

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ПЕТРОЗАВОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
ИНСТИТУТ МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
09.03.04 – ПРОГРАММНАЯ ИНЖЕНЕРИЯ

ОТЧЕТ О ПРОХОЖДЕНИИ ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ

Выполнил студент группы 22207
Костин Данила Александрович

Направление подготовки:
Системное и прикладное программное обеспечение

Место прохождения практики:
Кафедра ИМО, ЦИИ

Сроки прохождения практики:
30.05.22-09.06.22

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

Оценка _____

Дата _____

Содержание

Введение	3
1 Инструменты для разработки	4
1.1 Среда разработки Gazebo	4
1.2 SDF формат описания робота	4
2 Разработка модели четырёхколёсного робота	5
2.1 Подготовка мира	5
2.2 Основа четырёхколёсного робота	5
2.3 Реализация движения через терминал	9
Заключение	11

Введение

В настоящее время робототехника является одним из самых востребованных направлений в развитии автоматизированных технологических систем и активно используется в таких областях, как медицина, телекоммуникация, военное и промышленное дело, а также образование.

Моделирование давно уже стало неотъемлемой частью разработки какой бы то ни было технологии или продукта. Оно позволяет сберечь время и деньги, а в некоторых случаях даже позволяет не подвергать сам продукт, людей и окружающую среду различного рода опасностям. В сфере робототехники актуальность моделирования преобладает еще в большей степени ввиду высокого уровня сложности технических систем.

Развитие робототехники и технологий моделирования привело к появлению целого класса программного обеспечения – робототехнических симуляторов, в которых происходит разработка и тестирование созданной модели в условиях, приближённых к условиям реального мира.

Цель практики: разработка модели четырёхколёсного четырехколесного робота в среде Gazebo с использованием формата описания SDF.

Задачи учебной практики:

1. Овладеть навыками работы в *Gazebo* - среде моделирующий взаимодействие робота с физическим миром.
2. Изучить формат *SDF* (*SimulationDescriptionFormat*)
3. Используя возможности формата *SDF* и среды *Gazebo*, разработать модель робот-тележки.

1 Инструменты для разработки

1.1 Среда разработки Gazebo

Gazebo представляет собой среду для симулирования работы виртуальных роботов с различными сенсорами в окружении всевозможных объектов. Приложение состоит из графической части и части по имитированию взаимодействия твердых объектов, позволяя моделировать динамику и кинематику механизмов роботов.

Gazebo состоит из двух разделов: Gazebo-сервер и Gazebo-клиент. Моделирование выполняется на сервере Gazebo, который действует как программно-аппаратная часть сервиса. Графический интерфейс пользователя является клиентской стороной пользовательского интерфейса к программно-аппаратной части сервиса и действует как клиент Gazebo.

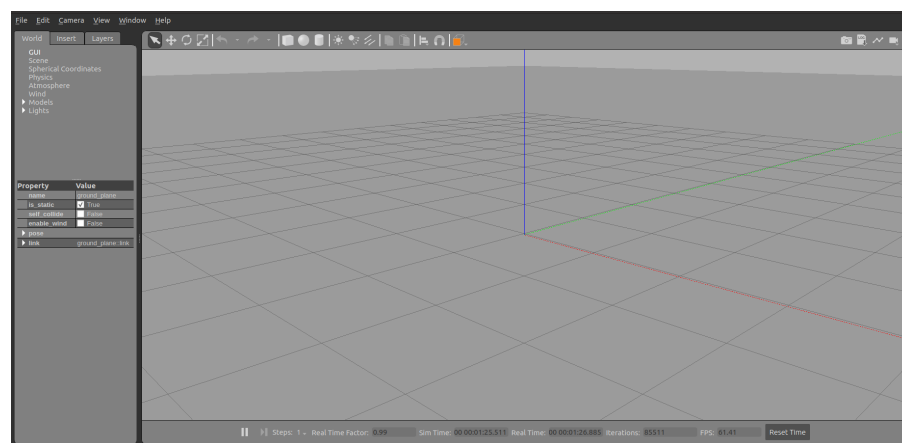


Рис. 1 – Интерфейс среды Gazebo

Gazebo имеет свой редактор моделей, позволяющий без программирования создавать трехмерные объекты. Он также включает библиотеку уже готовых моделей. Однако среда позволяет использовать модели, созданные в форматах *SDF*, *URDF*.

1.2 SDF формат описания робота

SDF, или Simulation Description Format, - это формат XML, который описывает объекты и среды для симуляторов роботов, визуализации и управления. Первоначально он разрабатывался как часть симулятора робота *Gazebo*. SDF способен описывать все аспекты роботов, статических и динамических объектов, освещения, рельефа местности и даже физики.

Последняя версия SDF - 1.9, которая включает в себя множество элементов или тегов, описывающих какой-либо элемент, будь то мир, будь то робот. Каждый файл, описыва-

ющий объект, должен начинаться с объявления, что файл имеет формат XML, и только после этого возможно добавление различных тегов, определенных форматом описания SDF.

2 Разработка модели четырёхколёсного робота

2.1 Подготовка мира

Первыми шагами перед созданием модели было создание настраиваемого мира. Отдельный мир создаётся для удобства и скорости запуска будущей модели робота в среде Gazebo.

Для этих целей было создано два файла формата XML:

- *world.world* – файл с описанием SDF, подключающий к создаваемому миру плоскость поверхности, а также саму модель робота – *new_robot*.
- *start.launch* – файл настроек мира, в котором указан аргумент *paused*, установленный на *true* для остановки течения времени в симуляции при запуске, а также путь к файлу *world.world*.

Для запуска мира с заданными параметрами используется утилита *roslaunch* с параметрами пакета *catkin*, где были созданы файлы мира, а также сам файл настройки мира, расширения «.launch».

Сделав тестовый запуск данной команды через терминал мир, запустится Gazebo с теми настройками мира, указанными в файле *world.world*, но так как в данный момент не была встроена модель робота в папку, где Gazebo производит локальный поиск моделей, то мир будет пуст.

2.2 Основа четырёхколёсного робота

Для создания модели необходимо создать два файла:

- *model.sdf* – основной файл, содержащий описание составных частей робота и их соединения с помощью тегов SDF.
- *model.config* – файл, обозначающий название модели, отображаемое, например, в панели добавления моделей в Gazebo, а также ссылается на файл с описанием модели (то есть на файл формата «.sdf»), с указанием версии SDF.

Рассмотрим подробнее этапы построения модели – файл `model.sdf`.

Как уже было ранее отмечено, любой файл описания робота должен начинаться с объявления того, что его формат - XML. Вся описывающая составные элементы робота часть файла начинается с базового тега – `< sdf >`.

В данном базовом теге вправе описываться сразу несколько моделей, для их разграничения используется тег `< model >`

Все теги, находящиеся внутри тега `< model >` относятся к конкретной модели. Основными тегами при описании робота были:

- `< static >` – данный тег отвечает за неподвижность текущей модели. Очевидно, что для подвижного робота данный тег должен принимать *false*.
- `< link >` – тег, внутри которого происходит описание отдельного элемента робота, например, колеса или детали корпуса, а также указание разлнчных свойств объекта.
- `< pose >` – тег, производящий размещение текущего элемента в трехмерном пространстве, и параметры поворота вокруг, соответственно, осей *x*, *y*, *z* – крен (roll), тангаж (pitch) и рыскание (yaw), визуализация которых представлена на рисунке 2.

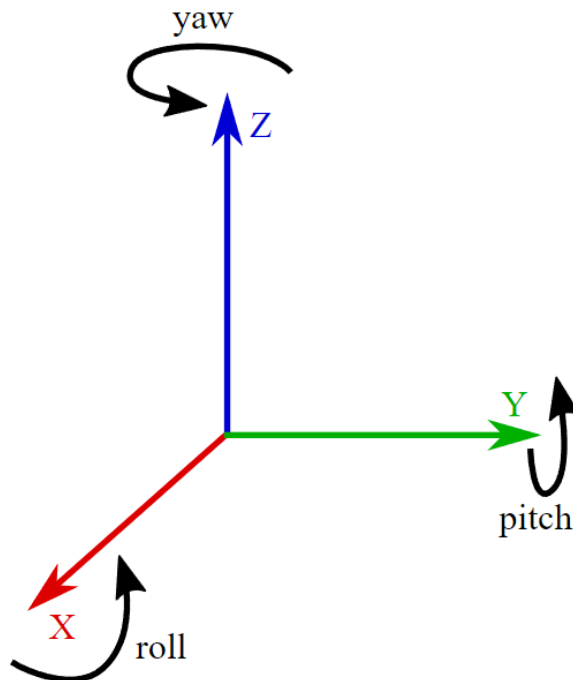


Рис. 2 – Тег `pose` - крен, тангаж и рыскание

- `< collision >` – задаёт форму, по которой будет происходить коллизия элемента с миром. Например, создаваемый робот может иметь сложную форму, и при симуля-

ции данного робота в среде Gazebo каждый интервал проверки коллизии робота будет производится много вычислений, производительность Gazebo будет падать. Для уменьшения количества проверок используется геометрические фигуры, имеющие меньшее число граней, но покрывающие весь объем робота.

- `< visual >` – задаёт форму объекта, которая будет визуально отображена при запуске модели в Gazebo.
- Теги `< box >` и `< cylinder >` – соответственно трехмерные геометрические фигуры параллелепипеда – корпус, – и цилиндра – колесо.

Перейдём к практической части описания построения робота с введением кода.

Ниже представлен листинг основной части робота - корпуса, но котором будут закреплены колёса.

```
1  <link name="chassis">
2    <pose>0 0 0.1 0 0 0</pose>
3    <collision name="collision">
4      <geometry>
5        <!-- Parallelepiped 0.4x0.2x0.1 -->
6        <box>
7          <size>0.4 0.2 0.1</size>
8        </box>
9      </geometry>
10   </collision>
11   <visual name="visual">
12     <geometry>
13       <box>
14         <size>0.4 0.2 0.1</size>
15       </box>
16     </geometry>
17   </visual>
18 </link>
```

Листинг 1 – Корпус

Из данного листинга 1 видно, что корпус, называемый *chassis* представляет собой параллелепипед, его коллизией также является параллелепипед того же размера.

После задания основного корпуса, необходимо добавить 4 колеса, крепящиеся к корпусу и возможностью их вращения. Это можно сделать, используя специальный тег, но об

этом позже. Сначала необходимо создать модель колеса. Простое колесо представимо в виде цилиндра, повернутого на 90 градусов, радиуса 0.1 и высотой 0.05.

```
1  <link name="left_wheel">
2    <pose>0.15 0.13 0.1 0 1.5707 1.5707</pose>
3    <collision name="collision">
4      <geometry>
5        <cylinder>
6          <radius>.1</radius>
7          <length>.05</length>
8        </cylinder>
9      </geometry>
10   </collision>
11   <visual name="visual">
12     <geometry>
13       <cylinder>
14         <radius>.1</radius>
15         <length>.05</length>
16       </cylinder>
17     </geometry>
18   </visual>
19 </link>
```

Листинг 2 – Левое переднее колесо

Как видно из листинга 2, строка 2, колесо повернуто по крену на 0, по тангажу и рысканию на $\Pi/2$. Аналогично были созданы 3 остальных колеса, с помощью `< pose >` они были поставлены на свои места.

Теперь осталось рассмотреть последний, вероятно, самый важный тег для связи элементов робота с помощью определенного типа соединения. Такой тег – `< joint >`. Для соединения колёс с корпусом создаётся 4 тега `< joint >`, использующих атрибут `type`, принимающий одно из множества значений типа соединений. Подходящей тип соединения для колеса с корпусом – тип *revolute*.

```
1  <joint type="revolute" name="left_wheel_hinge">
2    <pose>0 0 -0.03 0 0 0</pose>
3    <child>left_wheel</child>
4    <parent>chassis</parent>
5    <axis>
6      <xyz>0 0 1</xyz>
7    </axis>
```

Листинг 3 – Соединение левого переднего колеса и корпуса

В результате, повторив запуск мира, как представлено в пункте 2.1, Gazebo запустится с уже загруженной и вставленной моделью. Данная модель – база для последующего улучшения вторым человеком моей группы по данной теме. На рисунке 3 приведен скриншот получившейся модели, запущенной в Gazebo. Отметим, что она ещё не текстурирована, далее она будет улучшена.

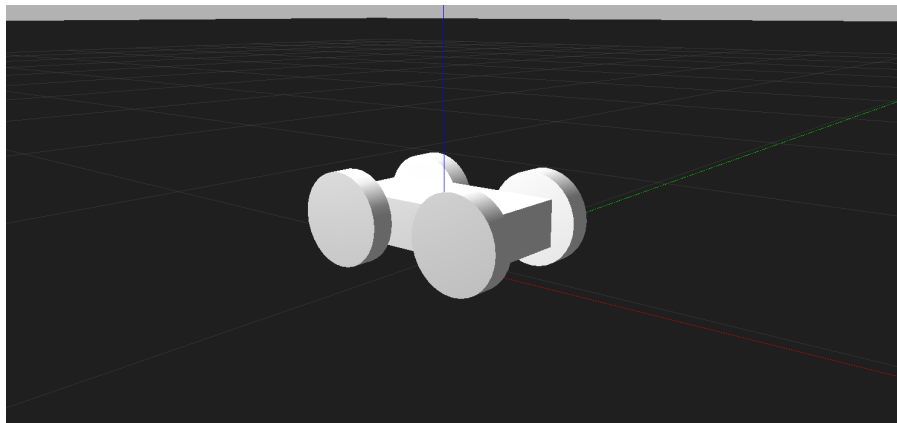


Рис. 3 – База четырёхколёсного робота

2.3 Реализация движения через терминал

Движение робота в среде Gazebo может реализовываться различными способами, например, с помощью функции интерфейса *ApplyForce* – выбрав в открывшемся окне направление толчка и силу. Такой способ годится лишь для тестирования полученной модели на устойчивость, но не годится к более длительным испытаниям передвижения.

Другой способ реализации простого движения четырёхколёсного робота является использование специального плагина для управления перемещением через терминал.

Вернёмся к файлу `model.sdf`. Подключим в него плагин при помощи тега `< plugin >`.

```

1 <plugin filename="libgazebo_ros_diff_drive.so" name="differential_drive_controller">
2   <alwaysOn>true</alwaysOn>
3   <updateRate>10</updateRate>
4   <leftJoint>left_wheel_hinge</leftJoint>
5   <rightJoint>right_wheel_hinge</rightJoint>
6   <wheelSeparation>0.25</wheelSeparation>
7   <wheelDiameter>0.2</wheelDiameter>
8   <wheelTorque>7</wheelTorque>

```

```

9      <commandTopic>robot/cmd_vel</commandTopic>
10     <odometryTopic>robot/odom_diffdrive</odometryTopic>
11     <odometryFrame>odom</odometryFrame>
12     <robotBaseFrame>chassis</robotBaseFrame>
13 </plugin>

```

Листинг 4 – Подключение плагина

Разберёмся с листингом 4: тег `< plugin >` от начала и до его закрытия содержит необходимые для работы плагина параметры, например, плагину важно знать соединения колес с корпусом, поэтому в параметрах `< leftJoint >` и `< rightJoint >` указываются названия соответствующих тегов `joint`. `< wheelTorque >` означает крутящий момент колеса, соответственно, чем он выше, тем быстрее робот передвигается. Также для возможности будущего обращения для управления моделью необходимо подключить данный плагин к топику через тег `< commandTopic >`.

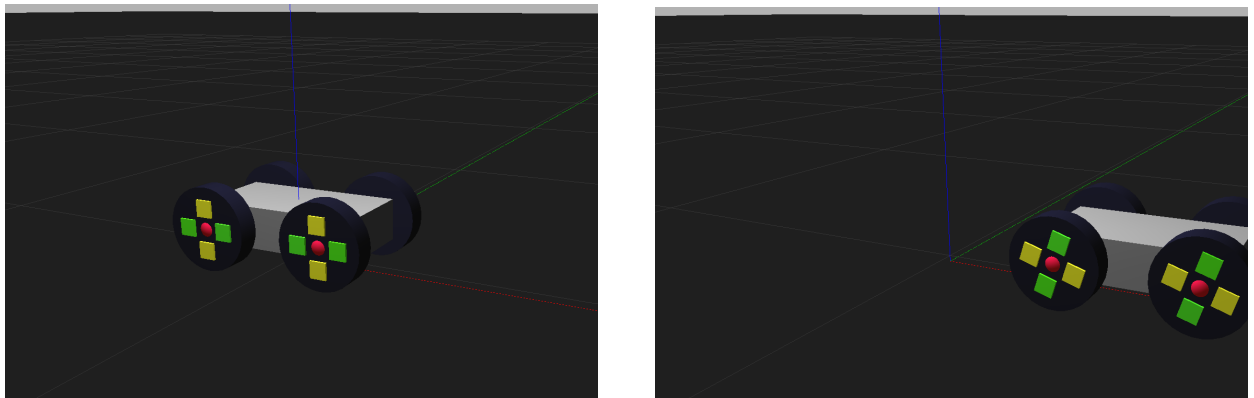


Рис. 4 – Робот начал движение

После производится запуск Gazebo по алгоритму из пункта 2.1. В другом терминале происходит вызов команды `roslaunch turtlesim turtle_teleop_key/turtle1/cmd_vel := /robot/cmd_vel`. Нажатие на стрелки на клавиатуре приводит в движение робота в соответствующем направлении.

К последнему этапу наш четырёхколёсный робот представляет собой текстурированную и более детализированную модель, представленную на рисунке 5.

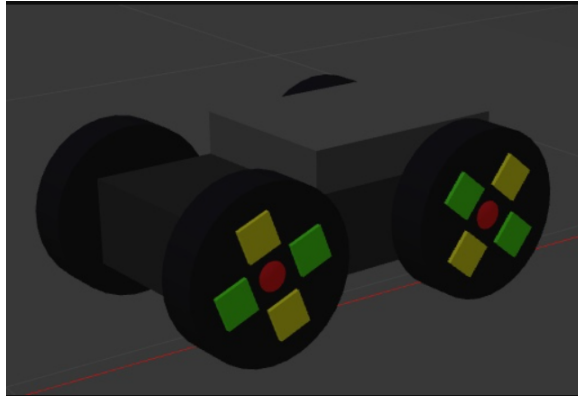


Рис. 5 – Готовая модель четырёхколёсного робота

Заключение

В результате работы была создана модель четырёхколёсного робота.

Были выполнены поставленные задачи:

- Была изучена среда для разработки роботов – Gazebo;
- Был изучен формат описания роботов SDF;
- На основе SDF была разработана детализированная модель робота;
- Дополнительной рассмотренной возможностью среды Gazebo стала реализация простого передвижения робота.

Список литературы

1. МОДЕЛИРОВАНИЕ РОБОТОТЕХНИЧЕСКИХ СИСТЕМ: И.А. Бжихатлов, Санкт-Петербург, 2018 г.
2. SDF format – официальный сайт разработчика SDF [Электронный ресурс]. - Режим доступа: <http://sdformat.org/>, свободный - (07.06.2022).
3. Официальный сайт Gazebo [Электронный ресурс]. - Режим доступа: <https://classic.gazebosim.org/>, свободный - (07.06.2022).
4. “Вводное руководство по работе с Gazebo” [Электронный ресурс]. - Режим доступа: <https://pandia.ru/text/80/149/14173.php>, свободный - (07.06.2022).
5. Gazebo plugin for ROS notes [Электронный ресурс]. - Режим доступа: <https://programmersought.com/article/36814336604/>, свободный - (07.06.2022).