

ISASE 2020

# An Operator Interface for Autonomous Vehicles

Zhi WANG<sup>1</sup>, Hideyasu SAI<sup>2</sup>, Kazuo OGIWARA<sup>3</sup>, Daishi WATABE<sup>4</sup>, Yukimichi SAITO<sup>5</sup>,  
Masayoshi WADA<sup>6</sup>

<sup>1</sup> Saitama Institute of Technology, 1690 Fusaiji, Fukaya, Saitama  
t7001vuk@sit.ac.jp

<sup>2</sup> Saitama Institute of Technology, 1690 Fusaiji, Fukaya, Saitama  
sai.hideyasu@sit.ac.jp

<sup>3</sup> Saitama Institute of Technology, 1690 Fusaiji, Fukaya, Saitama

<sup>4</sup> Saitama Institute of Technology, 1690 Fusaiji, Fukaya, Saitama  
dw@sit.ac.jp

<sup>5</sup> Mikuni Life &Auto, 456-1 Maguchi, Kazo, Saitama  
yukimichi\_saitou@mikuni-la.co.jp

<sup>6</sup> Tokyo University of Agriculture and Technology, 2-24-16 Naka, Koganei, Tokyo  
mwada@cc.tuat.ac.jp

**Abstract:** In this paper, we propose an interface for autonomous vehicles that can build trust between users and computers. Several open-source pieces of software (such as Autoware and Apollo) are widely used for autonomous driving. A few researchers have also tested their algorithms on simulators such as LGSVL. To identify a broader range of problems that would not be possible by means of simulator testing, such as the sudden appearance of pedestrians in front of the autonomous vehicle, bad weather, or roads in poor condition, we attempt to test these algorithms under live conditions. Further, we design an interface for an actual autonomous vehicle. The interface is aimed at quickly tuning the parameters of various algorithms, monitoring the status of the vehicle, and most importantly, notifying the operator, driver, and passengers concerning the drive mode of the vehicle and help develop trust in autonomous vehicles. The interface was designed to provide as much relevant information as possible and enable more accessible communication with the vehicle. Accordingly, this interface can help to improve the safety of autonomous vehicles.

**Keywords:** *Autonomous Vehicle, Operator Interface, Human Machine Interface*

## 1. INTRODUCTION

In recent years, the development of autonomous vehicles has made rapid progress. An increasing number of companies and schools have joined together in researching autonomous cars, backed by a Japanese government that strongly supports the development of autonomous vehicles. Based on the cross-ministerial Strategic Innovation Promotion Program (SIP), several autonomous experiments have been carried out in sparsely populated areas, areas with aging populations, and certain downtown areas. The authors of this paper have researched autonomous cars for over three years, and have in their possession a Prius (5-seater) and a Liesse (24-seater), supported by the technology to realize autonomous driving. These cars are based on open-source software, Autoware and equipped with sensors such as LiDAR, Radar, cameras, GNSS and Mobileye, among

others. The car is controlled by means of an acceleration, brake, and steer actuator via a controller area network (CAN) bus. In this paper, we propose a Human-Machine Interface (HMI) for an autonomous vehicle that differs from the traditional interface that not only displays data concerning the acceleration, braking and steering but can be used to monitor the status of autonomous vehicles in real-time.

## 2. BACKGROUND

### 2.1 Autonomous Driving Technology

Over the years, considerable research has been conducted on autonomous vehicles, which can broadly be categorized as perception, fusion, decision, and control. For vehicle control, a few well-known algorithms, such as the proportion integration differentiation controller (PID controller) or model predictive controller (MPC

controller), are widely used. However, adjusting the parameters governing MPC is a difficult task. For obstacle detection using cameras, several popular methods such as You Only Look Once (YOLO) exist, and methods such as Point Pillars have been developed to be used with LiDAR. For traffic light detection, traditional methods based on the HSV color space can be implemented, or deep learning methods can be employed. Localization can be achieved either by means of LiDAR or Global Navigation Satellite System (GNSS). Using LiDAR, a representative method that is employed is normal distributions transform matching (NDT matching). However, several difficulties are encountered while processing a large number of 3D points at high speed. While employing the GNSS, real-time kinematic (RTK), and inertial measurement units (IMU) can be used to supplement measurements to provide high precision positional information. This data can be relied on in areas without many buildings that could potentially block the signal of the GNSS. On solving the aforementioned four problems plaguing autonomous vehicles, a steady communication stream needs to be established with the vehicle. Communicating with autonomous vehicles is also a crucial problem. In the early stages of autonomous vehicle development, Sadigh[1] proposed a solution to the problem of human interaction with autonomous vehicles, that helps display sufficient and crucial information to the driver. Autonomous driving at different levels requires an appropriate interface to allow the driver or operator to establish accurate situational awareness, and reengage in the driving task [2]. Based on these works, not only is data taken from the CAN bus or sensors and displayed via the interface, but an interface is proposed for the operator and designed to enable the commissioning of autonomous driving, making it easier and more efficient.

## 2.2 Human Machine Interface

The autonomous car employed for this study is a Prius that uses an interface called HEV. The interface was created by the ZMP company. HEV displays information such as the steering angle, the throttle value while accelerating and braking, and other information like battery performance, temperature, and even enables the acquisition and setting of parameters concerning the PID controller, among others. For Autoware, there also exists a built-in interface, based on this interface, that enables commands to be sent to the autonomous vehicle to control actions such as acceleration, braking, and steering. Moreover, actions such as lane changing or avoiding obstacles can be controlled by means of this interface. In

addition, commands concerning the condition of the traffic light can also be sent to the vehicle. These functions, in Autoware, can enable a vehicle to be smarter. However, this interface is unable to obtain information from the autonomous vehicle, and it is difficult to provide security guarantees. To solve these problems, an interface for autonomous vehicles has been designed that can help users monitor the status of the vehicle while the autonomous vehicle is being run. This interface can also help the operator and driver determine whether the vehicle is in autonomous driving mode, and is useful in helping drivers transition their vehicles from manual mode to program mode at a correct time even when the vehicle is operating at high speeds.

## 3. Operator Interface

### 3.1 Vehicle CAN Data

We use Kvaser to communicate with the Autoware computer and the CAN bus of the vehicle microcomputer. Data from the CAN bus can be acquired as the actual value and data from Autoware computer can be obtained as the target value. To monitor the status of the vehicle in real-time, it is necessary to display the acceleration, braking and steering information via the interface.

### 3.2 Driving Mode

For autonomous vehicles, there are program and manual modes (see Figure 1). The throttle brake can be set to automatic mode and the steering can be set to manual mode, or the steering can be set to automatic mode and the throttle braking to manual mode. The driving mode can be changed at any time. It is necessary for drivers and operators to be aware of the control mode while driving. The interface displays the driving mode and the throttle braking and steering separately.

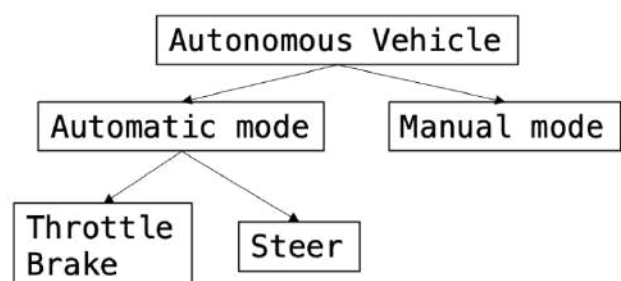


Figure 1: Control mode of autonomous vehicle

### 3.3 Offset distance of the path

Autonomous vehicles are at a stage of rapid development, and we are committed to making the car track the trajectory as smoothly as possible. However,

several complications arise during the process. To overcome these problems, we calculate the Euclidean distance from the current pose to the closest waypoint as the offset distance of the path. By displaying the offset distance, we can tune the course easily, and try to keep the vehicle in the middle of the road.

Offset distance of the path is calculated by means of Equation (1).  $\alpha$  is the current pose and  $\beta$  is the closest waypoint.

$$distance = \sqrt{(\alpha.x - \beta.x)^2 + (\alpha.y - \beta.y)^2} \quad (1)$$

### 3.4 Sensor information

Autonomous vehicles are usually equipped with a variety of sensors. The normal operation of various sensors is often affected by many factors. We use GNSS and LiDAR for localization. The GNSS localization error is crucial for autonomous vehicles, as it directly determines whether an autonomous vehicle can run automatically. It is usually determined by latitude, longitude, and altitude, shown as equation 2.

$$error = \begin{cases} \text{standard deviation of latitude} \\ \text{standard deviation of longitude} \\ \text{standard deviation of altitude} \end{cases} \quad (2)$$

The NDT matching score is the most important for LiDAR- lower the score, the better the LiDAR data fits the 3D map, and the higher the accuracy of the estimated position. NDT score can be calculated using Equation 3. NDT matching score is the most important for LiDAR- lower the score, the better the LiDAR data fits the 3D map, and the higher the accuracy of the estimated position. NDT score can be calculated using Equation 3.

$$score = \sum_i \exp\left(\frac{-(x'_i - q_i)^t \sum_i^{-1} (x'_i - q_i)}{2}\right) \quad (3)$$

All the information is shown on the interface in real-time. The background is displayed in light blue when the localization accuracy is high enough and is red when the error is too large, or the accuracy is too low.

It is necessary to show the localizer being used to highlight the GNSS localization error and NDT matching score. It is often displayed as “NDT + ODOM” or “GNSS + GYRO,” indicating localization by NDT matching and odometry information or by GNSS and gyro information.

### 3.5 Beep

Due to the emergence of models using auditory salience direction to guide visual attention [3], we use beep

information as part of the interface. When the current position of the vehicle is too far from the waypoint or localization error is too large, the automatic mode will be cut off and change to manual mode. The computer will beep to notify the driver or operator. It is very dangerous if the driver or operator cannot sense that the automatic mode is cut off for the first time.

### 3.6 Temporary stop line and other information

To tune parameters of the autonomous vehicles, we also show other information in interface, such as the yaw, the roll, the pitch of a vehicle, the stop distance from the temporary line, the number of waypoints, among others. For current autonomous driving, much information is provided by the ADAS map containing location information of traffic lights, crosswalks, temporary stops, lanes, etc. Therefore, whether the information can be used correctly and accurately is a current problem, and this interface confirms the information quickly.

## 4. Results

This section shows the two interfaces discussed above and the interface proposed in this article. Figure 2 shows the interface of the HEV mounted on the Prius, showing data from the CAN bus.

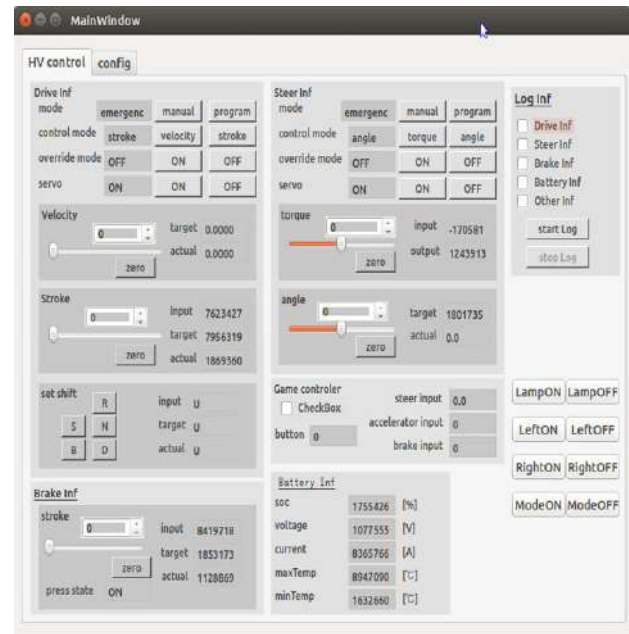


Figure 2: HEV interface

Figure 3 shows the interface build-in Autoware, that does not show any information from CAN bus but can

send various commands to the microcomputer to control the vehicle and make it smarter.

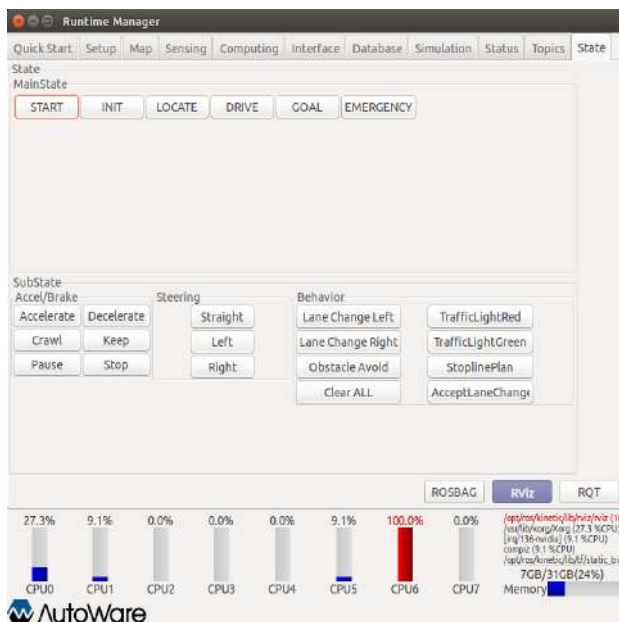


Figure 3: Autoware build-in interface

Figure 4 is our proposed interface. It is similar to Figure 2, and shows some CAN data, sensor information, and control commands in the interface. We also use color as auxiliary information to indicate whether the autonomous vehicle can work normally.

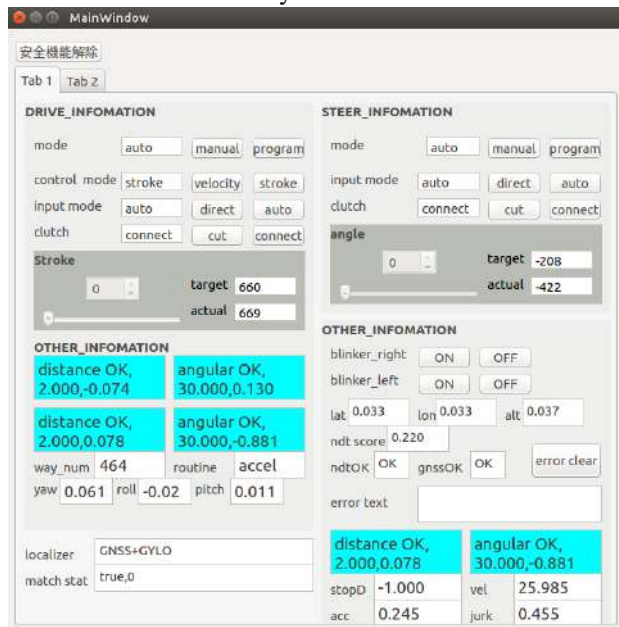


Figure 4: Proposed interface used for operator

### 5. Future work

Autonomous vehicles have been an invention 100 years in the making. Since the 1920s, automakers have experimented with autonomous driving. Autonomous vehicles regained popularity recently and have become a

reality under certain conditions. Simultaneously, interfaces have evolved with autonomous driving levels. We propose a convenient interface for autonomous vehicle operators, who can use it to find problems in autonomous driving and debug them quickly. It is necessary to design an interface for drivers or passengers. There is no extra debugging data, sensor data, or CAN data because these data are not of concern to passengers or drivers. However, what passengers need is to be able to ride in autonomous vehicles safely and be informed of the travel time to their destination. The interface should be personalized and communicate politely with drivers or passengers and should notify them via some simple color changes or beeps.

### REFERENCES

- [1] D. Sadigh, K. Driggs-Campbell, R. Bajcsy, S. S. Sastry, and S. A. Seshia, "User interface design and verification for semi-autonomous driving," *HiCoNS 2014 - Proc. 3rd Int. Conf. High Confid. Networked Syst. (Part CPS Week)*, pp. 63–64, 2014, doi: 10.1145/2566468.2576851.
- [2] S. Debernard, C. Chauvin, R. Pokam, and S. Langlois, "Designing Human-Machine Interface for Autonomous Vehicles," *IFAC-PapersOnLine*, vol. 49, no. 19, pp. 609–614, 2016, doi: 10.1016/j.ifacol.2016.10.629.
- [3] E. M. Kaya and M. Elhilali, "Modelling auditory attention," *Philos. Trans. R. Soc. B Biol. Sci.*, vol. 372, no. 1714, 2017, doi: 10.1098/rstb.2016.0101.
- [4] Autoware Official Website. <https://www.autoware.ai> [Accessed Jan 25, 2020]
- [5] Apollo Official Website. [http://apollo.auto/index\\_cn.html](http://apollo.auto/index_cn.html) [Accessed Jan 25, 2020]
- [6] Autoware branch for Saitama Institute of Technology <https://github.com/saihideyasu/autoware> [Accessed Jan 25, 2020]
- [7] LGSVL Official Website. <https://www.lgsvlsimulator.com> [Accessed Jan 25, 2020]