UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

Wybrane aspekty pojazdów autonomicznych

Oprogramowanie

Rafał Szczepański
e-mail: szczepi (at) umk.pl
www.umk.pl/~szczepi

03.06.2024

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

Plan prezentacji

- Zadania, wymagania, właściwości
- Robot Operating System
- CARLA
- Autoware Foundation
- OpenPilot

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

Oprogramowanie

Podstawowe zadania:

- dostęp do komponentów,
- modelowanie środowiska,
- synteza zachowań,
- zapewnienie narzędzi do debugowania i teletransmisji danych.

Istotne właściwości:

- przejrzysta (modułowa) struktura,
- możliwość wykorzystania komponentów w innych projektach,
- wydajność,
- odporność na błędy.

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

Oprogramowanie

Zaawansowane oprogramowanie przeznaczone do autonomicznych pojazdów mobilnych zwykle bazuje na **strukturach** (ang. framework).

Stosowanie struktur pozwala na przygotowanie aplikacji w sposób modułowy, poprzez łączenie elementów oprogramowania (np. modułów, komponentów, usług) w sieci.

Sterowanie rozproszone – wykorzystuje połączone w sieć komputery / sterowniki mikroprocesorowe.

Sterowanie rozproszone nakłada na sieć komunikacyjną dodatkowe wymagania dotyczące:

- przepustowości,
- opóźnień.

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

Oprogramowanie

Aspekty uwzględniane podczas tworzenia struktur:

- odporność,
- wydajność,
- łatwość obsługi,
- elastyczność,
- rozszerzalność,
- niezawodność,
- skalowalność,
- praca w czasie rzeczywistym,
- interoperacyjność,
- możliwość przenoszenia na inne platformy.

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

Istotne cechy z punktu widzenia programowania robota mobilnego:

- Wątki,
- Wysokopoziomowe języki programowania zorientowane obiektowo,
- Sterowanie niskopoziomowe,
- Łatwość prototypowania.,
- Komunikacja miedzyprocesowa,
- Wydajność,
- Wsparcie społeczności,
- Dostępność zewnętrznych bibliotek.

- Wszystkie cechy posiada oprogramowanie nazwane:

Robot Operating System

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

Robot Operating System (ROS) – podstawowe informacje

- Projekt rozpoczął się na Uniwersytecie Stanforda w 2007 roku,
- Aktualnie jest opracowywany i utrzymywany przez organizację *Open Robotics,*
- Dzięki wbudowanej funkcjonalności pozwala na: wielowątkowość, sterowanie niskopoziomowe, obsługę pakietów, przekazywanie informacji między procesami.
- Obsługuje języki programiwania C/C++ oraz Python,
- W niedalekiej przyszłości obsługiwał będzie również C# oraz Java,
- Integracja z bibliotekami tj. OpenCV,
- Wbudowane implementacja zaawansowanych algorytmów do planowania ścieżki, lokalizacji, mapowania, regulacji itp.,
- Wbudowany symulator Gazebo umożliwiający testowanie opracowywanych algorytmów symulacyjnie,
- Pakiet RViz umożliwiający wizualizację danych ze skanerów laserowych, kamer, mapy, pozycję i orientację robota oraz dane użytkownika.

UNIWERSYTET MIKOŁAJA KOPERNIKA W TORUNIU
Wydział Fizyki, Astronomii i Informatyki Stosowanej

Robot Operating System (ROS)

- zbiór bibliotek oraz narzędzi ułatwiających przygotowanie oprogramowania robota
- podstawowe składniki:
  - sterowniki
  - algorytmy
  - narzędzia programistyczne
- funkcjonalność:
  - utworzenie sieci procesowej robota
  - wsparcie podczas tworzenia oprogramowania robota
  - wsparcie na poziomie testowania robota
  - komunikacja i wymiana materiałów ze społecznością

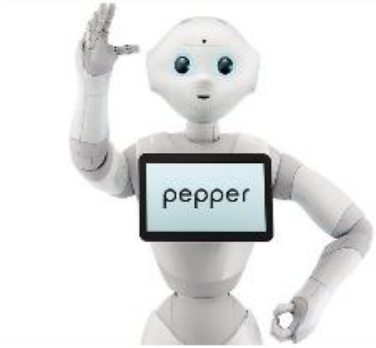
UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

Roboty wspierające ROS:

a) TurtleBot 4 (www.clearpathrobotics.com/turtlebot-4/)
b) Pepper (www.aldebaran.com/en/pepper)
c) Husky (https://clearpathrobotics.com/husky-unmanned-ground-vehicle-robot/)
d) Universal Robot UR3 (https://www.universal-robots.com/cb3/)

a)

c)



b)

d)

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

Czujniki wspierające ROS:

a) TeraRanger (www.terabee.com)
b) Xsense MTi IMU (www.xsens.com/products/)
c) RPLidar A3 (https://www.slamtec.com/en/Lidar/A3)
d) Intel RealSense (https://realsense.intel.com)

a)

c)

b)

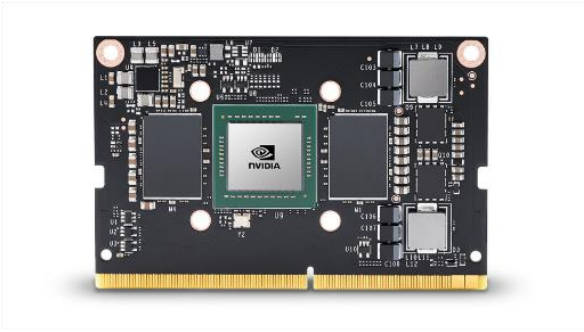
d)

UNIWERSYTET MIKOŁAJA KOPERNIKA W TORUNIU
Wydział Fizyki, Astronomii i Informatyki Stosowanej

Komputery jednopłytkowe wspierające ROS:

a) Raspberry Pi 4 (https://www.raspberrypi.com/products/raspberry-pi-4-model-b/)
b) Odroid XU4 (https://www.hardkernel.com/shop/odroid-xu4-special-price/)
c) NVDIA TX2 (https://www.nvidia.com/pl-pl/autonomous-machines/embedded-systems/jetson-tx2/)
d) Intel NUC (www.intel.com/content/www/us/en/products/boards-kits/nuc.html)

a)



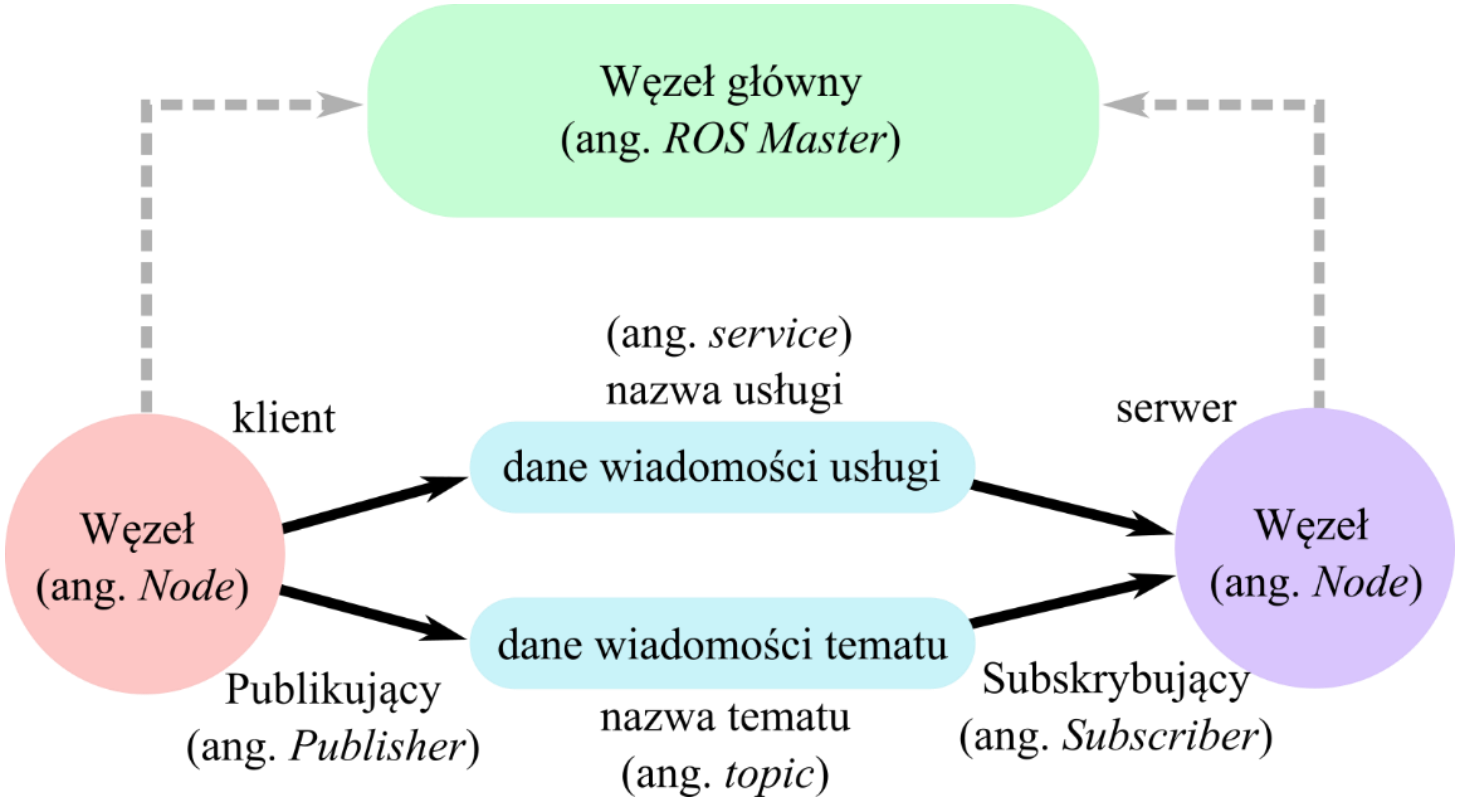
c)

b)

d)

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

Koncepcja Robot Operating System



UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

Koncepcja Robot Operating System

UNIWERSYTET MIKOŁAJA KOPERNIKA W TORUNIU
Wydział Fizyki, Astronomii i Informatyki Stosowanej

Najważniejsze pojęcia związane z Robot Operating System:

- *ROS node (węzeł)*: Proces wykorzystujący interfejs programowania aplikacji ROS,
- *ROS master (węzeł główny)*: Program pośrednik, który łączy węzły
- *ROS parameter server (serwer parametrów)*: Program uruchamiany równocześnie z ROS master. Pozwala na przechowywanie różnych parametrów i wartości.
- *ROS topic (temat)*: Nazwa łącza, przez które węzły mogą przekazywać wiadomości. Każdy węzeł może subskrybować lub publikować dowolną liczbę tematów.
- *ROS message (wiadomość)*: Wiadomości są przekazywane przez tematy. Istnieją wbudowane typy wiadomości, jak również użytkownik może stworzyć własną.
- *ROS service (usługa)*: Usługi pozwalają na zastosowanie mechanizmu pytanie-odpowiedź. Usługa wywołuje funkcje węzła dostarczającego usługę (serwer), która może zostać wywołana przez inny węzeł (klienta) w dowolnym momencie.
- *ROS bag (worek)*: Metoda pozwalająca zapisać i odtworzyć zapisane tematy. Również wykorzystywana do logowania danych robota w celu późniejszego ich wykorzystania.

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

Przykładowe węzły: *talker* i *listener*

Przejdźmy teraz do prostego przykładu, gdzie uruchomimy węzeł główny (*ROS master*), przykładowy węzeł publikujący (*talker*) i węzeł subskrybujący (*listener*). Tematem jaki będzie publikowany to */chatter*. Uruchamiając trzy konsole (można wykorzystać program *Terminator*) należy w każdej z nich uruchomić kolejno:

Węzeł główny:

```
$ roscore
```

węzeł publikujący temat */chatter*:

```
$ rosrun roscpp_tutorials talker
```

węzeł subkrybujący temat */chatter*:

```
$ rosrun roscpp_tutorials listener
```

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

Przykładowe węzły: *talker* i *listener*

ros@ros-VirtualBox: ~

roscore http://ros-VirtualBox:11311/ 132x9

```
auto-starting new master
process[master]: started with pid [3090]
ROS_MASTER_URI=http://ros-VirtualBox:11311/

setting /run_id to c4d796ee-aaa3-11ed-8dea-7f04338e31ed
process[rosout-1]: started with pid [3100]
started core service [/rosout]
```

ros@ros-VirtualBox: ~ 132x11

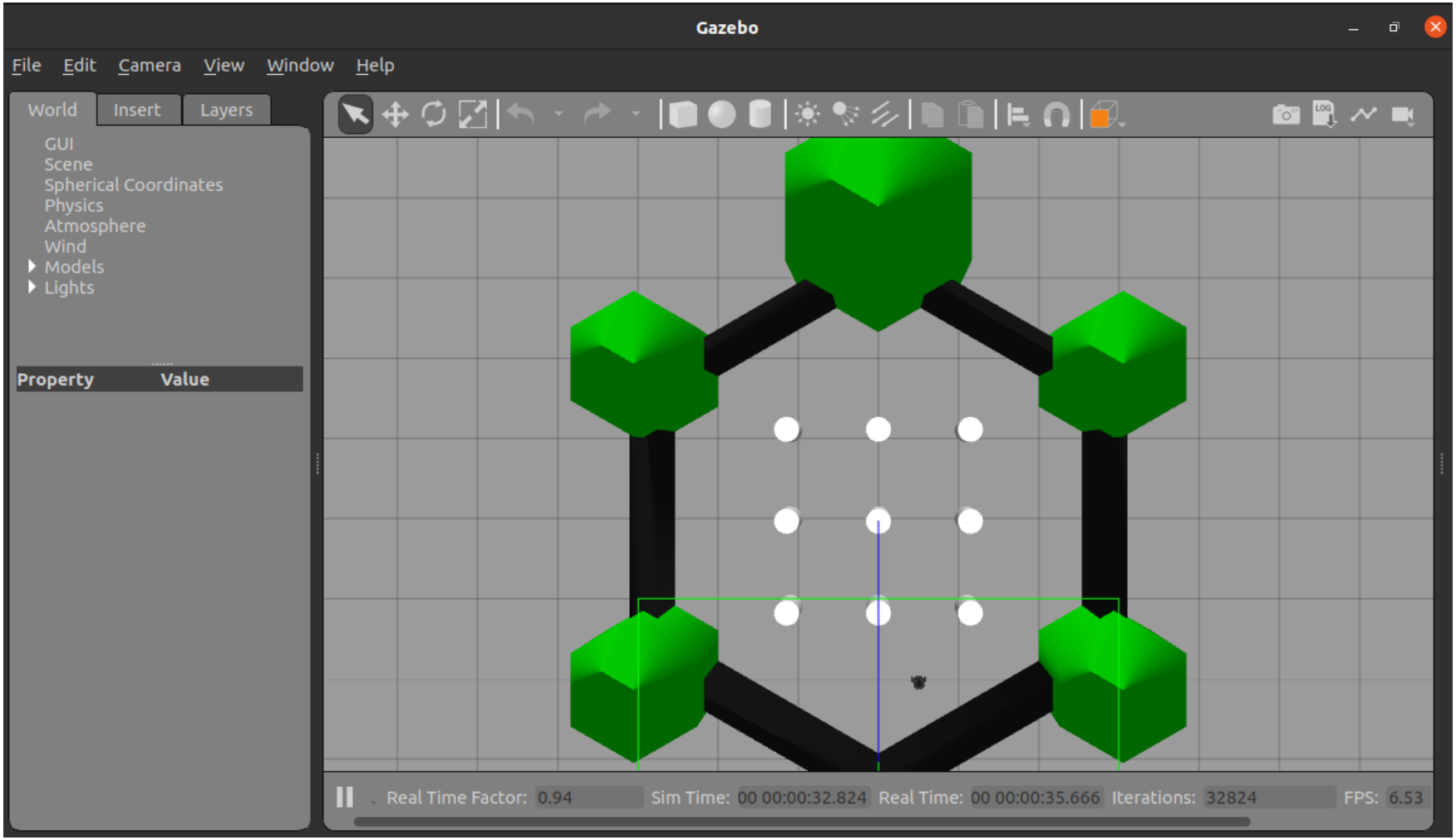
```
ros@ros-VirtualBox:~$ rosrun roscpp_tutorials talker
[ INFO] [1676185646.463091810]: hello world 0
[ INFO] [1676185646.563457835]: hello world 1
[ INFO] [1676185646.672944123]: hello world 2
[ INFO] [1676185646.763302746]: hello world 3
[ INFO] [1676185646.864816020]: hello world 4
[ INFO] [1676185646.964616172]: hello world 5
[ INFO] [1676185647.065436275]: hello world 6
[ INFO] [1676185647.190838244]: hello world 7
[ INFO] [1676185647.272571957]: hello world 8
[ INFO] [1676185647.373560142]: hello world 9
```

ros@ros-VirtualBox: ~ 132x11

```
ros@ros-VirtualBox:~$ rosrun roscpp_tutorials listener
[ INFO] [1676185646.764065436]: I heard: [hello world 3]
[ INFO] [1676185646.865175073]: I heard: [hello world 4]
[ INFO] [1676185646.964953339]: I heard: [hello world 5]
[ INFO] [1676185647.065688615]: I heard: [hello world 6]
[ INFO] [1676185647.191713097]: I heard: [hello world 7]
[ INFO] [1676185647.273196611]: I heard: [hello world 8]
[ INFO] [1676185647.373953479]: I heard: [hello world 9]
[ INFO] [1676185647.472972171]: I heard: [hello world 10]
[ INFO] [1676185647.564075730]: I heard: [hello world 11]
[ INFO] [1676185647.663669711]: I heard: [hello world 12]
[ INFO] [1676185647.777128839]: I heard: [hello world 13]
```

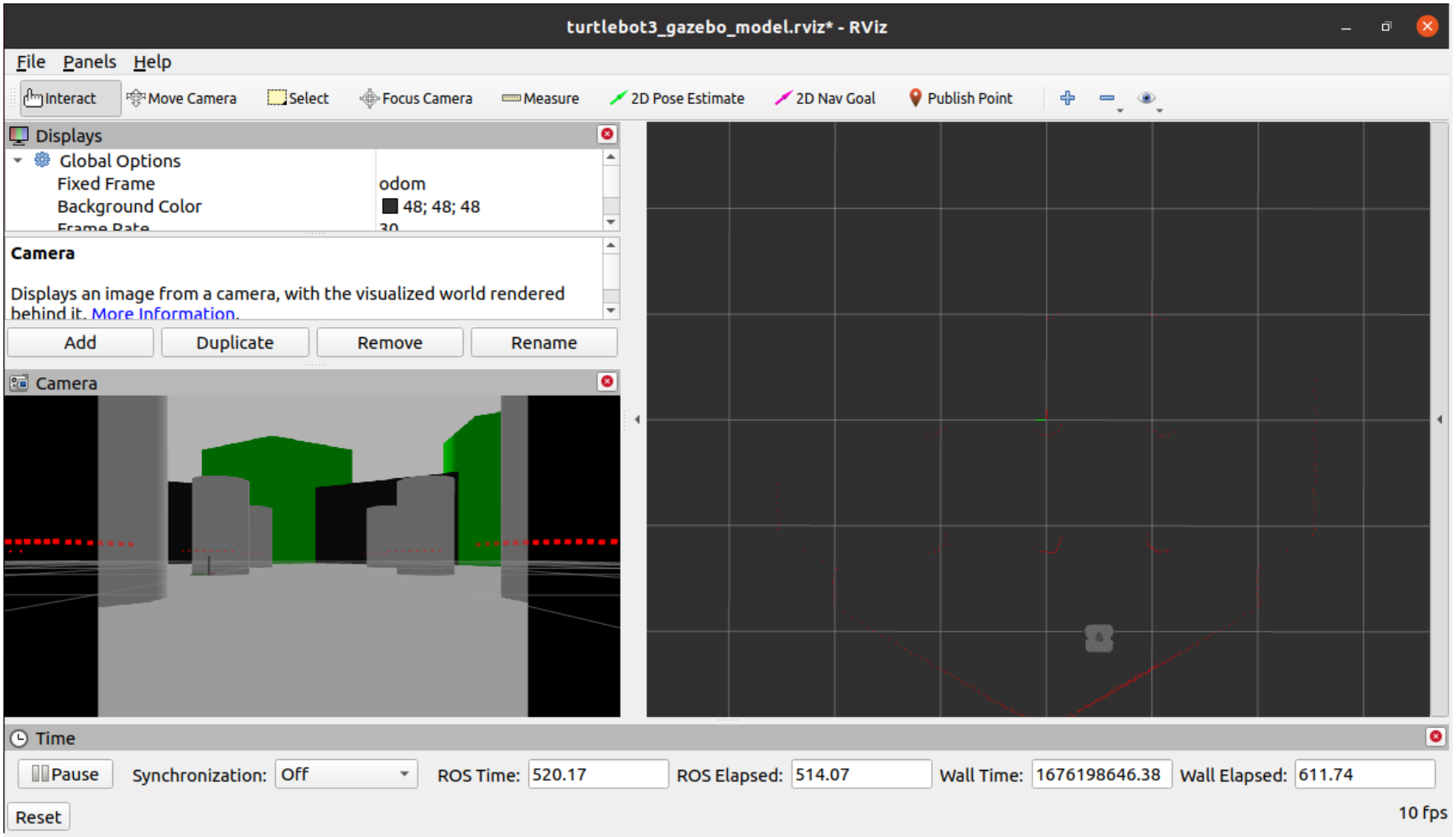
UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

Gazebo – środowisko symulacyjne



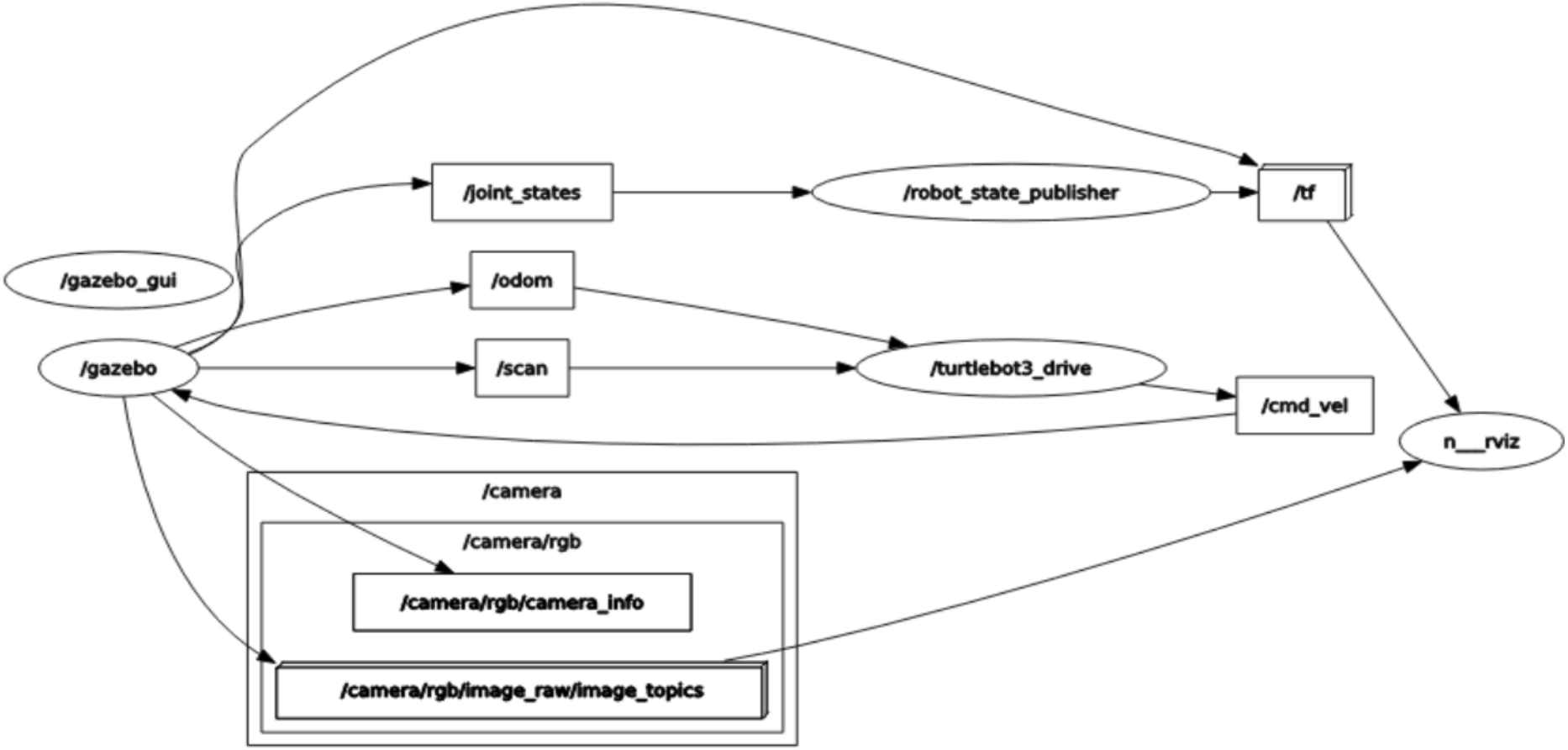
UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

RViz – wizualizacja danych



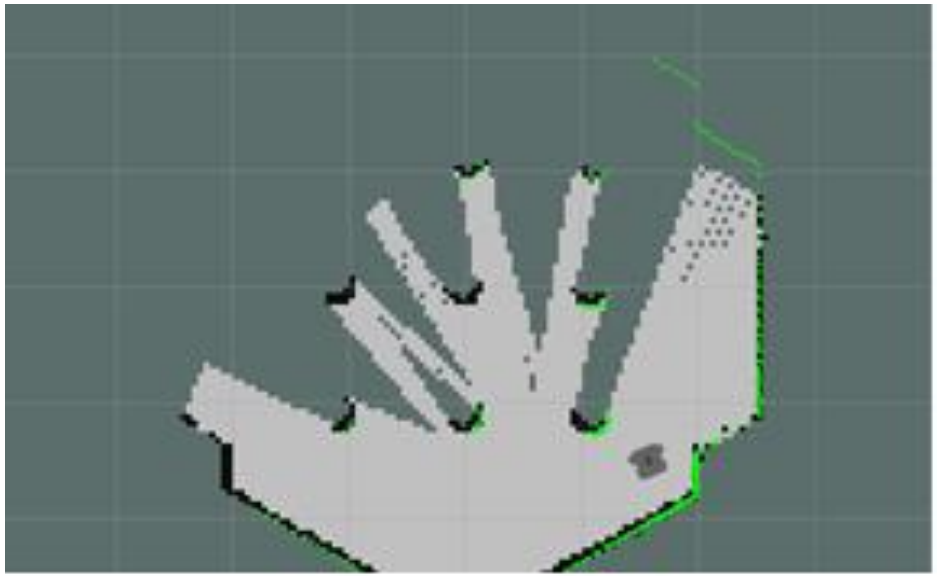
UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

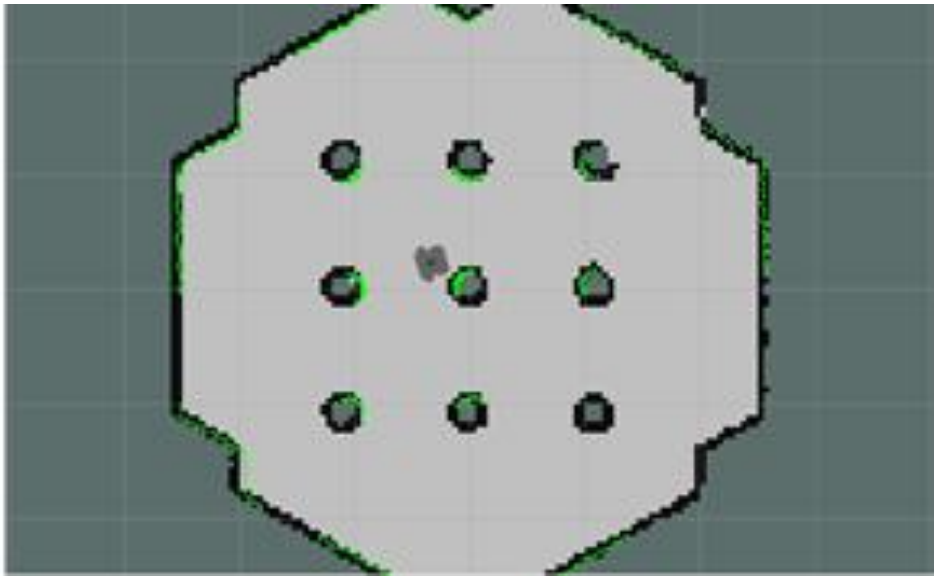
Gazebo i Rviz – bezkolizyjny ruch robota TurtleBot 3



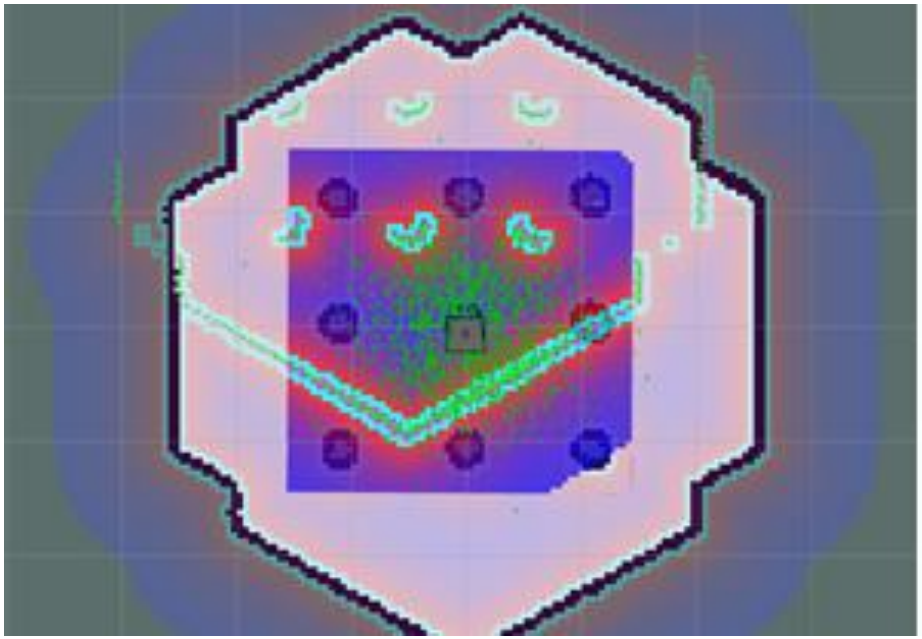
UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

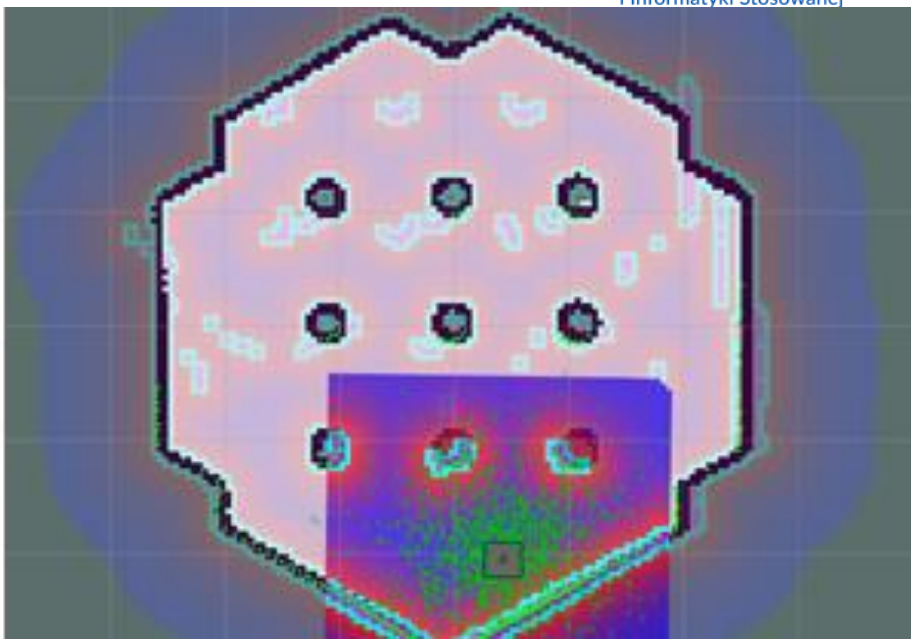
Gazebo i Rviz – mapowanie terenu

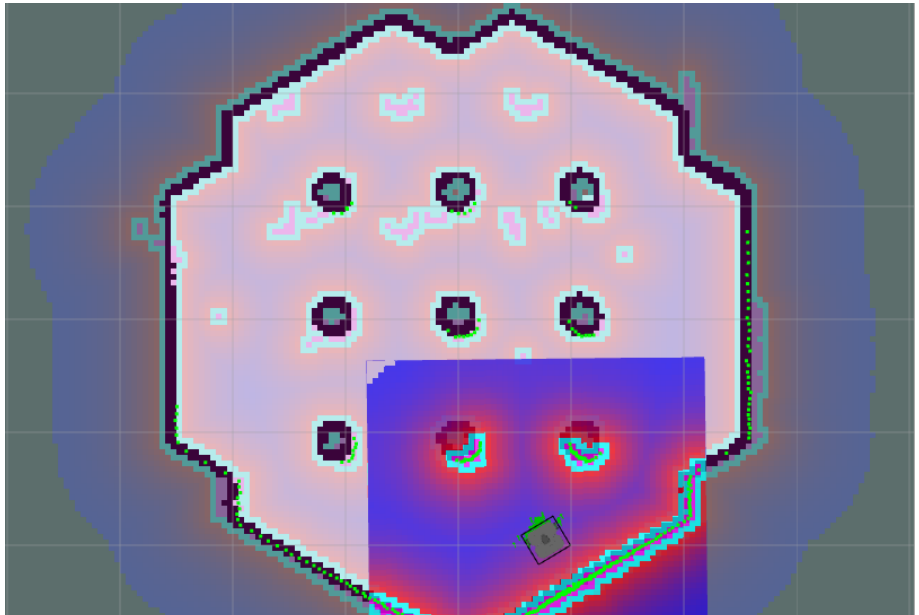




UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

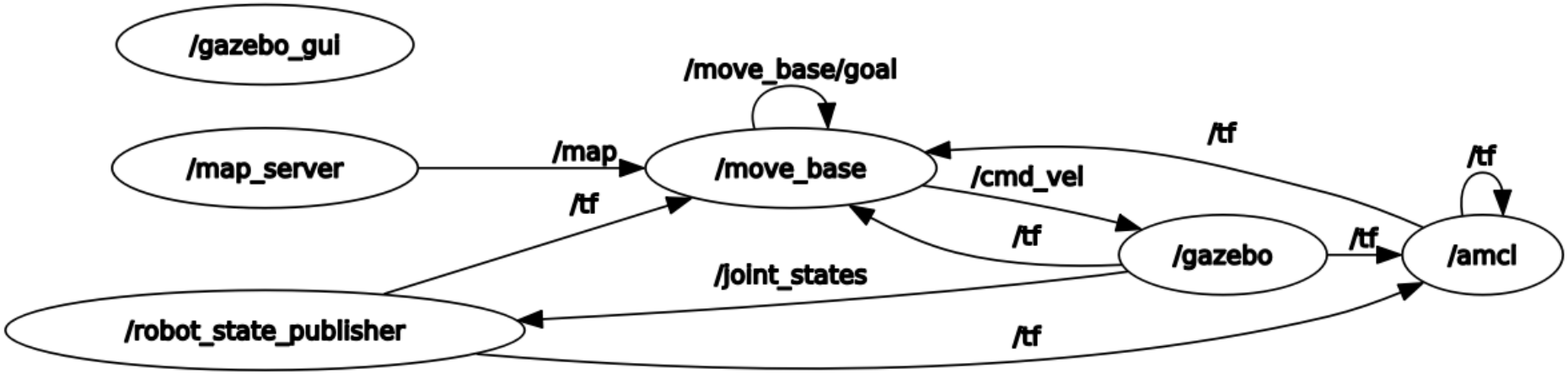


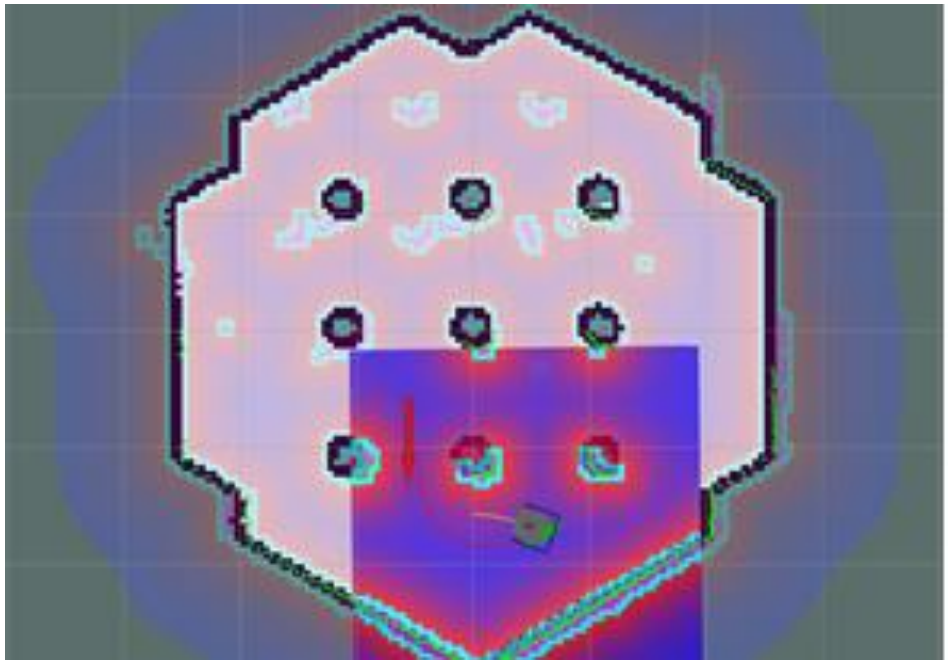


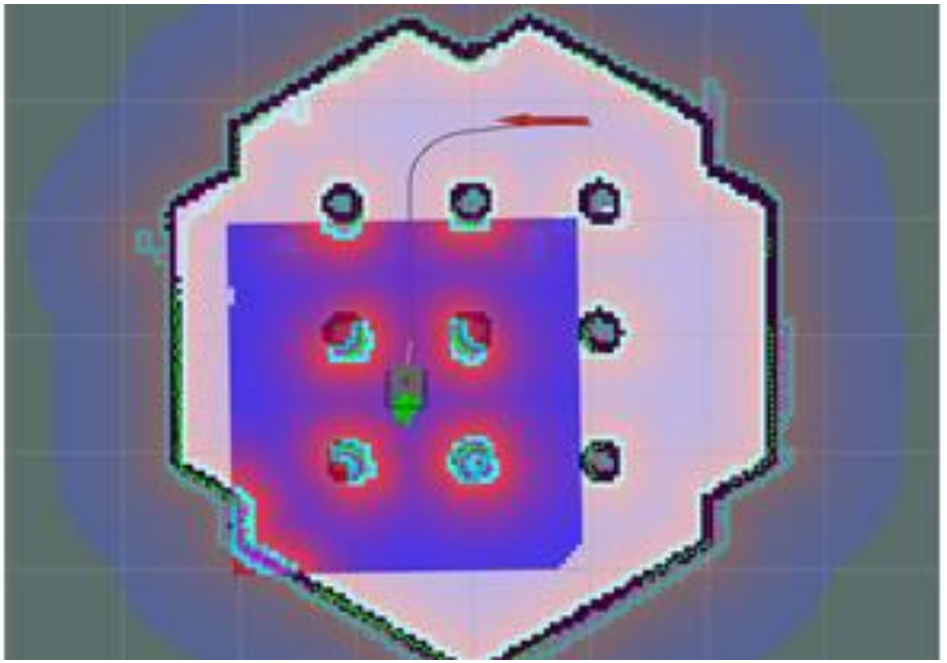


UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

Gazebo i Rviz – Algorytmy lokalizacji i planowania ścieżki

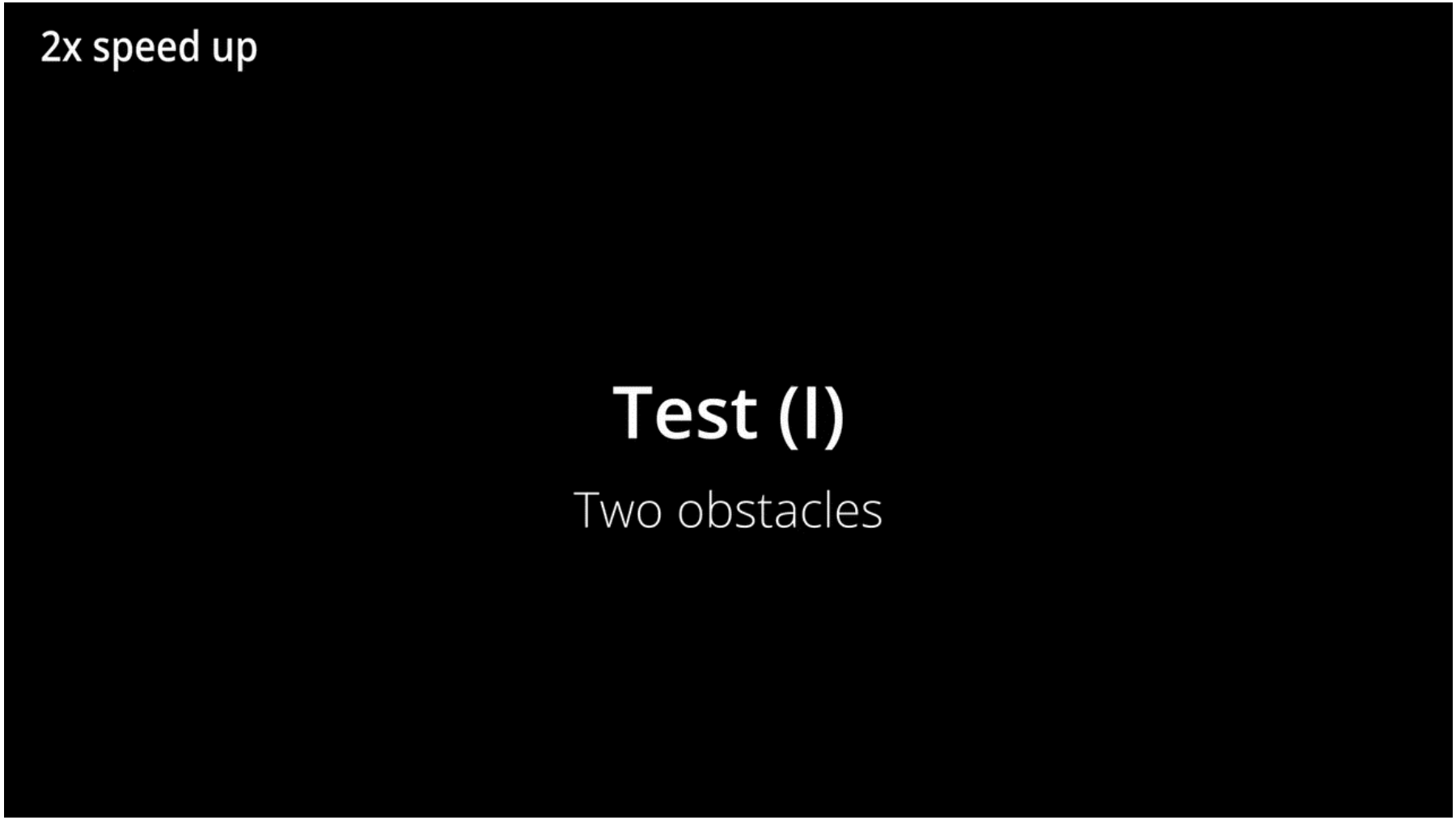






UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

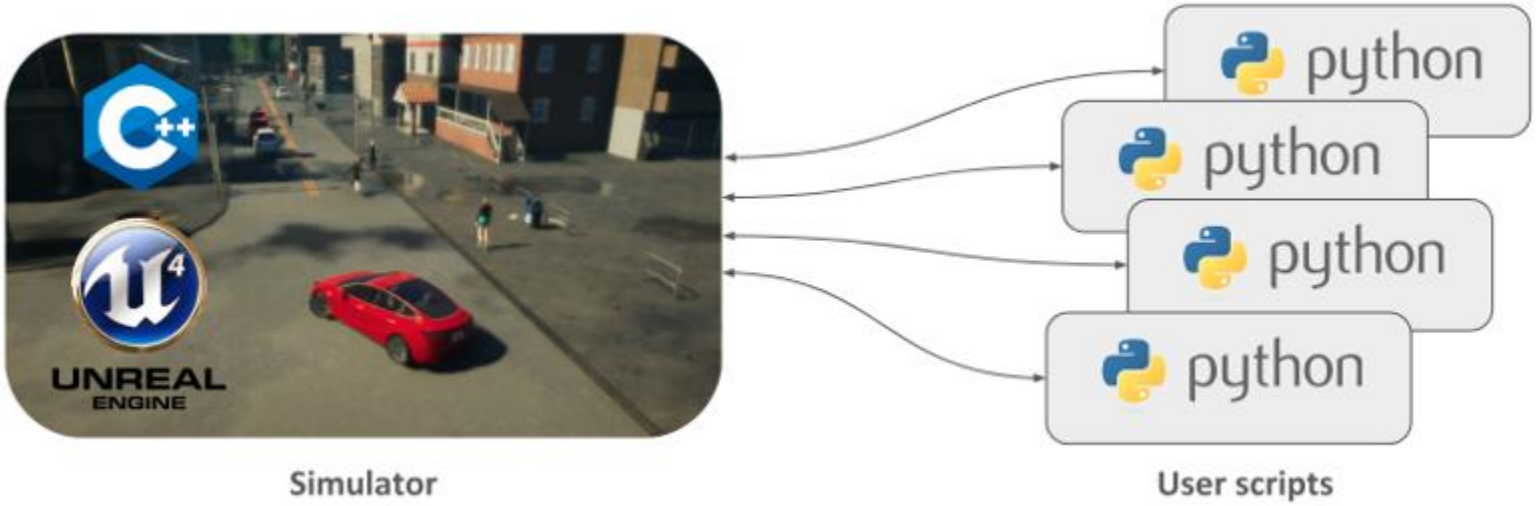
ROS w rzeczywistym robocie mobilnym



UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

CARLA simulator

- Projekt open-source
- Bazuje na silniku Unreal
- Wykorzystuje standard OpenDRIVE do definicji dróg i środowiska
- Wbudowane pojazdy poruszające się autonomicznie
- Możliwość pisania własnych skryptów do sterowania autonomicznym pojazdem w języku Python



UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

CARLA simulator

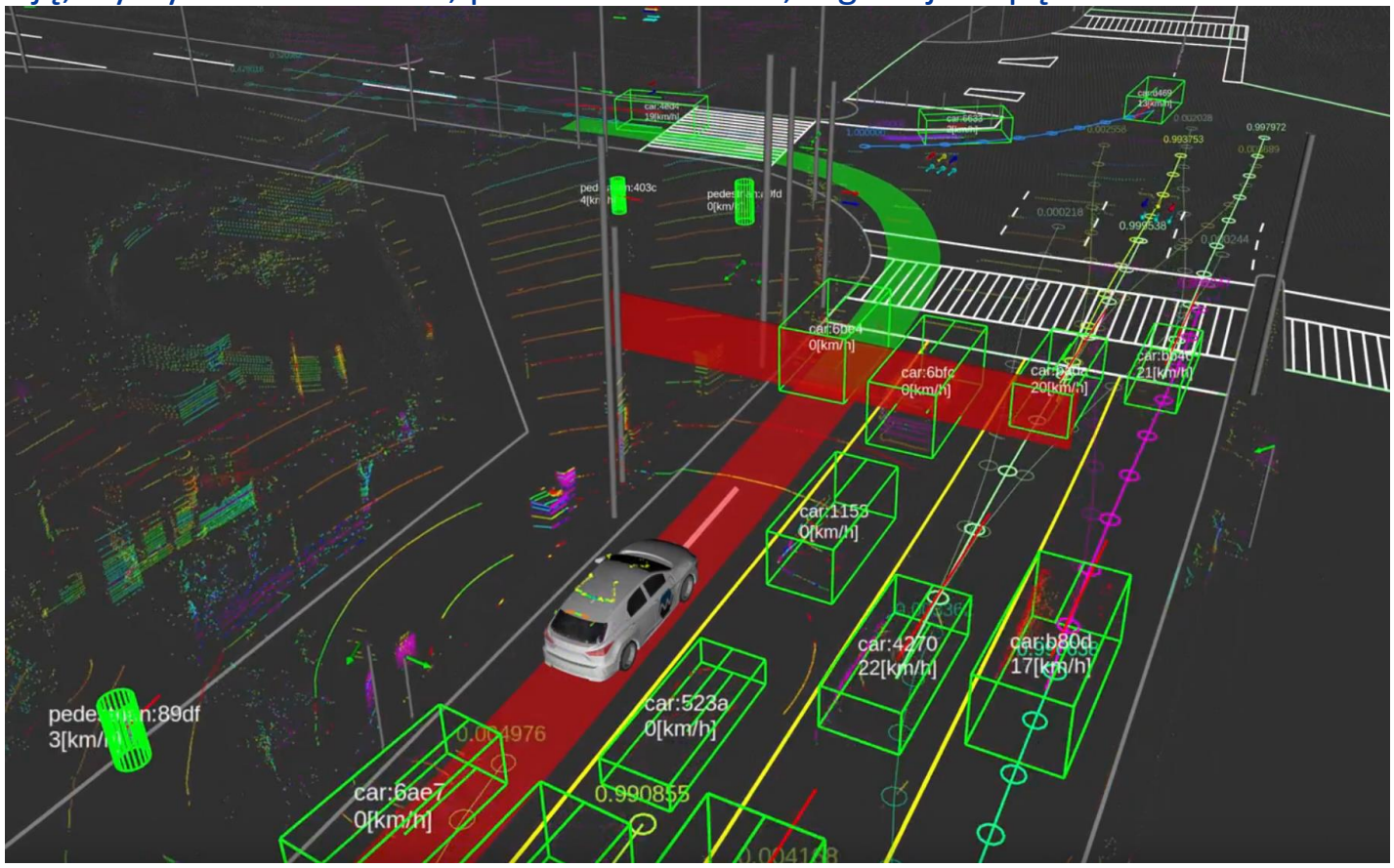
UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

CARLA simulator

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

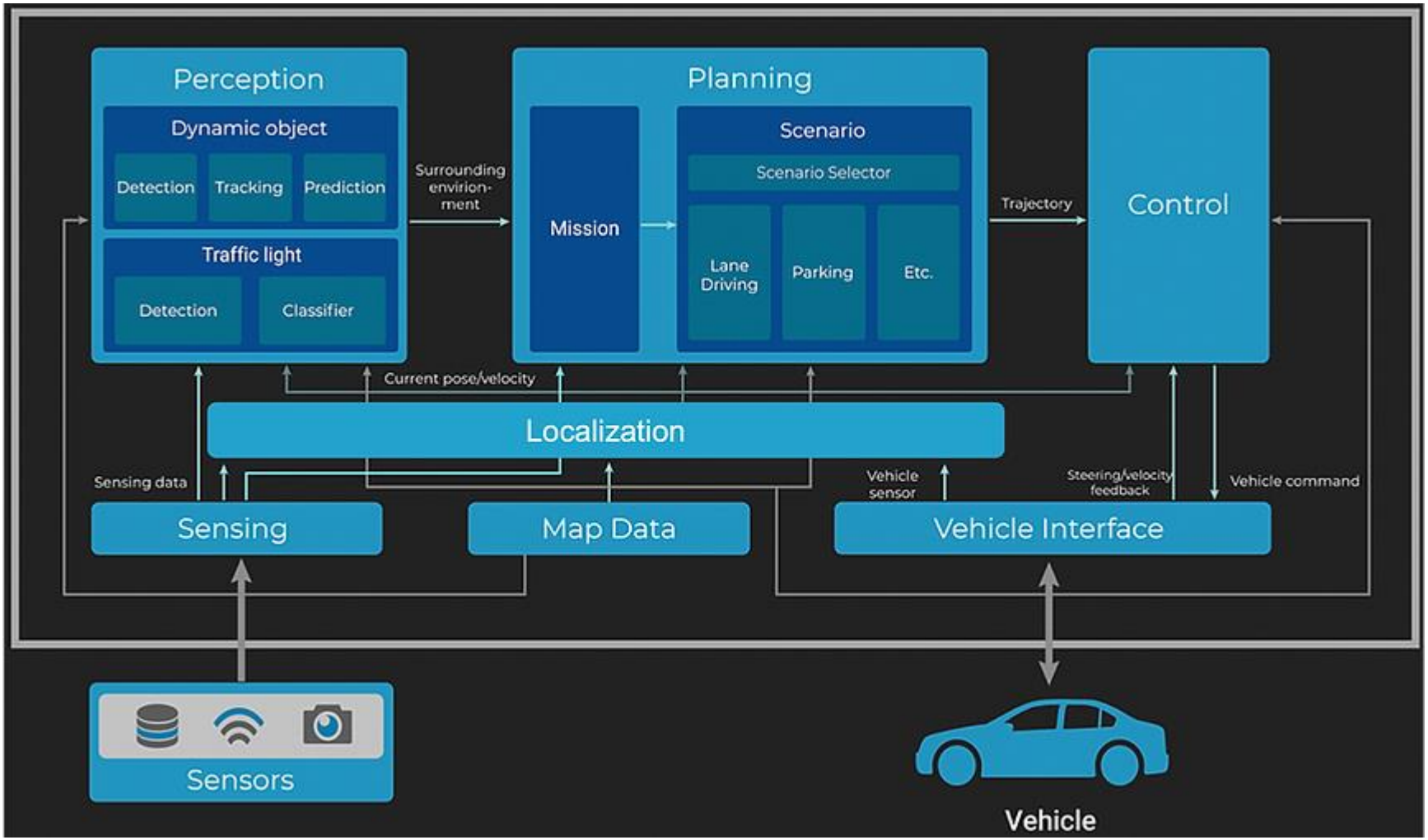
Autoware Foundation

- Projekt open-source
- Zintegrowany z ROS2
- Gotowy stos dla autonomicznych pojazdów z wbudowanymi metodami dotyczącymi: lokalizację, wykrywanie obiektów, planowanie ścieżki, regulacja napędów



UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

Autoware Foundation



UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

Autoware Foundation

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

Autoware Foundation

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

OpenPilot

- Projekt open-source,
- Wbudowane funkcje tj.: Adaptive Cuise Control (ACC), Automated Lane Centering (ALC), pilnowanie czy kierowca patrzy na drogę, przyśpieszanie, hamowanie, podążanie za drogą,
- Działa z dowolnym samochodem obsługującym sieć CAN jeżeli wspierana jest opcja ADAS.

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

OpenPilot

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

OpenPilot

UNIWERSYTET
MIKOŁAJA KOPERNIKA
W TORUNIU
Wydział Fizyki, Astronomii
i Informatyki Stosowanej

Youtube channel: AI DRIVR