Adding Orientation Source (IMU etc.) to ekf_localizer package #4973

meliketanrikulu started this conversation in **Design**

meliketanrikulu on Jul 11 Collaborator edited

Category

Design

Labels

component:localiza...

meeting:localizatio...

2 participants

Problem :

We observed that the performance of EKF decreased while the vehicle was moving in roll and pitch. Related issue is here
Currently, 1D Filter is used for roll, pitch and position z(height) estimation in the ekf_localizer package. The angular velocity data that comes with the twist message and the orientation data in the imu data are not used in the calculations. It is calculated only by reference to the NDT poses that come as input.

Solution :

IMU provides us with orientation information at a very high frequency. By using this data , we can localize roll, pitch and position z more accurately.
In this PR , it is seen that this problem has been solved by adding imu to ekf.

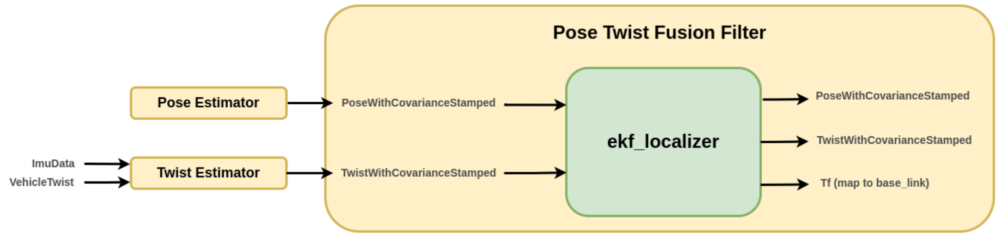
Proposal :

Current Autoware Localization Architecture:

Autoware Node Diagram is here

Recommended Architecture is here

Autoware's localization architecture is simply as follows:

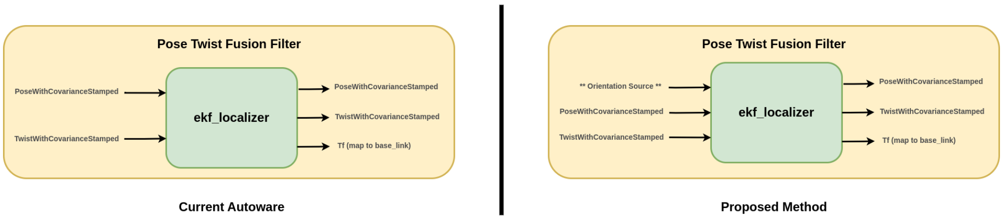


We see that the PR described above is not suitable for this architecture. Because the ekf_localizer package does not use Imu Data as input.

Proposed Changes:

Currently ekf_localizer uses 1D Filter by taking only NDT poses as input to estimate roll, pitch and position Z. If we have a source that provides accurate orientation and high frequency data, it is recommended to use this source directly. In the PR above, IMU was used as a reliable orientation source. However, different sources can also be used here. For example, the autoware orientation message can be used as a source.

The proposed method is as follows.


Pose Twist Fusion Filter
PoseWithCovarianceStamped
TwistWithCovarianceStamped
ekf_localizer
PoseWithCovarianceStamped
TwistWithCovarianceStamped
Tf (map to base_link)
Pose Twist Fusion Filter
** Orientation Source **
PoseWithCovarianceStamped
TwistWithCovarianceStamped
ekf_localizer
PoseWithCovarianceStamped
TwistWithCovarianceStamped
Tf (map to base_link)

Current Autoware

Proposed Method

Discussion Topics:

1. I see that adding an external orientation source to ekf_localizer contributes positively to the performance of ekf_localizer. PR. What are your suggestions for improving this?
2. In case of adding orientation source, which message type would you recommend to add ekf_localizer?
3. Apart from that, what are your suggestions to improve the roll pitch and z estimation performance of ekf_localizer?

↑ 1 👍 1

2 comments

Oldest Newest Top

SakodaShintaro on Jul 12 Collaborator

Thank you for creating the discussion. 😄

I agree that we should improve the roll, pitch and z of the EKF. However, rather than adding an orientation source, I feel that we should make appropriate use of the current architecture of the EKF as it is.

One of the shortcomings of current EKF is that it uses a very simple 1D-Filter for roll and pitch. I think this can be improved by using twist as an input and guaranteeing latency. This does not require any changes to the interface, and can be achieved by using the current twist input.

Currently, the only delay time taken into account is z.
https://github.com/autowarefoundation/autoware.universe/blob/19c1d0379942ba3a50907f381c38531ce6a494c3/localization/ekf_localizer/src/ekf_localizer.cpp#L191-L194
(And this is likely not very useful as the pitch is likely to be inaccurate)

How about doing something similar for roll and pitch?
That is, multiply the angular velocity by the delay_time and then apply it to the orientation of the pose before inputting it to the `update_simple_1d_filters`. With this policy, there are few codes that should be changed.

Or you could create a 2D-Filter that estimates z, roll and pitch using the values and their rate of change. More additional codes may be needed in this method, but a good estimation could be made.

Of course, implementing a 6DoF Kalman filter would be a good option.

I think we should be cautious about adding orientation because there is a lot of room for improvement even in the current architecture.

↑ 2 👀 1 0 replies

meliketanrikulu on Jul 22 Collaborator Author

Sorry for replying late and thanks for your suggestions @SakodaShintaro . I tried adding delay compensation to EKF. I saw that the result changed positively. I opened a PR for this. Here is the PR:
autowarefoundation/autoware.universe#8095
I would be very happy if you review it. Thanks

↑ 1 👍 1 0 replies