Autoware API for Remote / Direct Control #5369

Unanswered **messi49** asked this question in **Feature requests**

messi49 last month

Backgroud

In the development of vehicles without steering wheels, accelerators, or brakes, as well as remote operation systems, there is a need to perform low-level controls—such as steering, acceleration, and braking—through Autoware.

Proposal

Develop an API that enables low-level control such as steering, accele, and brake.

The objects to be manipulated are as follows.

- steering wheel
- Accele
- Brake
- Gear
- Turn signals
- Horn
- Wipers
- Headlights
- Parking brake

A system must be in place to handle the above interface (IF) safely.
When using the control API, there must be a mechanism to ensure a safe stop if an issue occurs between Autoware and the application. It should be clear which safety measures are guaranteed by the API and which are the responsibility of the application side.

It would be beneficial to add a pre-crash safety system for operations conducted via the control API.

↑ 1 👀 1

Category

🙏 **Feature requests**

Labels

None yet

5 participants

5 comments · 10 replies

Oldest Newest Top

isamu-takagi 3 weeks ago Maintainer edited ▾

Thank you for the suggestion. I will design a vehicle remote control API based on the following
These pages are used as references for vehicle interface information.

- https://autowarefoundation.github.io/autoware-documentation/main/design/autoware-interfaces/components/vehicle-interface/
- https://autowarefoundation.github.io/autoware-documentation/main/design/autoware-architecture/vehicle/

The only required interfaces are steer and speed, everything else is optional. Therefore, I will create an interface for each operation target and provide a list of functions to be provided (I am considering this as a common mechanism for AD API).

For speed control, I will only design pedal control this time. However, I will also be able to take other models, such as target acceleration and trajectory. For steering control, it will specify the tire angle.

For other optional interfaces, first implement the following, which are currently widely implemented:

- gear
- turn indicators
- hazard lights

Regarding safety

- Applications must guarantee the correctness of the commands they publish.
- Autoware must detect disconnection of the required interfaces during remote operation and perform MRM.
- If you want to use it at level 2 or 3:
  - Autoware needs to detect disconnection even during autonomous driving.
  - Application must determine whether the driver is ready to take over immediately.

↑ 1 👍 1 0 replies

RogerJung 2 weeks ago

Hello, I am from NEWSLAB, a research team in Taiwan.

Based on our experience, we have created a simplified version of the testbed for remote control using a Logitech G923 controller and a 1:8 scale vehicle. We have already mapped the steering control, throttle, and brake messages to the Autoware AckermannControlCommand.

Here is the code link, and I hope this information is helpful to you.

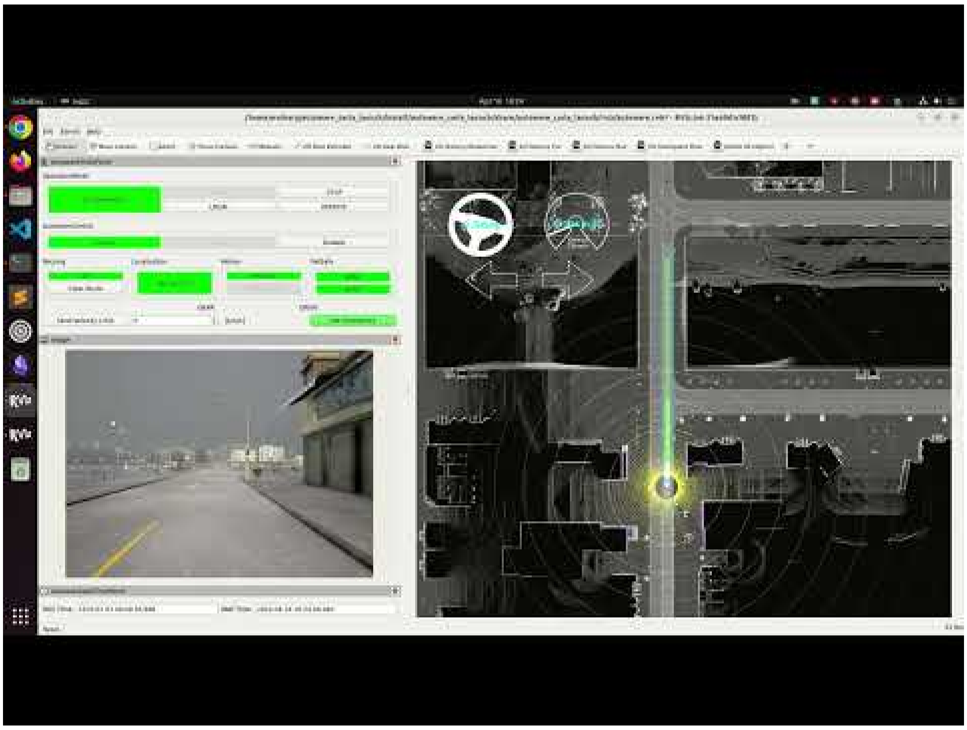
↑ 1 ❤️ 2 0 replies

hsule last week

Hello,

We are a team from ADLINK Technology, working on a fleet management system prototype, `zenoh_autoware_fms`, based on Zenoh. GitHub repo here. Our goal is to enable fleet management through a web interface that allows users to interact with Autoware via Zenoh, supporting remote driving, vehicle monitoring, destination setting, and more.

To understand our remote driving use case more clearly, see the demo video below:



Key Requirement: Our web interface is designed for direct speed setting rather than pedal control. Currently, we use the `/external/selected/control_cmd` topic to manage longitudinal speed and steering angle, an API specifically for target speed would be helpful.

Autoware Topics & Services in Use

Below is the list of topics and services currently in use in the FMS:

Topic

Category	Name	Type
Pose	/api/vehicle/kinematics	autoware_adapi_
Pose	/api/routing/route	autoware_adapi_

Category	Name	Type
Pose	/planning/mission_planning/goal	geometry_msgs/
Pose	/control/gate_mode_cmd	tier4_control_ms
Status	/api/external/get/cpu_usage	tier4_autoware_r
Status	/api/external/get/vehicle/status	tier4_autoware_r
Teleop	/api/external/get/vehicle/status	tier4_autoware_r
Teleop	/control/gate_mode_cmd	tier4_control_ms
Teleop	/api/external/set/command/remote/shift	tier4_autoware_r
Teleop	/external/selected/control_cmd	autoware_contro
Camera	/sensing/camera/traffic_light/image_raw	sensor_msgs/ms

Service

Category	Name	Type
Pose	/api/operation_mode/change_to_autonomous	autoware_a

Category	Name	Type
Teleop	/api/operation_mode/change_to_remote	autoware_a

Please let us know if you have any question. Thank you!

↑ 1 👍 2 2 replies

isamu-takagi last week Maintainer

> /planning/mission_planning/goal (AW currently does not provide an API for setting goals)

FYI: You can use [/api/routing/set_route_points]() to set goal (`waypoints` field can be empty).

👍 1

hsule 3 days ago

@isamu-takagi, thanks for your help! We're using `/api/routing/set_route_points` now, and it works well.

isamu-takagi last week Maintainer

@RogerJung **@hsule** (**@messi49**)

Thank you for sharing. After seeing your code, I realized that there is a demand for control with `/external/selected/control_cmd` , so I would like to consider supporting it. Control.msg has velocity and acceleration, which model are you using?

- target velocity (and optionally, min/max acceleration/jerk)
- target acceleration (and optionally, min/max jerk)

↑ 1 5 replies

isamu-takagi last week Maintainer

And we also need to decide how to handle gear:

- There is no gear, and a positive value moves forward, a negative value moves backward.
- There is gear, and a positive value moves in the gear direction, a negative value stops.

evshary last week Collaborator

Do we need to consider other configurations? Besides REVERSE, PARKING, DRIVE, there is something like NEUTRAL or LOW.
https://github.com/tier4/tier4_autoware_msgs/blob/tier4/universe/tier4_external_api_msgs/msg/GearShift.msg

evshary last week Collaborator

Also, a little off-topic, we would like to know your opinion on video streaming (for remote driving usage). I know that this is not efficient to send with ROS message type, but should we provide an API to share sensor data?

isamu-takagi last week Maintainer

> Do we need to consider other configurations? Besides REVERSE, PARKING, DRIVE, there is something like NEUTRAL or LOW.

@evshary There are a lot of gears defined in the vehicle interface. We could provide an API to retrieve a list of gear supported by the vehicle and let users select from there, but this is probably not necessary for most use cases.

- https://github.com/autowarefoundation/autoware_msgs/blob/main/autoware_vehicle_msgs/msg/GearCommand.msg
- https://github.com/autowarefoundation/autoware_msgs/blob/main/autoware_vehicle_msgs/msg/GearReport.msg

In most cases, DRIVE, REVERSE, PARKING and NEUTRAL will probably be enough. Therefore, I would like to support only this for now. What do you think?

evshary last week Collaborator

@isamu-takagi Thanks for your answer. That makes sense to me.

isamu-takagi last week Maintainer

> Also, a little off-topic, we would like to know your opinion on video streaming (for remote driving usage). I know that this is not efficient to send with ROS message type, but should we provide an API to share sensor data?

@evshary To ensure safety, 360-degree video and audio need to be transmitted at a sufficient frequency, but we cannot guarantee that the API can meet this requirement. This is because Autoware cannot check whether the video has been received, and therefore cannot trigger MRM properly.

↑ 1 👍 1 3 replies

evshary last week Collaborator

I see. Then users need to have their own ways of getting the video/audio. BTW, could you tell me what the abbreviation MRM is?

isamu-takagi last week Maintainer edited ▾

MRM stands for "minimum risk maneuver". This is the action (e.g. emergency stop) that Autoware should take when it detects an anomaly.

Also, this is just a thought, it might be possible to implement streaming API if there was a mechanism for applications receiving video or audio to detect abnormalities and report them to Autoware. But in the first step, I would like to only consider entering commands.

evshary last week Collaborator

I agree with you. Let's focus on the control commands first. Thank you for the explanation.