

# ROS 기반 자율주행 알고리즘 성능 검증을 위한 시뮬레이션 환경 개발

곽지섭\* · 이경수\*\*,<sup>†</sup>

## Development of Simulation Environment for Autonomous Driving Algorithm Validation based on ROS

Jisub Kwak\*, Kyongsu Yi\*\*,<sup>†</sup>

*Key Words: Autonomous driving(자율주행), ROS(로봇 운영 체제), Evaluation scenarios(평가 시나리오), Virtual Sensor(가상 센서), Motion Planning(거동 계획)*

### ABSTRACT

This paper presents a development of simulation environment for validation of autonomous driving (AD) algorithm based on Robot Operating System (ROS). ROS is one of the commonly-used frameworks utilized to control autonomous vehicles. For the evaluation of AD algorithm, a 3D autonomous driving simulator has been developed based on LGSVL. Two additional sensors are implemented in the simulation vehicle. First, Lidar sensor is mounted on the ego vehicle for real-time driving environment perception. Second, GPS sensor is equipped to estimate ego vehicle's position. With the vehicle sensor configuration in the simulation, the AD algorithm can predict the local environment and determine control commands with motion planning. The simulation environment has been evaluated with lane changing and keeping scenarios. The simulation results show that the proposed 3D simulator can successfully imitate the operation of a real-world vehicle.

### 1. 서론

최근 자동차 산업에서는 자율주행 연구가 활발히 이뤄지고 있다. 구글, 테슬라, 현대자동차 등 여러 기업들이 세계 곳곳에서 자율주행 자동차의 상용화를 위해 기술을 개발하고 있으며, 고도의 주행 보조 시스템을 탑재한 차량이 시중에 유통되고 있다. SAE International에서는 자율주행차량의 발전단계를 6단계로 정의했으며, 운전자가 주행의 모든 것을 통제하고 책임지는 0단계부터, 차량의 자체 주행 시스템으로만 주행이 이뤄지는 5단계까지 나뉘었다.

현재 상용 차량에 탑재된 주행 보조 시스템은 완벽한 자율주행이 불가능하기 때문에 비상시에는 운전자의 개입을 전제로 한다.

따라서 운전자가 필요하지 않은 5단계 완전 자율주행 목표를 달성하기 위해 해당 연구에 대한 개발이 활발하게 진행중이며, 이에 대한 알고리즘의 성능을 검증하고 분석하기 위해서는 실차 환경에서의 점검이 필수적이다. 하지만, 카메라, 라이다, GPS를 비롯한 다양한 센서가 탑재된 자율주행 차량을 제작하기 위해서는 상당한 비용을 필요로 하며, 차량을 포함한 실차 실험을 위한 환경을 조성하고 실험을 진행하는 것 또한 많은 시간과 인력이 소요된다. 따라서 실차 실험 환경을 모방할 수 있는 시뮬레이션 환경이 조성된다면 가상 환경으로 실차 실험을 대체할 수 있으며, 시뮬레이션 환경을 통해 검증된 알고리즘에 대해

\* 서울대학교 기계공학부, 석박통합과정

\*\* 서울대학교 기계공학부, 교수

<sup>†</sup>교신저자: [kyi@snu.ac.kr](mailto:kyi@snu.ac.kr)

E-mail: [kjs675252@snu.ac.kr](mailto:kjs675252@snu.ac.kr)

서만 실차 실험을 진행한다면 막대한 비용과 시간을 절감하는 효과를 가져다 줄 것이다. 또한 새롭게 개발한 알고리즘은 안전성이 검증되지 않은 상태이므로 이를 바로 차량에 적용하여 실험을 진행할 경우에는 실험자와 운전자의 안전을 위협할 수 있기 때문에 시뮬레이션 환경에서 새로운 알고리즘을 적용하는 것이 안전성 측면에서 우수하다.

본 논문에서 검증한 알고리즘에 활용된 로봇 운영 체제(ROS)는 로봇 프로그래밍에 널리 사용되는 미들웨어 플랫폼이며 오픈소스로 제공된다.<sup>(1)</sup> ROS는 노드간의 메시지 통신을 통해 데이터를 주고받는다. 오픈소스로 제공되기 때문에 범용성이 높으며, 데이터를 가시화하기에 편리하다. 하지만 Carsim, Carmaker와 같은 차량 시뮬레이터는 차량 센서들을 자유롭게 구성하는 것에 제약이 있으며, 차량의 정교한 동역학적 특성에 중점을 둔 시뮬레이터이다. 또한 ROS와의 연동에 있어서도 어려움이 있다. 따라서 이를 해결하기 위한 새로운 플랫폼을 도입하여 시뮬레이션 환경을 구축하였으며, 본 논문에서는 LGSVL을 사용하였다.

LGSVL은 LG전자 북미R&D 센터의 이름을 따라 이름이 붙여진 차량 시뮬레이터이며 Unity를 연동해서 사용하는 3D 차량 시뮬레이터이다.<sup>(2)</sup> 이를 통해 자율주행 차량의 센서 구성을 자유롭게 설정할 수 있으며, 차량의 상태 변수 뿐만 아니라 차량 센서의 데이터 출력을 가시화하고 분석할 수 있다. 본 논문에서는 LGSVL을 이용하여 ROS 기반 자율주행 알고리즘의 시뮬레이션 환경을 통합개발하고, 주행 시나리오에 대한 시뮬레이션을 통해 자율주행 알고리즘의 성능을 검증하였다. 또한 이를 실제 차량 실험과 비교하여 시뮬레이션 환경이 실제 차량 주행 환경을 성공적으로 모방하는지에 대해 분석하였다.

## 2. 자율주행 차량

### 2.1. 자율주행 차량 구성

본 논문에서는 LGSVL에서 기본예제로 제공하는 'JAGUAR 2015 XE' 중형세단 차량을 시뮬레이션 차량 모델로 선정하였으며 Fig. 1에 나타나 있다. 주변 도로 상황을 인지하기 위해 라이다 센서를 사용하였으며, 차체 상단에 VLP-16 모델을 탑재했다. 라이다 센서의 경우 매개 변수 설정을 통해 고성능 다채널 라이다를 모델링 할 수 있지만, 실차 환경과 유사한 시뮬레이션 환경을 구성하기 위해 실차에서 사용된 동일한 모델을 채택하였다. Fig. 2는 라이다 센

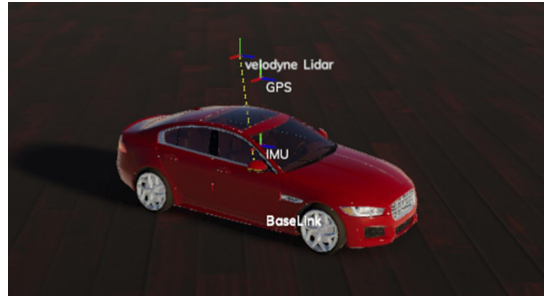


Fig. 1 Sensor configuration of simulation vehicle 'JAGUAR 2015 XE'

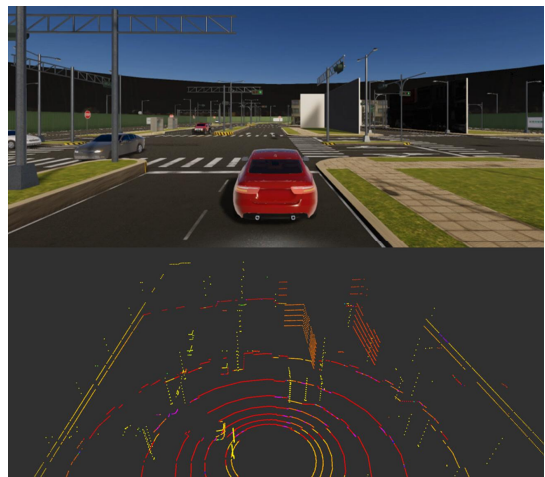


Fig. 2 Lidar sensor data from LGSVL simulator

서 데이터가 출력되는 모습이며, 이를 이용하여 실시간으로 주변 환경 정보를 인지할 수 있다.

추가로 GPS와 IMU 센서를 사용하여 차량의 현재 위치, 속도, 요각을 측정하도록 하였다. 이는 차량 시스템이 주변 환경을 인지하고 향후 거동을 계획하기 위하여 사용된다.

### 2.2. 자율주행 차량 모델링

#### 2.2.1. 차량 알고리즘 구성

본 논문에서 활용된 자율주행 알고리즘의 구조는 Fig. 3과 같다. 해당 알고리즘은 두 개의 모듈로 구성되어 있는데, 인지 모듈과 판단 모듈이 각각에 해당한다. 먼저 인지 알고리즘은 Lee(2020)에 의해 제안된 Geometric model-free approach and static obstacle map based moving object detection and tracking 알고리즘으로서 라이다 센서 데이

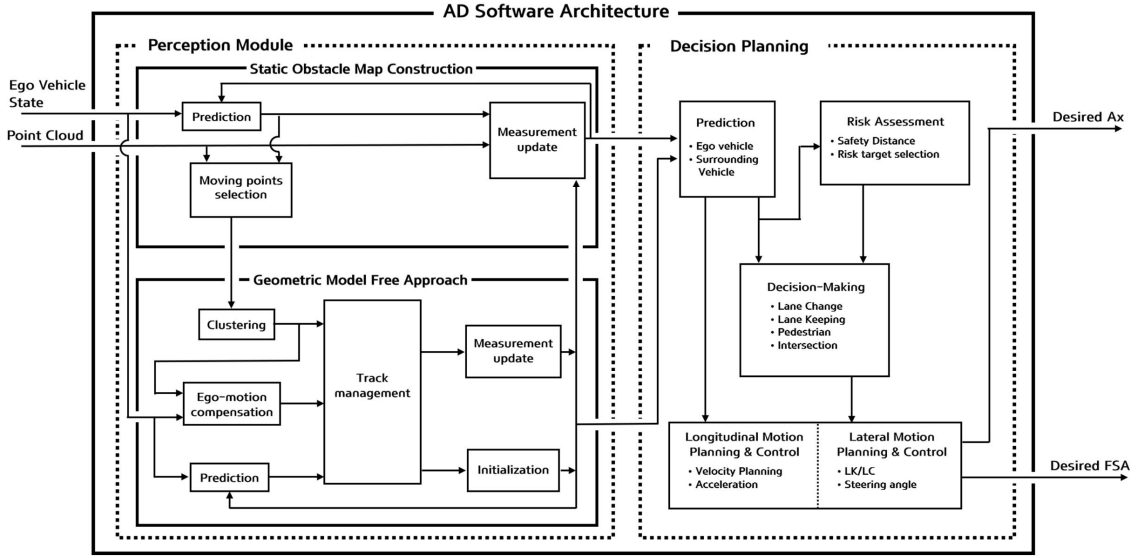


Fig. 3 Autonomous driving algorithm architecture

터와 자차량의 속도, 요각 정보를 바탕으로 정지 물체와 운동 물체를 추정하며 Extended Kalman filter와 Particle Filter를 이용하여 이를 업데이트 한다.<sup>(3)</sup>

다음으로 판단 알고리즘은 Chae(2018)에 의해 제안된 Probabilistic prediction based lane change 알고리즘으로서 인지 알고리즘에 의한 인지 데이터와 자차량의 상태 정보를 이용해 차량의 거동 계획을 확률적 예측을 통해 결정한다.<sup>(4)</sup>

### 2.2.2. 차량 동역학 모델

자율주행 알고리즘은 시간에 따라 변하는 주행 환경에 즉각적으로 반응해야 하므로 실시간으로 동작해야 한다. 즉 알고리즘의 계산 복잡도를 줄이는 것이 중요하다. 따라서 차량의 동역학 모델은 종 방향 동역학 모델과 횡 방향 동역학 모델로 구분되어 있다. 종 방향 동역학 모델은 차량의 종 방향 가속도를 계산하기 위해 설계되었다. 입력 지연이 고려된 종 방향 모델의 상태 공간 모델링은 식 (1)과 같다.

$$\begin{aligned} \dot{\bar{x}}_{lon} &= A_{lon}\bar{x}_{lon} + B_{lon}u_{lon} \\ s.t. \quad A_{lon} &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -1/\tau_a \end{bmatrix}, B_{lon} = \begin{bmatrix} 0 \\ 0 \\ 1/\tau_a \end{bmatrix} \end{aligned} \quad (1)$$

여기서  $\bar{x}_{lon} = [p \quad v \quad a]^T$ 는 상태변수이며 각각의 성분은 위치, 속도, 가속도를 의미한다.  $u_{lon} = a_{des}$ 은 시스템의 입력에 해당하며,  $\tau_a$ 는 종 방향 가속도의 액추에이터 지연을 의미한다.

횡 방향 동역학 모델은 bicycle model과 차선 중심으로부터의 error dynamics을 결합하여 설계되었다. 횡 방향 모델의 상태 공간 모델링은 식 (2)와 같다.

$$\begin{aligned} \dot{\bar{x}}_{lat} &= A_{lat}\bar{x}_{lat} + B_{lat}u_{lat} + F_{lat}\rho \\ s.t. \quad A_{lat} &= \begin{bmatrix} \frac{2C_f + 2C_r}{-mv} & \frac{2C_f l_f + 2C_r l_r}{-mv} & 0 & 0 \\ \frac{2C_f l_f - 2C_r l_r}{-I_z} & \frac{2C_f l_f^2 + 2C_r l_r^2}{-I_z v} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ v & 0 & 0 & v \end{bmatrix}, \\ B_{lat} &= \begin{bmatrix} \frac{2C_f}{mv_x} \\ \frac{2C_f l_f}{I_z} \\ 0 \\ 0 \end{bmatrix}, F_{lat} = \begin{bmatrix} 0 \\ 0 \\ -v_x \\ 0 \end{bmatrix} \end{aligned} \quad (2)$$

이 때  $\bar{x}_{lat} = [\beta \ \gamma \ e_\psi \ e_y]^T$ 는 상태변수이며  $\beta$ 는 횡 방향 슬립 각,  $\gamma$ 는 무게중심에 대한 회전속도,  $e_\psi$ 는 경로 중심선에 대한 방향 오차,  $e_y$ 는 경로 중심선에 대한 횡 방향 오차다.  $u_{lat} = \delta_f$ 은 시스템의 입력에 해당하며, 목표 조향각이다.  $\rho$ 는 도로의 곡률을 뜻하며,  $C_f, C_r$ 은 각각 전, 후륜의 코너링 강성이다.  $l_f, l_r$ 은 각각 차량의 무게중심으로부터 전, 후륜 축까지의 거리를 뜻한다.  $I_z$ 는 차량 중심에 대한 요 회전 관성이다.

### 3. 시뮬레이션 환경 구성

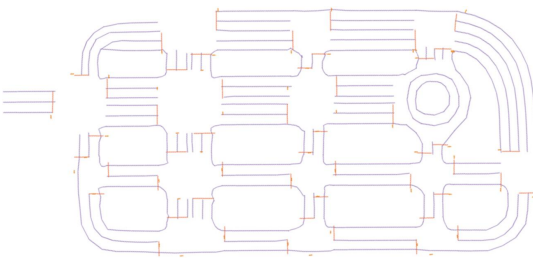
#### 3.1. 고정밀 지도 모델링

시뮬레이션 차량에 적용된 자율주행 알고리즘은 고정밀 지도(HD map)를 기반으로 동작한다. 따라서 LGSVL에서 제공하는 3D 그래픽 맵을 2D 고정밀 지도로 변환하였다. 이를 위해 두가지 프로그램을 사용하였다. 첫째는 Unity로서 3D 및 2D 비디오 게임의 개발 환경을 제공하는 게임 엔진이다. 둘째는 지리 정보 체계 응용 프로그램의 일종인 QGIS이며, 이는 데이터 시각화, 편집, 분석을 제공하는 크로스 플랫폼으로서 오픈 소스이다.

시뮬레이션 환경을 구축하기 위한 예제로 LGSVL에서



(a)



(b)

Fig. 4 (a) 3D shalun map (b) 2D high definition map of shalun

제공하는 Shalun맵을 채택하였다. 맵의 구성은 Fig. 4(a)의 모습과 같으며 원형교차로를 포함한 다양한 형태의 평면 교차로가 구현되어 있어 다양한 시나리오에 대한 시뮬레이션이 가능하다.

Unity는 ZXY 왼손 좌표계를 사용하므로 3D 지도의 좌표 정보를 XYZ 오른손 좌표계로 전환하여 2D 지도로 추출하였다. 추출한 2D 지도를 QGIS에서 편집하여 차선, 정지선에 대한 고정밀 지도를 각각 제작하였다. 최종적으로 제작된 2D 고정밀 지도는 Fig. 4(b)와 같다.

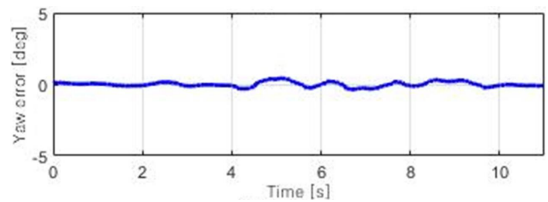
#### 3.2. 성능 검증 시나리오

차량 주행은 차선유지와 차선변경으로 이뤄지므로 자율주행 알고리즘의 성능을 평가하기 위해 차선변경과 차선유지를 포함한 세가지 시나리오를 개발하였다. 첫째는 직선 차로 구간에서의 차선유지, 둘째는 직선 차로 구간에서의 차선변경, 셋째는 직선 및 곡선 차로 구간을 포함한 코스주행 시나리오다. 3번째 시나리오에 대해서는 실차 실험과 데이터를 비교하였으며 개발한 시뮬레이션 환경이 실제 환경을 성공적으로 모방하는지에 대해 분석하였다. 실차 실험에서 사용된 차량은 VLP-16 라이더를 장착하였다.

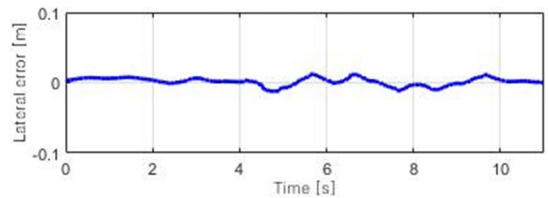
### 4. 시뮬레이션 결과

#### 4.1. 차선 유지 시나리오

차선유지 시나리오는 목표 속도 30kph로 진행되었으며 Fig. 5(a)~(b)는 차량 상태변수에 대한 정보이다. 차량은



(a) Yaw error



(b) Lateral error

Fig. 5 Simulation result 1: lane keeping

성공적으로 등속도를 유지하였으며 횡방향 거동에 대한 오차를 나타내는  $e_\psi, e_y$  또한 작은 값을 유지하였다.

#### 4.2. 차선 변경 시나리오

차선 유지 시나리오는 목표 속도 30kph로 진행되었으며 Fig. 6(a)~(c)는 차량 상태변수에 대한 정보이다. 차량의 목표 속도는 경로의 곡률에 의해 결정되는데, Fig. 6(a)에 따르면 차선을 변경하는 과정에서 초기 속도 30kph보다 작은 속도로 주행하였음을 확인할 수 있다. Fig. 6(c)에서 경로 중심에 대한 횡방향 오차는 5cm 내외로 나타났으며 충분한 성능을 보여주었다.

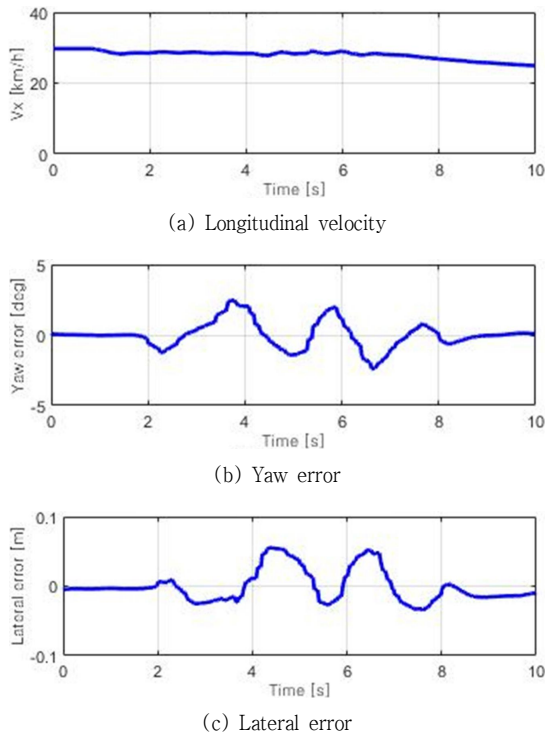


Fig. 6 Simulation result 2: lane change

#### 4.3. 코스 주행 시나리오

코스 주행 시나리오는 목표 속도 35kph로 진행되었으며 Fig. 7(a)~(c)는 차량 상태변수에 대한 정보이다. Fig. 7(a)에서 볼 수 있듯이 자율주행 알고리즘은 곡선 주로에서 경로의 곡률에 의해 차량의 거동계획을 결정하고, 차량의 속도를 10kph 부근으로 제한하였다. 해당 시나리오에

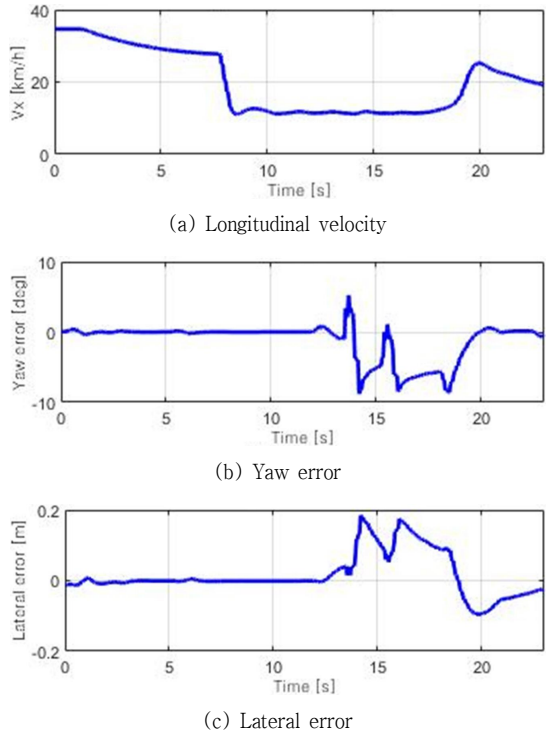


Fig. 7 Simulation result 3: Course driving

포함된 곡선 경로는 반경 12m이며 Fig. 7(c)에 따르면 이 구간에서  $e_y$ 의 최대값은 0.2m로 나타났다. Fig. 7(b)에서 볼 수 있듯이  $e_\psi$ 의 최대값은 10 deg로 나타났다.

시뮬레이션이 실차 환경을 성공적으로 모방하는지에 대해 분석하기 위해 해당 시나리오에 대한 실차 실험을 병행하였다. 실차 실험 또한 목표 속도 35kph로 진행되었으며 곡선 주로의 곡률 또한 12m로 동일하게 설정하였다. Fig. 8(a)~(c)는 차량 상태변수에 대한 정보이다. Table 1은 각각의 시뮬레이션과 실차 실험 데이터를 정리한 결과이다. 코스 주행 시나리오에 대한 시뮬레이션 결과가 차량 실험 결과와 유사함을 확인할 수 있다.

Table 1 Vehicle lateral state variable of each test

	Test 1	Test 2	Test 3	
			Simulation	Real vehicle
$e_y, \max (m)$	0.012	0.055	0.185	0.223
$e_y, \min (m)$	-0.012	-0.035	-0.095	-0.078
$e_\psi, \max (deg)$	0.385	2.451	5.235	4.973
$e_\psi, \min (deg)$	0.281	-2.371	-8.807	-8.381

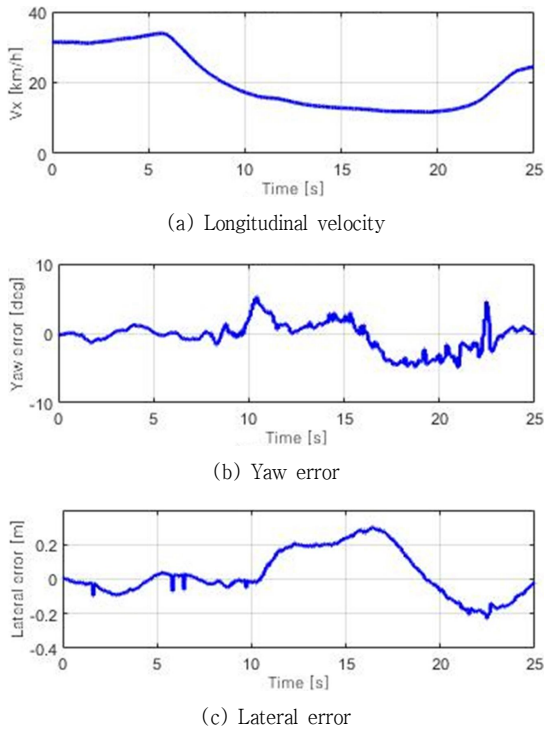


Fig. 8 Vehicle test result: Course driving

## 5. 결론

본 논문은 ROS 기반의 자율주행 알고리즘 성능 검증을 위한 시뮬레이션 환경을 개발하였다. 이를 위해 LGSVL을 채택하였으며, 이는 메시지 통신을 기반으로 센서 데이터를 주고받는 ROS 기반 자율주행 알고리즘을 분석하기 위함이다. 시뮬레이션 차량 모델에는 라이다, GPS, IMU 센서가 탑재되었으며 이를 통해 차량 알고리즘은 가상환경속에서의 주행환경과 차량의 운동 상태를 파악한다. 인지 정보를 바탕으로 차량 알고리즘은 차량의 거동 계획을 수립하며 이를 위한 제어 입력이 계산된다. 제어 입력은 시뮬레이터 차량에 인가되며, 시뮬레이터 차량은 실시간으로 변화하는 주변 환경에 대한 센서 정보를 차량 알고리즘에 전송한다. 일련의 순환 과정을 반복하며 시뮬레이터와 차량 알고리즘은 closed-loop 시스템을 구성하게 된다. 이는 3가지 시나리오에 대해서 그 성능이 검증되었으며, 실제 차량 실험 데이터와의 비교를 통해 시뮬레이션 환경

이 실제 차량 환경을 성공적으로 모델링한다는 것을 확인하였다.

본 연구는 향후 다양한 시나리오에 대한 차량 시뮬레이션을 가능하게 할 것으로 기대된다. 실제 상황에서 구현하기 힘든 위험상황을 구현하거나 기상조건을 자유롭게 설정하며 시간과 공간의 제약 없이 자율주행 알고리즘의 성능을 분석할 수 있고 이는 자율주행 차량의 인지 성능과 이에 따른 판단 및 제어 성능을 향상시키는 데에 기여할 것이다.

## 후 기

본 연구는 국토교통부/국토교통과학기술진흥원의 지원으로 수행되었음(과제번호 21AMDP-C162182-01).

## 참고문헌

- (1) A. Hussein, F. Garcia and C. Olaverri-Monreal, 2018, "ROS and Unity Based Framework for Intelligent Vehicles Control and Simulation", 2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES), Madrid, Spain, pp. 1~6.
- (2) G. Rong et al., 2020, "LGSVL Simulator: A High Fidelity Simulator for Autonomous Driving", 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), Rhodes, Greece, pp. 1~6.
- (3) H. Lee, J. Yoon, Y. Jeong and K. Yi, "Moving Object Detection and Tracking Based on Interaction of Static Obstacle Map and Geometric Model-Free Approach for Urban Autonomous Driving", in IEEE Transactions on Intelligent Transportation Systems.
- (4) H. Chae, Y. Jeong, S. Kim, H. Lee, J. Park and K. Yi, 2018, "Design and Vehicle Implementation of Autonomous Lane Change Algorithm based on Probabilistic Prediction", 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, pp. 2845~2852.