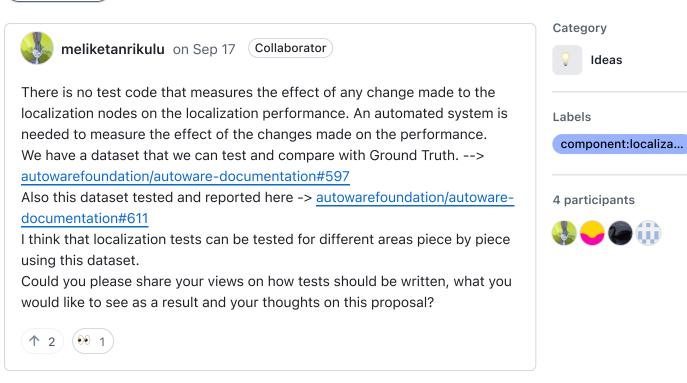


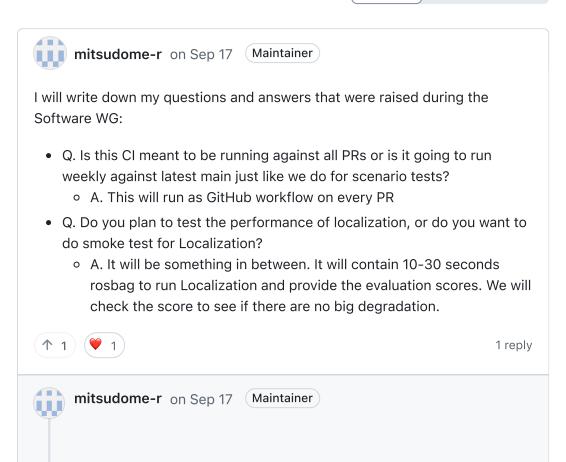
Adding Component Tests to CI/CD Pipeline for Localization #5250

Unanswered meliketanrikulu asked this question in Ideas



3 comments · 3 replies

Oldest Newest Top



@meliketanrikulu

It would be nice if you can write down how you are going to implement the test in the workflow as well before doing actual implementation so that I have a better idea of what you are planning to do? (e.g., are you planning to add new GitHub Action? What kind of configuration file do you plan to place and in which GitHub: autoware, autoware.universe, autoware.launch, etc.)





SakodaShintaro on Sep 18 (Collaborator)

I will briefly explain the activities at TIER IV.

(1) For changes that do not affect behavior or performance (i.e., refactoring), we conduct a deterministic test to ensure that the output (the value of /localization/kinematic_state) matches down to the precision of floating-point values. Recently, TIER IV implemented such a test privately for NDT+EKF configurations. This is related to the following pull request.

autowarefoundation/autoware.universe#8766

Since it is sufficient to execute the test with a single rosbag and map, we have set up a GitHub workflow in a private repository to run this test for each pull request that changes localization modules (it takes about 20 minutes per test). While this is currently kept confidential, there is a possibility that it could be made public.

(2) For changes that do affect behavior or performance, we perform a comprehensive evaluation using multiple rosbags and maps. TIER IV currently has around 30 datasets for evaluation. This evaluation is based on autoware's diriving_log_replayer, meaning we use ros2 bag play and ros2 bag record. By analysing the recorded rosbag, we verify the mean error and check the health of various /diagnostics topics. Although the evaluation contains some randomness, the large amount of data allows us to mitigate its effects. Since this evaluation includes data from TIER IV's customers and is also related to one of our products, Web.auto, it might be difficult to make it public.



2 replies



meliketanrikulu on Sep 18 (Collaborator)

For changes that do affect behavior or performance, we perform a comprehensive evaluation using multiple rosbags and maps. TIER IV currently has around 30 datasets for evaluation. This evaluation is based on autoware's diriving_log_replayer, meaning we use ros2 bag play and ros2 bag record. By analysing the recorded rosbag, we verify the mean error and check the health of various /diagnostics topics.

Although the evaluation contains some randomness, the large amount of data allows us to mitigate its effects. Since this evaluation includes data from TIER IV's customers and is also related to one of our products, Web.auto, it might be difficult to make it public.

Hello <u>@SakodaShintaro</u> . Thanks for your comments and for your sharings.

I think the test you mentioned is related to this document --> https://tier4.github.io/driving_log_replayer/quick_start/localization/. Is it correct? I performed the test in this document. I understand that it is tested based on the NDT score. Does it include any additional tests besides this? Our plan is to detect localization errors by using post-processed, high-accuracy GNSS/INS data as ground truth and comparing them with NDT+EKF autoware localization outputs . Is there such a comparison in your tests or what other metrics do you recommend using for comparison?



SakodaShintaro on Sep 18 (Collaborator)

The criteria of driving_log_replayer are listed here:

https://github.com/tier4/driving_log_replayer/blob/027e0252c73f0bb7

428cd181f8ee9b0571bc37db/driving_log_replayer/driving_log_replay
er/localization.py

It checks:

- nvtl
- The lateral deviation in NDT optimization.

However, I believe this is not sufficient.

driving_log_replayer also outputs a result_bag to
~/driving_log_replayer_output .

The rosbag includes the following topics:

https://github.com/tier4/driving_log_replayer/blob/027e0252c73f0bb7 428cd181f8ee9b0571bc37db/driving_log_replayer/launch/localization.l aunch.py#L19-L29.

To compare with the reference pose, I extract ekf or ndt poses from the rosbag using this script:

https://github.com/SakodaShintaro/misc/blob/98d473af5b941f3f8b12480e28f734ce1d4665f0/python_lib/extract_pose_from_rosbag.py.

Then, I use another script to compare the trajectories: https://github.com/SakodaShintaro/misc/blob/98d473af5b941f3f8b124 80e28f734ce1d4665f0/python_lib/compare_trajectories.py.

These are personal scripts I wrote, so I can't guarantee their quality.

Currently, we use the <code>/localization/kinematic_state</code> output from Autoware, which was recorded at the time the original input rosbag was collected, as the reference_pose. Of course, it's not a perfect choice, but we have verified that the pose is reliable to some extent by checking it visually in Rviz.



