EVALUATION TASK FOR MOTION ROBOTICS SOFTWARE DEVELOPER

General notes and requirements for the task:

- The code produced in the context of this task should be stored in code repository and shared via accessible link to corresponding GIT repository. The repository should include contents of your ROS workspace.
- While developing, take advantage of the VCS, use it as you would use it in real work situation.
- Choose the ROS distro according to your best knowledge.
- We expect that the software is developed keeping in mind the best practices and software development conventions. Write the code as if you already worked for Cleveron.
- Provide README file that describes how to set up and run your solutions according to task numbers.
- We reccomend using different launch file for running each provided soluton.
- You can use all resources from the internet – different ROS packages, code examples, libraries, etc. But you must provide the information about used resources (in README file) so we could tell which part of the task is custom solution and which is taken from the internet.
- We expect comprehensive explanations of the solutions provided in the README file. You should explain how did you approach the problem, what alternatives are there to solve the problem. Even if you could not come to final practical and working solution, you should provide some discussion and options, algorithms or methods that could be used for resolving named problem. Every visual representation of your ideas are also very welcome.
- Using ROS tools to visualize your solutions is welcome.

Software development tasks using ROS

In this section you should develop ROS packages and nodes that will resolve the problems described under the tasks. The ROS project should be structured according to ROS standard. You can use any programming language, as long as you use it in context of ROS. **The final output should be provided as some published ROS message or service.**

test-task-1

We provide a bag file containing recorded sequence of positions (positions.bag) that can be treated as 2D coordinates on 2D plane. The coordinates are published as geometry_msgs/Point. The sequence represents a path of driving non-holonomic vehicle, points are received in frequency of 5Hz. All the units are according to REP-3 conventions: https://www.ros.org/reps/rep-0103.html.

1. Create a ROS node that will listen to vehicle positions from the bag file and publishes estimated linear and angular speeds of the vehicle.
2. (bonus) 5 Hz is rather slow publishing rate for position information. Create a node that can estimate robots position between two points and publish position information with the rate of 30Hz.

test-task-2

We provide a bag file (imu.bag) containing recorded sequence of IMU sensor measurements (sensor_msgs/Imu). Message includes angular velocity and linear acceleration.

1. The IMU data is very noisy. Use some method to your best knowledge, to filter the data. Publish the filtered data to a new topic.
2. Integrate the filtered IMU data and publish current linear velocity approximates.

bonus-test-task

1. The IMU data and positions data from previous tasks are in correspondance. Altough, IMU sensor may not be positioned on vehicle according to REP103 conventions, some rotation conversion may be needed.
2. You can fuse the IMU and positions data to get better speed and position estimates. Publish them on some ROS topic.