

Ros2 with Ignition Gazebo

Polytech Angers - Mobile Robotics

- [ROS2 bridge](#)
- [Start everything with a launch file](#)

From now, you should have your robot that is moving with ignition gazebo. This is fun, but we do not want to use gazebo for the command: we want to use ros2 to control our robot. So far, the topics from gazebo are not available on ros2

ROS2 bridge

Start your gazebo simulation and check if you can see the topic with ros2:

```
docker@ros2:~/wdir$ ros2 node list
docker@ros2:~/wdir$ ros2 topic list
/parameter_events
/rosout
```

Nope...

It is possible to use `rosbridge` to do a bridge between ros2 and gazebo topics. Using the following command will connect a `/cmd_vel` topic from ros2 to a `/cmd_vel` topic in gazebo.

```
docker@ros2:~/wdir$ ros2 run ros_gz_bridge parameter_bridge
/cmd_vel@geometry_msgs/msg/Twist]ignition.msgs.Twist
```

- `ros2 run ros_gz_bridge parameter_bridge` to start the bridge
- `/cmd_vel` the topic name
- `geometry_msgs/msg/Twist` the type of the topic from ros2 side
- `] from ros2 to gazebo`
- `ignition.msgs.Twist` the topic type from gazebo side

and wait for the following result:

```
[INFO] [1680250360.443094577] [ros_gz_bridge]: Creating ROS->GZ Bridge:
[/cmd_vel (geometry_msgs/msg/Twist) -> /cmd_vel (ignition.msgs.Twist)]
(Lazy 0)
```

now you should have the topic `/cmd_vel` available to ros2:

```
docker@ros2:~/wdir$ ros2 topic list
/cmd_vel
/parameter_events
/rosout
```

You can try to publish a message on the `/cmd_vel` topic to check if it is processed by the gazebo simulation (this command has been introduced in [1_ros2_introduction](#)).

```
docker@ros2:~/wdir$ ros2 topic pub --once /cmd_vel geometry_msgs/msg/Twist
"{linear: {x: 2.0, y: 0.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z: 2.0}}"
```

The arguments of the command are:

- `--once`: an optional argument to publish this message only once
- `/cmd_vel`: the name of the topic
- `geometry_msgs/msg/Twist`: the type of message
- `"{linear: {x: 2.0, y: 0.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z: 1.0}}"`: the value of the message. This should be consistent with the type of the message as shown before.

Now we are able to control the simulated robot of gazebo from ros2 topics. It would be nice to be able to use your `my_teleop` node (`1_3_ros2_tp1_teleop_turtle`) to control this new robot. The problem is that the command topic from the `my_teleop` node is `/turtle1/cmd_vel` but the command we need for gazebo is `/cmd_vel`...

Remark: If you do not have the `my_teleop` node working properly, you can use the one provided by the `turtlesim` package instead:

```
ros2 run turtlesim turtle_teleop_key
```

Start everything with a launch file

Using what you have learned so far, create a new launch file named `teleop_gazebo_launch.py` that

- Start the gazebo simulation with the world built before (with the robot);
- Start the `teleop` node (yours or the `turtlesim` one) remapping the `/cmd_vel` topic;
- Start a bridge node to do the bridge between the ros2 `/cmd_vel` and the ign `/cmd_vel`
 - you can use the `arguments=['bla bla bla',]` parameter of Node class to add the parameters of the bridge

By starting the launch file you should be able to control the robot using the keyboard.

Node: the `-r` option of the `ign gazebo` command allows to run the simulation at start... may be handy not to have to click play on each try...