**Лабораторная работа №4**

**Выполнила:** Исламова Сания

**Группа:** НПИмд-01-24

**Задание:** Создать планирующий агент. Он должен уметь перемещаться из одной точки в другую. Он должен уметь распознавать среду.

Для выполнения данной лабораторной работы будем использовать ROS2, Nav2.

1. Мы создадим карту окружающей среды с помощью SLAM.
2. Мы заставим робота перемещаться по этой карте с помощью функций и инструментов навигации.

**SLAM** (Simultaneous Localization and Mapping) — метод навигации, который используется для определения местоположения и ориентации автономных роботов на заранее неизвестной местности, а также для обновления или дополнения уже известных карт окружающего пространства.

Принцип работы SLAM заключается в следующем:

1. Роботу необходимо в каждый момент времени знать своё местоположение.
2. Он постепенно сканирует окружающее пространство при помощи сенсоров, составляя карту местности.
3. Карта строится постепенно, по мере исследования роботом новых областей.
4. Основным источником информации о местоположении робота является одометрия, полученная тем или иным образом (колеса, компьютерное зрение, IMU или их комбинация).
5. По мере выстраивания карты робот начинает сверяться с ней.
6. Если аппарат понимает, что текущие показания одометрии не соответствуют показаниям карты, происходит корректировка одометрии.

Используется Ubuntu 22.04 jammy на VirtualBox, ROS2 Humble.

Установка пакетов Nav2:

*sudo apt install ros-humble-navigation2 ros-humble-nav2-bringup*

Поскольку мы будем использовать симуляцию робота в Gazebo, давайте установим пакеты для этого робота (Turtlebot3):

*sudo apt install ros-humble-turtlebot3\**

Сначала нам нужно экспортировать переменную среды, чтобы выбрать, какую версию Turtlebot3 мы хотим использовать (burger, waffle, waffle\_pi).

Добавим строку в конец нашего файла .bashrc:

*Export TURTLEBOT3\_MODEL=waffle*

После этого нужно выполнить команду source .bashrc или открыть новый терминал.

Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Затем откроем пустой мир в Gazebo / имитациz робота в мире Gazebo:

*ros2 launch turtlebot3\_gazebo turtlebot3\_world.launch.py*

Заставим его двигаться. В новом терминале запускаем узел телеуправления:

*ros2 run turtlebot3\_teleop teleop\_keyboard*

*Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание*

*Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание*

Для запуска робота в пустом мире и функциональности SLAM и RViz, выполняем следующее:

*ros2 launch turtlebot3\_gazebo turtlebot3\_world.launch.py*

*ros2 launch turtlebot3\_cartographer cartographer.launch.py use\_sim\_time:=True*

(аргумент «use\_sim\_time», потому что мы работаем в Gazebo и хотим использовать время Gazebo)

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Мы видем TF-данные робота, а также LaserScan от датчика Lidar (красные точки). Некоторые пиксели уже серые/чёрные. Когда мы заставим робота двигаться, пиксели будут разделены на 3 категории:

* Свободное пространство (станет белым),
* Препятствие (станет черным),
* Неизвестное пространство (останется серым).

Теперь заставим робота двигаться в мире. Запускаем телеоператорский узел Turtlebot3:

*ros2 run turtlebot3\_teleop teleop\_keyboard*

Робот начнёт перемещаться и появится результат, можно сохранять карту.

Чтобы сохранить карту, открываем новый терминал и вводим команду:

*ros2 run nav2\_map\_server map\_saver\_cli -f my\_map*

Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

Автоматически созданное описание

Должно быть 2 новых файла:

* my\_map.yaml
* my\_map.pgm

Изображение выглядит как снимок экрана, программное обеспечение, Мультимедийное программное обеспечение, Значок на компьютере

Автоматически созданное описание

Файл PGM — это изображение, представляющее карту, с белыми (свободное пространство), чёрными (препятствия) и серыми (неизвестное пространство) пикселями.

Изображение выглядит как снимок экрана, Мультимедийное программное обеспечение, Графическое программное обеспечение, программное обеспечение

Автоматически созданное описание

Файл YAML содержит важную информацию о карте. Она будет использоваться стеком навигации.

Изображение выглядит как снимок экрана, текст, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Установим и настроим Cyclone DDS для ROS2:

*sudo apt install ros-humble-rmw-cyclonedds-cpp*

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Отрываем файл параметров:

sudo gedit /opt/ros/humble/share/turtlebot3\_navigation2/param/waffle.yaml

И заменяем строку:

Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

Автоматически созданное описание

Заставим робота двигаться по карте.

Запуск робота:

*ros2 launch turtlebot3\_gazebo turtlebot3\_world.launch.py*

В другом терминале запускаем стек навигации и указываем карту, которую сохранили, в качестве аргумента:

*ros2 launch turtlebot3\_navigation2 navigation2.launch.py use\_sim\_time:=True map:=path/to/my\_map.yaml*

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

**Оценка позы в 2D и цели навигации**

Нажмём на кнопку «2D pose estimate» в RViz. Затем нажмём на карту, где находится робот.

Изображение выглядит как текст, снимок экрана, Мультимедийное программное обеспечение, программное обеспечение

Автоматически созданное описание

Можно отдавать навигационные команды. Выбираем «Nav2 Goal», затем нажмите на карту, чтобы выбрать положение + ориентацию, и робот должен начать двигаться в этом направлении.

*Ссылка на видео «Перемещение робота по карте»:* [*https://disk.yandex.ru/i/KyMcKo0t6mFVrA*](https://disk.yandex.ru/i/KyMcKo0t6mFVrA)