

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 ПРАКТИКЕ

Выполнил студент группы 22207:

Устинов Данил Андреевич

Направление подготовки:

Системное и прикладное программное обеспечение

Место прохождения практики:

Кафедра ИМО, ЦИИ

Сроки прохождения практики:

30.05.2022-09.06.2022

Руководитель:

к.т.н., доцент

Богоявленская Ольга Юрьевна

Оценка _____

Дата _____

Содержание

Введение	3
1 Используемые инструменты	4
1.1 Gazebo	4
1.2 SDFormat	5
2 Разработка модели четырехколесного робота	6
2.1 Подготовка	6
2.2 Детализация четырехколесного робота	6
2.3 Добавление модели в мир	9
3 Заключение	11
4 Список литературы	12

Введение

Робототехника — это быстро развивающаяся прикладная наука, занимающаяся разработкой автоматизированных технических систем и являющаяся важнейшей технической основой развития производства. Основной целью создания роботов является замена человека машиной в опасных для его жизни профессиях. Помимо этого робот может быть использован в бытовых, производственных целях и не только. Например, роботы-пылесосы, роботы-сиделки и другие.

Поэтому очень важно уметь создавать модели таких роботов.

Цель: Создать модель четырехколесного робота в среде Gazebo(SDF).

Задачи:

- 1) Научиться взаимодействовать со средой разработки Gazebo;
- 2) Изучить формат описания моделей SDF(Simulation Description Format);
- 3) Разработать модель четырехколесного робота в формате SDF.

1 Используемые инструменты

1.1 Gazebo

Gazebo — это 3D-динамический симулятор с возможностью точного и эффективного моделирования популяций роботов в сложных внутренних и наружных условиях.

Будучи похожим на игровые движки, Gazebo обеспечивает реалистичную визуализацию окружающей среды, включая высококачественное освещение, тени и текстуры. Он может моделировать датчики, такие как лазерные дальномеры, камеры (в том числе широкоугольные) и другие.

Приложение позволяет моделировать динамику и кинематику механизмов роботов (включая взаимодействия с телами внешней среды) и формировать правдоподобные показания виртуальных датчиков.

Gazebo создана на движке с высоким качеством графики и удобным программно-графическим интерфейсом.

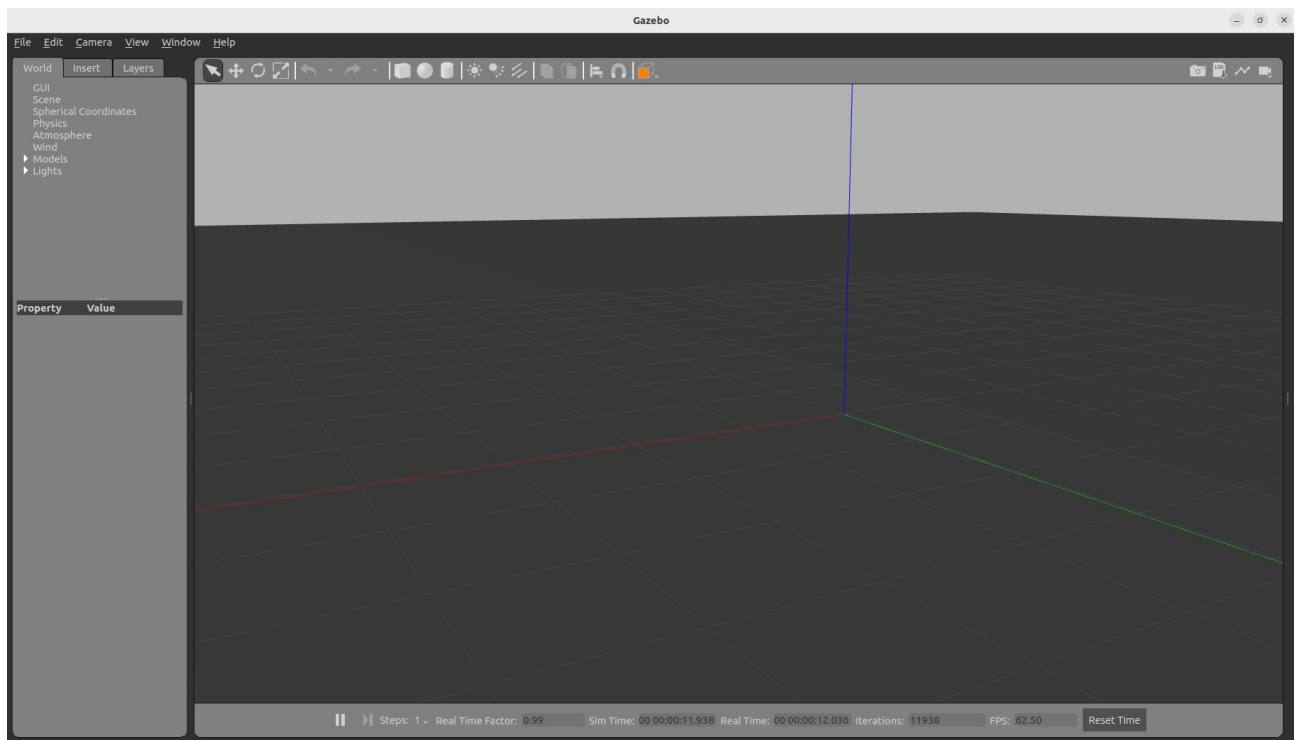


Рис. 1: Среда разработки Gazebo.

Моделирование роботов в среде Gazebo осуществляется с помощью SDF-файла. Однако также для описания моделей можно использовать формат URDF.

1.2 SDFFormat

SDFFormat(Simulation Description Format) — представляет собой формат XML, который описывает объекты и среды для симуляторов роботов, визуализации и управления.

SDF первоначально разрабатывался, как часть симулятора робота Gazebo. С годами SDFFormat стал способным описывать все аспекты роботов, статические и динамические объекты, освещение, рельеф местности и даже физику.

Перед тем, как начать описывать модель или мир необходимо указать, что файл имеет формат XML. После этого используя различные теги SDF-а, можно приступать к описанию объектов.

2 Разработка модели четырехколесного робота

2.1 Подготовка

Для того, чтобы иметь возможность добавлять робота в мир, создадим два файла в директории `home/.gazebo/models`. Первый файл будет иметь расширение `.config` - это файл, необходимый для описания создаваемой модели, ее автора и для ссылки на второй, создаваемый нами файл, расширения `.sdf`, где будем описывать нашего робота.

```
<?xml version="1.0"?>
<model>
  <name>My Robot</name>
  <version>1.0</version>
  <sdf version='1.9'>model.sdf</sdf>

  <author>
    <name>Ustinov Danil</name>
    <email>ustinov@petrsu.ru</email>
  </author>

  <description>
    Model of a four-wheeled robot
  </description>
</model>
```

Рис. 2: Составляющая файла `.config`

2.2 Детализация четырехколесного робота

Вся работа будет происходить в файле `model.sdf`. Внутри тегов `<sdf>` и `<model>` будет описываться наш робот. В `<sdf>` указываем используемую версию — 1.9. Внутри тега `<model>` указываем с помощью `<static>`, что наша модель будет нестатична, для этого внутри тега указываем значение `false`.

Каждая часть нашего робота описывается внутри тега `<link>`. Например, колесо, корпус и т.д. Внутри данного тега указываются различные свойства, которыми должна обладать деталь.

Моей задачей стало визуальное оформление робота и его детализация.

Начнем с колес:

Внутри каждого тега `<link>` с помощью тега `<material>` задается цвет. Цвет можно задать двумя способами:

- С помощью тегов `<ambient>` — цвет объекта, когда тот находится в тени, `<diffuse>` — это основной цвет, который виден при освещении прямыми лучами света, `<emissive>` - излучающий цвет (цвет самоосвещения) и `<specular>` — цвет света зеркального отражения (зеркальное отражение - это тип отражения, характерный для света, отраженного от блестящей поверхности). Внутри каждого тега указываются четыре значения `red/green/blue/alpha`.
- С помощью тега `<script>`, внутри которого указывается название цвета (в теге `<name>`), доступного из встроенного скрипта (`Gazebo/Название цвета`).

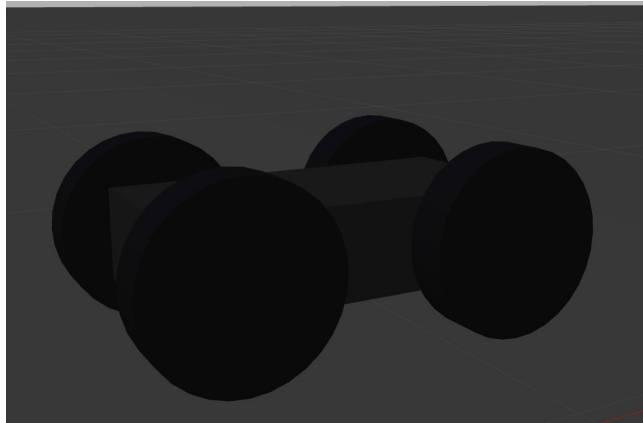


Рис. 3: Задаем цвет колесам и шасси.

Теперь сделаем что-то похожее на диски, благодаря этому при движении робота будет сразу понятно, что модель передвигается.

Для этого я использую четыре куба и одну сферу. Сфера размещается по центру колеса, а кубы, имитирующие спицы, по четырем сторонам света.

Для каждого фрагмента колеса, который описывается в теге `<link>`, задаем соответствующую позицию с помощью тега `<pose>`. Данный тег принимает 6 аргументов. Первые три — это значения по осям x , y и z . Остальные параметры r , p , y (roll, pitch, yaw) отвечают за поворот объекта вокруг соответствующих координатных осей.

Далее с помощью тега `<collision>` задаем объект по которому будет происходить коллизия с окружающим миром. Внутри с помощью тега `<geometry>` описываем геометрическую фигуру, для сферы используется тег `<sphere>` внутри него указывается радиус, для параллелепипеда, соответственно `<box>` и его размеры в трех измерениях.

Теперь с помощью тега `<visual>` задаем объект, по аналогии с тегом `<collision>`, который будет отображаться при добавлении модели в Gazebo. Также задаем цвет нашему объекту. Объекты из тега `<collision>` и `<visual>`, соответственно, будут совпадать.

```

<link name="left_sphere">
  <pose>0.15 0.13 0.1 0 0 0</pose>
  <collision name="collision">
    <geometry>
      <sphere>
        <radius>0.03</radius>
      </sphere>
    </geometry>
  </collision>
  <visual name="visual">
    <material>
      <ambient>0.9 0.1 0.1 1</ambient>
      <diffuse>0.9 0.1 0.4 1</diffuse>
      <specular>0 0 0 0</specular>
      <emissive>0 0 0 1</emissive>
    </material>
    <geometry>
      <sphere>
        <radius>0.03</radius>
      </sphere>
    </geometry>
  </visual>
</link>

```

Рис. 4: Пример кода для сферы

Таким образом создаем четыре куба и сферу. Располагаем их по соответствующим координатам и с помощью тега `<joint>` соединяем их с колесом. Данному тегу можно задать тип соединения и имя. Тип соединения может быть разным, например, `revolute`, который использовался для соединения колес и шасси, помимо него также есть `revolute2`, `gearbox`, `screw`, `fixed` и другие. В нашем случае будем использовать `fixed`, так как нужно жестко зафиксировать объекты вместе, исключив любое движение. Таким образом повторяем все эти действия для оставшихся трех колес.

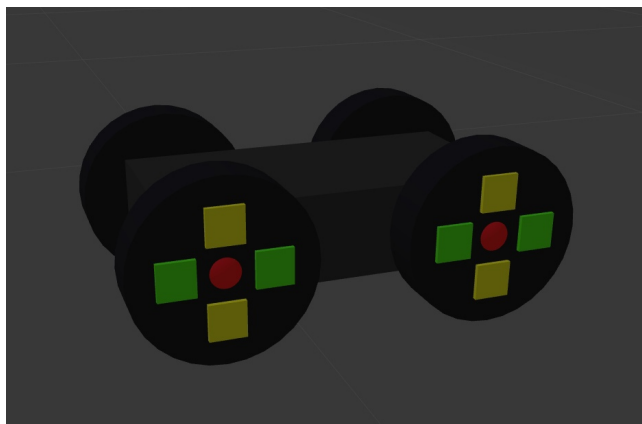


Рис. 5: Демонстрация колес

Теперь добавим параллелепипед чуть меньших размеров, чем исходный, для имитации

отсека под двигатель и аккумуляторную батарею. По аналогии с предыдущими объектами создаем его, используя соответствующие теги. И соединяем с шасси, также используя тип соединения fixed.

```
<link name='trunk'>
  <pose>-0.075 0 0.175 0 0 0</pose>
  <collision name="collision">
    <geometry>
      <box>
        <size>0.25 .2 .05</size>
      </box>
    </geometry>
  </collision>
  <visual name='visual'>
    <material>
      <script>
        <name>Gazebo/Grey</name>
      </script>
    </material>
    <geometry>
      <box>
        <size>0.25 .2 .05</size>
      </box>
    </geometry>
  </visual>
</link>

<joint type="fixed" name="trunk_with_chassis">
  <pose>0 0 -0.025 0 0 0</pose>
  <child>trunk</child>
  <parent>chassis</parent>
</joint>
```

Рис. 6: Пример кода для отсека

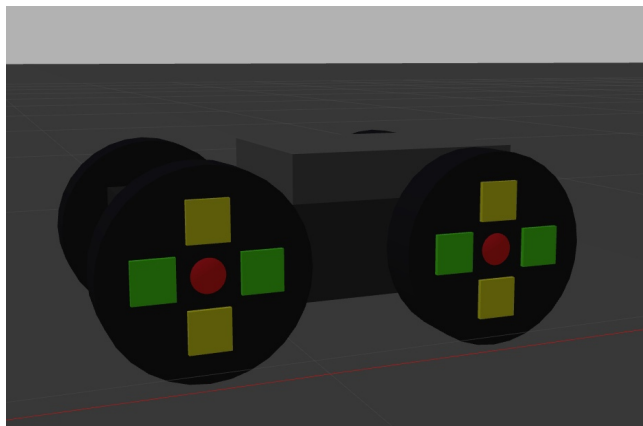


Рис. 7: Итоговый результат

2.3 Добавление модели в мир

Для того, чтобы проверить, что робот корректно отображается в мире. Открываем Gazebo, переходим во вкладку Insert, там в соответствующей директории должно отоб-

разиться имя нашего робота, которое было указано в файле .config. Щелкаем на него и теперь мы можем расположить модель в любом месте нашего мира.

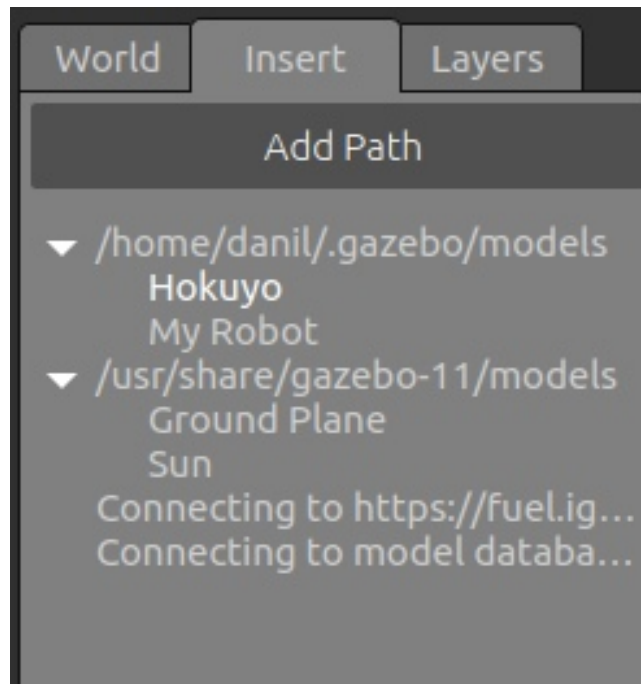


Рис. 8: Добавление модели в мир

Для проверки того, что модель может двигаться, воспользуемся встроенным интерфейсом Apply Force. Для этого зададим силу толчка и его направление.

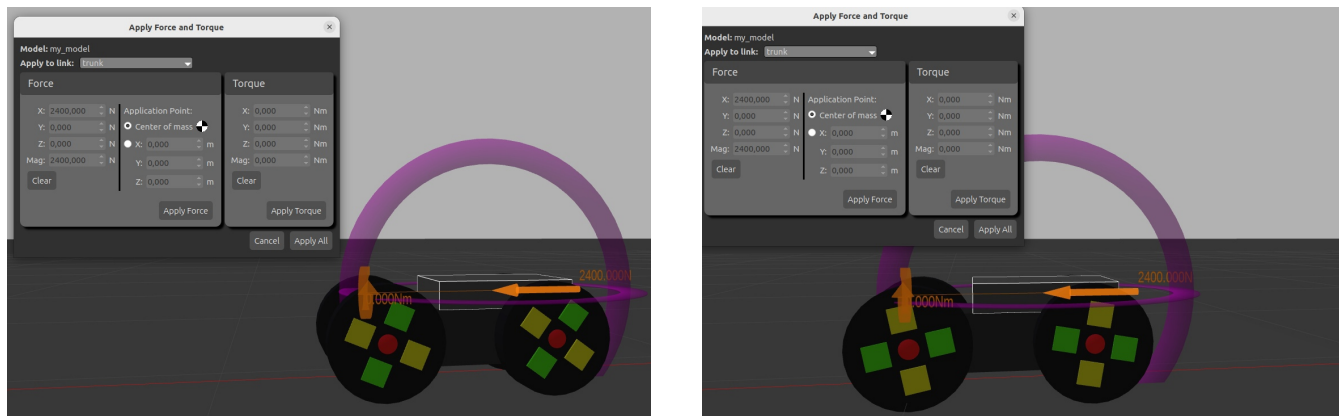


Рис. 9: Демонстрация движения робота

Далее второй человек из моей группы реализует управление роботом через терминал.

3 Заключение

В ходе работы были выполнены следующие задачи:

1. Получены навыки работы со средой разработки Gazebo;
2. Изучен формат описания моделей SDF(Simulation Description Format);
3. С помощью формата SDF разработана детализированная модель четырехколесного робота.

В итоге была создана модель четырехколесного робота с возможностью передвижения.

4 Список литературы

1. SDFFormat - документация[Электронный ресурс] / URL : <http://sdformat.org/spec>.
(дата обращения: 03.06.2022)
2. Сxem.net [Электронный ресурс] / URL : <https://cxem.net/software/gazebo.php>. (дата обращения: 06.06.2022)
3. Gazebo - tutoriales [Электронный ресурс] / URL : <https://classic.gazebosim.org/tutorials>.
(дата обращения: 06.06.2022)
4. Вводное руководство по работе с Gazebo [Электронный ресурс] / URL :
<https://pandia.ru/text/80/149/14173.php>. (дата обращения: 03.06.2022)