

Problem Statement

With the use of the ROS framework the basic functionalities of mapping, localization and navigation shall be achieved for an Autonomous Mobile Robots (AMR).

In the real world scenario, there are a number of issues which are not completely handled in the widely available open source modules. The following scenario is an example for the same.

Whenever an AMR is powered on within a known environment (with map), the first step is localization. That is informing the AMR its current location with reference to the existing map. Based on how good the initial localization estimation is, the further navigational performance varies. Hence this is a very important step for any AMR.

But doing a manual localization every time could be a tedious task. So why don't we think of a better alternative? Which could auto localize itself from the last known position. So here comes the problem statement.

The candidate is expected to develop a **C++** based software module that shall be having the following functionalities

- Updating the last known position of the AMR into a local text file.
- Whenever the navigation modules are re-launched, the localization should happen from the last known position. That's the last known coordinates should be read from the local text file and use that for initialization of AMR.

Ideally the module shall be having the following publishers/subscriber

Subscribers

- `/robot_pose*` - Robot pose with reference to the `/map` frame

*This shall be available by adding the node `robot_pose_publisher` from the package `robot_pose_publisher` to the navigation launch files

https://github.com/GT-RAIL/robot_pose_publisher

Publishers

- `/initialpose**` : Initializing the position of AMR with reference to `/map` frame

Important: This topic should be published only once .Otherwise continuously initialization of AMR will be happening,which is not appropriate.

Test Case

Since you are testing in a simulation environment,whenever the Gazebo modules are launched the AMR will always be spawned at a fixed location.

So for testing the functionality ,follow the steps mentioned below.

Spawn the robot using Gazebo modules
eg) `roslaunch turtlebot3_gazebo turtlebot3_world.launch`

Launch the localization-navigation modules like AMCL,move_base
eg) `roslaunch turtlebot3_navigation turtlebot3_navigation.launch`

Now after making the robot move to some random position (teleoperation/autonomous navigation) with localization intact, close the navigation launch file alone keeping the gazebo world launch file running .

Now on relaunching the navigation launch file (*with your developed package in it*),the AMR is expected to do the auto -localization

Expected output

C++ based software module,using **classes** that shall be able to do the auto localization of the AMR ,with the launch of navigation modules

Note:

- Candidate can make use of any existing projects for AMR functionality (preferably **turtlebot3**)
 - It would be much appreciated if you could make use of the OOP concept with the usage of **classes**
 - It would be much appreciated,if you could avoid the hard coding of location of local file (with coordinate values of AMR) in your cpp
- Hint: <https://wiki.ros.org/Packages#C.2B-.2B->