[Proposal] Using GNSS/INS and NDT Together in Localization #4134

meliketanrikulu started this conversation in **Design**

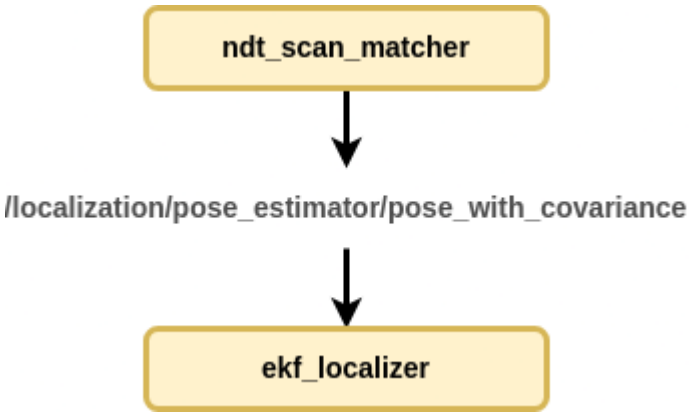
meliketanrikulu on Jan 30 Collaborator edited

In the localization system currently used in AW, we can use GNSS/INS and NDT separately. There are cases where NDT and GNSS work well and badly. For example, we know that NDT does not work well if there are bridges or open areas where there are not enough features. Likewise, GNSS does not perform well enough when passing near tall buildings or under trees.

Using these two pose sources together will ensure that localization works properly in these different situations. Our goal is to obtain the healthiest localization output by using GNSS and NDT together without restarting AW when moving from one situation to another while driving.

To do this, we need to feed both pose sources to EKF. EKF basically uses covariance values as reference to evaluate the poses it receives. There is no suitable real-time method to calculate covariance values of NDT. However, GNSS provides us with its own covariance accurately.

Default Autoware Architecture:

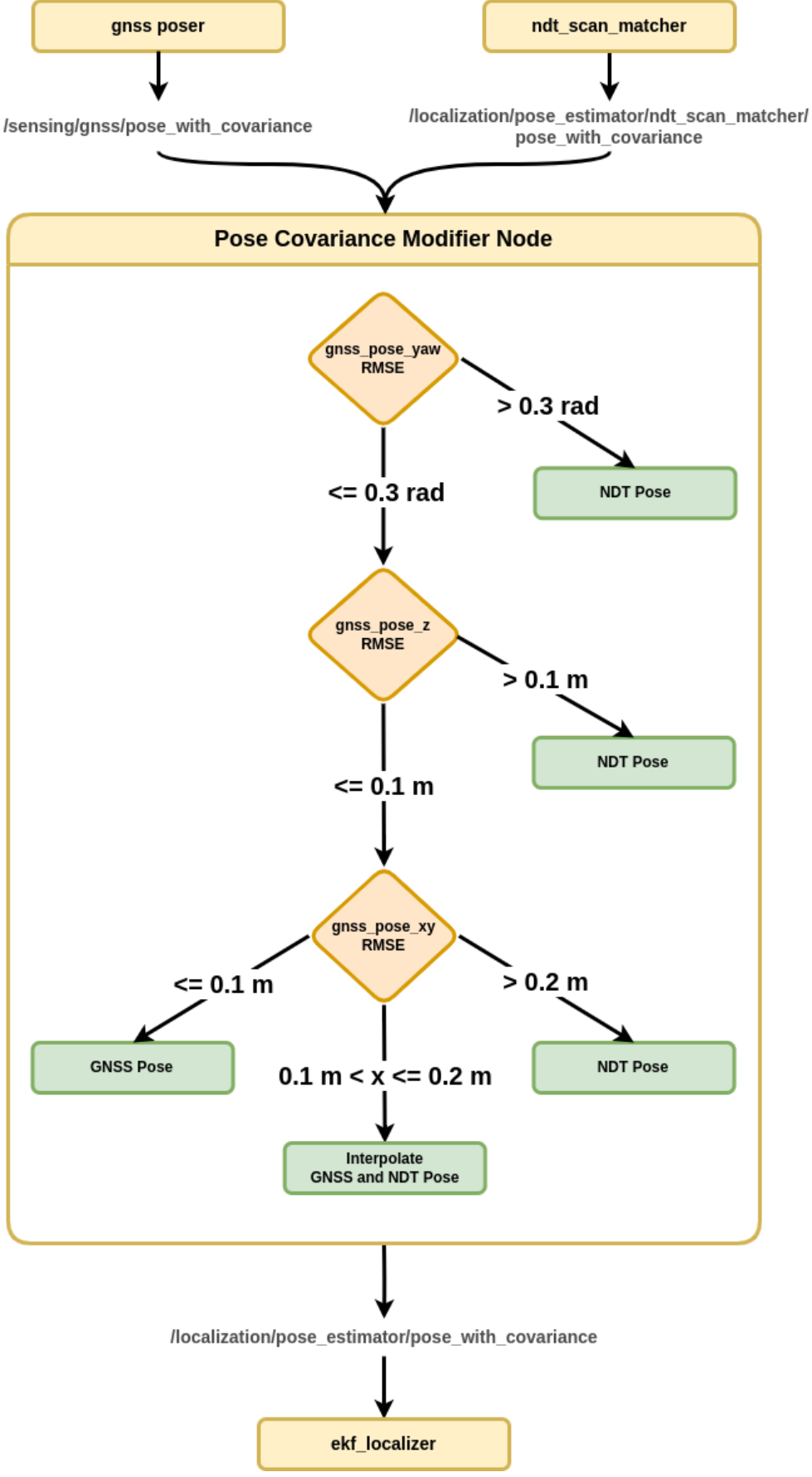


This method recommends assigning covariance to NDT by reference to GNSS covariance values.
Here is the proposal diagram:

Category
Design

Labels
component:localiza...

4 participants



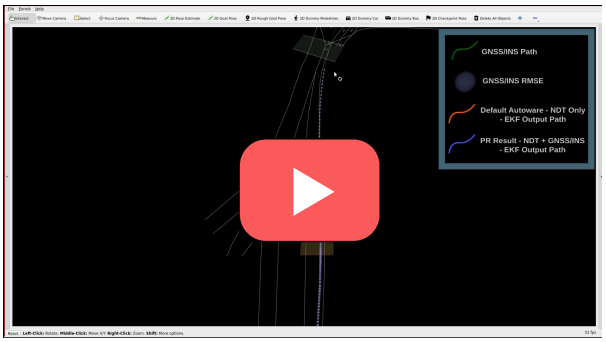
Case I --> GNSS RMSE values in most reliable boundaries (GNSS Position RMSE ~ <= 0.10 meters):
In cases where GNSS covariance values are low, driving directly with GNSS seems appropriate.
For this reason, while the covariance values of GNSS are within reliable limits, the covariance of NDT is set to a range that EKF will perceive as outlier.

Case II --> When GNSS RMSE values start to go beyond reliable limits (~0.10 meters <= GNSS Position RMSE <= ~0.25 meters):
When GNSS covariance values start to increase, NDT covariance values are gradually reduced. This process should be set dependent on GNSS covariance values.

Case III --> When GNSS RMSE values are completely outside the reliable limits (GNSS Position RMSE >= ~0.25 meters):
In this case, it is accepted that GNSS cannot produce reliable data and NDT covariance is reduced to its minimum value (0.15 meters). At the same time, at this point, the covariance values of GNSS were moved to a range that EKF would consider as outlier, thus ensuring that EKF did not evaluate these values. In other words, driving is carried out only by reference to NDT poses.

In this way, a method that allows driving in different environments together with NDT and GNSS is proposed. First version tests have been carried out.

Example Test Video:



It is seen that there are fast transitions at some points in these tests. Efforts are being made to soften these transitions.

The method can be improved with your suggestions. We would be happy to hear any suggestions or opinions you would like to add.

↑ 2 👀 1

5 comments · 10 replies

Oldest | Newest | Top

KYabuuchi on Jan 31 Collaborator

Hi! Thank you for the interesting proposal.

Currently, I am proposing a pose_estimator_arbiter for realizing multi-pose_estimator, and the PR is under review.
autowarefoundation/autoware.universe#6144 I am concerned about potential conflicts with your proposal. 😨 I believe it would be fine to implement your proposal by extending my pose_estimator_arbiter. This is because my pose_estimator_arbiter already supports pose_estimators other than NDT, and by adding additional rules, we can control the ON/OFF status of each pose_estimator. example video

So, to better understand your proposal, I have a few questions:

- What does RMSE for NDT POS and GNSS POS mean? Does it refer to the covariance of pose_with_covariance?
- Why manipulate the variance of NDT in each case? If the covariance of GNSS varies, naturally, EKF should favor the one with smaller variance.

I appreciate your clarification on these points.

↑ 1 0 replies

meliketanrikulu on Jan 31 Collaborator Author

Hello, Thanks for the update. Yes, we can collect them under one package with your work.I will examine it in more detail.
About the questions:

1. RMSE ' s for NDT POS and GNSS POS means square root of covariance.
2. Why I manipulate the NDT covariance values ? When I do not change the covariance of NDT, even if the GNSS error is low, it uses both pose sources and produces a new pose in the middle of the two (close to the pose which have low covariance value). Because it does not accept NDT poses as outliers. I cannot calculate the NDT covariance in the current system. So I don't actually know how accurately it works. But I know GNSS accuracy and we are trust this . For this reason, I want to use GNSS as the base pose source when the GNSS error is low. However, this situation changes when the GNSS error begins to increase. When the GNSS error starts to increase, I want to use both pose sources together, and when it increases completely, I want to use only NDT. At points where I want poses to be perceived as outliers, covariance can be turned on/off instead of manipulating, but in the case where I want to use NDT and GNSS together, I need to determine the covariance of NDT according to GNSS.
   I hope it was clear explanation. Please ask if anything is missing. Thank you

↑ 1 👍 1 4 replies

KYabuuchi on Jan 31 Collaborator

Thanks for the detailed explanation. I now understand well. 😄 I will try to figure out how to integrate your ideas with my multi-pose_estimator feature.

KYabuuchi on Feb 1 Collaborator

@meliketanrikulu In the ndt_scan_matcher package, there is a function called "regularization" that corrects the convergence position using GNSS. You can find more information about it here. Instead of introducing a covariance_switch_node, a similar effect can be achieved by expanding the existing regularization feature of the ndt_scan_matcher.

Currently, the regularization constrains the NDT convergence only in the longitudinal directions using GNSS. Expanding this to include 3DoF constraints when GNSS RMSE is low seems like another approach to achieving the goal.
What do you think about this idea? Thank you.

meliketanrikulu on Feb 1 Collaborator Author edited

Hello **@KYabuuchi** . Thanks for your response
As you said, Regularization in NDT only makes a lateral correction. It can be worked on and expanded, but as far as I know, the covariance values of regularization_pose are not used in the regularization process in ndt. So it will not work well in places where the gnss error increases. Because there is no control for this.

In addition, we want to make more extensive use of GNSS. When the GNSS error is low, we want to use GNSS instead of a source whose error we do not know, such as NDT.
However, when the GNSS error starts to increase, we want to use NDT together with GNSS for a while. When the GNSS error increases completely, we want to switch completely to NDT.

However, there is a point that I forgot to mention here: We want to drive with GNSS in places where the GNSS error is low, but the error of GNSS in z increases faster. When we drive with GNSS, we generally want to use the altitude of NDT, since the error in GNSS z increases faster. I had to make some changes to EKF to further evaluate the error in z. I will create a PR soon, it will be more understandable there.

When I examined your PR, I realized that you switched between NDT and Yabloc depending on whether the PC map was available or not. Other than that, are you using a switch condition? Did I understand this part correctly?
This proposal is a little different from your work; in fact, there is no switch mechanism like in your work. We aim to use GNSS and NDT together. Yabloc may also be here instead of NDT. However, the main issue here is to use the pose source whose error we do not know according to the GNSS error, but to use a source that knows its line as a base.

👀 1

KYabuuchi on Feb 2 Collaborator

> When I examined your PR, I realized that you switched between NDT and Yabloc depending on whether the PC map was available or not. Other than that, are you using a switch condition? Did I understand this part correctly?

In my PR, I provide only the switching mechanism, and the rules for switching are not yet implemented. The intention is to add various rules to this pose_estimator_arbiter in the future, allowing the switching of multiple pose_estimators.

Some switching rules are planned to be implemented, and they are expected to be switched based on configuration like plugins. Switching rules based on the availability of a point cloud map are just one example of such rules.

👍 1

KYabuuchi on Feb 2 Collaborator

The Localization team at TIER IV discussed this proposal. Honestly, from an architectural perspective, we prefer not to use GNSS and NDT in this way. 😢 We understand that your proposed method will improve performance in some environments and that it is easy to use. However, we should avoid putting situation-specific functionality into autoware.universe.

For example, there are the following concerns:

- Directly editing covariance to make the EKF ignore NDT pose is somewhat hack-like.
- Lowering NDT's covariance when GNSS accuracy is high is logically questionable.
- Ideally, NDT should estimate its own covariance. Moreover, there is already a feature for NDT to estimate covariance (though it is off by default, and there are concerns about processing overhead).

As an alternative, though it may take some time to implement, we suggest:

- Merge the pose_estimator_arbiter. (The switching rules are not implemented yet.)
- Implement a rule in the pose_estimator_arbiter that checks the covariance of GNSS and switches between pose_estimators based on it.

I will describe the detailed plan and its advantages and disadvantages later...

↑ 1 👀 1 3 replies

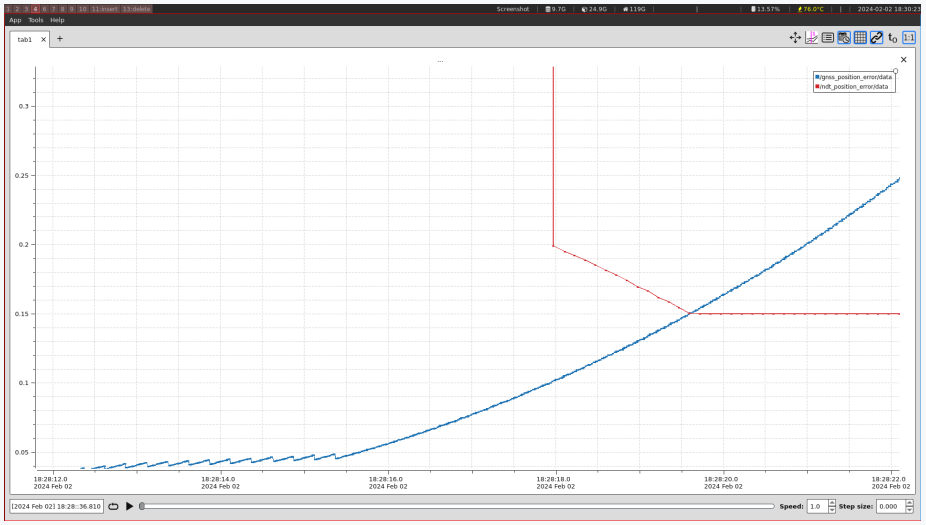
meliketanrikulu on Feb 2 Collaborator Author

Thanks for your quick response. I understand your concerns and agree that this is not the best solution. The correct solution is for NDT to calculate its own covariance and I follow your work on this subject. I know that your work on NDT covariance calculation is not yet ready to be used in real time, and this is actually a long-term research topic. It would be best to use this once completed. I also think that giving the covariance of NDT as a constant is also a wrong approach.

We are trying to create a feasible method in which we can use NDT and GNSS together in the most efficient way. We want to not only turn NDT on/off according to the GNSS error, but also use both together. The advantages of this method will be the following.

1. Instead of using NDT with fixed values, we will be able to make maximum use of GNSS in EKF, where we rely on covariance values.

2. When the GNSS error begins to increase, transitions from GNSS to NDT will occur more smoothly.
   An example transition between NDT and GNSS would be like this:



Also, what I want to explain is this:
The reason why I want to continue sending with high covariance when the GNSS error is low, instead of turning off NDT completely, is that even if the errors in x, y, yaw increase, I want to continue to benefit from NDT if the error in z does not increase or vice versa.
(Mahalonobis distance x, y at the current aw is calculated by taking the yaw components as reference, and if it exceeds the threshold, it does not use the values in z. I recommend making a small change here, too. But it is a subject open to discussion.)

In this way, EKF will outlier the components of NDT with high error values, while continuing to use other components.
After your PR is merged, we can add it to your package as an additional option. Those who want to use it this way can activate it and use it this way.

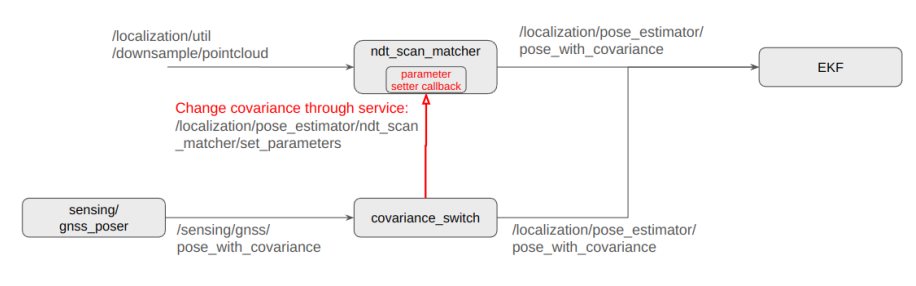
👀 1

KYabuuchi on Feb 6 Collaborator edited

Hi **@meliketanrikulu** Thank you for sharing your thoughts.
Sorry if my comments below seem critical. But I believe we need to consider this proposal very carefully. 🙏

The original plan I had in mind was for the `pose_estimator_arbiter` to toggle GNSS and NDT without changing covariance. Unfortunately, incorporating the covariance adjustments you desire into the current implementation is challenging.
At present, the pose_estimator_arbiter only manages the ON/OFF status of the estimator. I am planning to add functionality to gradually switch between multiple pose_estimators in the future, but the architectural idea is still incomplete. Additionally, we need to modify all the pose_estimator to align features such as covariance adjustments, etc., so it will take time to achieve. 😢

As another idea, How about implementing a service callback to adjust the covariance in NDT?
With this approach, there is no need to modify the interface (node configuration or topic names) between NDT and EKF. It will behave the same as before if we don't launch covariance_switch_node. Users can launch covariance_switch only when they want to use GNSS for position estimation assistance.
What do you think about this idea?



By the way, connecting the output of `/sensing/gnss_poser`, which doesn't belong to pose_estimator, to EKF requires careful consideration. (The current pose_estimator options are ndt, yabloc, eagleye, and artag. These can be launched as runtime options when starting Autoware, and their output topic names are maintained.)

If adopting such a change, when developing future functionalities, we would need to consider:

- Whether EKF should assume the input pose like from gnss_poser, that cannot perform position estimation on its own.
- Whether pose_initializer should activate/deactivate gnss_poser .
- Whether the system_monitor should monitor gnss_poser and invoke an emergency state.

Inconsistent node configurations make determining specifications and adding functionalities more troublesome.

👀 1

meliketanrikulu on Feb 7 Collaborator Author

Thank you **@KYabuuchi** . I think we can use NDT parameter setter callback. I will create draft PR we can talk on this.

😄 1

meliketanrikulu on Mar 6 Collaborator Author edited by xmfcx

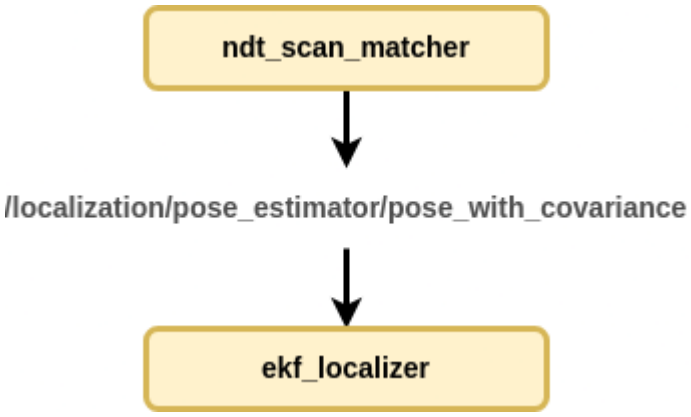
Hello **@KYabuuchi**
As we discussed, I created a working architecture by activating NDT by calling the /set_parameters service from outside.

This is how current PR currently works. (like here--> https://github.com/orgs/autowarefoundation/discussions/4134#discussioncomment-8378581)

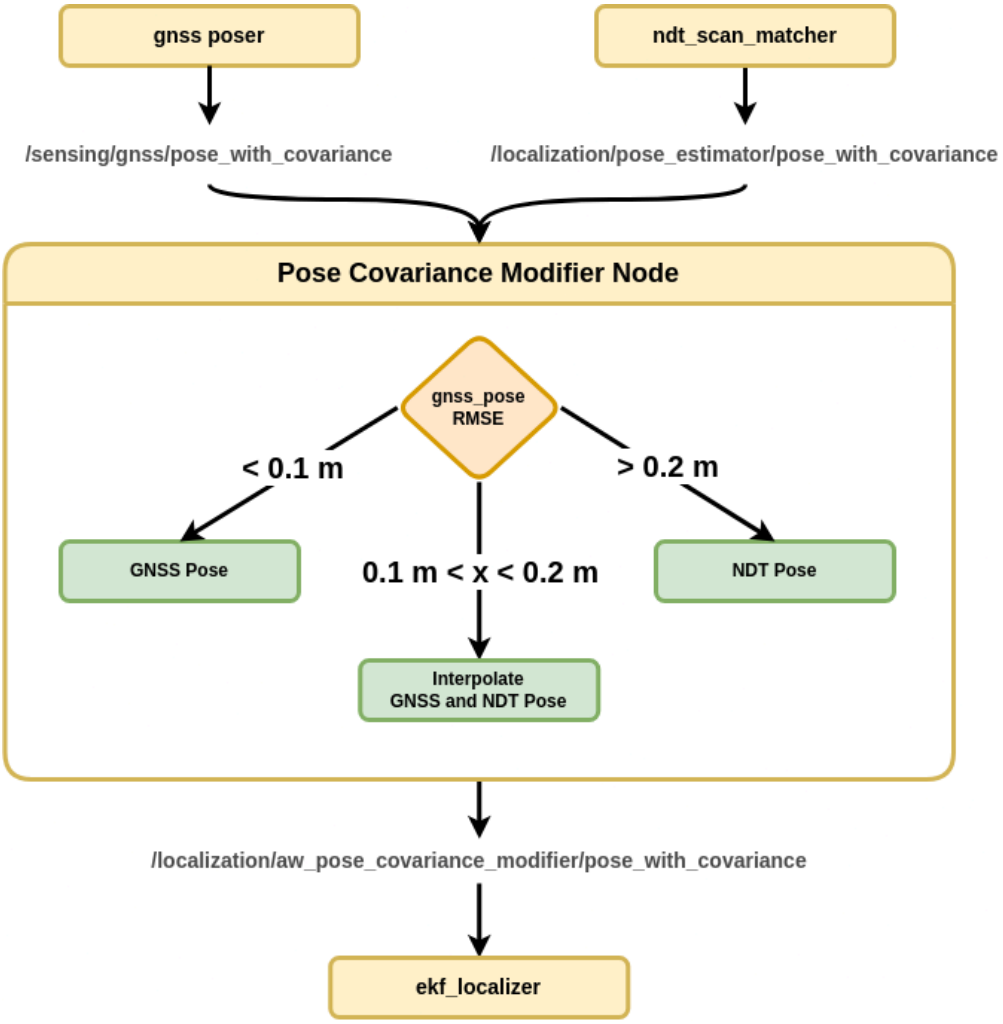
However, when we talked about this PR with our team again, we decided to change the structure a little more. I plan to move all the processes made without interfering with the NDT node to the extra package that I will add. I will update the PR accordingly.

In this way, we decided that all operations would be carried out by an external package without adding any extra features to the NDT in Default AW or complicating the code. Below I share diagrams to explain the new architecture we are planning.

Existing Autoware Architecture



Proposed Change



I will update PR according to this architecture as soon as possible. Please share your opinions with us

↑ 1 👀 1 3 replies

xmfcx on Mar 6 Maintainer edited

@meliketanrikulu the node name prefix has to be `autoware_`. `aw_` is not acceptable.

👍 1

TaikiYamada4 on Mar 7 Collaborator

Hi @meliketanrikulu !
Thank you for your valuable proposal!
May I ask some questions about the architecture of this proposal?

1. Is this node tend to be launched for any cases? As far as I know, there is a quite large range of GNSS sensors around the world. Some doesn't provide the covariance and some doesn't provide the orientation of the sensor. So I feel we can't make an assumption that the GNSS provides reliable data, and this feature has to be optional so that the users can toggle the on/off for this node.

2. How does this module deals with the asynchornized input of GNSS and LiDAR pointclouds? I assume that this module will mix the GNSS and NDT pose when 0.1 m < x < 0.2 m, and I'm curious how to balance them when both sensors have a differenet publication frequency.

3. How did you decide the threshold 0.1 m and 0.2 m? Is there some theoretical background to it?

I look forward to your thoughts and answers to the questions above. Thank you once again for your contribution and engagement.

meliketanrikulu on Mar 12 Collaborator Author

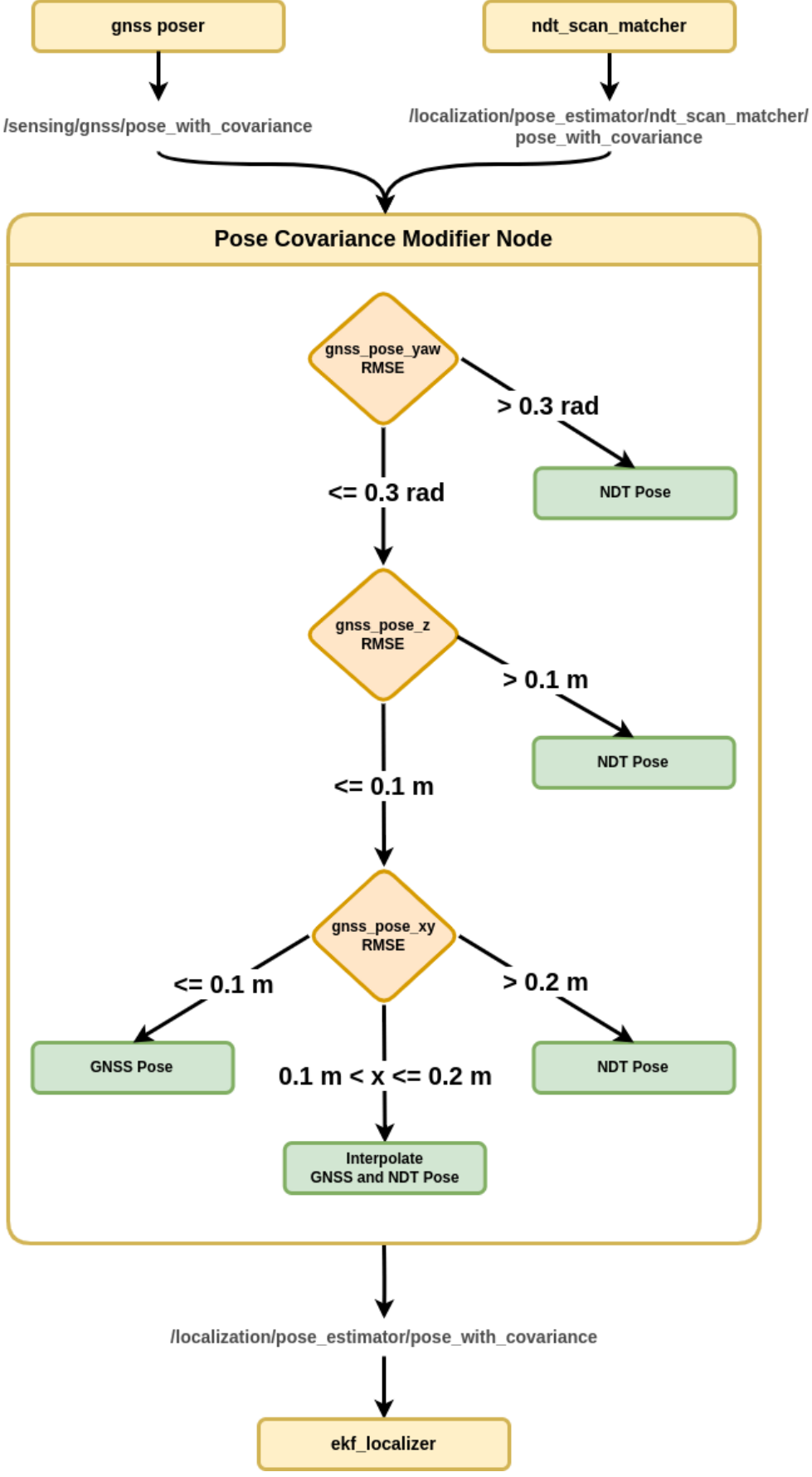
Hello **@TaikiYamada4** . Sorry for replying late. Thank you for your interest and comment.
First of all, this feature will remain false in default autoware and those who want to use it will be able to turn it on with a single parameter change.

- Apart from that, Yes, you are right, not all GNSS sensors provide RMSE information and you can only use this package if your GNSS sensor calculates the error value. That's why I added this to the "importance notes" section of the README.md file.
- We decided the threshold values by experiencing the acceptable error range of localization. When looking at the position covariance values in Default NDT, we also took into account that the position error value was accepted as 15 cm. However, I added these threshold values to the parameter file. In other words, those who wish can use these values by changing them for different applications and scenarios.
- Regarding your other question, the synchronization problem needs future work. This is one of the shortcomings of the package. If you have any suggestions on this subject, I'd be happy to hear them.

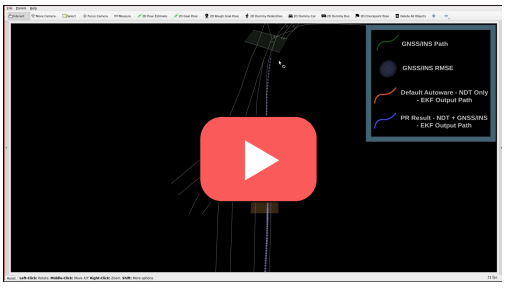
❤️ 1

meliketanrikulu on Apr 18 Collaborator Author edited

We decided that instead of changing the topic names entering the EKF, it would be more appropriate to change the name of the topic that is the pose output of NDT. I updated proposal architecture:



I also updated the test video:



↑ 1 0 replies