

# Proposal to integrate Eagleeye, a GNSS/IMU-based localizer, into Autoware #3257

rsasaki0109 started this conversation in Show and tell



rsasaki0109 on Feb 7, 2023

Collaborator

edited

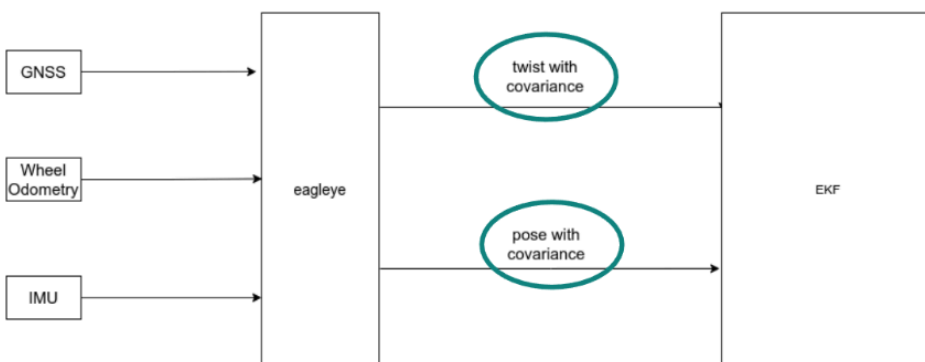
[@yukkysaito](#) [@mitsudome-r](#) [@xmfcx](#) ([@n-patiphon](#))

On behalf of [MAP IV. Inc.](#), I am pleased to announce a proposal to integrate [Eagleeye](#), our open-source GNSS/IMU-based localizer, into the localization stack of Autoware. Eagleeye uses low-cost GNSS and IMU sensors to provide vehicle positioning and orientation, offering a cost-effective alternative to LiDAR and point cloud-based localization.

Although the current version of Eagleeye requires wheel speed information, there are plans to remove this requirement in the future, making Eagleeye even more accessible to Autoware users. With this planned integration, Autoware users will have the option to choose between the existing LiDAR and point cloud-based localization stack or the GNSS/IMU-based Eagleeye localizer, depending on their specific needs and operating environment.

In our planned integration, there are two ways to utilize Eagleeye's result in the Autoware localization stack:

1. Feed both Twist and Pose from Eagleeye into the EKF localizer.



Category



Show and tell

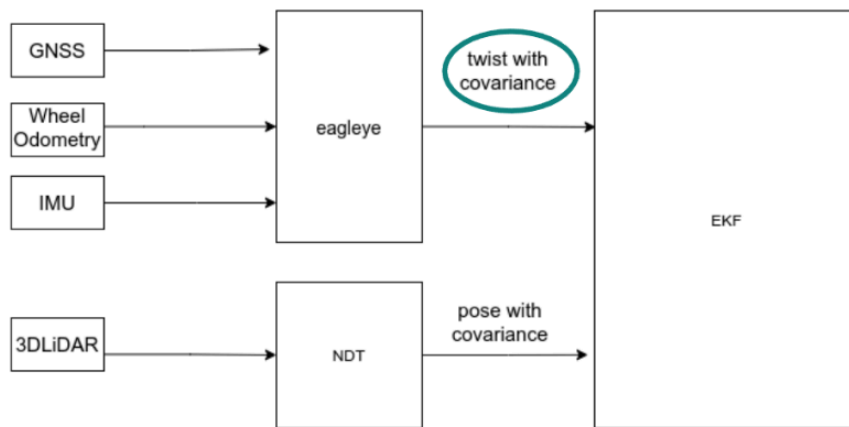
Labels

component:localiza...

8 participants



## 2. Feed only Twist into the EKF localizer.



By using both Twist and Pose from Eagleye, Autoware can localize without using LiDARs. Meanwhile, feeding only Twist into the EKF localizer is expected to improve the stability of NDT scan matching by providing a more accurate guess pose for scan matching.

We have already started working on the proposed integration in Q4 of 2022, and several tasks have already been completed to make Eagleye compatible with Autoware and to allow for testing the integration:

- [Modifying Eagleye to support TwistWithCovarianceStamped](#) as its speed input to match Autoware's specifications
- Adding a [fix](#) to handle [EKF not initialized without NDT](#)
- Adding a [new geoid model](#) to Eagleye
- We have also modified `autoware_launch` on a private fork of Autoware for testing. This modified `autoware_launch` allows users to easily enable the use of Eagleye outputs in Autoware.

As for the current progress, we are now in the testing phase. We have just finished the first testing on a real vehicle. We performed the test in an Urban area surrounded by high buildings. The vehicle was driven by a human driver (NOT in an autonomous mode). The goal of this test was to determine if the estimated positions were accurate enough in actual driving scenarios. The results were promising as the estimated positions remained within the lane throughout the test. The following is the video of the test.

[https://www.youtube.com/watch?v=\\_AJmJfGuEoo](https://www.youtube.com/watch?v=_AJmJfGuEoo)

We plan to finish implementation and testing by the end of Q1 of 2023. During this time, we plan to have another two tests. One with a human driver and the other in full autonomous driving mode. Moreover, we will transfer changes of `autoware_launch` to a public fork of Autoware in preparation for opening a PR. We have regular discussions regarding the details and progress of this integration in the [Mapping Work Group](#), but we will also post future updates in this Discussion as well.

Please note that this integration is not meant to replace the current localization stack but to provide Autoware users with another option. We believe that the integration of Eagleye will enhance the functionality and versatility of Autoware.

We invite the Autoware community to stay tuned for updates on the integration process and to provide feedback and suggestions. Your contributions are essential for the continued improvement and development of Autoware.

Thank you for your support, and we look forward to your feedback!



2



5

5 comments · 11 replies

Oldest

Newest

Top



**yukkysaito**

on Feb 7, 2023

Maintainer

Nice activity 👍 If you have any problems, I will do our best to support you, so please feel free to ask anything.



1



1

0 replies



**rsasaki0109**

on Feb 10, 2023

Collaborator

Author

I submitted pull requests.

[#3261](#)

[autowarefoundation/autoware\\_launch#200](#)

[autowarefoundation/autoware.universe#2848](#)



1



1

3 replies



**mitsudome-r**

on Feb 10, 2023

Maintainer

[@rsasaki0109](#) It would be nice if you can also create PR to Autoware Documentation under "How to Guide" to explain how people can use Eagleeye.



**rsasaki0109**

on Feb 10, 2023

Collaborator

Author

edited ▼

That's important, I would like to create a PR.

Work is in progress on the following branches

[https://github.com/MapIV/autoware-documentation/tree/add\\_eagleeye\\_in\\_how-to-guides](https://github.com/MapIV/autoware-documentation/tree/add_eagleeye_in_how-to-guides)

PR

[autowarefoundation/autoware-documentation#334](#)



**rsasaki0109**

on Apr 26, 2023

Collaborator

Author

PRs merged except document.



2



1



**n-patiphon** on Feb 16, 2023 Collaborator

edited ▾

Hi [@meliketanrikulu](#)! To follow up on your question last night, I checked with the developer ([@rsasaki0109](#)), and it seems like there won't be any changes to the EKF localizer in this integration. However, I've just heard that there are some modifications needed in `pose_initializer` to keep AD API working properly.



1



1

2 replies



**meliketanrikulu** on Feb 16, 2023 Collaborator

Hello [@n-patiphon](#) . Thanks for information. Yes, `pose_initializer` may need to be changed to trigger EKF.



1



**rsasaki0109** on Mar 9, 2023 Collaborator Author

I have addressed this issue.

[autowarefoundation/autoware.universe#2904](https://autowarefoundation/autoware.universe#2904)



1



**VRichardJP** on Apr 27, 2023 Collaborator

Hi!

Sorry to dig out an old discussion. Have you made any comparison between the following setups:

1. eagleye Twist & pose -> EKF
2. eagleye Twist + NDT pose -> EKF
3. gyro odom Twist + NDT pose -> EKF (default autoware)
4. GNSS pose + gyro odom Twist + NDT pose -> EKF (not implemented in autoware EKF, but it is not difficult to add)

In particular, I am curious how 2. and 4. differ, as localization would use the same data but with different algorithms.

Thanks!



1

1 reply



**rsasaki0109** on Apr 27, 2023 Collaborator Author

edited ▾

[@VRichardJP](#)

For setups 1-3, we have conducted detailed evaluations and plan to share the results eventually (perhaps in Mapping WG). Setup 4 is also currently under evaluation.

For setup 2, compared to setup 3, the performance of Twist is improved, which results in a better localization outcome. This is particularly helpful in environments with fewer distinctive features, where NDT can become unstable. The improved Twist helps reduce the number of iterations needed for NDT.

As for setup 4, while it is indeed possible, it is currently under investigation with an estimated completion timeline around September. For example, issues such as misaligned absolute coordinates in the point cloud map may cause the localization to not work properly. If this combination works well, I believe it will enable more robust localization.



**enescingoz** on Feb 23

**@rsasaki0109** is there any updated documentation to run eagleye with autoware?



5 replies



**rsasaki0109** on Feb 25

Collaborator

Author

Please check the link provided.

<https://github.com/autowarefoundation/autoware-documentation/blob/main/docs/how-to-guides/integrating-autoware/launch-autoware/localization/eagleye/index.md>



**noyansongur** on Mar 5

**@rsasaki0109** Thank you for your response. Just to follow up on **@enescingoz**'s question, we are trying to run Eagleeye with one of the provided datasets on [Autoware documentation](#), specifically rosbag replay data. However, we have not been able to localize using Eagleeye. Is there a dataset that you confirmed to be working with Autoware and that you can share with us?

Thank you very much in advance!



**rsasaki0109** on Mar 5

Collaborator

Author

This PR's rosbag has been verified to work with Autoware and Eagleeye. [autowarefoundation/autoware.universe#6144](#)



**noyansongur** on Mar 5

Thank you [@rsasaki0109](#) for your prompt response. I just downloaded the rosbag and it looks like there is no twist topics in there. All I can see are the following topics:

```
/clock | Type: roscpp_msgs/msg/Clock
/sensing/camera/traffic_light/camera_info | Type:
sensor_msgs/msg/CameraInfo
/sensing/camera/traffic_light/image_raw | Type:
sensor_msgs/msg/Image
/sensing/gnss/pose | Type: geometry_msgs/msg/PoseStamped
/sensing/gnss/pose_with_covariance | Type:
geometry_msgs/msg/PoseWithCovarianceStamped
/sensing/gnss/ublox/nav_sat_fix | Type: sensor_msgs/msg/NavSatFix
/sensing/gnss/ublox/navpvt | Type: ublox_msgs/msg/NavPVT
/sensing/imu/tamagawa/imu_raw | Type: sensor_msgs/msg/Imu
/sensing/lidar/top/pointcloud_raw_ex | Type:
sensor_msgs/msg/PointCloud2
/vehicle/status/velocity_status
```

I am curious what would be the following config parameters for this rosbag to run eagley as pose\_source and twist\_source?  
especially:

- twist type and twist topic
- sub-gnss type and topic
- use\_gnss\_mode

Thank you so much once again!



**rsasaki0109** on Mar 5

Collaborator

Author

edited ▼

The /vehicle/status/velocity\_status topic, related to wheel velocity, is converted into the twist topic by a node

`vehicle_velocity_converter` .

The /sensing/gnss/ublox/navpvt contains speed from GNSS and is set as gnss.twist\_source.

The gnss.pose\_source is /sensing/gnss/ublox/nav\_sat\_fix.

There's no need to configure the sub-gnss type and topic as it is not present in this rosbag.

The use\_gnss\_mode doesn't need to be changed from the standard settings.