POLITECNICO DI MILANO

AUTONOMOUS VEHICLES

S. Arrigoni

Important prerequisites

Let’s initialize Gazebo

Install examples for Gazebo before test it!

Install in your ws folder! (ws_new?)

1. 1. Install Simulation Package

The **TurtleBot3 Simulation Package** requires `turtlebot3` and `turtlebot3_msgs` packages as prerequisite. Without these prerequisite packages, the Simulation cannot be launched.
Please follow the PC Setup instructions if you did not install required packages and dependent packages.

```
$ cd ~/catkin_ws/src/
$ git clone -b kinetic-devel https://github.com/ROBOTIS-GIT/turtlebot3_simulations.git
$ cd ~/catkin_ws && catkin_make
```

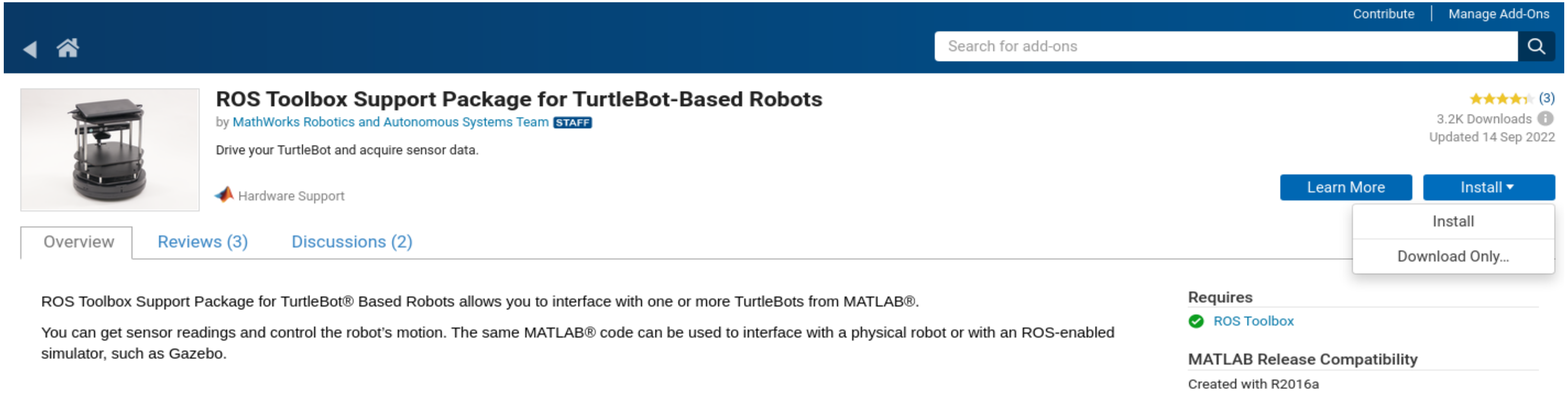
https://emanual.robotis.com/docs/en/platform/turtlebot3/simulation/

Software requirements

Not mandatory: (just if you want to collect data directly from Matlab)

- Matlab 2022a(recommended) / 2022b

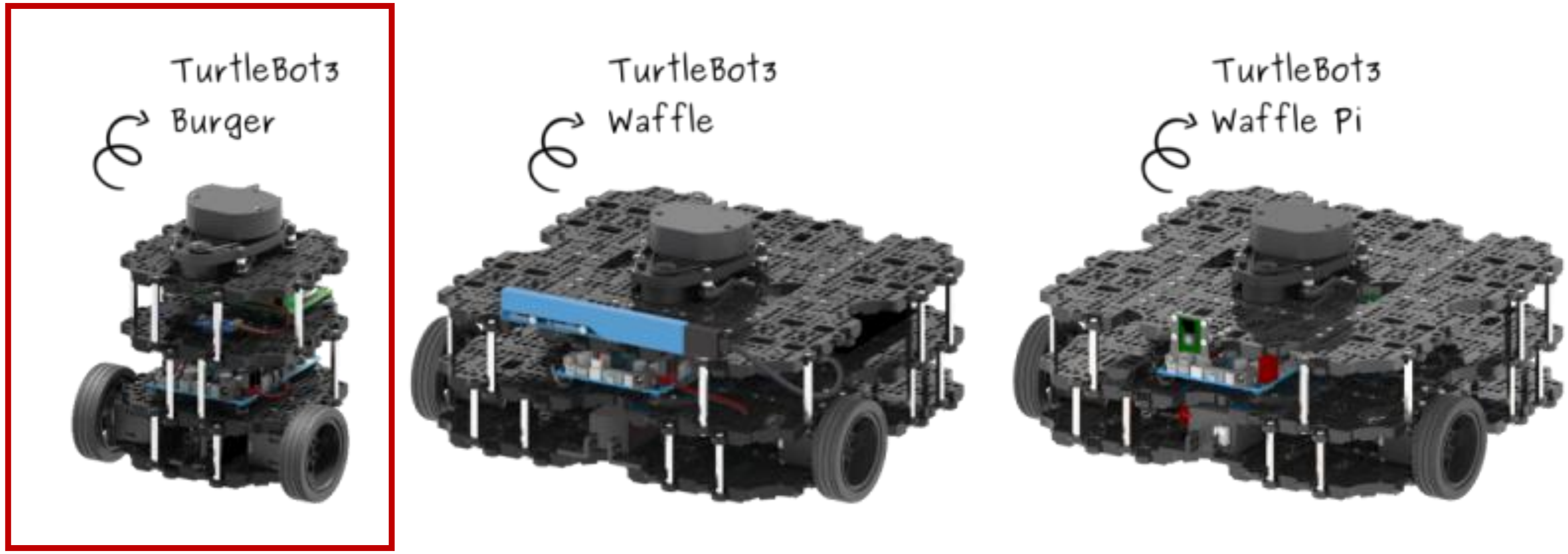
Install from add-on explorer: “ROS Toolbox Support Package for TurtleBot-Based Robots”



TURTLEBOT3

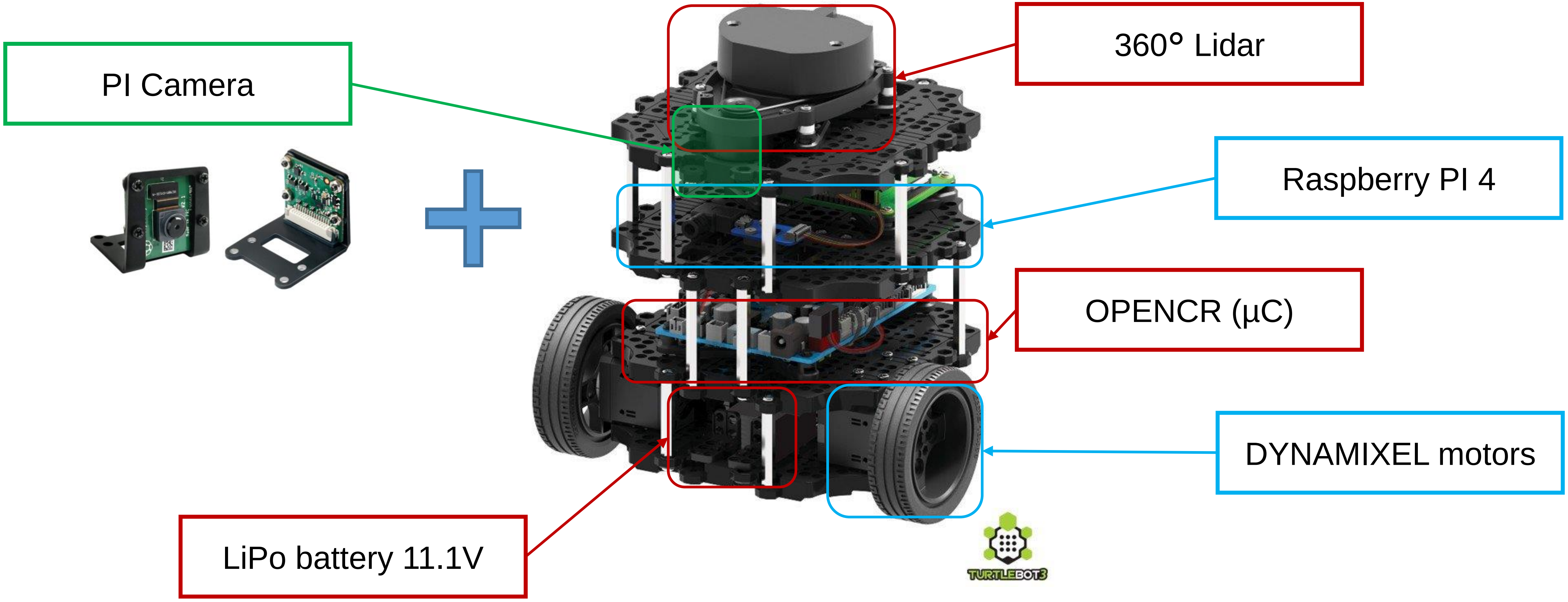
Intro

Turtlebot3



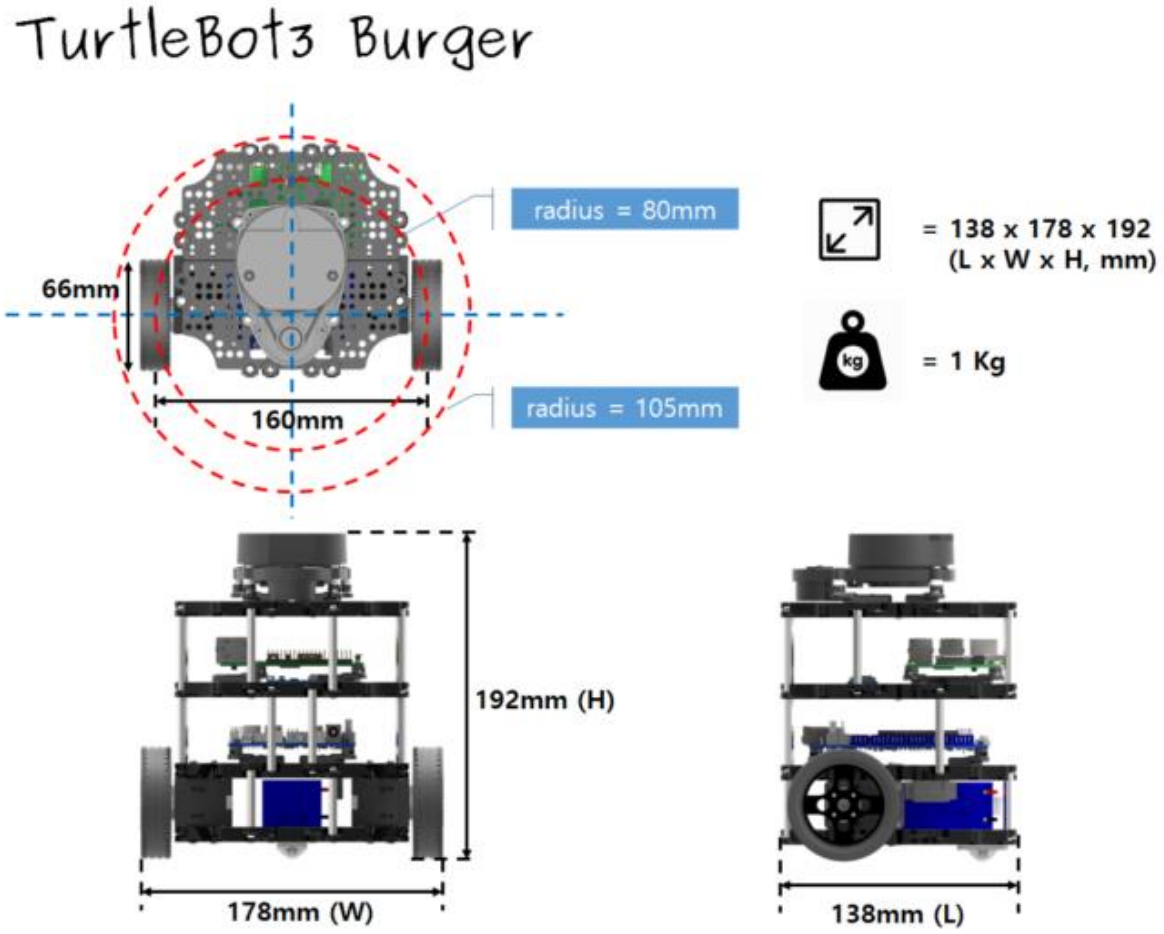
Source: [*https://emanual.robotis.com/*](https://emanual.robotis.com/)

Turtlebot3


PI Camera
360° Lidar
Raspberry PI 4
OPENCR (µC)
DYNAMIXEL motors
LiPo battery 11.1V
TURTLEBOT3

Turtlebot3

Maximum translational velocity	0.22 m/s
Maximum rotational velocity	2.84 rad/s (162.72 deg/s)
Maximum payload	15kg
Size (L x W x H)	138mm x 178mm x 192mm
Weight	1kg
Threshold of climbing	10 mm or lower
Expected oper / charg time	2h 30m
SBC (Single Board Computers)	Raspberry Pi 4
Actuator	XL430-W250
LDS(Laser Distance Sensor)	360 Lidar LDS-02
Camera	PI Camera
IMU	Gyroscope 3 Axis Accelerometer 3 Axis
Battery	Lithium polymer 11.1V 1800mAh / 19.98Wh 5C



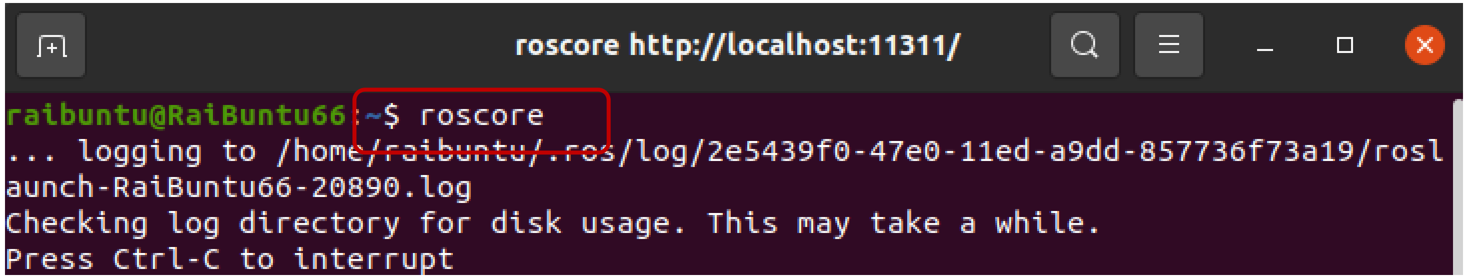
Introduction to

ROS + MATLAB

GAZEBO

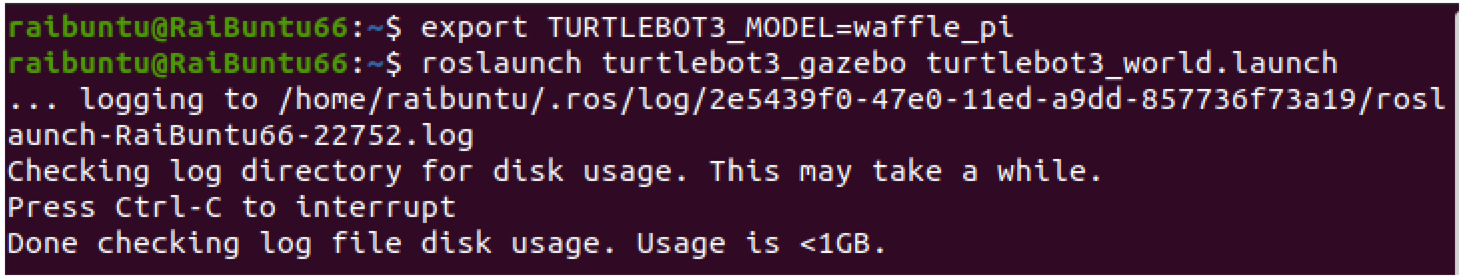
TURTLEBOT3

Let’s initialize ROS + Gazebo environment



Initialize: Ros

Initialize: Gazebo + Turtlebot3



export TURTLEBOT3_MODEL=waffle_pi
roslaunch turtlebot3_gazebo turtlebot3_world.launch

Let's use:

Connect to virtual robot

ipaddress = 'localhost'; % # IP of real turtlebot / or other PC
tbot = turtlebot(ipaddress)

```
>> tbot= turtlebot('localhost')
The value of the ROS_HOSTNAME environment variable, localhost, will be used to set the advertised address for the ROS node.

tbot =

  turtlebot with properties:

           Velocity: [1x1 struct]
         ColorImage: [1x1 struct]
          GrayImage: [1x1 struct]
         DepthImage: [1x1 struct]
         PointCloud: [1x1 struct]
          LaserScan: [1x1 struct]
           Odometry: [1x1 struct]
      OdometryReset: [1x1 struct]
                IMU: [1x1 struct]
    TransformFrames: {2x1 cell}
         TopicNames: {30x1 cell}
```

Let’s use:

#1: read Odometry

position and orientation as [x y z] coordinates and [yaw pitch roll] angles

```
>> odom = getOdometry(tbot)

odom =

  struct with fields:

       Position: [-1.9999 -0.4996 -0.0010]
    Orientation: [0.0023 0.0032 -3.3219e-06]
```

Let’s use:

#2: read camera : Let’s make an example!

```
%% assign topic to control it!
tbot.Velocity.TopicName = '/cmd_vel';
%% camera
desiredRate = 2;
rate = rateControl(desiredRate);


reset(rate)
for i = 1:20
setVelocity(tbot,0.01,0.5)
img = getColorImage(tbot,0);
imshow(img)
waitfor(rate);
end
```

Define *topic_name* to control turtlebot

Define control / visualization frequency

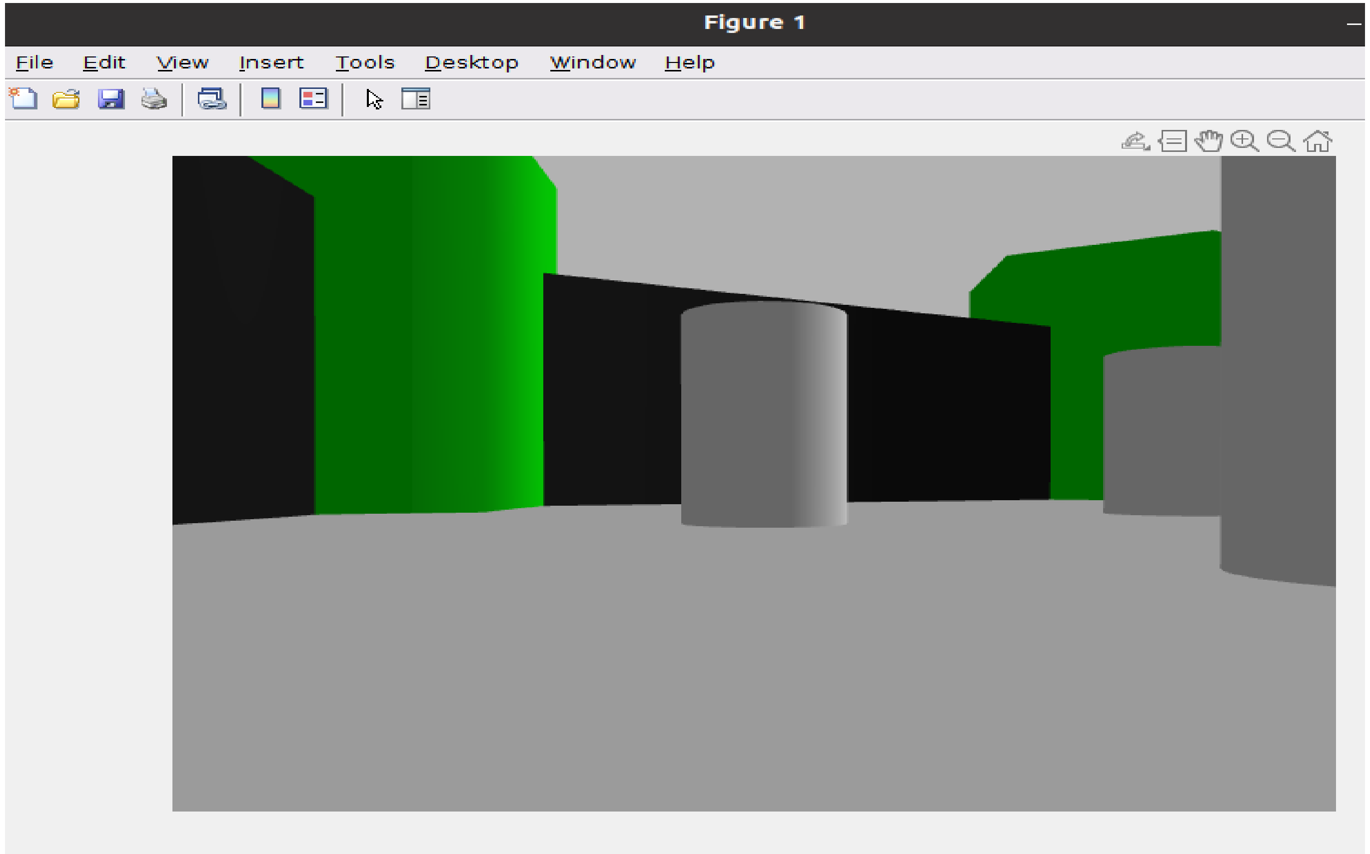
Provide control inputs to turtlebot

Acquire / visualize camera

Let’s use:

#2: read camera : Let’s make an example!

expected result



Let’s use:

MATLAB®

#3: read laser scanner: Let’s make an example!

```
[scan,scanMsg] = getLaserScan(tbot);
% scan = range [meters] and angle [radiants]
% scanMsg = object of ROS message
plot(scanMsg)
```

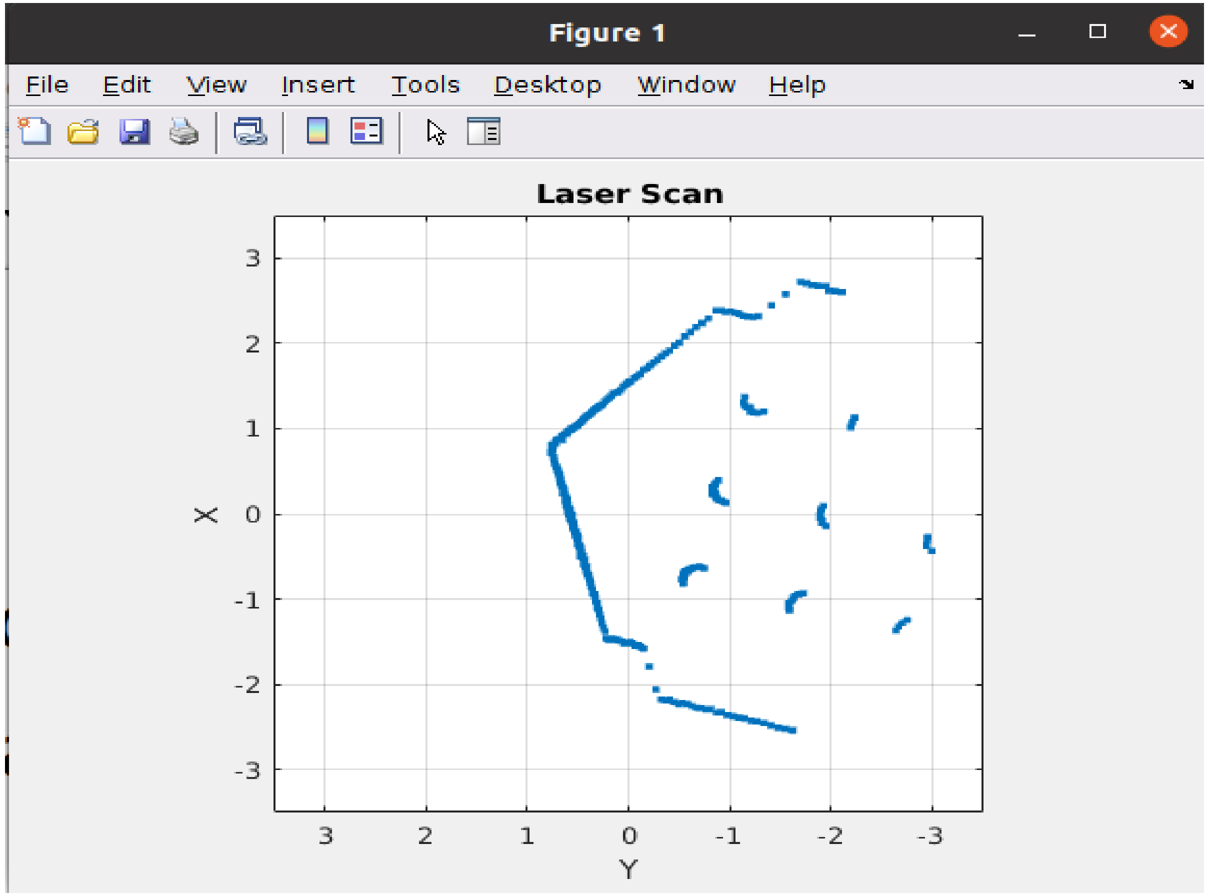
acquire scan data

visualize results (x,y)

Let’s use:

#3: read laser scanner: Let’s make an example!

expected result



Let's use:

SIMULINK®

Initialize ROS in MATLAB

rosinit

Create a new simulink file and initialize it to ROS (*last lesson*)

Let’s use:

#1 read generic sensors:

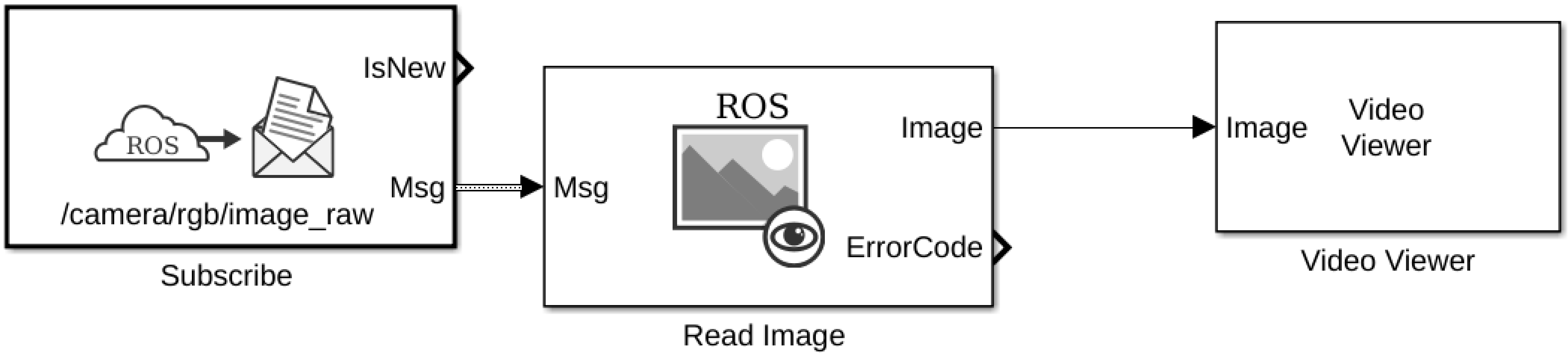
✓ You can always read all data coming from ROS topics (*subscriber*)

Let’s analyze how to deal with specific sensor data from Turtlebot3:

- ✓ Camera
- ✓ Laser scanner

Let's use:

#2: camera



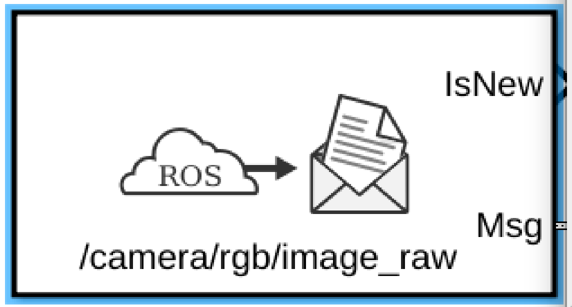
Read ROS topic

Convert it into image

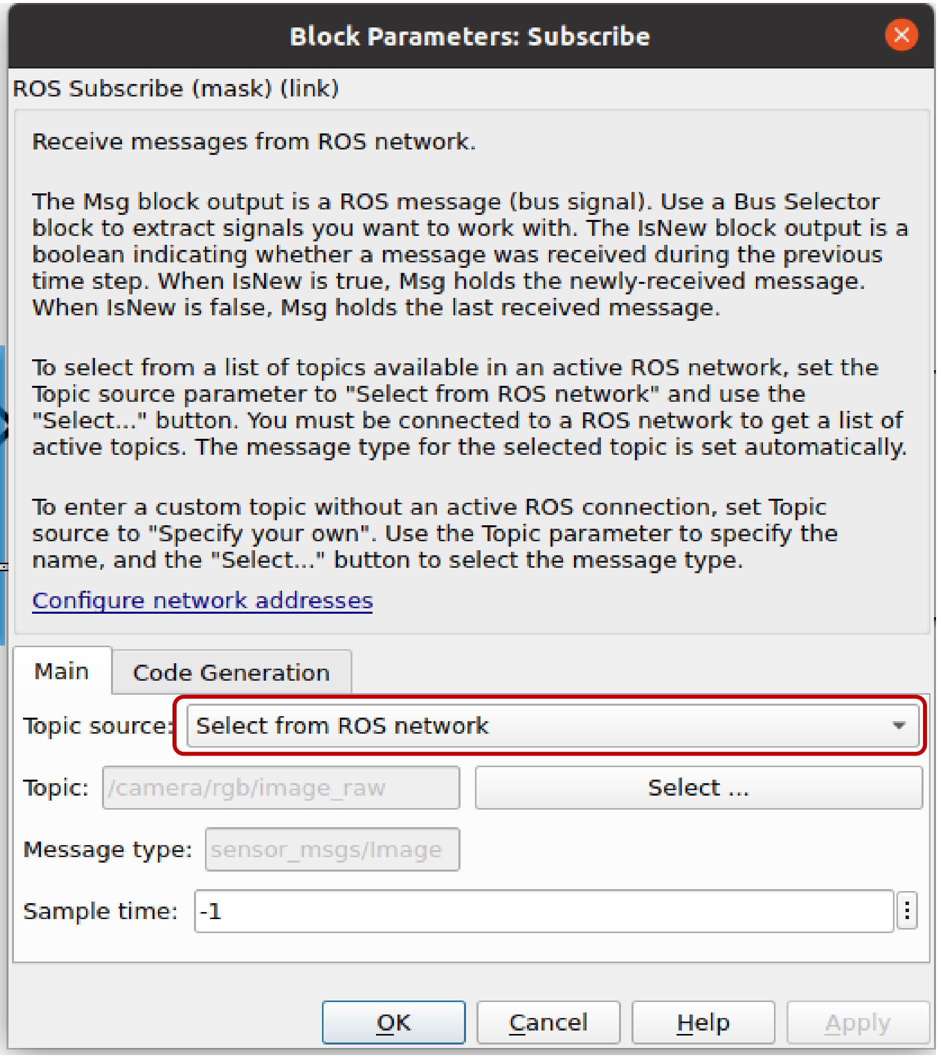
Visualize it

Let's use:

#2: camera

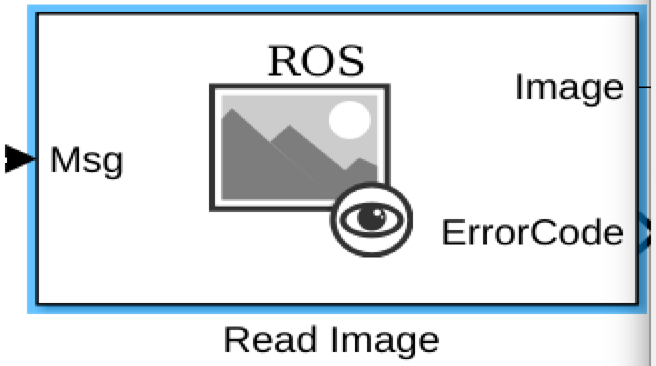


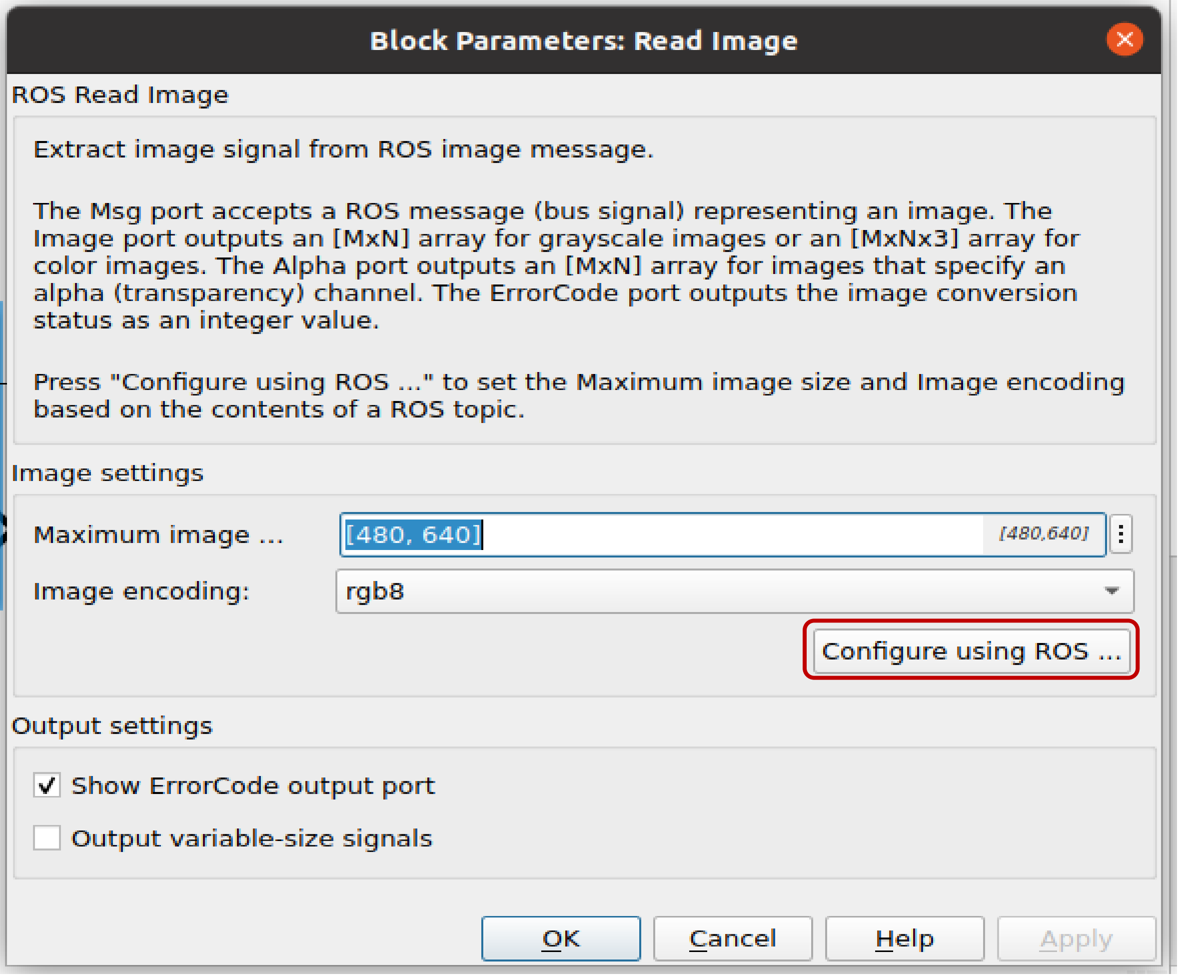
Subscribe



Let's use:

#2: camera



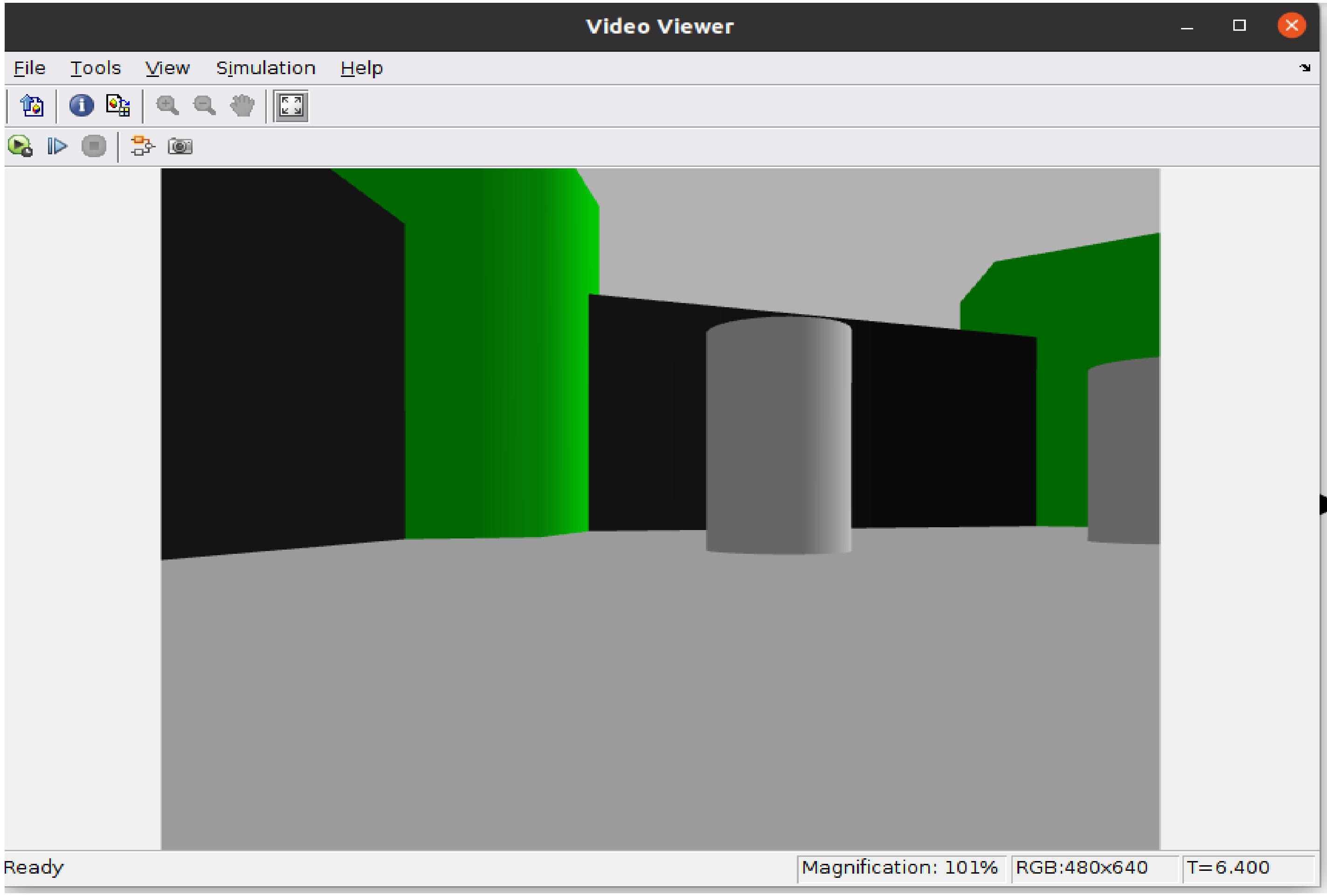


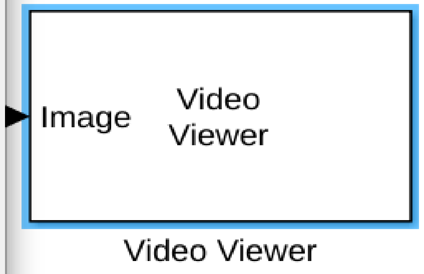
Let’s use:

SIMULINK®

#2: camera

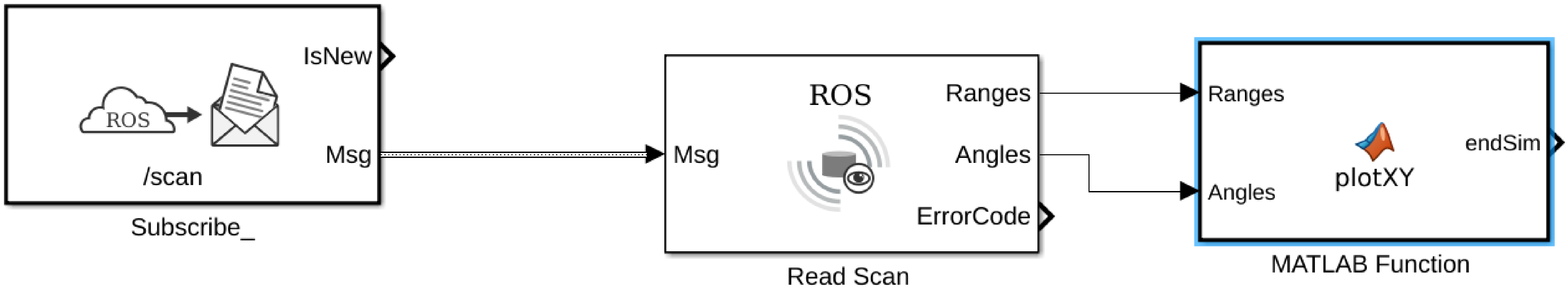
expected result





Let’s use:

#3: laser scanner



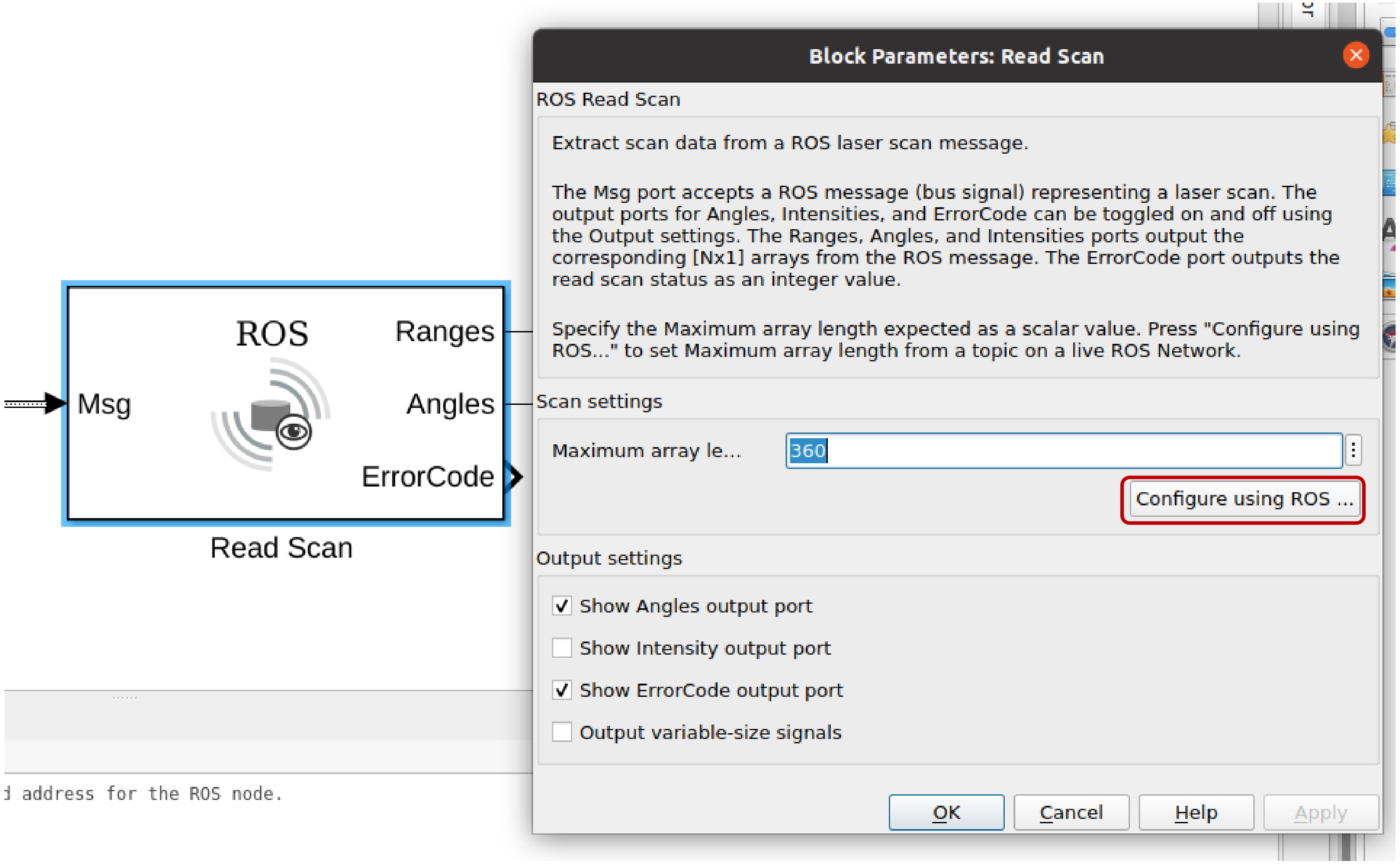
Read ROS topic

Extract ranges / angles

Visualize them in x-y

Let's use:

#3: laser scanner



Let’s use:

SIMULINK®

#3: laser scanner

From data: if != 0

Poject in x-y plane

Plot it in a figure

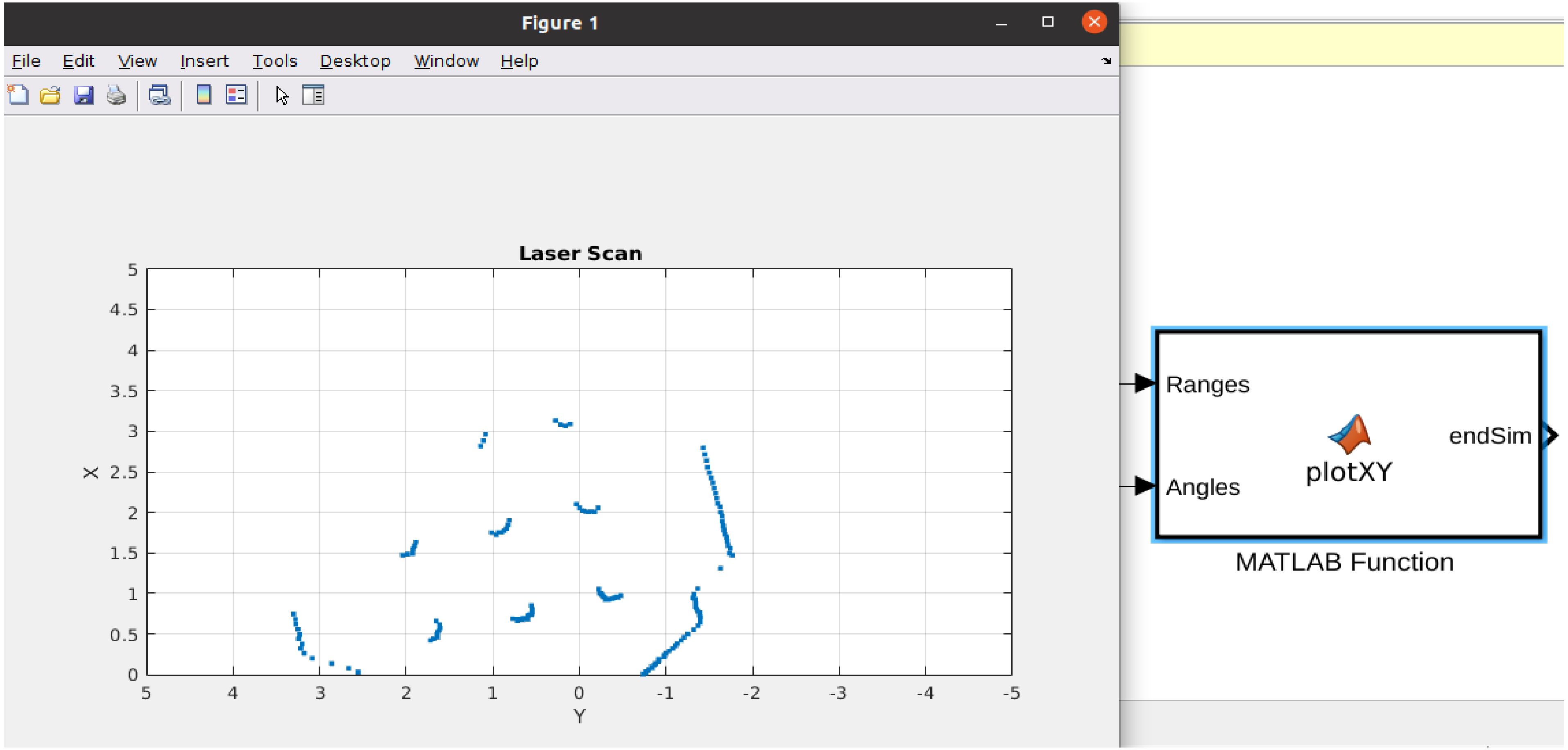
example_2 ▸ MATLAB Function

```matlab
function endSim = plotXY(Ranges, Angles)
% Read Scan block will output all zeros until the Subscriber blocks
% outputs a message
endSim = any(Ranges>0);% =1 if at least one range is not zero
if endSim
    x = Ranges.*cos(Angles);
    y = Ranges.*sin(Angles);
    % only return finite values
    xValid = x(isfinite(x));
    yValid = y(isfinite(y));
    plot(xValid,yValid,marker=".",LineStyle="none", MarkerSize=8)
    title('Laser Scan')
    xlabel("X")
    ylabel("Y")
    set(gca, 'YDir','reverse')
    grid on
    axis equal
    ylim([-5 5])
    xlim([0 5])
    view([90 -90])
end
```

Let's use:

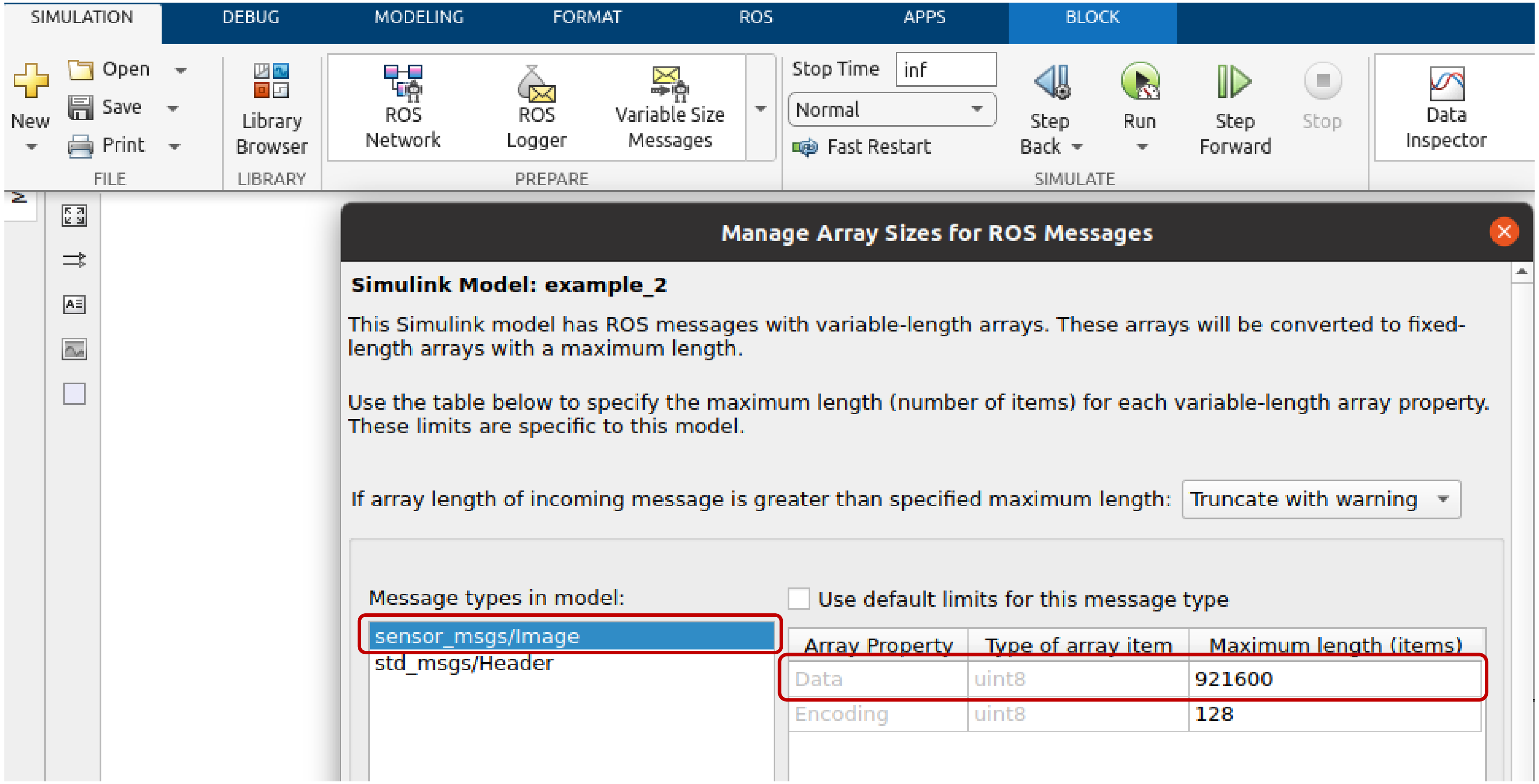
#3: laser scanner

expected result



Let's use:

#fix for both data processing#



Use of the Bag

Let’s save all messages!

Save all messages

rosbag record -a

Let’s replay them!

- **terminal**

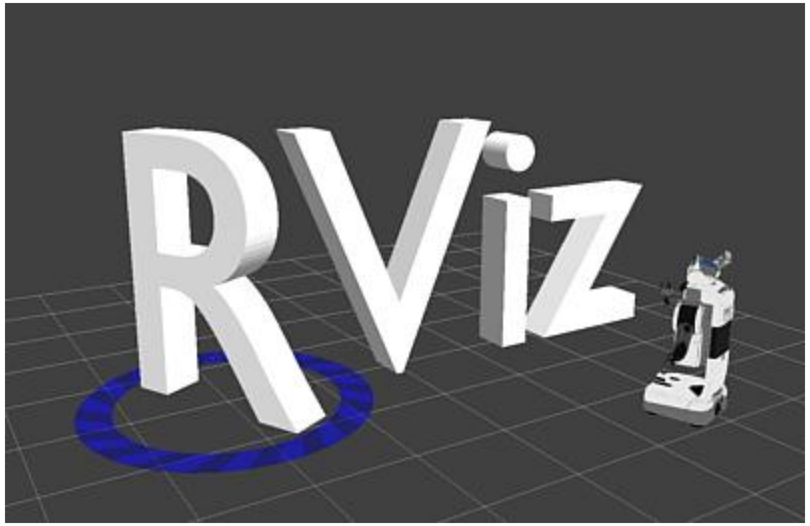
  rosbag play nome.bag

- **graphics**

  rqt_bag

Let’s use Rviz

Rviz = a 3D visualizer for ROS



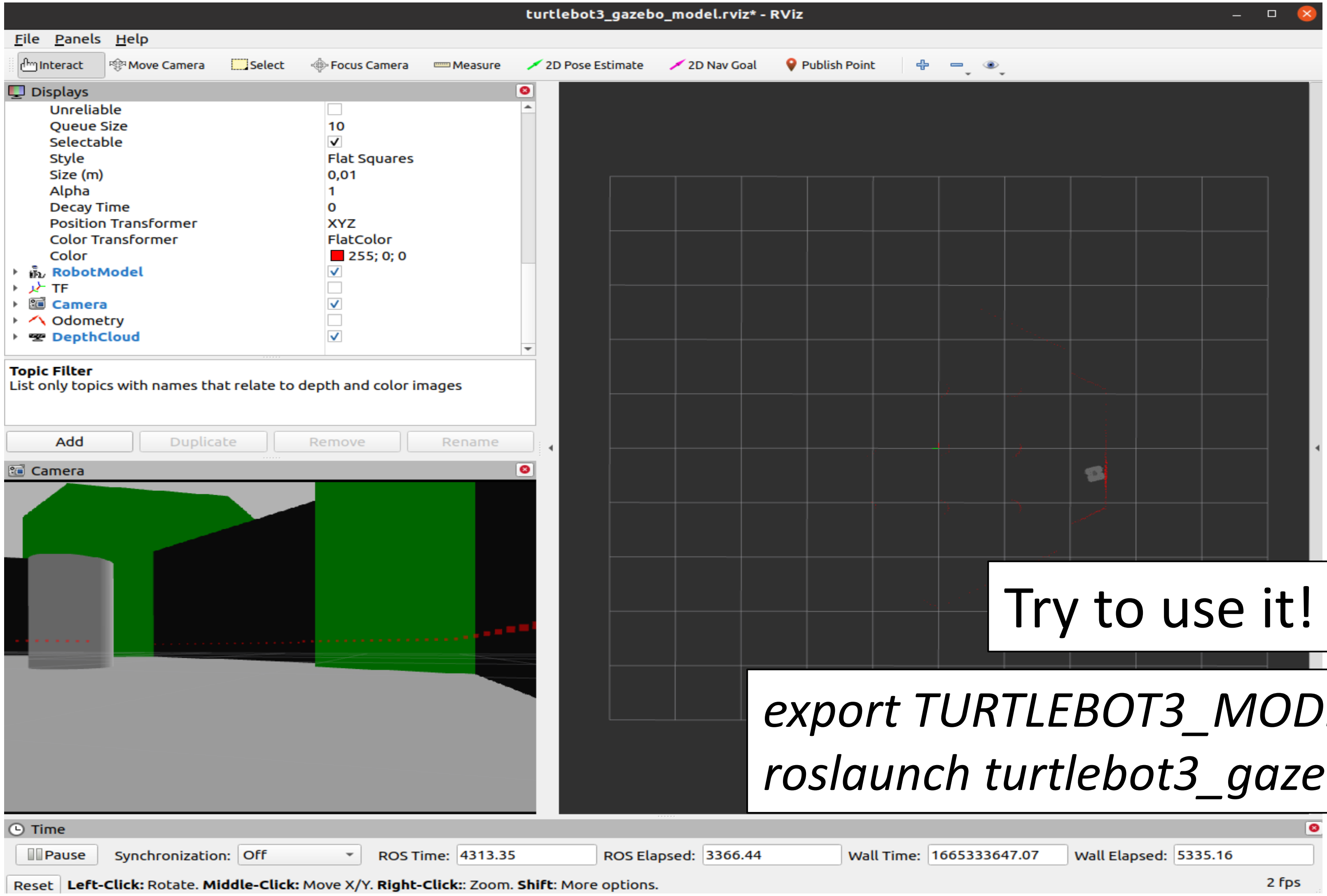
A powerful 3D visualization tool for ROS that allows to:

- view the **robot model**
- **display** and/or log sensor information
- **replay** the logged sensor information

It can displays:

- 3D sensor data from stereo cameras, lasers, Kinects, and other 3D devices in the form of **point clouds** or **depth images**;
- 2D sensor data from webcams, RGB cameras, and 2D laser rangefinders in the form of **image data**.

Let’s use Rviz



Try to use it!

export TURTLEBOT3_MODEL=waffle_pi
roslaunch turtlebot3_gazebo turtlebot3_gazebo_rviz.launch

That’s it for today…

See you next time!

S. Arrigoni

POLITECNICO
MILANO 1863