[Proposal] Runtime Localization Module Switching for Multi-Pose Estimators #3878

KYabuuchi started this conversation in **Design**

KYabuuchi on Sep 28, 2023 Collaborator edited

On behalf of TIER IV, I plan to introduce a mechanism to launch multiple pose_estimators.

Background

Autoware currently supports NDT (LiDAR-based), YabLoc (camera-based), Eagleye (GNSS/INS & vehicle velocity -based) and AR-tag based localization as pose_estimators.

Currently, it is possible to launch multiple pose_estimators and fuse them using a Kalman filter by editing launch files. However, this approach is not preferable due to computational costs. Particularly, NDT and YabLoc are computationally intensive, and it's not recommended to run them simultaneously. Also, even if both can be activated at the same time, the Kalman Filter may be affected by one of them giving bad output.

This proposal suggests a mechanism to launch multiple pose_estimators, and allow for pausing and resuming specific ones according to the situation. Also, this feature will not have any impact on using a single pose_estimator as before.

Method

My proposed approach is to add a new package named `pose_estimator_manager`.
This package comprises switching-rules and sub-managers corresponding to each pose_estimator.

- Sub-managers controls the pose_estimator activity by relaying inputs or outputs, or by requesting a service.
- Switching-rules determine which pose_estimator to use by subscribing to the map or self-location.

Which sub managers and switching rules are instantiated depends on the runtime arguments at startup.
Following figure shows the node configuration when NDT, YabLoc and Eagleye are run independently, and when all of them are used simultaneously.

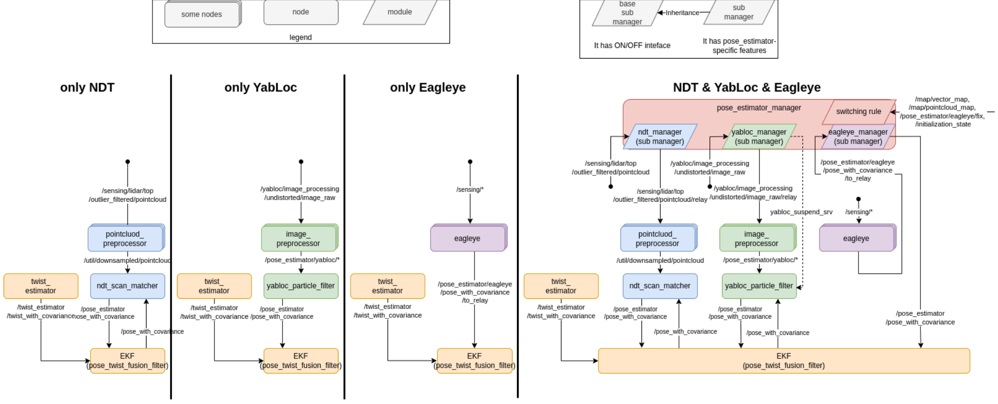
Category

Design

Labels

component:localiza...

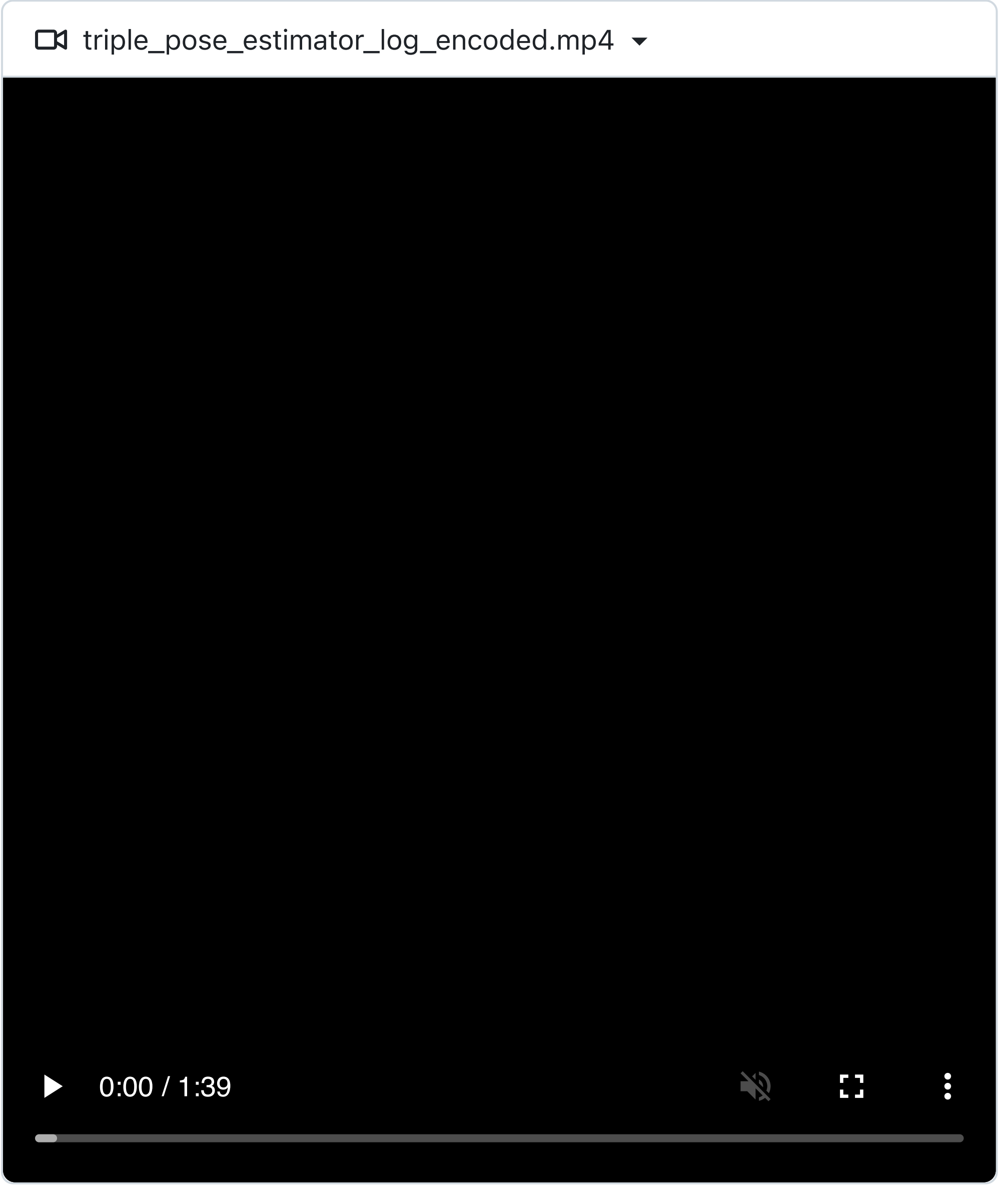
4 participants


some nodes
node
module
legend
base sub manager
Inheritance
sub manager
It has ON/OFF inteface
It has pose_estimator-specific features
only NDT
only YabLoc
only Eagleye
NDT & YabLoc & Eagleye
/sensing/lidar/top /outlier_filtered/pointcloud
pointcluod_ preprocessor
/util/downsampled/pointcloud
twist_ estimator
ndt_scan_matcher
/twist_estimator /twist_with_covariance
/pose_estimator /pose_with_covariance
/pose_with_covariance
EKF (pose_twist_fusion_filter)
/yabloc/image_processing /undistorted/image_raw
image_ preprocessor
/pose_estimator/yabloc/*
yabloc_particle_filter
/sensing/*
eagleye
/pose_estimator/eagleye /pose_with_covariance /to_relay
pose_estimator_manager
switching rule
/map/vector_map, /map/pointcloud_map, /pose_estimator/eagleye/fix, /initialization_state
ndt_manager (sub manager)
yabloc_manager (sub manager)
eagleye_manager (sub manager)
/sensing/lidar/top /outlier_filtered/pointcloud/relay
/yabloc/image_processing /undistorted/image_raw/relay
yabloc_suspend_srv

Demonstration

`logging_simulator.launch.xml` in dataset created by AWSIM.

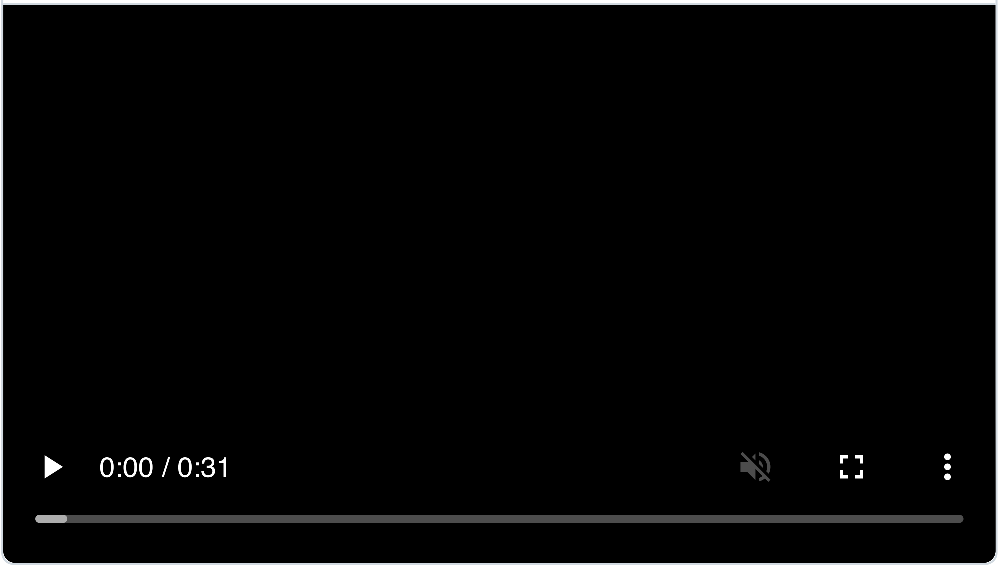
NDT, YabLoc and Eagleye switching between each other and performing localization.



`e2e_simulator.launch.xml` with AWSIM.

NDT and YabLoc switching between each other and performing autonomous driving.

double_pose_estimator_e2e_encode.mp4 ▾



Details

I will post proposal details and other ideas considered in this discussion later. Please wait.

↑ 4 🎉 6

6 comments · 5 replies

Oldest Newest Top

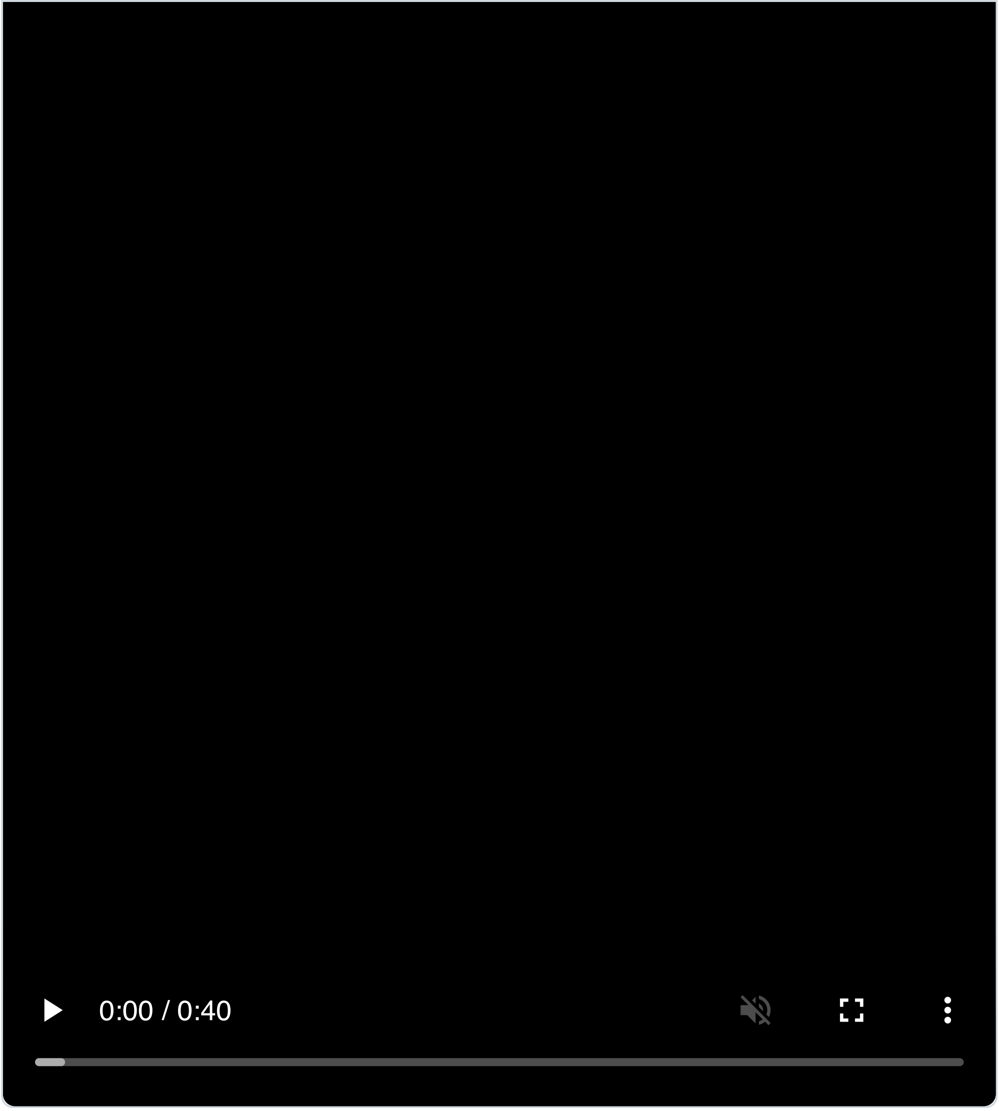
KYabuuchi on Nov 13, 2023 Collaborator Author

I give an update here.

Supporting landmark-based localizer

I have implemented support for the recently added landmark_based_localizer.
In this video, four different pose_estimators, NDT, YabLoc, Eagleye, and ArTag-based, are switched while estimating self-position.

quatro_swithing_for_discussion.mp4 ▾

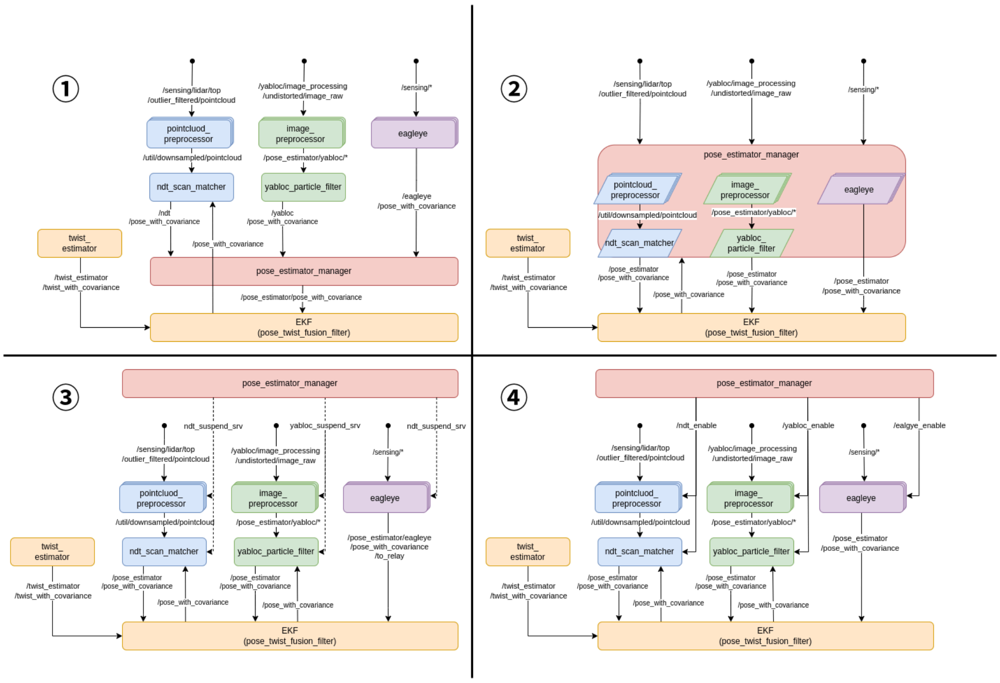


Details

The fundamental concept of the proposal is explained in the above post. The reason why I came up with the idea is that NDT and YabLoc have multiple nodes with high computational load, and we do not want to execute the highly loaded pose_estimators at the same time.

Other ideas

I devised other ideas as follows:



1. Launch all pose_estimators simultaneously and select which estimation results to use at a later stage.
   This approach depends on the computational performance; however, even the simultaneous launch of YabLoc and NDT might not be within acceptable computational limits. 😞

2. Modularize each pose_estimator and manage them internally within the pose_estimator_manager.
   Modularizing YabLoc and Eagleye, which are composed of multiple nodes, is super challenge. 😞

3. Instruct start and stop operations through a service.
   It is necessary to implement a server in all nodes that comprise the pose_estimator.

4. Communicate start and stop instructions by publishing topics.
   It is necessary to implement a subscriber in all nodes that comprise the pose_estimator.

Note that 3 and 4 in the above figure are undesirable because they include functions that are not needed when executed alone. However, my proposal does not prohibit them, and there is no issue if such an approach is chosen for arbitrating the future pose_estimator.
This proposal only proposes to implement a manager node with some sub_manager to stop each pose_estimator.
Since the structure of arbitrating by relay of topic does not require additional implementations, I adopted the method for NDT and YabLoc.

Future prospect

Honestly, I am not confident that the this approach is perfect.
There could be issues that we cannot see at this point, and I expect that the design may be changed in the future.
However, I intend to create a PR for this method because it at least covers the pose_estimators currently implemented and is an easy-to-revert implementation.

↑ 1 👍 1 0 replies

amadeuszsz on Dec 18, 2023 Collaborator

It is something what we are looking for. I just wonder how to maximize flexibility of this idea. I guess that the new localization modules will come in the future. Moreover, users may want to bind a custom localization to the system. Third version with forcing an API for each localization module seems reasonable but the `pose_estimator_manager` can be node independent, basing only on list of services/topics to operate. Regarding EKF, I would rather to change its API and align it with your contribution - input is parameterized and considers `twist_with_covN` and `pose_with_covN` topics. The only issue is with possible RVIZ plugin and switching future unknown modules, but I think it can be solved easily.

Please correct me if I'm wrong with my ideas. I have seen the recent changes in PR draft so I believe the topic is alive.

↑ 1 👍 1 1 reply

KYabuuchi on Dec 18, 2023 Collaborator Author

@amadeuszsz Thanks for your interest in runtime localization switching.
I have not had time to work on it, but this topic is in progress.

> The pose_estimator_manager can be node independent, basing only on a list of services/topics to operate.

While it would be ideal for the pose_estimator_manager to be independent of nodes, I think it might be challenging to achieve.

> Regarding EKF, I would rather change its API and align it with your contribution - input is parameterized and considers twist_with_covN and pose_with_covN topics.

Sorry. I do not understand the meaning of this change. At the moment, each pose_estimator outputs pose_estimator/pose_with_covariance, and only one callback in the EKF processes it.

👍 1

amadeuszsz on Dec 18, 2023 Collaborator

@KYabuuchi

> Sorry. I do not understand the meaning of this change. At the moment, each pose_estimator outputs pose_estimator/pose_with_covariance, and only one callback in the EKF processes it.

I thought about changing the current EKF package to handle multiple poses and simply configure the number of inputs by config file (see the example in 3rd party ROS pkg).

I'm aware that this approach will require extra effort, but let me know if it seems reasonable from your point of view.

↑ 1 👀 1 1 reply

KYabuuchi on Dec 18, 2023 Collaborator Author

At least the currently implemented localization modules output PoseWithCovarianceStamped and can be handled by the existing EKF callbacks, so I do not think we need it right away.
I believe it would be worth reconsidering this feature when new localization modules are introduced in the future.

👍 1

isouf on Jan 16

@KYabuuchi thank you very much for your proposal and your PR #5846. I have a question that I would appreciate if you can provide any direction.

I would like to fuse `ndt` with `eagleye` pose_estimators using the `ekf_localizer`.

According to your earlier post:

> Currently, it is possible to launch multiple pose_estimators and fuse them using a Kalman filter by editing launch files.

I had a look at the `ekf_localizer.launch.xml` file and I could only see two input topics:

1. `input_pose_with_cov_name`
2. `input_twist_with_cov_name`

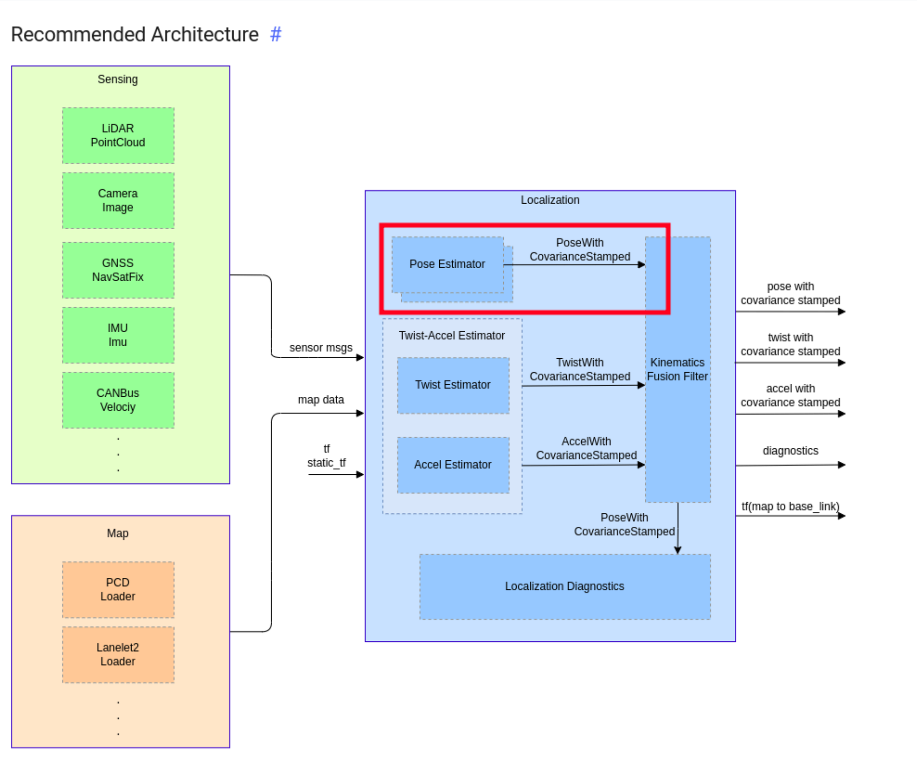
How can I add a second `input_pose_with_cov_name` topic (one for `ndt` and one for `eagleye`)? Will I need to modify the source code or I can do it by just editing the launch files?

↑ 1 2 replies

KYabuuchi on Jan 17 Collaborator Author

@isouf The `ekf_localizer` does not provide a mechanism to subscribe to multiple topics.
If you want to use multiple pose_estimators simultaneously, you can simply launch each pose_estimator.
Both NDT and Eagleye publish `/localization/pose_estimator/pose_with_covariance`, and the ekf_localizer subscribes to it, performing observation updates appropriately.

reference : https://autowarefoundation.github.io/autoware-documentation/main/design/autoware-architecture/localization/#recommended-architecture



Recommended Architecture #

👍 1

isouf on Jan 17

@KYabuuchi thank you very much for your help, it is clear to me now. I will update the `pose_twist_estimator.launch.xml` based on your PR changes to allow me to launch multiple pose_estimators.

👍 1

meliketanrikulu on Jan 31 Collaborator

Hello @KYabuuchi . We've talked about this before in this discussion(https://github.com/orgs/autowarefoundation/discussions/4134). As you say, these two proposals overlap, so it would be beneficial to move forward together. I have some questions.

1. Here you check what condition to switch. Did I understand correctly that the switch here involves using or not using more than one pose source?
2. How do you determine covariance values when you use all pose resources together? Are you using fixed covariance values for NDT and Yabloc? In this case, if NDT and yabloc do not work properly, it will continue to affect the system in the same way. In my previous tests, I could not obtain suitable results in the tests I conducted without making any direct corrections to covariance values. Because it is not defined as outlier when the GNSS error increases (gnss poses mahalonobis distance smaller than pose_gate_dist threshold every time even I set this threshold as 5.0 . EKF considers it as outlier when the mahalonobis distance exceeds this value and discards that value.) it distorts the

result. At the same time, in environments where NDT does not work properly, NDT will continue to disrupt the system in the same way.

3. In your approach, what do you think about adding the switching method according to the gnss covariance values that I explained in my proposal to the pose estimator manager?
   Thank you very much for your work and comments. I appreciate your clarification on these points.

↑ 1 1 reply

KYabuuchi on Feb 8 Collaborator Author edited ▾

@meliketanrikulu I sincerely apologize for the delayed response. I failed to notice the mention in this thread. 🙇

> Here you check what condition to switch.

Currently, there are no implemented switching conditions. (pose_estimator_manager just activates all) I've just implemented an architecture that allows adding various conditions in the future. I have prepared some sample rules, but they are not yet practical. (Originally, I had implemented a map-based condition, but the PR was too large, so I reduced it.)

> the switch here involves using or not using more than one pose source?

This is correct. The current pose_estimator_manager (actual package name is pose_estimator_arbiter) manages only the ON/OFF status of each pose_estimator.

> 2

I understand your concerns. Unfortunately, at the current stage, pose_estimators like NDT or YabLoc publish constant covariance. Therefore, when both NDT and YabLoc are launched, inappropriate results may occur if one of them outputs incorrect poses.

However, I think adjusting covariance in the post-processing stage of the pose_estimator is not a correct strategy. Ideally, covariance estimation should be done by the pose_estimator itself, not in the post-processing stage. If the estimation result is inappropriate, the pose_estimator should not output it. NDT, a relatively mature pose_estimator, is partially realizing that functionality.
As each pose_estimator improves, I trust your concerns will be naturally addressed.

In case my policy had problems, the implementation of my proposal has less impact on other packages. We are easy to revert.

> 3

Perhaps, I believe the answers provided on the original discussion page should be sufficient for this question. If there are still unclear points, please let me know.

KYabuuchi on Feb 8 Collaborator Author

It took some time, but my PR #6144 was merged a few days ago. There are several differences from what was originally proposed in this discussion, so I'll summarize the changes:

- Package Name: The proposed name was "pose_estimator_manager," but it became "pose_estimator_arbiter."
- Supported Estimators: The initial proposal did not consider ar_tag_based_localizer, but it is supported in the PR.
- Switching Rules: The original proposal suggested switching based on referencing the map. However, in the PR, I did not introduce this switching rule. Currently, pose_estimator_arbiter keeps all pose_estimators turned on.

Although the switching rule using a lanelet2 map is efficient, including the information for pose_estimator_arbiter in the official map specifications has a wide range of effects. For this reason, I excluded this rule this time.

↑ 1

0 replies