

**LABWORK 1**  
**TURTLEBOT 2 BASICS** (Max score: 40)  
16-311: INTRODUCTION TO ROBOTICS (FALL 2017)

OUT: October 10, 2017, at 4:30pm

DUE: October 10, 2017 at 7:pm

## Instructions

## Homework Policy

You can discuss the exercises with your classmates, but you should write up your own solutions. If you find a solution in any source other than the material provided in the course, you must mention the source.

## Submission

Hand out your answers (handwritten or typewritten) at the end of the lecture or email a pdf to the instructor within one hour from the end of the lecture.

## Contents

<b>1 Sensors on your robot (10 points)</b>	<b>1</b>
<b>2 Move it (20 points)</b>	<b>1</b>
<b>3 See through (10 points)</b>	<b>2</b>

## 1 Sensors on your robot (10 points)

1. Launch a ROS package that brings up the minimal set of ROS nodes providing access to the core functionalities of your turtlebot.
2. Use ROS commands to gather information about active sensors. List the sensors, with a brief description of what they measure. Specify whether they are event-based or not.
3. By simple manual interaction with the robot, check that every sensor is doing its job (e.g., hit the robot to check bumpers). For each sensor, report the ROS topic that you have used to check it and the response of the sensor, quantifying the result of the experiment (e.g., I rotated the robot of approximately 45 degrees, and from topic `xyz` I could read this value in the field `fxzy` of data type `dtxyz`, or I could observe that heading was changing from `angle1` to `angle2`, ...).

## 2 Move it (20 points)

1. Perform the steps required to move your robot with both the keyboard and with the joystick (as specified in the document *Hands-on Turtlebot 2*). Move the robot across the desks in the classroom. Call an instructor to validate this step.
2. From the command line, send `Twist()` commands to the robot to perform the following:

- Stop-and-go motion: move straight by advancing a *bit* every second with a speed of 0.1 m/s. Once the robot reaches one meter distance from the starting point, stop it.. Visually tag the one-meter distance target using the tape, and check the distance by visual inspection of the estimates from the odometry topic. Report the deviation between the expected and the reached point.
- Let the robot rotate about an ICR distant 0.3m from the initial position of the robot with an angular velocity of 0.2 rad/s. After completing two full revolutions, measure the deviation. Compare the results with what is being reported by Odometry.

☛ In both, the two experiments above, before starting with the experiment, reset the odometry. This can be done by sending an `Empty` command to the topic `/commands/reset_odometry` subscribed by the kobuki base.

☛ Be aware that there are a few different, yet equivalent ways to publish messages by using the command line. ROS makes use of the YAML syntax (<http://wiki.ros.org/ROS/YAMLCommandLine>). The following example illustrates the different possibilities.

Let's consider messages of type `geometry_msgs/PointStamped`, that represent a point in the space and include also an header. The message data structure, as shown below, is composed of two parts:

```
Header header
  uint32 seq
  time stamp
  string frame_id
Point point
  float64 x
  float64 y
  float64 z
```

And these are all equivalent ways of publishing the same data on the `pt` topic using YAML syntax:

```
$rostopic pub pt geometry_msgs/PointStamped '{stamp: now, frame_id: base_link}' '[1.0, 2.0, 3.0]'
$rostopic pub pt geometry_msgs/PointStamped '[0, now, base_link]' '[1.0, 2.0, 3.0]'
$rostopic pub pt geometry_msgs/PointStamped '{header: {stamp: now, frame_id: base_link}, point: [1.0, 2.0, 3.0]}'
$rostopic pub pt geometry_msgs/PointStamped '{header: {stamp: now, frame_id: base_link}, point: {x: 1.0, y: 2.0, z: 3.0}}'
```

### 3 See through (10 points)

How to perform the following steps is illustrated in the document *Hands-on Turtlebot 2*.

- Complete the steps to get access to the depth camera, visualize the color image, and visualize the point cloud in `rviz`.
- Perform the steps for creating a ROS network with your robot and your laptop (the 'workstation').
- Check whether the camera image is being displayed with continuity or not on the workstation.
- If the step above is successful, connect the joystick and navigate the robot while enjoying watching the images on your laptop (well, it would help to have a wireless joystick ...).

Call an instructor to validate these steps.