Modelado y Simulación de Robots - GIRS

Práctica 3: Simulación de Robots usando middleware

Parte A

Objetivo

Configurar correctamente el modelo de robot utilizado en la práctica 2 (el que se realizo en Blender) para poder ser visualizado, operado y controlado usando el software disponible en `ROS 2`.

Actividades

Al final estas dos primeras sesiones se debería tener un paquete de ROS, en el cual el robot se encuentre totalmente configurado y compatible con REP-103 y REP-105.

El robot además de estar plenamente funcional debe contener:

- un sensor IMU ubicado en el centro del robot, o dentro de la base del mismo si su geometría no lo permite.
- una cámara frontal posicionada en la parte delantera del robot
- una cámara que vaya unida al eslabón final del manipulador robot

Entregables

Un paquete de `ROS 2`, donde se encuentre la configuración del robot, tradicionalmente estos paquetes se suelen llamar `robot_description`.

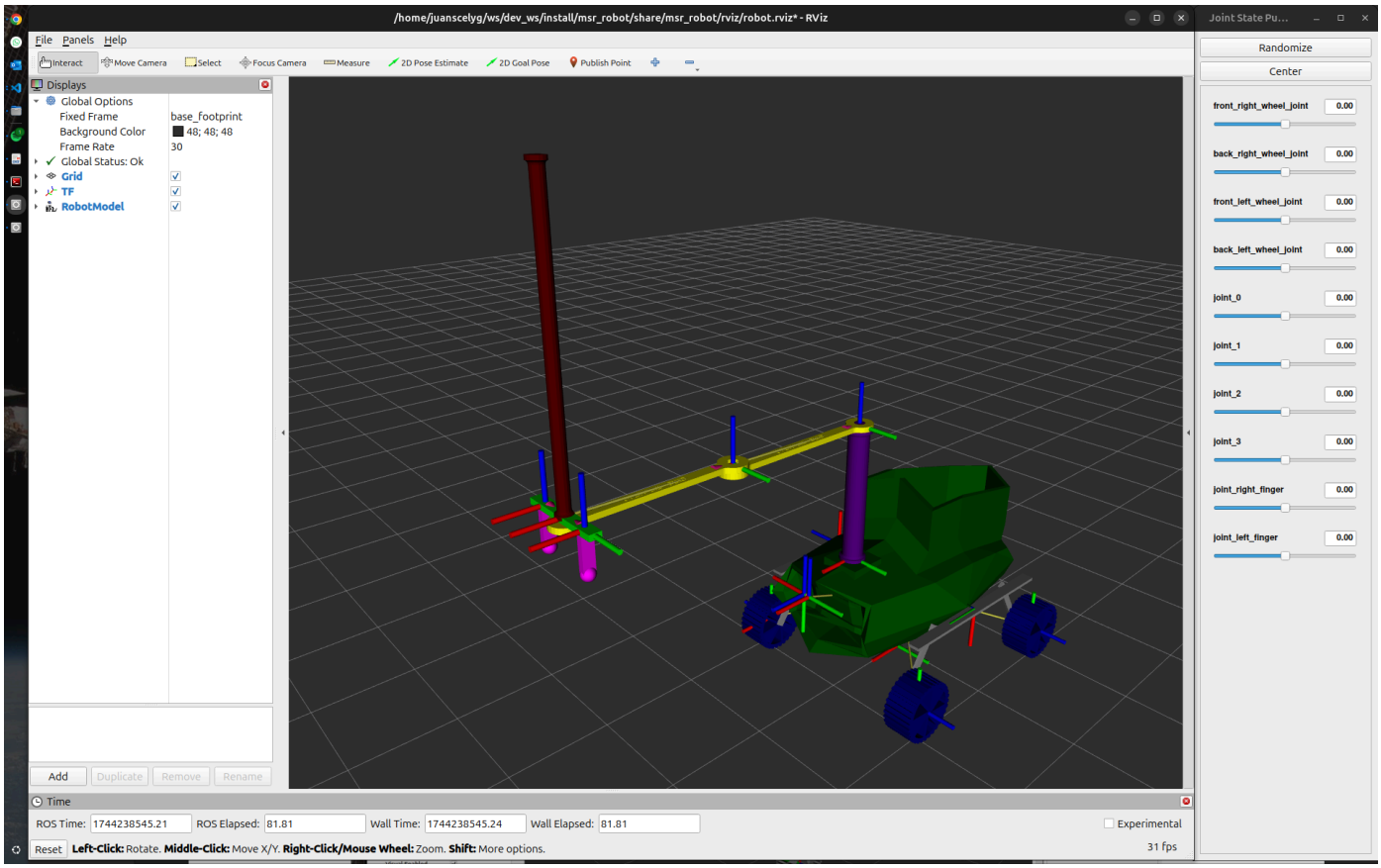
Una estructura ideal del paquete podría ser la siguiente:

```
├── CMakeLists.txt
├── config
│   └── config.yaml
├── launch
│   └── robot_state_publisher.launch.py
├── meshes
│   └── dae
│       ├── base.dae
│       ├── base_gripper.dae
│       ├── Cube.000.dae
│       ├── Cylinder.dae
│       ├── eslabon.dae
│       ├── eslabon_vertical.dae
│       └── finger.dae
├── package.xml
├── robots
│   └── robot.urdf.xacro
├── rviz
│   └── robot.rviz
└── urdf
    ├── arm
    │   ├── gripper.urdf.xacro
    │   └── scara.urdf.xacro
    ├── base
    │   └── robot_base.urdf.xacro
    ├── ros2_control.urdf.xacro
    ├── sensors
    │   ├── camera.urdf.xacro
    │   ├── imu_sensor.urdf.xacro
    │   └── sensors.urdf.xacro
    ├── utils
    │   └── utils.urdf.xacro
    └── wheels
        └── rover_wheel.urdf.xacro
```

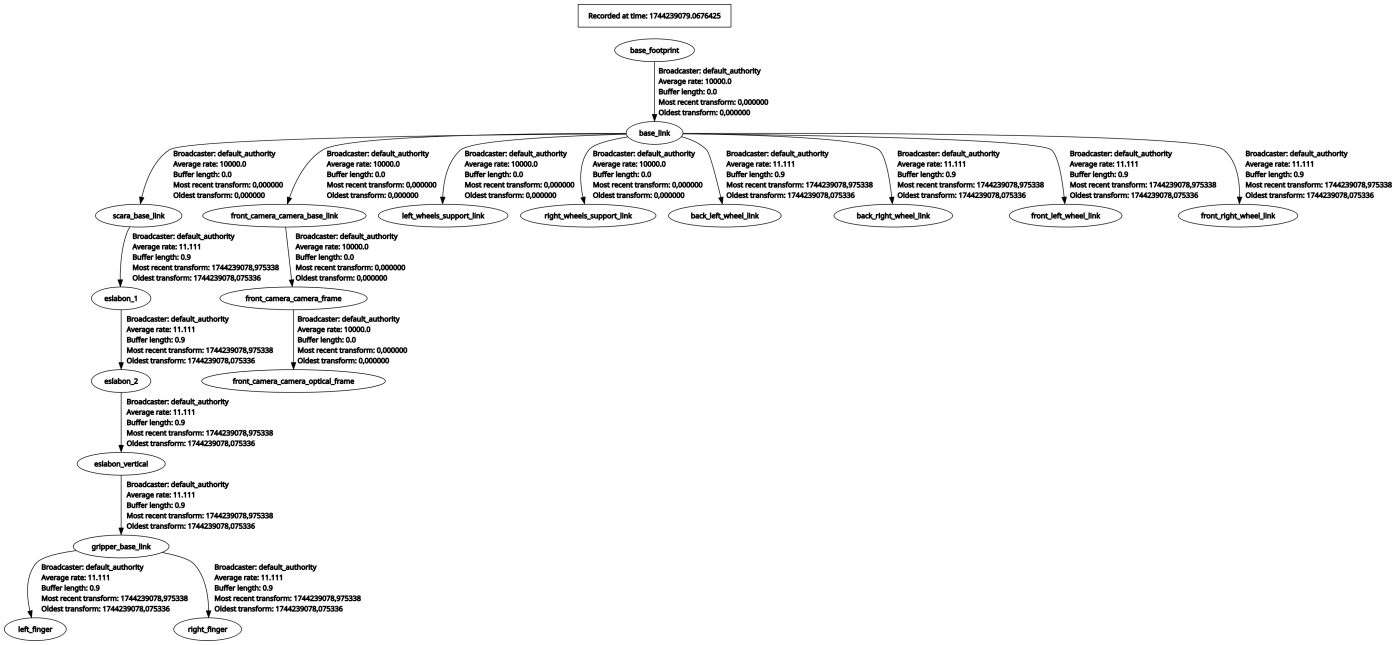
El paquete debe contener:

- Un modelo del robot en URDF totalmente compatible con REP-103 y REP-105.
- El modelo del robot debe contener un link denominado `base_footprint`, el cual debe ser padre del link `base_link`, el cual a su vez debe ser padre del resto de los links del robot. Importante que cumpla con la orientación correspondiente.

- Una carpeta meshes donde se encuentren los modelos 3D del robot, tanto en formato `STL` o en formato `DAE` .
- Una carpeta `urdf` donde se encuentre el modelo del robot en formato `URDF` . En este caso es indispensable que se haya usado `xacro` para la creación del modelo, ya que es el formato que se utiliza en `ROS 2` .
- Una carpeta `robots` donde esté el archivo principal del robot, que en este caso puede ser `*.urdf.xacro`
- Una carpeta `launch` donde se encuentre el archivo de lanzamiento del robot, que en este caso puede ser `*.launch.py`
- El paquete debe permitir visualizar el robot en RViz y manipular los joint a través de `joint_state_publisher_gui` . Como se visualiza a continuación:



- Debe existir una imagen o PDF que visualice el árbol de transformadas entre los links del robot. Este se puede obtener a través de `rqt_tf_tree` o con `tf2_tools` y el nodo `view_frames` . La salida debe ser similar a la siguiente:



Anexos

1. Añadir las opciones de XACRO a un archivo URDF

Añade en la linea de la creación del robot la linea de declaración de `xacro`:

```
<?xml version="1.0"?>
<!-- created with Phobos 1.0.1 "Capricious Choutengan" -->
<robot name="robot" xmlns:xacro="http://wiki.ros.org/xacro">
```

2. Creación de un macro usando XACRO

Creación de un macro llamado `solid_cuboid_inertia`, que recibe como parametros los valores de `m w h d`:

```
<xacro:macro name="solid_cuboid_inertia" params="m w h d">
    <inertia    ixx="${(m*(h*h+d*d))/12}" ixy = "0" ixz = "0"
                iyy="${(m*(w*w+d*d))/12}" iyz = "0"
                izz="${(m*(w*w+h*h))/12}"/>
</xacro:macro>
```

3. Como usar un macro en XACRO

Se usa un macro llamandolo por su nombre y pasandole los parametros que recibe:

```
<xacro:solid_cuboid_inertia m="0.5" w="0.1" h="0.1" d="0.1"/>
```

4. Llamar archivos URDF desde otro archivo URDF

Para llamar un archivo URDF desde otro archivo URDF se usa la siguiente sintaxis:

```
<xacro:include filename="$(find <nombre_del_paquete>)/urdf/<nombre_del_archivo>.urdf.
```