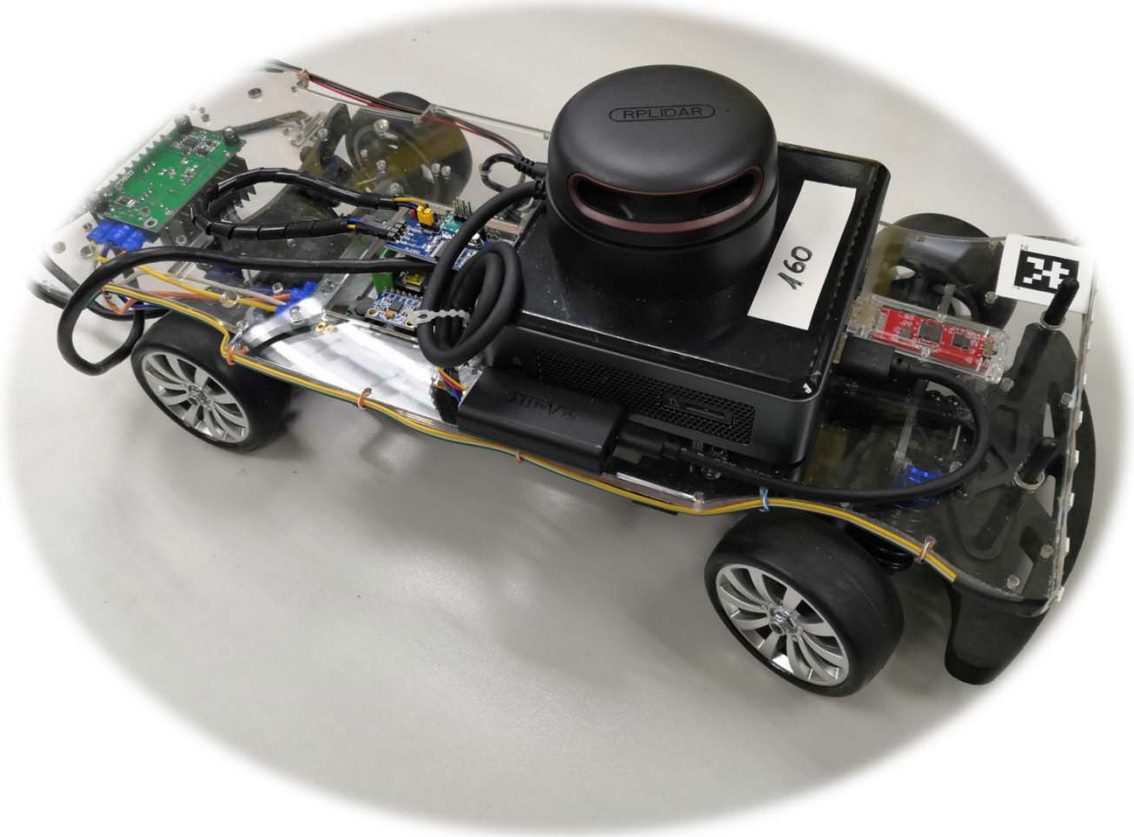
Autonomous Car Autominy

Dr. Roger Miranda Colorado

Creating a Publisher Node for the Autominy Simulator



CONACYT
Consejo Nacional de Ciencia y Tecnología

IPN CITEDI
Centro de Investigación y Desarrollo de Tecnología Digital

Instituto Politécnico Nacional-CITEDI

Creating a Publisher Node-Simulator

The next step is to design a program in **Python** to control the autonomous vehicle. The first program that we will create will make use of the next topics:

- **/actuators/speed**
- **/actuators/steering_normalized**

We also need to know the type of messages that these topics use. For the topic /actuators_speed, we obtain this information by using the command:

- **$ rostopic info /actuators/speed**

and we obtain as a result the message type:

- **autominy_msgs/SpeedCommand**

We can do the same for the topic /actuators/steering_normalized. Then, we use the command:

- **$ rostopic info /actuators/steering_normalized**

And we obtain the message type:

- **autominy_msgs/NormalizedSteeringCommand**

Creating a Publisher Node-Simulator

Then, we can obtain information of the message structure through the command:

- **$ rosmsg show autominy_msgs/Speedcommand**

Then, we obtain the structure of this message:

```
std_msgs/Header header
        uint32 seq
        time stamp
        string frame_id
float64 value
```

We may follow the same procedure for obtaining the structure of the other message:

- **$ rosmsg show autominy_msgs/NormalizedSteeringCommand**

Then, we obtain the structure of this message:

```
std_msgs/Header header
        uint32 seq
        time stamp
        string frame_id
float32 value
```

Creating a Publisher Node-Simulator

Once we know the type of messages and the topics we are going to use, we create our publisher node. The structure and description of the node is given in the following:

```python
#!/usr/bin/env python
#Publisher program for topics /actuators/speed and /actuators/steering_normalized

import rospy
from autominy_msgs.msg import SpeedCommand
from autominy_msgs.msg import NormalizedSteeringCommand

def nodePublisher():
    rospy.init_node('publisher01Autominy',anonymous=False)
    pubSpeed = rospy.Publisher('/actuators/speed',SpeedCommand,queue_size=10)
    pubSteering = rospy.Publisher('/actuators/steering_normalized',NormalizedSteeringCommand,queue_size=10)
    rate = rospy.Rate(0.2) #Five seconds
    msgSpeed = SpeedCommand()
    msgSteering = NormalizedSteeringCommand()

    while not rospy.is_shutdown():
        msgSpeed.value = 0.3
        msgSteering.value = 1
        pubSpeed.publish(msgSpeed)
        pubSteering.publish(msgSteering)
        rate.sleep()

        msgSpeed.value = 0
        msgSteering.value = 1
        pubSpeed.publish(msgSpeed)
        pubSteering.publish(msgSteering)
        rate.sleep()

if __name__ == '__main__':
    try:
        nodePublisher()
    except rospy.ROSInterruptException:
        pass
```

Creating a Publisher Node-Simulator

After creating our node, we have to make it **executable**. Then, we open up a new terminal and use the following command:

- **$ chmod +x [program_name]**

```
roger@roger-HP-Spectre-x360-Convertible:~/ckAutominy2022/src/pack01Autominy/scripts$ chmod +x program01a
.py
```

Then, we verify that our program is executable with the command:

```
roger@roger-HP-Spectre-x360-Convertible:~/ckAutominy2022/src/pack01Autominy/scripts$ ls
program01a.py  program01.py  program02.py
```

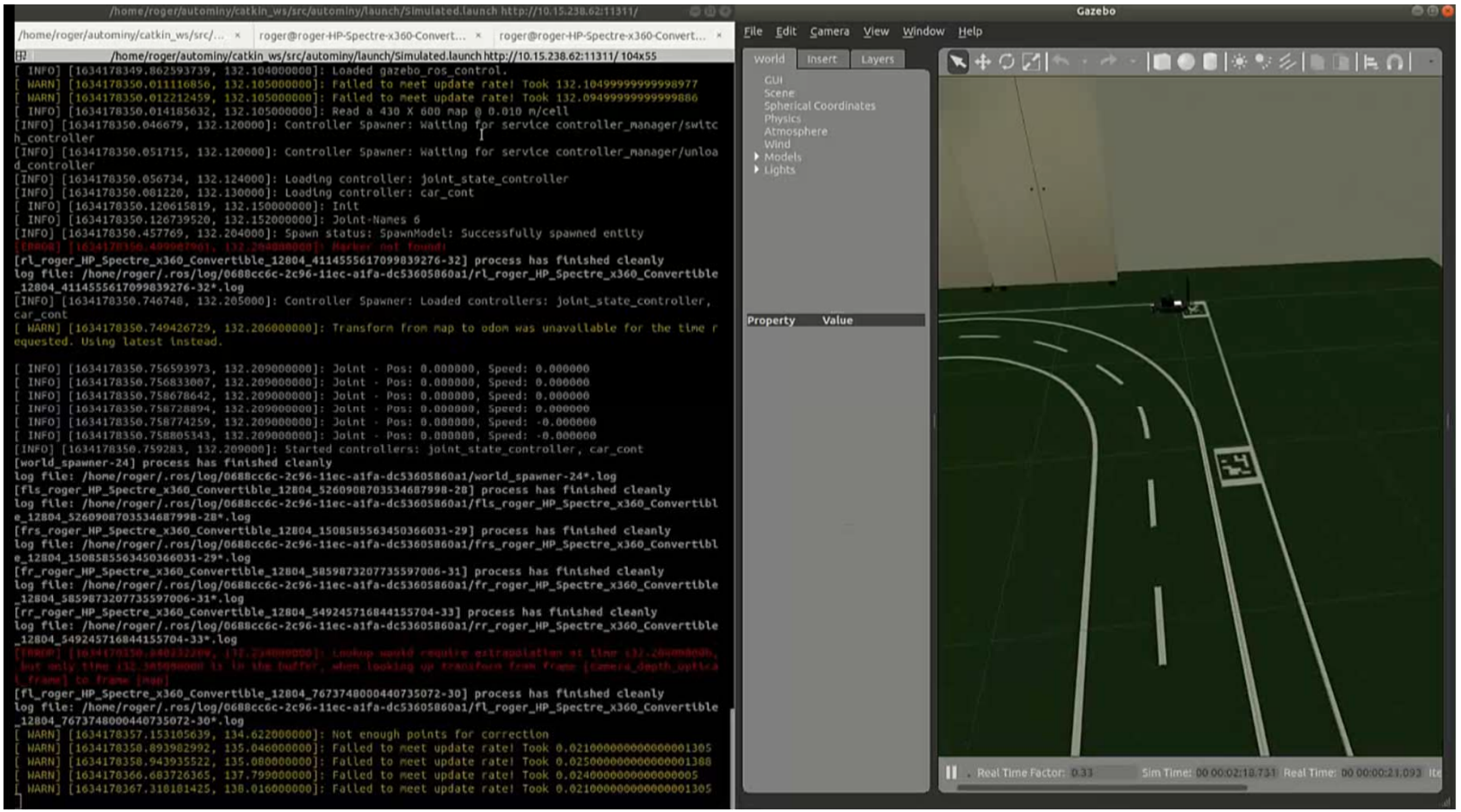
Now, we have to execute the **launch file**:

- **$ roslaunch autominy Simulated.launch**

```
roger@roger-HP-Spectre-x360-Convertible:~/ckAutominy2022/src/pack01Autominy/scripts$ roslaunch autominy
Simulated.launch
```

Creating a Publisher Node-Simulator

The previous command opens up a new window with the **Gazebo simulator** of the autonomous vehicle. This is depicted in the following picture:

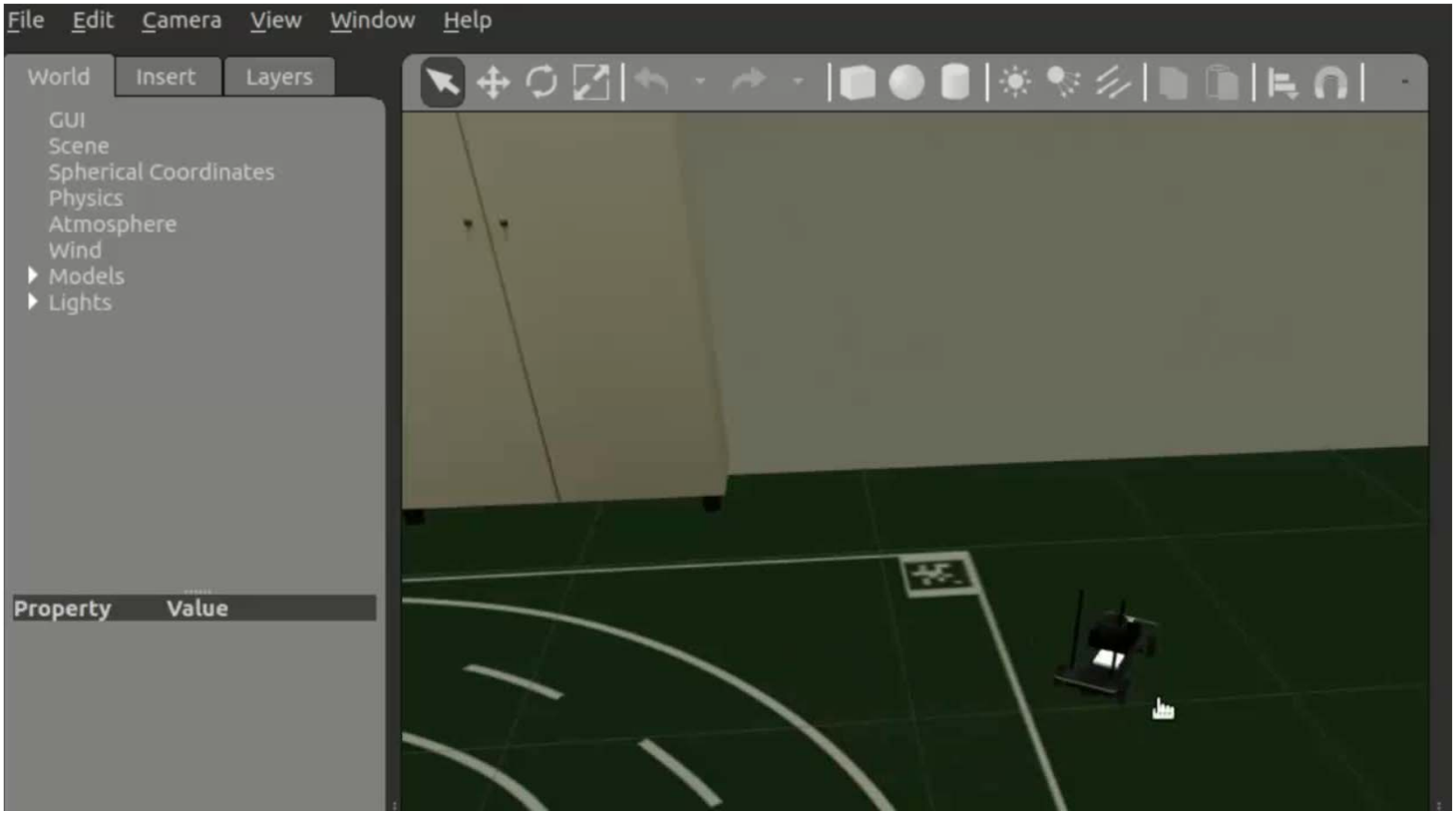


Creating a Publisher Node-Simulator

Finally, we can execute the publisher node with the following command:

- **$ rosrun [package_name] [program_name]**

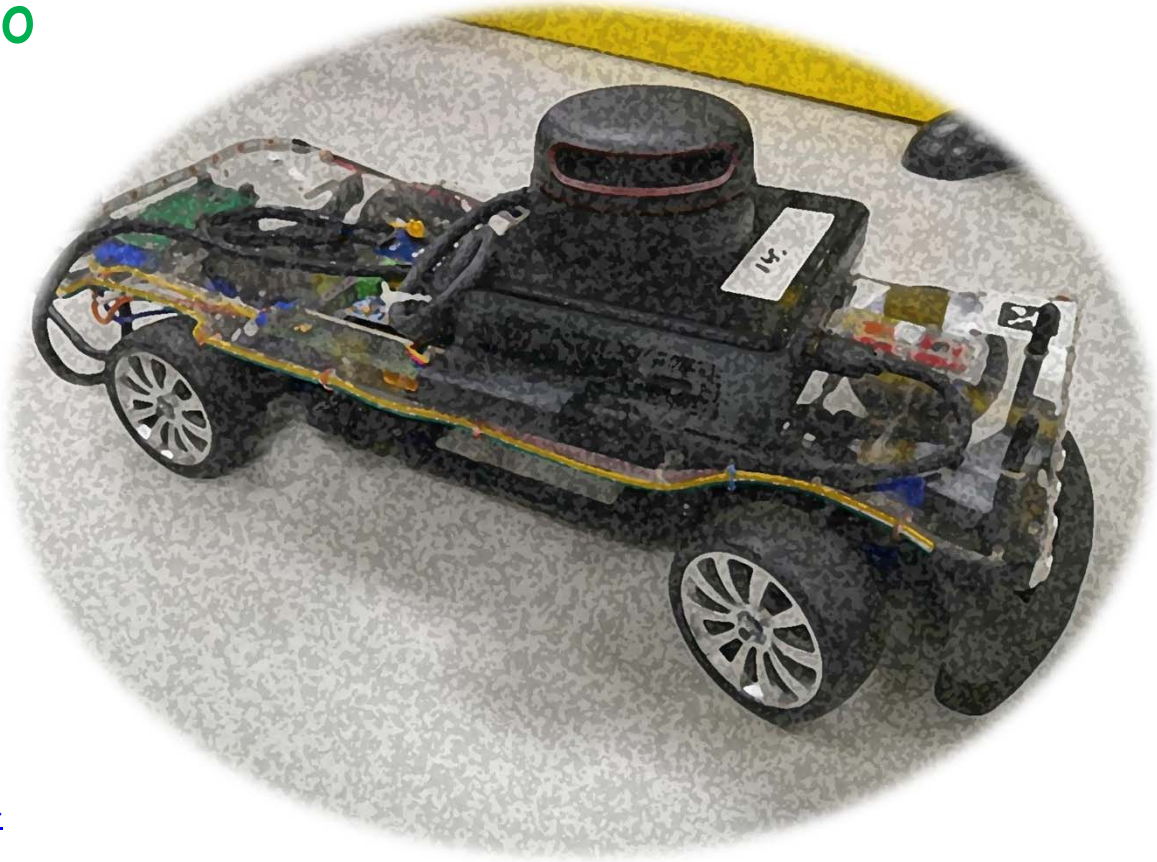
```
roger@roger-HP-Spectre-x360-Convertible:~/ckAutominy2022/src/pack01Autominy/scripts$ rosrun pack01Automi
ny program01a.py
```



Creating a Publisher Node-Simulator

The previous procedure is now described in the following video:

Dr. Roger Miranda Colorado



Researchgate:
https://www.researchgate.net/profile/Roger-Miranda-Colorado-2

Google Scholar:
https://scholar.google.com/citations?hl=es&user=NmzkrSwAAAAJ&view_op=list_works&sortby=pubdate

Pure-IPN:
https://ipn.elsevierpure.com/es/persons/roger-miranda-colorado-3

Youtube:
https://www.youtube.com/channel/UCeGT1lfNnJt695XGzEI4IxA