Telemetry and Basestation Solutions #2664

Unanswered **HaoruXue** asked this question in **Feature requests**

HaoruXue on Jun 13, 2022 Collaborator

Background

While working with Autoware on scaled robotic vehicles and IAC race cars, I realize how much a solid telemetry and basestation (BS) solution targeting developers is needed. They are used to

- View and toggle recording of real-time vehicle data
- Command the launch of autonomy systems
- Modify the launch parameters
- Sending the vehicle into different operation modes
- Provide debugging tools and interfaces

Here are some thoughs on what's the goal of such solution, and what are examples of existing solutions.

Purpose

The BS is not to be confused with human machine interface (HMI) which was in a couple of previous software WG discussions targeting end-user. It should focus more on the developer experience, who is working on a in-progress vehicle. The difference being:

1. BS displays more internal data that helps debugging and evaluation, e.g. planning FPS, lateral error, detection confidence
2. BS has an easy interface with online param tuning
3. BS has more relaxed rules about starting/restarting autonomy stack in the middle of a run
4. On a vehicle that is unable to carry a BS and an engineer onboard (scaled cars, IAC race cars, etc.), BS needs to maintain performance in a low bandwidth, long-range, lossy wireless communication
5. BS should be more customizable and easy to evolve with the autonomy software
6. BS user can be considered to have rich Autoware, ROS and Linux system knowledge and can be expected to use these tools to customize their basestation

Category

🙏 Feature requests

Labels

None yet

1 participant

Existing Examples

Rviz and Rqt are classic examples of BS solution in the ROS world. Together they provide ways of visializing 2D and 3D data, and enable the manual manipulation of ROS topics, parameters, and services. However, they are also easy to crash in the middle of a run, have little optimization for remote, lossy, and low-bandwidth scenario, and does not integrate with the ROS launch system. They are good examples of what would work for developers sitting comfortably in front of a simulation computer, but not necessary for on-site developers, especially in racing ODD.

Foxglove and ROS Board are more modern-themed telemetry tools, both supporting web-based deployment. But in general they are weak on the "control" side on BS where the developer needs an easy access to modify parameters, alter robot behaviors, or launch and re-launch systems on the fly.

Many teams in the IAC competition develop their own mini telemetry solutions around ROS to address their needs. The Hawaii-UCSD team uses Z1 Analyzer which is a motosport-grade telemetry tool optimized for real-time, but short of any commanding capability. Some other teams simply build their own mini tool in Matplotlib.

Conclusion

In general I see a void in the realm of telemetry and basestation solutions, which targets developers in an on-site AV development, especially in the racing ODD and university education/research. Any thoughts on what Autoware can do with this, or any other existing solutions that may help with this would be appreciated.

↑ 1

0 comments