Localization_Lab_1

Task 1

Create a **ros2 python node** that subscribes to the imu message published by gazebo in the Simulation_demo package.

This node should do the following:

1- **Print a message when the heading is between [-2, 2] degrees.**

The printed sentence format should be as the following:

"The robot is nearly heading north .. Heading is: <a> degrees" where a is the current value.

2- **Print a warning if the absolute linear acceleration x or absolute angular speed z exceeded the following limits** :

- Absolute Linear limit : 0.3 m/s^2
- Absolute Angular limit : 0.3 rad/sec

The printed sentence format should be as the following:

"Warning !! .. linear acceleration x exceeded the limit . Current acceleration is <a> m/s^2" where a is the current value.

"Warning !! .. angular velocity z exceeded the limit . Current Angular velocity is <a> rad/sec" where a is the current value.

Use attached functions.py when needed.

Task 2:

Create a ros2 python node that uses the imu data extracted from the **attached** CSV file **imu_data.csv** to publish an imu message with a topic name **"zed2_imu"**
and a frame named **"zed2_imu_link"** with frequency **30 Hz**.

- Do all required unit conversions to publish the right values according to imu Ros message standard.
- Set a suitable values for covariance given that :
    - This imu always outputs a very accurate **angular velocity in x and y** and an accurate linear **acceleration** in all directions.
    - The accuracy of both output **orientation**(**yaw**) and **angular velocity z** depends on the Angular velocity z value. They are accurate If the absolute Angular velocity z is smaller than 0.3 rad/s and become less accurate when the absolute Angular velocity z exceeds this value.
- Assume any missing information.
- Any reasonable covariance value is Ok. They just should make sense relative to each other.

The **imu_data.csv** file has 7 columns: **accX, accY, accZ, angX, angY, angZ, yaw_deg**
Note that:
accX, accY, accZ are in “g”
angX, angY, angZ are in rad/sec
Yaw_deg is in degrees

Use attached functions.py when needed.