

# Roboty Mobilne 1

## Zadanie planowania

### SLAM

## Zadania

1. Zadaniem jest stworzenie symulacji służącej do nawigacji robota, systemu detekcji kolizji oraz mapowania obszaru na podstawie dostępnych czujników.

- (a) Uruchomić symulację robota *waffle* w środowisku *turtlebot3\_dqn\_stage4* lub *turtlebot3\_world* lub innym wybranym.

```
$ export ROS_DOMAIN_ID=xyz
$ export TURTLEBOT3_MODEL=waffle
$ ros2 launch turtlebot3_gazebo turtlebot3_world.launch.py
```

- (b) Wykorzystując symulację i wybrana implementację algorytmu SLAM np. *slam\_toolbox* lub *turtlebot3\_cartographer*

```
$ ros2 launch turtlebot3_cartographer cartographer.launch.py \
  use_sim_time:=True
$ ros2 launch slam_toolbox online_async_launch.py
```

- (c) Wykorzystując wybrany moduł do sterowania należy możliwie dokładnie przeskanować cały dostępny obszar

```
$ ros2 run turtlebot3_teleop teleop_keyboard
$ ros2 run turtlebot3_gazebo turtlebot3_drive
```

- (d) Uzyskane dane należy zapisać do pliku

```
$ ros2 run nav2_map_server map_saver_cli -f ~/Desktop/map
```

- (e) W kolejnym kroku należy wczytać stworzoną mapę. Najbezpieczniej jest wyłączyć wszystkie dotychczas działające procesy i uruchomić od nowa symulację oraz moduł do nawigacji.

```
$ ros2 launch turtlebot3_navigation2 navigation2.launch.py \
  use_sim_time:=True map:=$HOME/Desktop/map.yaml
```

- (f) W pierwszej kolejności należy spozycjonować robota, a następnie można wyznaczać mu punkt i orientację docelową (przyciski *2D Pose Estimate* i *2D Goal Pose* należy przytrzymać i przeciągnąć).

2. Zadanie planowania realizowane jest z wykorzystaniem topika *goal\_pose*. Polecenie może zostać wywołane z parametrami w następujący sposób

```
$ ros2 topic pub --once \
  /goal_pose geometry_msgs/PoseStamped \
  "{header: {stamp: {sec: 0}, frame_id: 'map'}, \
```

```
pose: {position: {x: 2.0, y: 1.0, z: 0.0}, \
orientation: {w: 1.0}}}"
```

Wykorzystując *goal\_pose* zaimplementować ruch po krawędziach wielokąta foremnego. Jako parametr powinny być przekazywane dwie liczby:  $n$ -liczba boków i  $m$ -długość boku. Miara kąta między kolejnymi krawędziami wynosi

$$\theta = \frac{\pi(n-2)}{n}.$$

Pozycje docelowe robota, to współrzędne kolejnych wierzchołków. Robot powinien zakończyć swoją podróż w punkcie początkowym. Każdorazowo robot dojeżdżając do wierzchołka powinien być zorientowany zgodnie z kolejną krawędzią.

Dla prostoty obliczeń należy przyjąć, że punkt  $(0,0)$  jest centroidem, a więc znajduje się w geometrycznym środku wielokąta. Dla tak przyjętych założeń macierz obrotu o pojedynczy kąt przedstawia się następująco

$$R = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}.$$

Idąc dalej, kolejne punkty wyliczane są zgodnie ze wzorem

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}.$$