Exercise Session 5

Theory

- Service calls

Exercise

Use the node you implemented in Exercises 2 & 3 and add a service server that can start/stop the robot. This functionality could be used as an emergency stop.

1. Implement a service server (Lecture 4, Slide 8) that you can start and stop the robot. Use the `std_srvs/SetBool` service type for this task. Load the service name from the parameter server.
2. Run the simulation and call the service you have implemented from the terminal using `rosservice call` to start and stop the robot (Lecture 4, Slide 6-9). You need to implement the start/stop request handling logic such that you can call the service multiple times without restarting the simulation.

OPTIONAL

A. Create a *separate node* that stops the robot if it is *too close to an obstacle* using the laser measurements. Use the service you have implemented above.
B. Create a *separate node* that stops the robot *after a crash has occurred* with the stop service you have implemented above. For this, plot and analyze the data of the IMU under the topic `/imu0` with `rqt_multiplot` and develop a method to detect a crash.

Evaluation

- ❏ Stop SMB by using the service call.
  - ❏ Is the functionality implemented correctly? [20%]
  - ❏ Is the service called correctly? [20%]
  - ❏ Does the robot actually stop? [10%]
- ❏ Start SMB using the service call.
  - ❏ Is the functionality implemented correctly? [20%]
  - ❏ Is the service called correctly? [20%]
  - ❏ Does the robot actually start? [10%]

OPTIONAL

- ❏ Automatically triggered emergency stop when too close to an obstacle [Bonus 25%]
- ❏ Automatically triggered emergency stop after crashing with an obstacle [Bonus 25%]

Note that all the functionally must be implemented inside the class and not in the main() function where the class is instantiated. You will lose 20% points if you don't adhere to this rule. Optional tasks can be implemented directly in the executable file.