AUTONOMOUS NAVIGATION LAB

autonav.inha.ac.kr

Study on Full Scale Testbench Based Vehicle-In-The-Loop Simulation Testing for Automated Vehicles

by Yong-Ha Lee, Jong-Hoon Won*

Department of Electrical and Computer Engineering, Inha University

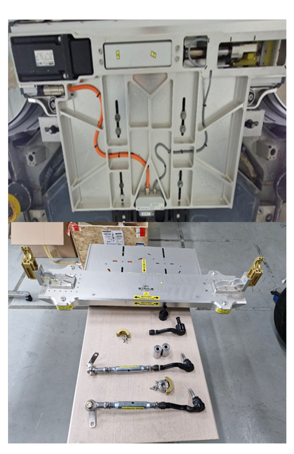
인하대학교 INHA UNIVERSITY 1954

Introduction

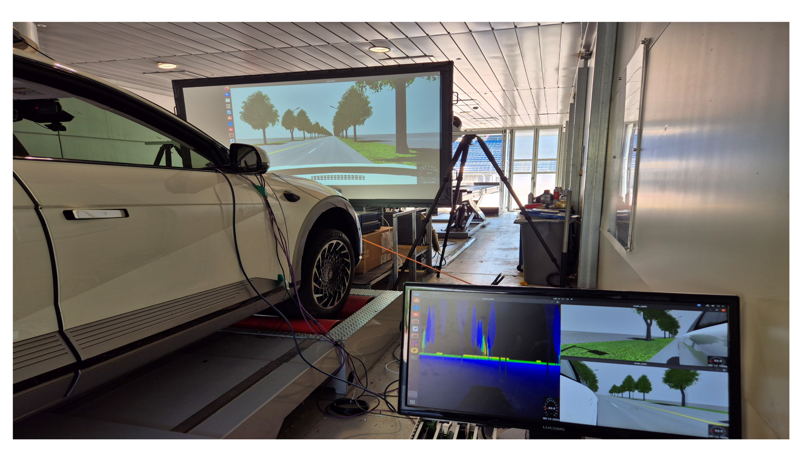
❖ **Testbench-Based Full-Scale ViLS**
- Enables testing of physical properties without spatial constraints.
- Utilizes a chassis dynamometer to capture vehicle behavior.
- Delivers sensor data about to the autonomous vehicle controller.

❖ **Challenges in Current Methods**
- Integration of LiDAR:
  - Significant challenge in incorporating LiDAR into the loop for full-scale testing.
  - Existing methods struggle to conduct VILS testing that includes LiDAR sensors.
- Integration of Omni-directional Cameras:
  - 360-degree configurations can lead to image distortion, affecting data accuracy.
  - Implementing multiple cameras for a complete view can be economically inefficient.

❖ **Proposed Solution**
- Sensor Data Injection Method:
  - Implements a solution to provide high-capacity sensor data directly to the autonomous vehicle controller.

< Automated vehicle and chassis dynamometer and Steering force emulator >

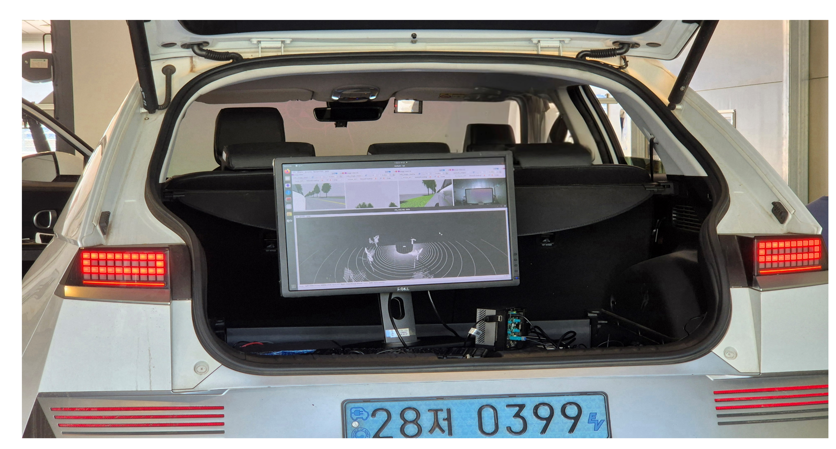


< ViLS testing demonstration with sensor data injection >

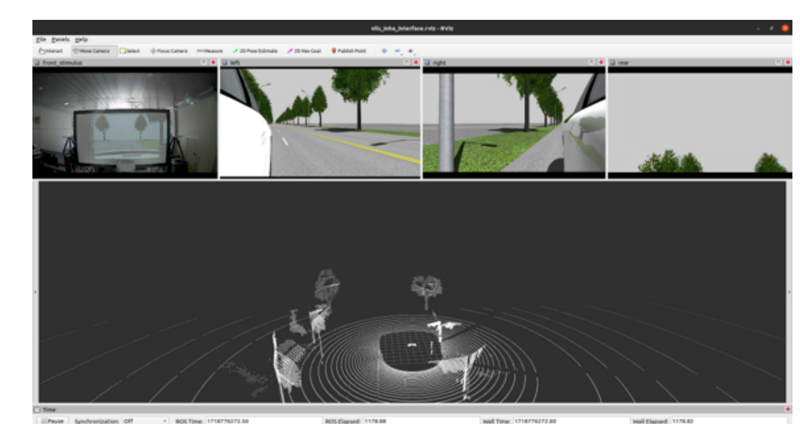
Sensor Data Injection

❖ **Purpose of Sensor Data Injection Method**
- A method for delivering simulated sensor outputs from a driving simulator to the autonomous driving controller of a real vehicle.
- Although sensor data stimulation presents technical challenges, it is crucial for executing fully autonomous driving algorithms by providing high-volume, high-level data directly from LiDAR and surround-view cameras to the autonomous driving controller.

❖ **Importance of Proposed Method**
- Enables the construction of a full-scale ViLS system, including LiDAR and omnidirectional cameras, where stimulation is not feasible.
- Provides a flexible system for LiDAR sensor injection that is independent of the manufacturer's parsing methods.
- Implements real-time data injection through a single Ethernet channel by threading multiple sensors for parallel operation.
- Validates the effectiveness of the injection system through transmission tests and data analysis with the autonomous driving controller of actual vehicles.



< Sensor data injection into the automated vehicle's controller >



< GUI displaying injected sensor data in the automated vehicle >

Transmission Method

❖ **Transmission Method for Injection**
- (1) For LiDAR data, transmission of point cloud data in batches of 4500 points every 2500 μs, ensuring timely updates and maintaining the integrity of the data.
- (2) For camera data, transmission of camera image data over UDP in compressed JPEG chunks, where the image is divided into chunks based on the image size and chunk size, transmitted at intervals of 25 m/s to ensure smooth data flow.

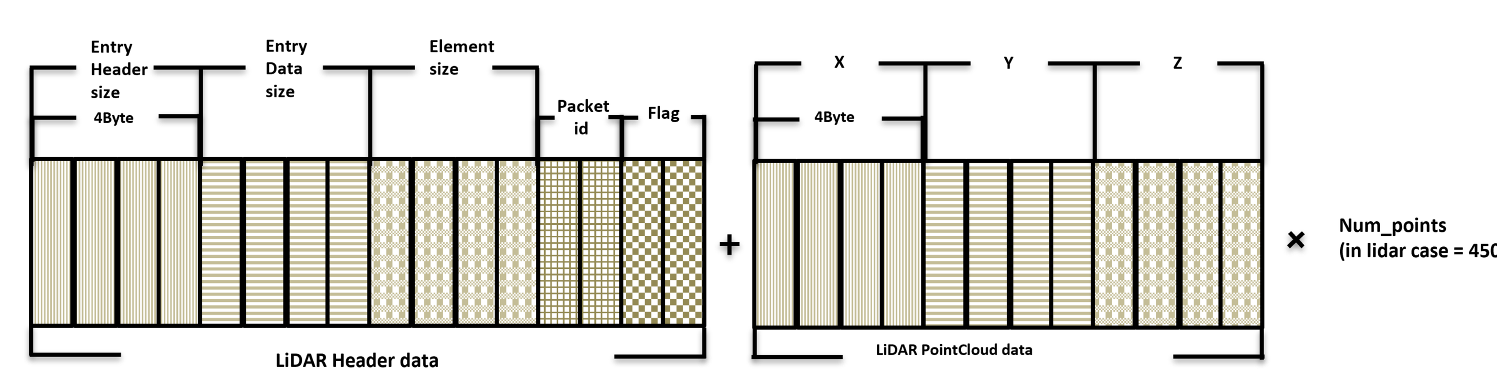
$$TransmitLiDARData = \sum_{i=0}^{num_{points} \div 4500 - 1} Send\left(\{X_i, Y_i, Z_i\}\right) \cdot \delta\left(t_i - t_{prev} \geq 2500\right) \tag{1}$$

$$TransmitCameraData = \sum_{i=0}^{img_{size} \div chunk_{size} - 1} Send\left(CompressedImageChunk_i\right) \cdot \delta\left(t_i - t_{prev} \geq 25000\right) \tag{2}$$

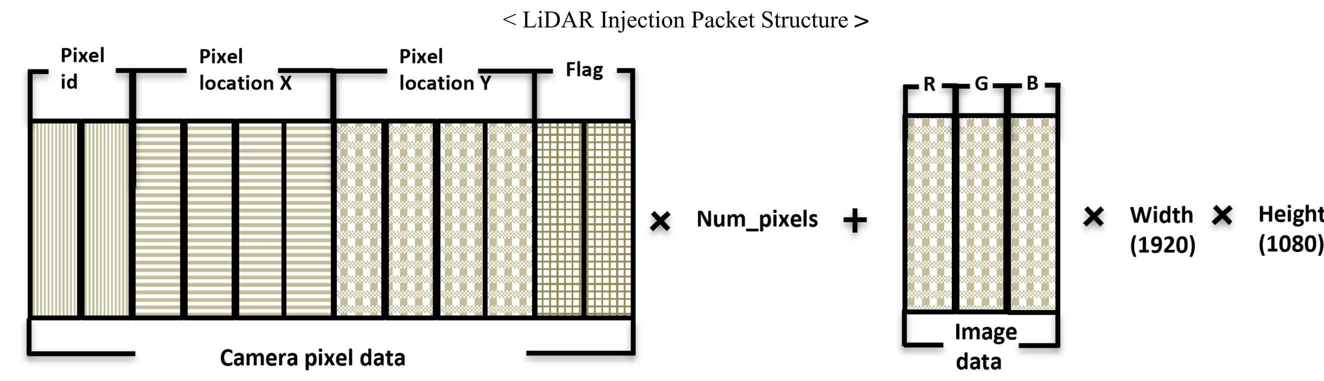
Where:
num_{points} : The total number of LiDAR points to be processed. $Send\left(\{X_i, Y_i, Z_i\}\right)$: Function that sends a block of 4500 points.
$\delta(t_i - t_{prev} \geq 2500)$: A condition that ensures data transmission occurs every 2500 microseconds
img_size: Total size of the image data (in bytes or pixels). chunk_size: The size of each image chunk being transmitted
$Send(CompressedImageChunk_i)$: Function that sends a compressed chunk of the image.
$\delta(t_i - t_{prev} \geq 25000)$: Condition ensuring data transmission occurs every 25000 microseconds (to maintain a 40Hz cycle).
t_i : Current time. t_{prev} : Previous transmission time.

Sensor Packet Structure

❖ **Sensor Data Packet Structure for Injection Method**
- We designed a specific packet structure for direct sensor data transmission via the injection method.
- For LiDAR, we constructed packets based on point cloud positions instead of using a specific parsing method from any LiDAR manufacturer to ensure universal applicability.
- For camera data, JPEG compression is implemented to enhance transmission efficiency.



< LiDAR Injection Packet Structure >



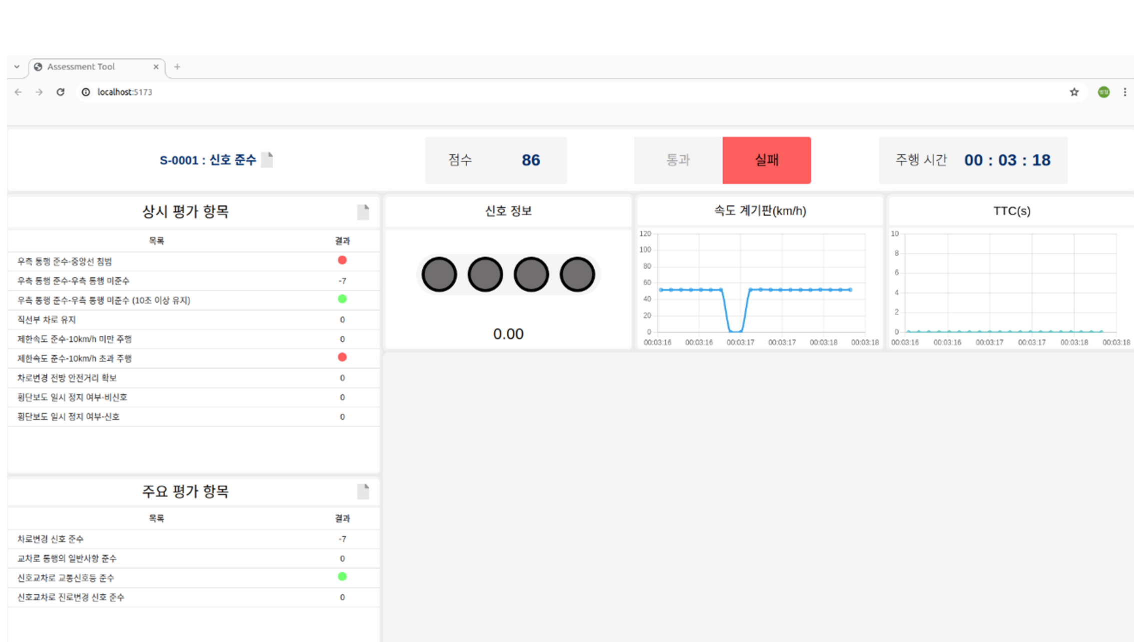
< Camera Injection Packet Structure >

Experimental Simulation

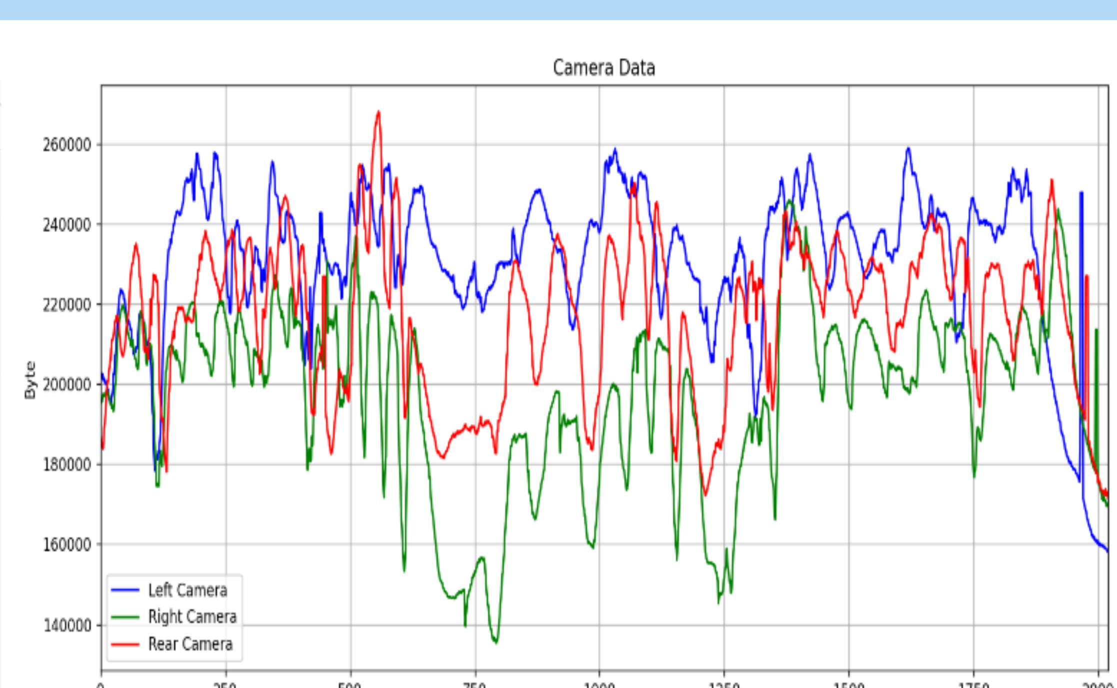
❖ **Camera Data Experiment in ViLS**
- Simulated front Full HD camera and 3 HD side/rear cameras.
- When OTA stimulation was available for the front camera, focused on injecting side/rear camera data with LiDAR data.
- The y-axis represents the byte length of the JPEG compressed camera data, while the x-axis shows the number of generated camera data packets in the entire sample scenario

❖ **LiDAR Data Experiment in ViLS**
- Injected data starting from 16-channel to 80-channel multi-LiDAR point clouds (360° scan).
- Targeted 20 Hz for single-frame processing to assess controller capacity for large LiDAR data.

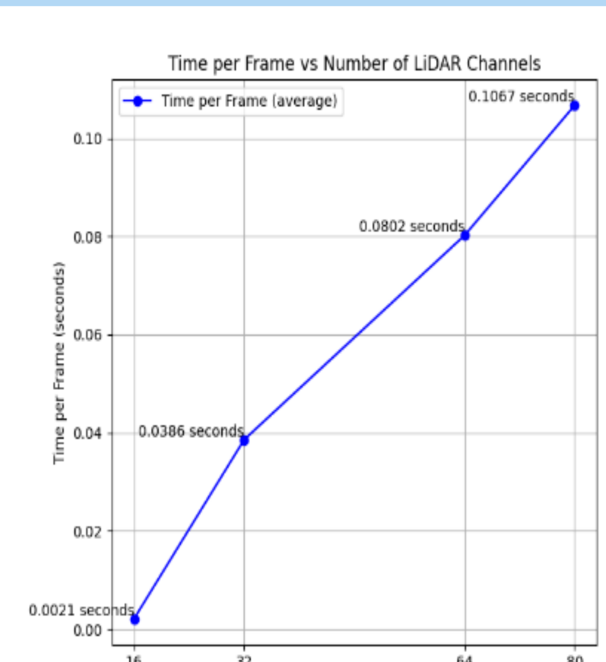
❖ **Result Overview**
- LiDAR and camera injections were performed using a single Ethernet cable with a UDP method. The analysis focused on the impact of the number of LiDAR channels, while for the cameras, the packet size transmitted for scenarios involving three HD cameras was evaluated using the ViLS automated evaluation tool. The final driving performance of the autonomous vehicle was assessed through these evaluations

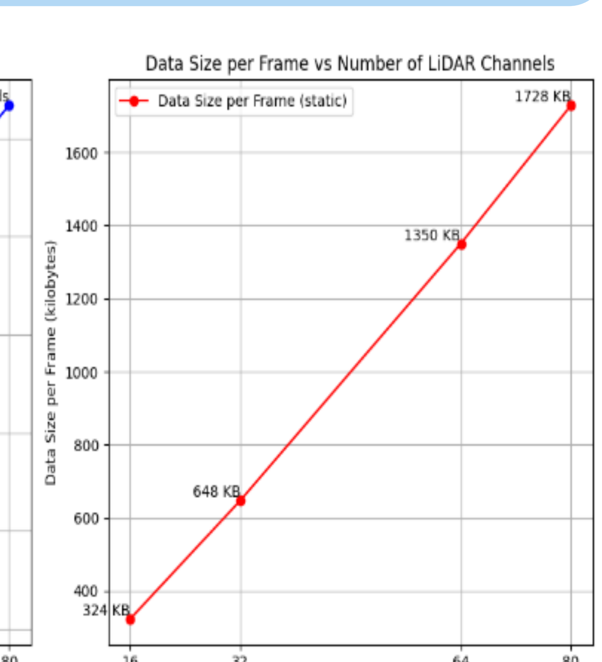


< ViLS automated evaluation tool GUI >



< Injected camera data: y-axis shows byte length of JPEG-compressed data; x-axis indicates the number of generated data packets. >





< Injected LiDAR data: Left y-axis shows time (sec) per data frame; right y-axis shows data volume (kb) per frame; x-axis indicates the number of LiDAR channels. >