Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕТРОЗАВОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Институт математики и информационных технологий

Отчет о прохождении производственной практики

Выполнил студент группы 22207
Клюшов Никита Константинович

Направление подготовки:
Системное и прикладное программное обеспечение

Место прохождения практики:
Кафедра информатики и математического обеспечения,
Центр искусственного интеллекта

Сроки прохождения практики:
30.05.2022-09.06.2022

Руководитель практики:
к.т.н., доцент Богоявленская Ольга Юрьевна

Оценка _____

Дата _____

Петрозаводск 2022 г.

Перечень терминов

В ходе описания отчёта по производственной практике будут использоваться следующие термины:

- Gazebo - программа с открытым исходным кодом, моделирующая работу робота.
- ROS - Robot Operating System - программная платформа, назначение которой – написание программного обеспечения робота
- SLAM - Simultaneous Localization and Mapping - метод навигации.
- Turtlebot3 Burger - модель мобильного робота.

Содержание

Введение

Роботы – это автоматизированные машины, способные взаимодействовать с окружающим миром подобно человеку. О них люди мечтали еще с древних времен, и вот сейчас эти механизмы появляются в нашей жизни и уже активно используются во многих сферах жизнедеятельности человека. Основное их предназначение – сделать нашу жизнь более комфортной, улучшить условия труда, освободить людей от сложных и опасных работ и увеличить производительность.

Одно из направлений, в котором роботы приносят пользу уже сейчас - исследование потенциально пригодных для обитания людей планет. Так, NASA - федеральное правительство США, ответственное за гражданскую космическую программу, исследования в области аэронавтики и космических исследований - в прошлом году успешно отправило на Марс космического робота «Настойчивость» (Perseverance). Среди задач робота — слежение за пыльными бурями, геологические исследования и эксперименты по получению кислорода и топлива из богатой углекислым газом атмосферы планеты. «Настойчивость» стал пятым марсоходом, отправленным США на Красную планету. Теоретически, он сможет подготовить к отправле на Марс первых людей.

Для того, чтобы проверить работоспособность робота и протестировать его в "боевых"условиях, робот сначала тестируется в виртуальной среде, так как это может дать представление о поведении робота, но в отличие от реальных испытаний, тестирование в виртуальной среде менее затратно по времени и требует намного меньших финансовых затрат.

Цель практики: Автономное движение и навигация мобильного робота.
Задачи практики:

1. Изучить фреймворк «ROS»;
2. Изучить 3D-симулятор робототехники «Gazebo»;
3. Научиться взаимодействовать с утилитой «RViz»;
4. Научиться запускать виртуальную среду для тестировки уже существующего робота;
5. Научиться работать с созданием и использованием карт местности;
6. Научиться работать с автономным движением мобильного робота.

1 Фреймворк «ROS»

Robot Operating System (ROS) — это фреймворк для программирования роботов. ROS упрощает работу с роботами и даёт возможность разработчикам сосредоточиться на своей конкретной задаче. ROS предоставляет сервисы, такие как аппаратная абстракция, низкоуровневое управление устройствами, реализация часто используемых функций, передача сообщений между процессами и управление пакетами (плагинами).

ROS спроектирована как слабо связанная система, в которой процесс, называемый узлом (node), должен отвечать за одну задачу. Узлы общаются друг с другом, используя сообщения, проходящие через логические каналы, называемые темами (topics). Каждый узел может отправлять или получать данные от другого узла, используя шаблон проектирования издатель-подписчик (publish–subscribe pattern).

Для ROS, уже реализованы драйвера, позволяющие единым образом работать со многими устройствами, такими как контроллеры, GPS, камеры, лазерные дальномеры и т. п.

2 Установка «ROS»

Установка ROS проводилась на дистрибутиве Ubuntu 20.04.4 LTS.

Настройка sources.list:

```
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $
    (lsb_release -sc) main" > /etc/apt/sources.list.d/ros-
    latest.list'
```

Настройка ключей:

```
sudo apt install curl
curl -s https://raw.githubusercontent.com/ros/rosdistro/
    master/ros.asc | sudo apt-key add -
```

Установка ROS

```
sudo apt install ros-noetic-desktop-full
```

Была использована полная версия дистрибутива noetic.

Настройка главного скрипта:

```
source /opt/ros/noetic/setup.bash
```

Настройка зависимостей:

```
sudo apt install python3-rosdep python3-rosinstall
    python3-rosinstall-generator python3-wstool build-
    essential
```

3 Среда разработки «Gazebo»

Gazebo — это динамический 3D-симулятор с открытым исходным кодом, который развивается Open Source Robotic Foundation и довольно тесно взаимодействует с ROS. Gazebo позволяет точно и эффективно моделировать роботов как в сложных условиях помещений, так и снаружи.

Симулятор состоит из сервера gzserver, который занимается просчетом физики, столкновений и симуляцией сенсоров. К серверу могут подсоединяться клиенты, например gzclient (для десктопа) и gzweb (для браузера). Именно они занимаются рендерингом моделей.

Все это дает возможность тестировать сложные робототехнические системы в виртуальном пространстве гораздо быстрее и без риска нанести ущерб дорогостоящим настоящим роботам.

4 Установка «Gazebo»

Gazebo включен в полный установочный пакет ROS, поэтому не нуждается в отдельной установке.

5 Утилита «RViz»

RViz (сокращение от “ROS visualization”) - это программный инструмент для 3D-визуализации роботов, датчиков и алгоритмов. Он позволяет увидеть восприятие роботом своего мира (реального или смоделированного).

Цель RViz - дать возможность визуализировать состояние робота. Он использует данные датчиков, чтобы попытаться создать точное изображение того, что происходит в окружающей среде робота.

6 Установка «RViz»

Установка RViz:

```
sudo apt-get install ros-fuerte-visualization
```

7 Выбор робота

Для изучения тестирования в виртуальной среде мною был выбран виртуальный робот TurtleBot3 модели «Burger».

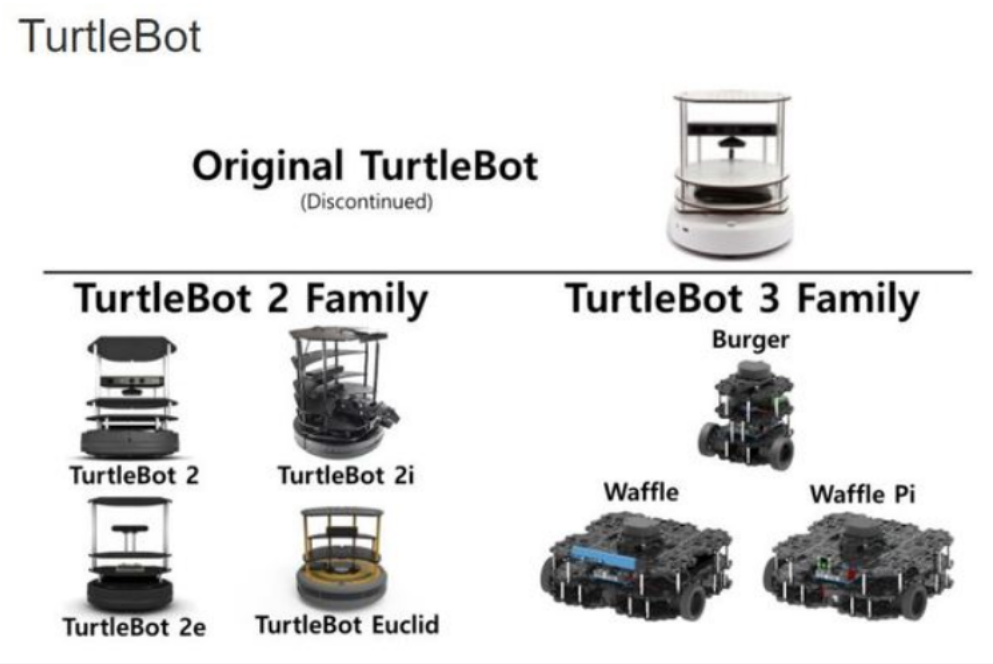

Рис. 1: Семейство роботов TurtleBot

8 Установка робота

Установка зависимых пакетов:

```
cd ~/catkin_ws/src/

git clone https://github.com/ROBOTIS-GIT/turtlebot3_msgs
   .git

git clone https://github.com/ROBOTIS-GIT/turtlebot3.git

cd ~/catkin_ws && catkin_make
```

Установка файлов симуляции TurtleBot3:

```
cd ~/catkin_ws/src/

git clone https://github.com/ROBOTIS-GIT/
   turtlebot3_simulations.git

cd ~/catkin_ws && catkin_make
```

9 Запуск виртуальной среды

Запуск робота в Gazebo:

```
roslaunch turtlebot3_gazebo turtlebot3_world.launch
```

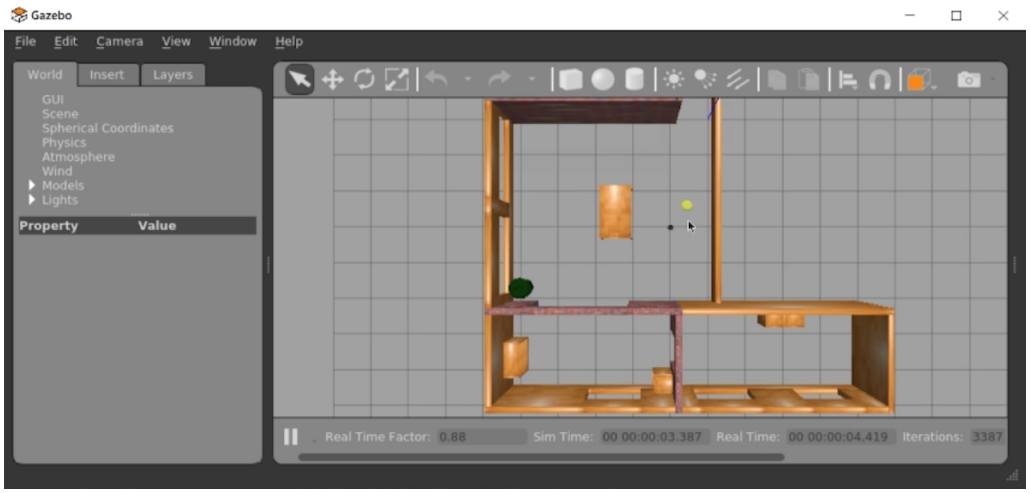

Рис. 2: Внешний вид среды разработки «Gazebo»

Запуск метода SLAM(simultaneous localization and mapping — одновременная локализация и построение карты) в RViz:

```
roslaunch turtlebot3_slam turtlebot3_slam.launch
    slam_methods:=gmapping
```

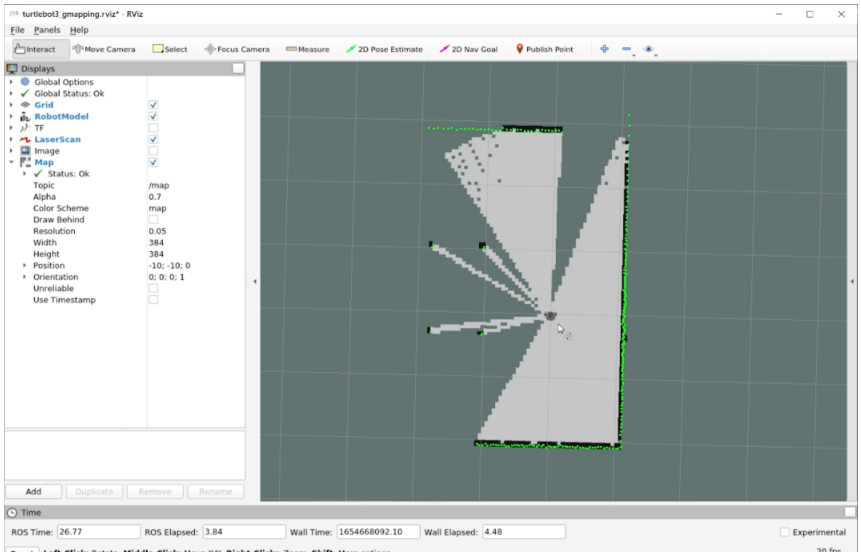

Рис. 3: Внешний вид утилиты RViz. По центру располагается робот, который датчиками получает информацию о препятствиях

На приведённом выше изображении уже можно увидеть работу метода SLAM - препятствия определяются датчиками робота и их местоположение сохраняются на карте.

Симуляция автоматического движения робота:

```
roslaunch turtlebot3_gazebo turtlebot3_simulation.launch
```

textbfили
Ручное управление роботом через консоль с помощью клавиатуры:

```
roslaunch turtlebot3_teleop turtlebot3_teleop_key.launch
```

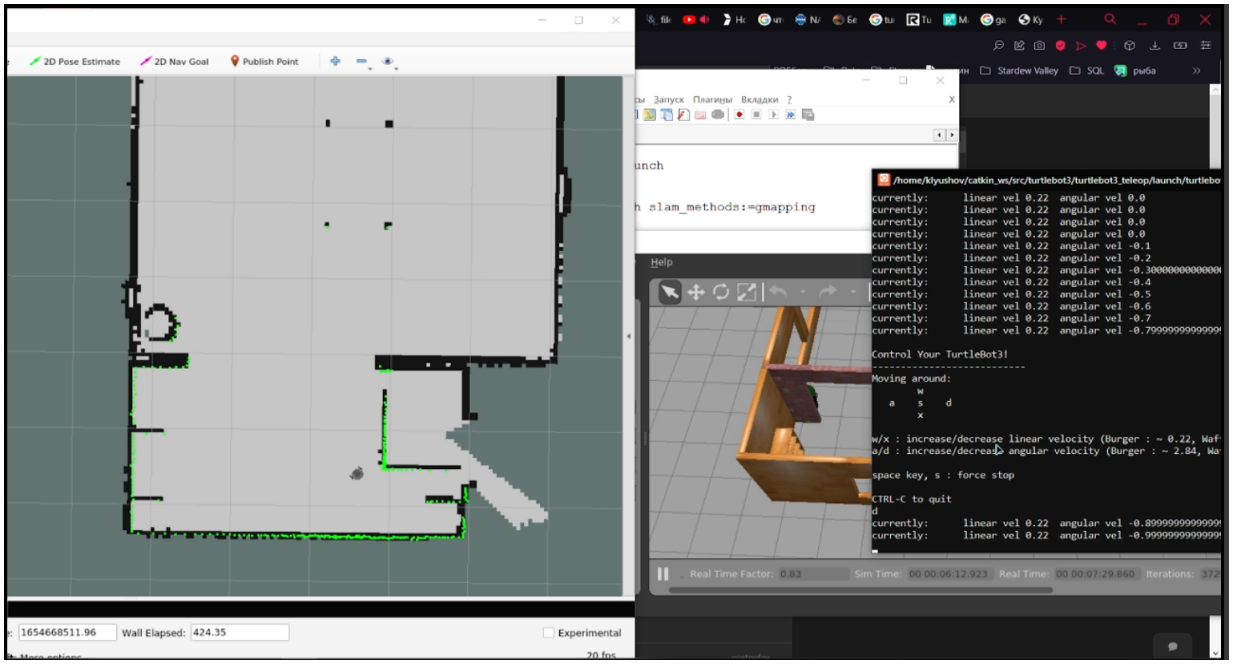

Рис. 4: Изучение роботом карты методом SLAM. Слева - изученная часть карты в RViz.

Спустя некоторое время, когда робот изучит некоторую часть карты, эту карту можно экспортировать в виде .yaml файла:

```
rosrun map_server map_saver
```

```
klyushov@WIN-O4MLJLNKSDP:~$ rosrun map_server map_saver
[ INFO] [1654687536.688615300]: Waiting for the map
[ INFO] [1654687537.017587600, 64.509000000]: Received a 384 X 384 map @ 0.050 m/pix
[ INFO] [1654687537.032232200, 64.511000000]: Writing map occupancy data to map.pgm
[ INFO] [1654687537.042375500, 64.518000000]: Writing map occupancy data to map.yaml
[ INFO] [1654687537.045728000, 64.518000000]: Done

klyushov@WIN-O4MLJLNKSDP:~$ ls
catkin_ws  map.pgm  map.yaml
```

Рис. 5: Сохранение карты

Теперь можно навигацию робота по уже созданной карте. Для этого в следующей команде надо в качестве параметра передать созданную карту:

```
roslaunch turtlebot3_navigation turtlebot3_navigation.
    launch map_file:=$HOME/map.yaml
```

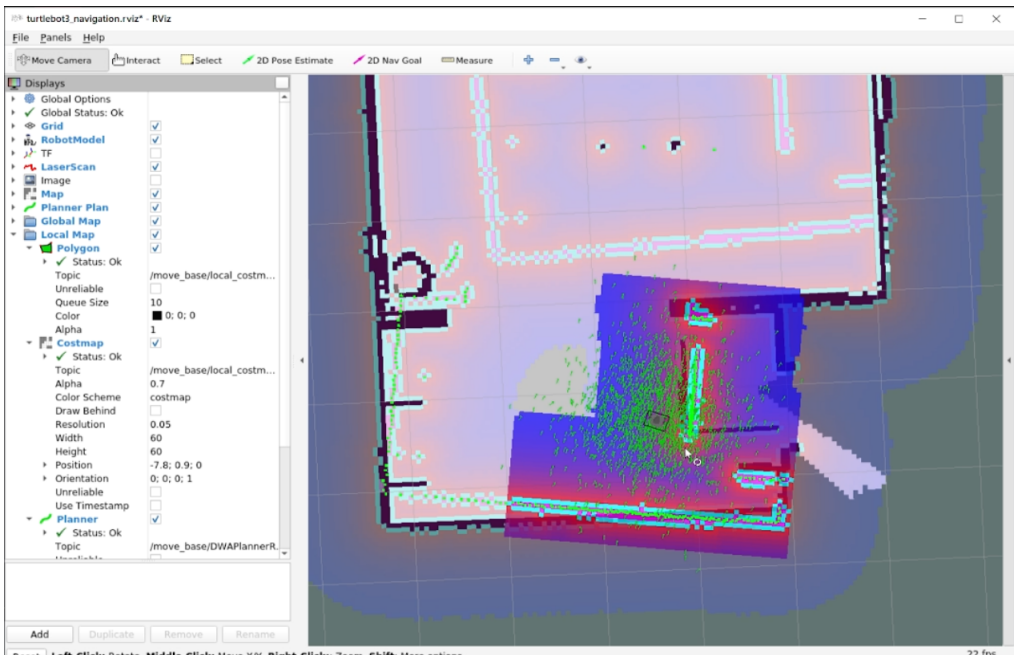

Рис. 6: Навигация по переданной карте.

Так как с самого начала робот не знает, где находится, нужно с помощью кнопки «2D Pose Estimate» инициализировать его начальное положение и направление взгляда, то есть сообщить роботу где конкретно на карте он находится.

И теперь с помощью кнопки «2D Nav Goal» можно указать роботу конечную точку и направление взгляда и робот будет двигаться к заданной точке автоматически, избегая препятствия.

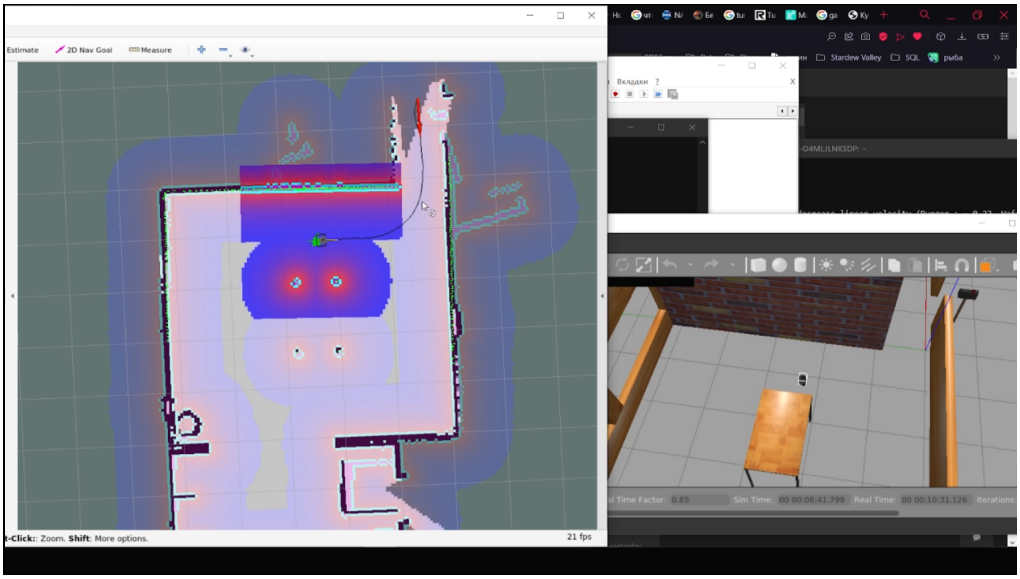

Рис. 7: Перемещение робота по переданной карте в конкретную точку. Слева окно RViz: в центре робот, чёрная линия - путь, красная стрелка - конечная точка с направлением взгляда; справа снизу окно Glazebo: робот едет к конечной точке.

При этом датчики робота продолжают получать информацию о препятствиях. Робот сопостовляет свою информацию о препятствиях с данными на загруженной карте и накладывает первое изображение на второе.

Также стоит уточнить, что SLAM не используется при автоматическом движении робота по загруженной карте, и поэтому навигация робота по ещё не открытой местности, по неправильной карте или без соответствующей начальной инициализации может привести к непредвиденным проблемам.

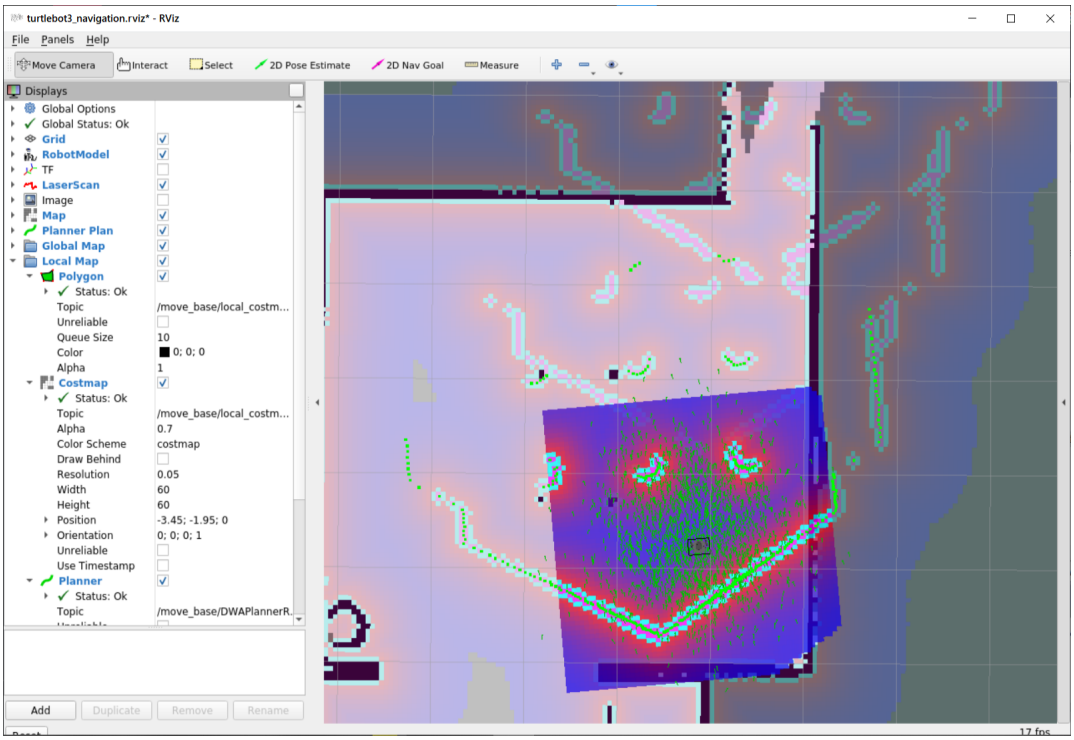

Рис. 8: Использование неправильной карты. Робот пытается наложить информацию о препятствиях с датчиков на карту и препятствия не совпадают.

Заключение

В ходе данной практики удалось поработать с фреймворком ROS, средой разработки Glazebo и утилитой Rviz, а также опробовать симуляцию автономного движения мобильного робота с обходом препятствий (как автоматического, так и с помощью задания координат начала и конца пути) и построения и использования карты местности, которую автоматически строит робот благодаря своим датчикам.

Список литературы

[1] Официальная документация ROS [Электронный ресурс]: Статья. - Режим доступа: http://wiki.ros.org

[2] Gazebo [Электронный ресурс]: Статья. - Режим доступа: https://gazebosim.org/home

[3] How to Launch the TurtleBot3 Simulation With ROS [Электронный ресурс]: Статья. - Режим доступа: https://automaticaddison.com/how-to-launch-the-turtlebot3-simulation-with-ros/gazebo

[4] Setting up ROS in Windows through WSL2 [Электронный ресурс]: Статья. - Режим доступа: https://jack-kawell.com/2020/06/12/ros-wsl2/

[5] Документация по работе с TurtleBot3 [Электронный ресурс]: Статья. - Режим доступа: https://emanual.robotis.com/docs/en/platform/turtlebot3/overview/overview

[6] OpenAI Gym+ROS+Gazebo: обучение автономного робота в домашних условиях. [Электронный ресурс]: Статья. - Режим доступа: https://habr.com/ru/post/441218/