Politechnika Wrocławska

Lab05 - Implementacja robota 2.0

Roboty Mobilne 1

Michał Markuzel, 275417

15.06.2025

Spis treści

1 Cel ćwiczenia

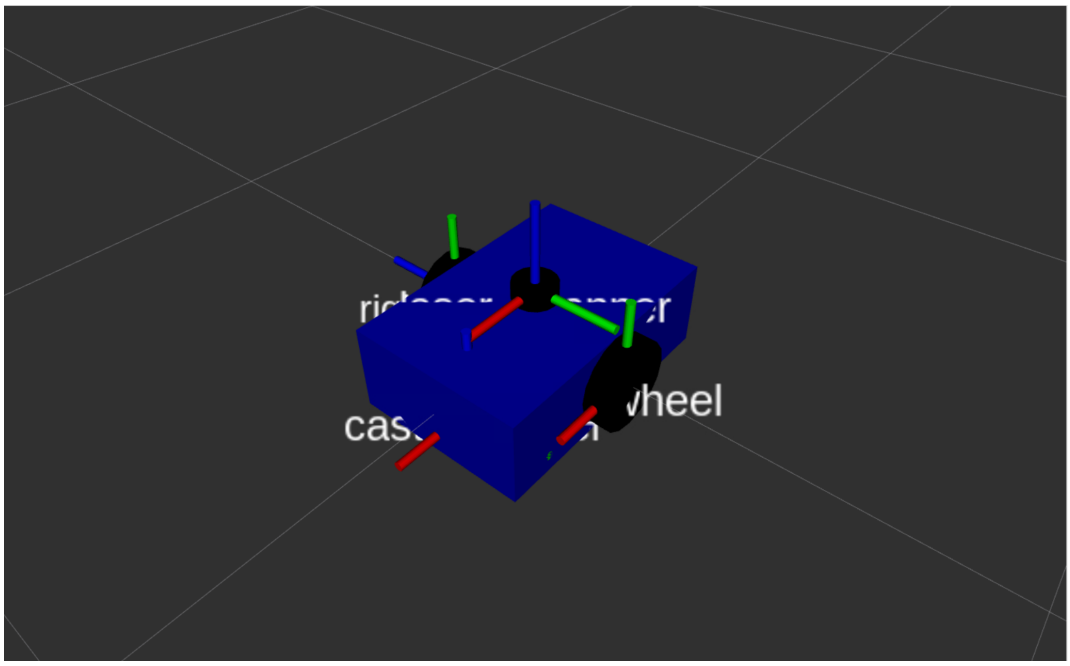
Celem ćwiczenia było zaimplementowanie modelu robota mobilnego klasy 2.0 w środowisku ROS2 z następującymi wymaganiami:

- Podstawa w kształcie sześcianu o wymiarach 0.4m × 0.6m × 0.2m
- Skaner laserowy umieszczony na robocie
- Koła główne o promieniu 0.1m
- Koło podporowe (castor) w postaci sfery o promieniu 0.05m

2 Model robota

2.1 Wizualizacja modelu

Na rysunku 1 przedstawiono model zaimplementowanego robota w środowisku RViz.



Rysunek 1: Model robota w środowisku RViz

Jak widać na rysunku 1, model zawiera wszystkie wymagane elementy:

- Niebieską podstawę o zadanych wymiarach
- Dwa czarne koła główne
- Czarne koło podporowe (castor)
- Czarny cylinder reprezentujący skaner laserowy na górze robota

2.2 Struktura pakietu

Implementacja została zrealizowana w pakiecie ROS2 o nazwie `my_robot_description`. Struktura pakietu przedstawia się następująco:

```
my_robot_description/
          CMakeLists.txt
          config/
                display.rviz
          launch/
                display.launch.py
          package.xml
          urdf/
              robot.urdf.xacro
              robot_core.xacro
              robot_gazebo.xacro
```

2.3 Model robota

Model robota został zdefiniowany w pliku `robot_core.xacro`. Główne elementy modelu to:

2.3.1 Parametry robota

```
<xacro:property name="base_length" value="0.6"/>
<xacro:property name="base_width" value="0.4"/>
<xacro:property name="base_height" value="0.2"/>
<xacro:property name="wheel_radius" value="0.1"/>
<xacro:property name="wheel_width" value="0.05"/>
<xacro:property name="caster_radius" value="0.05"/>
```

2.3.2 Podstawa robota

```
<link name="base_link">
    <visual>
        <geometry>
            <box size="${base_length} ${base_width} ${base_height}"/>
        </geometry>
        <material name="blue"/>
    </visual>
    <!-- ... -->
</link>
```

2.3.3 Koła główne

```
<link name="right_wheel">
    <visual>
        <geometry>
            <cylinder radius="${wheel_radius}"
                      length="${wheel_width}"/>
        </geometry>
        <material name="black"/>
    </visual>
    <!-- ... -->
</link>
```

2.3.4 Koło podporowe (castor)

```
<link name="caster_wheel">
    <visual>
        <geometry>
            <sphere radius="${caster_radius}"/>
        </geometry>
        <material name="black"/>
    </visual>
    <!-- ... -->
</link>
```

2.3.5 Skaner laserowy

```
<link name="laser_scanner">
    <visual>
        <geometry>
            <cylinder radius="0.05" length="0.04"/>
        </geometry>
        <material name="black"/>
    </visual>
    <!-- ... -->
</link>
```

3 Uruchomienie i wizualizacja

Model robota może być uruchomiony i zwizualizowany w środowisku RViz za pomocą przygotowanego pliku launch:

```
ros2 launch my_robot_description display.launch.py
```

Plik launch konfiguruje następujące elementy:

- Robot State Publisher - publikuje transformacje (tf) dla robota
- Joint State Publisher - zarządza stawami robota
- RViz - narzędzie do wizualizacji modelu

4 Wnioski

W ramach ćwiczenia zrealizowano:

- Implementację modelu robota zgodnie z zadanymi wymaganiami
- Konfigurację wizualizacji w środowisku RViz
- Prawidłowe umiejscowienie wszystkich elementów robota
- Poprawne działanie modelu kinematycznego

Model robota został zaimplementowany w sposób modularny, co umożliwia łatwe modyfikacje i rozszerzenia w przyszłości. Wszystkie wymiary i parametry są zdefiniowane jako zmienne, co ułatwia ich dostosowanie do potrzeb.