

# Gazebo - Méthode Actionneurs

Une autre méthode pour contrôler tous les modules est d'utiliser des plugins intégrés à Gazebo qui agiront comme des moteurs attachés aux liaisons.

Pour la barrière, en plus de la liaison de type *revolute* on appliquera le plugin *gazebo\_ros\_control*:

```
<gazebo>
  <plugin name="gazebo_ros_control" filename="libgazebo_ros_control.so">
    <robotNamespace>/barrier1</robotNamespace>
  </plugin>
</gazebo>

<transmission name="barrier1_transmission">
  <type>transmission_interface/SimpleTransmission</type>
  <actuator name="motor1">
    <mechanicalReduction>1</mechanicalReduction>
  </actuator>
  <joint name="base_to_barrier">
    <hardwareInterface>hardware_interface/EffortJointInterface</hardwareInterface>
  </joint>
</transmission>
```

Ensuite, il faut ajouter une configuration qui permettra à Gazebo d'interpréter ce moteur:

*config.yaml*

```
barrier_1:
  joint_1_state_controller:
    type: joint_state_controller/JointStateController
    publish_rate: 50

  gear_to_barrier_1_position_controller:
    type: effort_controllers/JointEffortController
    joint: gear_to_barrier_1
```

Enfin, on pourra publier sur le topic généré par le plugin:

```
pub = rospy.Publisher('/barrier1/gear_to_barrier_1_position_controller/command', Float64, queue_size=10)
```

---

Revision #1

Created 21 June 2023 15:41:28 by Leveque Robin

Updated 21 June 2023 15:52:12 by Leveque Robin