Project Title

3D modelling and Simulation of Robots using ROS Kinetic and Gazebo (with Ubuntu 16.04)

Developed by: MSc. Shiva Agrawal

Place: Germany

Date: October 2018

Content

List of Figures

Abbreviations

ROS	Robot Operating System
URDF	Unified Robot Description Format
XACRO	XML Macro
XML	eXtensible Markup Language

1. Introduction

1.1. Project description

The aim of this project is to learn, understand and demonstrate the process of developing 3D models of robots using URDF and XACRO with ROS (Robot Operating System). After development of the model, the model is checked for its correction and then Gazebo simulator is used to visualize and simulate the robots.

ROS Kinetic installed on Ubuntu 16.04 is used for the project work.

The project is developed using ROS wiki [1], Effective Robotics programming with ROS [2], Mastering ROS for Robotics Programming [3], ROS wiki for URDF [4], ROS wiki for XACRO [5] and other online reference.

1.2. Outline

- Chapter 1: Project description and outline of the project report
- Chapter 2: ROS and related packages
- Chapter 3: 3D model development and visualization
- Chapter 4: results
- Chapter 5: Conclusion
- Chapter 6: References

2. ROS and related packages Python Packages

2.1. ROS (Robot Operating System)

ROS (Robot Operating System) provides libraries and tools to help software developers create robot applications. It provides hardware abstraction, device drivers, libraries, visualizers, message-passing, package management, and more. ROS is licensed under an open source, BSD license. The official website of the ROS www.ros.org is full with everything one needs to start with ROS.

Different versions of the ROS are available for the user purpose, but I have used ROS Kinetic.

2.2. ROS package

Software are organized in ROS using packages. Each package then further contains one or more nodes (a process that performs computation), dataset, configuration files, 3rd party software piece, robot model files, launch files, service and parameter files, etc. [6]

Common files and directories inside a package directory are:

- include/package_name: C++ include headers
- msg/: Folder containing Message (msg) types
- src/package_name/: Source files (C++ / Python)
- srv/: Folder containing Service (srv) types
- scripts/: executable scripts
- launch/: contain launch files to execute multiple nodes at a time
- urdf/: contains xacro and urdf files of robot model
- CMakeLists.txt: CMake build file
- package.xml: Package catkin/package.xml

NOTE: detail description of package development is out of scope of this document. Readers are requested to check ROS wiki.

2.3. Gazebo

Gazebo offers the ability to accurately and efficiently simulate variety of robots in complex indoor and outdoor environments. It provides a robust physics engine, high-quality graphics, and convenient programmatic and graphical interfaces. Best of all, Gazebo is free with a vibrant community [7].

2.4. URDF

It is abbreviation for Unified Robot Description Format. This is XML based format with predefined tags which can be used to create 3D model. It contains tags to create joints, links, colors, meshes, etc. for robot. We can also develop sensors and working environment using URDF files. ROS metapackage 'robot_model' contains various packages to design and create the 3D models of robots. Urdf is one of the packages of this meta package.

2.5. XACRO

It is abbreviation for XML Macros. The URDF representation for 3D model development is good but it has limitations of not able to define variables, equations, functions and other macros in it. Hence when the robot model is complex, URDF limits the user flexibility. These limitations of URDF are overcome by XACRO files. These are also sort of XML type files with different tag representations.

When the model is created using XACRO then there are two ways to visualization and use it in Gazebo and other visualization and/or simulation tools.

1. Directly launch the xacro file
2. Converting xacro file to urdf file and then launch it

3. Robot Model development

Whether it is mobile robot or robotic arm, the basics and the process for the development is same.

Consider the robot-arm of 6 DOF (Degree of Freedom). Such arm consists of links (rigid parts of the robot with specific length) and joints (connecting two links together with specific type of motion). Joints could be revolute, prismatic, etc.

Similarly consider 4-wheel mobile robot with Ackermann drive. Such robot can mimic the behavior of car which can be further used for research and development of autonomous driving projects. For the mobile robot, the chassis is one link and 4 wheels are other for links. Every wheel is connected with the chassis with revolute joint.

To develop the model using ROS, **robot_model** [8] meta package is used which is installed during the ROS installation full version. This meta package contains multiple packages to develop different type of models, parsing libraries and inter-conversion functions which converts one type of model description format to another type.

Models in this project are developed using two approaches

1. Using URDF files
2. Using XACRO files

The process flow is shown as below for approach 1 i.e. using URDF file

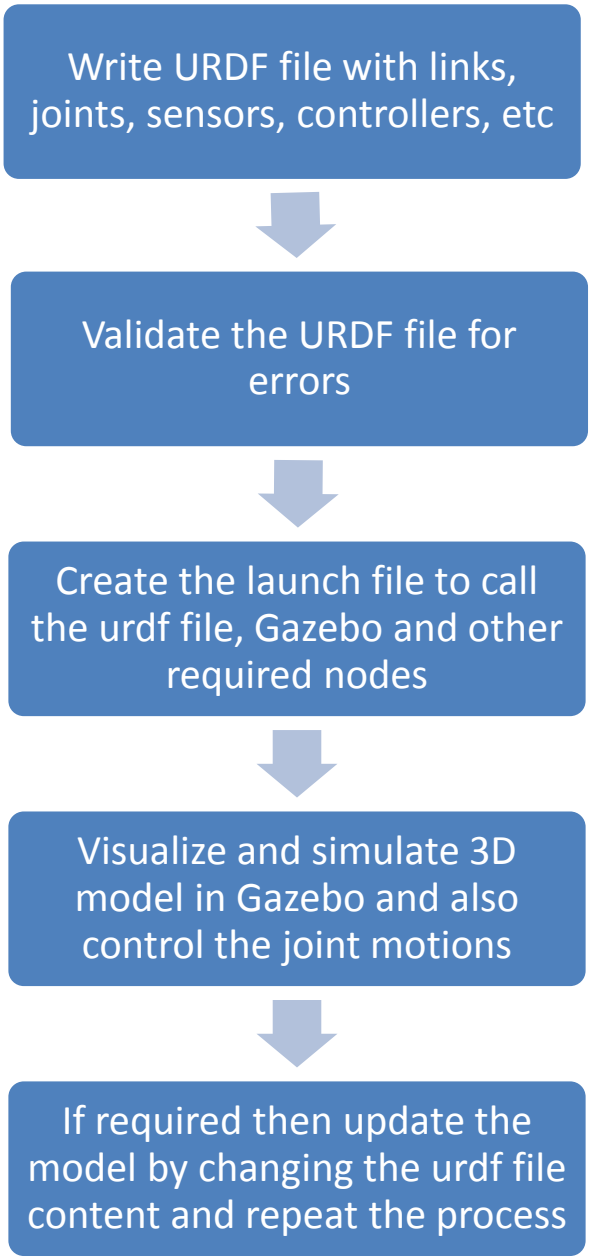

Figure 1 - Process flow for 3D model development

For approach 2, using XACRO files, create the XACRO file for the required robot model and then convert it into URDF file.

The command used to convert XACRO to URDF in ROS kinetic is

```
rosrun xacro xacro --inorder model.xacro > model.urdf
```

Use terminal and then enter the above command. Once the file is converted into URDF format then the process flow given above for approach 1 can be followed.

The commands to validate the URDF file is *check_urdf filename.urdf*

The ROS package is available in the **src** folder of the project repository. Inside the package, all the urdf and xacro files are available under URDF folder and all the launch files under LAUNCH folder.

The hierarchy diagram can be generated for each links available in URDF and can be saved as pdf file. Such generated pdf files are also available in **doc** folder of the project repository along with this project report.

One example of such hierarchy for differential drive ROBOT is given below. It contains one main link as body/base, then two wheels, two castor wheels and one sensor (Hokuyo)

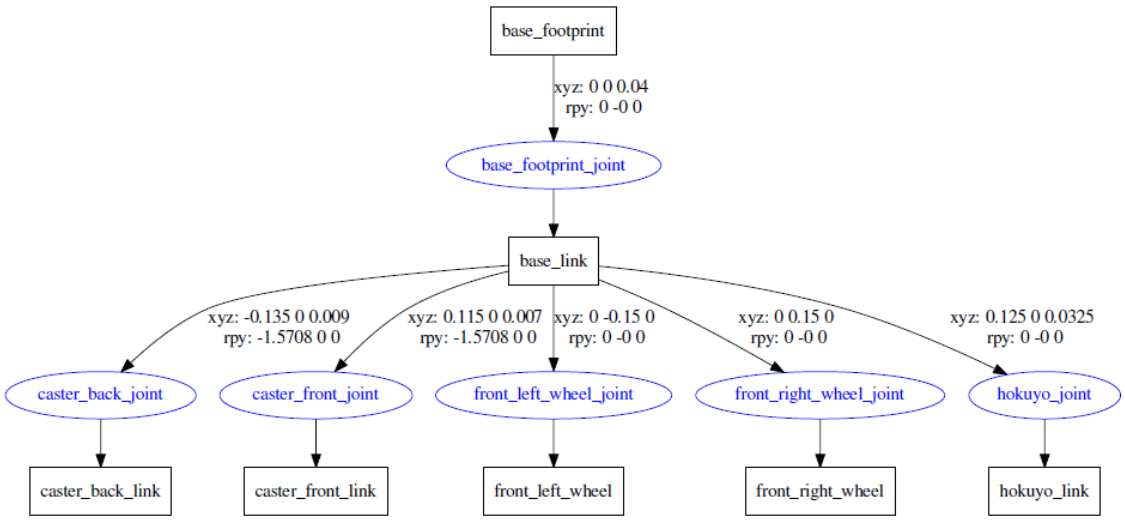

Figure 2 - Links Hierarchy of differential wheel mobile robot

4. Results

In this project, 2 models are developed and visualized in Gazebo.

1. Differential drive Robot

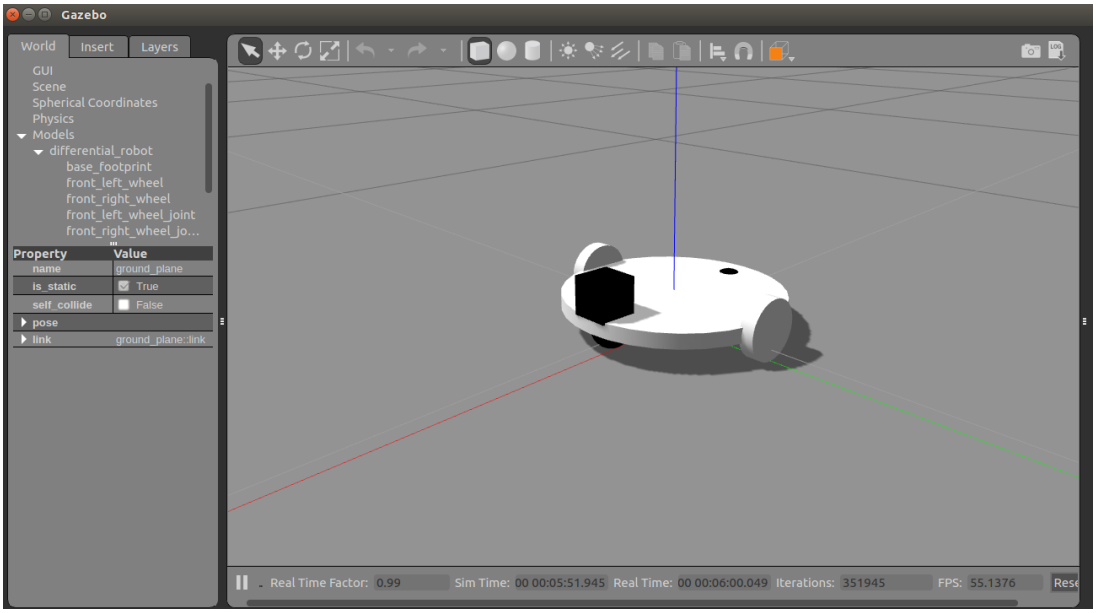

Figure 3 - Differential drive Robot in Gazebo

2. 7 DOF robotic arm

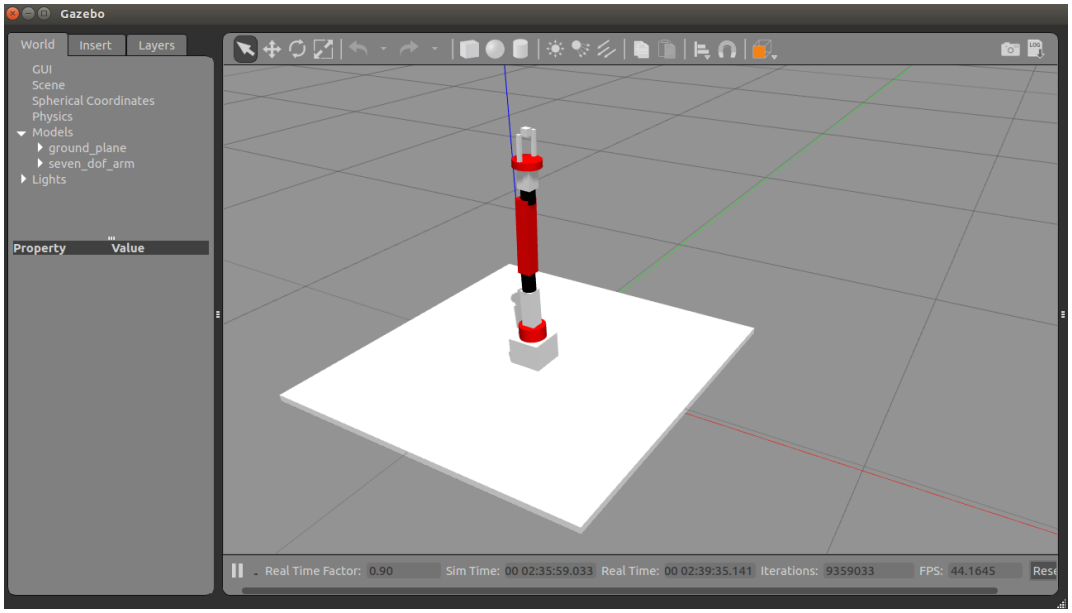

Figure 4- seven DOF arm in Gazebo

All the above screenshots are available in the **test** folder of project repository. As the project is kind of visualization and simulation which can be run by using launch files (available inside package), no separate test files are required to run it.

5. Conclusion

For developing any robotic application, it is important to test the robot extensively. This is only possible by using some simulation tools. Hence development of 3D models of the robot, incorporating sensors, controllers, etc. and then working with such models in one of the powerful simulation tools like Gazebo was the prominent aim of the project which is properly achieved.

6. References

[1] "ROS," ROS, [Online]. Available: http://wiki.ros.org/.

[2] A. Mahatani, Effective Robotics Programming with ROS - 3rd edition, Packt.

[3] L. Joseph, Mastering ROS for Robotics Programming, Packt.

[4] "URDF," ROS open source, [Online]. Available: http://wiki.ros.org/urdf.

[5] "xacro," ROS open source, [Online]. Available: http://wiki.ros.org/xacro.

[6] "packages," ROS open source, [Online]. Available: http://wiki.ros.org/Packages.

[7] "gazebosim," Open Source Robotics Foundation, [Online]. Available: http://gazebosim.org/.

[8] "robot_model meta package," ROS open source, [Online]. Available: http://wiki.ros.org/robot_model.