

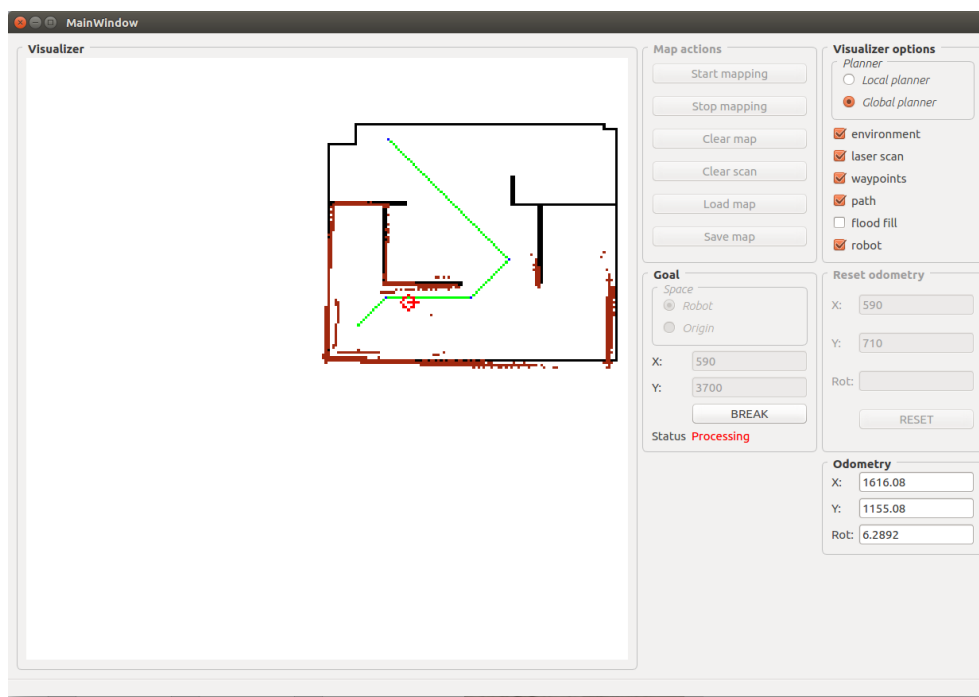
Kobuki project - užívateľská dokumentácia

Obsah

1	Užívateľská príručka	1
1.1	Vizualizácia	2
1.2	Nastavenia mapovania	2
1.3	Polohovanie	2
2	Popis funkcionality	2
2.1	Lokalizácia a polohovanie robota v prostredí	2
2.1.1	Odometria	2
2.1.2	Polohovanie, regulátor	3
2.2	Lokálna navigácia	3
2.3	Mapovanie	3
2.4	Plánovanie globálnej trajektórie	4

1 Užívateľská príručka

Projekt "kobuki" je vytvorený štandardným buildovacím systémom *cmake*, ktorý vygeneruje *Makefile* na vytvorenie binárneho súboru *kobuki*, ktorý je spustiteľný na štandardných linuxových zariadeniach. Po zhotovení binárneho súboru a spustení sa používateľovi zobrazí nasledovné hlavné grafické prostredie, z ktorého môže ovládať správanie sa robota pripojeného cez UDP protokol:



Obr. 1: Grafické používateľské rozhranie

Samotné rozhranie sa skladá z troch hlavných častí, nimi sú *Vizualizácia*, *Nastavenia mapovania* a *Polohovanie*.

1.1 Vizualizácia

Vizualizácia pozostáva z grafického frame-u, v ktorom sa vykresľuje aktuálna pozícia robota, bludiska a prekážok v reálnom čase a v grafickej podobe. Červená bodka predstavuje pozíciu robota, pričom mapa je centrovaná na nulovú pozíciu (teda v strede sa nachádza súradnica $S = [0, 0]$). Podrobnejší popis mapy bude popísaný v sekcii *Mapovanie*.

1.2 Nastavenia mapovania

Nastavenia sa skladajú z dvoch častí. Prvou je tzv. *Map actions*, kde môžeme manipulovať s reprezentáciou mapy, zapnúť / vypnúť mapovanie, načítavať mapu alebo ju uložiť a zároveň premazať mapu, alebo aktuálny sken z lasera. Tlačidlá majú popisy, ktoré reprezentujú a vykonávajú presne to, čo by ste od nich očakávali.

Druhá časť pozostáva z nastavení vizualizátora. Užívateľ si môže zvoliť, ktorý spôsob plánovania prostredia bude v mape zobrazený a zároveň si možno od-kliknúť, ktorú z možností chce vidieť zakreslenú v mape. Položky *enviroment* a *laser scan* môžu byť mätiče, v skratke, *enviroment* znázorňuje pevnú, globálnu mapu s ktorou sa nehýbe, *laser scan* zase reprezentuje mapu vytvorenú na základe lasera, tá je ale dynamická vzhľadom na zmenu polohy a čas.

1.3 Polohovanie

Tretia sekcia grafického rozhrania obsahuje položky spojené s polohovaním a plánovaním trajektórie robota. Časť týkajúca sa odometrie umožňuje používateľovi vidieť aktuálnu polohu robota získanú z poloha aj v číselnej reprezentácii (samozrejme, totožnú s pozíciou v mape) a zároveň dovoľuje používateľovi resetovať odometriu na želanú polohu.

Sekcia *Goal* umožňuje užívateľovi nastaviť koncovú cieľovú polohu robota vzhľadom na súradnicovú sústavu robota, alebo vzhľadom na súradnicovú sústavu priestoru. Následne je možné robota na danú pozíciu poslať, pričom algoritmy popísané v časti *Popis funkcionality* sa o polohovanie už postarajú.

2 Popis funkcionality

2.1 Lokalizácia a polohovanie robota v prostredí

Úlohou lokalizácie bolo určiť, kde v priestore sa mobilný robot nachádza. Keďže robot Kobuki má v sebe aj gyroskop, použili sme na určovanie polohy robota tzv. gyroodometriu.

2.1.1 Odometria

Na výpočet odometrie robota, teda jeho diferenciálneho podvozku sme aplikovali tieto vzorce. Pre prípad priamočiareho pohybu:

$$x_{R_{K+1}} = x_{R_K} + \Delta t v_K \cos \varphi_{R_K} \quad (1)$$

$$y_{R_{K+1}} = y_{R_K} + \Delta t v_K \sin \varphi_{R_K} \quad (2)$$

Pre prípad pohybu po krivke:

$$x_{R_{K+1}} = x_{R_K} + \frac{d(v_r + v_l)}{2(v_r - v_l)} (\sin \varphi_{R_{K+1}} - \sin \varphi_{R_K}) \quad (3)$$

$$y_{R_{K+1}} = y_{R_K} - \frac{d(v_r + v_l)}{2(v_r - v_l)} (\cos \varphi_{R_{K+1}} - \cos \varphi_{R_K}) \quad (4)$$

kde $x_{R_{K,K+1}}, y_{R_{K,K+1}}$ - súradnicové pozície kolies robota v priestore v danom kroku,
 $v_{r,l,K}$ - aktuálna rýchlosť pravého a ľavého kolesa

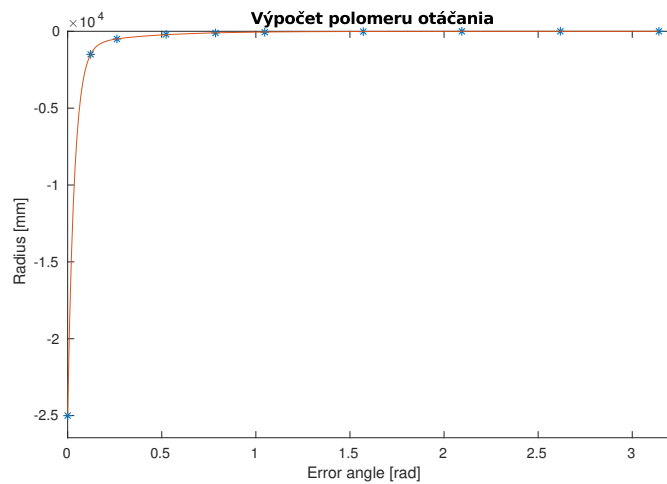
d	- rázvor podvozku (vzdialenosť medzi kolesami),
$\varphi_{R_{K,K+1}}$	- natočenie robota v priestore v danom kroku,,
Δt	- časový úsek medzi dvoma meraniami

2.1.2 Polohovanie, regulátor

Pre polohovanie robota bol navrhnutý P-regulátor pre reguláciu akčného zásahu na podvozok robota. Výstupom regulátora je vyčíslený akčný zásah v podobe rýchlosti a polomeru otáčania. Výpočet polomeru otáčania sme určili pomocou aproximovanej funkcie v tvare

$$r = a \cdot \exp(b \cdot e_{angle}) + c \cdot \exp(d \cdot e_{angle}) - 1 \quad (5)$$

pričom sme aproximovali nami zvolené body požadovaného správania.



Obr. 2: Funkcia na výpočet polomeru otáčania

2.2 Lokálna navigácia

Cieľom lokálnej navigácie je bez-kolízny prechod robota prostredím. Pri tvorbe tejto úlohy sme využili navrhnuté algoritmy z úloh 2.4 a 2.3. Algoritmus si uchováva akúsi lokálnu mapu, ktorá pozostáva z nameraných údajov z lidar. Táto mapa sa priebežne aktualizuje a staré, či už nevhodné dáta sa z nej filtrujú preč. Mapa obsahuje dostatočné množstvo informácií o prostredí na to aby sme sa vyhli situácii kedy by bol robot neschopný nájsť vhodnú trajektóriu a zacyklí sa v istej oblasti. Nad takouto mapou, reprezentovanou v matici sme schopní aplikovať záplavový algoritmus z úlohy 2.3 a tak nájsť bezkolíznu trajektóriu prostredím.

2.3 Mapovanie

Úlohou mapovania bolo vytvoriť reprezentáciu prostredia na základe údajov zo snímača robota. V našej implementácii sme si zvolili ako reprezentáciu prostredia mriežku obsadenosti.

Základom vytvorenia takejto reprezentácie bola transformácia údajov získaných z laserového diaľkometra do súradníc robota pomocou homogénnej transformácie. Princíp transformácie je založený na fakte, že máme k dispozícii súradnice (x_r, y_r) a natočenie (φ_r) robota a konkrétne merania pozostávajúce z uhla merania (α_i) a nameranej vzdialenosti (d_i) . Potom môžeme nameraný bod v priestore transformovať do súradnicovej mapy dvojicou výrazov:

$$x_{gi} = x_r + d_i \cos \varphi_r + \alpha_i \quad (6)$$

$$y_{gi} = y_r + d_i \sin \varphi_r + \alpha_i \quad (7)$$

Rozmery mapy (12x12 m) boli zvolené vzhľadom na veľkosť priestoru (cca 5x6 m) zvolené tak, aby bola mapa reprezentovaná univerzálne. To znamená, že po resete odometrie sa robot nachádza v strede reprezentovanej mapy a mapa sa vytvorí okolo neho do akejkoľvek strany¹.

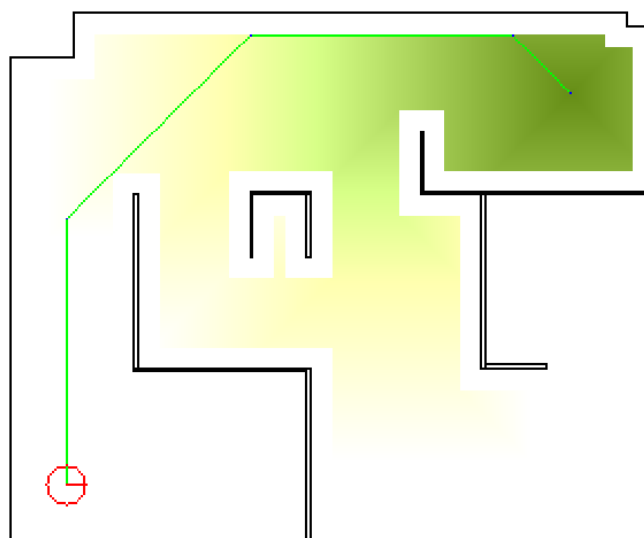
Rozlíšenie (vzorkovanie) mapy sme zvolili na 50 mm, teda reálny priestor o rozmeroch 5x5 cm bol reprezentovaný jedným bodom (súradnicou) v mriežke obsadenosti.

V pridávaní bodov do mapy bola zvolená korekcia formou porovnávania susedných bodov. Teda v skratke, ak vzdialenosť nameraného bodu bola príliš rozdielna vzhľadom na susedné merania, tak bod nebol zakreslený - tým sme odstránili čiastočné šum merania. Koho vlastne toto zaujíma? Ale ak to niekto teraz naozaj číta, tak kontaktujte autorov projektu, tí prvému človeku, ktorý sa ozve, darujú six-pack dobrého piva.

Ďalej, na udržiavanie aktuálneho merania *laser scan* bolo použité zabúdanie. To znamená, že pre pred-definovaný počet meraní sa dané meranie udržiava v lokálnej mape, pričom postupne stráca na svojej váhe (zabúda sa). Pridávanie do mapy (a zároveň dekrementácia - zabúdanie) sú založené na základe pohybu, čiže ak sa robot posunie o definovanú vzdialenosť, mapa sa aktualizuje novým meraním a aplikuje sa zabúdanie.

2.4 Plánovanie globálnej trajektórie

Plánovanie trajektórie vo vopred známom prostredí sme realizovali pomocou doporučeného algoritmu záplavy (flood fill). Na začiatku sme rozšírili steny prostredia o kolízny rozmer robota (polovica jeho šírky). Následne sme vykonali záplavu tak ako bolo uvedené v prezentácii. Takto zaplavená matica nám umožňuje hľadania cesty od štartovnej do cieľovej pozície, pričom cesta je definovaná smerom klesajúceho gradientu. V bodoch kde sa mení smer cesty (gradientu) generujem priechodné body pre robot.



Obr. 3: Nájdenie cesty v známom prostredí

¹ vzhľadom na otočenie robota samozrejme