

자율주행 시뮬레이션 환경을 위한 차량 구동 및 제동 제어기 개발

곽지섭* · 이경수**,[†]

Development of Throttle and Brake Controller for Autonomous Vehicle Simulation Environment

Jisub Kwak*, Kyongsu Yi**,[†]

Key Words: Autonomous driving(자율주행), Longitudinal Control(종방향 제어), Vehicle simulation(차량 시뮬레이션), Low-level Controller(하위 제어기)

ABSTRACT

This paper presents a development of throttle and brake controller for autonomous vehicle simulation environment. Most of 3D simulator control autonomous vehicle by throttle and brake command. Therefore additional longitudinal controller is required to calculate pedal input from desired acceleration. The controller consists of two parts, feedback controller and feedforward controller. The feedback controller is designed to compensate error between the actual acceleration and desired acceleration calculated from autonomous driving algorithm. The feedforward controller is designed for fast response and the output is determined by the actual vehicle speed and desired acceleration. To verify the performance of the controller, simulations were conducted for various scenarios, and it was confirmed that the controller can successfully follow the target acceleration.

1. 서론

최근 자동차 산업의 핵심은 자율주행 연구이며 이는 차량의 하드웨어 구조를 넘어 차량의 소프트웨어에 대한 고도화된 연구를 통해 알 수 있다. 이에 따라 세계 곳곳에서는 자율주행 연구가 활발히 이뤄지고 있다. 현대자동차, 구글, 테슬라 등 세계 여러 기업들이 자율주행 자동차의 상용화를 위해 인력과 자본을 투자하고 있으며, 고도의 주행 보조 시스템을 탑재한 차량이 상용화에 성공되어 시중에 널리 유통되고 있다.

또한 운전자의 개입을 필요로 하지 않는 5단계 완전자율주행이라는 목표를 달성하기 위해 관련 연구가 활발히 진행되고 있으며, 알고리즘의 성능을 검증하고 분석하기 위해서는 실제 차량에서의 실험이 필수적으로 요구되고 있다. 그러나 카메라, 라이다, GPS 등 자율주행에서 사용되는 대부분의 센서들이 높은 가격을 형성하고 있으며, 이들을 모두 탑재한 자율주행차를 제작하기 위해 상당한 비용이 필요하다. 이를 제외하고도 실제 차량 실험을 위한 도로 환경과 인프라를 조성하고 실험을 진행하는 것도 많은 시간과 인력을 필요로 한다. 설령 많은 시간과 인력을 투자한다 하더라도 날씨와 기후와 같은 초자연적인 측면에 대한 변인통제가 불가능하기 때문에 실제 환경에서 알고리즘의 성능을 평가할 수 있는 정밀한 실험을 진행하는 것에는 많은 제약이 따른다. 따라서 실제 자동차 실험 환

* 서울대학교 기계공학부, 석박통합과정

** 서울대학교 기계공학부, 교수

[†]교신저자: kyi@snu.ac.kr

E-mail: kjs675252@snu.ac.kr

경을 모방할 수 있는 시뮬레이션 환경이 만들어지면 실제 자동차 테스트를 가상 환경으로 대체할 수 있다. 단순히 실제 환경을 대체한다는 장점 이외에도 시뮬레이션 환경은 개발자가 다양한 환경에 대한 실험을 자유롭게 진행할 수 있다는 장점이 있다. 악천후 상황, 도로 환경 모델링을 비롯한 다양하고 자유로운 변인통제가 가능하기 때문에 시뮬레이션 환경의 도입은 자율주행 분야에서 필수적이다. 또한 새로 개발된 알고리즘의 안전성이 아직 검증되지 않았기 때문에 새로운 알고리즘을 바로 실제 차량에 적용하여 실험을 진행할 경우 실험자와 운전자의 안전을 위협할 수 있다. 따라서 안전성 측면에서도 시뮬레이션 환경을 우선적으로 채택하는 것이 좋다.

이를 위해서는 시뮬레이션 환경이 실제 차량과 유사하다는 것이 보장되어야 하며, 본 논문에서는 이를 위해 시뮬레이션 차량에 필요한 제어 인터페이스를 제안하였고, 이를 바탕으로 자율주행 알고리즘을 평가하고 검증할 수 있는 시뮬레이션 환경을 구축하였다.

본 논문에서 검증한 알고리즘에 활용된 로봇 운영 체제(ROS)는 로봇 프로그래밍에 널리 사용되는 미들웨어 플랫폼이며 오픈소스로 제공된다.⁽¹⁾ ROS는 오픈소스로 제공되기 때문에 범용성이 높으며, 데이터를 가시화하기에 편리하다. 널리 사용되는 차량 시뮬레이터로서는 Carsim, Carmaker 등이 있지만 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.⁽²⁾

하지만 LGSVL은 차량 동역학 특성을 정교하게 반영하지 못하여 차량을 제어할 수 있는 제어가 구현되어있지 않다. 따라서 이를 해결하기 위해 추가적인 제어기 제작이 필요했으며, 본 연구에서 차량 하위 제어기를 개발하였다. 추가적으로 개발한 하위 제어기의 성능을 검증하기 위해 3가지 시나리오를 설정하여 시뮬레이션을 진행하고 실제 차량의 데이터와 유사한지 분석하였다.

2. 자율주행 차량

본 연구에서는 LGSVL에서 기본예제로 제공하는 'HYUNDAE NEXO' 차량을 시뮬레이션 차량 모델로 선정하였으며 Fig. 1에 나타나 있다. 라이다 센서를 이용하여 주변 도로 상황을 인지하며, 차체 상단에 VLP-16 모델을 탑재했다. 라이다 센서의 경우 매개 변수 설정을 통해 고성능 다채널

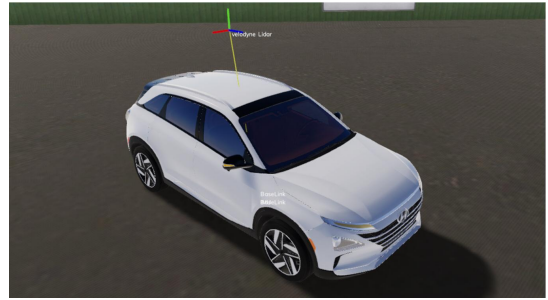


Fig. 1 Sensor configuration of simulation vehicle

라이다를 모델링 할 수 있지만, 실차 환경과 유사한 시뮬레이션 환경을 구성하기 위해 실차에서 사용된 동일한 VLP-16 모델을 채택하였다.

추가로 GPS와 IMU 센서를 장착하여 차량의 현재 위치, 속도, 요각을 측정하도록 하였다. 이는 차량 시스템이 주변 환경을 인지하고 향후 거동을 계획할 때 차량의 상태에 대한 정보를 얻기 위해 사용된다.

2.1. 차량 알고리즘 구성

본 논문에서 활용된 자율주행 알고리즘의 구조는 Fig. 2와 같다. 해당 알고리즘은 두 개의 모듈로 구성되어 있는데, 각각은 인지 모듈과 판단 모듈이다. 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 데이터를 기반으로 차량의 운동상태를 추정한다. 자차량의 속도, 요각 정보를 바탕으로 정지 물체와 운동 물체를 개별적으로 추정하며 Extended Kalman filter와 Particle Filter를 이용하여 이를 업데이트 하여 자차량의 운동상태와 주변 차량 및 물체들을 인지한다.⁽³⁾

다음으로 판단 알고리즘은 Chae(2018)에 의해 제안된 Probabilistic prediction based lane change 알고리즘으로서 인지 모듈에 의한 인지 출력 데이터와 자차량의 상태 정보를 이용해 차량의 주변 환경을 파악하고 차량의 거동 계획을 확률적 예측을 통해 결정한다.⁽⁴⁾

알고리즘의 출력은 목표 가속도와 목표 회전각이며 시뮬레이션의 차량제어 입력은 페달입력과 회전각이다. 따라서 알고리즘의 횡방향 출력은 그대로 시뮬레이션의 제어입력으로 사용 가능하지만 종방향 출력인 목표 가속도의 경우 그대로 시뮬레이션에 적용할 수 없으므로 추가적인 제어기가 필요하다. 이에 대해서는 3장에서 자세히 설명할 예정이다.

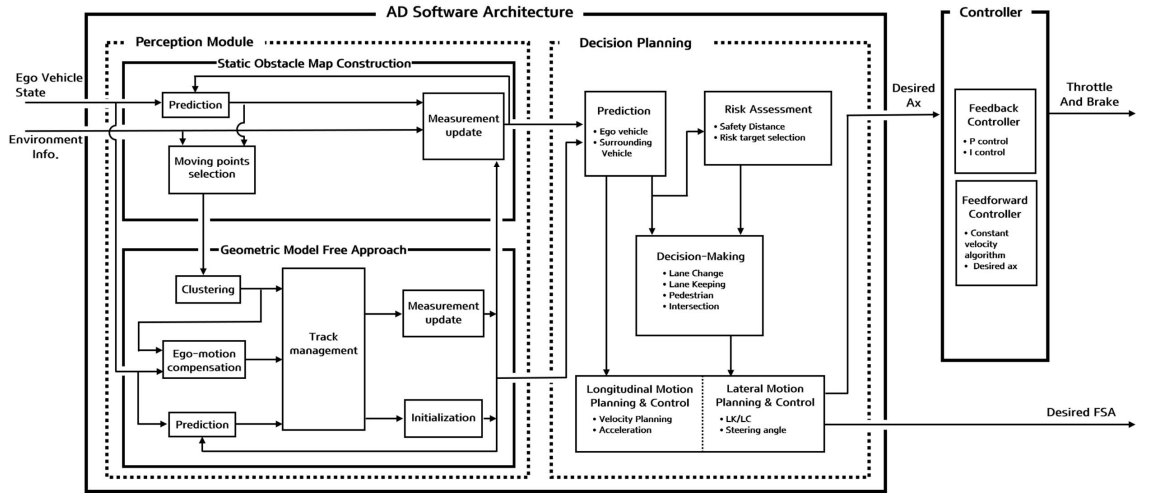


Fig. 2 Autonomous driving algorithm architecture

2.2. 차량 동역학 모델

차량 시스템은 탑승자의 안전에 직접적인 영향을 끼치므로 자율주행 알고리즘은 시간에 따라 변하는 주행 환경에 즉각적으로 반응해야 하며 실시간으로 동작해야 한다. 즉 자율주행 알고리즘의 경우 계산 복잡도를 줄이는 것이 매우 중요하다. 따라서 차량의 동역학 모델은 종 방향 동역학 모델과 횡 방향 동역학 모델로 분리되어 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.

$$\dot{\bar{x}}_{lon} = A_{lon}\bar{x}_{lon} + B_{lon}u_{lon}$$

$$s.t. A_{lon} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -1/\tau_a \end{bmatrix}, B_{lon} = \begin{bmatrix} 0 \\ 0 \\ 1/\tau_a \end{bmatrix} \quad (1)$$

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

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

이 때 $\bar{x}_{lat} = [\beta \ \gamma \ e_\psi \ e_y]^T$ 는 상태변수이며 β 는

횡 방향 슬립 각, γ 는 무게중심에 대한 회전속도, e_ψ 는 경로 중심선에 대한 방향 오차, e_y 는 경로 중심선에 대한 횡 방향 오차다. $u_{lat} = \delta_f$ 는 시스템의 입력에 해당하며, 목표 조향각이다. ρ 는 도로의 곡률을 뜻하며, C_f, C_r 은 각각 전, 후륜의 코너링 강성이다. l_f, l_r 은 각 차량의 무게

$$\dot{\bar{x}}_{lat} = A_{lat}\bar{x}_{lat} + B_{lat}u_{lat} + F_{lat}\rho$$

$$s.t. A_{lat} = \begin{bmatrix} \frac{2C_f + 2C_r}{mv} & \frac{2C_f l_f + 2C_r l_r}{mv} & 0 & 0 \\ -\frac{2C_f l_f - 2C_r l_r}{I_z} & -\frac{2C_f l_f^2 + 2C_r l_r^2}{I_z v} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ v & 0 & v & 0 \end{bmatrix},$$

$$B_{lat} = \begin{bmatrix} \frac{2C_f}{I_z} \\ \frac{mv_x}{I_z} \\ 0 \\ 0 \end{bmatrix}, F_{lat} = \begin{bmatrix} 0 \\ 0 \\ -v_x \\ 0 \end{bmatrix} \quad (2)$$

중심으로부터 전, 후륜 축까지의 거리를 뜻한다. I_z 는 차량 중심에 대한 요 회전 관성이다.

3. 구동 및 제동 제어기

본 연구에서 검증하고자 하는 자율주행 알고리즘은 차량의 목표가속도를 계산한다. 하지만 시뮬레이터 차량은 목표 가속도가 아닌 페달 입력을 이용해 차량을 제어한다. 따라서 자율주행 알고리즘을 시뮬레이터에서 구현하기 위해 추가적인 종방향 하위 제어기가 필요하였고, 이는 차량의 목표 가속도를 차량의 페달 입력으로 변환해주는 역할을 한다. 본 연구에서 제안하는 종방향 하위 제어기의 전체 구조는 Fig. 3과 같다. 제어기는 크게 두 개의 제어기로 구성되며 이는 각각 피드백 제어기와 피드포워드 제어기이다.

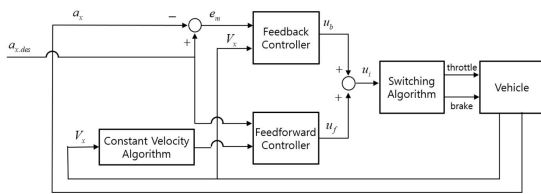


Fig. 3 Throttle and brake controller architecture

3.1. 피드백 제어기 구성

본 연구에서 제안하는 피드백 제어기는 비례제어기와 적분제어기로 이루어져 있다. 비례제어기는 제어기의 목표값과 현재 측정값의 차이에 따라 비례하는 출력을 만들어주어 오차를 감소시키는 역할을 한다. 제어 목표값은 자율주행 알고리즘의 출력값인 목표 가속도로 설정하였으며 현재 측정값은 차량의 실제 가속도로 설정하였다. 비례 제어기의 비례계수를 작게 설정할 경우 응답성을 보장할 수 없으며 크게 설정할 경우 과도한 오버슈트를 발생시켜 차량 시스템의 고장을 야기할 수 있다. 따라서 적절한 비례계수 설정을 위한 과정이 필수적이며, 이는 경험적으로 채택되었다.

적분제어기는 제어기의 목표값과 현재 측정값의 차이를 적분한 값을 오차로 정의하고 이에 비례하는 출력을 계산해주는 제어기이다. 적분제어기의 적분계수 또한 적절한 값으로 설정 해야하며, 경험적으로 채택되었다. 비례 제어기와 적분제어기에서 각각 계산된 출력값을 합하여 최종적인 피드백 제어기의 출력을 결정한다.

3.2. 피드포워드 제어기 구성

피드포워드 제어기는 두 종류의 알고리즘으로 이루어져 있으며 시스템이 추종해야하는 목표출력을 효과적으

Table 1 Steady-state velocity result

Acceleration				
Throttle	0.025	0.05	0.075	0.1
Vx[kph]	21.8	45.3	63.7	74.8
Deceleration				
Throttle	0.075	0.05	0.025	0
Vx[kph]	68.5	50.7	28.9	0.7

로 따라가기 위해 고안된 제어기이다. 첫 번째 알고리즘은 현재 차량 속도에 의해 제어입력이 결정되며 이를 위해 차량 반응특성을 관측하였다. 실험 방법은 Throttle 입력에 대한 정상상태 속도를 측정하였다. Table 1은 실험 수치와 그에 따른 결과를 정리한 것이다.

이와 같이 Throttle 입력을 단계적으로 가했을 때 차량은 초기에는 가속도 운동을 하게 되지만, 시간이 지남에 따라 구름저항, 공기저항, 타이어의 마찰력을 비롯한 저항에 의해 점차 가속도가 사라지게 되며 등속도 운동에 접어들게 된다. 이는 시뮬레이터 내부에 적용되어 있는 동역학 모델의 특성을 반영한 결과이며, 가속상황과 감속상황에 대해 각각 실시하였다. 가속상황과 감속상황에서의 데이터의 중간값을 이용하여 구간별 속도에 따른 Throttle 값을 선형화 하였다. 이와 같이 사전에 정의된 테이블을 바탕으로 첫번째 알고리즘에서는 차량의 실시간 속도를 입력으로 받으며 이에 대응하는 Throttle 출력을 계산한다.

이어서 피드포워드의 두번째 알고리즘은 목표 가속도를 입력으로 받아 이에 비례하는 출력을 계산한다. 이는 차량의 현재 상태를 고려하지 않으며 목표 가속도를 추종하는데 있어서 반응성을 높이기 위해 고안된 알고리즘이다. 피드포워드의 두 알고리즘에서 계산된 출력을 이용해 하나의 제어 입력 u_f 가 계산되며, 피드백 제어기에서 계산된 u_b 가 더해진 u_t 가 최종 제어 입력이 된다. 스위칭 알고리즘에서는 u_t 의 값에 따라 Throttle 입력과 Brake 입력값을 계산하며, 이 값이 최종적으로 시뮬레이터 차량을 제어하는 제어 입력으로 사용된다.

3.3. 성능 검증 시나리오

본 연구에서 개발한 제어기의 성능을 평가하기 위해 차선변경과 차선유지를 포함한 세가지 시나리오를 개발하였다. 첫째는 직선 차로 구간에서의 차선변경, 둘째는 직선구간을 포함한 외선코스주행, 셋째는 도심지역에서 빈번하게 발생하는 교차로를 포함한 주행 시나리오이다.

4. 시뮬레이션 결과

4.1. 차선 변경 시나리오

차선변경 시나리오는 목표 속도 30kph로 진행되었으며 Fig. 4(a)~(c)는 차량 상태변수에 대한 정보이다. 차량은 차선변경 구간에서 등속도를 유지하였으며 이는 설계된 종방향 제어기가 안정적으로 차량을 제어하고 있음을 나타낸다.

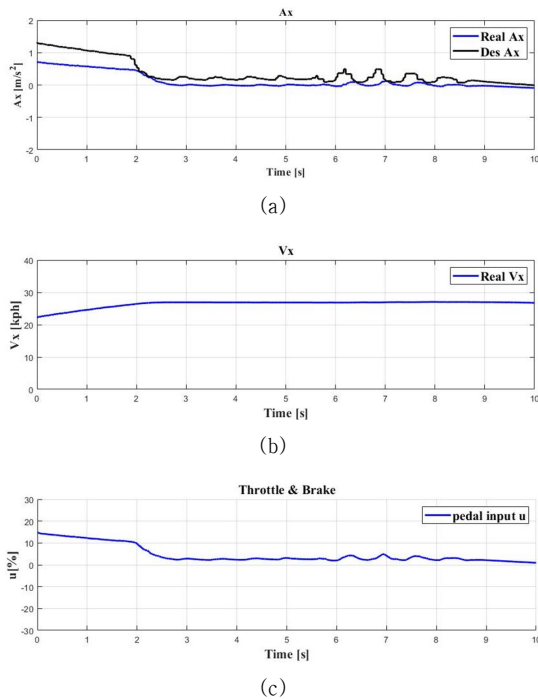


Fig. 4 Simulation result 1: lane change

4.2. 외선순환 시나리오

외선순환 시나리오는 목표 속도 20kph와 35kph로 진행되었으며 Fig. 5(a)~(c)는 목표 속도 20kph에서 차량 상태변수에 대한 정보이며, Fig. 6(a)~(c)는 목표 속도 35kph에서의 차량 상태변수에 대한 정보이다.

두 시나리오에 대해서 차량의 실제 가속도는 알고리즘에 의해 계산된 목표 가속도를 성공적으로 추종하였으며, 이는 설계된 구동 및 제동 제어기의 출력이 유효한 결과를 의미한다.

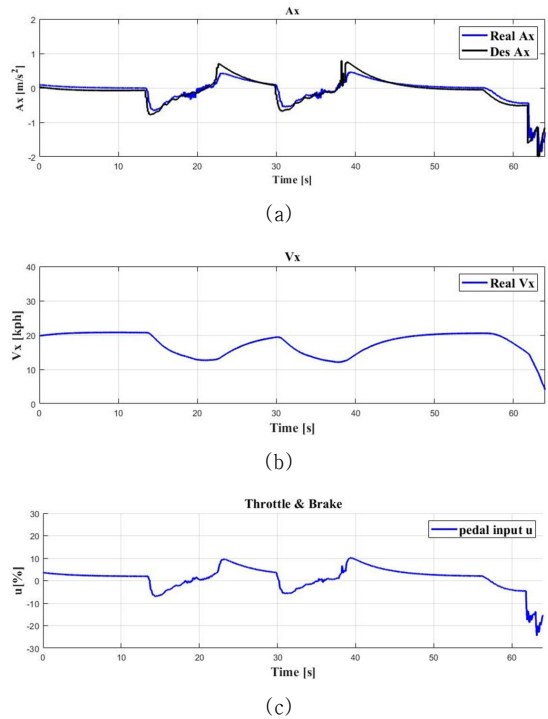


Fig. 5 Simulation result 2_1: outerloop (20kph)

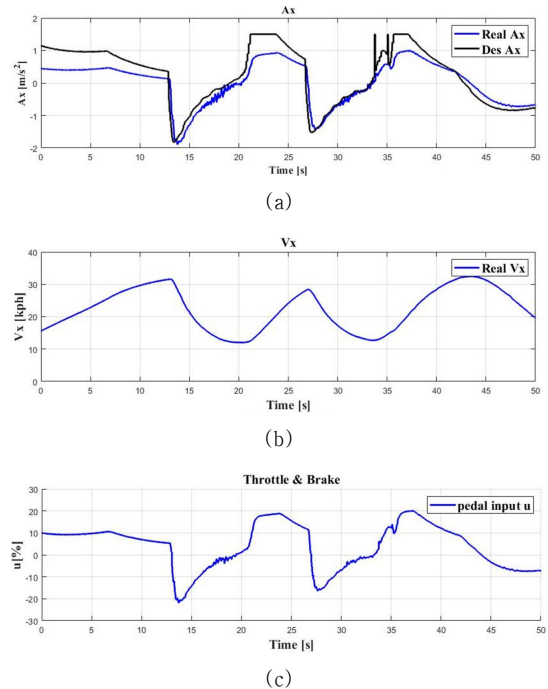


Fig. 6 Simulation result 3: outerloop (35kph)

4.3. 교차로 시나리오

교차로 시나리오는 목표 속도 35kph로 진행되었으며 Fig. 7(a)~(c)는 차량 상태변수에 대한 정보이다. 직선구간과 교차로가 반복됨에 따라 차량의 가속 및 감속 시나리오가 주기적으로 발생하였다. 교차로 구간에서 차량의 제동 및 구동 입력은 10% 이내로 인가되었으며 차량의 실제 가속도는 1m/s^2 이내의 값으로 측정되었다. 따라서 급격한 가속 및 제동은 발생하지 않았으며 차량이 목표 가속도를 성공적으로 추종하였음을 확인할 수 있다.

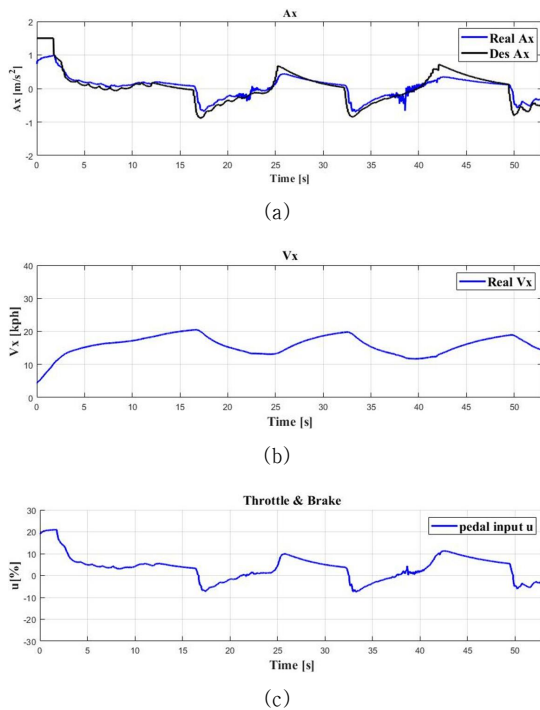


Fig. 7 Simulation result 3: intersection

5. 결론

본 연구에서는 차량 시뮬레이션에 적용할 중방향 구동 및 제동 제어기를 설계하였다. 제어기의 입력은 목표 가속도와 차량 센서로부터 측정된 실제 가속도와 실제 속도이다. 제어기의 출력은 Throttle & Brake 페달 입력이며, 이를 통해 시뮬레이터 차량을 제어하게 된다. 제어기는 두 개의 제어기로 구성되어 있으며, 각각은 피드백 제어기와 피드포워드 제어기이다.

피드백제어기는 비례제어와 적분제어로 이루어져 있으

며 목표 가속도와 실제 차량 가속도의 오차를 줄이기 위해 설계되었다. 피드포워드 제어기는 시스템이 추종해야하는 목표출력을 효과적으로 따라가기 위해 고안된 제어기이며, 현재 차량 속도에 따라 제어입력을 결정하기 위해 차량 반응특성을 관측하였다. 제어기의 성능을 검증하기 위해 3개의 시나리오에 대한 시뮬레이션을 진행했으며, 목표 가속도를 성공적으로 추종하는 제어가 보장됨을 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.