Lien du tutoriel :

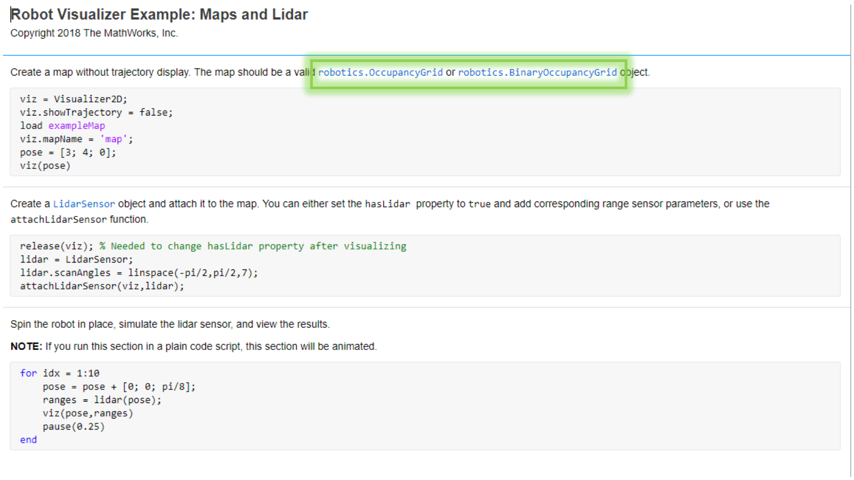
https://wiki.nps.edu/display/MRC/Assignment+5%3A+Mathworks+Mobile+Robotics+Simulation+Toolbox

---

1) Git Clone

mobile-robotics-simulation-toolbox

2) Map



Robot Visualizer Example: Maps and Lidar

Copyright 2018 The MathWorks, Inc.

Create a map without trajectory display. The map should be a valid `robotics.OccupancyGrid` or `robotics.BinaryOccupancyGrid` object.

```
viz = Visualizer2D;
viz.showTrajectory = false;
load exampleMap
viz.mapName = 'map';
pose = [3; 4; 0];
viz(pose)
```

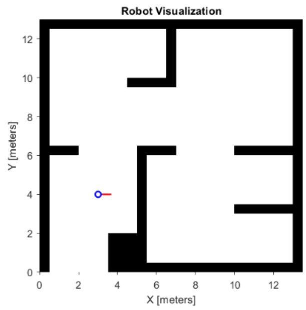
Create a `LidarSensor` object and attach it to the map. You can either set the `hasLidar` property to `true` and add corresponding range sensor parameters, or use the `attachLidarSensor` function.

```
release(viz); % Needed to change hasLidar property after visualizing
lidar = LidarSensor;
lidar.scanAngles = linspace(-pi/2,pi/2,7);
attachLidarSensor(viz,lidar);
```

Spin the robot in place, simulate the lidar sensor, and view the results.

NOTE: If you run this section in a plain code script, this section will be animated.

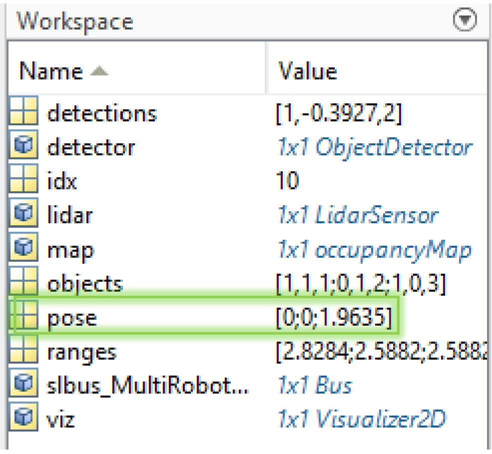
```
for idx = 1:10
    pose = pose + [0; 0; pi/8];
    ranges = lidar(pose);
    viz(pose,ranges)
    pause(0.25)
end
```



⇨ https://youtu.be/tZda3BoA-zY

3) Variables

On pourra lier l’IHM aux variables Matlab générées par les fonctions de mobile_robotics_simulation_toolbox pour mettre à jour les données provenant de la carte Raspberry **que l’on connectera via ROS.**



Workspace

Name	Value
detections	[1,-0.3927,2]
detector	*1x1 ObjectDetector*
idx	10
lidar	*1x1 LidarSensor*
map	*1x1 occupancyMap*
objects	[1,1,1;0,1,2;1,0,3]
pose	[0;0;1.9635]
ranges	[2.8284;2.5882;2.588
slbus_MultiRobot...	*1x1 Bus*
viz	*1x1 Visualizer2D*

A faire :

- ROS

NPS Wiki Espaces Rechercher Se connecter

Multi Robot Control

Pages

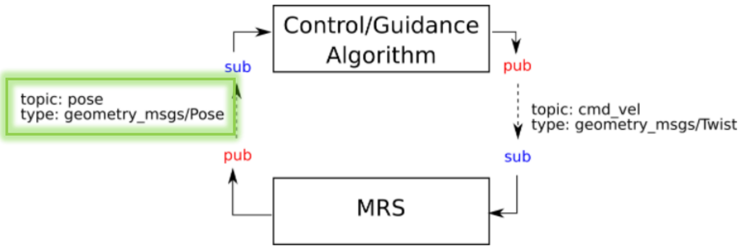
ARBORESCENCE DES PAGES

- ROS Laptop Computer Etiquette
- File lists
- Leading Discussion of a Research Pa
- MRC Assignments
  - MRC Assignments 2016
  - MRC Assignments 2017
  - MRC Assignments 2018
  - MRC Assignments 2019
  - Under Development, Coming Soo
  - Assignment 1: Introduction to Lin
  - Assignment 2: ROS Tutorials Part
  - Assignment 3: ROS and MATLAB/
  - Assignment 4: Robot Simulation
  - **Assignment 5: Mathworks Mobi**
  - Assignment 6: Basis Behaviors
  - Assignment 7: Basis Behaviors, Pa
  - Assignment 8: Composing Higher
- MRC Books
- Research Papers
- Syllabus
- Hokuyo Laser Rangefinder on Pione
- Microstain GPS/IMU Initialization Ex
- RFID Detection Post-Processing
- HistoricalPages
- Special Topics
- Ubuntu NPS Wireless
- Updating ME4823 Laptops
- Current Tasks
- MRC Schedule 20-3
- Article Discussion Schedule 20-3

Exercise 2: ROS interface for MRS

In Exercise 1 the interface between the simulation and the controller is all within Simulink. This allows us to test the algorithm, but not the actual interface we would use to implement the algorithm on hardware.

Create a new Simulink model where the interface between the controller and the simulation (MRS) is a ROS interface as illustrated below.



It should be possible to accomplish this as a single Simulink model where the model both publishes and subscribes to the two ROS topics (cmd_vel and pose).

- Save the final simulink model as `mrc_hw5/ex2/waypoint_ros.slx`
- And save an image file for the model as `mrc_hw5/ex2/waypoint_ros.png`. See Print Models to Image File Formats.

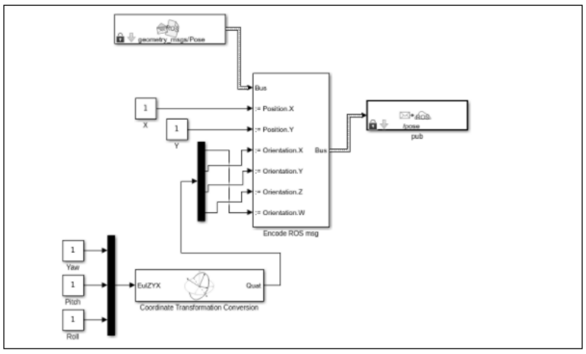
Hint: Publishing a `geometry_msgs/Pose` ROS message

Publishing a standard 3D Pose message (not the 2D `turtlesim/Pose` message) with Simulink requires a bit of setup.

First, take a look at the `geometry_msgs/Pose` documentation to understand the contents of the message. Publishing the message requires filing the appropriate fields. A couple things to consider art that you will...

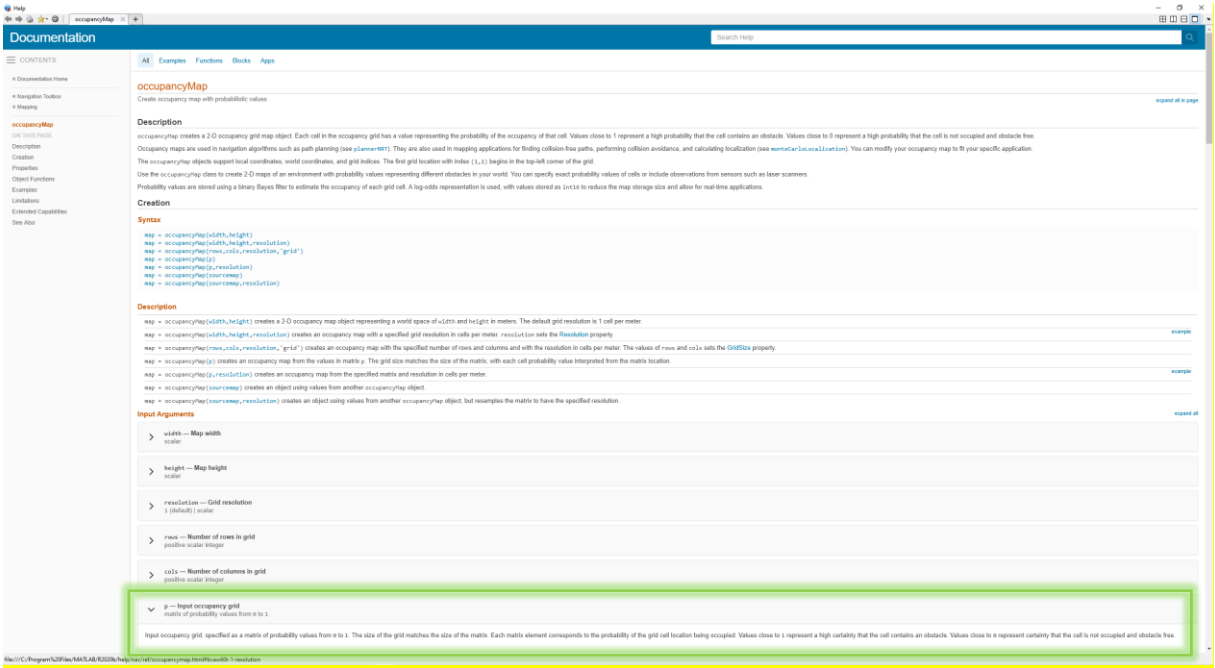
- need to covert Euler angles (roll, pitch, yaw) into a quaternion.
- populate the quanternion in the ROS message in the correct order, noting that the output of the coordinate conversion does not have the same convention as the input to the ROS message.

Here is a working example:



You should be able to download it here: https://gitlab.nps.edu/multirobotcontrol/mrc_examples/-/tree/master/simulink
The name of the model is pub_pose_example.slx.

- OccupancyMap – IHM pour pouvoir reproduire la map à la main depuis des mesures manuelles



- Faire bouger un robot sur la map avec l’IHM (avant de faire la connexion ROS)

→ **EN COURS DE DÉVELOPPEMENT**