

1 Implementacja kodu

Na drugim laboratorium stworzyliśmy dwie mapy w rosie: „korytarz” i „piętro”. Na tych mapach będziemy testować stworzony w ramach projektu drugiego: node do nawigacji robota mobilnego (Tiago), parametry bibliotek do nawigacji, plik launchowego, pliki „świata”, które zawierają informacje o naszych dwóch mapach.

Korzystamy z gotowych bibliotek `costmap_2d`, `global_planner`, `base_local_planner`, `rotate_recovery` i `base_local_planner`, do inicjalizacji działania lokalnego i globalnego planner'a, zachowania „odkleszczania” i lokalnej mapy kosztów.

Nasz węzeł ROS:

- Subskrybuje `/move_base_simple/goal` – otrzymujemy informacje o punkcie docelowym zadane przez nas w Rvizie
- Korzysta z lokalnego i globalnego planner'a, lokalnej mapy kosztów i „odkleszczania”
- Wysyła żądania zmian prędkości do sterownika robota
- Realizuje bezpieczny ruch do kolejnych pozycji
- Zwraca stan wykonania

Pliki:

`proj2.cpp` - https://github.com/STERO-22Z/mobile_STERO-22Z-Kwarcinski-Marchlewicz/blob/tiago/src/proj2.cpp

`tiago_proj2.launch` - https://github.com/STERO-22Z/mobile_STERO-22Z-Kwarcinski-Marchlewicz/blob/tiago/src/proj2.cpp

pliki parametrów - https://github.com/STERO-22Z/mobile_STERO-22Z-Kwarcinski-Marchlewicz/tree/tiago/param

pliki światów - https://github.com/STERO-22Z/mobile_STERO-22Z-Kwarcinski-Marchlewicz/tree/tiago/worlds

konfiguracja RVIZa - https://github.com/STERO-22Z/mobile_STERO-22Z-Kwarcinski-Marchlewicz/tree/tiago/param

2. Rezultaty:

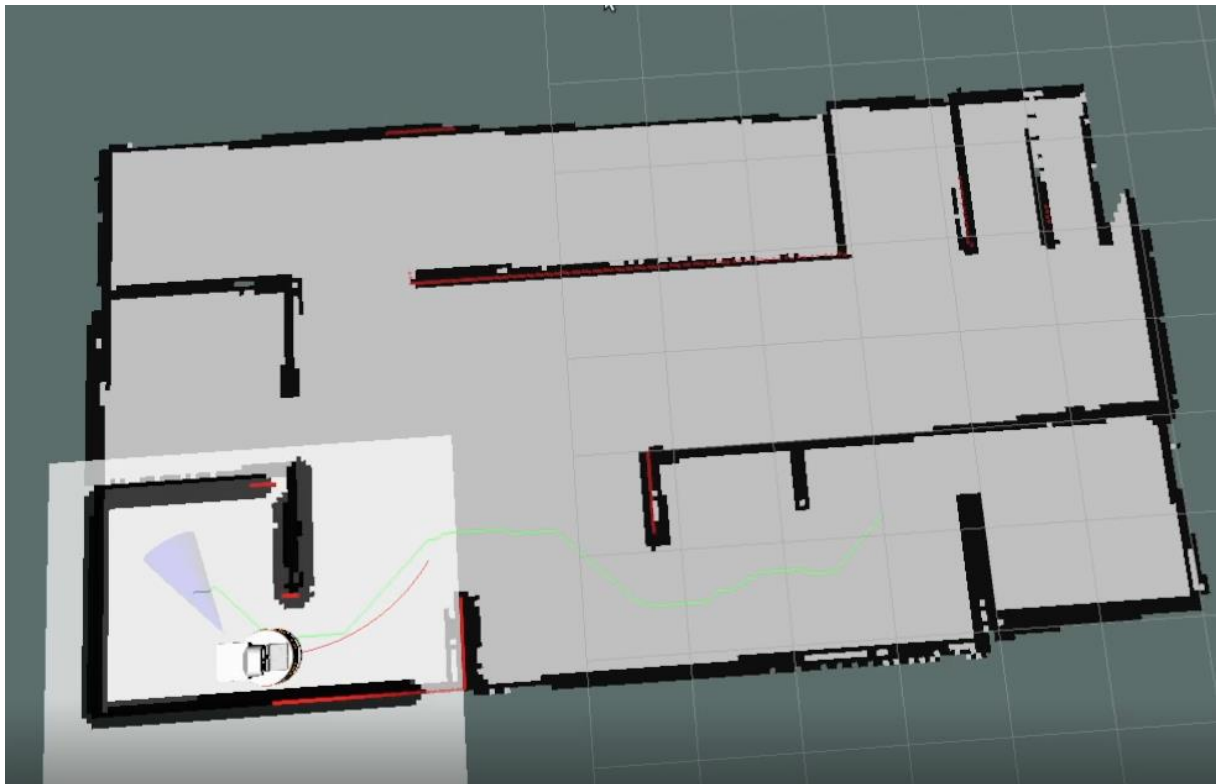
Ostatecznie otrzymaliśmy poprawną symulację. Robot poruszał się poprawnie po mapach. Czasami poruszając się po globalnym planie, najczęściej w miejscach, gdzie algorytm planowania A* generował „ostre” zakręty, robot uruchamiał „RecoveryBehaviors” robił obrót o około 360 stopni i kontynuował jazdę na zadanej trasie. Wjeżdżał poprawnie w wejścia o szerokości do obwodu robota (54 centymetry – z dokumentacji Tiago).

Z tego co wyczytaliśmy na forum rosa błędy widoczne na filmiku ekstrapolacji wynikają z działania na maszynie wirtualnej.

Na widocznym poniżej zrzucie ekranu widać jedną z dwóch map.

Widać na niej:

- Zieloną trajektorie – plan globalny
- Czerwoną trajektorie – plan lokalny
- Wybielony kwadrat z robotem w centrum – obszar „generacji” plany lokalnego, przemieszcza się razem z robotem
- Ciemno-szary kolor obok ścian – działanie warstwy inflacji (żeby robot w miarę możliwości nie jeździł obok ścian)



3. Linki do filmików z symulacji działania robota*:

Działanie w „labiryncie”:

<https://1drv.ms/v/s!AkMYzt8ywVnohmpfC3FBFb1ugR1x?e=iX9s7Z>

Działanie w korytarzu z dostawioną przeszkodą:

<https://1drv.ms/v/s!AkMYzt8ywVnohmlClhwpLHgiNVb0?e=iEplcA>

Dodaliśmy w Gazebo przeszkodę, której nie było w planie globalnym. Jak widać robot ją zauważa i próbuje ominąć. W korytarzu widoczne jest niedostrojenie mapy, lecz nie wpływa to negatywnie na działanie robota – do celu dojeżdża korzystając głównie z mapy lokalnej.

*zaleca się znaczne przyspieszenie, z uwagi na niską liczbę klatkę na maszynie wirtualnej (średnio 1-10 fps) nie przyspieszyliśmy symulacji w programie do obróbki, żeby widz lepiej mógł wczuć się w realia naszej pracy.