# POLITECNICO DI MILANO

## *AUTONOMOUS VEHICLES*

*S. Arrigoni*

# Introduction to ROS + MATLAB

POLITECNICO MILANO 1863

# Software requirements

- Matlab 2022a(recommended) / 2022b

- Type in command window: rosinit

```
>> rosinit
Launching ROS Core...
Invalid Python executable: ''. Use pyenv function to set the path to the Python executable and retry the
command.
```

- Install python (3.9.2) from https://www.python.org/ and type

```
>> pyenv('Version','/usr/bin/python3.9')
```

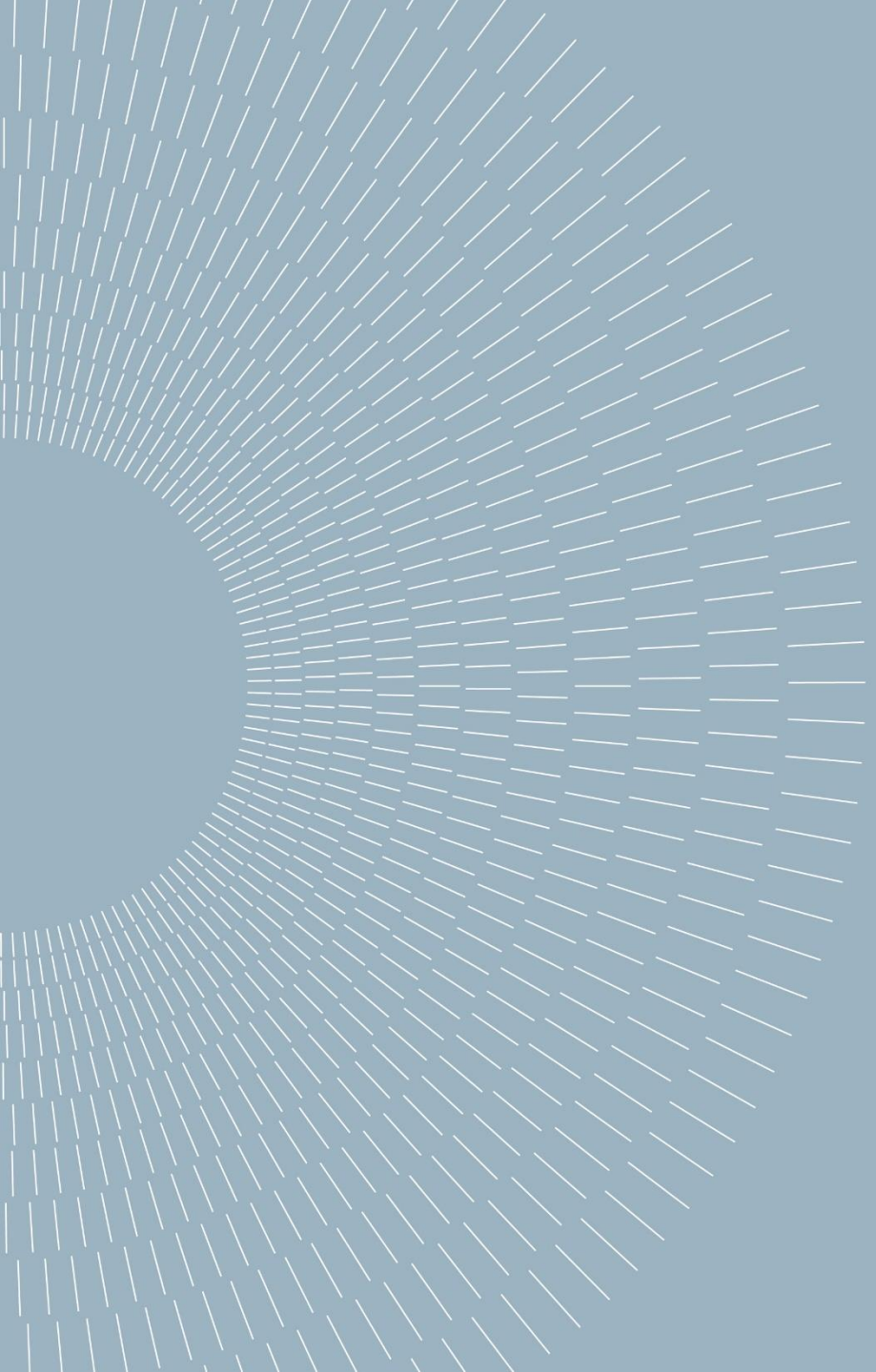
# Software requirements

- Matlab 2022a(recommended) / 2022b

- Type in command window (again): rosinit

```
>> rosinit
Launching ROS Core...
Creating a Python virtual environment...Done.
Adding required Python packages to virtual environment...Done.
Done in 0.60235 seconds.
Initializing ROS master on http://192.168.1.77:11311.
Initializing global node /matlab_global_node_70367 with NodeURI http://arrige66:59118/ and MasterURI ht
>>
```

*If you install 2022b you will have to fix a couple of things more…*



# ROS + MATLAB

## Intro

POLITECNICO MILANO 1863

# Modified Publisher (python)

```
#!/usr/bin/env python
import rospy
from std_msgs.msg import Float32

def say_something():
pub = rospy.Publisher('rndm_number', Float32, queue_size=10)
rospy.init_node('publisher_node', anonymous=False)
rate = rospy.Rate(1) # 10hz
start=rospy.get_time()
while not rospy.is_shutdown():
        timer=rospy.get_time()
        msg = timer-start
        rospy.loginfo(msg)
        pub.publish(msg)
        rate.sleep()
if __name__ == '__main__':
try:
         say_something()
except rospy.ROSInterruptException:
        pass
```

*Float type*

*New topic name and frequency*

*Loginfo: display and store in /rosout*

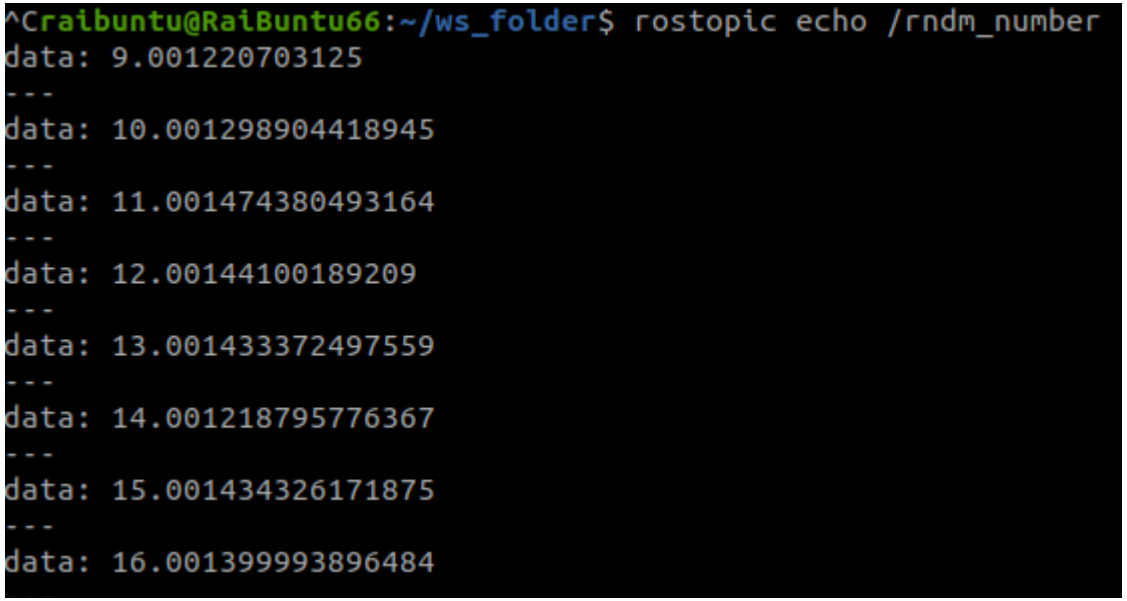
# Verify

Compile the node and run it!

Check nodes *rosnode list*

Check topics *rostopic list*

See topic *rostopic echo /rndm_number*



# Let’s connect Matlab to ROS

## Initialize Ros Master

*rosinit*

*rosinit(‘ip_address of the master if any’)*

```
>> rosinit
The value of the ROS_MASTER_URI environment variable, http://localhost:11311, will be used to connect to the ROS master.
The value of the ROS_HOSTNAME environment variable, localhost, will be used to set the advertised address for the ROS node.
Initializing global node /matlab_global_node_83857 with NodeURI http://localhost:44665/ and MasterURI http://localhost:11311.
>>
```

# Let’s connect Matlab to ROS

Check if it’s working

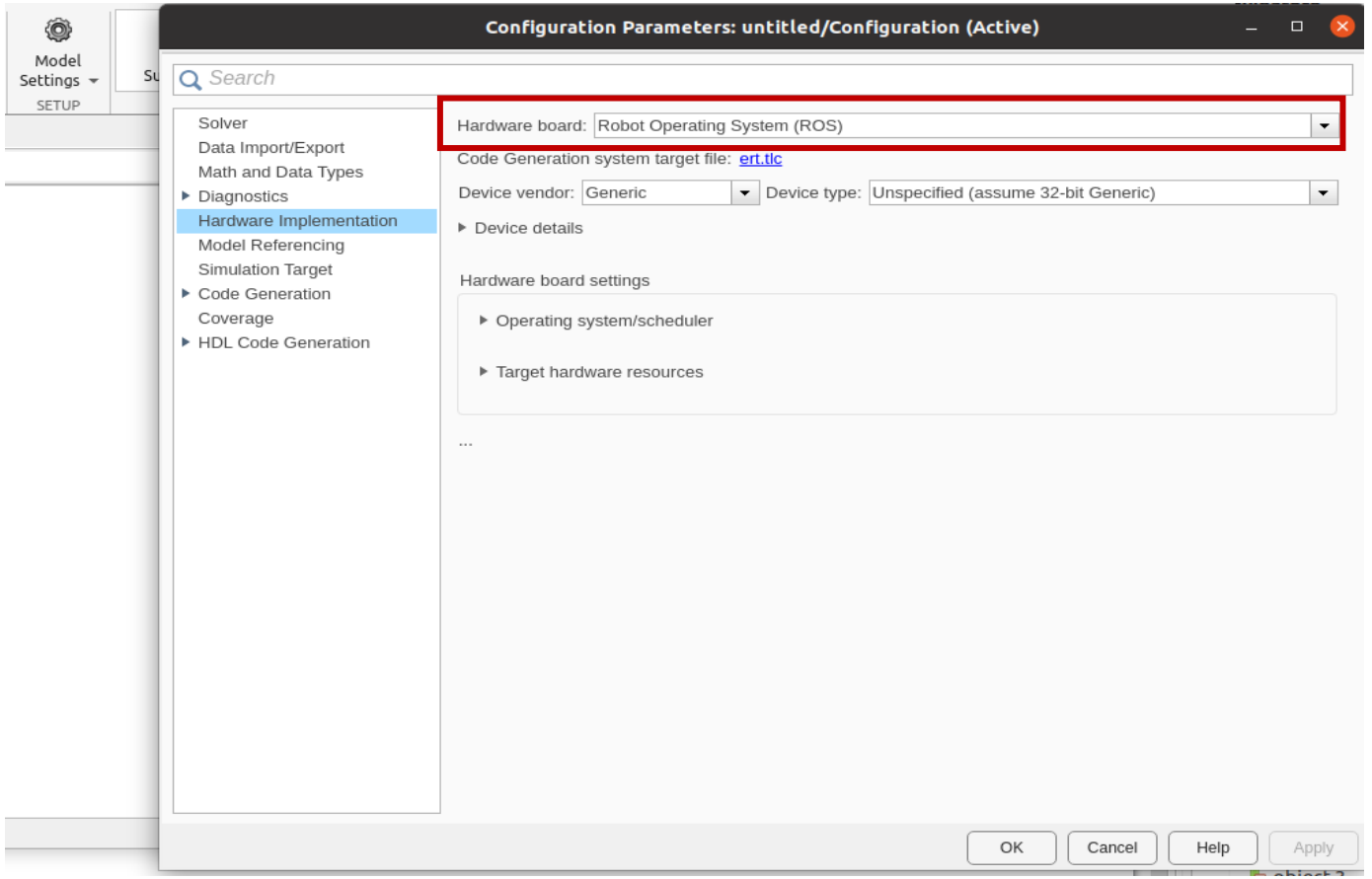
*rostopic list*

*rostopic echo*

```
>> rostopic list
/rndm_number
/rosout
/rosout_agg
/tf
>> rostopic echo /rndm_number

  Data :  212.0012664794922
---

  Data :  213.00146484375
---

  Data :  214.0012359619141
---
```

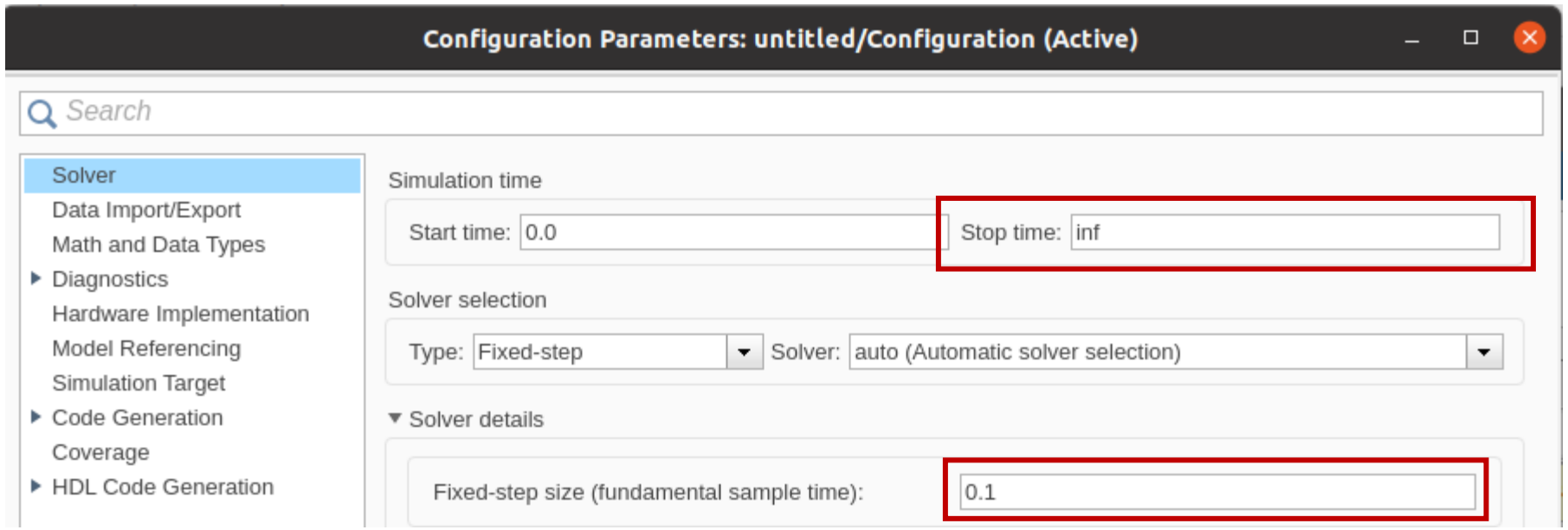
# Let’s define a Simulink node

## Initialize Simulink with ROS Toolbox 1/3



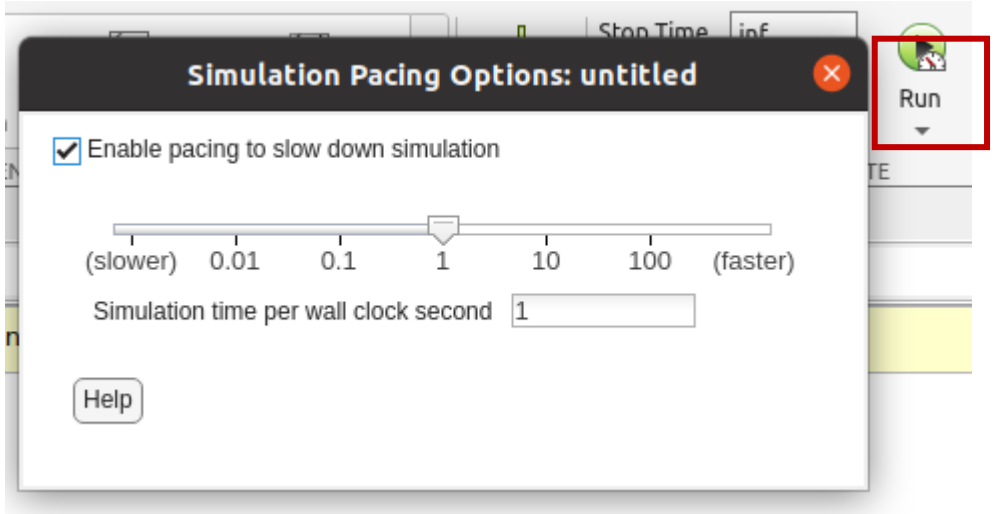
# Let’s define a Simulink node

## Initialize Simulink with ROS Toolbox 2/3



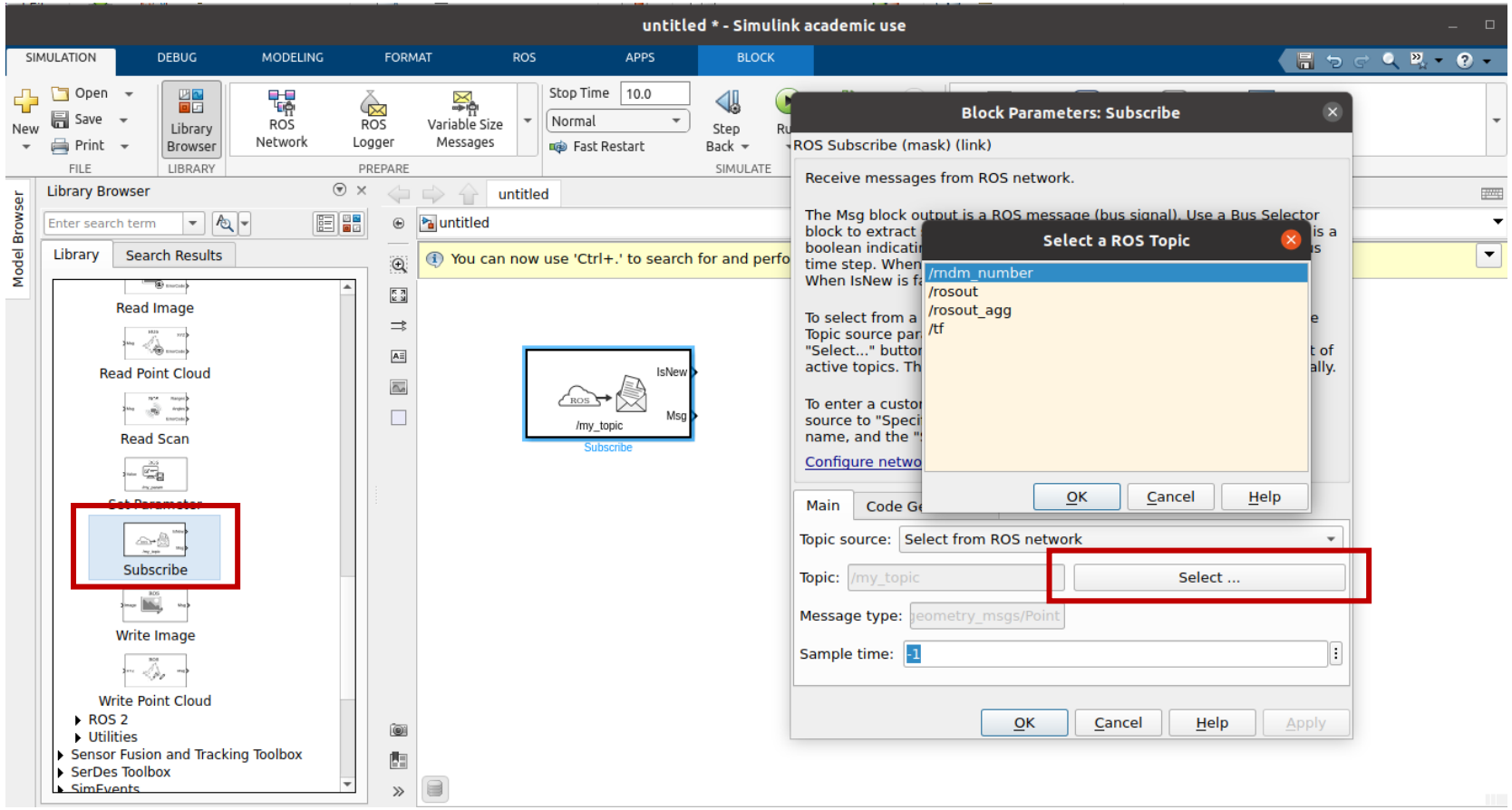
# Let’s define a Simulink node

## Initialize Simulink with ROS Toolbox 3/3



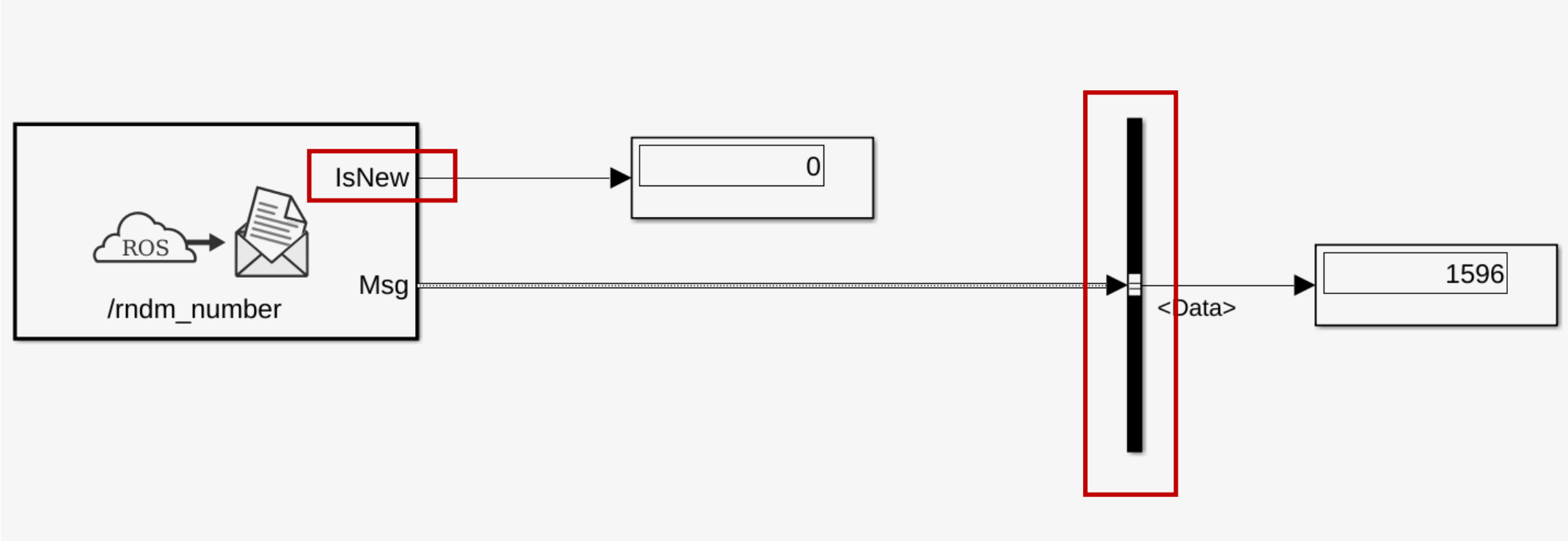
# Let’s define a publisher - subscriber

## Let’s define the subscriber



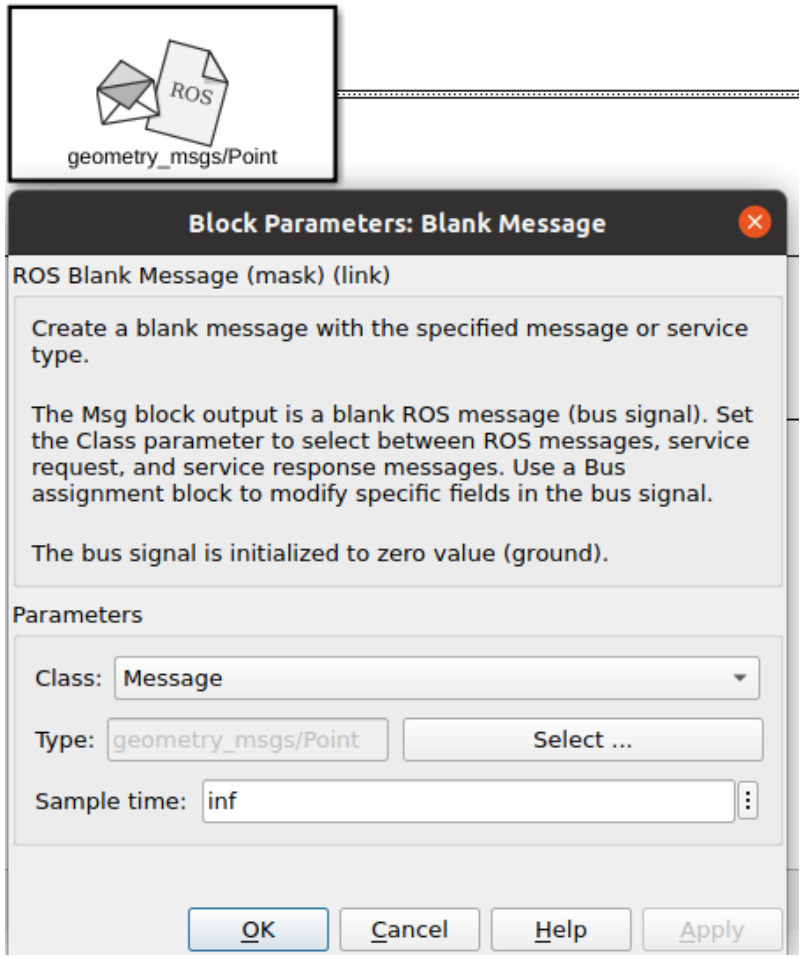
# Let’s define a publisher - subscriber

## Subscriber block



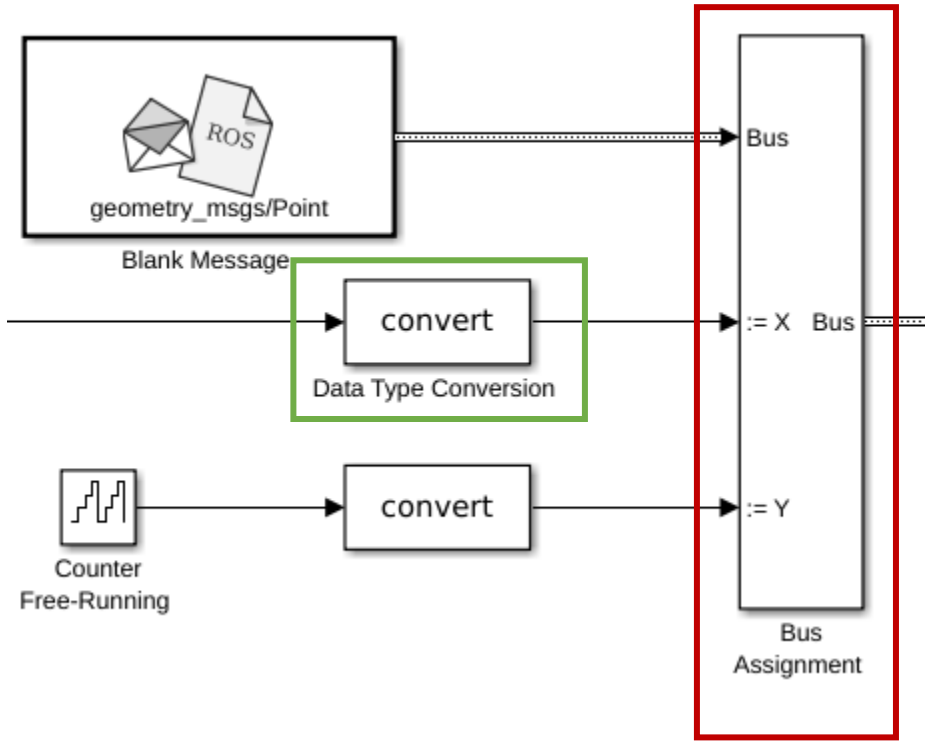
# Let’s define a publisher - subscriber

Let’s define the publisher:
blank message



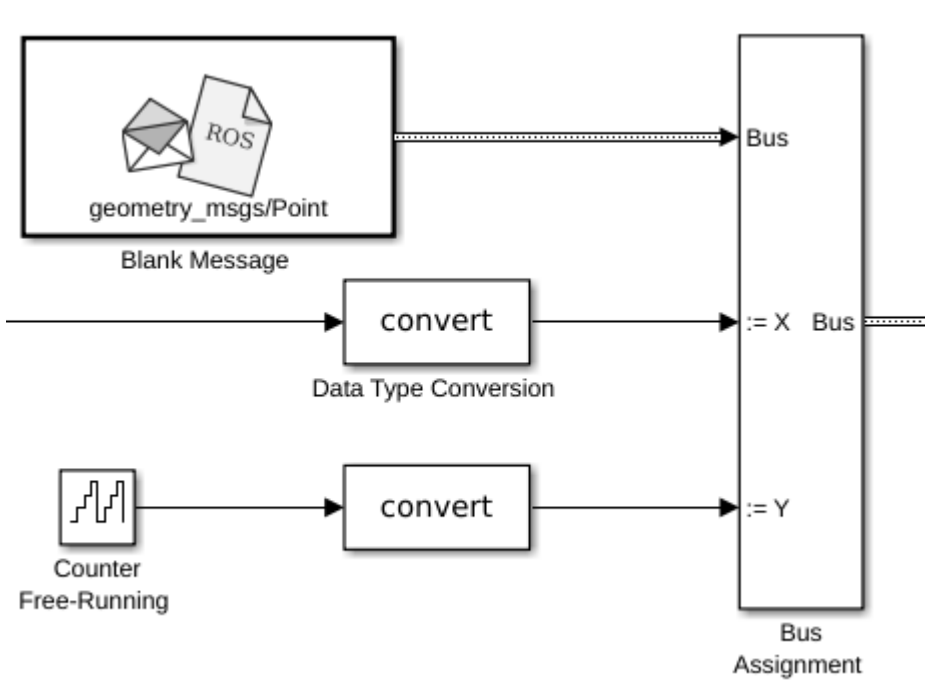
# Let’s define a publisher - subscriber

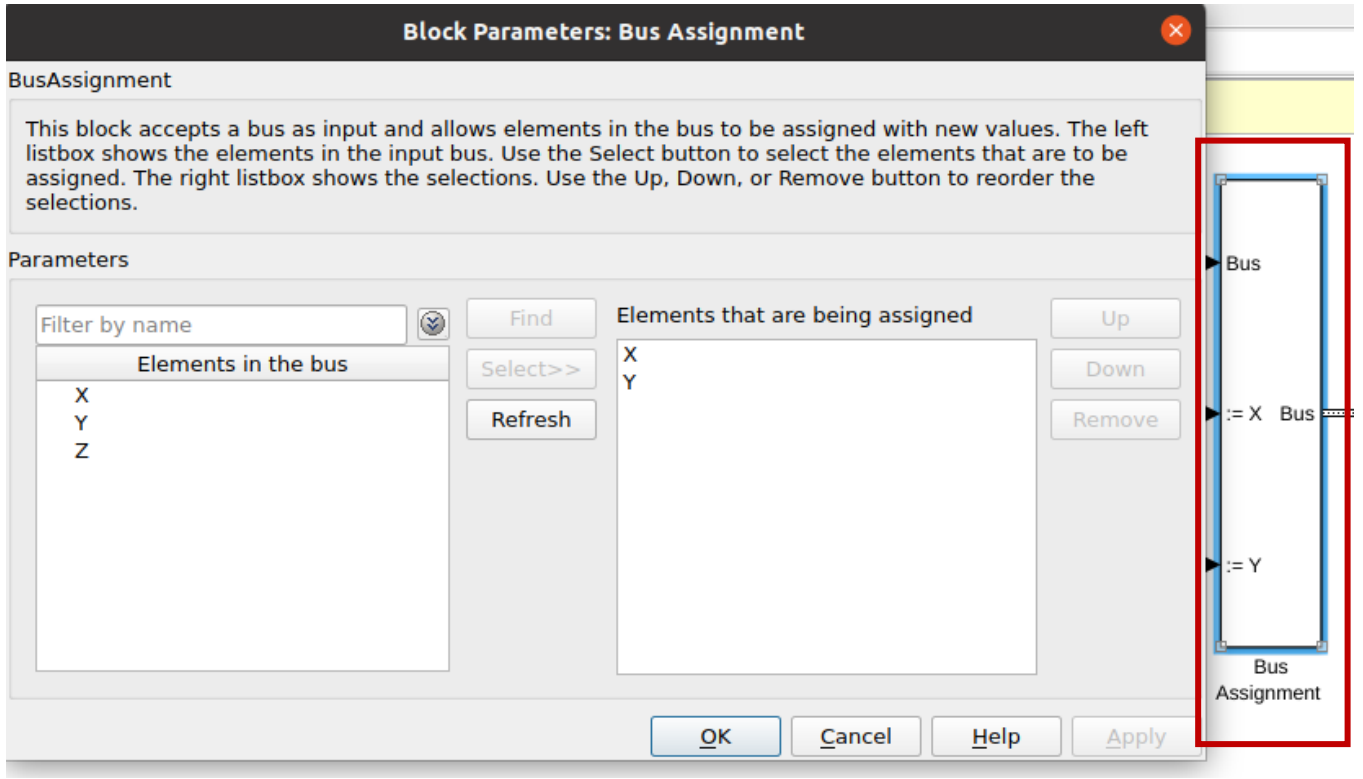
Let’s add some data in the message



# Let’s define a publisher - subscriber

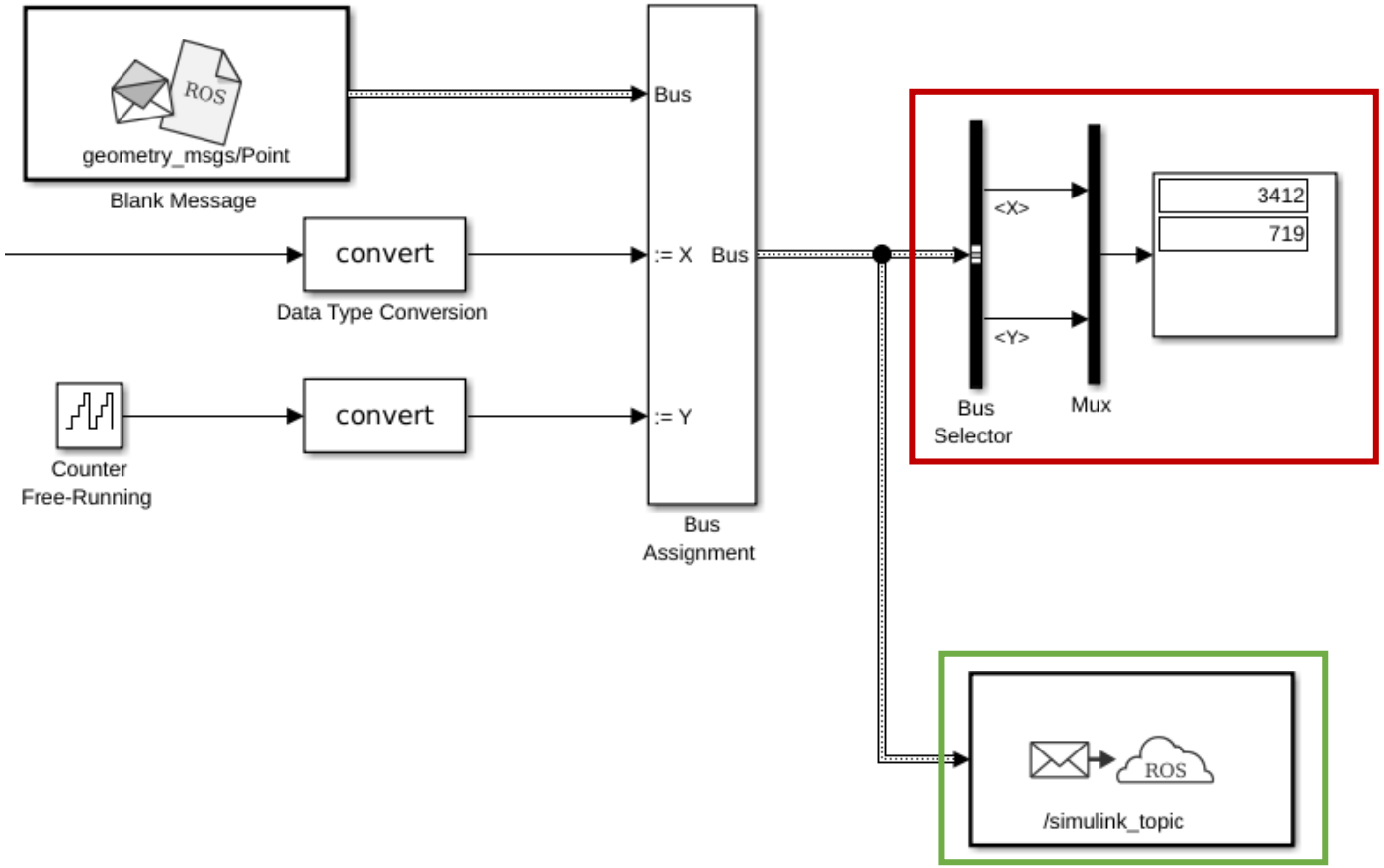
## Bus assignment



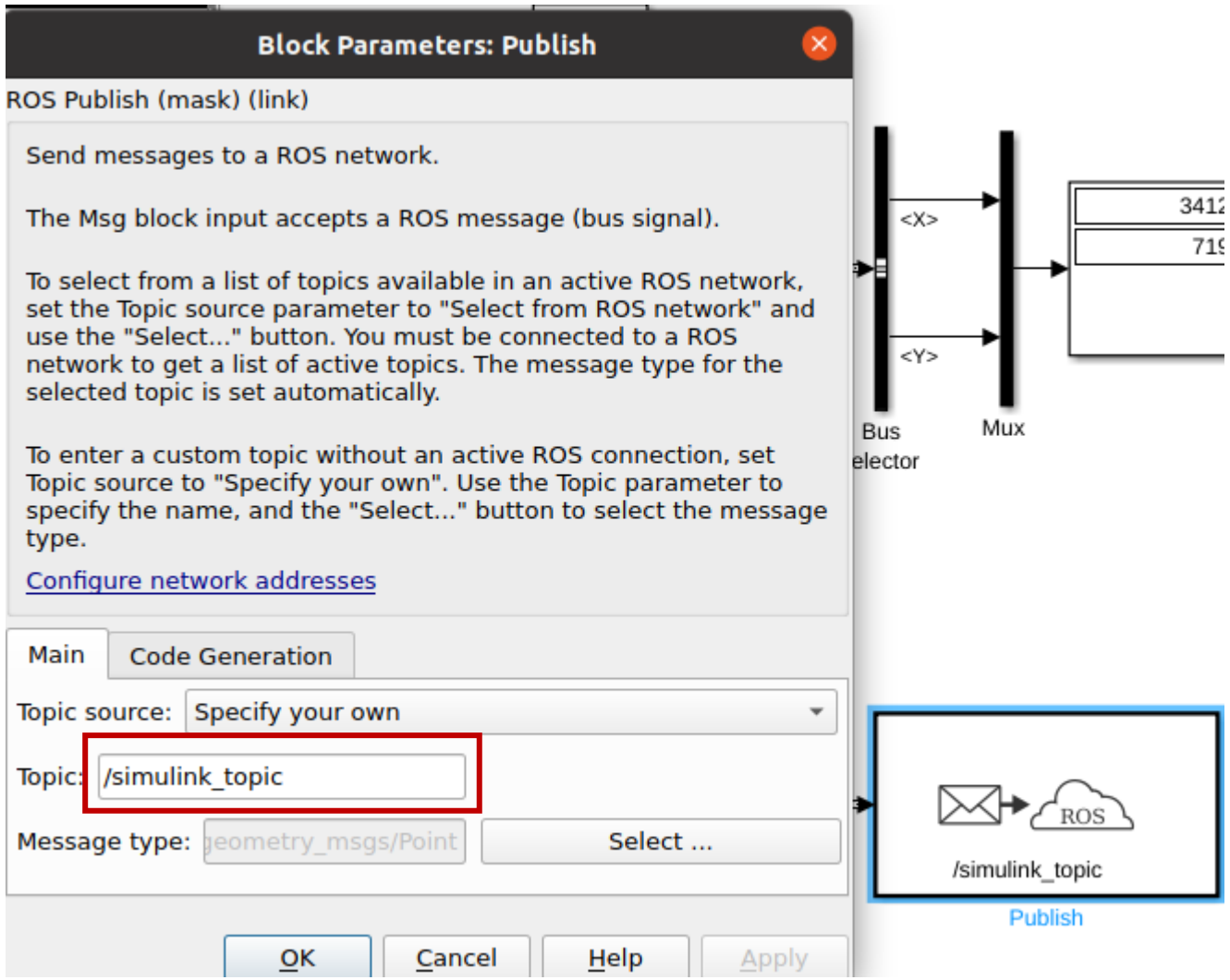
# Let’s define a publisher - subscriber

## Let’s complete the publisher



# Let’s define a publisher - subscriber

computation block



# Verify

Run the node and check it!

Check topics *rostopic list*

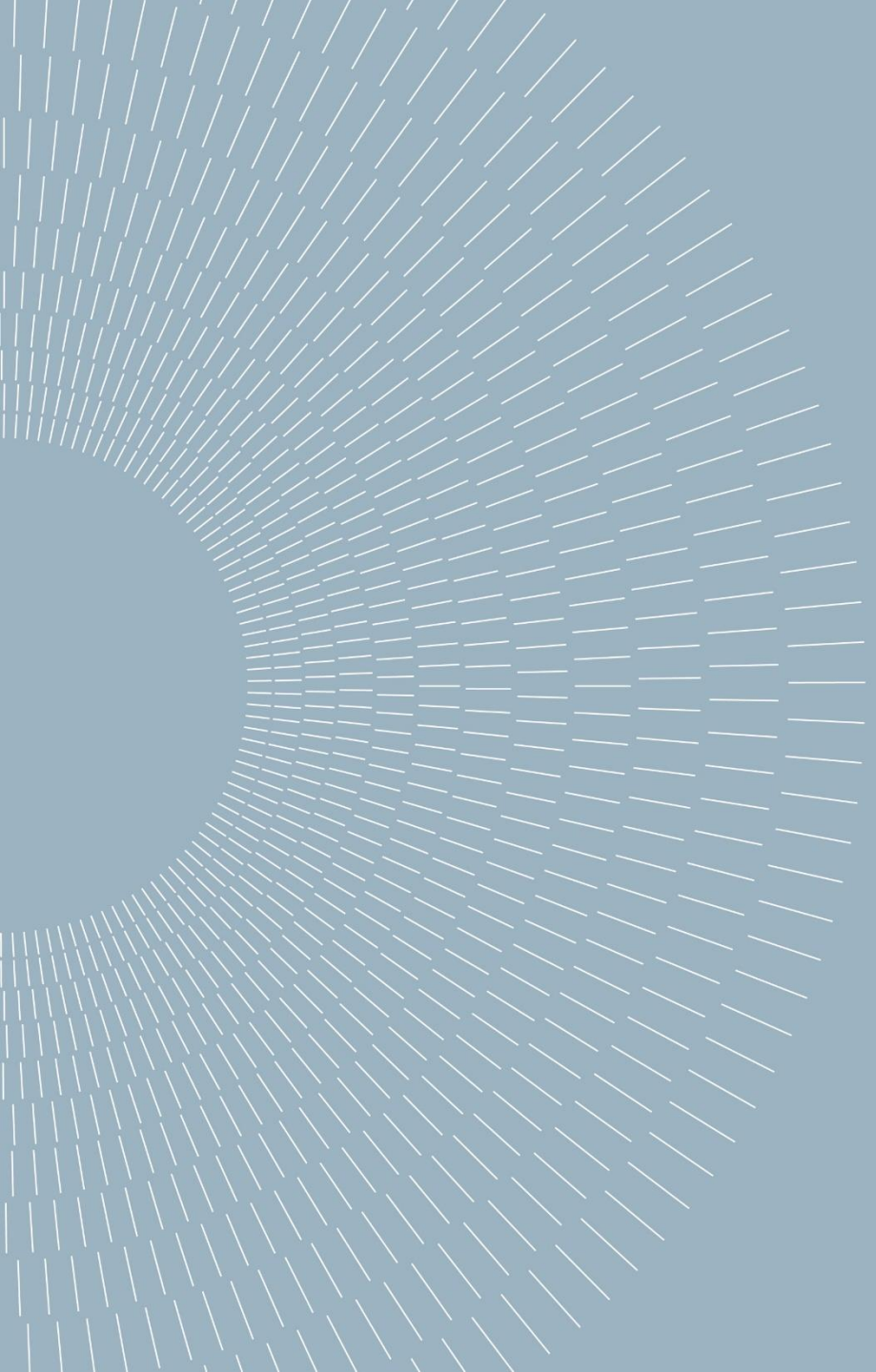
```
^Craibuntu@RaiBuntu66:~/ws_folder$ rostopic list
/rndm_number
/rosout
/rosout_agg
/simulink_topic
/tf
```

see topic *rostopic echo /simulink_topic*

```
^Craibuntu@RaiBuntu66:~/ws_folder$ rostopic echo /simulink_topic
x: 37.000980377197266
y: 216.0
z: 0.0
```

verify frequency *rostopic hz / simulink_topic*

```
raibuntu@RaiBuntu66:~/ws_folder$ rostopic hz /simulink_topic
subscribed to [/simulink_topic]
average rate: 11.398
        min: 0.000s max: 0.101s std dev: 0.02949s window: 12
average rate: 10.690
        min: 0.000s max: 0.101s std dev: 0.02220s window: 22
average rate: 10.457
        min: 0.000s max: 0.101s std dev: 0.01852s window: 32
average rate: 10.342
        min: 0.000s max: 0.101s std dev: 0.01621s window: 42
average rate: 10.273
        min: 0.000s max: 0.102s std dev: 0.01460s window: 52
average rate: 10.228
```



# ROS + GAZEBO

## Intro

POLITECNICO MILANO 1863

# Let’s initialize Gazebo

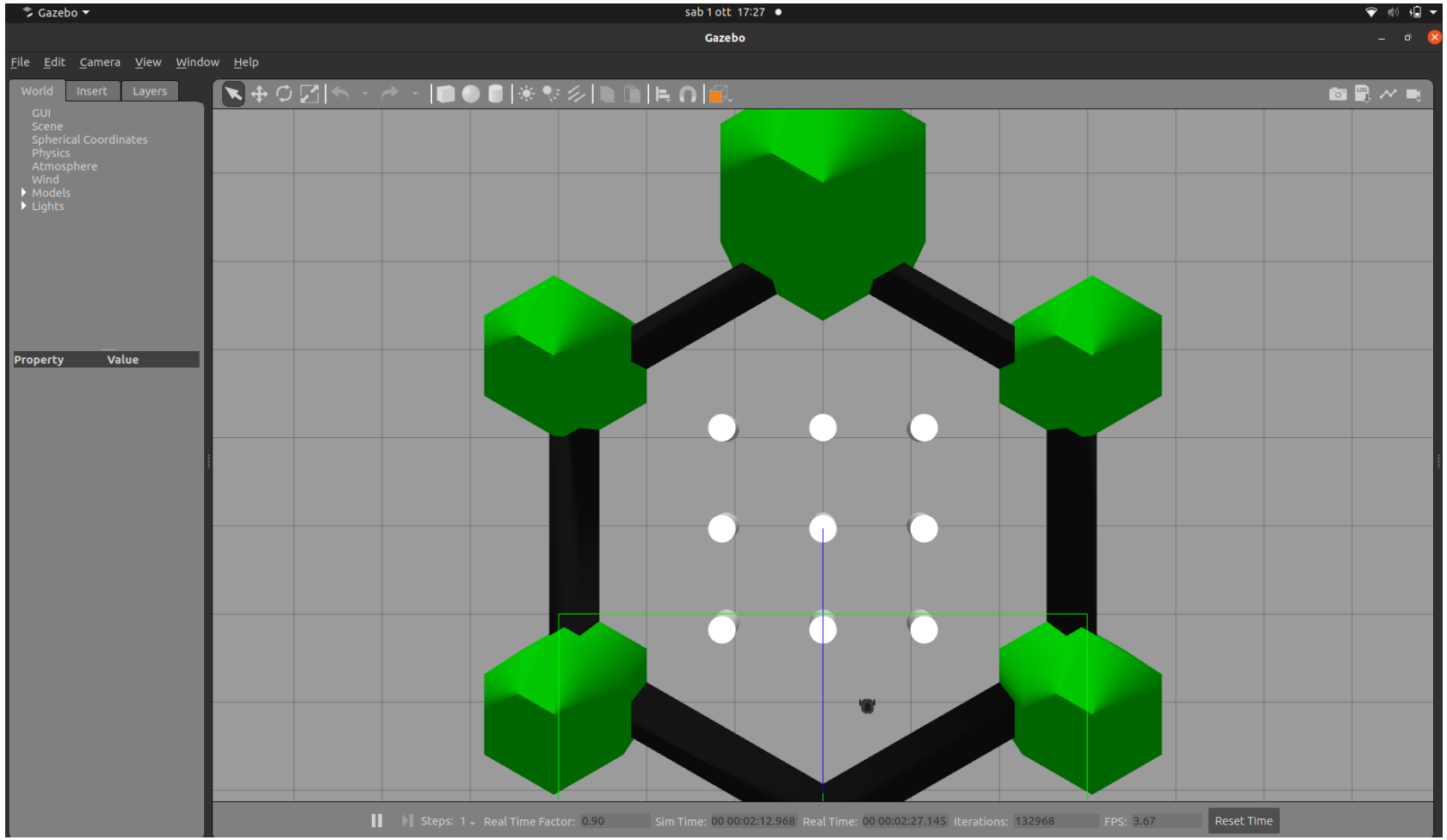
Let’s start an example in ROS 1/2

Open a new terminal and initialize gazebo

```
$ export TURTLEBOT3_MODEL=burger
$ roslaunch turtlebot3_gazebo turtlebot3_world.launch
```

# Let’s initialize Gazebo

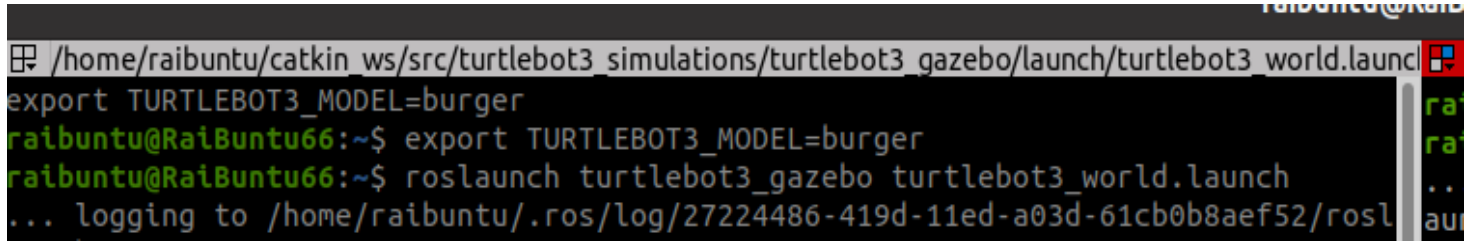
Gazebo

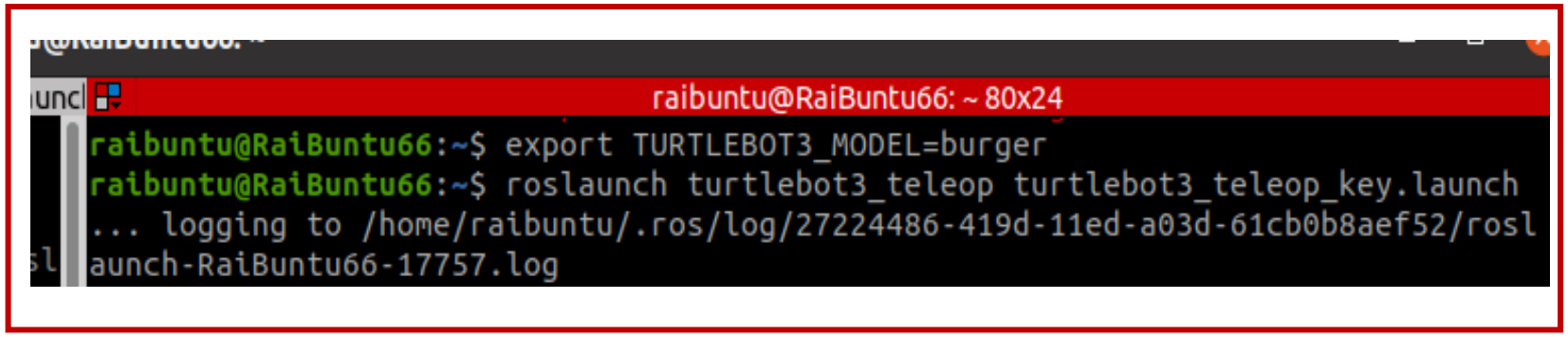


# Let’s initialize Gazebo

## Let’s start an example in ROS 2/2

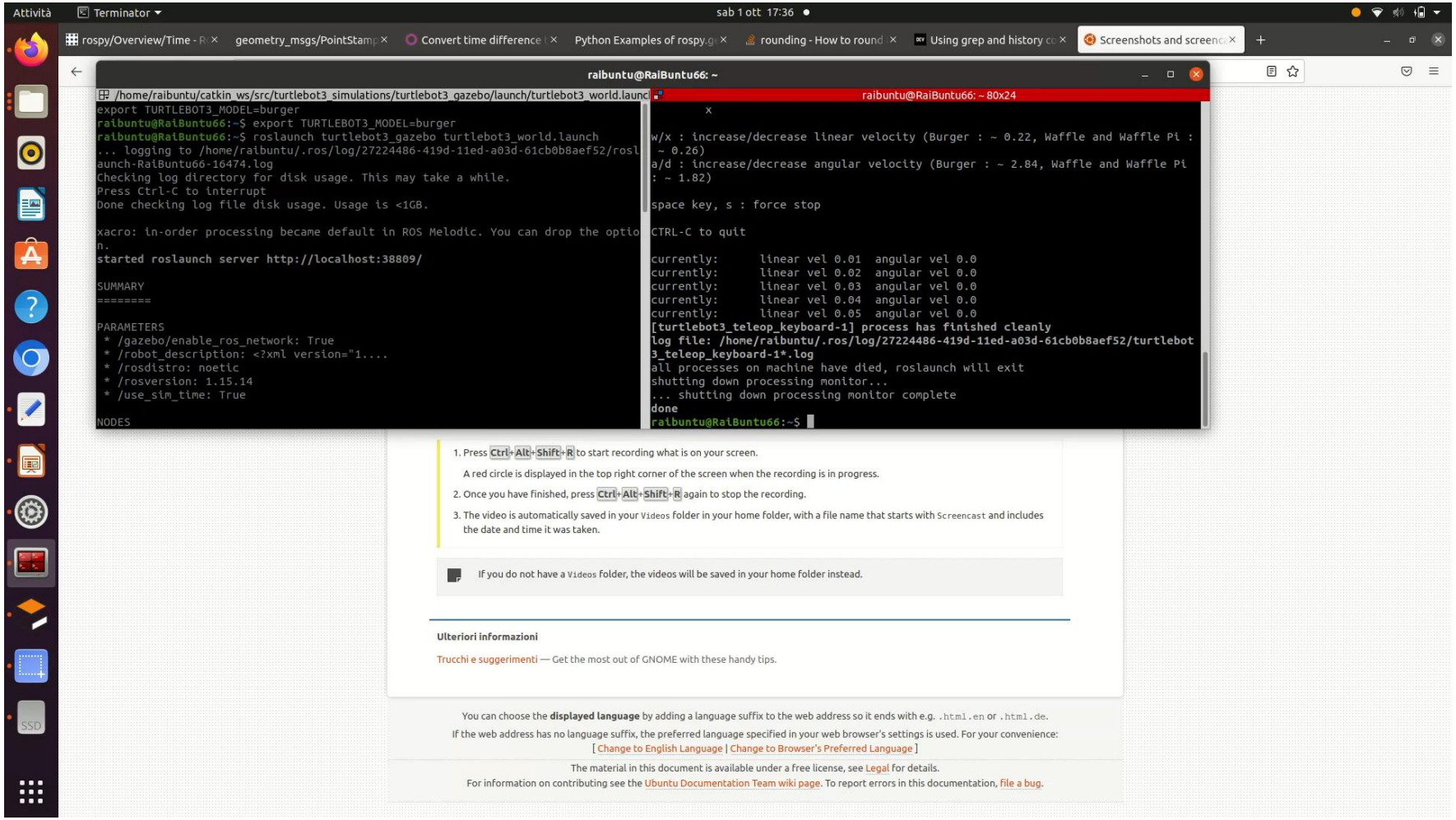
in a new terminal let’s run a control example



# Let’s initialize Gazebo

Expected result…




# That’s it for today…

# See you next time!

*S. Arrigoni*