SLOVENSKÁ TECHNICKÁ UNIVERZITA BRATISLAVA

Fakulta elektrotechniky a informatiky

Riadenie mobilných robotov

Dorottya Szarka a Matej Havetta

2024/2025

Obsah

[Úloha 1. Lokalizácia a polohovanie robota v prostredí 3](#_Toc198422601)

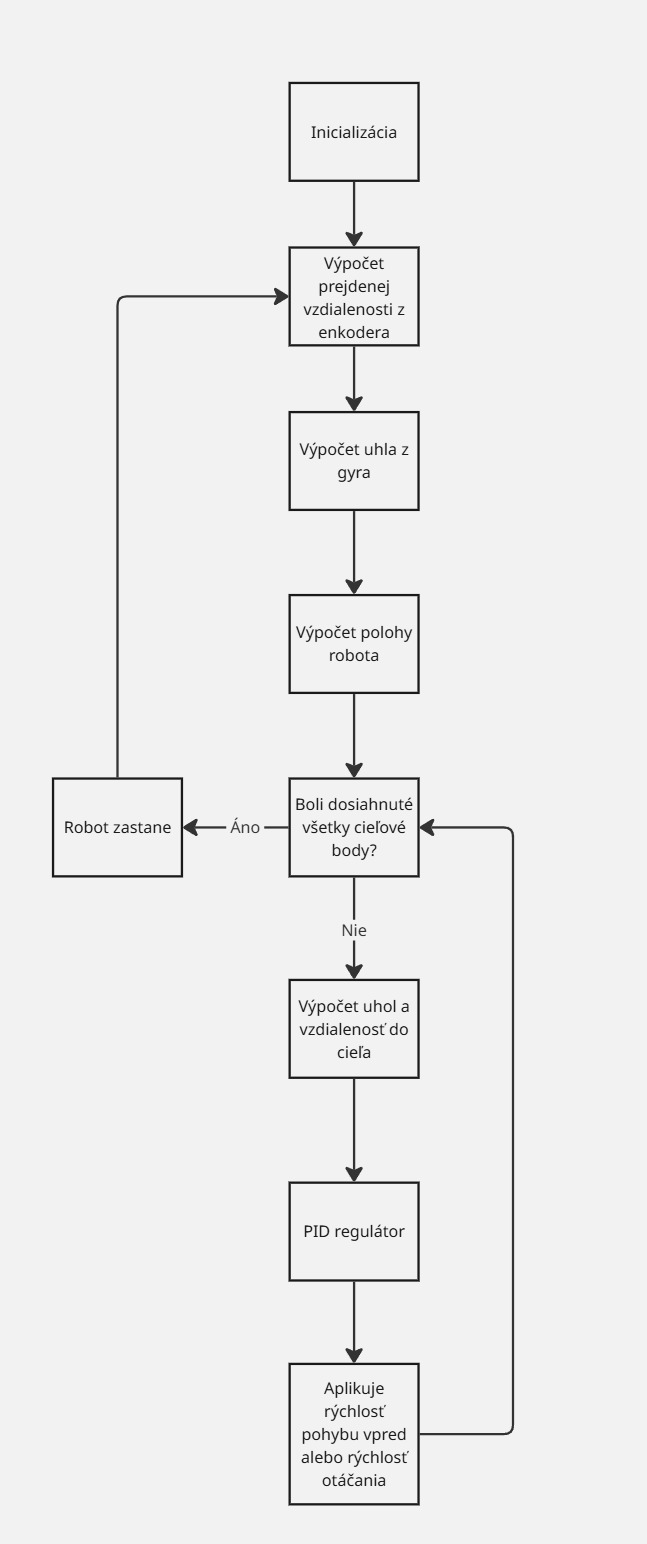
[Odometria + Gyroskop 4](#_Toc198422602)

[Výpočet uhla správneho natočenia 4](#_Toc198422603)

[Výpočet vzdialenosti do cieľa 4](#_Toc198422604)

[Úloha 2. Navigácia 5](#_Toc198422605)

## Úloha 1. Lokalizácia a polohovanie robota v prostredí



Obrázok 1 Vývojový diagram

### Odometria + Gyroskop

Vypočítame prejdenú vzdialenosť každého kolesa samostatne z dát enkódera.

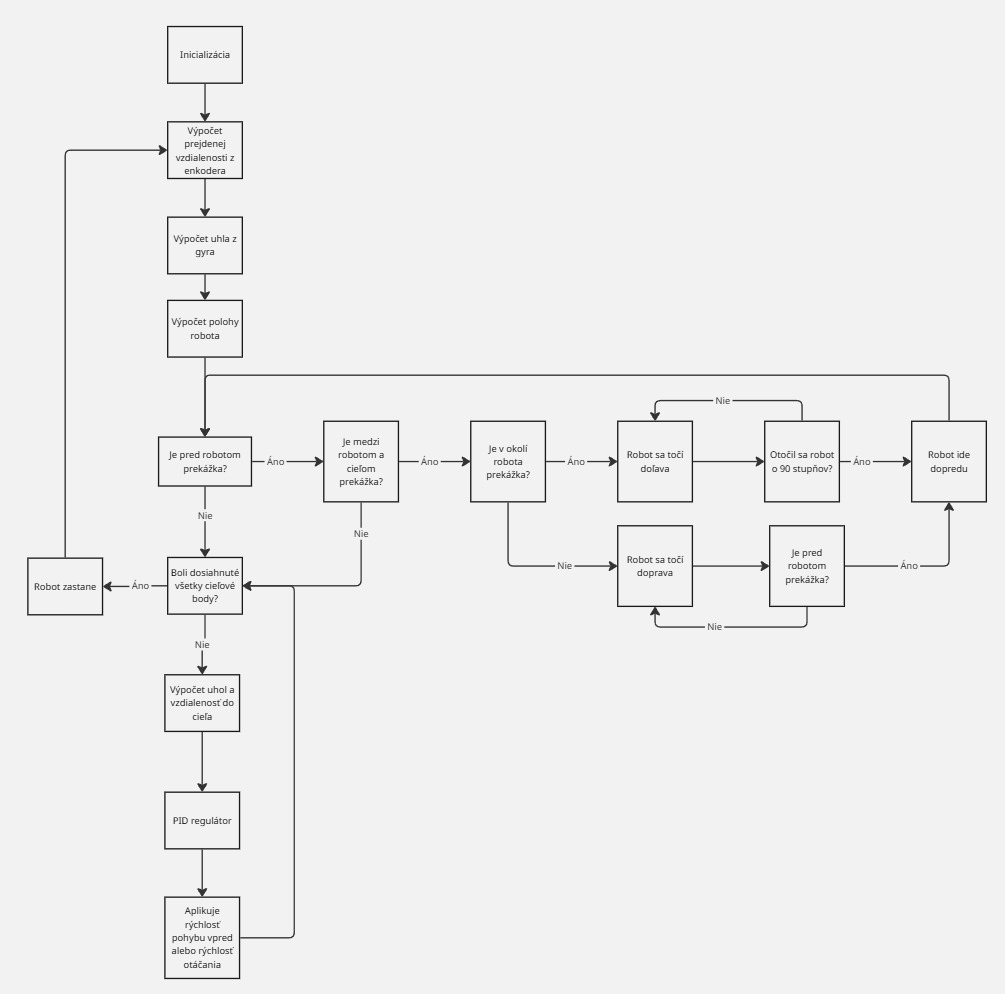
Následne vypočítame prejdenú vzdialenosť od posledného výpočtu.

Vypočítame kde sa s robotom nachádzame v priestore a to aj na základe dát z gyroskopu.

### Výpočet uhla správneho natočenia

### Výpočet vzdialenosti do cieľa

## Úloha 2. Navigácia



Obrázok 2 Vývojový diagram

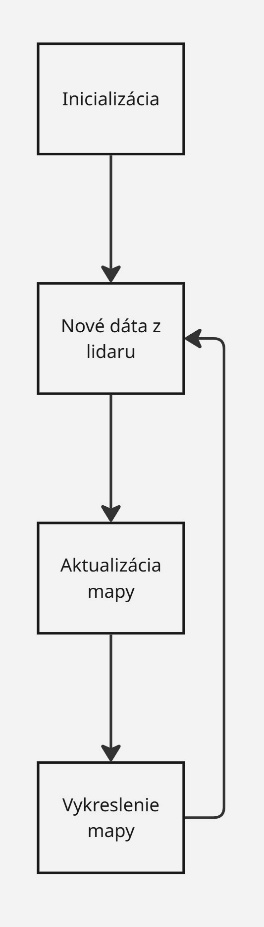
Robot sa snaží dostať do cieľa. V prípade stretu s prekážkou prejde do módu jazdy popri stene. Mód jazdy popri stene sa ukončí keď medzi robotom a cieľom nie je prekážka.

## Úloha 3. Mapovanie

V úlohe 3 vytvárame mapu prostredia na základe dát zo lidaru. Tento senzor poskytuje merania vzdialeností v 360° prehľad toho čo sa nachádza okolo robota. Dáta z lidaru chodia s vyššou frekvenciou, než akou dostávame informácie o polohe a orientácii samotného robota. Z tohto dôvodu sa môže stať, že pri spracovaní konkrétneho merania z lidaru ešte nemáme k dispozícii presnú polohu robota v čase, kedy dané meranie vzniklo.

Aby sme zabezpečili, že lidarové meranie budú správne umiestnené do globálnych súradníc mapy, je potrebné odhadnúť polohu robota v čase, kedy lidar daný bod zaznamenal. Tento odhad robíme pomocou **lineárnej interpolácie** medzi dvoma známymi bodmi (pozíciami a orientáciami robota), ktoré obklopujú časový moment merania. Interpolované hodnoty potom použijeme vo výpočte globálnych súradníc bodu:

Riadenie je rovnaké ako v úlohe 1.



Obrázok 3 „Lifecycle“ mapy

## Úloha 4. Plánovanie dráhy

Výsledkom úlohy 4 je aktualizovaná *cost mapa*, ktorá obsahuje vzdialenosti (v počte krokov) od cieľa do každej dostupnej (voľnej) bunky.

Na začiatku sa reálne súradnice cieľového bodu (goal) prevedú do grid súradníc. Tento prevod sa realizuje pomocou mierky (scale = 0.1) a posunu (offsetX, offsetY), ktoré zabezpečujú správne zarovnanie mapy – teda konverziu z centimetrov do mriežky. Následne sa overí, či cieľový bod leží v rámci platného rozsahu mapy a či nepredstavuje prekážku (napr. stenu).

Po overení platnosti cieľa algoritmus vytvorí novú *cost mapu* – kópiu pôvodnej mapy – do ktorej bude zapisovať výsledky šírenia. Následne inicializuje frontu buniek (FIFO fronta), do ktorej vloží cieľovú bunku ako počiatočný bod šírenia. Cieľovej bunke je priradená hodnota 2, čím sa začína šírenie. Hodnota 2 je zvolená tak, aby sa dala odlíšiť od nevyplnených (nulových) buniek alebo od buniek s prekážkami (-2).

Ak sa počas šírenia narazí na bunku s hodnotou -1 (cieľ), algoritmus sa predčasne ukončí a vráti aktuálny stav *cost mapy*. Ak je cieľ nedostupnú, algoritmus pokračuje dovtedy, kým nie sú spracované všetky dostupné oblasti.

## Github linky k repozitárom s jednotlivými úlohami

Úloha 1.

<https://github.com/Matej-Havetta/RMR-zad1>

Úloha 2.

<https://github.com/Matej-Havetta/RMR-zad1/tree/2_zad>

Úloha 3.

<https://github.com/Matej-Havetta/RMR-zad1/tree/3_mapping>

Úloha 4.

<https://github.com/Matej-Havetta/RMR-zad1/tree/4_trajectory_planning>