**Creating and Simulating a Robot in ROS2 & Gazebo using URDF and Fusion 360**

This documentation provides a **step-by-step guide** to simulate a mobile robot that was modeled in **Fusion 360**. The robot is described using **URDF/Xacro**, simulated in **Gazebo**, and controlled via **ROS2**.

## **1. Robot Design in Fusion 360**

## **Overview:**

* The robot was first modeled in Fusion 360 with the following components:  
  + Main chassis
  + Two side wheels for motion control
  + Two caster wheels (which rotate freely in all directions for balance)

### **Exporting Model:**

* Export each part (e.g., chassis, wheels, caster wheels) as STL or STEP files.
* Ensure that the orientation is consistent:  
  + X-axis → Forward
  + Y-axis → Left
  + Z-axis → Up

## 

## 

## **2. Setting Up the ROS2 Workspace**

### **Create and Initialize Workspace:**

mkdir -p ~/ros2\_ws/src

cd ~/ros2\_ws/src

ros2 pkg create --build-type ament\_cmake <robot\_description\_package>

### **Build and Source:**

cd ~/ros2\_ws

colcon build

source install/setup.bash

## **3. Create URDF Files Using Xacro**

* Use Xacro (XML Macros) to simplify and modularize your URDF file.
* File naming should follow: robot\_name.urdf.xacro (e.g., tea\_bot.urdf.xacro).
* Create a urdf directory inside your package and place the .xacro files and mesh files there.
* Define:  
  + base\_link as the root link, centered at the robot’s origin.
  + Wheel joints and caster wheels using accurate origin and axis values.

**Coordinate Convention:**

* X → Forward
* Y → Left
* Z → Up

## **4. URDF Structure and Link Setup**

### **Base Link**

* Must be the **root link** and placed at the **origin (0,0,0)** for simplicity.

<link name="base\_link">

<visual>...</visual>

<collision>...</collision>

<inertial>...</inertial>

</link>

**Connecting Wheels and Casters**

Each wheel is added as a separate link and connected via a **joint**:

<joint name="left\_wheel\_joint" type="continuous">

<parent link="base\_link"/>

<child link="left\_wheel"/>

<origin xyz="0 0.175 0" rpy="0 0 0"/>

<axis xyz="0 0 1"/>

</joint>

* For right wheel, set axis as:

<axis xyz="0 0 -1"/>

### **Inertia**

Either manually add:

<inertial>

<origin xyz="..." rpy="..."/>

<mass value="..."/>

<inertia ixx="..." iyy="..." izz="..." .../>

</inertial>

## **5. Launching in RViz (URDF Preview)**

### **Run URDF with robot\_state\_publisher:**

Create rsp.launch.py:

from launch import LaunchDescription

from launch\_ros.actions import Node

def generate\_launch\_description():

return LaunchDescription([

Node(

package='robot\_state\_publisher',

executable='robot\_state\_publisher',

output='screen',

parameters=[{'robot\_description': Command(['xacro ', PathJoinSubstitution([

FindPackageShare('your\_package'),

'urdf',

'robot.urdf.xacro'

])])}]

)

])

Then run:

ros2 launch <your\_package> rsp.launch.py

### **Note:**

* If the robot doesn’t appear correctly:  
  + Click "Reset" in RViz
  + Relaunch the file
  + Verify all paths and file names

## **6. Spawning Robot into Gazebo**

### **Command to spawn:**

ros2 run gazebo\_ros spawn\_entity.py -entity my\_robot \

-file ~/ros2\_ws/src/<your\_package>/urdf/robot.urdf \

-x 0 -y 0 -z 0

## 

## 

## **7. Adding ros2\_control for Motion**

### **Add Control Plugin to URDF:**

<gazebo>

<plugin name="diff\_drive\_controller" filename="libgazebo\_ros2\_control.so">

<left\_wheel>left\_wheel\_joint</left\_wheel>

<right\_wheel>right\_wheel\_joint</right\_wheel>

<wheel\_separation>0.35</wheel\_separation>

<wheel\_radius>0.05</wheel\_radius>

<command\_topic>cmd\_vel</command\_topic>

</plugin>

</gazebo>

### **Set Control Parameters:**

You may also define:

* **Max torque**
* **Acceleration**
* **PID gains**
* In the robot control YAML or inside the plugin

## **8. Teleoperation Control**

### **Install teleop\_twist\_keyboard:**

sudo apt install ros-<distro>-teleop-twist-keyboard

### 

### **Run the teleop node:**

ros2 run teleop\_twist\_keyboard teleop\_twist\_keyboard

* Use WASD keys to move the robot
* Ensure cmd\_vel is the subscribed topic in your plugin

## **9. Friction Settings (Optional)**

To enhance movement:

* Set different **friction coefficients** for left/right wheels
* Use SDF format or Gazebo <surface> tag in URDF

Example:

<surface>

<friction>

<ode>

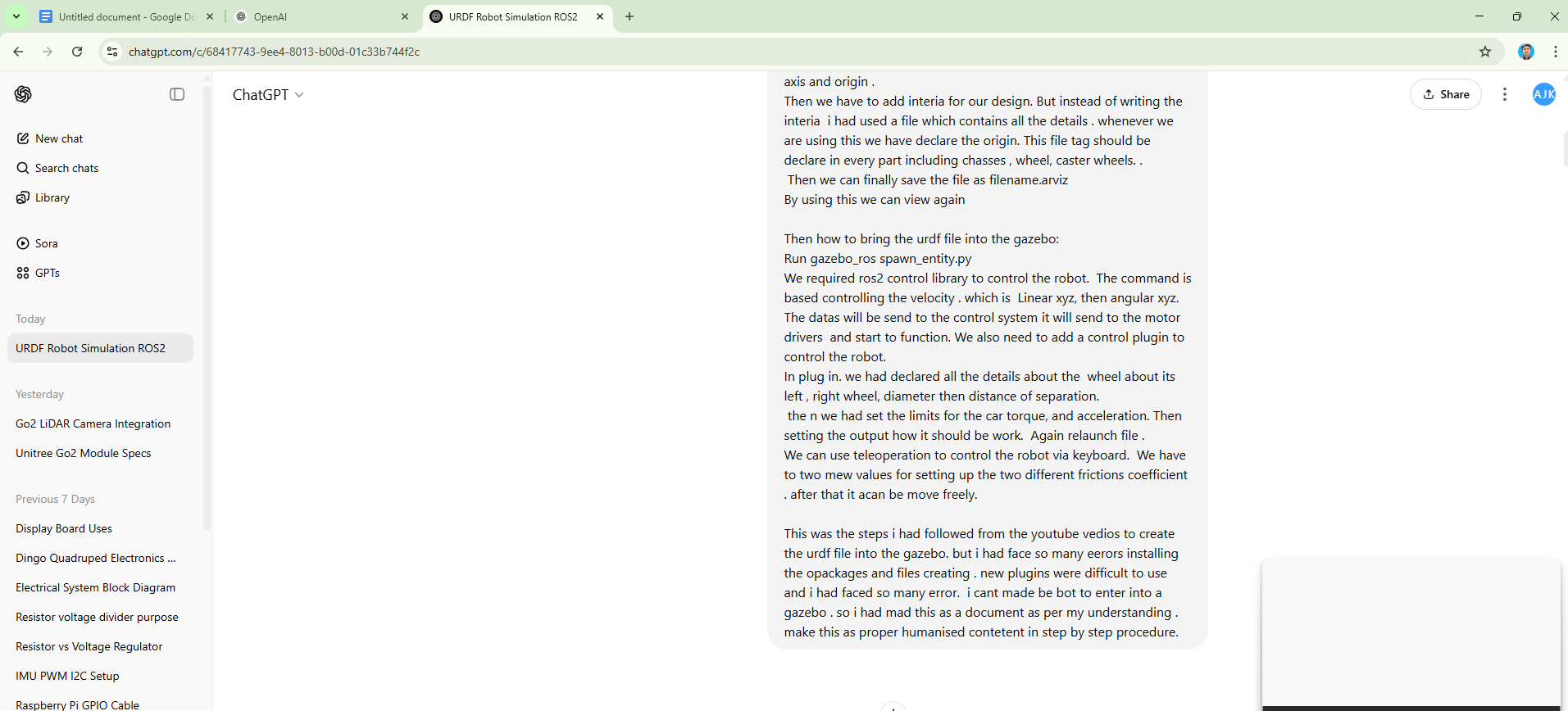
<mu>1.0</mu>

<mu2>1.0</mu2>

</ode>

</friction>

</surface>



My Work :[Link](https://chatgpt.com/c/68417743-9ee4-8013-b00d-01c33b744f2c)