

Modelado y Simulación de Robots

Apoyo Práctica 4

Grado en Ingeniería de Robótica Software

Teoría de la Señal y las Comunicaciones y
Sistemas Telemáticos y Computación

Sergio Paniego Blanco
sergio.paniego@urjc.es

Sumario

- Ejemplo mover joints del robot en URDF.
- Acceder a información relevante de los topics.
- Ejemplo rospy (*Publish*).
- Ejemplo PyQt5.

Ejemplo 1. Mover joints en Gazebo

- Creamos package y carpetas relevantes

```
catkin_create_pkg joint_ros_gz_tutorial controller_manager  
joint_state_controller robot_state_publisher  
mkdir joint_ros_gz_tutorial/launch  
mkdir joint_ros_gz_tutorial/urdf  
mkdir joint_ros_gz_tutorial/config
```

- Añadimos modelo sencillo URDF.
- El plugin va dentro de `<gazebo>` por ser URDF.
- Se añaden los detalles en `<transmission>`.

Ejemplo 1. Mover joints en Gazebo

- Se añade la configuración del controlador en un nuevo archivo *.yaml*

```
husky_robot:  
  joint_state_controller:  
    type: joint_state_controller/JointStateController  
    publish_rate: 20  
  
  joint_velocity_controller:  
    type: effort_controllers/JointVelocityController  
    joint: front_left_wheel  
    pid: {p: 1.0, i: 1.0, d: 0.0}
```

Ejemplo 1. Mover joints en Gazebo

- En el *.launch* se añade el detalle del *.yaml*

```
<launch>
  <param name="robot_description" textfile="$(find joint_ros_gz_tutorial)/urdf/husky_robot.urdf" />
  <node name="urdf_spawner" pkg="gazebo_ros" type="spawn_model" respawn="false" output="screen"
args="-urdf -model husky_robot -param robot_description" />
  <!-- loads the controllers -->
  <rosparam file="$(find joint_ros_gz_tutorial)/config/config.yaml" command="load" />
  <node name="controller_spawner" pkg="controller_manager" type="spawner" respawn="false"
output="screen" ns="/husky_robot" args="joint_velocity_controller joint_state_controller"/>
  <!-- converts joint states to TF transforms -->
  <node name="robot_state_publisher" pkg="robot_state_publisher" type="robot_state_publisher"
respawn="false" output="screen">
    <remap from="/joint_states" to="/husky_robot/joint_states" />
  </node>
</launch>
```

Ejemplo 1. Mover joints en Gazebo

- Se lanza el robot con roslaunch

```
roslaunch joint_ros_gz_tutorial joint_world.launch
```

- Podemos mover el joint publicando en el *Topic*

```
rostopic pub -1 /husky_robot/joint_velocity_controller/command std_msgs/Float64 "data: 3.0"
```

Ejemplo 2. Acceder a posición del robot

- ¿Cómo accedemos a la posición del robot?

```
rostopic list
```

- Movemos el robot y volvemos a comprobar la posición

```
rostopic echo -n 1 /gazebo/model_states
```

Ejemplo 3. Publish topics con rospy

- Con la librería rospy, publicamos al topic anterior para comprobar la funcionalidad con python.
- Más tutoriales: http://wiki.ros.org/rospy_tutorials

Ejemplo 3. Publish topics con rospy

```
import rospy
from std_msgs.msg import Float64

def move_joint():
    rospy.init_node("move_joint")
    rate = rospy.Rate(1)
    pub = rospy.Publisher('/husky_robot/joint_velocity_controller/command', Float64, queue_size=10)
    msg = Float64()
    msg.data = 1.0
    i = 0
    while i < 10:
        print(msg)
        pub.publish(msg)
        print('Publishing')
        rate.sleep()
        i += 1
    rospy.spin()

move_joint()
```

Ejemplo 3. Publish topics con rospy

- Lanzamos el código Python

```
python3 rospy_publish_example.py
```

Ejemplo 4. Ejemplo GUI con PyQt5

- Creamos una GUI sencilla con un par de botones
- Otros tutoriales:
 - <https://build-system.fman.io/pyqt5-tutorial>
 - <https://pythonpyqt.com/pyqt-label/>
 - <https://www.techwithtim.net/tutorials/pyqt5-tutorial/buttons-and-events/>

Ejemplo 4. Ejemplo GUI con PyQt5

```
from PyQt5.QtWidgets import QApplication, QLabel, QWidget, QPushButton, QVBoxLayout

def btn_1_clicked():
    print('clicked btn 1!')

def btn_2_clicked():
    btn_3 = QPushButton('New backward')
    layout.addWidget(btn_3)

app = QApplication([])

window = QWidget()
layout = QVBoxLayout()
layout.addWidget(QLabel('Move robot'))

btn_1 = QPushButton('Forward')
btn_1.clicked.connect(btn_1_clicked)
layout.addWidget(btn_1)

btn_2 = QPushButton('Backward')
btn_2.clicked.connect(btn_2_clicked)
layout.addWidget(btn_2)

window.setLayout(layout)
window.show()
app.exec_()
```

