

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:
Использование URDF в Gazebo для совмещения с ROS

Выполнил студент группы 22207

Лашко Григорий Максимович

Направление подготовки:

**Системное и прикладное про-
граммное обеспечение**

Место прохождения практики:

Кафедра ИМО, ЦИИ

Сроки прохождения практики:

30.05-09.06

Руководитель практики:

**к.т.н, доцент Богоявленская Ольга
Юрьевна**

Оценка _____

Дата _____

Петрозаводск 2022 г.

Содержание

1	GAZEBO	3
1.1	Работа с Gazebo	3
2	Robot Operating System (ROS)	4
2.1	Что дает готовая платформа	5
2.2	Gazebo и ROS	6
3	Что такое SDF	7
3.1	Обзор	8
3.2	Компоненты SDF-моделей	8
4	Что такое URDF	10
4.1	Обзор преобразования в Gazebo	10
5	Конвертирование URDF в SDF	14
5.1	При каких обстоятельствах необходимо конвертировать между URDF и SDF?	14
5.2	URDF в SDF	14
5.3	SDF в URDF	14
6	Заключение	16
7	Список литературы	17

1 GAZEBO

Gazebo - это программный пакет, моделирующий взаимодействие робота или даже популяции роботов с физическим миром. Детально описав робота можно тестировать как работу алгоритмов, так и физическую реализацию робота в виртуальной среде, до того, как собирать его в железе. Физический движок использующийся в Gazebo позволяет учитывать такие тонкости взаимодействия робота со средой как трение, мощность моторов, имитация сенсоров и камеры и т. д.



Рис. 1: logo GAZEBO

1.1 Работа с Gazebo

Сначала в xml-формате создается модель робота, каждая функциональная часть (база, колеса, камера и т. д.) описывается отдельно, затем эти части соединяются вместе в модель робота.

К таким частям робота, как камеры, моторы, сенсоры и т. д. необходимо подключать плагины, которые пишутся на C++ и описывают функциональность этих частей. Например, простейший плагин на мотор прикладывает момент силы к некоторой оси при получении соответствующей команды (например, ROS-сообщения).

Можно подключать плагины не только непосредственно к модели и сенсорам, но и к миру целиком.

2 Robot Operating System (ROS)

Robot Operating System (ROS) - это гибкая платформа (фреймворк) для разработки программного обеспечения роботов. Это набор разнообразных инструментов, библиотек и определенных правил, целью которых является упрощение задач разработки ПО роботов.

Создание действительно надежного, универсального программного обеспечения для роботов чрезвычайно сложная задача. С точки зрения робота, проблемы, которые кажутся тривиальными для людей, часто требуют очень сложных технических решений. Часто разработка такого решения не под силу одному человеку.

ROS была создана, чтобы стимулировать совместную разработку программного обеспечения робототехники. Каждая отдельная команда может работать над одной конкретной задачей, но использование единой платформы, позволяет всему сообществу получить и использовать результат работы этой команды для своих проектов.



Рис. 2: logo ROS

2.1 Что дает готовая платформа

- **Повторное использование программных модулей.** Разработанный программный модуль, легко запускается и переиспользуется в любом другом приложении. Вопросы установки зависимостей и других библиотек хорошо проработан и автоматизирован.
- **Готовый протокол коммуникации.** Основная проблема комплексных робототехнических систем, это решение задач коммуникации в рамках одного приложения. Для решения этих задач ROS содержит все необходимые утилиты. Любой программный модуль может быть представлен как отдельный процесс, взаимодействующий с другими процессами по сетевому протоколу. Такой подход позволяет создавать независимые и простые в повторном использовании программные модули, которые возможно запустить/остановить/модифицировать на любом устройстве.
- **Развитость средств разработки и отладки.** ROS предоставляет готовые инструменты для отладки, инструмент 2D-визуализации (rqt), и инструмент 3D-визуализации (RViz), инструмент 3D симуляции (Gazebo).
- **Активное и открытое сообщество.** Активное и открытое сообщество Сообщества разработчиков робототехники из академического мира и промышленности, были относительно закрытыми до последнего времени. Но сейчас мы видим активное, и главное открытое сотрудничество всех участников. В центре этого изменения – программная платформа с открытым исходным кодом. В случае ROS существует более 5000 пакетов, которые были разработаны и выложены в общий доступ. Описание этих пакетов, инструкций и другой полезной информации – превышает 18 000 Wiki страниц.

- **Собственная экосистема.** Вокруг ROS сформирована собственная экосистема (по аналогии с платформами Android и Apple). В ней существуют разработчики аппаратных платформ, разработчики программных модулей, энтузиасты и компании-производители промышленного оборудования, единое место распространения и хранения готовых модулей, тысячи станций документации. Все участники взаимодействуют и работают в рамках единой платформы.

2.2 Gazebo и ROS

Gazebo не обязателен ROS для функционирования, в нем успешно можно работать вообще без ROS'а. Однако это менее удобно. Именно наличие некоторого количества уже написанных плагинов взаимодействующих методами ROS (сообщения, сервисы, параметры) и интегрируют Gazebo в ROS. ROS Indigo работает с Gazebo 2.2, ROS Jade работает с Gazebo 5.0.

3 Что такое SDF

SDF (Simulation Description Format) иногда сокращенно SDF, представляет собой формат XML, который описывает объекты и среды для симуляторов роботов, визуализации и управления. Первоначально разработанный как часть симулятора робота Gazebo, SDFormat был разработан с учетом научных приложений роботов. С годами SDFormat стал стабильным, надежным и расширяемым форматом, способным описывать все аспекты роботов, статических и динамических объектов, освещения, рельефа местности и даже физики.

Роботы Вы можете точно описать все аспекты робота, используя SDFormat, будь то простое шасси с колесами или гуманоид. В дополнение к кинематическим и динамическим атрибутам для робота можно определить датчики, свойства поверхности, текстуры, трение в суставах и многие другие свойства. Эти функции позволяют использовать SDFormat как для моделирования, визуализации, планирования движения, так и для управления роботом.

Среды Моделирование требует богатой и сложной среды, в которой модели существуют и взаимодействуют. SDFormat предоставляет средства для определения широкого спектра сред. Несколько огней могут быть включены в окружающую среду, местность (вымышленную или основанную на DEM), улицы из OpenStreetMaps и любую модель, предоставленную из магазина реквизита (в онлайн-хранилище 3D-моделей).

3.1 Обзор

Модели SDF могут варьироваться от простых форм до сложных роботов. Он ссылается на `<model>` тег SDF и, по сути, представляет собой набор ссылок, соединений, объектов столкновения, визуальных эффектов и плагинов. Создание файла модели может быть затруднено в зависимости от сложности желаемой модели. На этой странице вы найдете несколько советов о том, как создавать свои модели.

3.2 Компоненты SDF-моделей

- **Ссылки:** Ссылка содержит физические свойства одного тела модели. Это может быть колесо, или звено в совместной цепи. Каждая ссылка может содержать множество коллизионных и визуальных элементов. Попробуйте уменьшить количество ссылок в ваших моделях, чтобы повысить производительность и стабильность. Например, модель стола может состоять из 5 звеньев (4 для ножек и 1 для верха), соединенных через суставы. Однако это слишком сложно, тем более что суставы никогда не будут двигаться. Вместо этого создайте таблицу с 1 ссылкой и 5 элементами столкновения.
 - **Столкновение:** элемент столкновения инкапсулирует геометрию, которая используется для проверки столкновений. Это может быть простая форма (которая предпочтительнее) или треугольная сетка (которая потребляет больше ресурсов). Ссылка может содержать много коллизионных элементов.
 - **Визуальный:** Визуальный элемент используется для визуализации частей ссылки. Ссылка может содержать 0 или более визуальных элементов.
 - **Инерционный:** Инерционный элемент описывает динамику

ческие свойства звена, такие как масса и матрица инерции вращения.

- **Датчик:** Датчик собирает данные из мира для использования в плагилах. Ссылка может содержать 0 или более датчиков.
- **Light:** Элемент `light` описывает источник света, прикрепленный к ссылке. Ссылка может содержать 0 или более огней.
- **Суставы:** сустав соединяет два звена. Отношения между родителями и детьми устанавливаются вместе с другими параметрами, такими как ось вращения и пределы соединения.
- **Плагины:** Плагин - это общая библиотека, созданная третьей стороной для управления моделью.

4 Что такое URDF

URDF или Объединенный Формат Описания Робототехники, является спецификацией XML, используемой в академической среде и промышленности для моделирования систем мультител, такого как руки роботов-манипуляторов для производства сборочных конвейеров и аниматронных роботов для парков развлечений. Хотя URDFS являются полезным и стандартизированным форматом в ROS, им не хватает многих функций и они не были обновлены для удовлетворения развивающихся потребностей робототехники. URDF может указывать только кинематические и динамические свойства одного робота в изоляции. URDF не может указать позу самого робота в мире. Это также не универсальный формат описания, поскольку он не может указывать петли соединения (параллельные связи), и ему не хватает трения и других свойств. Кроме того, он не может указывать вещи, которые не являются роботами, такие как огни, карты высот и т. Д.

На стороне реализации синтаксис URDF нарушает правильное форматирование с интенсивным использованием атрибутов XML, что, в свою очередь, делает URDF более негибким. Также нет механизма обратной совместимости.

4.1 Обзор преобразования в Gazebo

Есть несколько шагов, чтобы заставить робота URDF правильно работать в Gazebo. Ниже приведен обзор шагов, которые затем подробно рассматриваются в остальной части этого урока:

Элемент<gazebo> <gazebo>Элемент является расширением URDF, используемым для указания дополнительных свойств, необходимых для целей моделирования в Gazebo. Это позволяет указать свойства, найденные в формате SDF, которые не включены в формат URDF. Ни один из элементов в <gazebo>элементе не требует-

ся, потому что значения по умолчанию будут автоматически включены. Существует три различных типа `<gazebo>` элементов - один для `<robot>` тега, один для `<link>` тегов и один для `<joint>` тегов. В этом уроке мы обсудим атрибуты и элементы внутри каждого типа `<gazebo>` элемента.

Для преобразования требуется:

- Добавьте `<gazebo>` элемент для каждого `<link>`
- Добавьте `<gazebo>` элемент для каждого `<joint>`
- Добавить `<gazebo>` элемент для `<robot>` элемента
- Добавьте `<link name="world"/>` ссылку, если робот должен быть жестко привязан к миру

`<collision>` и `<visual>` элементы Эти теги работают практически так же в Gazebo, как и в Rviz. Важно указать оба, потому что в отличие от некоторых приложений ROS, Gazebo не будет использовать ваши `<visual>` элементы в качестве `<collision>` элементов, если вы явно не укажете `<collision>` элемент. Вместо этого Gazebo будет рассматривать вашу ссылку как "невидимую" для лазерных сканеров и проверки столкновений. **Использование правильных цветов и текстур** Стандартный URDF может указывать цвета с помощью тега, например, в RRBot: `<material name="orange"/>`

С оранжевым цветом, определенным отдельно, например, в файле материалы.хacro:

```
<material name="orange">
  <color rgba="{255/255}_ {108/255}_ {10/255}_ 1.0" />
</material>
```

К сожалению, этот метод указания цветов ссылок не работает в Gazebo, поскольку он использует сценарии материалов OGRE для раскраски и текстурирования ссылок. Вместо этого для каждой ссылки должен быть указан тег материала Gazebo, например:

```
<gazebo reference="link1">
  <material>Gazebo/Orange</material>
</gazebo>
```

<inertial> Элемент Чтобы физический движок Gazebo работал правильно, <inertial>элемент должен быть задокументирован на странице элемента ссылки URDF. Чтобы ссылки не игнорировались в Gazebo, их масса должна быть больше нуля. Кроме того, ссылки с нулевым основным моментом инерции (i_{xx} , i_{yy} , i_{zz}) могут привести к бесконечному ускорению при любом конечном приложении крутящего момента.

Определение правильных значений для каждой ссылки требуется для получения точных физических приближений в Gazebo. Это может быть выполнено путем проведения различных измерений деталей роботов или с помощью программного обеспечения CAD, такого как Solidworks, которое включает функции для аппроксимации этих значений. Для начинающих вы также можете просто составить значения.

Пример элемента инерции из первой ссылки RRBot:

```
<inertial>
  <origin xyz="0_0_0_{height1/2}" rpy="0_0_0"/>
  <mass value="1"/>
  <inertia
    ixx="1.0 " ixy="0.0 " ixz="0.0 "
    iyy="1.0 " iyz="0.0 "
    izz="1.0 "/>
</inertial>
```

В этом примере работа как матрица массы, так и матрица инерции составляют значения, поскольку этот робот не имеет реального аналога.

Настройка модели Если ваша модель робота ведет себя неожиданно в Gazebo, это, вероятно, потому, что ваш URDF нуждается-

ся в дальнейшей настройке, чтобы точно представить свою физику в Gazebo. Почитайте литературу для получения дополнительной информации о различных свойствах, доступных в Gazebo, которые также доступны в URDF через `<gazebo>` тег.

5 Конвертирование URDF в SDF

5.1 При каких обстоятельствах необходимо конвертировать между URDF и SDF?

Как упоминалось выше, в некоторых особых случаях могут использоваться только модели формата URDF или SDF. Например, визуализация Rviz может использовать только URDF, а моделирование параллельных роботов Gazebo может использовать только SDF. В это время, если наш существующий формат модели не соответствует требованиям, нам нужно конвертировать.

5.2 URDF в SDF

Поскольку sdf является родным форматом Gazebo, относительно просто преобразовать .urdf в .sdf. Gazebo официально предоставляет еще один метод командной строки, который также может конвертировать urdf в sdf. Откройте терминал в каталоге, где находится urdf, и выполните следующую команду. среди них my_model.urdf-название оригинального urdf.

```
gz sdf -p my_model.urdf > my_model.sdf
```

5.3 SDF в URDF

По сравнению с .urdf в .sdf, хотя официального способа конвертировать .sdf в .urdf нет, к счастью, некоторые люди разработали жизнеспособный инструмент - пакет pysdf. Адрес проекта на github выглядит следующим образом: [andreasBihlmaier/pysdfgithub.com](https://github.com/andreasBihlmaier/pysdf)

Функциональный пакет также очень прост в использовании, разделен на 4 шага:

- Загрузите пакет функций pysdf с github и поместите его в каталог src в рабочей области ROS.

- Откройте терминал из каталога, в котором находится файл модели sdf, подлежащий преобразованию, и выполните следующую команду. среди них my_model.sdf имя исходного файла модели SDF.

```
roslaunch pysdf sdf2urdf.py my_model.sdf result_model.urdf
```

6 Заключение

Вот мы и разобрали что такое GAZEBO и ROS, какие стандарты они используют и в чем их отличие.

Разобрали как при необходимости конвертировать SDF и URDF и какие сложности возникают

7 Список литературы

<https://programmersought.com/article/52536716739/>

<http://wiki.ros.org/urdf/Tutorials>

https://classic.gazebosim.org/tutorials?tut=ros_urdf

<http://sdformat.org/>

<http://docs.voltbro.ru/starting-ros/ros-about.html>

<https://programmersought.com/article/52536716739/>

<https://cxem.net/software/gazebo.php>