

Localization Architecture Design #2455

mitsudome-r started this conversation in **General**



mitsudome-r on Dec 22, 2021

Maintainer

edited ▾

I would like to initiate the discussion for the new Localization module that we are going to implement for Core/Universe.

The overview of the AD stack is being discussed [here](#), and this discussion thread would focus on the details of the localization module.

Tier IV has prepared the initial proposal in the following slides:

[Core_Universe Localization Design Proposal English Version.pdf](#)

↑ 1

👍 3

Category



General

Labels

component:localiza...

8 participants



0 comments · 17 replies

Oldest

Newest

Top



mitsudome-r on Dec 22, 2021

Maintainer

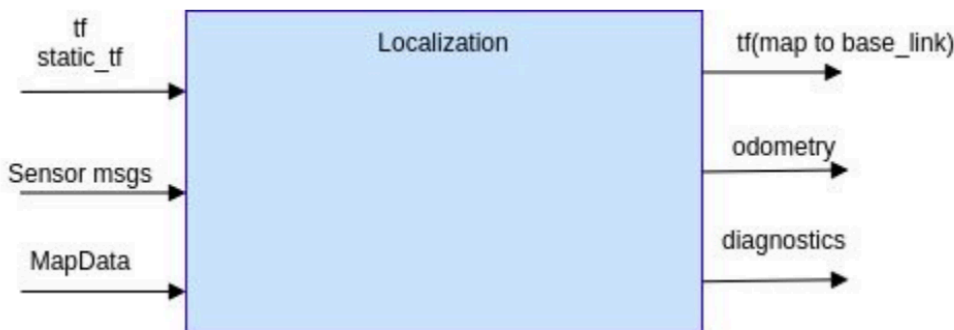
Author

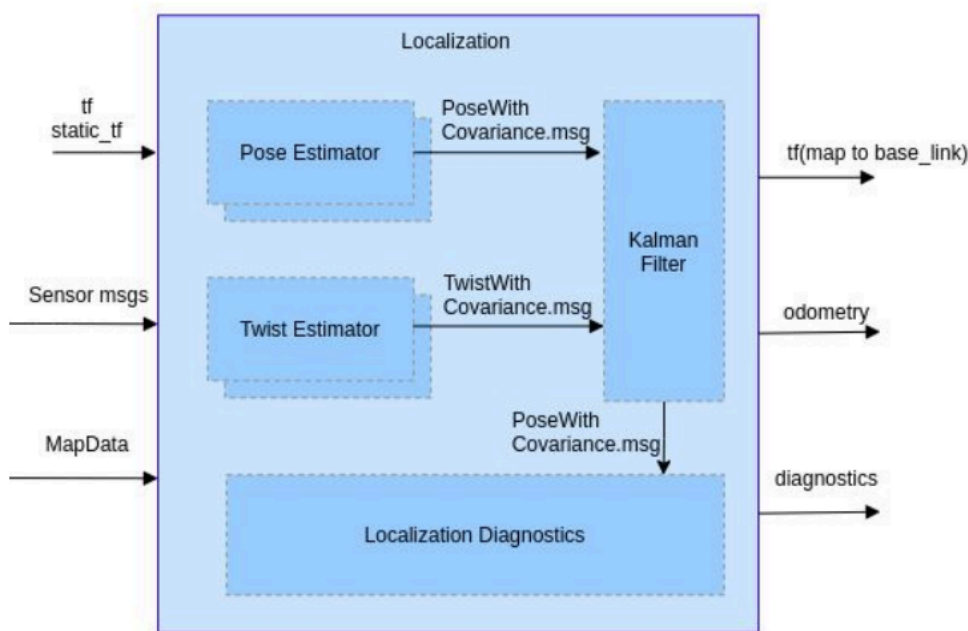
Here's the meeting minutes from the discussion we had today.

https://docs.google.com/document/d/1_4VbWuliRXWsc_u6vuDmNARjmT8QAOSmPvjPpZN7PK8/edit?usp=sharing

Summary:

In general, the proposal was accepted.





- We have agreed on overall input/output for the localization
- We also agreed to abstract the inputs to Kalman Filter and only allow PoseWithCovariance and TwistWithCovariance
- We will be starting without `odom` frame in Core architecture. If anyone can come up with a use case where we really need the frame, then we can have another discussion

Next Action:

- Tier IV will be creating a draft for localization design document based on today's discussion. Any review would be welcome
- We can further discuss about actual node level implementation that we would be expecting to have during Bus ODD offline using this thread.

The summary is just from the people who participated to the meeting, and if there are any people who weren't able to attend, feel free to comment your concerns or proposals in this thread.

↑ 2

2 replies



xczhanjun on Feb 6, 2022

edited ▼

How about add an end 2 end AI module together with the Kalman Filter to test data-driven path?



xmfcx on Mar 9, 2022 Maintainer

[@xczhanjun](#) As long as it can replicate the inputs and outputs of these specified modules, any module can be replaced partly or as a whole.

We are discussing what the inputs and outputs of these modules should be here.



maxime-clem on Feb 20, 2022 Collaborator

I would like to propose an additional feature: localization always produces an estimate of the current acceleration.

Motivation

In the current architecture, modules have to calculate the acceleration using the change of twist over time. Even if a sensor provides more accurate values they cannot be used without breaking the separation between hardware and software.

Possible Implementation

Localization uses acceleration information from sensors (e.g., IMU) if available. Otherwise localization estimates the acceleration using the change in twist over time.

↑ 3

3 replies



mitsudome-r on Feb 21, 2022 Maintainer Author

[@YamatoAndo](#) Do you have any opinion on this?



YamatoAndo on Feb 21, 2022 Collaborator

I agree with this propose.

Currently, the acceleration is calculated in each planning module as shown in the code below, but I also think the localization module should produce it.

https://github.com/autowarefoundation/autoware.universe/blob/tier4/proposal/planning/behavior_velocity_planner/include/behavior_velocity_planner/planner_data.hpp#L152-L182

https://github.com/autowarefoundation/autoware.universe/blob/tier4/proposal/planning/behavior_velocity_planner/src/node.cpp#L126-L127



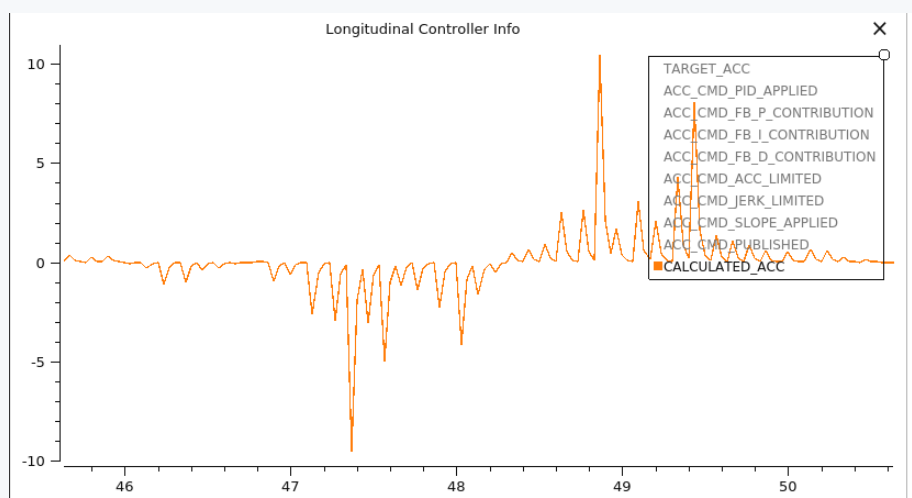
1



brkay54 on Feb 21, 2022 Collaborator

edited ▼

I agree with [@YamatoAndo](#) but also we should change the acceleration data on all modules. I plotted the estimated acceleration of trajectory follower node as you can see below.



Estimated acceleration is not reliable I think, 9-10 m/s² acceleration is not possible in AVP Demo. If localization module will produce the measured acceleration as an option, I think all modules should use that data.



YamatoAndo on Feb 24, 2022 Collaborator

In the current architecture, the localization module outputs the Odometry type.

If we add the acceleration to the output, which one do you think would be best for the output?

1. Pose, Twist and Accel are combined into one data type.
 - User-defined message type consisting of std_msgs/Header, geometry_msgs/PoseWithCovariance, geometry_msgs/TwistWithCovariance and geometry_msgs/AccelWithCovariance
2. Pose and Twist are output as Odometry type, and Accel is output separately.
 - nav_msgs/Odometry
 - geometry_msgs/AccelWithCovarianceStamped
3. Pose, Twist and Accel are output separately.
 - geometry_msgs/PoseWithCovarianceStamped
 - geometry_msgs/TwistWithCovarianceStamped
 - geometry_msgs/AccelWithCovarianceStamped

↑ 1

5 replies



maxime-clem on Feb 25, 2022 Collaborator

I like both options 1 and 2. I think if few nodes use the acceleration information then option 2 is probably better.



YamatoAndo on Feb 25, 2022 Collaborator

Option 3 seems to be popular in Tier IV.



maxime-clem on Feb 25, 2022 Collaborator

I do not have a strong opinion.

What I liked about option 2 was that only nodes that need acceleration would subscribe to the corresponding topic, making it clear which node uses the acceleration data.

Option 3 actually has the same advantage so this is also fine.



xmfcx on Mar 9, 2022 Maintainer

edited ▾

@[maxime-clem](#) @[YamatoAndo](#) I also support option 3. And mark all 3 with same timestamps. And use [exact_time_policy](#) for near latency-less communication of these messages.

They should all be together or they will feel disorganized and confuse people.



xmfcx on May 27, 2022 Maintainer

In [autowarefoundation/autoware_msgs#16](#) I've went with the option 1 to keep things simple.



meliketanrikulu on Jun 2, 2022 Collaborator

Hello, do you plan to include gnss data in ekf? I think GNSS is an important source of localization. Any progress on this? We talked about this issue. -->[autowarefoundation/autoware.universe#800 \(comment\)](#)



3 replies



YamatoAndo on Jun 2, 2022 Collaborator

edited ▾

@[kminoda](#) is working on supporting multi input pose in EKF, starting this week.

If his work is done, we can add gnss data to the estimation as well.



kminoda on Jun 2, 2022 Collaborator

Thanks for your interest! Yes, I am currently working on multiple sensor inputs into the ekf_localizer package, which is targeted for the configurations like follows:

- IMU+odometry, and NDT
- IMU+odometry, GNSS, and NDT
- IMU+odometry, GNSS doppler, GNSS, and NDT

In the proposal, ekf_localizer will subscribe to

`/localization/pose_estimator/pose_with_covariance` and

`/localization/twist_estimator/twist_with_covariance` . If you want

to use both NDT and GNSS for the pose information source, you need to put both information into

`/localization/pose_estimator/pose_with_covariance` .

Here is the PR (draft): [autowarefoundation/autoware.universe#1027](#)

(As the module is responsible for the final output of localization results, it may take some time for performance validation.)



meliketanrikulu on Jun 2, 2022 Collaborator

Thanks for your work. [@kminoda](#)

