# **Robot modelling & Gazebo simulation activities**

# **Introduction**

You will find an incomplete model for a differential drive robot. This activity will give clarify some knowledge and give you practice about the ROS modelling using URDF and Xacro to be used in Rviz and Gazebo, by completing a simplified version of a Differential drive robot.

Inside the activity workspace folder, you will find the following folders:.

- puzzlebot\_control – contains ROS control files

- puzzlebot\_gazebo – contains Robot model files

- puzzlebot\_world - Contains worlds for gazebo

We are going to be working with puzzlebot\_gazebo. Here the list of files inside this package.

puzzlebot\_gazebo

├── CMakeLists.txt

├── launch

│ ├── config1.rviz

│ ├── puzzlebot\_gazebo.launch

│ └── puzzlebot\_joints\_test.launch

├── meshes

│ ├── camera.stl

│ ├── chassis.stl

│ └── wheel.stl

├── package.xml

└── urdf

├── macros.xacro

├── materials.xacro

├── parameters.xacro

├── puzzlebot.gazebo

└── puzzlebot.xacro

Diagram 1. List of files inside gazebo model

**Note:It is an standard and recommended as good practices to keep this files splitted in folders.**

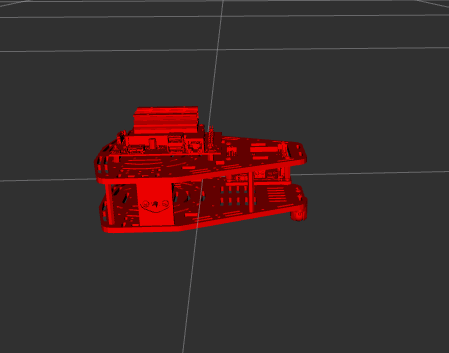
**Warning: Do not modify any other file thats not specified in the activity, unless you are told to do so.**

# **Activity #1: URDF modelling and Xacro simplification** - (time 35 min) :

Inside puzzlebot \_gazebo, if you run the launch file

roslaunch puzzlebot\_gazebo puzzlebot\_joints\_test.launch

like the one shown in the picture.



The first task is divided in 4 mini sub task.

1. Task 1.a: Create one wheel for the robot. The dimensions are already defined in the parameters.xacro file. Use the defined variables and add the joint.
2. Task 1.b: Based on task 1.a, Using xacro to create a macro for the wheel ( move this to them macro.xacro file) This macro will add you added add a simetrical wheel on the other side of the wheel chassis and add a joint to it.

Note: Move the macro to the macro.xacro is best practices.

1. Task 1.c: Add meshes to all link ( wheels ).

**Hint: Identify where the meshes are, and check how was added to the base\_link link.**

1. Task 1.d .Check the wheels are movable with joint\_state publisher\_gui, load rviz config (gazebo file and test joints. Once you run it a testes with the gui.

# **Activity #2: Gazebo tags, and ROS Control and Gazebo simulation** - (time 35 min) :

In this section you need to add some Gazebo tags., to make it compatible for Gazebo visualization and siimulation.

1. Task 1. Open the world with the following command :

gazebo

Then, add walls to create a 4x4 metre room around the origin of the world, add a bookshelf from the model library, and save the world as room.world in puzzlebot\_world/world. This will not involve the robot model yet.

1. Task 2. Add Gazebo tags, including ros\_control plugin.

This part is mainly done for you. The full idea of this tag is that you uncomment the necessary lines and get familiarized with the folders.

1. Task 3 Add Transmisiones. Transmisions are the link between the URDF/Xacro model and ROS control. The Cheat Sheet provided has an example so you can add one to each link. Please, mind the name of the joints.
2. Task 4. Add ROS controller. As mentioned dring the presentation, ROS controllers include some basic already-made pieces of code, but also you can create the one if you design requires it. For this, the part of the code that will be used is one already done. So you can loaded and test it. For the sake of time, if you want to see a costumised Controller, please go to the full model of Puzzlebot.
3. To test fully run the

roslaunch puzzlebot\_gazebo spawn\_puzzlebot\_gazebo.launch