při které je maximální očekávaný zisk informace. Zisk informace je zde reprezentován jako očekávaná změna entropie po provedení dané akce. Tato metoda dokáže rozeznat velké množství různých objektů s pravděpodobností přesahující 95 % již po pár dotycích. Na rozdíl od předchozího přístupu řeší tato práce úlohu s nižším počtem dimenzí.

Přístup využívající pohybu manipulátoru ke změně polohy klasifikovaného objektu popsal A. Doumanoglou a spol. [8]. Cílem je tak jako v předchozích případech klasifikovat objekt. V tomto případě se jedná o typ oblečení (triko, mikina, kalhoty, krat'asy). K dispozici je robot s dvěma rameny, který zvedne kus oblečení na předem nedefinovaném místě. Pomocí aktivního vnímání je pak manipulováno s objektem tak, aby byl pomocí stereo kamery klasifikován do správné třídy oblečení. První akce po zvednutí kusu oblečení je chycení objektu za nejnižší možný bod. Ke klasifikaci je využíváno metody aktivních náhodných lesů (Active Random Forests). Tato metoda je naučena na sadě dat obsahující hloubkový obraz ze stereo kamery, třídu objektu, pozici bodu úchopu a polohu oblečení. Tato data jsou uložena pro různé body úchopů, objekty všech tříd, a pro všechny úhly pohledu s rozlišením 1° . Pokud se nepodaří objekt klasifikovat rovnou, je vybírána akce aktivního vnímání. Akce je vybírána na základě informativnosti aktuálního měření, víry ve viditelnost objektů zájmu v aktuálním měření a ceny za vykonání potenciální akce. Na rozdíl od metody popsané M. Bjorkmanem a spol. je v tomto případě využíváno, naučených hodnot pro volbu další akce. Tento způsob využívá přímo klasifikátoru k tomu, aby byla zvolena příští akce manipulátoru.

4 Teorie

Tato kapitola popisuje teoretický základ nutný k implementaci. V první části této kapitoly jsou popsány základy kinematiky robotů. Dále je zde popsána struktura a principy využívaných knihoven ROS a "MoveIt!". V dalších částech jsou popsány struktury dat, se kterými se pracuje. V poslední části je popsán algoritmus Gibbsova vzorkování a navržený algoritmus pro aktivní vnímání.

4.1 Základní pojmy kinematiky robotů

Při psaní této kapitoly bylo čerpáno hlavně z knihy [13]. Cílem této kapitoly je definice inverzní kinematické úlohy. Pro vysvětlení pojmu inverzní kinematické úlohy, je potřeba nejprve zavést několik jiných pojmů. Kinematika řeší obecně pohyb mechanismů s tím, že bere v potaz pouze geometrické uspořádání, rychlosti a zrychlení jednotlivých částí manipulátoru. Není tedy brán ohled na síly a momenty sil působící v mechanismu. Jedná se o základní aspekt robotiky pro design robotů, analýzu, řízení a simulaci pohybu. Robotický mechanismus si můžeme představit jako soustavu pevných těles (členů), propojených pomocí kloubů. Klouby mohou být různých typů (například posuvný nebo rotační). Typ mechanismu může být sériový, paralelní nebo hybridní. Toto rozdělení je určeno posloupností jednotlivých kloubů a těles.

Transformace

V kinematice se využívá různých reprezentací polohy mechanismu pro řešení jednotlivých kinematických úloh. Pro určení polohy jednotlivých těles je využíváno kombinace rotace a translace. Translace je reprezentována pomocí vektoru \mathbf{p} , označujícího směr a velikost posunu.

$$\mathbf{p} = \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix}. \quad (1)$$

Rotace může být reprezentována několika způsoby. Mezi tyto způsoby spadá například reprezentace pomocí rotační matice, Eulerových úhlů nebo kvaternionů. Obecně vypadá rotační matice takto:

$$\mathbf{R} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}. \quad (2)$$

Rotační matice obsahuje devět elementů, pro definici orientace ale stačí pouze tři. Těmito třemi parametry mohou být například Eulerovy úhly. Přepočtení mezi rotační maticí a

Eulerovými úhly vypadá následovně.

$$\beta = \text{Atan2}(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2}), \quad (3)$$

$$\alpha = \text{Atan2}\left(\frac{r_{21}}{\cos(\beta)}, \frac{r_{11}}{\cos(\beta)}\right), \quad (4)$$

$$\gamma = \text{Atan2}\left(\frac{r_{32}}{\cos(\beta)}, \frac{r_{33}}{\cos(\beta)}\right). \quad (5)$$

Úhly α, β, γ se také označují *yaw, pitch, roll*. Toto označení pro Eulerovy úhly je využíváno i v této diplomové práci. Další možností reprezentace orientace je reprezentace pomocí kvaternionu. Kvaternion je čtyřrozměrný vektor s prvky $\epsilon_0, \epsilon_1, \epsilon_2, \epsilon_3$, které se z rotační matice dají vypočítat následujícími rovnicemi:

$$\epsilon_0 = \frac{1}{2}\sqrt{1 + r_{11} + r_{22} + r_{33}} \quad (6)$$

$$\epsilon_1 = \frac{r_{32} - r_{23}}{4\epsilon_0}, \quad (7)$$

$$\epsilon_2 = \frac{r_{13} - r_{31}}{4\epsilon_0}, \quad (8)$$

$$\epsilon_3 = \frac{r_{21} - r_{12}}{4\epsilon_0}. \quad (9)$$

V této práci je využíváno interpretace rotace pomocí Eulerových úhlů (v prostředí matlab, kapitola 5.4) a pomocí kvaternionů (v "MoveIt!" a souvisejících programech, kapitola 5.3). Poloha každého tělesa je definována využitím transformační matice ${}^j\mathbf{T}_i$.

$${}^j\mathbf{T}_i = \begin{pmatrix} {}^j\mathbf{R}_i & {}^j\mathbf{p}_i \\ \mathbf{0}^T & 1 \end{pmatrix}, \quad (10)$$

kde ${}^j\mathbf{T}_i$ je transformační matice, transformující vektory ze souřadnicového systému i do souřadnicového systému j . Dále ${}^j\mathbf{R}_i$ je rotační matice a ${}^j\mathbf{p}_i$ je translační vektor, značící rotaci a posun ze souřadnicového systému i do souřadnicového systému j . Využitím matic ${}^j\mathbf{T}_i$ dokážeme popsat polohu jednotlivých členů celého mechanismu.

Reprezentace manipulátoru

Mějme manipulátor (mechanismus) obsahující N členů. Pro řešení kinematických úloh, je důležité vytvořit referenční souřadný systém $\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i$ pro každý člen i , kde $i \in N$, robotického manipulátoru. Kloub označený jako i je umístěn mezi členy $i - 1$ a i . Jedním ze způsobů, jak určit souřadnicové systémy jednotlivých částí a jak zapsat transformace mezi nimi, je například Denavit-Hartenbergova notace. Využitím Denavit-Hartenbergovy

notace přidělíme jednotlivým členům souřadné systémy. Tato notace udává, že osa \mathbf{z}_i je umístěná vždy v ose kloubu i , kde osa kloubu značí osu, kolem které se kloub otáčí (rotační kloub), nebo podél které se kloub pohybuje (posuvný kloub). Dále tato notace udává, že osa \mathbf{x}_{i-1} leží ve směru společné normály os \mathbf{z}_{i-1} a \mathbf{z}_i . Transformační matice ${}^{i-1}\mathbf{T}_i$ mezi souřadnicovými systémy náležící členům $i-1$ a i vypadá obecně takto

$${}^{i-1}\mathbf{T}_i = \begin{pmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (11)$$

kde a_i je vzdálenost z \mathbf{z}_{i-1} do \mathbf{z}_i podél osy \mathbf{x}_{i-1} , α_i je úhel mezi \mathbf{z}_{i-1} a \mathbf{z}_i podél osy \mathbf{x}_{i-1} , d_i je vzdálenost z \mathbf{x}_{i-1} do \mathbf{x}_i podél osy \mathbf{z}_i a θ_i je úhel mezi \mathbf{x}_{i-1} a \mathbf{x}_i podél osy \mathbf{z}_i . Tyto čtyři parametry se nazývají Denavit-Hartenbergovy parametry. Zatímco α_i a a_i závisí pouze na geometrickém a mechanickém složení manipulátoru, tak θ_i a d_i závisí na aktuální poloze kloubů.

Přímá kinematická úloha

Nyní je již možné přejít k pojmu přímé a inverzní kinematické úlohy. Vzhledem k tomu, že manipulátor, se kterým je pracováno v této diplomové práci je sériový, tak je tato část soustředěna pouze na případ sériového manipulátoru. Řešení přímé kinematické úlohy pro sériový manipulátor spočívá v nalezení pozice a orientace koncového efektoru vzhledem k rámu. V praxi známe geometrickou strukturu manipulátoru i polohy jednotlivých kloubů (kloubové souřadnice \mathbf{q}) získaných pomocí senzorů. Přímá kinematická úloha je řešena počítáním jednotlivých transformací mezi rámem a koncovým efektozem manipulátoru. Výpočet přímé kinematické úlohy pro robot s šesti stupni volnosti (šest kloubů s kloubovými souřadnicemi $\mathbf{q} = (q_1, q_2, q_3, q_4, q_5, q_6)$) vypadá následovně:

$${}^0\mathbf{T}_6(\mathbf{q}) = {}^0\mathbf{T}_1(q_1) {}^1\mathbf{T}_2(q_2) {}^2\mathbf{T}_3(q_3) {}^3\mathbf{T}_4(q_4) {}^4\mathbf{T}_5(q_5) {}^5\mathbf{T}_6(q_6). \quad (12)$$

Hodnoty jednotlivých prvků transformačních matic ${}^{i-1}\mathbf{T}_i$ jsou známé a matice ${}^0\mathbf{T}_6$ udává relativní polohu koncového efektoru vzhledem k rámu. Vypočítáním této rovnice získáme právě jedno řešení.

Inverzní kinematická úloha

Opakem přímé kinematické úlohy je úloha inverzní. Úkolem inverzní kinematické úlohy pro sériový manipulátor je nalezení kloubových souřadnic \mathbf{q} manipulátoru, pokud známe

polohu a orientaci koncového efektoru, vzhledem k rámu, a geometrickou strukturu jednotlivých částí. V nejčastějším případě se řeší úloha pro sériový manipulátor se šesti stupni volnosti. V tomto případě známe transformační matici ${}^0\mathbf{T}_6$ (viz. rovnice (12)) a úkolem je zjistit kloubové souřadnice $\mathbf{q} = (q_1, q_2, q_3, q_4, q_5, q_6)$. Jedná se o řešení sady nelineárních rovnic. Jejich výpočtem můžeme zjistit, že řešení neexistuje nebo můžeme získat rovnou několik různých řešení. Inverzní kinematická úloha bývá řešena buď analyticky nebo numericky. Analytický výpočet je sice rychlejší, ale není obecný. Z toho důvodu je často využíváno metod numerických, které nejsou závislé na geometrické struktuře robotu. V této diplomové práci bylo při implementaci inverzní kinematické úlohy využito knihoven, využívajících právě numerické řešení.

4.2 Statické síly v manipulátoru

Analýza statických sil v manipulátoru popisuje vztahy mezi silami a momenty sil v koncovém efektoru a v jednotlivých kloubech. Analýza sil je využívána například pro interakci robotu s prostředím. Často nejsou síly v jednotlivých motorech měřeny pomocí senzorů, ale jsou vypočítány z proudových odběrů v jednotlivých motorech.

Pro přepočítání sil je nejdříve nutné nadefinovat Jacobiho matici. Jacobiho matice je obecně matice velikosti $6 \times N$, kde N je počet stupňů volnosti manipulátoru. Jedná se o matici obsahující všechny první parciální derivace vektorové funkce. Uvažujme případ, kdy se jedná o manipulátor se šesti stupni volnosti. Naše vektorová funkce vypadá takto:

$$\mathbf{f}(\mathbf{q}) = (f_x(\mathbf{q}), f_y(\mathbf{q}), f_z(\mathbf{q}), f_\alpha(\mathbf{q}), f_\beta(\mathbf{q}), f_\gamma(\mathbf{q})), \quad (13)$$

kde funkce f_x , f_y a f_z vracejí pozici koncového efektoru a funkce f_α , f_β a f_γ jeho natočení. Pokud je Jacobiho matice $\mathbf{J}(\mathbf{q})$ čtvercová, vypadá přepočítání sil z jednotlivých kloubů do koncového efektoru následovně:

$$\mathbf{J}(\mathbf{q}) = \begin{pmatrix} \frac{\partial f_x}{\partial q_1} & \frac{\partial f_x}{\partial q_2} & \frac{\partial f_x}{\partial q_3} & \frac{\partial f_x}{\partial q_4} & \frac{\partial f_x}{\partial q_5} & \frac{\partial f_x}{\partial q_6} \\ \frac{\partial f_y}{\partial q_1} & \frac{\partial f_y}{\partial q_2} & \frac{\partial f_y}{\partial q_3} & \frac{\partial f_y}{\partial q_4} & \frac{\partial f_y}{\partial q_5} & \frac{\partial f_y}{\partial q_6} \\ \frac{\partial f_z}{\partial q_1} & \frac{\partial f_z}{\partial q_2} & \frac{\partial f_z}{\partial q_3} & \frac{\partial f_z}{\partial q_4} & \frac{\partial f_z}{\partial q_5} & \frac{\partial f_z}{\partial q_6} \\ \frac{\partial f_\alpha}{\partial q_1} & \frac{\partial f_\alpha}{\partial q_2} & \frac{\partial f_\alpha}{\partial q_3} & \frac{\partial f_\alpha}{\partial q_4} & \frac{\partial f_\alpha}{\partial q_5} & \frac{\partial f_\alpha}{\partial q_6} \\ \frac{\partial f_\beta}{\partial q_1} & \frac{\partial f_\beta}{\partial q_2} & \frac{\partial f_\beta}{\partial q_3} & \frac{\partial f_\beta}{\partial q_4} & \frac{\partial f_\beta}{\partial q_5} & \frac{\partial f_\beta}{\partial q_6} \\ \frac{\partial f_\gamma}{\partial q_1} & \frac{\partial f_\gamma}{\partial q_2} & \frac{\partial f_\gamma}{\partial q_3} & \frac{\partial f_\gamma}{\partial q_4} & \frac{\partial f_\gamma}{\partial q_5} & \frac{\partial f_\gamma}{\partial q_6} \end{pmatrix}. \quad (14)$$

Jacobiho matice slouží k přepočtu rychlostí a sil z kloubových souřadnic do souřadnic koncového efektoru a naopak.

V našem případě je počítána síla v koncovém efektoru manipulátoru. K tomu je využíváno výše zmíněné Jacobiho matice $\mathbf{J}(\mathbf{q})$ a momentů sil $\boldsymbol{\tau}$ v kloubech, které jsou počítány přímo z proudových odběrů. Pokud je matice $\mathbf{J}(\mathbf{q})$ regulární, platí následující rovnice.

$$\mathbf{f} = (\mathbf{J}(\mathbf{q})^T)^{-1} \cdot \boldsymbol{\tau}, \quad (15)$$

kde \mathbf{f} je vektor obsahující sílu a moment síly v koncovém bodě manipulátoru.

4.3 Odometrie

Odometrie (inkrementální odometrie) [14] je nejčastějším způsobem měření rychlosti robotu. Při této metodě transformujeme data z enkodérů umístěných v motorech na rychlost robotu a jeho směr. Senzor je nejčastěji umístěn na hřídeli motoru a měří pootočení kola. Tyto senzory posílají informace o rychlosti točení kola, ze kterých počítáme rychlost celého robotu. Jako senzor v motorech se často využívá Hallova sonda. Tato metoda je citlivá na smýkání kol, funguje tedy jen na některých terénech.

V naší úloze je poloha počítána dvěma způsoby. Prvním z nich je základní IMU odometrie, kde se jedná o klasickou výše zmíněnou odometrii. Poloha je v tomto případě počítána následujícím způsobem. Ke směrnici natočení robotu, získané pomocí gyroskopu a matematického modelu, se přičítá změna polohy spočítána z rychlosti pásů. Tento způsob výpočtu kumuluje chybu určení polohy. Druhým způsobem výpočtu polohy v projektu je takzvaná ICP odometrie. Tento způsob vychází z předchozího výpočtu, navíc ale využívá dat z laseru ke korekci chyby při určování polohy robotu. Jak již název napovídá, korekce polohy je prováděna metodou ICP (Iterative Closest Point), kde se spojují měření z laseru a mapy uložené ve shluku 3D bodů, čímž se zjistí chyba v odometrii, která se napraví. Oproti původní IMU odometrii je zde výpočet polohy přesnější a nekumuluje se chyba. V úloze mapování (kapitola 5.4.6) je využívána hlavně IMU odometrie, a to z toho důvodu, že při špatné viditelnosti laseru by metoda korekce polohy stejně nefungovala.

4.4 ROS

ROS (Robot Operating System) [15] je soubor nástrojů, knihoven a konvencí usnadňující vývojářům robotických aplikací implementaci komplexních programů. Využitím ROS se snadno vytváří multi-vláknové aplikace, které jsou podstatnou součástí robotiky. Základní strukturu ROS tvoří takzvané uzly (nodes), které označují jednotlivé procesy. Využitím knihoven ROS je jednoduchá komunikace mezi uzly, které mohou být psány v různých programovacích jazycích. Uzly mezi sebou komunikují pomocí zpráv skrz témata (topics) využitím TCP/IP protokolu. Zprávy v ROSu mají pevně definovaný typ a jsou

mezi uzly distribuovány na principu vydavatel/odběratel (publisher/subscriber). Vydavatel slouží k posílání zpráv z uzlu do tématu. Odběratel poslouchá téma a přijímá zprávy, které se v daném tématu vyskytují. Dále existují uzly typu servis (service), které čekají na žádost (request) na daném tématu a vrací odpověď (reply).

4.5 MoveIt!

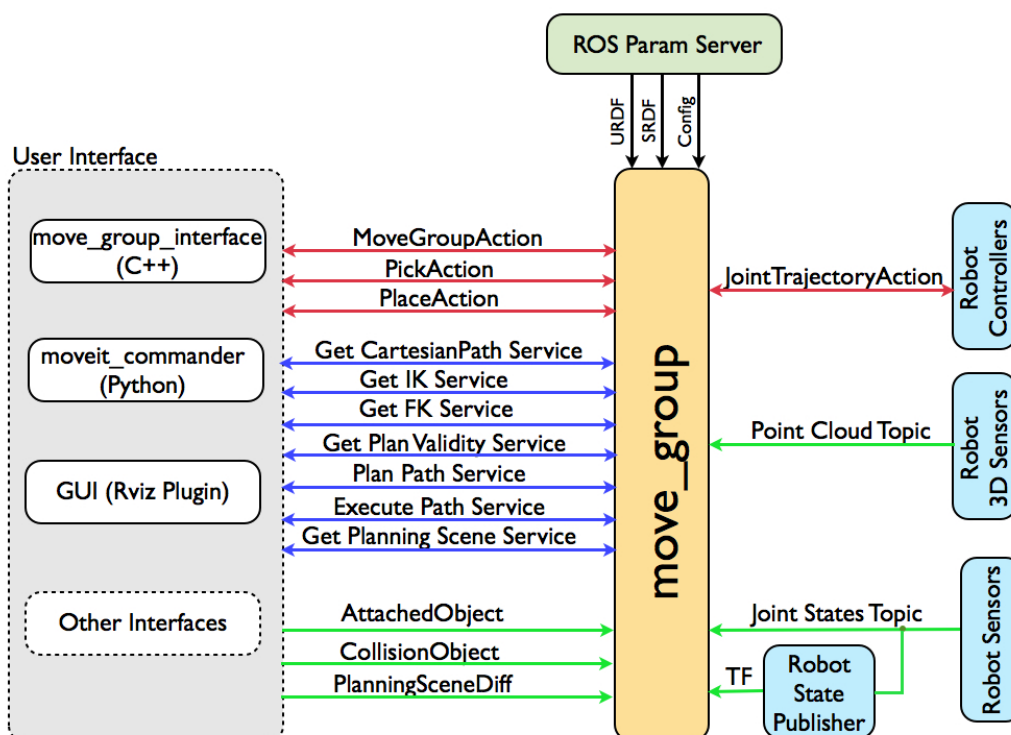
”MoveIt!”[4] je software vytvořený v prostředí ROS, umožňující vytvářet pokročilé robotické aplikace pro robotické manipulátory. ”MoveIt!” poskytuje podporu kinematiky, plánování trajektorie a pohybu, sledování kolizí a interakci robotu s okolím. Momentálně ”MoveIt!” podporuje přes 65 různých manipulátorů a robotů, a díky rozšířené komunitě uživatelů, se počet podporovaných robotů stále rozšiřuje.

V této části diplomové práce je popsána struktura balíčku ”MoveIt!”, který je později využit při implementaci inverzní kinematické úlohy manipulátoru Jaco Arm. Architektura celého ”MoveIt!” je zobrazená na Obrázku 6. Koncept ”MoveIt!” se dělí na čtyři základní části

- *move_group*,
- *ROS parametry (ROS Param Server)*,
- *rozhraní robotu (Robot Interface)*,
- *uživatelské rozhraní (User Interface)*.

Na tomto schématu (Obrázek 6) je vidět, že jednou z hlavních komponent ”MoveIt!” je *move_group*, se kterým komunikují další komponenty pomocí zpráv skrz vyznačená témata. Tento uzel slouží jako integrátor spojující všechny jednotlivé komponenty ”MoveIt!”.

Část konceptu *ROS parametry (ROS Param Server)*, poskytuje *move_group* tři typy informací. První z nich je *URDF (Unified Robotic Description Format)* popis, zachycující kinematickou strukturu celého robotu. V *URDF* popisu jsou zaznamenány virtuální kolize jednotlivých kloubů a ramen. Je zde zaznamenán tvar jednotlivých ramen a typ kloubů manipulátoru (rotační, posuvný atd.). Další informací, kterou komponenta *ROS parametry* poskytuje je *SRDF (Semantic Robot Description Format)* popis. Tento popis definuje jednotlivé skupiny manipulátoru. Existuje možnost rozdělit manipulátor do několika skupin, pro které se plánuje pohyb zvlášť. Důvod pro rozdělení manipulátoru do několika skupin je takový, že snížíme počet stupňů volnosti manipulátoru, pro který plánujeme pohyb. Příkladem může být robot, který má dvě robotická ramena, ale chceme pohybovat jen jedním. Pomocí popisu *SRDF* se také často odděluje manipulátor a například prsty koncového nástroje, jako je tomu v našem případě. Poslední informací, kterou poskytuje komponenta



Obrázek 6: Architektura MoveIt! [4]

ROS parametry je takzvaná *konfigurace (config)*. Zde jsou definovány dodatečné informace o manipulátoru, jako jsou například omezení jednotlivých kloubů, maximální rychlosti motorů nebo významné využívané polohy (např. domácí poloha manipulátoru).

Komunikace *move_group* s robotem je poskytována skrz *rozhraní robotu (robot interface)* pomocí zpráv a akcí. Do *rozhraní robotu* spadají komponenty, které jsou na schématu (Obrázek 6) vpravo od *move_group*. Je nutné podotknout, že *rozhraní robotu* není implementováno v rámci knihoven "MoveIt!", ale úzce s ním souvisí. "MoveIt!" očekává na zadaných tématech ROS vydavatele a odběratele. Komponenty z *rozhraní robotu* jsou využívány k zjištění informace o aktuální pozici jednotlivých kloubů, k získání 3D bodů reprezentující překážky, nebo k ovládání motorů manipulátoru. Komponenta označená ve schématu jako *Robot Sensors* poskytuje informaci o aktuální pozici jednotlivých kloubů. Informace o poloze manipulátoru je využita i pro získání transformace, kde se kromě transformace mezi jednotlivými rameny a klouby, doplní i informace o transformaci manipulátoru vůči zbytku robotu nebo vůči globální mapě. Další část *rozhraní robotu* má název *3D senzory robotu (Robot 3D Sensors)*. Tato část slouží k získání a předání informace o prostředí, ve kterém se robot nachází. Na tématu označeném jako *Point Cloud Topic* je očekáván shluk 3D bodů (pointcloud) reprezentující překážky. Takovéto shluky jsou získány ze senzorů robotu

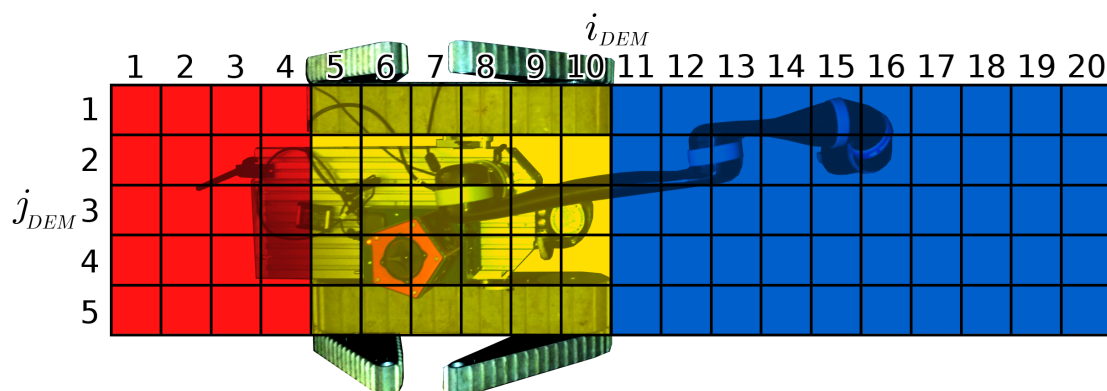
jako je například laser nebo stereo kamera. V rámci *move_group* jsou pak shluky 3D bodů převedeny na takzvanou OktoMapu (OctoMap), udávající obsazenost prostoru. OktoMapa je 3D mapa obsazenosti prostoru, vytvářená na principu Oktálového stromu [16]. Převod shluku bodů na OktoMapu je prováděn v komponentě *Planning Scene Monitor*, umístěné uvnitř *move_group*. Tato komponenta *move_group* přijímá informace o poloze manipulátoru a shluk 3D bodů ze sensorů robotu, které pomocí takzvaného *octomap_server* převádí na OktoMapu. Třetí částí rozhraní robotu jsou ovladače robotu (*Robot Controllers*). Tyto ovladače jsou využívány k řízení motorů a samotného provádění pohybu manipulátoru.

Část konceptu "MoveIt!", řešící přístup k *move_group*, je *uživatelské rozhraní (user interface)*. Přístup k *move_group* je možný třemi základními způsoby. Jedním z nich je pomocí grafického rozhraní *Rviz*. V *Rviz* je možné vizualizovat většinu ROS témat. Grafické rozhraní je užitečné například pro vizualizaci reprezentace prostředí, nebo pro vizualizaci plánu před tím, než je proveden. Přístup k *move_group* je také možný přímo z programu nebo funkce a to dvěma způsoby. Jedním z nich je *move_group_interface*, využívaný v programovacím jazyce C++, druhým je *move_group_comander*, který je využíván programovacím jazyce Python. V části *uživatelského rozhraní* je nadefinován cíl pohybu a jeho omezení. Mimo jiné se zde také specifikuje, který plánovač má být využit pro vytvoření trajektorie. "MoveIt!" totiž využívá knihovnu plánovačů OMPL [17], která nabízí velké množství algoritmů pro naplánování trajektorie manipulátoru.

4.6 DEM

Důležitým pojmem, který je v implementovaných programech využíván, je takzvaný DEM (z anglického Digital elevation map). DEM je mřížkový model prostředí popisující nejbližší okolí robotu. Jedná se o mapu 20×5 polí mřížky, s rozlišením 10×10 cm na pole. V každém z těchto polí je uložena hodnota, určující výšku terénu v tomto konkrétním poli. Souřadnicový systém DEM je v této práci označován pomocí i_{DEM} a j_{DEM} (viz Obrázek 7). V tomto obrázku je zobrazeno rozdělení DEM na tři části podle umístění robotu v mapě. První část $i_{DEM} = \langle 1, 4 \rangle$ DEM (červená) určuje terén za robotem, druhá část $i_{DEM} = \langle 5, 10 \rangle$ (žlutá) je část DEM pod robotem a poslední část $i_{DEM} = \langle 11, 20 \rangle$ (modrá) mapuje prostředí před robotem. Rozdělení DEM na tři části je důležité kvůli jeho tvorbě.

DEM je tvořen z dat získaných pomocí laseru. Tato data jsou uložena do shluků 3D bodů. Existuje však několik různých shluků bodů, vytvářených robotem v prostředí ROS. Jedním z nich je dynamický shluk 3D bodů (*dynamic_pointcloud*), který je získáván přímo z dat s laseru. Dynamický shluk 3D bodů představuje aktuální prostředí v zorném poli laseru. Dalším pro nás důležitým shlukem 3D bodů je statický shluk (*static_pointcloud*). Tento typ shluku bodů představuje mapu prostředí, kterou robot sestavil. Statický shluk



Obrázek 7: Umístění robotu v DEM

3D bodů je vytvořen z dynamických shluků, které jsou na sebe skládány pomocí metody ICP (Iterative Closest Point).

Část před robotem (modrá) je tvořena přímo z dynamického shluků 3D bodů. Algoritmus pro tvorbu DEM prohlédne shluk 3D bodů a najde body, které spadají do intervalu ohraničeného souřadnicemi x a y jednotlivých polí v DEM. Z těchto bodů se spočítá průměrná hodnota, která se vzápětí uloží do daného pole DEM. Část pod robotem (červená) a za robotem (žlutá) je počítána z dynamického shluku 3D bodů, a to stejným způsobem, který je popsán výše. Důvodem k tomu je, že senzory robotu neumožňují získání aktuálních dat z těchto míst. Zatímco u části za robotem jsou data získána z dynamického shluku rovnou, u části pod robotem jsou data nejdříve zpracována. Není možné, aby terén pod robotem zasahoval do těla robotu. Z toho důvodu jsou body, které jsou ve výšce vyšší než je tělo mobilního robotu, sníženy tak, aby do robotu nezasahovaly. Důležitým faktem je, že hodnoty výšky terénu v polích DEM jsou orientovány v z -souřadnici globální mapy. Hodnoty jednotlivých polí DEM, jsou výšky terénu, vztahované ke stabilizovanému souřadnému systému robotu. Střed souřadného systému robotu je umístěn 9 cm pod těžištěm (při nulovém naklonění mobilního robotu bez namontované robotické ruky) a osy x a y jsou orientovány tak, jako podvozek robotu. Střed stabilizovaného souřadného systému robotu, je umístěn ve stejném místě jako střed souřadného systému robotu, osy x a y jsou ale umístěny vodorovně. Při tvorbě DEM se stává, že na některých polích DEM nejsou ve shlucích 3D bodů žádná naměřená data. V takovém případě je hodnota pole DEM označena 0 a dodatečná informace je uložena ve vektoru příznaků (viz. kapitola 4.7).

4.7 Vektor příznaků

DEM popisovaný v předchozí části má tu výhodu, že je přímo z dvourozměrného pole vidět, která pole sousedí s kterými. Této výhody je využíváno například při grafické vi-

zualizaci prostředí v okolí robotu. DEM není na žádném ROS tématu vysílán ve formátu popisovaném výše. Existuje však vektor příznaků, ve kterém je mimo jiné celý DEM uložen. Tento vektor je o velikosti 210 polí a je vysílán na ROS tématu *model_request*.

číslo pole	1	2	...	5	6	...	10	...	100
obsah	DEM _{1 1}	DEM _{1 2}	...	DEM _{1 5}	DEM _{2 1}	...	DEM _{2 5}	...	DEM _{20 5}

Tabulka 1: Uspořádání DEM ve vektoru příznaků.

Prvních 100 prvků obsahuje DEM a to ve formátu popsáném v Tabulce 1. Pokud použijeme analogii z Obrázku 7, tak je DEM ukládán po sloupcích i_{DEM} od konce za robotem směrem před robot. Další 10 polí vektoru příznaků obsahuje sensorická data. Pořadí, ve kterém jsou sensorická data uložena do vektoru příznaků, je zobrazeno v Tabulce 2. V této

číslo pole	101	102	103	104	105	106	107	108	109	110
obsah	<i>pitch</i>	v_r	v_l	rear _r	main _r	front _r	rear _l	main _l	front _l	<i>roll</i>

Tabulka 2: Uspořádání sensorických dat ve vektoru příznaků.

tabulce jsou uvedeny zkratky jednotlivých měření, která budou v následujícím textu uvedena v závorce. Mezi sensorická data je uloženo naklonění robotu (*roll*, *pitch*), dopředné rychlosti pravého (v_r) a levého (v_l) hlavního pásu a proudy v jednotlivých motorech. Proudové odběry motorů jsou uloženy na 104. až 109. poli vektoru příznaků. Jsou zde uloženy proudy v hlavních pásech (main_r, main_l), v předních ramenech (front_r, front_l) a v zadních ramenech (rear_r, rear_l). Další 100 prvků (111. až 210.) obsahuje informaci o počtu bodů ve 3D shluku, ze kterých byla spočítána výška konkrétního pole DEM. Pokud je na tomto místě hodnota 0, je výška tohoto pole v DEM neznámá. Seřazení informace o počtu bodů spadajících do intervalů jednotlivých polí DEM je stejné jako v seřazení DEM a je zobrazeno v Tabulce 3.

číslo pole	111	112	...	115	116	...	120	...	210
obsah	bod _{1 1}	bod _{1 2}	...	bod _{1 5}	bod _{2 1}	...	bod _{2 5}	...	bod _{20 5}

Tabulka 3: Uspořádání informace o počtu bodů při tvorbě DEM ve vektoru příznaků.

4.8 Gibbsovo vzorkování

Ve většině případů řešených v reálném světě je získání přesného pravděpodobnostního modelu nemožné. Z toho důvodu existují metody, snažící se o vytvoření co nejpřesnější

aproximace reálného pravděpodobnostního modelu. Jednou ze tříd metod, které tento problém řeší, jsou metody Monte Carlo, založené na vzorkování. Vzorkování je v teorii pravděpodobnosti metoda, při které dochází k výběru prvku na základě pravděpodobnostního rozdělení. Jedná se o náhodný výběr prvku z populace, kde každý prvek populace má svou pravděpodobnost výběru, určenou právě pravděpodobnostní distribucí. Monte Carlo metody lze definovat jako stochastické výpočetní techniky, využívající pseudonáhodná čísla k řešení problémů, pro které je použití klasických výpočetních metod složité nebo nemožné. Základní myšlenkou Monte Carlo metod je určit střední hodnotu z dané náhodné veličiny (náhodného vektoru) s určitým pravděpodobnostním rozdělením. A to v případech, kdy sdružená pravděpodobnost náhodného vektoru není známa nebo se z ní složitě vzorkuje.

Rozšířením Monte Carlo metod jsou metody nazvané MCMC (zkratka z ang. *Markov Chain Monte Carlo*), mezi které spadá i algoritmus Gibbsova vzorkování. MCMC metody využívají Markovovy řetězce a vzorkování pro získání výsledného pravděpodobnostního rozdělení. Markovovy řetězce jsou takové řetězce, kde příští stav závisí pouze na stavu aktuálním. V tomto případě jsou stavy Markovova řetězce tvořeny jednotlivými vzorky. Přechodová funkce mezi stavy pak závisí na podmíněných pravděpodobnostech dané proměnné podmíněné ostatními proměnnými. Při $T \rightarrow \infty$, kde T je počet přechodů mezi stavy, konverguje stav do ekvilibria. Pokud jsou dobře modelované podmíněné pravděpodobnosti, toto ekvilibrium aproximuje reálný pravděpodobnostní model.

Pro Gibbsovo vzorkování je potřebné znát podmíněné pravděpodobnosti $p(x_i|\mathbf{x}_{-i})$, kde x_i je hodnota i -tého prvku vektoru $\mathbf{x} = (x_1, x_2, \dots, x_N)$ a \mathbf{x}_{-i} označuje vektor \mathbf{x} bez i -tého prvku. Mějme pravděpodobnostní rozdělení $p(\mathbf{x}) = p(x_1, x_2, \dots, x_N)$, ze kterého chceme vzorkovat. A předpokládejme, že všechny proměnné x_i , kde $i \in \{1, 2, \dots, N\}$, jsou v nějakém počátečním stavu Markovova řetězce. V každém kroku procedury Gibbsova vzorkování nahradíme jednu z proměnných za hodnotu vybranou z pravděpodobnostního rozdělení podmíněného ostatními proměnnými. Nahradíme tedy hodnotu prvku x_i za jinou hodnotu získanou výběrem vzorku z pravděpodobnostního rozdělení $p(x_i|\mathbf{x}_{-i})$. Tento postup je opakován ve smyčce pro všechny proměnné s tím, že vždy je vybrán náhodný vzorek z pravděpodobnostního rozdělení dané podmíněné pravděpodobnosti.

Gibbsovo vzorkování je metoda vzorkování z minimálně dvourozměrné pravděpodobnostní distribuce. Příklad Gibbsova vzorkování je uveden na nejjednodušším možném případě s náhodným vektorem $\mathbf{x} = (x_1, x_2)$. Naším cílem je získat střední hodnoty z dvourozměrného pravděpodobnostního rozdělení $p(x_1, x_2)$. Předpokládejme, že x_1 a x_2 mají hodnotu $x_1^{(t)}$ a $x_2^{(t)}$, která byla získána v t -té iteraci algoritmu. V případě počáteční inicializace jsou hodnoty nastaveny libovolně. V iteraci $t + 1$ se při prvním kroku vybere vzorek $x_1^{(t+1)}$ z pravděpodobnostního rozdělení

$$p(x_1|x_2^{(t)}), \tag{16}$$

kde $x_2^{(t)}$ je hodnota z předchozí iterace. V dalším kroku se vybere vzorek $x_2^{(t+1)}$ z pravděpodobnostního rozdělení

$$p(x_2|x_1^{(t+1)}). \quad (17)$$

Tímto způsobem, po provedení dostatečného množství iterací, získáme odhad střední hodnoty sdružené pravděpodobnosti $p(x_1, x_2)$. Pseudokód popisující algoritmus Gibbsova vzorkování pro N -rozměrného pravděpodobnostního rozdělení vypadá následovně:

Algoritmus 1: Gibbsovo vzorkování

1. Inicializace $\{x_i : i = 1, \dots, N\}$
 2. For $t = 1, \dots, T$:
 - vyber vzorek $x_1^{(t+1)} \sim p(x_1|x_2^{(t)}, x_3^{(t)}, \dots, x_N^{(t)})$
 - vyber vzorek $x_2^{(t+1)} \sim p(x_2|x_1^{(t+1)}, x_3^{(t)}, \dots, x_N^{(t)})$
 - ⋮
 - vyber vzorek $x_j^{(t+1)} \sim p(x_j|x_1^{(t+1)}, \dots, x_{j-1}^{(t+1)}, x_{j+1}^{(t)}, \dots, x_N^{(t)})$
 - ⋮
 - vyber vzorek $x_N^{(t+1)} \sim p(x_N|x_1^{(t+1)}, x_2^{(t+1)}, \dots, x_{N-1}^{(t+1)})$
-

Existují případy, ve kterých jsou některé proměnné, které odpovídají měřením, známy. Například v našem konkrétním případě známe vždy proměnné odpovídající naklonění robotu a proudům v jednotlivých pásech. Pokud jsou některé proměnné známy, není pro tyto proměnné prováděn výběr vzorku z pravděpodobnostního rozdělení. Neznámé hodnoty musí být na začátku inicializovány do nějakého stavu. Počáteční stav se často volí náhodný, tudíž nevyovídá nic o výsledném pravděpodobnostním rozdělení. Z toho důvodu se při implementaci algoritmu často zahazuje několik prvních vzorků. Z ostatních vzorků je pak brán buď konečný stav, nebo hodnota získaná z několika posledních stavů. [18, 19]

4.9 Výpočet podmíněných pravděpodobností metodou maximální věrohodnosti

Jedním ze základních problémů, při implementaci Gibbsova vzorkování je to, že potřebujeme znát jednotlivé podmíněné pravděpodobnosti

$$p(x_i|\mathbf{x}_{-i}) \quad (18)$$

pro každou proměnnou x_i , kterou chceme vyčíslit. V přístupu, který byl zvolen pro odhad jednotlivých pravděpodobnostních rozdělení je předpokládáno, že $p(x_i|\mathbf{x}_{-i})$ má normální pravděpodobnostní rozdělení. V dalším textu se zaměříme na výpočet střední hodnoty

pouze pro jednu proměnou $x_i = y$ při známém měření $\mathbf{x}_{-i} = \mathbf{x}$, kde \mathbf{x} je vektor o velikosti N .

Střední hodnota y_μ pravděpodobnostního rozdělení dané proměnné y , je závislá na ostatních proměnných \mathbf{x} . Odhad střední hodnoty pravděpodobnostního rozdělení můžeme zapsat jako

$$y_\mu(\mathbf{x}) = \boldsymbol{\theta}^T \cdot \mathbf{x}, \quad (19)$$

kde hodnoty jednotlivých polí vektoru $\boldsymbol{\theta}$ udávají, jak který prvek vektoru \mathbf{x} ovlivňuje střední hodnotu y_μ . Cílem je najít takový vektor $\boldsymbol{\theta}$, využitím kterého bude hodnota y_μ nejvíce odpovídat střední hodnotě pravděpodobnostního rozdělení $p(y|\mathbf{x})$. Mějme naměřená data

$$\mathcal{D} = \begin{pmatrix} y_1 & y_2 & \dots & y_K \\ \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_K \end{pmatrix}, \quad (20)$$

kde K je počet měření. Vektor $\boldsymbol{\theta}$, pomocí kterého střední hodnota $y_\mu(\mathbf{x})$ nejvíce odpovídá střední hodnotě pravděpodobnostního rozdělení $p(y|\mathbf{x})$ označíme jako $\boldsymbol{\theta}^*$. Pro hledání vektoru $\boldsymbol{\theta}^*$ lze využít metodu maximální věrohodnosti [18]. Využitím této metody nalezneme koeficienty, které nejvěrohodněji odpovídají modelu z naměřených dat. Věrohodnost je definovaná jako:

$$\mathcal{L}(\boldsymbol{\theta}; \mathcal{D}) = p(\mathcal{D}|\boldsymbol{\theta}). \quad (21)$$

Pomocí metody maximální věrohodností můžeme vypočítat $\boldsymbol{\theta}^*$ následující rovnicí:

$$\boldsymbol{\theta}^* = \operatorname{argmax}_{\boldsymbol{\theta}} p(\mathcal{D}|\boldsymbol{\theta}) = \operatorname{argmax}_{\boldsymbol{\theta}} p(y_1, x_1, \dots, y_K, x_K|\boldsymbol{\theta}), \quad (22)$$

Za předpokladu, že jsou na sebe jednotlivá měření nezávislá, můžeme sdruženou pravděpodobnost $P(\mathcal{D}|\boldsymbol{\theta})$ z rovnice (22) faktorizovat (tj. rozdělit na součin jednotlivých pravděpodobnostních rozdělení)

$$\operatorname{argmax}_{\boldsymbol{\theta}} p(y_1, x_1, \dots, y_K, x_K|\boldsymbol{\theta}) = \operatorname{argmax}_{\boldsymbol{\theta}} \prod_{i=1}^K p(y_i, x_i|\boldsymbol{\theta}). \quad (23)$$

Sdružená pravděpodobnost lze rozepsat jako:

$$\operatorname{argmax}_{\boldsymbol{\theta}} \prod_{i=1}^K p(y_i, \mathbf{x}_i|\boldsymbol{\theta}) = \operatorname{argmax}_{\boldsymbol{\theta}} \prod_{i=1}^K p(y_i|\mathbf{x}_i, \boldsymbol{\theta}) \cdot p(\mathbf{x}_i|\boldsymbol{\theta}). \quad (24)$$

Vzhledem k tomu, že podmíněná pravděpodobnost $p(\mathbf{x}_i|\boldsymbol{\theta})$ nezávisí na $\boldsymbol{\theta}$, můžeme předchozí rovnici přepsat do následujícího tvaru.

$$\operatorname{argmax}_{\boldsymbol{\theta}} \prod_{i=1}^K p(y_i|\mathbf{x}_i, \boldsymbol{\theta}) \cdot p(\mathbf{x}_i|\boldsymbol{\theta}) = \operatorname{argmax}_{\boldsymbol{\theta}} \prod_{i=1}^K p(y_i|\mathbf{x}_i, \boldsymbol{\theta}) \cdot p(\mathbf{x}_i). \quad (25)$$

Nyní je vidět že $\operatorname{argmax}_{\boldsymbol{\theta}}$ nezávisí na $p(\mathbf{x}_i)$ a tudíž lze předchozí rovnici zjednodušit na

$$\operatorname{argmax}_{\boldsymbol{\theta}} \prod_{i=1}^K p(y_i|\mathbf{x}_i, \boldsymbol{\theta}) \cdot p(\mathbf{x}_i) = \operatorname{argmax}_{\boldsymbol{\theta}} \prod_{i=1}^K p(y_i|\mathbf{x}_i, \boldsymbol{\theta}). \quad (26)$$

V dalším kroku využíváme skutečnosti, že argmax funkce je možné libovolně transformovat na jinou neklesající funkci. Předchozí rovnice lze tedy přepsat do následujícího tvaru:

$$\operatorname{argmax}_{\boldsymbol{\theta}} \prod_{i=1}^K p(y_i|\mathbf{x}_i, \boldsymbol{\theta}) = \operatorname{argmax}_{\boldsymbol{\theta}} \sum_{i=1}^K \ln p(y_i|\mathbf{x}_i, \boldsymbol{\theta}). \quad (27)$$

Při předpokladu Gaussova pravděpodobnostního rozdělení podmíněné pravděpodobnosti $p(y_i|\mathbf{x}_i, \boldsymbol{\theta})$, můžeme předchozí tvar přepsat následovně:

$$\operatorname{argmax}_{\boldsymbol{\theta}} \sum_{i=1}^K \ln p(y_i|\mathbf{x}_i, \boldsymbol{\theta}) = \operatorname{argmax}_{\boldsymbol{\theta}} \sum_{i=1}^K \ln \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_i - \boldsymbol{\theta}^T \cdot \mathbf{x}_i)^2}{2\pi\sigma^2}}. \quad (28)$$

Část výrazu $\frac{1}{\sqrt{2\pi\sigma^2}}$ neovlivňuje výsledný $\operatorname{argmax}_{\boldsymbol{\theta}}$ takže jí můžeme vynechat.

$$\operatorname{argmax}_{\boldsymbol{\theta}} \sum_{i=1}^K \ln e^{-\frac{(y_i - \boldsymbol{\theta}^T \cdot \mathbf{x}_i)^2}{2\pi\sigma^2}} = \operatorname{argmax}_{\boldsymbol{\theta}} \sum_{i=1}^K \frac{-(y_i - \boldsymbol{\theta}^T \cdot \mathbf{x}_i)^2}{\sqrt{2\pi\sigma^2}}. \quad (29)$$

Jmenovatel opět neovlivní výsledek a tudíž můžeme výsledný výraz zjednodušit na

$$\operatorname{argmax}_{\boldsymbol{\theta}} \sum_{i=1}^K \frac{-(y_i - \boldsymbol{\theta}^T \cdot \mathbf{x}_i)^2}{\sqrt{2\pi\sigma^2}} = \operatorname{argmax}_{\boldsymbol{\theta}} \sum_{i=1}^K -(y_i - \boldsymbol{\theta}^T \cdot \mathbf{x}_i)^2 = \operatorname{argmin}_{\boldsymbol{\theta}} \sum_{i=1}^K (y_i - \boldsymbol{\theta}^T \cdot \mathbf{x}_i)^2. \quad (30)$$

Čímž získáváme rovnici pro výpočet optimální hodnoty $\boldsymbol{\theta}^*$:

$$\boldsymbol{\theta}^* = \operatorname{argmin}_{\boldsymbol{\theta}} \sum_{i=1}^K (y_i - \boldsymbol{\theta}^T \cdot \mathbf{x}_i)^2. \quad (31)$$

Tento výraz lze také vypočítat maticově využitím pseudoinverze.

$$\boldsymbol{\theta}^* = X^\dagger Y, \quad (32)$$

kde $X = (\mathbf{x}_1 \ \mathbf{x}_2 \ \dots \ \mathbf{x}_K)^T$ a $Y = (y_1 \ y_2 \ \dots \ y_K)^T$. Využitím takto získaného vektoru $\boldsymbol{\theta}^*$ v rovnici (19) získáme odhad střední hodnoty $p(y|\mathbf{x})$. K vytvoření pravděpodobnostního rozdělení je třeba určit rozptyl σ^2 . Pro určení rozptylu bylo využito střední hodnoty a jednotlivých hodnot ze sady dat. Rozptyl lze vypočítat jako

$$\sigma^2 = \frac{1}{K} \sum_{i=1}^K (\boldsymbol{\theta}^{*T} \mathbf{x}_i - y_\mu)^2, \quad (33)$$

kde K značí počet měření v sadě dat \mathcal{D} , \mathbf{x}_i je i -tý vektor \mathbf{x} ze sady dat \mathcal{D} , $\boldsymbol{\theta}^*$ je získána z rovnice (32) a y_μ je střední hodnotu pravděpodobnostního rozdělení, získaná pomocí rovnice (19).

4.10 Algoritmus pro aktivní vnímání

Na základě předchozí kapitoly byla provedena úvaha o tom, jakým způsobem ovlivňuje $\boldsymbol{\theta}^*$ danou střední hodnotu. Musíme si uvědomit, že pro každý prvek i vektoru příznaků o velikosti M , existuje vektor $\boldsymbol{\theta}_i^*$ o velikosti $M - 1$. V tomto vektoru je obsažena informace, udávající jak je hodnota daného prvku vektoru, ovlivněna hodnotami ostatních prvků. Tato skutečnost vychází z rovnice

$$x_{i|\mu} = \boldsymbol{\theta}_i^{*T} \cdot \mathbf{x}_{-i} = (\theta_{i|1} \quad \theta_{i|2} \quad \dots \quad \theta_{i|M-1}) \cdot \begin{pmatrix} x_1 \\ \vdots \\ x_{i-1} \\ x_{i+1} \\ \vdots \\ x_M \end{pmatrix}. \quad (34)$$

Kromě této informace, existuje ještě opačná informace, která se dá využít. Nejen že je hodnota prvku x_i závislá na hodnotách ostatních prvků z vektoru \mathbf{x}_{-i} , ale i hodnoty ostatních prvků jsou přímo ovlivněny hodnotou x_i . Informace jak daný prvek ovlivňuje ostatní prvky je ukryta ve vektorech $\boldsymbol{\theta}_j^*$, $j \neq i$, náležícím ostatním prvkům vektoru příznaků. Následující rovnice ukazuje případ kdy $i < j$

$$x_{j|\mu} = \boldsymbol{\theta}_j^{*T} \cdot \mathbf{x}_{-j} = (\theta_{j|1} \quad \dots \quad \theta_{j|i} \quad \dots \quad \theta_{j|M-1}) \cdot \begin{pmatrix} x_1 \\ \vdots \\ x_i \\ \vdots \\ x_{j-1} \\ x_{j+1} \\ \vdots \\ x_M \end{pmatrix}. \quad (35)$$

Prvek $\theta_{j|i}$ udává, jak je střední hodnota hodnota prvku $x_{j|\mu}$ závislá na hodnotě x_i .

Mějme množinu M prvků, kde prvky jsou jednotlivé pole vektoru příznaků. Označme množinu obsahující všech M prvků (indexů) písmenem C . Tato množina lze rozdělit na dvě podmnožiny. Podle toho, jestli je hodnota v poli vektoru příznaků známá nebo není. Necht' množina N označuje obsahuje prvky s neznámými (odhadnutými) hodnoty a množina Z obsahuje prvky kde jsou hodnoty známé (naměřené), s tím že platí:

$$N \cap Z = \emptyset, \quad (36)$$

$$N \cup Z = C. \quad (37)$$

Na základě provedené úvahy byly vytvořeny dvě hodnoty, které jsou počítány pro každý prvek z množiny N . Na základě těchto dvou hodnot je následně určeno, které pole vektoru příznaků bude nejlepší prozkoumat příští akcí manipulátoru. Těmito hodnotami je:

- Přesnost (accuracy),
- Užitečnost (usefulness).

Přesnost měření prvku j je definována následovně. Jedná se o součet všech polí $\theta_{j|i}^*$ z vektoru $\boldsymbol{\theta}_j^* = (\theta_{j|1}, \theta_{j|2}, \dots, \theta_{j|M-1})$, kde $i \in Z$. Přesnost odhadu hodnoty pole s indexem j můžeme spočítat pomocí následující rovnice:

$$acc_j = \sum_{i \in Z} \theta_{j|i}^*, \quad (38)$$

kde $\theta_{j|i}^*$ je i -tý prvek vektoru $\boldsymbol{\theta}_j$. Jedná se o hodnotu udávající jak k výpočtu střední hodnoty pole s indexem j přispěla ostatní známá pole vektoru příznaků. Dále platí, že čím je větší přesnost daného prvku, tím méně ho při modelování ovlivní měření ostatních polí.

Druhou důležitou hodnotou je užitečnost. Prvek s největší užitečností odpovídá poli ve vektoru příznaků, které po prozkoumání nejvíce zvýší přesnost ostatních prvků. Užitečnost prvku j je součet všech $\theta_{i|j}^*$, kde $i \in N$, kterými prvek j přispívá polím s indexy i , při výpočtu střední hodnoty. Výpočet užitečnosti prvku j lze zapsat jako:

$$use_j = \sum_{i \in N} \theta_{i|j}^*, \quad (39)$$

kde $\theta_{i|j}^*$ je j -tý prvek vektoru $\boldsymbol{\theta}_i$. Hodnota užitečnosti nám říká, který bod nejvíce ovlivní výpočet středních hodnot neznámých polí, při jejich modelování.

Kombinací užitečnosti a přesnosti získáváme rozhodující kritérium pro výběr prvku, který bude v dalším kroku vybrán pro haptickou exploraci. Zatímco pole s největší užitečností je výhodné prohledat, tak u pole s největší přesností neočekáváme výrazný rozdíl mezi modelovanou a měřenou hodnotou. Rovnice pro výběr akce aktivního vnímání je následující:

$$j^* = \operatorname{argmax}_j \sum_{i \in N} \theta_{i|j}^* - \sum_{i \in Z} \theta_{j|i}^* = \operatorname{argmax}_j use_j - acc_j, \quad (40)$$

kde j^* značí pole, které bude prozkoumáno.

Navržený algoritmus, na rozdíl od většiny ostatních metod aktivního vnímání, nevyužívá číselné hodnoty modelu pro výběr akce. V této metodě záleží pouze na tom, která pole jsou známa. Vzhledem k této skutečnosti je velmi důležité, aby data, ze kterých jsou počítány jednotlivé vektory $\boldsymbol{\theta}^*$, byla dobře naměřena. Je potřeba, aby byla data dostatečně pestrá a odpovídala reálnému světu. Vytvořením špatné sady dat tato metoda ztrácí účinnost.

5 Implementace

V této kapitole je podrobně popsáno řešení zadaných problémů. Pro vyřešení níže popsaného problému je nutné splnit všechny body ze zadání práce. Nejprve je zde popsán obecně přístup k řešení zadaných úloh. V dalších částech této kapitoly je postupně popsáno řešení jednotlivých úkolů. V rámci řešení globálního problému jsou zde postupně popsány tyto implementace řešící jednotlivé úlohy: inverzní kinematika manipulátoru (kapitola 5.3.1), servis pro zjištění výšky a tuhosti terénu (kapitola 5.3.2), algoritmus aktivního vnímání (kapitola 5.4.4), Gibbsovo vzorkování (kapitola 5.4.5) a propioceptivní mapování (kapitola 5.4.6).

5.1 Rozbor zadaného problému

Jedním z úkolů této diplomové práce byla implementace programu, který dokáže s využitím aktivního vnímání zrekonstruovat terén v okolí robotu. Na toto zadání se dá dívat třemi pohledy, s rozdílnými rozsahy rekonstrukce.

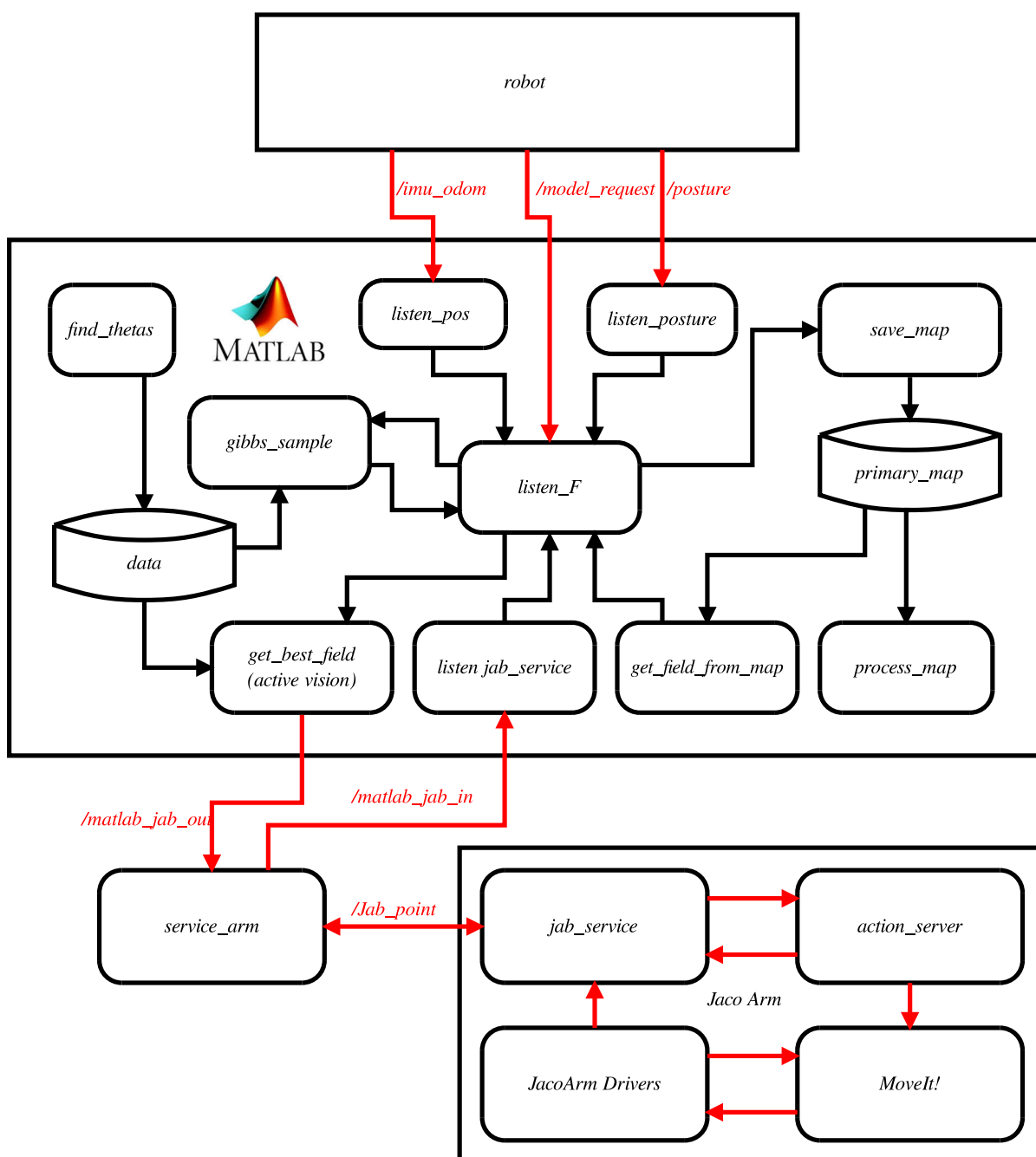
1. Rekonstrukce z části zakrytého DEM.
2. Rekonstrukce celého DEM.
3. Částečná rekonstrukce globální mapy.

První z pohledů si můžeme představit tak, že robot pomocí laseru nedokáže zjistit výšku terénu v daném místě. Taková situace může nastat, pokud je například na cestě kaluž a paprsky laseru se odrazí. Jiný případ kdy tato situace nastává je, když není laserem vidět těsně před robot, například kvůli vysokému naklonění robotu (*pitch*). Není tedy jisté, jestli je tento terén průjezdný. Využitím Gibbsova vzorkování a servisu pro zjištění výšky terénu v bodě pomocí robotické ruky, můžeme chybějící část terénu zrekonstruovat. V tomto případě je předpokládáno, že robot je v prostředí, kde senzory fungují normálně a tudíž si pomocí laseru vytváří mapu, kterou doplní o rekonstruované hodnoty.

Druhý a třetí pohled na úlohu rekonstrukce terénu spolu úzce souvisí. Představa u těchto dvou bodů je následující. Robot nemá žádné body z laseru a tudíž je celý DEM neznámý. Tato situace může nastat například při silném zadýmení prostoru, nebo při hustém dešti. My tuto situaci modelujeme vymazáním daných hodnot z vektoru příznaků po jeho přijmutí. DEM můžeme pomocí Gibbsova vzorkování modelovat využitím sensorických dat z vektoru příznaků (naklonění robotu, proudy v motorech). V porovnání s předchozím přístupem je zde celý DEM neznámý, máme tedy 100 neznámých hodnot ve vektoru příznaků, které jsou vzorkovány a pouze 8 využívaných hodnot, které jsou známy. Využitím haptické

5. IMPLEMENTACE

explorace je možné změřit výšku některých polí z DEM, čímž je snížen počet vzorkovaných proměnných. Je důležité si uvědomit, že pracovní prostor manipulátoru není neomezený a z celého DEM je možné prohlédnout jen nepatrnou část. Například část umístěnou pod robotem není možné prozkoumat. Při implementaci byl zvolen takový přístup, kde využí-



Obrázek 8: Schéma souboru programů.

váme i hodnoty naměřené dříve. Po pohybu mobilního robotu vpřed se body, které byly prozkoumány před pohybem, posunou v DEM pod robot. Tyto body mohou být využity i k rekonstrukci nového DEM. Informace o prohledaných místech jsou ukládány do mapy.

Mapa je tvořena ze dvou důvodů. Jedním z těchto důvodů je získání doplňujících informací o DEM v místech, kam robot nedosáhne, ale už je dříve změřil. A druhým důvodem je získat částečný model prostředí, ve kterém se robot pohybuje. Úloha, která je zde řešena se nazývá proprioceptivní mapování. Proprioceptivní mapování si můžeme představit jako tvorbu mapy s využitím pouze vnitřních senzorů robotu. V našem případě se jedná o gyroskop, proudové odběry v pásech a ramenech mobilního robotu a proudové odběry v kloubech manipulátoru. Do této mapy jsou ukládány body z těsného okolí robotu, které byly vymodelovány nebo naměřeny. Pro určení globální polohy bodů využíváme IMU odometrii. Vytváření mapy přináší hlavní výhodu v tom, že při pohybu vpřed je možné získat z mapy výšku části DEM pod robotem, na kterou by robot nemohl pomocí ruky sáhnout. Nevýhoda této metody je taková, že IMU odometrie není přesná a to ve všech třech osách. Může se tedy stát, že změna výšky určená z odometrie není správná, nebo posun v osách x, y není přesný. Největší chyby odometrie vznikají při otáčení na místě a na površích, kde se pásy smýkají. Pro co nejmenší chyby v odometrii je nutné pohybovat se robotem opravdu pomalu a zamezit smýkání.

Tato kapitola se soustředí hlavně na problém rekonstrukce globální mapy. Jedná se totiž o komplexní problém, který řeší jak problém rekonstrukce celého DEM, tak rekonstrukci částečně zakrytého DEM. V případě, že bychom chtěli rekonstruovat pouze z části zakrytý DEM stačí pouze zavolat funkci pro Gibbsovo vzorkování.

Pro řešení problému rekonstrukce mapy byl vytvořen soubor programů. Tento soubor programů využívá k rekonstrukci aktivní vnímání a Gibbsovo vzorkování. Nápomocné je to, že se pro modelování DEM načítají i body, které již byli prohledány dříve. Pro lepší orientaci při popisu jednotlivých částí programu je na Obrázku 8 znázorněno schéma souboru programů a funkcí. Je důležité zdůraznit, že na tomto schématu nejsou zobrazeny všechny programy a ROS uzly, ale pouze ty nejdůležitější. Červeně jsou zde značeny ROS témata. Celé schéma se dá rozdělit na tři části. První část je pojmenovaná jako *robot*. V této části je zahrnuto mnoho ROS uzlů, které se starají o funkčnost robotu. Druhá část, označená *Matlab*, řeší problém tvorby modelu prostředí, Gibbsovo vzorkování a výběr akce pro haptickou exploraci. Poslední částí jsou programy starající se o provedení haptické explorace a pohyb manipulátoru. Tyto tři části, jsou v dalším textu podrobněji popsány.

5.2 Robot

Část schématu označená jako *robot* (Obrázek 8 nahoře) nebyla mnou implementována. Jedná se ale o nezbytně nutnou část pro fungování celého konceptu. Uvnitř této části

5. IMPLEMENTACE

jsou ovladače a uzly ROS, které zajišťují funkcionalitu mobilního robotu. Konkrétně jde o uzly zajišťující výpočet odometrie, tvorbu mapy ze shluků 3D bodů získaných z laseru, tvorba DEM, vysílání vektoru příznaků, transformace mezi jednotlivými částmi robotu a podobně. V našem programu je pro nás důležité odebírat vypočtenou polohu robotu (ROS téma *imu_odom*), vektor příznaků (ROS téma *model_request*), a mód konfigurace ramen mobilního robotu (ROS téma *posture*).

No robotu se spustí potřebné ROS uzly pomocí příkazů:

```
export ROS_IP=192.168.2.115 // IP robotu
roslaunch inso inso.launch
roslaunch adaptive_traversability reinf_process_corrMATLAB3_robot.launch
```

První řádek je důležitý k tomu, aby zprávy na tématech, které jsou přijímány v matlabu byly viditelné. Podpůrné programy se spustí na počítači pomocí následujících příkazů:

```
export ROS_MASTER_IP=192.168.2.115 // IP robotu
export ROS_IP=192.168.2.175 // IP PC
roslaunch nifti_mapping_launchers mapAndNav.launch
roslaunch adaptive_traversability reinf_process_corr_base.launch
```

První dva řádky zajistí připojení k ROS robotu a viditelnost publikovaných témat v prostředí matlab.

5.3 Manipulátor

V této části kapitoly je popsána část schématu označená *Jaco Arm*. Ve schématu na Obrázku 8 je tato část zobrazena dole. Je v ní řešen pohyb manipulátoru a to jak pro haptickou exploraci, tak obecně pro pohyb robotické ruky v prostředí shluků 3D bodů. Část *Jaco Arm* můžeme rozdělit do čtyř komponent zobrazených ve schématu, kde některé komponenty označují jeden script a jiné celý soubor programů. Každá z komponent má ale svou důležitou funkcionalitu. Komponenty označené jako *"MoveIt!"* a *action_server* slouží k plánování pohybu manipulátoru. Tyto dvě části jsou popsány v kapitole 5.3.1. Další komponentou je *jab_service*, který slouží k získání informace o výšce terénu. Tento servis je popsán v kapitole 5.3.2. Poslední komponentou jsou ovladače robotické ruky (*JacoArm Drivers*). Ovladače jsou nutné ke komunikaci s reálným hardware. Jsou využívány například k získání pozice manipulátoru, k získání momentů sil v jednotlivých kloubech nebo i

k získání výsledné kartézské síly v koncovém bodě. Ovladače jsou také potřebné k ovládní motorů v manipulátoru. Ovladače ruky jsou poskytnuty od Kinovy a až na nepatrné změny (např. posílání sil na dané ROS téma) nebyly upravovány.

Pro spuštění části programů, které řeší pohyb manipulátoru a haptickou exploraci je nutné na robotu spustit příkaz:

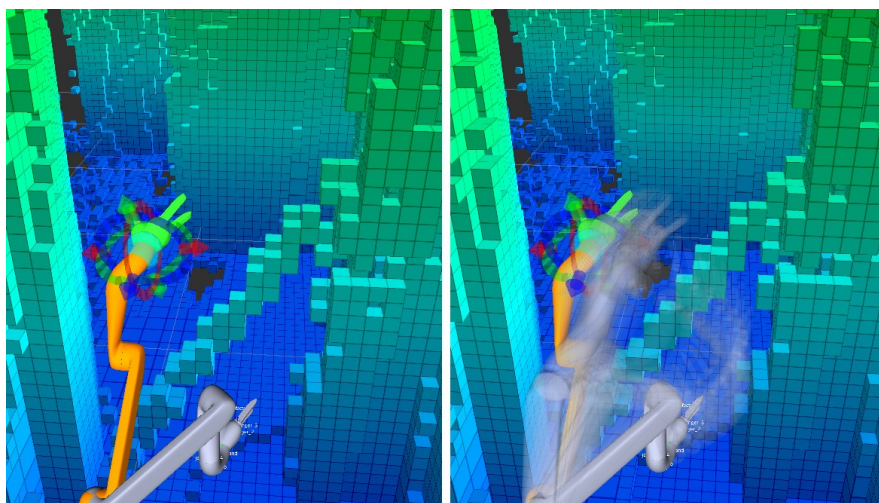
```
roslaunch jaco_moveit jabbing.launch
```

5.3.1 Inverzní kinematická úloha v prostředí s překážky

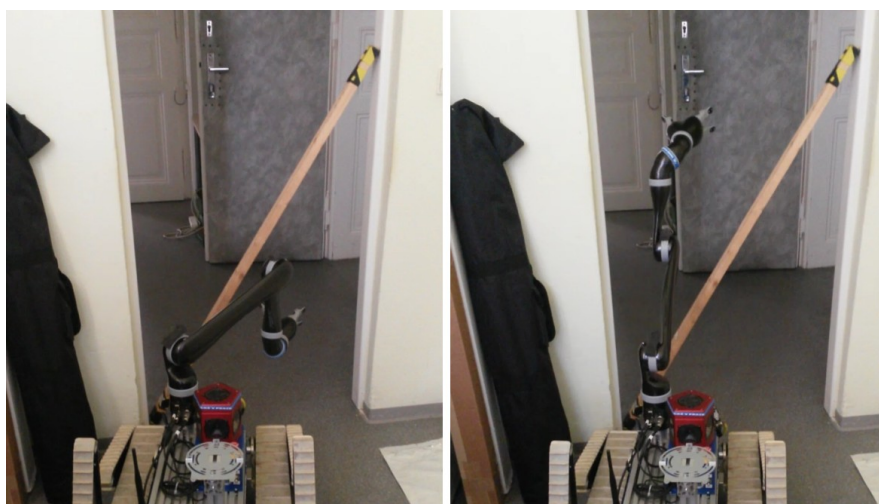
Jednou z důležitých částí této práce byla implementace inverzní kinematiky robotického manipulátoru Jaco Arm. Pro splnění tohoto úkolu byla vybrána knihovna "MoveIt!" (kapitola 4.5) ze systému ROS (kapitola 4.4). Při implementaci kinematiky pomocí "MoveIt!" v prostředí ROS bylo vycházeno z předchozí práce Matěje Balgy a bylo postupováno podle návodu umístěného na stránkách [4]. K přípravě knihovny "MoveIt!" pro konkrétní manipulátor, bylo využito MoveIt setup assistant. Jedná se o nastavení ROS parametrů (*ROS Param Server*) z Obrázku 6. Pomocí tohoto programu se nastaví jednotlivé kloubové limity, typy kloubů, skupiny manipulátoru a významné polohy manipulátoru. Zároveň se do "MoveIt!" nahraje také *URDF* popis manipulátoru. V *SRDF* popisu jsme rozdělili robotickou ruku na dvě skupiny. Jednou z nich je koncový efektor, obsahující pouze prsty robotické ruky. Druhou skupinou je manipulátor, který obsahuje všech šest hlavních kloubů. Při dalším plánování trajektorií je dále plánován pohyb pouze pro skupinu manipulátor, která má šest stupňů volnosti. Překážky v prostředí, ve kterém se pohyb manipulátoru plánuje, jsou reprezentovány pomocí OktoMapy. Tato OktoMapa je získávána ze statického shluku 3D bodů využitím *octomap_server*. Prostor obsazenosti, ve kterém plánovač hledá cestu, tvoří OktoMapa, předem definované objekty (*collision objects*) a model robotu.

Z uživatelského rozhraní konceptu "MoveIt!" (Obrázek 6) je využíváno *Rviz*, kde je možné mimo jiné vizualizovat plán a *moveit_commander*. Vizualizace plánu v *Rviz* je zobrazena na Obrázku 9. V *Rviz* je možné vytvořit cíl i pomocí interaktivní značky. Stačí v grafickém rozhraní vybrat polohu, které má manipulátor dosáhnout a spustit plánovač (tlačítko Plan). Po kontrole plánu je možné vykonat pohyb manipulátoru stisknutím tlačítka Execute. *Rviz* je používán hlavně pro debug a zobrazování naměřených hodnot. Plánování pohybu v *Rviz* v této práci dále není využíváno.

Jak již bylo zmíněno výše, kromě *Rviz* je z uživatelského rozhraní využíván také *moveit_commander*. Jeho instance je vytvořena v programu nazývaném *action_server*, který



(a) Vizualizace v prostředí *Rviz*. Počáteční poloha manipulátoru je značena bíle, koncová poloha oranžově a naplánovaná trajektorie průhledně.



(b) Počáteční poloha manipulátoru (vlevo) a koncová poloha manipulátoru (vpravo) v reálném světě.

Obrázek 9: Plánování pohybu využitím "MoveIt!".

je využíván pro komunikaci s *move_group*. Program *action_server* byl implementován v jazyce Python a je nezbytnou součástí vykonání pohybu. Tento program přijme cíl a nechá "MoveIt!" naplánovat trajektorii. Po kontrole správnosti trajektorie vyšle příkaz k vykonání pohybu. K plánování i vykonávání pohybu využívá právě výše zmíněného *move_group*. Program *action_server* dokáže přijmout cílovou polohu ve třech různých formátech. Jedním ze způsobů předání cíle je 3D bod (position). Druhým způsobem je 3D bod a kvaternion (pose), definující natočení konce manipulátoru. A třetí způsob je předání cíle v kloubových souřadnicích. Tento program dále rozhoduje, jestli je plán, který vygeneroval "Mo-

veIt!” vyhovující. ”MoveIt!” občas vygeneruje trajektorie, které nejsou optimální, například při pohybu koncového efektoru manipulátoru o pár centimetrů níže se může stát, že robot změni konfiguraci prvního kloubu, což je nežádoucí. Proto je kontrolováno, jestli je začátek a konec trajektorie kloubových souřadnic v daných mezích. V případě nenalezení plánu, je pokus pětkrát zopakován. Pokud není nalezen plán ani poté, vypíše se chybová hláška a cíl se zahodí. Jak již bylo uvedeno dříve, ”MoveIt!” využívá knihovnu plánovačů OMPL. Tato knihovna obsahuje velké množství algoritmů pro generování trajektorie. Při implementaci byl jako základní plánovač zvolen algoritmus KPIECE (z anglického Kinodynamic Planning by Interior-Exterior Cell Exploration), a to z důvodu velmi rychlého nalezení řešení.

5.3.2 Servis pro zjištění výšky terénu

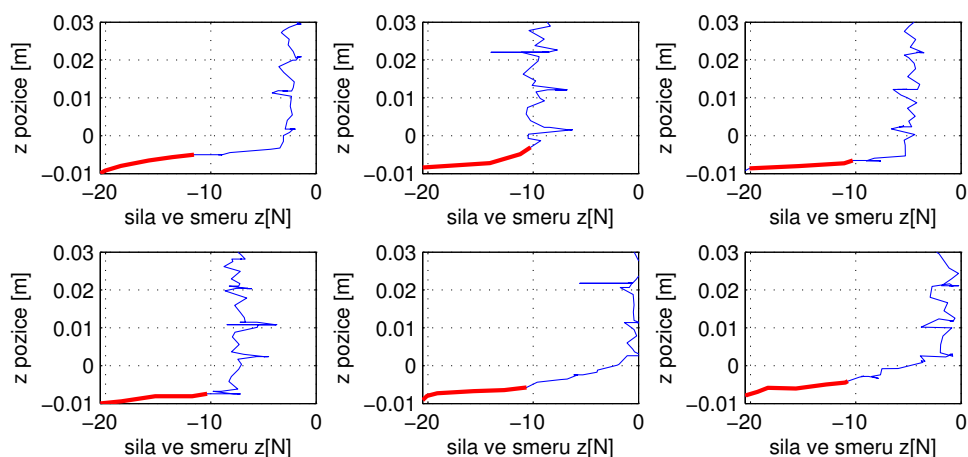
Pro zjištění výšky terénu pomocí robotického manipulátoru byl implementován ROS servis. Tento program je implementován v programovacím jazyce Python a je pojmenován jako *jab_service* (viz. Obrázek 8). Servis na vstupu očekává x, y, z souřadnice vztažené k základně robotického manipulátoru. Výstup servisu je boolean, označující prezenci překážky, a tři čísla typu float se souřadnicemi bodu, kde byla nalezena překážka. Souřadnice jsou vztažené k rámu manipulátoru.

Řízení pohybu robotu je rozděleno do dvou částí. V první části se koncový bod manipulátoru pohybuje podél osy z , od výšky zadané na vstupu až po překážku nebo konec pracovního prostoru manipulátoru. V druhé části se manipulátor vrací do takové pozice, ve které je mobilní robot stabilní a ruka nepřekáží ve výhledu laseru a ostatním sensorům. Při nenalezení plánu pro další prohledávání v ose z vrátí program false a souřadnice $(0, 0, 0)$. Pro pohyb robotu je využitý výše popsáný *action_server*. Cíl plánu pohybu se pro první část programu předává jako 3D bod a kvaternion. Pro návrat do stabilní pozice je využit cíl v podobě kloubových souřadnic. Pohyb podél osy z není plynulý a to hlavně kvůli stabilizaci sil v jednotlivých kloubech manipulátoru. Cíl pro další měření kontaktu se vždy generuje o 1 cm pod minulým cílem. Po každém zastavení manipulátoru se zkontroluje síla a pokud je velikost síly působící na koncový bod manipulátoru menší než $-20 N$, je tento bod prohlášen za bod kontaktu s terénem.

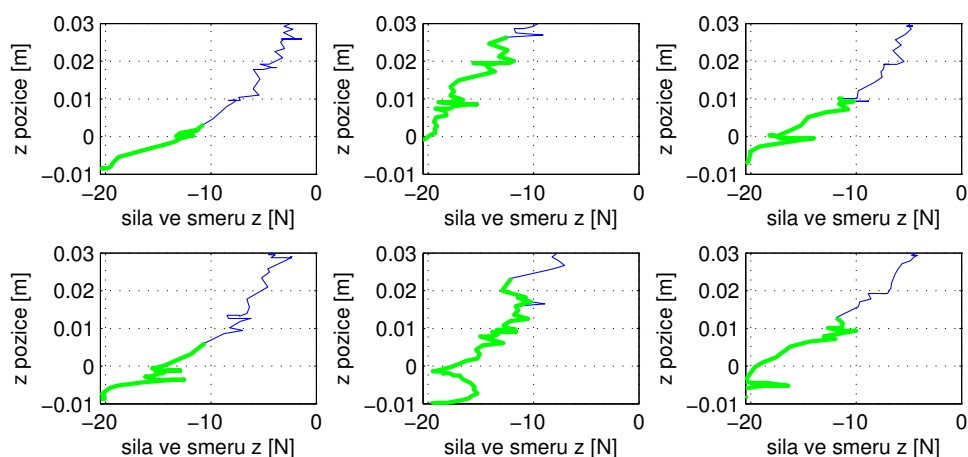
Součástí tohoto programu je také klasifikace tuhosti terénu, který je aktuálně prohledáván. Rozdělili jsme klasifikované objekty (prostředí) do dvou tříd, v závislosti na tuhosti objektu. Mějme tedy třídu měkkých objektů a třídu tvrdých objektů. Do měkkých objektů spadají materiály jako je molitan, zmuchlaný textil a další poddajné materiály. Do třídy tvrdých objektů spadají materiály, které jsou nepoddajné, například beton, dřevo nebo udusaná hlína. Servis odebírá ROS témata s informací o aktuální pozici koncového bodu (*jaco_arm_driver/out/tool_position*) a s informací o síle v koncovém bodu manipulátoru

5. IMPLEMENTACE

(*jaco_arm_driver/out/tool_wrench*). Z těchto zpráv jsou čteny informace o souřadnici z . Jsou vytvořena dvě pole typu fronta o velikosti 50. Jedno pole je pro ukládání sil a druhé pro ukládání výšky koncového bodu. Pokaždé, když přijde zpráva na dané téma, uloží se do odpovídajícího pole. Při prohledávání souřadnic x, y podél osy z se po každém provedení pohybu kontroluje, jestli síla v koncovém bodu manipulátoru je menší než -20 N . Je-li tomu tak, spustí se proces klasifikace tuhosti. Při klasifikaci tuhosti se prochází pole s posledními 50 hodnotami sil a pozic. Z těchto polí se zjišťuje, jaký je rozdíl ve výšce, kde



(a) Měření síly v ose z během prohledávání různých polí DEM podél z souřadnice pro tvrdý povrch



(b) Měření síly v ose z během prohledávání různých polí DEM podél z souřadnice pro měkký povrch

Obrázek 10: Příklad kartézských sil při prohledávání šesti různých souřadnic v DEM pro tvrdý povrch (nahore) a pro měkký povrch (dole).

začala být velikost síly vyšší než při pohybu bez kontaktu. Rozdíl počítáme mezi bodem, kde byla naměřena síla menší než $-20 N$ a prvním bodem, kde byla síla menší než $-10 N$. Pokud se síla mezi těmito body dostane nad hranici $-10 N$, počítá se velikost intervalu až od nového překročení hranice $-10 N$. Tyto intervaly jsou na Obrázku 10 zvýrazněny tučně. Rozdíl v z souřadnici mezi silou $-10 N$ a $-20 N$ jsme naměřili v různých místech pracovního prostoru pro objekty z obou tříd. Z naměřených hodnot bylo zjištěno, že klasifikaci lze provést na základě prahové hodnoty. Tato prahová hodnota byla nastavena na rozdíl $0,01 m$ mezi výškou, kde velikost síly začala být vyšší než v normálu a výškou při dotyku objektu. Pokud je rozdíl menší než daný práh, je klasifikovaný objekt prohlášen za tvrdý a naopak.

5.4 Matlab pipeline

Hlavní část celého programu obsahující funkce pro Gibbsovo vzorkování, výběr akce aktivního vnímání a mapování je napsána v programovacím jazyce matlab. V této části diplomové práce je popsána část souboru programů, která je uprostřed schématu na Obrázku 8. Postupně jsou zde popsány implementované funkce v prostředí matlab. Nejprve je v kapitole 5.4.1 popsána tvorba koeficientů θ^* z naměřených dat. Dále jsou v kapitole 5.4.2 popsáni odběratelé ROS témat. Mezi těmito odběrateli je i hlavní část programu řídící ostatní důležité funkce jako je: aktivní vnímání (kapitola 5.4.4), Gibbsovo vzorkování (kapitola 5.4.5), práce s mapou (kapitola 5.4.6) nebo komunikace se servisem pro zjištění výšky terénu (kapitola 5.4.3).

Program v Matlabu lze spustit pomocí scriptu:

```
read_robot
```

Tento skript nejprve nastaví spojení s prostředím ROS tak, aby zprávy na daných tématech byly viditelné i v matlabu. Vzhledem k tomu, že tato část celého konceptu se spouští na jiném počítači než na robotu, je potřeba správně v matlabu nastavit IP adresu robotu (ROS MASTER IP) a počítače, na kterém běží program v matlabu (ROS IP). Skript inicializuje globální struktury a proměnné využívané v prostředí, a poté spustí základní odběratele ROS témat.

Knihovny ROS pro matlab jsou k dispozici stále v omezené verzi. Momentálně není možné v prostředí matlab implementovat uzel typu servis, a stejně tak není možné servis z prostředí zavolat. Tento problém se projevuje v situaci, kdy je potřeba zavolat servis pro zjištění výšky terénu v daných souřadnicích.

5.4.1 Data

Pro Gibbsovo vzorkování i pro výběr další akce při aktivní haptické exploraci bylo potřeba naměřit dostatek dat. Z těchto dat se později určují střední hodnoty pravděpodobnostních rozdělení potřebných pro metodu Gibbsova vzorkování. Zaznamenáván byl vektor příznaků a odpovídající konfigurace ramen mobilního robotu. Data byla naměřena při různých situacích, ve kterých se robot může vyskytnout a to jak v interieru na rovném terénu s překážkami, tak v exteriéru na zvlněném terénu. Z naměřených vektorů příznaků byly odstraněny rychlosti, které dle předpokladu nemají vliv na výšku rekonstruovaných polí DEM.

Z naměřených dat jsou pro každé pole vektoru příznaků s indexem i vypočteny vektory θ_i^* a rozptyl σ_i^2 . Pro výpočet hodnot θ_i^* je využito rovnice (32). Vektory θ_i^* jsou uloženy v matici velikosti 107×108 .

$$\Theta = (\theta_1^* \ \theta_2^* \ \dots \ \theta_{108}^*) \quad (41)$$

Je důležité zdůraznit, že výpočet jednotlivých vektorů Θ je oddělen pro každou konfiguraci ramen mobilního robotu. Kromě matice Θ , je pro každou konfiguraci spočten vektor σ^2 , velikosti 108.

$$\sigma^2 = (\sigma_1^2 \ \sigma_2^2 \ \dots \ \sigma_{108}^2) \quad (42)$$

Tento vektor je vypočten pomocí rovnice (33) a udává rozptyl jednotlivých hodnot v sadě dat. Hodnoty Θ a σ^2 pro každou konfiguraci jsou uloženy do globální struktury označované jako *data*, ze které jsou dále využívány při Gibbsově vzorkování a aktivním vnímání.

5.4.2 ROS odběratelé

Základním prvkem pro získání informací o stavu mobilního robotu jsou odběratelé ROS témat. Jak již bylo zmíněno, tyto odběratelé jsou spuštěny skriptem v matlabu s názvem *read_robot*. Jedná se o tři odběratele přijímající informace z mobilního robotu a jednoho odběratele, který očekává informaci o výšce terénu získané pomocí haptické explorace:

- odběratel pozice (*listen_pos*),
- odběratel konfigurace ramen mobilního robotu (*listen_posture*),
- odběratel vektoru příznaků (*listen_F*),
- odběratel odpovědi ze servisu pro zjištění výšky terénu. (*listen_jab_service*).

Odběratel pozice i odběratel konfigurace jsou jednoduché funkce, které reagují na příchozí zprávu tím, že data ze zprávy uloží do globální proměnné. Odběratel pozice poslouchá na tématu *imu_odom* a ukládá do proměnné *pos*. Odběratel konfigurace mobilního robotu čte data na tématu *posture* a ukládá je jako *mode_c*. Odběratel odpovědi ze servisu pro zjištění výšky terénu je součástí pseudoservisu, který je popsán v kapitole 5.4.3.

Odběratele vektoru příznaků lze označit za stěžejní část celého programu v prostředí matlab. Tento odběratel očekává na tématu *model_request* zprávu s vektorem příznaků, na kterou reaguje. Pseudokód, popisující proces po příchodu vektoru příznaků, je popsán v Algoritmu 2. Ve verzi programu, kterou jsme implementovali, není kladen důraz na autonomnost robotu. Očekává se, že robot je ovládán operátorem, a to jak jeho pohyb, tak příkaz k haptické exploraci. Po příchodu vektoru příznaků do funkce je operátor dotázán na počet akcí, které má manipulátor vykonat. Operátor tedy zváží jak přesně potřebuje zmapovat prostředí a v závislosti na této skutečnosti, zadá číslo do konzole. Poté odběratel počká na nový vektor příznaků, se kterým začne dál pracovat. Důvodem k vyčkání na novou zprávu je, že vektor příznaků je publikován každé tři vteřiny a pokud bychom pracovali se starým vektorem příznaků, mohli bychom zpracovávat data, která již nejsou aktuální. Po zadání počtu akcí manipulátoru je očekáváno, že operátor nebude pohybovat mobilním robotem do té doby než akce manipulátoru skončí. V tuto chvíli se uloží aktuální pozice a mód konfigurace mobilního robotu z globálních proměnných do proměnné ve funkci, čímž zajistíme, aby se při běhu programu nezměnily hodnoty. V dalším kroku se upraví vektor příznaků tím způsobem, že se odstraní hodnoty polí DEM získané pomocí laseru. Důvodem k odstranění dat je to, že v tomto případě vytváříme mapu pouze z proprioceptivních senzorů.

Odběratel vektoru příznaků postupně zavolá funkce pro vykonání těchto akcí:

1. získání dříve naměřených dat z mapy,
2. výběr příští akce manipulátoru,
3. haptická explorace,
4. modelování zbývajících terénu pomocí Gibbsova vzorkování,
5. uložení naměřených dat do mapy.

Jednotlivé uvedené funkce jsou podrobněji popsány v dalším textu. Funkcí odběratele vektoru příznaků je správa dat mezi funkcemi a integrace všech funkcí do jednoho celku, řešícího tvorbu mapy.

Algoritmus 2: Odběratel vektoru příznaků.

```
Input:  $F$  vektor příznaků
if notset(jabnum) then
  | jabnum  $\leftarrow$  získej číslo od operátora
else
  |  $F \leftarrow$  smaž výšky polí z  $F$ 
  |  $DEM, DEM\_missing \leftarrow$  get_fields_from_map(pos)
  |  $F \leftarrow$  přepiš odpovídající pole  $F$  známými hodnotami  $DEM$ 
  | for  $i = 1, 2, \dots, jabnum$  do
  | |  $r, s \leftarrow$  find_best_field(DEM_missing, mode_c, data)
  | | publikuj  $r, s$  na ROS téma matlab_jab_out
  | | while  $\neg jab\_done$  do
  | | | wait // čeká na změnu globální proměnné jab_done
  | | end
  | |  $jab\_done \leftarrow false$ 
  | |  $F \leftarrow$  vlož změřenou výšku do  $F$ 
  | end
  |  $DEM, DEM\_var \leftarrow$  gibbs_sample( $F$ , mode_c, data)
  | save_map( $DEM$ ,  $DEM\_var$ , pos)
  | unset(jabnum)
end
```

5.4.3 Pseudoservis

Pod pojmem pseudoservis si můžeme představit soubor funkcí nahrazující klasický ROS servis. Důvodem k tvorbě pseudoservisu je to, že z matlabu prozatím není možné zavolat ROS servis. Servis funguje na takovém principu, že po přijetí zprávy provede danou akci a na stejné ROS téma vrátí výsledek. Přímé volání servisu z prostředí matlab bylo nahrazeno využitím ROS odběratele v matlabu a Python skriptu.

Proces zavolání servisu začíná přímo ve funkci odběratele vektoru příznaků, kde se v daném místě publikuje zpráva se souřadnicemi pole v DEM, které má být prozkoumáno. Tato zpráva je publikována na ROS téma *matlab_jab_out*. Tím se dostáváme ke skriptu *service_arm*, který mimo jiné odebírá toto téma. Tento skript po přijetí zprávy, transformuje souřadnice DEM do x, y souřadnic manipulátoru. Na tématu *jab_point* se poté zavolá servis pro zjištění výšky terénu. Po provedení prozkoumání terénu tímto servisem (kapitola 5.3.2), získá *service_arm* odpověď se souřadnicemi bodu, kde byl nalezen kontakt. Vrácené souřadnice bodu jsou uvedené vůči základně robotického manipulátoru. Je tedy nutné bod transformovat vzhledem ke stabilizovanému souřadnicovému systému robotu, tak jak je tomu u všech bodů v DEM. Transformace je provedená využitím knihoven

ROS. V případě, že prohledání pole DEM neproběhlo úspěšně, je výška transformovaného bodu nastavena na konstantní hodnotu 10 *m*. V odběrateli vektoru příznaků se pak tato hodnota odchyťává jako neúspěšný pokus. Souřadnice *z* transformovaného bodu je z programu *service_arm* publikována na ROS téma *matlab_jab_in*. V prostředí matlab tuto zprávu přečte dříve zmíněný odběratel *listen_jab_service*. Tento odběratel nastaví globální proměnou s výškou zjištěného terénu na změřenou hodnotu a hodnotu *jab_done* nastaví na true. Změna hodnoty *jab_done* se projeví tak, že v odběrateli vektoru příznaků proběhne ukončení smyčky čekání a program pokračuje dál (viz. Algoritmus 2).

5.4.4 Aktivní vnímání

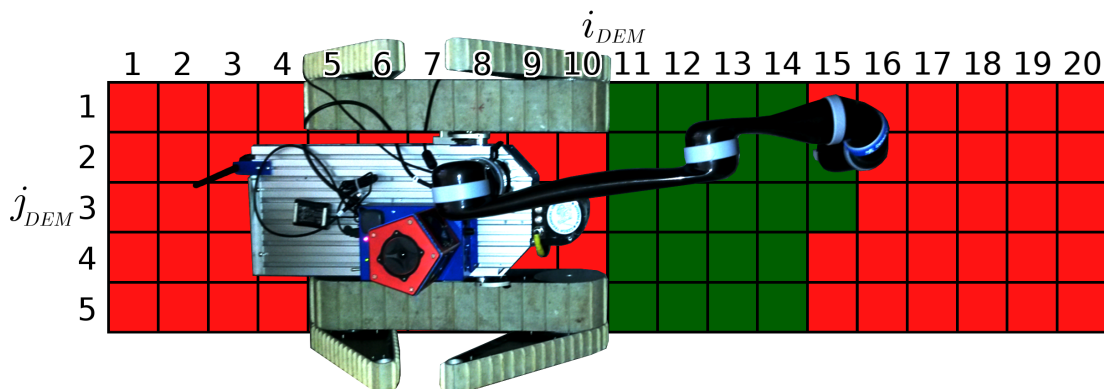
Aktivní vnímání zde představuje funkci zajišťující výběr akce manipulátoru, která bude nejužitečnější pro tvorbu modelu DEM. Při implementaci aktivního vnímání bylo využito skutečností popsaných v kapitole 4.10. Pro výběr pole, které je nejvýhodnější prozkoumat, bylo zvoleno kritérium užitečnosti a přesnosti. Rozdílem těchto dvou hodnot je získána hodnota H_j pro každé neznámé pole DEM j .

$$H_j = \sum_{i \in N} \theta_{i|j}^* - \sum_{i \in Z} \theta_{j|i}^*, \quad (43)$$

kde N je množina neznámých polí vektoru příznaků a Z je množina známých polí vektoru příznaků. Pro výpočet H_j se využívá celý vektor příznaků. Výběr pole j^* s maximální hodnotou H_j probíhá pouze z pracovního prostoru manipulátoru v DEM (viz. Obrázek 11).

$$j^* = \operatorname{argmax}_{j \in W} H_j, \quad (44)$$

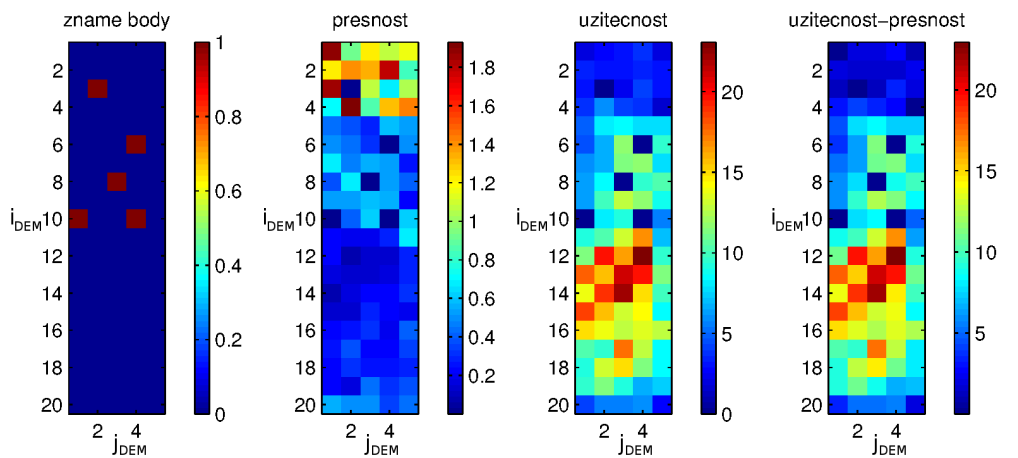
kde W značí množinu neznámých polí DEM, spadající do pracovního prostoru manipulátoru. Pracovní prostor je ovlivněn výškou, ve které začínáme prohledávat pole DEM.



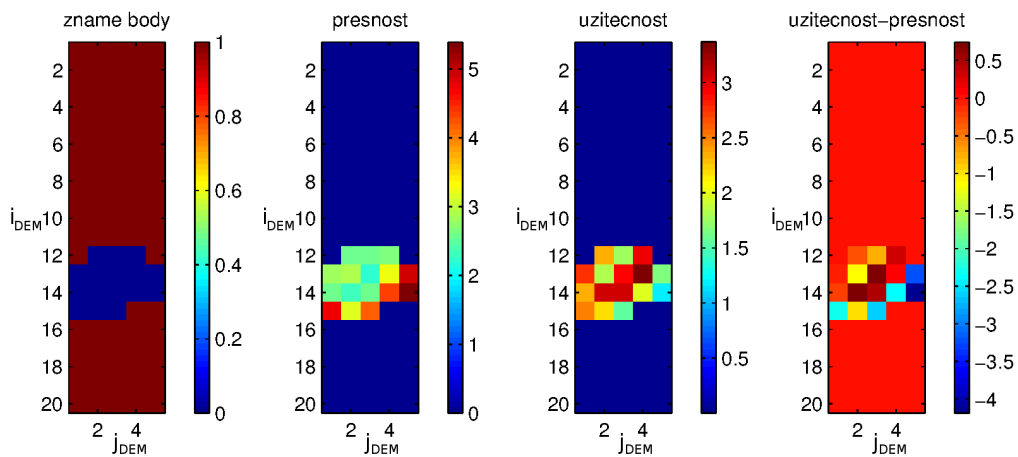
Obrázek 11: Pracovní prostor manipulátoru v DEM

5. IMPLEMENTACE

Příklad pracovního prostoru při prohledávání výšky terénu od 40 cm je zobrazen na Obrázku 11. V nejbezpečnějším případě by bylo vhodné prohledávat prostor od výšky, určené pracovním prostorem manipulátoru. Při experimentech byla tato hodnota nastavena na 40 cm z důvodu, že neočekáváme takto výraznou změnu terénu mezi robotem a prohledávaným polem DEM. V této implementované metodě záleží výběr příští akce na polích vektoru příznaků, které jsou známy, ale ne na jejich hodnotách. Pole vybrané pro další akci je vybráno hlavně v závislosti na tom, kam sáhl robot dříve a k jaké změně pozice robotu došlo mezi výběrem další akce.



(a) Hodnoty přesnosti, užitečnosti a jejich rozdílu při proprioceptivním mapování.



(b) Hodnoty přesnosti, užitečnosti v případě, že robot zná data z laseru a před ním se nachází neznámý terén (například kaluž).

Obrázek 12: Příklad hodnot spočítaných při dvou různých situacích.

Funkce pro výběr nejlepšího místa k prozkoumání (*get_best_field*) je implementována v programovacím jazyce matlab (Obrázek 8). Na vstupu očekává dvourozměrné pole s ozna-

čením neznámých polí DEM, konfiguraci ramen mobilního robotu a strukturu, obsahující koeficienty Θ . Výstupem z funkce jsou souřadnice pole DEM, které je dle algoritmu nejlepší prohledat.

Na Obrázku 12 jsou zobrazeny naměřené hodnoty pro jednotlivé pole DEM při dvou různých situacích. První situace nastává spíše při tvorbě mapy pouze z proprioceptivních dat. Při této situaci je známých polí DEM málo, což výrazně ovlivňuje spočítané hodnoty přesnosti i užitečnosti. Na Obrázku 12a je vidět, že hodnoty přesnosti jsou velmi nízké v porovnání s vysokými hodnotami užitečnosti. Výsledné hodnoty H_j jsou tak ovlivněny hlavně hodnotou užitečnosti. Druhá situace, zobrazená na Obrázku 12b, nastává hlavně v případě, že haptická explorace slouží k doplnění informace o neznámých polích DEM. Jedná se o případ, kdy využíváme data získaná z laseru a pomocí haptické explorace zjistíme výšku terénu v polích, kde je vysoká odrazivost, nebo nejsou v zorném poli laseru. Jako příklad si můžeme představit kaluž ležící před robotem. V tomto případě je vidět, že některé hodnoty přesnosti přesahují velikost hodnot užitečnosti. Výsledné hodnoty H_j jsou zde značně ovlivněny přesností jednotlivých polí.

5.4.5 Gibbsovo vzorkování

Při implementaci Gibbsova vzorkování bylo vycházeno z teorie popsané v kapitole 4.8. Iterativně je vybírán vzorek z jednotlivých podmíněných pravděpodobností $p(x_i|\mathbf{x}_{-i})$, kde \mathbf{x}_{-i} je upravený vektor příznaků. Vektor \mathbf{x}_{-i} je upravený tak, že jsou vynechány rychlosti pásů, prvek x_i a posledních 100 hodnot, označujících počet bodů, ze kterých je počítána výška odpovídajícího pole DEM (viz. kapitola 4.7).

$$\mathbf{x}_{-i} = (f_1 \ f_2 \ \dots \ f_{i-1} \ f_{i+1} \ \dots \ f_{101} \ f_{104} \ \dots \ f_{110})^T, \quad (45)$$

kde f_i značí hodnotu i -tého prvku vektoru příznaků. Jak již bylo řečeno v teoretické části této práce, předpokládáme že podmíněná pravděpodobnost $p(x_i|\mathbf{x}_{-i})$ má normální rozdělení se střední hodnotou rovnu $x_{i|\mu}$ s rozptylem σ_i^2 . Střední hodnota je vypočítána jako

$$x_{i|\mu} = \boldsymbol{\theta}_i^T \cdot \mathbf{x}_{-i}. \quad (46)$$

Kde $\boldsymbol{\theta}_i$ jsou koeficienty vypočítané z naměřené sady dat pomocí rovnice (32). Rozptyl σ_i^2 normálního pravděpodobnostního rozdělení je vypočítán využitím rovnice (33). Výběr vzorku x_i je pak realizován využitím následující rovnice.

$$x_i = \boldsymbol{\theta}_i^T \cdot \mathbf{x}_{-i} + N(0, \sigma_i^2), \quad (47)$$

kde $N(0, \sigma_i^2)$ značí vzorek vybraný z normálního rozdělení s nulovou střední hodnotou a rozptylem σ_i^2 .

V případě neznámých hodnot ve vektoru příznaků, jsou tyto hodnoty inicializovány náhodně z intervalu $\langle -0, 2; 0, 2 \rangle$. Výběr vzorku je proveden pro každou neznámou hodnotu upraveného vektoru příznaků, a to v každém kroku iterace. Odhadovaná střední hodnota náhodného vektoru $\mathbf{x} = (x_1 \ x_2 \ \dots \ x_{100})$ je určena jako střední hodnota vzorků za posledních 2400 iterací. Celkový počet iterací byl zvolen 2500, prvních 100 hodnot tedy nebereme při výpočtu v potaz. Stejně jako střední hodnota, tak i rozptyl jednotlivých polí je počítán z posledních 2400 vzorků.

Funkce zajišťující Gibbsovo vzorkování, je napsána v jazyce matlab. Jako vstup očekává vektor příznaků, mód konfigurace mobilního robotu a strukturu, obsahující Θ a rozptyly σ^2 . Výstup tohoto programu je *DEM* s doplněnými hodnotami na dříve neznámých polích a rozptyl těchto hodnot *DEM_var*. Pseudokód popisující algoritmus Gibbsova vzorkování je popsán na Algoritmu 3. Důležité je, že vektor příznaků je upravován tím způsobem, že jsou odstraněna pole obsahující rychlost pásů. Upravený vektor příznaků má tedy velikost 208.

Algoritmus 3: Gibbsovo vzorkování

Input: $(F, mode_c, data)$, kde F je vektor příznaků, $mode_c$ je konfigurace mobilního robotu a $data$ je struktura obsahující θ a σ^2 pro všechny konfigurace
Output: (DEM, DEM_var)
 $\Theta \leftarrow data(mode_c).theta$
 $\sigma^2 \leftarrow data(mode_c).sigma$
 $index \leftarrow$ získej indexy bodů, na kterých je F neznámý
 $F \leftarrow$ odeber rychlost z F
 $F(index) \leftarrow$ random
while $iter < 2500$ **do**
 foreach $i \in index$ **do**
 $F(i) \leftarrow$ vyber vzorek pomocí rovnice (47) pro θ_i a σ_i^2
 if $iter > 100$ **then**
 $v(i, iter - 100) \leftarrow F(i)$ // uloží $F(i)$ pro výpočet DEM , DEM_var
 end
 end
 $DEM \leftarrow mean(v)$
 $DEM_var \leftarrow var(v)$
end

5.4.6 Proprioceptivní mapování

Proprioceptivní mapování si můžeme představit jako tvorbu mapy pouze z vnitřních senzorů robotu. V našem případě se jedná o gyroskop, proudové odběry v motorech mobilního robotu a proudové odběry v kloubech manipulátoru. Mapa je primárně využívána pro ukládání naměřených hodnot s tím, že body z této mapy jsou využívány při modelování nových DEM. Důležité je, že takto lze získat i hodnoty v polích DEM, které jsou umístěné pod robotem a tudíž je není možné prozkoumat pomocí robotické ruky. Mapa bodů je uložena v globální struktuře. Struktura obsahuje souřadnice x , y a z každého bodu a hodnotu e , získanou jako rozptyl výšky pole při Gibbsově vzorkování. Pro celý proces mapování je důležitá odometrie. Z důvodu, že nevyužíváme data z laseru (viz. Kapitola 5.1), používáme nekorigovanou IMU odometrii, která do mapy zanáší chybu sama o sobě. Odometrie je publikovaná na ROS tématu *imu_odom*. Obsahuje pro nás podstatné informace o poloze robotu x , y , z a kvaternion určující natočení robotu. Z tohoto kvaternionu se počítají úhly *roll*, *pitch* a *yaw*.

Proces mapování je rozdělen do tří funkcí, kde každá plní jinou úlohu. První funkcí (*save_map*) je funkce k uložení naměřených a vymodelovaných dat do mapy. Druhá funkce (*get_field_from_map*) je využívána k vyjmutí bodů z mapy, které jsou v aktuálním okolí robotu. A třetí funkce (*process_map*) slouží pro tvorbu modelu finální mapy. Primární mapa může mít uloženo na jednom poli se souřadnicemi x , y několik různých bodů s výškou z_i a rozptylem e_i . Finální mapa, je taková mapa, kde je na každé souřadnici x , y maximálně jedna hodnota výšky z . K získání této výšky a k určení finální mapy, slouží právě tato funkce.

Pro uložení polí do mapy je, jak je již výše zmíněno, využívána funkce *save_map*. Tato funkce přijímá pole DEM, rozptyl jednotlivých polí získaný z Gibbsova vzorkování a pozici mobilního robotu, získanou z odometrie. Při tvorbě mapy jsou hodnoty x a y zaokrouhleny na decimetry. Vzhledem k tomu, že DEM má rozlišení 1 dm^2 na jedno pole a vzhledem k poměrně nepřesné odometrii, není tímto zaokrouhlením pozorovatelně zvýšená nepřesnost. Do mapy jsou ukládány takové body z DEM, které jsou umístěny v intervalu $i_{DEM} = \langle 5, 15 \rangle$. K přepočítání souřadnic DEM do globálních souřadnic je z kvaternionu vypočítán úhel natočení robotu podél osy z neboli *yaw*. Rovnice pro přepočet souřadnic DEM do globálních souřadnic vypadají následovně:

$$x_{global} = x_{odom} + 0,1 \cdot (i - i_{center}) \cdot \cos(yaw) + 0,1 \cdot (j - j_{center}) \cdot \cos(yaw - \pi/2), \quad (48)$$

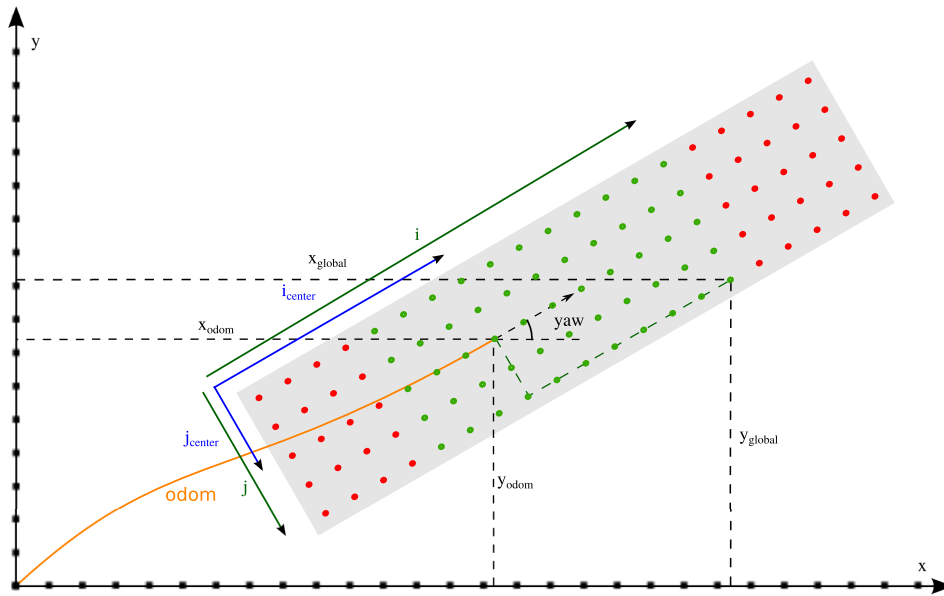
$$y_{global} = y_{odom} + 0,1 \cdot (i - i_{center}) \cdot \sin(yaw) + 0,1 \cdot (j - j_{center}) \cdot \sin(yaw + \pi/2), \quad (49)$$

$$z_{global} = z_{odom} + z_{dem}, \quad (50)$$

kde i , j značí souřadnice ukládaného bodu v DEM a $i_{center} = 8$, $j_{center} = 3$ jsou souřadnice středu robotu v DEM. x_{odom} , y_{odom} a z_{odom} značí pozici středu robotu v souřadnicovém

5. IMPLEMENTACE

systému mapy. Tyto hodnoty jsou získány z odometrie. Zatímco souřadnice x_{global} a y_{global} se zaokrouhlí na decimetry, hodnota souřadnice z_{global} je předána do primární mapy tak jak byla získána v rovnici (50). Ke každému bodu uloženému do mapy se uloží i rozptyl e , který byl vypočítán ve funkci pro Gibbsovo vzorkování. Pokud je bod objeven pomocí haptické explorační, je hodnota rozptylu nastavena na $e = 10^{-8}$. V případě, že byl bod získán při načítání bodů z mapy (viz funkce popsána níže), není již tento bod znovu ukládán. Pro lepší orientaci je na Obrázku 13 zobrazena situace ukládání bodu se souřadnicemi v DEM $i = 15, j = 5$.



Obrázek 13: Ukládání pole DEM do mapy.

Pro získání bodů z mapy je využívána funkce s názvem *get_field_from_map*. Tato funkce přijímá na vstup pozici robotu a jeho natočení. Výstupem z funkce je DEM, kde na místech kam robot dříve sáhl, jsou vyplněné hodnoty. V této funkci je využíváno toho, že mapa je zaokrouhlená na decimetry. Nejprve jsou pomocí rovnic (48) a (49) vypočítány souřadnice jednotlivých polí DEM v globálních souřadnicích. V dalším kroku je hledáno, jestli v mapě existuje bod prozkoumaný pomocí robotické ruky, který by měl stejné souřadnice x, y jako některá z globálních souřadnic polí DEM. Pokud je takový bod nalezen, vypočítá se jeho výška v DEM jako:

$$z_{dem_{ij}} = z_{global} - z_{odom}, \quad (51)$$

kde $z_{dem_{ij}}$ je výška daného bodu v DEM, z_{global} je výška bodu v globální mapě a z_{odom} je výška středu robotu získaná z odometrie.

Získání finální verze mapy není přímou součástí programu popsaného ve schématu na Obrázku 8. Funkce na zpracování mapy (*process_map*) není totiž spouštěna automaticky, ale spouští jí až operátor v případě, že chce vidět jak výsledná mapa vypadá. Na vstupu má tato funkce primární mapu a výstupem funkce je finální mapa. Primární mapa může obsahovat na daných souřadnicích x_{global}, y_{global} několik různých odhadů výšky. Cílem této funkce je určit výslednou hodnotu výšky terénu pro každé souřadnice mapy, které obsahují alespoň jednu uloženou výšku. Pro výpočet výšky mapy v jednotlivých místech je využito váženého průměru uložených výšek v daných souřadnicích x_{global}, y_{global} . Váhy jsou určeny rozptylem vzorkovaných hodnot při Gibbsově vzorkování. Kde čím nižší je rozptyl daných vzorků, tím vyšší je váha měření. Postupně jsou procházeny všechny souřadnice mapy, na které byla uložena alespoň jedna hodnota. Pro každé souřadnice splňující tuto podmínku je počítána finální výška mapy pomocí následující rovnice:

$$z_{final} = \frac{\sum_{i=1}^N \frac{z_i}{e_i}}{\sum_{i=1}^N \frac{1}{e_i}}, \quad (52)$$

kde N je počet uložených bodů v daných souřadnicích x_{global}, y_{global} v mapě, z_i je hodnota výšky bodu i uložená v mapě a e_i je rozptyl vzorků při odhadu výšky z_i (nebo konstanta v případě měření pomocí haptické explorační).

6 Experimenty

V této části diplomové práce jsou popsány jednotlivé experimenty zobrazující výsledky, kterých bylo dosaženo. V první části je popsán experiment ukazující přesnost zjištění výšky terénu (kapitola 6.1). V kapitole 6.2 je popsán experiment zobrazující úspěšnost klasifikace tuhosti terénu. Dále je zde zobrazen experiment zobrazující kvalitu algoritmu pro aktivní vnímání 6.3. V kapitole 6.4 jsou popsány experimenty zobrazující modely prostředí získané pomocí Gibbsova vzorkování při různých sensorických vstupech. V poslední části je popsán experiment zobrazující výslednou mapu a závislost kvality rekonstrukce mapy na počtu akcí manipulátoru (kapitola 6.5).

6.1 Přesnost měření výšky pomocí manipulátoru

Jedním z úkolů této diplomové práce bylo implementovat servis pro zjištění výšky terénu v souřadnicích souřadného systému robotu. Experiment provedený v této části, ukazuje odchylku měřených dat v DEM. Měření probíhalo ve čtyřech výškových úrovních terénu. Pro každou úroveň výšky bylo provedeno 10 měření na různých souřadnicích polí DEM s tím, že těchto 10 souřadnic bylo pro každou výškovou úroveň stejných. V dalším kroku byly spočítány odchylky jednotlivých měření od reálné hodnoty. Tyto odchylky jsou uvedeny v Tabulce 4.

č. měření	$z - z_{v1}$ [m]	$z - z_{v2}$ [m]	$z - z_{v3}$ [m]	$z - z_{v4}$ [m]
1	0,001	0,003	0,002	0,002
2	0,001	0,004	-0,001	0,000
3	0,009	-0,002	-0,005	-0,005
4	-0,001	-0,003	-0,005	-0,003
5	0,000	-0,002	0,000	-0,006
6	0,000	0,007	0,003	0,002
7	0,000	0,005	-0,009	0,003
8	-0,001	-0,002	-0,008	0,004
9	-0,002	0,002	-0,009	0,000
10	0,001	-0,005	-0,010	-0,004

Tabulka 4: Tabulka obsahující odchylky měření výšky od reálných hodnot. Kde z je měřená výška vztažená k podvozku a $z_{v1} = -0,154$ m, $z_{v2} = 0$ m, $z_{v3} = 0,154$ m, $z_{v4} = 0,264$ m jsou referenční výšky terénu vůči podvozku.

6. EXPERIMENTY

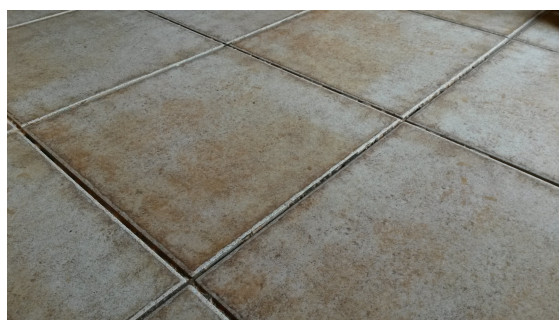
V tabulce je vidět, že maximální velikost odchylky je 1 *cm*. Chyby jsou způsobeny buď zkroucením pracovního nástroje v robotické ruce nebo nakloněním robota při měření. Směrodatná odchylka naměřených dat je rovna $\sigma = 0,0043$ *m*. Pokud bychom předpokládali normální rozdělení chyby, můžeme dle pravidla 2σ říct, že chyba určení výšky na 95 % nepřesáhne hodnotu $2\sigma = 0,0086$ *m*.

6.2 Klasifikace tuhosti terénu

Součástí programu pro zjištění výšky na daných souřadnicích souřadného systému robota je i zjištění tuhosti prozkoumávaného terénu. Tuhost může být reprezentována pomocí hodnoty rozdílu z souřadnice koncového efektoru, kde velikost síly ve směru z začala být větší než 10 *N* a výškou, kde velikost síly ve směru z byla větší než 20 *N*. Tento rozdíl závisí na tuhosti terénu nebo objektu následujícím způsobem. Čím je tento rozdíl větší, tím je objekt nebo terén měkký. Součástí implementovaného programu je klasifikace na měkké objekty a na tvrdé objekty v závislosti na hodnotě tohoto rozdílu. Pokud je hodnota větší nebo rovna 0,01 *m*, je objekt klasifikován jako měkký. Pokud je rozdíl menší, tak je objekt klasifikován jako tvrdý.



(a) Měřený objekt ze třídy měkkých objektů.



(b) Měřený objekt ze třídy tvrdých objektů.

Obrázek 14: Měřené objekty.

Tento experiment spočíval v naměření úspěšnosti klasifikace objektů. Bylo provedeno 15 měření pro obě třídy objektů. Těchto 15 měření bylo provedeno v různých bodech pracovního prostoru manipulátoru. Během měření byl manipulátor připevněn na mobilní robot. Měřené objekty jsou zobrazeny na Obrázku 14. Jednotlivé naměřené hodnoty pro měkký objekt jsou uvedeny v Tabulce 5. Hodnoty naměřené pro tvrdý objekt jsou uvedeny v Tabulce 6. Souřadnice x , y , z v těchto tabulkách označují bod, kde byl zjištěn kontakt s terénem. Hodnota označená jako rozdíl z , odpovídá rozdílu v z souřadnici ruky mezi první větší silou ve směru z a silou s velikostí ve směru z větší než 20. Poslední sloupec značí jestli klasifikátor správně klasifikoval třídu objektu.

6. EXPERIMENTY

č. měření	x [m]	y [m]	z [m]	rozdíl v z [m]	klasifikováno
1	0	-0,4	-0,241	0,017	správně
2	0	-0,5	-0,248	0,015	správně
3	0	-0,6	-0,251	0,011	správně
4	0,1	-0,6	-0,249	0,016	správně
5	0,1	-0,5	-0,243	0,020	správně
6	0,1	-0,4	-0,248	0,023	správně
7	0,1	-0,3	-0,254	0,001	špatně
8	-0,2	-0,3	-0,240	0,016	správně
9	0	-0,3	-0,260	0,006	špatně
10	0,3	-0,5	-0,250	0,009	špatně
11	0,4	-0,5	-0,249	0,014	správně
12	0,5	-0,5	-0,267	0,020	správně
13	-0,4	-0,5	-0,250	0,017	správně
14	0,2	-0,3	-0,240	0,020	správně
15	-0,3	-0,4	-0,251	0,015	správně

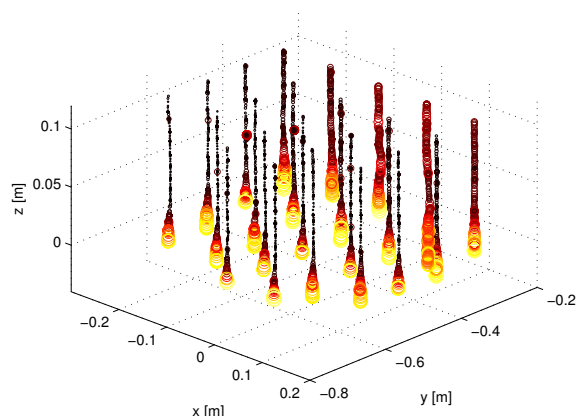
Tabulka 5: Měření kontaktu robotického manipulátoru s měkkými objekty.

č. měření	x [m]	y [m]	z [m]	rozdíl v z [m]	klasifikováno
1	0	-0,4	-0,292	0,004	správně
2	0	-0,5	-0,291	0,002	správně
3	0	-0,6	-0,297	0,011	špatně
4	0,1	-0,6	-0,298	0,012	špatně
5	0,1	-0,5	-0,297	0,002	správně
6	0,1	-0,4	-0,288	0,003	správně
7	0,1	-0,3	-0,290	0,003	správně
8	-0,2	-0,3	-0,297	0,003	správně
9	0	-0,3	-0,293	0,002	správně
10	0,3	-0,5	-0,299	0,002	správně
11	0,4	-0,5	-0,296	0,005	správně
12	0,5	-0,5	-0,300	0,003	správně
13	-0,4	-0,5	-0,302	0,002	správně
14	0,2	-0,3	-0,299	0,003	správně
15	-0,3	-0,4	-0,299	0,004	správně

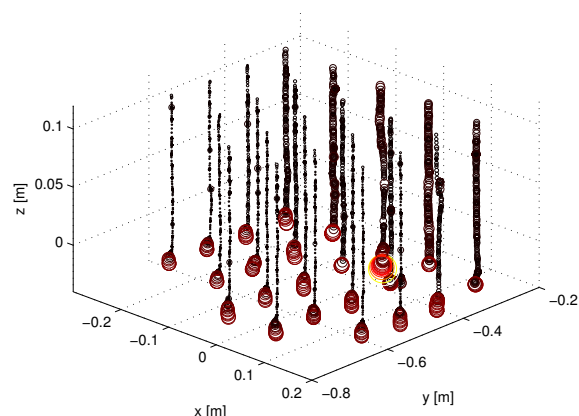
Tabulka 6: Měření kontaktu robotického manipulátoru s tvrdými objekty.

6. EXPERIMENTY

Z Tabulek 5 a 6 lze vidět několik informací. Na těchto změřených datech je hodnota úspěšnosti klasifikace měkkého objektu rovna 80 %. Pro tvrdý objekt je tato hodnota rovna 86,7%. Celková úspěšnost klasifikace měkkého a tvrdého objektu z těchto dat vychází 83,3%. Je důležité zmínit několik poznatků. Je problematické určit přesně sílu v koncovém efektoru v místech, které jsou blízko singulárním polohám manipulátoru. Například hodnoty sil, naměřené v blízkosti rámu robotu, jsou celkově vyšší i v klidovém stavu. K vychýlení těchto hodnot je potřeba vyvinout větší sílu. Další poznatek, který zvyšuje nepřesnost klasifikace je v opačných případech. V situaci, kdy robotická ruka sahá téměř na konec pracovního prostoru před robotem, se výrazně posune těžiště a tělo mobilního robotu se nakloní. Při kontaktu ruky s terénem se pak nejdříve zvedá tělo mobilního robotu, což v průběhu sil může vypadat jako měření měkkého předmětu. V grafech na Obrázku 15 jsou zobrazeny velikosti sil při měření výšek polí DEM v pracovním prostoru manipulátoru. Velikost sil ve směru z je v těchto grafech značena velikostí a její barvou značky s tím, že čím světlejší barva, tím větší je síla v daném bodě. V grafech je viditelné, že statické síly jsou výrazně vyšší blíž u rámu robotu, který je umístěn v souřadnicích $x = 0$, $y = 0$, $z = 0$.



(a) Velikosti sil ve směru z v souřadnicích x , y , z při měření kontaktu s měkkým objektem.



(b) Velikosti sil ve směru z v souřadnicích x , y , z při měření kontaktu s tvrdým objektem.

Obrázek 15: Velikosti změřených sil ve směru z na polích DEM z pracovního prostoru manipulátoru.

6.3 Užitečnost algoritmu pro aktivní vnímání

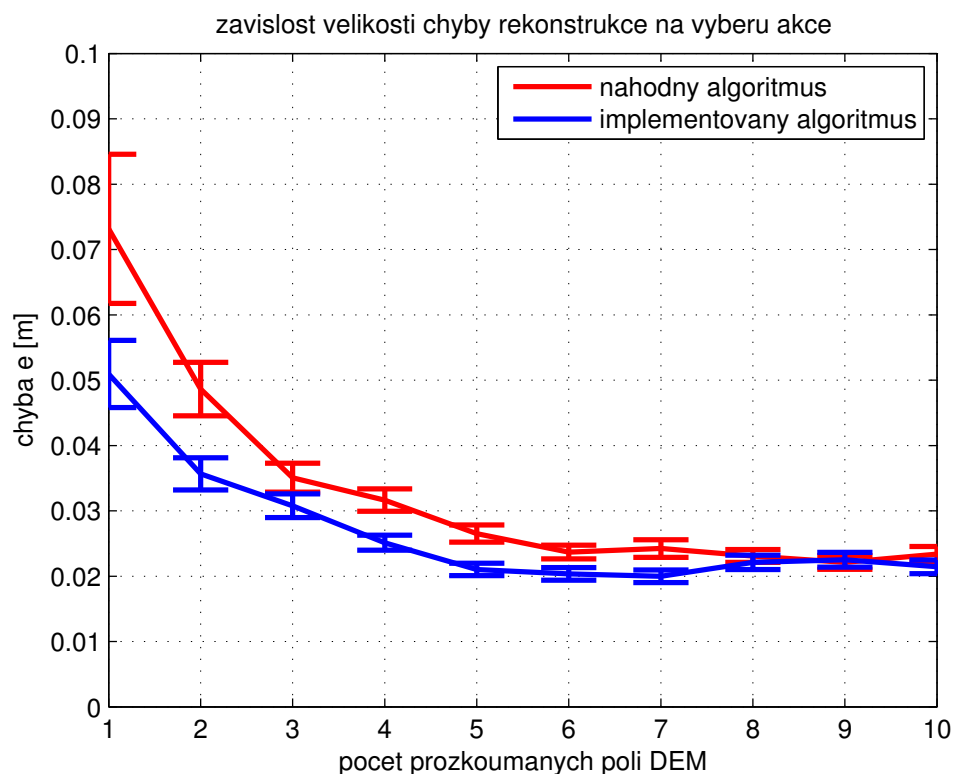
Ve většině případů bývá užitečnost aktivního vnímání porovnávána s příkladem, kdy jsou akce vybírány náhodně. Stejně tak je tomu i v tomto experimentu. Experiment spočívá v určení, zda je výhodné využít v této práci navrženou metodu pro aktivní vnímání (kapitola 5.4.4) místo náhodného generátoru souřadnic z pracovního prostoru manipulátoru.

Pro zobrazení naměřených výsledků bylo potřeba určit chybu rekonstrukce. Pokud by byla měřena chyba rekonstrukce celého DEM, byl by rozdíl chyby při využití implementovaného algoritmu a náhodného výběru souřadnic nepatrný. Protože každé pole DEM ovlivňuje při tvorbě modelu hlavně pole v jeho okolí. A vzhledem k tomu, že pracovní prostor manipulátoru pokrývá méně než jednu čtvrtinu celého DEM, není pro kvalitní rekonstrukci dostatek informací ze zbylých polí DEM. V následujícím experimentu je chyba počítána pouze na polích DEM, která spadají do pracovního prostoru manipulátoru a jsou neprozkoumaná. Jedná se tedy o průměrnou absolutní hodnotu chyby rekonstrukce polí v pracovním prostoru manipulátoru. Chyba e je počítána jako

$$e = \frac{1}{M} \sum_{i \in (W \cap N)} |z_i^{(laser)} - z_i^{(gibbs)}|, \quad (53)$$

kde N značí množinu neznámých polí DEM, W značí množinu polí DEM, které jsou v pracovním prostoru manipulátoru, M značí počet polí, které jsou v průniku množin W a N , $z_i^{(laser)}$ je referenční hodnota výšky pole získaná z laseru a $z_i^{(gibbs)}$ je hodnota výšky pole získaná pomocí Gibbsova vzorkování.

Experiment byl naměřen na dvacítce různých DEM, získaných během přejezdu palety. Při tomto experimentu nebylo využíváno bodů, které byly uloženy do mapy. Data z laseru byla před procesem Gibbsova vzorkování smazána. DEM byl tedy rekonstruován celý pouze ze známých sensorických dat a z prozkoumaných polí v pracovním prostoru. V grafu na Obrázku 16 jsou zobrazeny dvě křivky, ukazující velikost chyby e , v závislosti na počtu prozkoumaných polí z DEM. Z grafu je patrné, že při využití implementovaného algoritmu je chyba rekonstrukce polí v pracovním prostoru menší než při využití náhodného generátoru souřadnic. Rozdíl je viditelný hlavně při nízkém počtu akcí. Vzhledem k tomu, že cena za každou akci je poměrně vysoká, je často používáno méně akcí pro vytváření modelu prostředí. Z tohoto důvodu je užitečné využívat implementovaného algoritmu.



Obrázek 16: Porovnání průměrné chyby v odhadu výšky v poli DEM implementovaného algoritmu a náhodného generátoru

6.4 Rekonstrukce DEM

V této kapitole byly provedeny experimenty, zobrazující funkčnost Gibbsova vzorkování pro modelování části DEM. Měření bylo provedeno v základních situacích, ve kterých se robot může nacházet. Pro každou pozici robotu byla rekonstrukce provedena třikrát a grafy zobrazené níže v této kapitole, zobrazují průměrné hodnoty z těchto tří měření. Cílem bylo rekonstruovat DEM, pouze z proprioceptivních dat. V grafech uvedených v této kapitole je zobrazena pouze část DEM pod robotem a necelá část DEM před robotem. Pro lepší orientaci v grafech je důležité zdůraznit, že robot se nachází na pozicích $i_{DEM} = \langle 5, 10 \rangle$. Algoritmus Gibbsova vzorkování provedl 2500 iterací s tím, že model zobrazený v grafech je střední hodnota za posledních 2400 vzorků. Ve sloupcových grafech jsou zobrazeny úhly *pitch* a *roll* robotu ve stupních a proudy v motorech v ampérech.

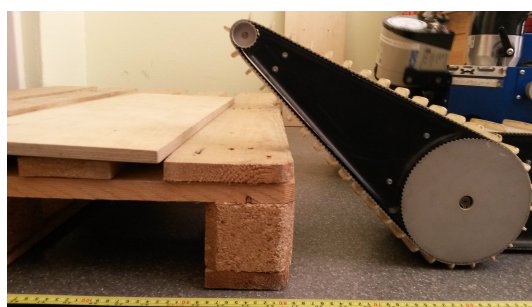
Prvním z této sady experimentů byl pokus o rekonstrukci terénu v situaci, kdy robot je na rovině a na jeho pásy nepůsobí žádné externí síly. Tato situace je zobrazena na Obrázku 17. Výsledkem rekonstrukce je očekávaná rovina. Vzhledem k tomu, že terén před robotem je v tomto případě úplně neznámý, může vypadat jakkoliv. Při Gibbově

6. EXPERIMENTY

vzorkování byl rozptyl vzorků v místě před robotem výrazně vyšší, než v místech pod robotem. Po některých iteracích je očekáváno zvýšení terénu, v jiných iteracích zase schod dolů. Střední hodnota po dostatečném množství vzorků je zobrazena na Obrázku 18.

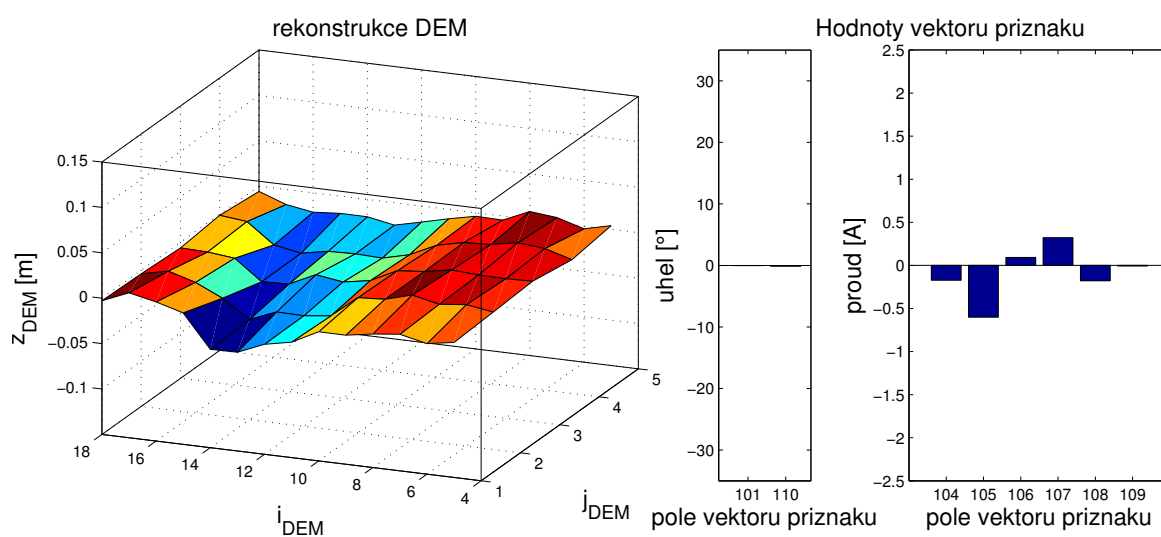


(a) Pohled z vrchu.



(b) Pohled ze strany, kde je vidět, že přední ramena se nedotýkají palety.

Obrázek 17: Situace, při které byl naměřen vektor příznaků, který je zobrazený na Obrázku 18. Robot je na rovině. Hodnoty proudů i naklonění jsou nepatrné.



Obrázek 18: Rekonstrukce DEM a známe hodnoty vektoru příznaků v situaci zobrazené na Obrázku 17.

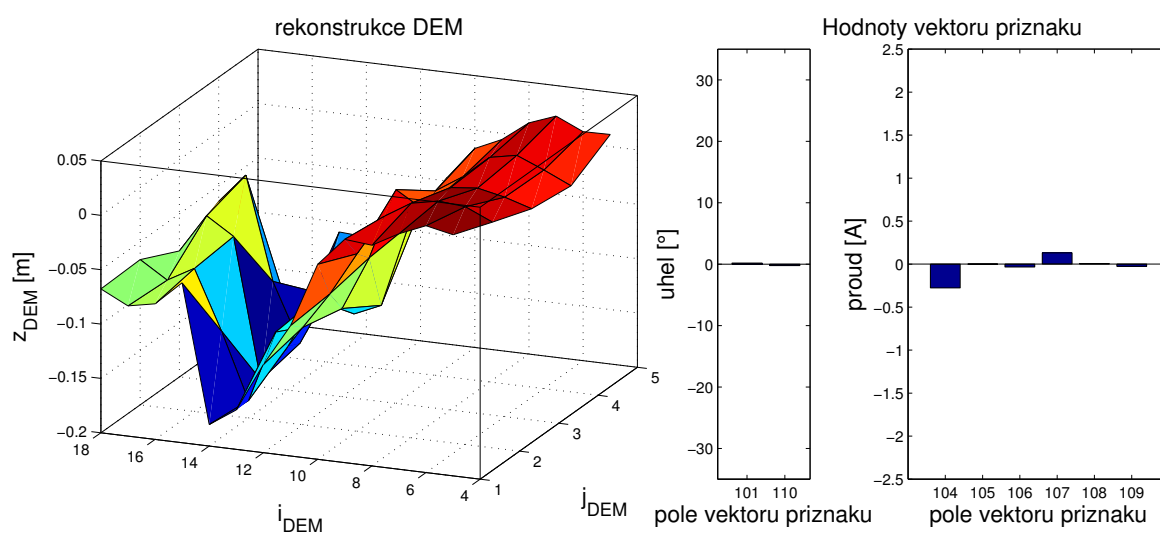
Druhý experiment, který byl změřen, představoval případ, kdy jsou senzorická data v robotu (naklonění a proudy) téměř stejná s předchozím případem. Rozdíl oproti předchozímu experimentu je takový, že v tomto experimentu robot provedl průzkum jednoho pole DEM. Situace je zobrazena na Obrázku 19. Proces Gibbsova vzorkování byl stejný jako v předchozím případě. Jediná změna je, že v při tomto experimentu bylo vzorkováno

6. EXPERIMENTY

99 hodnot vektoru příznaků. Známa byla sensorická část vektoru příznaků a jedno pole DEM. Výsledný rekonstruovaný model, je zobrazen na Obrázku 20.



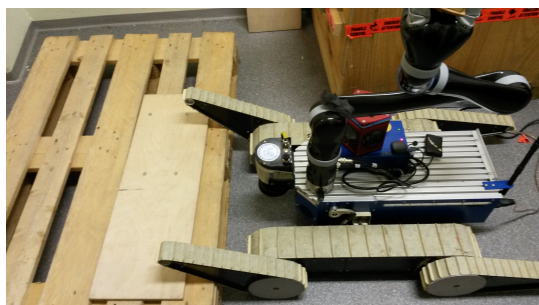
Obrázek 19: Situace, při které byl naměřen vektor příznaků, který je zobrazený na Obrázku 20. Před robotem je schod dolů, který je prozkoumán pomocí jedné akce manipulátoru.



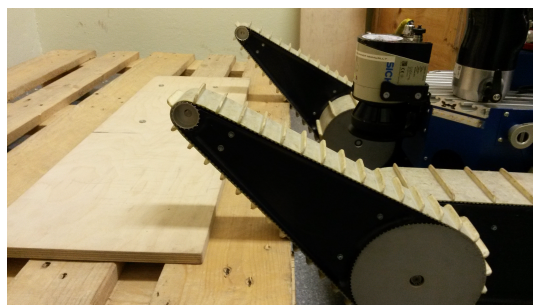
Obrázek 20: Rekonstrukce DEM a známe hodnoty vektoru příznaků v situaci zobrazené na Obrázku 19.

Třetí situace je na první pohled velmi podobná první. Robot je opět na rovině, rozdíl ovšem je v tom, že nyní se předními rameny opírá o paletu před sebou. Tento experiment má ukázat, že je při tvorbě modelu využívána informace o proudech v ramenech mobilního robotu. Situace je zobrazena na Obrázku 21. Ve sloupcových grafech na Obrázku 22 je vidět, že hodnoty proudů jsou již výraznější než v klidovém stavu. Hodnoty vektoru příznaků v polích 106 a 109 značí proudy v předních ramenech. Proudů v hlavních motorech jsou na 105. a 108. pozici. Vzorkování je v tomto případě prováděno pro všech 100 polí vektoru

6. EXPERIMENTY

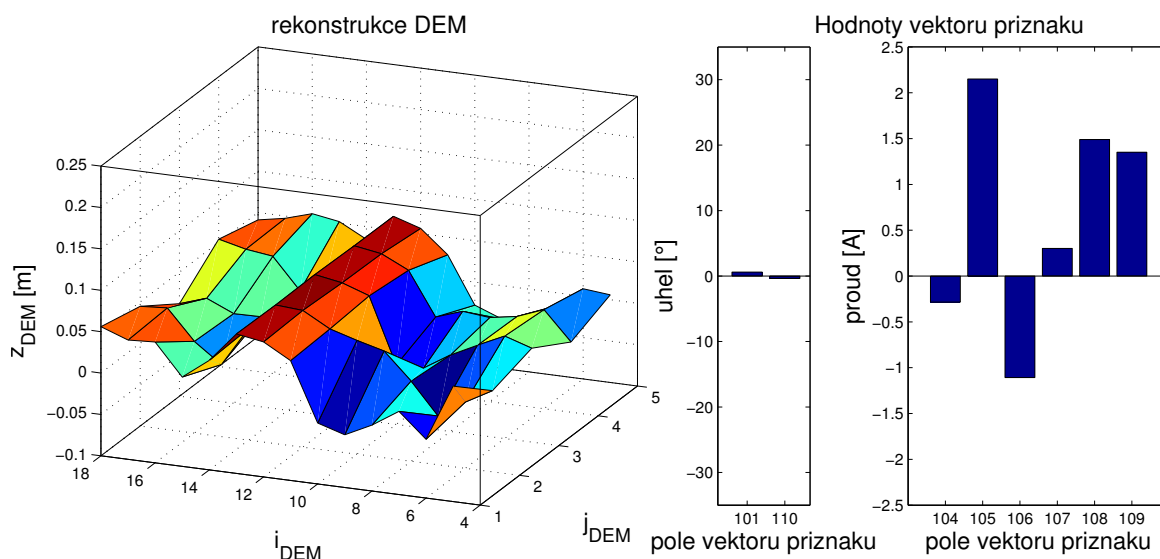


(a) Pohled z vrchu.



(b) Pohled ze strany, kde je vidět že přední ramena se dotýkají palety.

Obrázek 21: Situace, při které byl naměřen vektor příznaků, který je zobrazený na Obrázku 22. Robot je na rovině a dotýká se předními rameny schodu.



Obrázek 22: Rekonstrukce DEM a známe hodnoty vektoru příznaků v situaci zobrazené na Obrázku 21.

příznaků, které odpovídají polím DEM. Známé hodnoty jsou pouze sensorické hodnoty z vektoru příznaků. Rekonstrukce DEM je zobrazena v grafu na Obrázku 22. Z grafu je patrné, že v místě $i_{DEM} = 12$, je očekáván vyvýšený terén. Tento jev je způsoben, právě zvýšenými hodnotami proudů v předních ramenech v sensorické části vektoru příznaků.

Předchozí experiment byl rozšířen tak, že robot navíc vykoná jednu akci pro prozkoumání pole v DEM. Situace je zobrazena na Obrázku 23. Vzorkováno bylo 99 hodnot z vektoru příznaků. V grafu na Obrázku 23 je vidět výsledná rekonstrukce povrchu. Při porovnání modelů rekonstrukce DEM z tohoto experimentu a z experimentu předchozího, můžeme

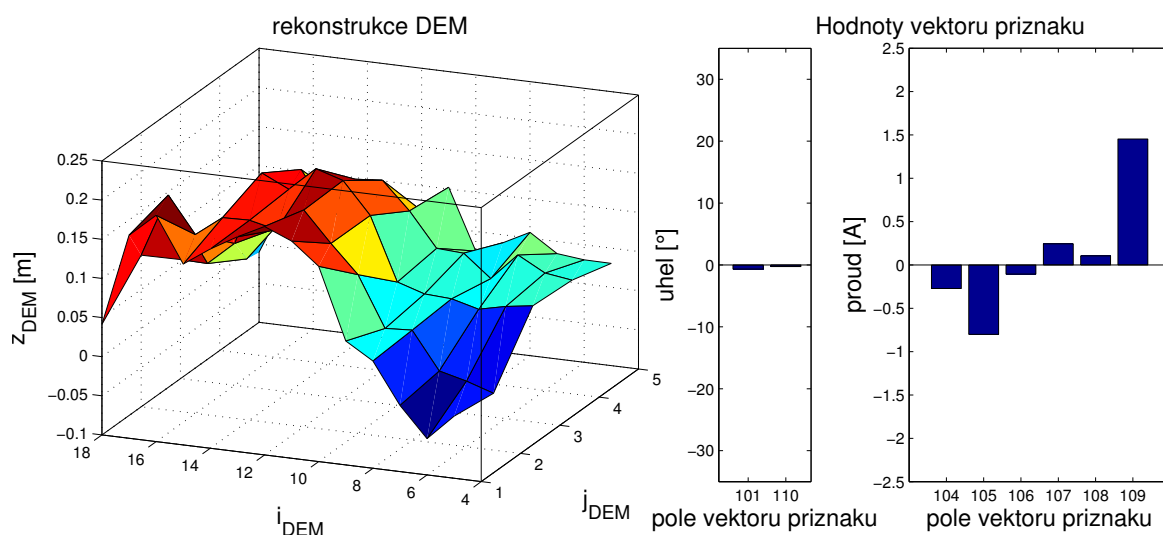
6. EXPERIMENTY



(a) Pohled z vrchu.

(b) Pohled ze strany.

Obrázek 23: Situace, při které byl naměřen vektor příznaků, který je zobrazený na Obrázku 24. Robot je na rovině, jeho přední ramena se opírají o schod a je prováděna haptická explorace jednoho pole.



Obrázek 24: Rekonstrukce DEM a známe hodnoty vektoru příznaků v situaci zobrazené na Obrázku 23.

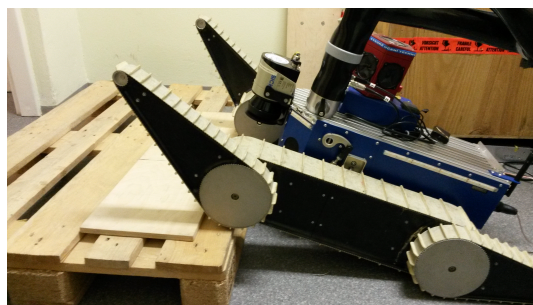
vidět, jak výsledek ovlivnila jedna přidaná informace o prostředí. Zatímco v předchozím případě bylo modelováno zvýšení terénu pouze v blízkosti dotyku robotu s paletou, v tomto případě je model vytvořen tak, že vyvýšený terén pokračuje dál. Kromě této informace je také vidět, že výška překážky více odpovídá realitě.

Další provedené experimenty se soustředí na to, jak vypadá model DEM při naklonění robotu. První experiment ověřující vliv naklonění je zobrazen na Obrázku 25. Jedná se o situaci, kdy je úhel naklonění robotu *pitch* zhruba 22° . Ze sloupcového grafu na Obrázku 26 je vidět, že kromě vyššího úhlu *pitch*, jsou vysoké také hodnoty proudů v zadních ramenech na polích 104 a 107 ve vektoru příznaků. Výsledný model získaný pomocí Gibbsova

6. EXPERIMENTY

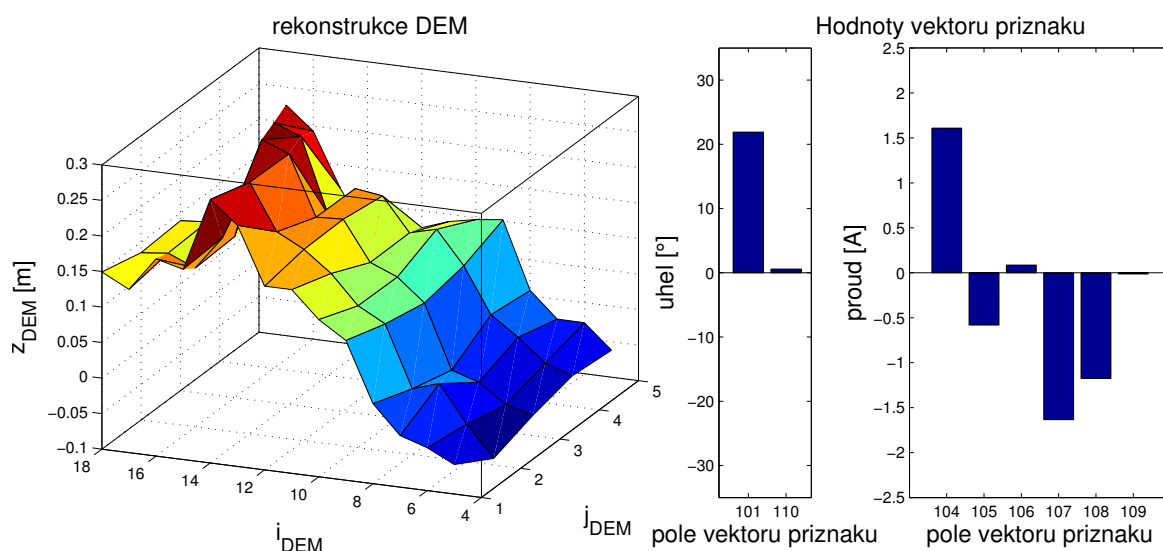


(a) Pohled z vrchu.



(b) Pohled ze strany.

Obrázek 25: Situace, zobrazující nájezd na paletu, bez haptické explorace.

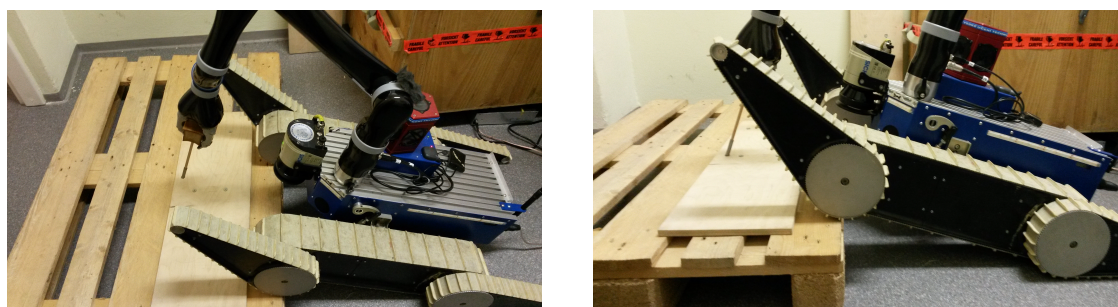


Obrázek 26: Rekonstrukce DEM a známe hodnoty vektoru příznaků v situaci zobrazené na Obrázku 25.

vzorkování očekává nakloněný terén do kopce. V modelu je také vidět, že nejde o pouhou nakloněnou rovinu, ale je pravděpodobné, že pod robotem je schod (mezi místem $i_{DEM} = 7$ a $i_{DEM} = 8$ je patrná změna výšky). Tento jev je projevem toho, že při měření sady dat, ze které je počítána Θ , byly naměřeny i přesně takové situace.

Stejně tak jako tomu bylo v situaci zobrazené na Obrázku 23, kdy bylo testováno, jak ovlivní výsledný model DEM, využití aktivního vnímání. Při stejných podmínkách jako v předchozí situaci bylo provedeno prozkoumání jednoho pole DEM. Tato situace je zobrazená na Obrázku 27. Využitím dodatečné informace o jednom poli v DEM byl vytvořen model (na Obrázku 28), který je oproti předchozímu případu bližší realitě. V modelu vytvořeném v tomto případě již není očekáváno další stoupání terénu.

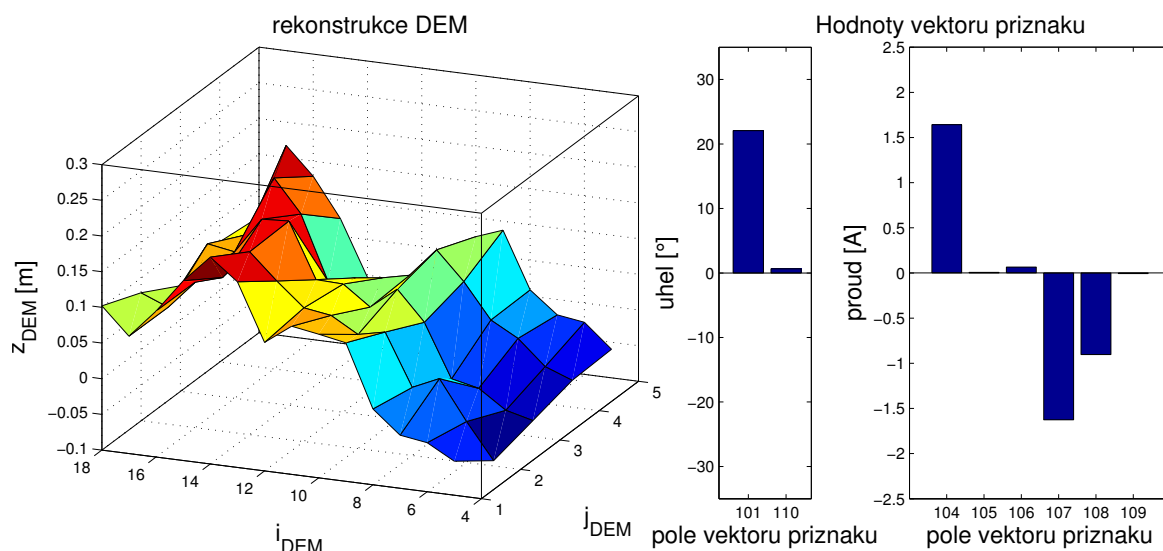
6. EXPERIMENTY



(a) Pohled z vrchu.

(b) Pohled ze strany.

Obrázek 27: Situace, zobrazující nájezd na paletu, s haptickou explorací.



Obrázek 28: Rekonstrukce DEM a známe hodnoty vektoru příznaků v situaci zobrazené na Obrázku 27.

V této kapitole byly provedeny experimenty ukazující jak vypadají modely DEM získané pomocí Gibbsova vzorkování. U všech experimentů bylo provedeno 2500 iterací, a bylo vzorkováno 99 nebo 100 proměnných, v závislosti na experimentu. Výsledky zobrazují, že i z malého množství známých proměnných je možné získat odhad tvaru DEM, který se podobá skutečnosti. Dále experiment ukazuje, že při použití manipulátoru k prohledání alespoň jednoho pole, se model DEM více přiblíží skutečnému prostředí. Když se podíváme na jednotlivé modely DEM získané pomocí Gibbsova vzorkování, můžeme vidět, že výšky některých polí DEM pod robotem by zasahovaly do těla mobilního robotu. Prakticky je nemožné, aby tyto hodnoty byly vyšší než je podvozek robotu. Tato skutečnost nepříjemně ovlivňuje chybu rekonstrukce DEM.

6.5 Rekonstrukce mapy

Jedním z experimentů, které byly provedeny, je rekonstrukce mapy. Jedná se o experiment, kdy robot pomocí IMU odometrie a modelování blízkého okolí (DEM), tvoří mapu prostředí, ve kterém se pohybuje. Základním problémem proprioceptivního mapování je integrace chyby v odometrii. Při tomto experimentu byl kladen důraz na opravdu přesné měření odometrie tím, že operátor ovládal robot opravdu pomalu. Aby bylo možné změřit chybu rekonstrukce, bylo zvoleno takové prostředí, kde jsou hodnoty výšek jednotlivých míst změřitelné. Tímto prostředím byla rovná podlaha, na které byla položena paleta, o výšce 0,144 m a délce 0,8 m. Paleta byla umístěna ve vzdálenosti 1,3 m od počáteční polohy středu robotu. Prostředí, které bylo mapováno, je zobrazeno na Obrázku 29. Výsledná



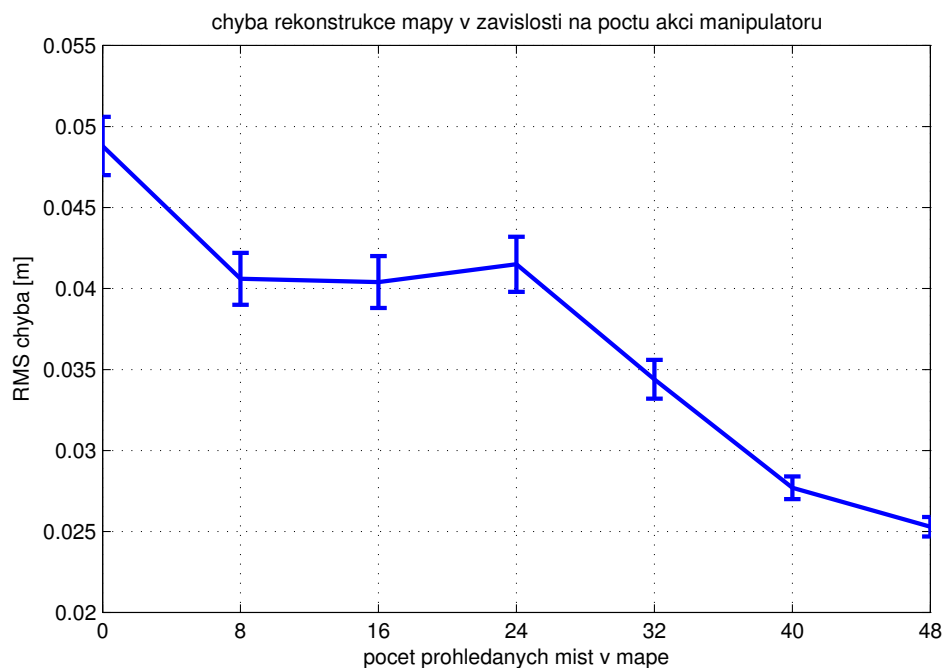
Obrázek 29: Skutečné prostředí, které bylo rekonstruováno.

mapa, na které byla počítána chyba je mapa, která byla zpracována jako finální mapa (funkce *process_map*). Celý experiment probíhal tím způsobem, že robot, ovládaný operátorem, přešel paletu. S tím, že vždy po zhruba 30 cm byl robot zastaven a byl proveden určitý počet akcí manipulátoru. Zastávek robotu bylo osm a počet prozkoumaných míst byl v rozmezí 0 až 6. Chyba rekonstrukce mapy je zobrazena v grafu na Obrázku 30. Na tomto grafu je zobrazeno, jak je závislá chyba rekonstrukce mapy na počtu prohledaných míst s tím, že pro každý počet prohledaných míst byla použita stejná data z odometrie. Chyba zobrazená v grafu je počítána pomocí vzorce

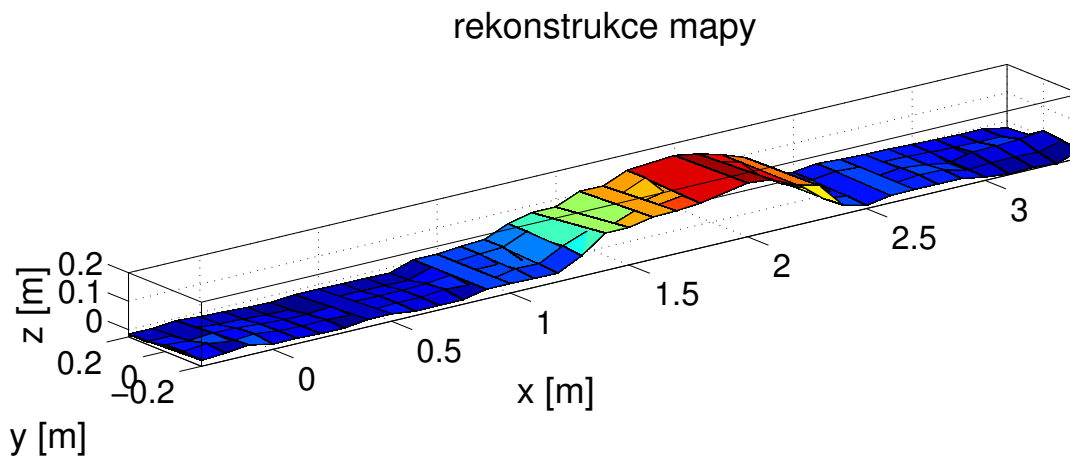
$$e = \sqrt{\frac{1}{M} \sum_{i=1}^M (z_i^{(final_map)} - z_i^{(true)})^2}, \quad (54)$$

kde M je počet bodů uložených do mapy, $z_i^{(final_map)}$ značí hodnoty ve finální mapě a $z_i^{(true)}$ značí přesnou výšku terénu v daném bodě.

6. EXPERIMENTY



Obrázek 30: Chyba rekonstrukce přejetí palety v závislosti na počtu prohledaných polí.



Obrázek 31: Rekonstrukce přejetí palety při osmi prohledaných polích.

V grafu na Obrázku 30 je vidět, že při vyšším počtu prozkoumaných polí chyba rekonstrukce mapy klesá. Výsledná finální mapa získaná pomocí programu pro proprioceptivní mapování je zobrazena na Obrázku 31. Na tomto obrázku je zobrazena mapa, vytvořená použitím pouze jedné akce ruky při každé zastávce robotu. Můžeme vidět, že chyba re-

6. EXPERIMENTY

konstrukce mapy je v tomto experimentu malá a model mapy téměř odpovídá skutečnému prostředí. Je důležité ještě jednou zdůraznit, že tento experiment byl měřen s opravdu velkou opatrností na chybu odometrie. Vzhledem k tomu, že chyba IMU odometrie se integruje, tak i menší chyba na začátku měření výrazně ovlivní výslednou chybu rekonstrukce.

7 Shrnutí realizace a budoucí práce

Během implementace a měření experimentů se objevilo několik komplikací, z nichž se některé nepodařilo úplně vyřešit a budou řešeny v budoucnu. Mezi hlavními problémy je jistě to, že ruka Jaco Arm není primárně určena k práci, ke které byla využívána. Vzhledem k tomu, že kontakt s terénem byl měřen pomocí odběru proudů v motorech a ne využitím taktilních nebo silových senzorů, stávalo se během měření, že se ruka začala přehřívat. V budoucnu je plánováno připevnit taktilní senzory na prsty manipulátoru. Poté by bylo jednodušší a zároveň i přesnější detekovat kontakt s terénem. Kromě toho, by se otevřely nové možnosti využití ruky Jaco Arm v projektu.

Dalším problémem, se kterým se potýkáme je to, že plánovač pohybu vrací při stejných podmínkách různá řešení. V implementovaném programu je kontrolováno, aby se během provádění haptické explorace (při pohybu podél osy z) neměnila konfigurace manipulátoru. Je tedy vyžadováno, aby manipulátor mezi prvním bodem, kde měří sílu a terénem, neměnil konfiguraci. Bohužel se ale občas stane, že neexistuje takový plán, na jehož konci robot dosáhne beze změny konfigurace na bod, který je 1 cm pod ním. V tomto případě není haptická explorace provedena. Při opětovném spuštění explorace stejných souřadnic se většinou vybere konfigurace, ve které existuje řešení až do konce pracovního prostoru, tudíž se nalezne kontakt s terénem. Tento problém neovlivňuje kvalitu modelované mapy, ale pouze čas, který je potřeba pro splnění úlohy. V budoucí práci je potřebné tyto nedostatky odstranit.

Některé komplikace byly zapříčiněny tím, že se na projektu stále pracuje. Různé části starající se o tvorbu shluků 3D bodů, tvorbu DEM nebo počítání odometrie tudíž byly měněny paralelně s vývojem tohoto programu. Momentálně je největší kámen úrazu v tvorbě DEM. I přesto, že v programu není využíváno přímo dat z laseru, jsou data potřebná. Jsou potřeba zaprvé pro tvorbu sady dat, ze kterých počítáme hodnoty Θ (kapitola 5.4.1) a zadruhé k určení chyby rekonstrukce DEM. Dříve byl celý DEM tvořen ze statického shluku 3D bodů. Tento shluk je tvořen slepením dynamických shluků 3D bodů metodou ICP. Problém nastává v situacích, kdy robot zaznamená dynamickou překážku. Tato překážka se zobrazí v dynamickém shluku, ze kterého se pomocí metody ICP dostane i do statického shluku, kde zůstává uložena. V situacích, kdy před robotem například projde člověk, se do statického shluku zanesou body odpovídající dřívější poloze člověka. Při tvorbě DEM, je pak výška v místech, kde člověk dříve stál, ovlivněna těmito body. Z důvodu zlepšení této situace byla předělána tvorba DEM, tak jak je to popsáno v kapitole 4.6. Část DEM před robotem je vytvářena přímo z dynamického shluku 3D bodů. Tímto způsobem se sice vyřešil problém s dynamickými překážkami před robotem, problém ale stále přetrvává v místech za robotem a pod robotem. Problémem v posledních měsících bylo také to, že laser občas zaznamenal přední ramena jako překážku a přidal jí do dynamického shluku, čímž se do-

stala i do statického shluku a později se zobrazila za robotem. Hlavně z tohoto důvodu jsou pole DEM za robotem při ukládání do mapy zahazována. Po uložení do mapy by se model prostředí zhoršil. Program pro tvoření DEM je momentálně upravován a po jeho předělání, bude nutné naměřit nová data pro spočítání podmíněných pravděpodobností. V souvislosti s tímto problémem, jsou všechny experimenty prováděny v konfiguraci *L-shape* zobrazené na Obrázku 3. Pro tuto konfiguraci bylo totiž naměřeno nejvíce vyhovujících dat.

V experimentu v kapitole 6.4 bylo zjištěno, že hodnoty polí, které jsou pod robotem, jsou v některých případech tak vysoké, že by zasahovaly do podvozku mobilního robotu. Tento problém bude potřeba vyřešit. Nabízí se několik způsobů řešení tohoto problému. Jedním ze způsobů je snížení nebo zvýšení celého DEM tak, aby seděl na podvozku. Druhou možností je seříznutí výšek polí tak, jako je tomu při tvorbě DEM ze statického shluku 3D bodů. A třetí nabízenou možností je, že při vzorkování, nebude povoleno vybrat vzorek s hodnotou vyšší než je výška podvozku. Tento problém nebyl vyřešen v rámci této práce a jeho řešení bude v budoucí práci zkoumáno.

Dále je v budoucích pracích plánováno rozšířit stávající program, zajišťující proprioceptivní mapování o lokalizaci. Jednalo by se o proprioceptivní SLAM (simultánní lokalizace a mapování). Momentálně je program velmi háklivý na nepřesnosti v IMU odometrii. V budoucnu bychom chtěli chyby v odometrii korigovat. Pokud máme dva po sobě jdoucí modely DEM a známe přibližný posun robotu mezi těmito DEM, mohli bychom například pomocí metody ICP posunout nový DEM tak, aby co nejvíce odpovídal předchozímu a tím korigovat chybu v odometrii. Během implementace již bylo provedeno několik pokusů o opravení odometrie. Bohužel se zatím nepodařilo získat takové výsledky, které by globálně prokázaly zlepšení odhadu pozice.

8 Závěr

Cílem této diplomové práce bylo navrhnout a implementovat program zajišťující kontaktní průzkum terénu pro mobilní robot. Pro splnění tohoto cíle bylo potřeba vyřešit několik úkolů. Kontaktní průzkum byl zprostředkováván pomocí robotického manipulátoru Jaco Arm. Pro vykonání kontaktního průzkumu bylo nutné implementovat inverzní kinematickou úlohu pro manipulátor v prostředí s překážkami. Pro řešení inverzní kinematické úlohy bylo využito knihovny "MoveIt!" z prostředí ROS. Další důležitou součástí práce byla implementace programu, který pomocí ruky zjistí výšku terénu v daných souřadnicích v souřadném systému robota. Ke zjištění kontaktu robotického manipulátoru s terénem bylo využito kartézské síly v koncovém efektoru manipulátoru, která je počítána přímo z proudových odběrů v jednotlivých kloubech. Tuhost terénu je odhadována pomocí průběhů sil během měření. Další řešenou úlohou byl návrh a implementace algoritmu pro aktivní vnímání, které je využito k výběru souřadnic, prozkoumávaných v rámci kontaktního průzkumu terénu. V rámci této práce byla provedena studie obecných principů aktivního vnímání, které bylo využito pro návrh algoritmu. Akce je v navrženém algoritmu vybírána na základě očekávané užitečnosti. V rámci implementace programu pro kontaktní průzkum terénu byl implementován algoritmus Gibbsova vzorkování. Tento algoritmus vytváří odhad tvaru terénu v okolí robotu na základě měření naklonění robotu, proudových odběrů v jednotlivých pásech a známých měření výšek terénu. Jednotlivé odhady tvarů terénu v okolí robotu jsou ukládány do primární mapy, ze které je později získán finální odhad mapy.

Celý program pro tvorbu mapy i jeho jednotlivé části byly experimentálně ověřeny a jednotlivé experimenty náležitě popsány. Výsledky získané pomocí provedených experimentů jsou dobré. Program pro zjištění výšky terénu v daných souřadnicích v souřadném systému robotu vykazuje přesnost určení do jednoho centimetru a úspěšnost klasifikace tuhosti terénu je vyšší než 80 %. Bylo provedeno měření ukazující, že navržený algoritmus pro aktivní vnímání je lepší než generátor náhodných míst pro prozkoumání. V praktické části práce byly dále ukázány odhady modelů prostředí, získané pomocí Gibbsova vzorkování při několika různých situacích. Ze zobrazených grafů je vidět, že modely vytvořené pomocí Gibbsova vzorkování jsou podobné reálnému tvaru terénu v okolí robotu. Poslední experiment ukazuje proprioceptivní tvorbu mapy. Střední kvadratická chyba výšky pole rekonstrukce mapy byla v tomto experimentu 4 cm při osmi prohledaných místech, což považuji za dobrý výsledek.

Reference

- [1] <http://www.nifti.eu/>. online, přístupné 17-03-2015.
- [2] K. Zimmermann, P. Zuzanek, M. Reinstein, and V. Hlavac. Adaptive traversability of unknown complex terrain with obstacles for mobile robots. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 5177–5182, May 2014.
- [3] <http://kinovarobotics.com/>. online, přístupné 17-03-2015.
- [4] Ioan A. Sucas and Sachin Chitta. “MoveIt!”. <http://moveit.ros.org/>. online, přístupné 17-03-2015.
- [5] <http://www.tradr-project.eu/>. online, přístupné 17-03-2015.
- [6] Ivana Kruijff-Korbayová, Francis Colas, Mario Gianni, Fiora Pirri, Joachim de Greeff, Koen Hindriks, Mark Neerincx, Petter Ögren, Tomáš Svoboda, and Rainer Worst. TRADR project: Long-term human-robot teaming for robot assisted disaster response. *Künstliche Intelligenz*, 2015.
- [7] M. Bjorkman, Y. Bekiroglu, V. Hogman, and D. Kragic. Enhancing visual perception of shape through tactile glances. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 3180–3186, Nov 2013.
- [8] Andreas Doumanoglou, Tae-Kyun Kim, Xiaowei Zhao, and Sotiris Malassiotis. Active random forests: An application to autonomous unfolding of clothes. In *Computer Vision – ECCV 2014*, volume 8693 of *Lecture Notes in Computer Science*, pages 644–658. Springer International Publishing, 2014.
- [9] Ricardo Martins, João Filipe Ferreira, and Jorge Dias. Touch attention bayesian models for robotic active haptic exploration of heterogeneous surfaces. *CoRR*, abs/1409.6226, 2014.
- [10] G.C.H.E. de Croon, I.G. Sprinkhuizen-Kuyper, and E.O. Postma. Comparing active vision models. *Image and Vision Computing*, 27(4):374 – 384, 2009.
- [11] A. Bierbaum, M. Rambow, T. Asfour, and R. Dillmann. A potential field approach to dexterous tactile exploration of unknown objects. In *Humanoid Robots, 2008. Humanoids 2008. 8th IEEE-RAS International Conference on*, pages 360–366, Dec 2008.

REFERENCE

- [12] A. Schneider, J. Sturm, C. Stachniss, M. Reisert, H. Burkhardt, and W. Burgard. Object identification with tactile sensors using bag-of-features. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 243–248, Oct 2009.
- [13] Bruno Siciliano and Oussama Khatib. *Springer Handbook of Robotics*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.
- [14] T. Krajník, J. Faigl, M. Vonásek, V. Kulich, K. Košnar, and L. Přeučil. Simple yet stable bearing-only navigation. *J. Field Robot.*, 2010.
- [15] Morgan Quigley, Ken Conley, Brian P. Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. ROS: an open-source Robot Operating System. In *ICRA Workshop on Open Source Software*, 2009.
- [16] <http://octomap.github.io/>. online, přístupné 27-04-2015.
- [17] I.A. Sucas, M. Moll, and L.E. Kavraki. The open motion planning library. *Robotics Automation Magazine, IEEE*, 19(4):72–82, Dec 2012.
- [18] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
- [19] D.J.C. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003.
- [20] K.S. Roberts. Robot active touch exploration: constraints and strategies. In *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, pages 980–985 vol.2, May 1990.

Příloha

Obsah příloženého CD

V tabulce 7 jsou uvedena jména všech kořenových adresářů příloženého CD s popisem obsahu.

Jméno adresáře	Popis obsahu
dp	diplomová práce ve formátu pdf
sources	zdrojové kódy
images	fotky robotu
video	videa

Tabulka 7: Obsah CD