

# Exercise Session 5

## Theory

- Service calls

## Exercise

Use the node you implemented in Exercise 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 can start and stop the robot. Use the [std\\_srvs/SetBool](#) service type for this task.
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 4/7).

## 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 analyse the data of the IMU under the topic `/imu/data` with `rqt_multiplot` and develop a method to detect a crash.

## Evaluation

- |  |       |
|--|-------|
| <input type="checkbox"/> Stop Husky using the service call.  | [50%] |
| <input type="checkbox"/> Start Husky using the service call. | [50%] |

## OPTIONAL

- |   |             |
|---|-------------|
| <input type="checkbox"/> Automatically triggered emergency stop when too close to an obstacle   | [Bonus 25%] |
| <input type="checkbox"/> Automatically triggered emergency stop after crashing with an obstacle | [Bonus 25%] |