

МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное
учреждение высшего образования
«Южный федеральный университет»

Институт математики, механики
и компьютерных наук им. И. И. Воровича

Кафедра прикладной математики и программирования

Ту Евсеев Максим

**Использование системы 3D-моделирования и физической
симуляции Gazebo в ROS 2**

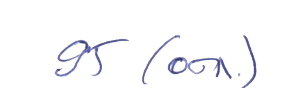

КУРСОВАЯ РАБОТА

по направлению подготовки

01.03.02 – Прикладная математика и информатика

Научный руководитель –

старший преподаватель кафедры ПМП Пучкин Максим Валентинович

	
оценка (рейтинг)	подпись руководителя

Ростов-на-Дону – 2021

Задание на курсовую работу бакалавра

Направление подготовки: прикладная математика и информатика

Образовательная программа бакалавриата:

Студент: Ту Евсеев Максим

Научный руководитель: ст. преподаватель М.В. Пучкин

Год защиты: 2021

Тема работы: Использование системы 3D-моделирования и физической симуляции Gazebo в ROS 2.

Цель работы: рассмотреть возможности системы 3D-моделирования и физической симуляции Gazebo применительно к моделированию робототехнических комплексов, провести анализ механизма интеграции в ROS 2, настройки и основных принципов разработки проектов, а также способы взаимодействия с узлами ROS 2.

Задачи работы:

- изучение функционала и возможностей системы Gazebo, системных требований, рассмотрение процесса установки и настройки приложения;
- анализ основных принципов построения моделей физических систем в среде Gazebo, способов описания моделей, определения физических параметров, и создания симуляций;
- рассмотреть демонстрационный проект, иллюстрирующий способы взаимодействия с элементами инфраструктуры ROS 2, а также возможности управления моделями из ROS 2;
- по результатам проведённой работы подготовить руководство по использованию системы Gazebo для моделирования робототехнических комплексов под управлением ROS 2.

Заведующий кафедрой

Научный руководитель

Студент

10 ноября 2020 г.

Г.А. Угольников

М.В. Пучкин

М. Ту Евсеев

СПРАВКА

Южный Федеральный Университет

о результатах проверки текстового документа
на наличие заимствований

ПРОВЕРКА ВЫПОЛНЕНА В СИСТЕМЕ АНТИПЛАГИАТ.ВУЗ

Автор работы: Ту Евсеев Максим
Самоцитирование
рассчитано для: Ту Евсеев Максим
Название работы: Курсовая работа М. Ту Евсеев
Тип работы: Курсовая работа
Подразделение: ИММиКН ЮФУ

РЕЗУЛЬТАТЫ

■ ОТЧЕТ О ПРОВЕРКЕ КОРРЕКТИРОВАЛСЯ: НИЖЕ ПРЕДСТАВЛЕНЫ РЕЗУЛЬТАТЫ ПРОВЕРКИ ДО КОРРЕКТИРОВКИ

ЗАИМСТВОВАНИЯ 9.65%
ОРИГИНАЛЬНОСТЬ 85.85%
ЦИТИРОВАНИЯ 4.5%
САМОЦИТИРОВАНИЯ 0%

ЗАИМСТВОВАНИЯ 9.65%
ОРИГИНАЛЬНОСТЬ 85.85%
ЦИТИРОВАНИЯ 4.5%
САМОЦИТИРОВАНИЯ 0%

ДАТА ПОСЛЕДНЕЙ ПРОВЕРКИ: 04.06.2021

ДАТА И ВРЕМЯ КОРРЕКТИРОВКИ: 04.06.2021 11:42

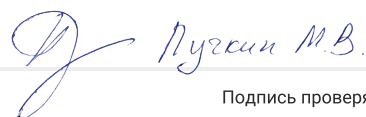
Модули поиска: ИПС Адилет; Библиография; Сводная коллекция ЭБС; Интернет Плюс; Сводная коллекция РГБ; Цитирование; Переводные заимствования (RuEn); Переводные заимствования по eLIBRARY.RU (EnRu); Переводные заимствования по Интернету (EnRu); Переводные заимствования издательства Wiley (RuEn); eLIBRARY.RU; СПС ГАРАНТ; Медицина; Диссертации НББ; Перефразирования по eLIBRARY.RU; Перефразирования по Интернету; Патенты СССР, РФ, СНГ; СМИ России и СНГ; Модуль поиска "ЮФУ"; Шаблонные фразы; Кольцо вузов; Издательство Wiley; Переводные заимствования

Работу проверил: Пучкин Максим Валентинович

ФИО проверяющего

Дата подписи:

4.06.2021


Подпись проверяющего



Чтобы убедиться
в подлинности справки, используйте QR-код,
который содержит ссылку на отчет.

Ответ на вопрос, является ли обнаруженное заимствование
корректным, система оставляет на усмотрение проверяющего.
Предоставленная информация не подлежит использованию
в коммерческих целях.

Отзыв о квалификационной работе (проект 3 курса)
студента бакалавриата М. Ту Евсеева
на тему: «Использование системы 3D-моделирования и физической
симуляции Gazebo в ROS 2»


Целью работы являлось рассмотрение возможностей системы 3D-моделирования и физической симуляции Gazebo применительно к моделированию робототехнических комплексов, анализ механизма интеграции в ROS 2, изучение процедуры настройки и разработки проектов, а также способы взаимодействия с узлами ROS 2.

Система Gazebo предназначена для полноценного физического моделирования различных устройств и механизмов, позволяет определять различные параметры моделируемых системы, обладает возможностью полноценного 3D-рендеринга в реальном времени, а также позволяет выполнять онлайн-управление моделируемой системы извне Gazebo. Изучение базовых возможностей указанной системы, и разработка демонстрационного приложения являлось основной целью курсовой работы.

Работа велась регулярно, поставленные задачи в целом были выполнены. В процессе были продемонстрированы достаточно высокие способности к изучению и анализу технической литературы, навыки разработки и использования сложных имитационных систем. Также следует отметить активное участие М. Ту Евсеева в семинарах робототехнической лаборатории и выступления по тематике проекта.

Работа выполнена на достаточно высоком уровне и заслуживает оценки «отлично» (95 баллов).

Старший преподаватель кафедры
прикладной математики и программирования



М.В. Пучкин

Содержание

Введение	4
1. Постановка задачи.....	5
2. Общие сведения о Gazebo и gazebo_ros_pkgs	6
2.1 Установка Gazebo	6
2.2 Установка gazebo_ros_pkgs.....	7
2.3 Тестирование Gazebo и интеграции ROS 2	7
3. Реализация проекта в Gazebo	9
3.1 SDFormat файла.....	9
3.2 Создание складского робота.....	10
3.2.1 Создание Model.config	10
3.2.2 Загрузка Mesh Files.....	10
3.2.3 Создание SDF модели	11
3.2.4 Постройка склада.....	12
3.3 Запуск робота с помощью ROS 2	13
3.3.1 Создание пакета.....	13
3.3.2 Настройка пакета	14
3.3.3 Создание узла.....	15
3.3.4 Перемещение робота по складу	15
3.3.5 Создание файла запуска и запуск	16
Заключение	18
Литература	19

Введение

Gazebo – универсальная среда симуляции для любых типов роботов. Он обеспечивает реалистичную визуализацию окружающей среды - высококачественное освещение, тени, текстуры. Он может моделировать датчики, которые "видят" моделируемую среду, такие как лазерные дальномеры, камеры (в том числе широкоугольные), датчики стиля Kinect и т. В Gazebo интегрирован физический движок ODE, рендеринг с при помощи OpenGL и код поддержки для управления движениями. Он является лидером в области моделирования роботов. Это инструмент, на который полагаются сотни тысяч пользователей и разработчиков по всему миру.

Разрабатываемый некоммерческой организацией OSRF (Open Source Robotics Foundation), имеет ряд преимуществ по сравнению с другими робототехническими симуляторами. Во-первых, он бесплатный и имеет открытый код. Во-вторых, он очень популярен среди мирового робототехнического сообщества и является официальным симулятором соревнований DARPA. В-третьих, Gazebo отлично интегрируется с программной платформой ROS (Robot Operating System), а значит разработанную вами программу управления виртуальным роботом в Gazebo и ROS будет относительно несложно перенести на реального робота. Целью работы является изучить основы Gazebo и интегрировать его в Ros 2.

1. Постановка задачи

Задачи курсовой работы:

1. Изучение основных принципов установки и настройки Gazebo, интеграция в Ros 2.
2. Изучение интерфейса, создание элементов построения роботов в Gazebo.
3. Реализация и управление проектом в Gazebo, иллюстрирующего основные принципы, на основе Ros 2.

2. Общие сведения о Gazebo и gazebo_ros_pkgs

Gazebo — это автономное приложение, которое можно использовать независимо от ROS или ROS 2. Интеграция Gazebo с любой версией ROS осуществляется с помощью набора пакетов, называемых `gazebo_ros_pkgs`. Эти пакеты обеспечивают мост между API C++ Gazebo и сообщениями служб от ROS 2.

Пакет ROS 2 `gazebo_ros_pkgs` представляет собой группу пакетов, состоящих из:

- `gazebo_dev`: содержит конфигурацию `cmake` для версии Gazebo для дистрибутива ROS. Таким образом, нижестоящие пакеты могут просто строить зависимость от `gazebo_dev` вместо того, чтобы искать Gazebo самостоятельно.
- `gazebo_msgs`: набор служб для взаимодействия с Gazebo из ROS 2.
- `gazebo_ros`: предоставляет необходимые классы и функции C++, которые могут использоваться другими плагинами, например, такими как `gazebo_ros::Node`, утилиты преобразования и тестирования.
- `gazebo_plugins`: Серия плагинов Gazebo, содержащих датчики и другие функции для ROS 2. Например, `gazebo_ros_camera` проецирует изображения ROS 2, а `gazebo_ros_diff_drive` содержит интерфейс для управления и анализа роботов через ROS 2.

2.1 Установка Gazebo

Предполагается, что уже имеется линукс (Ubuntu 20.04 LTS) с установленным на нем ROS 2.

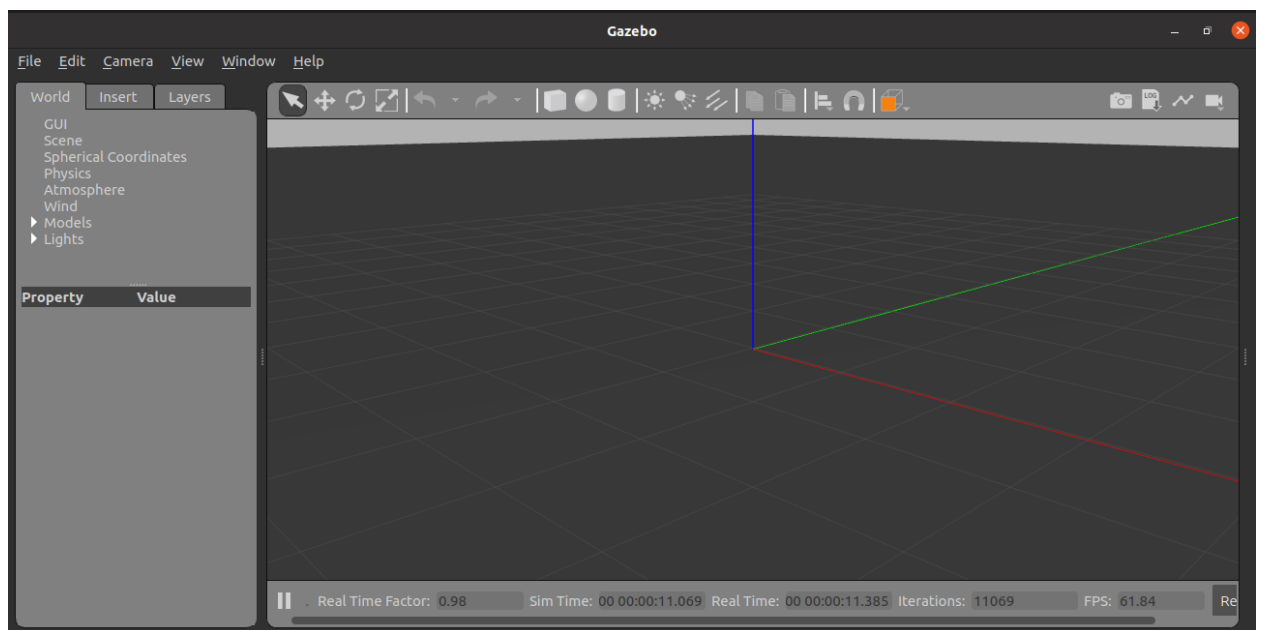
Выполним команду в терминале для установки (скрипт выполнит установку последней стабильной версии Gazebo, на данный момент это Gazebo11):

```
curl -sSL http://get.gazebosim.org | sh
```

Проверяем, что Gazebo установилась, прописав

```
gazebo
```

Должно появиться окно:



Это свидетельствует об успешной установке.

2.2 Установка gazebo_ros_pkgs

Откроем новое окно терминала и установим gazebo_ros_pkgs, который позволят использовать ROS 2 для взаимодействия с Gazebo.

```
sudo apt install ros-foxy-gazebo-ros-pkgs
```

2.3 Тестирование Gazebo и интеграции ROS 2

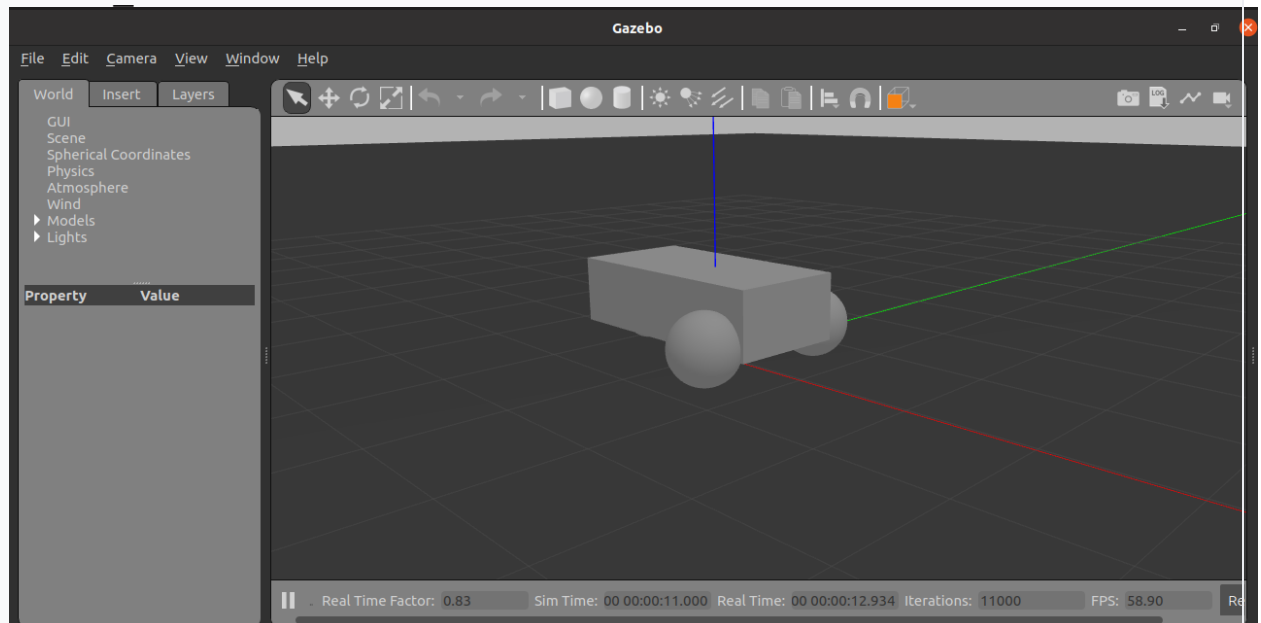
Предполагается, что среды ROS 2 и Gazebo были правильно настроены и установлены, теперь есть возможность загружать модели миров Gazebo, содержащие плагины ROS 2.

Установим еще несколько основных пакетов для корректной работы:

```
sudo apt install ros-foxy-ros-core ros-foxy-geometry2
```

Загрузим демо версию робота:

```
gazebo --verbose  
/opt/ros/foxy/share/gazebo_plugins/worlds/gazebo_ros_diff_  
drive_demo.world
```



Открыв второй терминал, можно посмотреть какие команды нам доступны:

```
gedit  
/opt/ros/foxy/share/gazebo_plugins/worlds/gazebo_ros_diff_  
drive_demo.world
```

```

<!--
Gazebo ROS differential drive plugin demo

Try sending commands:

  ros2 topic pub /demo/cmd_demo geometry_msgs/Twist '{linear: {x: 1.0}}' -1
  ros2 topic pub /demo/cmd_demo geometry_msgs/Twist '{angular: {z: 0.1}}' -1

Try listening to odometry:

  ros2 topic echo /demo/odom_demo

Try listening to TF:

  ros2 run tf2_ros tf2_echo odom_demo chassis
  ros2 run tf2_ros tf2_echo chassis right_wheel
  ros2 run tf2_ros tf2_echo chassis left_wheel
-->

```

Откроем новое окно терминала и введём следующую команду, чтобы заставить робота двигаться вперед со скоростью 1,0 метра в секунду:

```

ros2 topic pub /demo/cmd_demo geometry_msgs/Twist
'{linear: {x: 1.0}}' -1

```

Робот начнет двигаться вперед.

3. Реализация проекта в Gazebo

Мы будем создавать автономного складского робота, а также среду для его передвижения.

3.1 SDFFormat файла

SDFFormat (Simulation Description Format), сокращенно SDF, представляет собой формат XML, который описывает объекты и среды для симуляторов роботов, визуализации и управления. SDFFormat - стабильный, надежный и расширяемый формат, способный описывать все аспекты роботов, статических и динамических объектов, освещения, рельефа местности и даже физики.

3.2 Создание складского робота

3.2.1 Создание Model.config

Создаём папку для модели:

```
mkdir -p ~/.gazebo/models/small_warehouse
```

Создаём файл конфигурации модели. Этот файл будет содержать её описание:

```
gedit ~/.gazebo/models/small_warehouse/model.config
```

```
1 <?xml version="1.0"?>
2 <model>
3   <name>Mobile Warehouse Robot</name>
4   <version>1.0</version>
5   <sdf version='1.4'>model.sdf</sdf>
6
7   <author>
8     <name>sfu-user</name>
9     <email>testemail@gamil.com</email>
10  </author>
11
12  <description>
13    A basic differential drive mobile warehouse robot.
14  </description>
15 </model>
```

3.2.2 Загрузка Mesh Files

Mesh Files помогают сделать роботов более реалистичными как визуально, так и для различных датчиков.

```
cd ~/.gazebo/models
wget -q -R *index.html*,*.tar.gz --no-parent -r -x -nH
http://models.gazebosim.org/warehouse\_robot/
```

Теперь загрузим mesh file лазерного дальномера Hokuyo:

```
wget -q -R *index.html*,*.tar.gz --no-parent -r -x -nH  
http://models.gazebosim.org/hokuyo/
```

3.2.3 Создание SDF модели

Теперь создадим файл SDF. Этот файл будет содержать теги, необходимые для создания модели робота `mobile_warehouse_robot`. У нашего робота будет три колеса: два колеса спереди и колесо сзади. На вершине робота мы установим лазерный дальномер, который будет посылать лучи в радиусе 180 градусов для обнаружения препятствий.

```
gedit ~/.gazebo/models/mobile_warehouse_robot/model.sdf
```

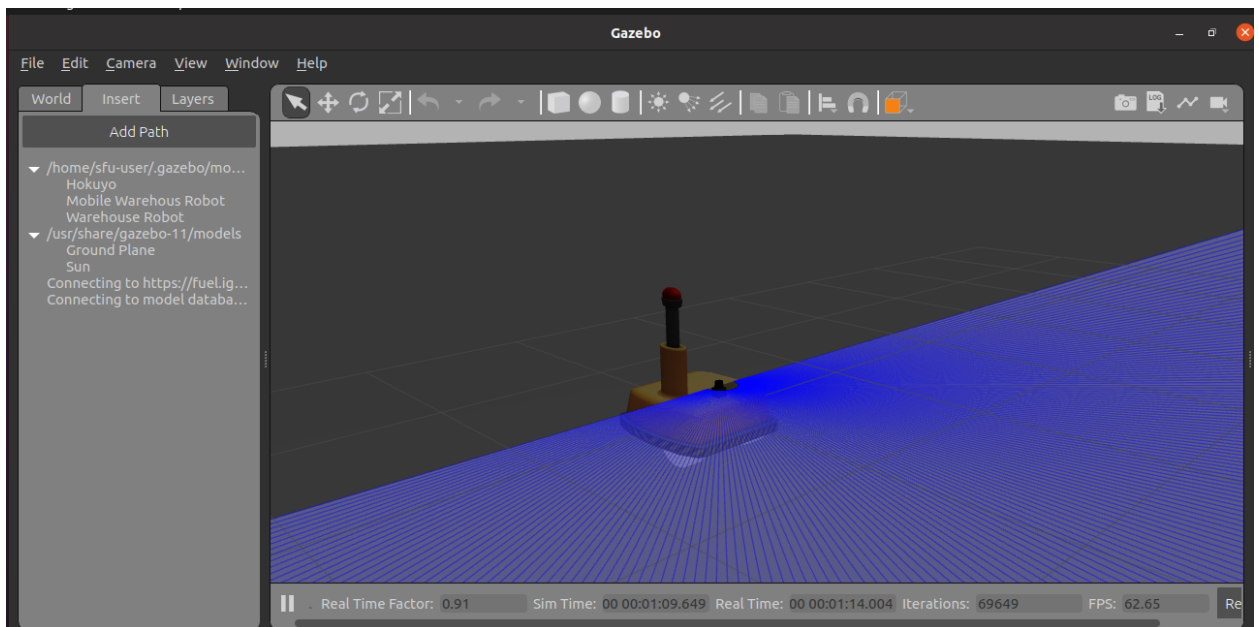
Копируем и вставляем SDF-файл отсюда:
https://drive.google.com/file/d/1BG5hOrfp_qC73uqRE0LMNyRu3bOOijOV/view?usp=sharing.

Теперь давайте запустим Gazebo, чтобы мы могли увидеть нашу модель. Введите следующую команду:

```
gazebo
```

С левой стороны перейдите на вкладку “Insert”.

На левой панели нажмите кнопку “Mobile Warehouse Robot”. Вы увидите складского робота. Вы можете разместить его в любом месте, щелкнув внутри среды.



Теперь поместим свою модель “Mobile Warehouse Robot” в окружающую среду.

3.2.4 Постройка склада

Построим склад для нашего робота, чтобы он мог передвигаться.

Повторим все шаги по созданию `model.config` и `model.sdf` для склада:

```
mkdir -p ~/.gazebo/models/small_warehouse
gedit ~/.gazebo/models/small_warehouse/model.config
gedit ~/.gazebo/models/small_warehouse/model.sdf
```

SDF-файл склада: https://drive.google.com/file/d/1ti9MsasSiAi_yWmY1ai1GNd-dOc-chAU/view?usp=sharing

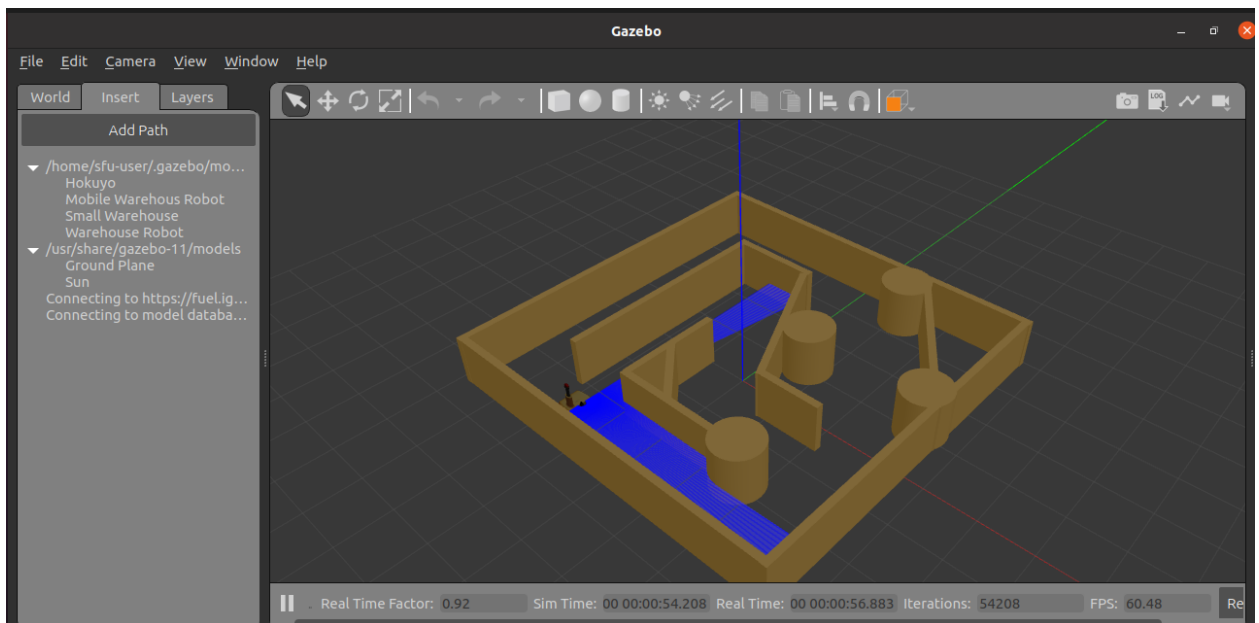
Можно создать свой собственный склад, используя интерфейс Gazebo. Gazebo позволяет добавлять фрагменты в мир, а затем сохранять мир в виде sdf-файла.

Теперь запустим Gazebo, чтобы мы могли увидеть нашу модель.

С левой стороны перейдите на вкладку “Insert”.

На левой панели нажмите кнопку “small_warehouse”, размещаем модель склада.

Клацнув по “Mobile Warehous Robot”, можем разместить нашего робота на модели.



3.3 Запуск робота с помощью ROS 2

Запустим робота и склад с помощью ROS 2.

3.3.1 Создание пакета

Программное обеспечение в ROS 2 организовано в пакеты. Каждый пакет может содержать смесь кода (например, узлы ROS 2), данных, библиотек, изображений, документации и т. Д. Каждая программа, которую вы пишете в ROS 2, должна быть внутри пакета.

Создадим рабочую область:

```
mkdir -p ~/dev_ws/src  
cd ~/dev_ws/src
```

Теперь давайте создадим пакет с именем `warehouse_robot_spawner_pkg`:

```
ros2 pkg create --build-type ament_python
warehouse_robot_spawner_pkg
```

3.3.2 Настройка пакета

Откроем свой `setup.py` файл:

```
gedit setup.py
```

Вставляем файл:

<https://drive.google.com/file/d/1E4X8Pup21QqWmkLl7Ru6clD8KtyE668e/view?usp=sharing>

Теперь нам нужно добавить наши модели:

```
mkdir ~/dev_ws/src/warehouse_robot_spawner_pkg/models
cd ~/.gazebo/models
cp -r small_warehouse
~/dev_ws/src/warehouse_robot_spawner_pkg/models
cp -r mobile_warehouse_robot
~/dev_ws/src/warehouse_robot_spawner_pkg/models
cp -r hokuyo
~/dev_ws/src/warehouse_robot_spawner_pkg/models
cp -r warehouse_robot
~/dev_ws/src/warehouse_robot_spawner_pkg/models
cd ~/dev_ws/src/warehouse_robot_spawner_pkg/models
```

Получили:

```
sfu-user@ubuntu:~/dev_ws/src/warehouse_robot_spawner_pkg/models$ dir
hokuyo  mobile_warehouse_robot  small_warehouse  warehouse_robot
```


Создадим sdf-файл, который обрабатывает генерацию среды Gazebo и склада:

```
mkdir ~/dev_ws/src/warehouse_robot_spawner_pkg/worlds
cd ~/dev_ws/src/warehouse_robot_spawner_pkg/worlds
gedit warehouse.world
```

Вставляем:

<https://drive.google.com/file/d/1bMmQSp5wHweJYkkL2PGvIPCa5AEtuUS/view?usp=sharing>

3.3.3 Создание узла

Нужно создать ROS 2 node, который создаст складского робота и соединит его с ROSE 2.

```
cd
~/dev_ws/src/warehouse_robot_spawner_pkg/warehouse_robot_s
pawner_pkg/
gedit spawn_demo.py
```

Вставляем: <https://drive.google.com/file/d/1wdyZTGGWwHrTM05z9xXU-ygOIoXvk6EY/view?usp=sharing>

3.3.4 Перемещение робота по складу

Теперь сделаем так, что бы патрульный робот патрулировал склад, используя свой лазерный дальномер.

Нужно создать два узла ROS 2. Один узел будет отвечать за оценку текущего состояния робота в окружающей среде (т. е. положение и ориентацию). Другой узел будет посылать роботу команды изменения скорости.

Вставляем:

<https://drive.google.com/file/d/1SZHBcaUpMymqw23PYz2UPyEyX1w3uu9w/view>

https://drive.google.com/file/d/1kwD9wU2fgR_Zv-ZCX2y42gGnT1-rNxb3/view

```
gedit robot_estimator.py
gedit robot_controller.py
```

3.3.5 Создание файла запуска и запуск

```
mkdir ~/dev_ws/src/warehouse_robot_spawner_pkg/launch/
cd ~/dev_ws/src/warehouse_robot_spawner_pkg/launch/
gedit gazebo_world.launch.py
```

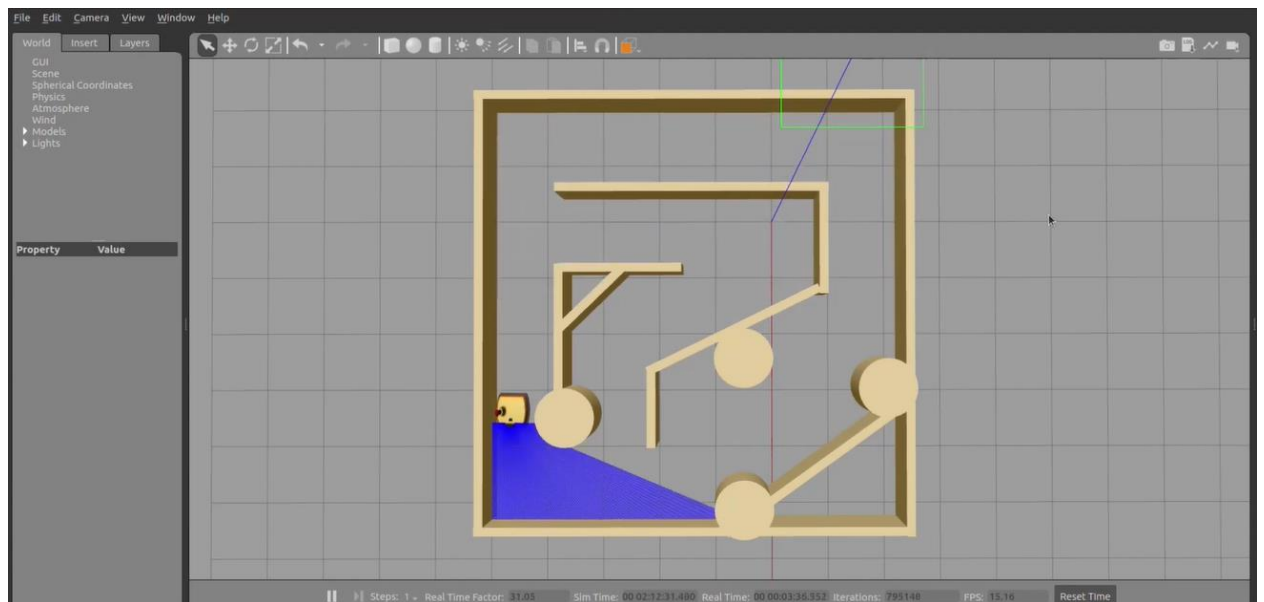
Вставляем:

<https://drive.google.com/file/d/1vdML3njWgcgSTNHqUIoRmFgoOMyrbB2f/view?usp=sharing>

```
cd ~/dev_ws/
colcon build --packages-select warehouse_robot_spawner_pkg
```

Введите следующую команду для запуска:

```
ros2 launch warehouse_robot_spawner_pkg
gazebo_world.launch.py
```



Можем наблюдать робота, патрулирующего склад.

Заключение

В ходе проделанной работы были изучены функционал и возможности системы Gazebo, системные требования, были рассмотрены процессы установки и настройки приложения. Был проведен анализ основных принципов построения моделей физических систем в среде Gazebo, способов описания моделей, определения физических параметров, и создания симуляций. Также рассмотрен демонстрационный проект, иллюстрирующий способы взаимодействия с элементами инфраструктуры ROS 2, а также возможности управления моделями из ROS 2.

По результатам проведённой работы подготовлено руководство по использованию системы Gazebo для моделирования робототехнических комплексов под управлением ROS 2.

Литература

1. Руководство по использованию Gazebo – электронный ресурс URL: <http://gazebosim.org/tutorials?cat=install>
2. Руководство по использованию ROS 2 – электронный ресурс URL: <https://docs.ros.org/en/foxy/Tutorials.html>
3. Как создать робота с помощью Gazebo и ROS 2 – электронный ресурс URL: https://automaticaddison.com/how-to-simulate-a-robot-using-gazebo-and-ros-2/#Install_gazebo_ros_pkgs