Laboratory exercise 7

Mobile robot navigation in Stage using apriltag detections

Name: Sérgio Miguel Rosa Estêvão **JMBAG:** 0036563035

Preparation and helpful instructions

- **Do not: consult, read, or examine ANY materials or solutions for assignments from the previous years' editions of this course, or from your colleagues. The assignment solution must *fully* be your work. In case signs of plagiarism are detected, you will get a zero score and may be subject to be reported to the Faculty's Ethics Committee.**
- **Do: consult the teaching staff (for this assignment, teaching assistant Vlaho-Josip Štironja) via Teams DM, if you have any problems, if anything is unclear, or if you need any help with any part of the assignment.**
- Review the lecture slides about mobile robot localization and navigation.
- Review the lecture slides about writing a ROS subscriber and publisher in C++.
- Write clean, readable, easy-to-understand code. Give meaningful names to variables. Make sure that you check that your solution runs without syntax errors and works as intended. (Remember that you can always ask for help!)

Assignments

Task 1: Uncompressing images from rosbag

In the `Task 1` folder, you'll find a file named `lab7.bag`, a rosbag containing two topics:

- `/usb_cam/image_raw/compressed`
- `/usb_cam/camera_info`

with message types `sensor_msgs/CompressedImage` and `sensor_msgs/CameraInfo`, respectively.

When dealing with images, specialized transport strategies like compression or streaming codecs are often necessary due to memory constraints. However, visualizing or utilizing these images in code may sometimes require uncompressing. Uncompressing (in ROS) is usually done when reading the compressed images within the subscriber but in order to simplify the process, we'll transform `lab7.bag` into `lab7_uncompressed.bag` using the provided Python script `lab7.py` (all required libraries inside the python file should already be installed by default if you have ROS Noetic).

Execute the command `python3 lab7.py` within the folder to perform this transformation. Once the transformation is complete, try to visualize the images in Rviz or any other visualization tool (e.g. image_view package) using the `lab7_uncompressed.bag` file.

Task 2: AprilTag Detection with ROS

AprilTag, developed by researchers at the University of Michigan, is a visual tag system designed to offer precise localization with minimal overhead. This system provides accurate position estimates by associating specific locations with identifiable tags.

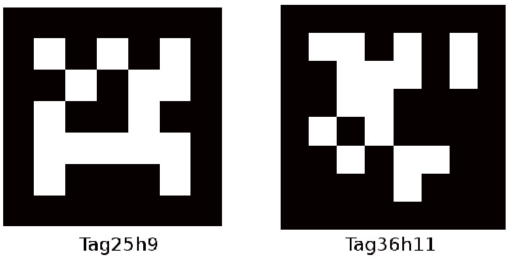

Figure 1: Examples of different AprilTag types.

In this task, we will detect the AprilTag from the `lab7_uncompressed.bag` file used in the previous task. The detection process will be facilitated using the `apriltag_ros` package.

1. Begin by installing the Apriltag library with the following command:

   ```
   sudo apt install ros-noetic-apriltag
   ```

2. Navigate to your source (src) folder within your workspace directory (e.g., `/catkin_ws/src`) and copy the `apriltag_ros` package from the Task 2 laboratory materials. This package is a personalized version with minor differences tailored for the RPS laboratory exercise.

3. After copying, ensure you are in the workspace directory and build the package using the command:

   ```
   catkin_make
   ```

   Verify that the compilation is successful; if not, contact the teaching assistant for assistance.

4. Source the `setup.bash` file from the `devel` folder:

   ```
   source devel/setup.bash
   ```

5. If everything is configured correctly, execute the following command:

   ```
   roslaunch apriltag_ros lab_7.launch
   ```

Ensure to play the rosbag and visualize the `/tag_detections_image` topic in image_view package. See more details on how to run the package within the documentation. If you do not have it installed by default install it using:

```
sudo apt-get install ros-noetic-image-view
```

Capture a screenshot when the `apriltag_ros` package successfully detects an AprilTag.

Task 3: Navigate mobile robot in Stage using apriltag detections

Stage

Following the steps presented in the lecture ”Mobile Robot Localization and Navigation”, run the Stage simulator with lab7.world environment. Run the robot keyboard teleoperation. Try moving the robot around the environment via the keyboard to make sure everything is working.

`apriltag_navigation` package

After making sure that everything is performed correctly in the final task you will have to make a package `apriltag_navigation` with which we will control mobile robots’ speed in the stage simulator using apriltag detections from `apriltag_ros`. Copy the package template `apriltag_navigation` provided in the material in `Task 3` folder and paste it into the workspace directory.

Navigate to the workspace directory and build the package using the command:

```
catkin_make
```

Ensure that the compilation is successful; if not, contact the teaching assistant for assistance.

Source the `setup.bash` file from the `devel` folder:

```
source devel/setup.bash
```

Implementing `apriltag_navigation` package logic

Implement logic same to the video (`lab7.mp4`) within the `apriltag_navigation` node. If `apriltag_ros` detects any apriltag in the rosbag file, the mobile robot should stop; otherwise, it should move at a constant speed during the rosbag duration.

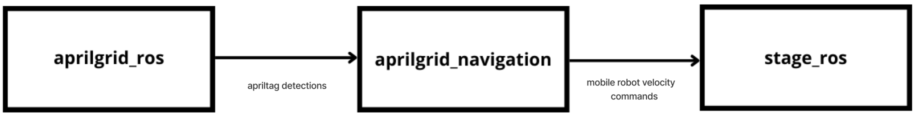

Figure 2: A scheme illustrating how the apriltag_navigation package is working.

Logic Summary:

- If any April tag is detected, the mobile robot **stops**.
- If no April tags are detected, the mobile robot moves **at constant speed during rosbag duration**.

Additionally, the whole setup has to be run from the same launch file (same as in the video (`lab7.mp4`)), and it should launch: `lab_7.launch` (from `apriltag_ros` package), `stage_ros`, `image_view`, your implemented `apriltag_navigation` node, and rosbag with uncompressed images (`lab7_uncompressed.bag`). Record a short video (max. 10 s) demonstrating how your package works (you can use Ubuntu screen recording (link or your mobile phone)).

Additional hints

- Use `rostopic info` to gather additional details about wanted topics.
- Use `/tag_detections` topic to check if an April tag is detected.
- Inspect the `msg` folder in the `apriltag_ros` package to understand how to manipulate custom `/tag_detections` messages.
- Refer to the this link on how to include another launch file in a launch file.
- In order to access `AprilTagDetectionArray` msg in `apriltag_nav.cpp` file use `apriltag_ros::AprilTagDetectionArray` form.
- In the video (`lab7.mp4`), the rosbag start is delayed, but you don't need to do the following in the lab.
- Please remember that you can modify any file within the `apriltag_navigation` package if it is needed.

Exercise submission

Create a zip archive containing exercise files. Organize the files in a zip archive into the following structure:

- Folder **Task_2** containing a screenshot from Task 2.
- Folder **Task_3** containing your whole `apriltag_navigation` package (including launch file) and short recorded video demonstrating the functionality of your system.
- This pdf with the filled-out details.

Upload on Moodle.