Matej Šutý

Report k semestrálnej práci
UIR 2022

Cieľ práce:
Preskúmať oblasť a vytvoriť mapu pomocou Hexapod robota.

Použité metódy:
Reaktívne vyhýbanie sa prekážkam: Robot sleduje najbližšiu prekážku v pravom a ľavom zornom poli a podľa toho, ktorá je bližšie sa rozhodne ísť ďalej od nej. Použil som konštanty z cvičenia a pridal som C_LINEAR_SPEED_BOOST konštantu na zrýchlenie robota ak sa nachádza blízko cieľa aby sa nespomaľoval.

Hľadanie trasy: Použil som A star algoritmu s manhattan distance heuristikou na obstacle growing grid. Do úvahy som bral 8-neighbourhood. Cieľ (centroid) sa nachádza spravidla v oblasti, ktorá je na hranici a teda je v obstacle growing grid nedostupný. Z toho dôvodou som povolil relaxáciu nájdenia cieľa, kde sa stačí dostať do vzdialenosti 1.5 * robot_size od cieľa. Metóda obsahuje fallback. Ak sa nenájde trasa priamo ku cieľu, metóda vráti informáciu o neúspechu a cestu, ktorá sa dostala najbližšie k cieľu.

Zjednodušenie trasy: Pomocou bresenham algoritmu som skracoval cestu a ponechal som iba minimum navigačných bodov pre robota. Cieľ sa nachádza v neprístupnej oblasti, takže som použil podobnú relaxáciu ako v prípade hľadania trasy. Zjednodušenie trasy vypočíta aj dlžku trasy, ktorá sa zohľadňuje pri výbere cieľa v P2.

M1: Mapa statickej veľkosti, ktorá sa určuje priamo v kóde v konštrutore Explorer.

Frontiers sa počítajú iba počas plánovania, preto sa na mape neupdatujú aj keď je nová objavená oblasť.

F1: Identifikácie hranice medzi neobjaveným a objaveným priestorom. Určenie centroidu v tejto oblasti.

F2: Vypočítanie viacerých centroidov na hraničných oblastiach podľa guideline. Filtroval som oblasti, ktoré mali menej ako 4 políčka.

F3: Vypočítanie potencionálneho informačného zisku IG navštívením centroidu v hraničnej oblasti. Vypočítal som to na základe entropie políčok v kruhovom okolí. Polomer kruhu som obmedzil na *laserscan.range_max* / C, napr. C=5 čo zodpovedá približne 1 jednotke vzdialenosti (robot má veľkosť 0.5). V tomto kruhu som z centroidu vypočítal N=100 raycastov na kružnicu. Od stredu do kružnice som pripočítal IG políčok až do momentu, kým som narazil na prekážku. Potom som pokračoval s ďalším raycastom.

P1 *nepoužité*: Vyberal som najbližší centroid hraničnej oblasti po tom ako som k nemu vypočítal trasu a zjednodušil ju. Centroidy, ku ktorým som nenašiel trasu som nebral do

úvahy. Obsahuje fallback - ak sa nenájde žiadna trasa, vyberie sa prvý frontier a najbližší bod, ktorý sa pri hľadaní trasy dostal najbližšie.

P2: Centroidy som zoradil zostupne podľa IG z F3 a hľadal som trasu k najvýhodnejšiemu centroidu. Ak celková dlžka trasy bola kratšia ako CLOSE_RANGE, odmietol som ju a hľadal som ďalšiu. Obsahuje fallback. Ak sa nenašla žiadna trasa, robot dostane pokyn spraviť kružnicu.

Spustenie projektu:
Spustite *Explorer.py* skript v Python3.

Nastavenie parametrov je pomocou argumentov (nižšie) pri spustení. Skript má nastavené default hodnoty, ktoré je vidno nižšie a sú vhodné pre mapu blocks_multirobot.

- rs : Robot size = 0.5
- r: Map resolution = 0.1
- width: Width = 100
- height: Height = 100
- Origin: Origin: = [-5, -5]

Taktiež je možné nastaviť rôzne konštanty v jednotlivých súboroch ale nie je to potrebné. Taktiež možete zmeniť intervaly výpočtov priamo v konštruktore Explorer.

Dodatočné knihovne, ktoré možno treba doinštalovať:
numpy
scipy.ndimage
skimage.measure
matplotlib.pyplot
Sklearn.cluster

Poznámka:
Na mojom pomalom počítači som si pomohol zrýchlením robota cez konštantu C_LINEAR_SPEED_BOOST v HexapodController. Je možné, že na rýchlejšom počítači to nebude potrebné.