CARLA 시뮬레이터 및 CAN 버스 연동 기반 자율주행 하드웨어 시뮬레이션 환경 구축

성영주, 임 혁, 윤승현
한국에너지공과대학교 (KENTECH)
{yj2023, hlim, syoon}@kentech.ac.kr,

Development of a CARLA Simulator and CAN Bus Integrated Hardware Simulation Environment for Autonomous Driving

Yeong Ju Seong, Hyuk Lim, Seunghyun Yoon
Korea Institute of Energy Technology (KENTECH)

요 약

자율주행 기술의 발전에 따라 알고리즘의 안전성과 신뢰성을 검증할 수 있는 현실성 있는 실험 환경의 필요성이 커지고 있다. 그러나 실제 차량을 이용한 검증은 비용과 위험 부담이 크며, 기존의 시뮬레이션 기반 환경은 물리적 제어 장치와의 상호작용이 제한적이라는 한계가 있다. 이에 본 논문은 CARLA 시뮬레이터와 CAN 버스 통신을 연동한 하드웨어 기반 자율주행 실험 환경을 제안한다. 조향 장치의 입력 신호는 CAN 버스를 통해 시뮬레이터 내부의 가상 차량으로 전달되며, 차량은 사용자의 조작에 실시간으로 반응한다. 또한 모든 제어 신호와 통신 데이터는 Python 인터페이스를 통해 모니터링 및 분석이 가능하도록 구성하였다. 제안된 환경은 자율주행 알고리즘의 성능 검증뿐 아니라, CAN 통신 보안 및 공격 시나리오 검증을 위한 실험 플랫폼으로 활용될 수 있다.

Ⅰ. 서 론

인공지능 기술이 발전하면서 산업 전반에 도입되고 있다. 자동차 분야에서도 센서 기반의 운전자 보조 시스템 기술들이 개발되고 적용되었다. 운전자 보조 기술들의 발전을 토대로, 운전자의 개입을 최소화하는 자율주행 시스템에 대한 연구도 활발히 진행되고 있다. 따라서 자동차용 인공지능 테스트를 위한 실험 환경 구축은 나날이 중요해지고 있다. 하지만 실제 도로에서의 자율주행 기능 테스트는 시간, 비용, 위험성 문제를 동반하였고, 해당 문제들을 해결하기 위한 대안으로 시뮬레이션 기반 테스트가 주목받게 되었다 [1]. 본 논문에서는 이러한 시뮬레이션 기반 테스트에 대한 필요를 충족하기 위한 방안으로 CARLA 시뮬레이터 기반의 가상 실험 환경 구축 방식을 제안한다.

CARLA는 자율주행 연구를 위한 오픈 소스 시뮬레이터로, 실제 도로 상황과 유사한 도시 환경, 건물, 차량 등의 디지털 에셋을 제공하고, 다양한 시나리오 환경과 센서 구성을 지원하면서 자율주행 알고리즘의 테스트 및 평가에 적극적으로 활용되고 있다. 본 논문에서는 기존 CARLA 시뮬레이터에 실제 차량에 사용되는 CAN 버스 프로토콜 [2]을 구축하고 물리적인 조향 장치를 연동하였다. CAN(Controller Area Network)은 차량 내부의 전자 제어 장치들 간의 통신을 위해 사용되는 표준 통신 규격이다. 호스트 컴퓨터의 개입 없이 동작하며 데이터 프레임을 버스에 전송하는 메시지 기반 통신이다. 이를 구현함으써 실제 주행과 유사한 환경을 구성할 수 있다. 해당 구현은 소프트웨어를 기반으로 자율주행 알고리즘의 검증에 집중하는 일반적인 실험 환경과 달리, 실제 하드웨어 요소와 연계되는 실증적인 환경을 구현했다는 차별점이 있다.

또한 자율주행 차량은 탐지, 통신, 제어 등의 여러 요소가 연동되는 복합적 시스템으로, 구성 요소의 보안 취약성은 시스템 안정성과 탑승자의 안전에 직접적인 해를 끼칠 수 있다 [3]. 본 논문에서 제안한 환경은 CAN 버스에 대한 보안 공격 시나리오를 실제와 유사한 조건에서 검증할 수 있는 테스트베드를 제공하며, 자율주행 차량 보안 연구를 위한 기반으로도 활용될 수 있다.

Ⅱ. 본론

2.1 시스템 구성 개요

본 논문에서 제안하는 시뮬레이션 환경은 Ubuntu 20.04 기반 CARLA 시뮬레이터, Python 기반 통신 인터페이스, CAN 버스 모듈, 물리적 조향 장치로 구성된다.

사용자는 물리적 조향 장치를 사용해 시뮬레이션 내 객체 차량을 직접 조작할 수 있고, 시뮬레이터는 차량의 물리적인 데이터를 제공한다. 해당 데이터는 통신 인터페이스에서 CAN 메시지로 변환되고 CAN 버스 모듈로 전달된다. 변환된 메시지들은 CAN 버스를 통해 객체 차량 내부의 계통간에서 송수신된다. 해당 환경은 외부 공격자가 차량의 CAN 버스에 접근하여 물리적 데이터를 조작하거나, 차량 내부 계통의 동작을 속이는 공격 시나리오의 구현 기반을 마련한다.

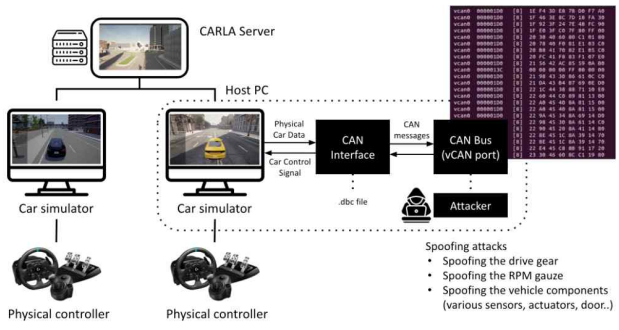

그림 1. 실험 환경 구성도

2.2 CAN 통신 구성

데이터 프레임은 각 제어 장치에 할당된 고유 CAN ID, 8 바이트 크기의 데이터 필드에 더하여 오류 검출 및 전송 제어를 위한 추가 필드들로 구성된다.

시뮬레이터에서 생성된 제어 명령을 CAN 메시지 프레임 형식으로 변환하여 실제 차량의 ECU 간 통신 형태를 모방하였다. 이 과정에서 cantools 라이브러리를 적용해서 사전에 정리된 DBC 파일을 기준으로 CAN 메시지를 해석하고, 제어 명령은 해석 양식에 맞춰 인코딩, 디코딩되며 실제 차량 내 통신과 유사한 구조를 적용하였다. 추가적으로 CAN 버스에 대한 공격 시나리오 상황에서 통신 내용의 변화를 관찰하기 위해 제어 명령의 CAN 메시지들을 시각화한 모니터링 환경을 구현하였다.

2.3 조향 장치 구성 및 연동

조향 장치에서 발생한 입력 신호는 실시간으로 수집된 후 시뮬레이터에 전달된다. 본 연구에서는 Logitech G923 핸들, 페달을 조향 장치로 사용하였으며, 해당 장치를 통해 입력된 사용자 조작에 반응하는 차량 동작을 시뮬레이션 환경 내에 구현할 수 있었다. 이 과정에서 실제 핸들과 페달 입력의 범위를 시뮬레이션 내부 객체 차량의 제어 범위에 맞춰 변환하는 추가적인 연산 모듈이 추가되었다.

그림 2. 조향 장치 환경

Ⅲ. 구현

구성된 시스템의 정상적인 동작 여부를 확인하기 위해 실제 주행과 유사한 시나리오를 구성하여 구현 환경을 시연하였다. 사용자가 조작하는 조향 장치에 입력이 발생하면 객체 차량이 제어 명령을 받아 사용자의 입력에 실시간으로 동기화된 동작을 수행하였다.

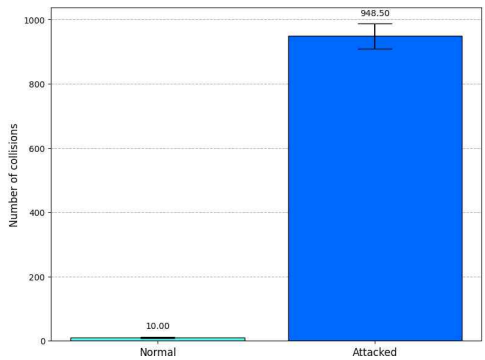

그림 3. 공격 유무에 따른 충돌 횟수

3.1 공격 시나리오 구현

시뮬레이션 차량에 기존 입력과 유사한 신호를 입력하는 스푸핑 공격을 가했을 때, 객체 차량은 공격을 통해 입력된 신호를 정상 입력으로 판단하고 비정상적인 움직임을 보인다.

스푸핑 공격이 충돌 횟수에 가하는 영향을 측정하기 위해 실험를 설계하였다. 시뮬레이터의 자율 주행 기능을 활용해 정상적인 자율주행 상황과 스푸핑 공격을 받는 자율주행 상황을 구성하였다. 각각의 상황에서 동일한 시나리오를 40초간 3회씩 반복 실행하고, 시나리오 수행 중 매 타임스탬프마다 충돌 여부를 확인하여 기록하였다. 해당 실험에서 스푸핑 공격은 받는 객체 차량의 가속 페달 신호를 최댓값으로 만들도록 설정하였다.

그림 4는 각 상황에서 충돌이 발생한 횟수를 비교하였으며, 스푸핑 공격을 받은 객체 차량이 일반적인 주행 상황에서보다 더 많은 충돌 횟수를 기록하였다.

해당 구현을 통해 CAN 버스를 활용한 데이터 송수신이 정상적으로 작동함이 확인되었다. 조향 장치를 통한 주행 명령 입력뿐만 아니라 임의로 생성된 공격 신호들까지 제어 계통에 정확히 전달할 수 있음을 알 수 있다.

Ⅳ. 결론

본 논문은 CARLA 시뮬레이터 내부에 파이썬 라이브러리를 통한 CAN 버스 프로토콜을 구현하고 물리적 조향 장치를 연동한 자율주행 알고리즘 검증을 위한 테스트베드를 구축하였다.

기존의 소프트웨어 기반 시뮬레이션 환경에 물리적인 제어 장치를 연동하면서 실제 차량과 유사한 환경을 구축하였으며, 기존 환경에 CAN 통신 기반으로 제어계 신호를 송수신하는 구조를 통해 차량 보안 연구를 위한 실험 조건을 제공하여 향후 연구를 위한 기틀을 마련하였다.

향후에 조향 장치와 시뮬레이션 환경 간의 상호 연동성을 보완하여, 자율 주행 기능이 활성화된 상황에서 조향 장치가 가상 환경의 변화에 반응하여 차량의 움직임에 완전히 동기화된 동작을 구현하도록 개선하는 과정이 필요하다. 또한, 본 연구에서 구현한 CAN 기반 프로토콜의 신뢰성을 검증하고, 직접 구축한 탐지 알고리즘을 시뮬레이션에 적용하는 테스트를 통해 자율주행 차량의 보안 안정성을 향상시키는 방안을 모색하여 시스템의 안정성 보강을 위한 지속적인 연구를 수행할 것이다.

ACKNOWLEDGMENT

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2021-II210379, Privacy risk analysis and response technology development for AI systems), and by the KENTECH Residential College (RC) Co-Curricular Program (CCP) (자율주행 시뮬레이션 환경 구축).

참 고 문 헌

[1] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “CARLA: An Open Urban Driving Simulator,“ Proceedings of the 1st Annual Conference on Robot Learning, pp. 1-16, 2017.

[2] A. A. Salunkhe, P. P. Kamble, and R. Jadhav, “Design and implementation of CAN bus protocol for monitoring vehicle parameters,“ 2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), pp. 301-304, 2016.

[3] K. Kim, J. S. Kim, S. Jeong, J.-H. Park, and H. K. Kim, “Cybersecurity for autonomous vehicles: Review of attacks and defense,“ Computers & Security, p. 102150, 2021.