Exercise Session 4

Theory

- ROS bag
- ROS time
- Rpt_multiplot
- ROS launch
- RViz
- Service

Exercise

The goal of this exercise is to work with data that was recorded on a real robot. The recorded bag file contains sensor measurements from wheel odometry, inertial measurement unit (IMU) and laser scanner

1. Launch your controller from the last exercise. Use rqt_mulitplot to plot the path of the simulated robot in the x/y-plane (Tip: use the topic */odometry/filtered*).
2. Using *rosbag* and *teleop_twist_board*, in which *teleop_twist_board* use to navigate robot from A to B in 15 seconds, *rosbag* use to record data received from sensor such as wheel odometry, inertial measurement unit (IMU) and laser scanner.
3. Use *rqt_mulitplot* to plot the path of the recorded robot in the x/y-plane.
   - Tip: Remember to set the parameter */use_sim_time* to *true*: http://wiki.ros.org/Clock
   - Tip: Play the bag-file with: *rosbag play mydata.bag –clock* which also publishes the time of the recorded data: http://wiki.ros.org/rosbag/Commandline.
4. Visualize the motion of smb by using TF markers in RViz. Add a *robot_state_publisher* node to your launch file and load the smb robot description to the parameter server. Now you can visualize the smb model in RViz. (Tip: Use the *control.launch* file from the *smb_control* package as reference). Note: The wheels jump behind the robot body a bit, but that is okay.

5. Implement a service server 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.
6. Run the simulation and call the service you have implemented from the terminal using *rosservice* call to start and stop the robot. You need to implement the start/stop request handling logic such that you can call the service multiple times without restarting the simulation.