

대열주행을 위한 V2V 통신 기반 자율협력주행 알고리즘 개발

김 동 환¹⁾ · 하 윤 철¹⁾ · 김 세 환¹⁾ · 이 상 엽¹⁾ · 박 기 홍^{*2)} · 박 현 배³⁾ · 전 재 석³⁾

국민대학교 자동차공학전문대학원¹⁾ · 국민대학교 자동차IT융합학과²⁾ · 자동차융합기술원 친환경모빌리티연구센터³⁾

Development of Cooperative Autonomous Driving Algorithm Using V2V Communication for Convoy Driving

Donghwan Kim¹⁾ · Yunchul Ha¹⁾ · Sehwan Kim¹⁾ · Sangyeop Lee¹⁾ · Kihong Park^{*2)} · Hyunbae Park³⁾ · Jaeseok Jeon³⁾

¹⁾Graduate School of Automotive Engineering, Kookmin University, Seoul 02707, Korea

²⁾Department of Automobile and IT Convergence, Kookmin University, Seoul 02707, Korea

³⁾Green Mobility R&D Center, JIAT, 6, Dongjangan 2-gil, Gunsan-si, Jeonbuk 54158, Korea

(Received 7 December 2020 / Revised 21 December 2020 / Accepted 21 December 2020)

Abstract : It is possible to improve the performance of the existing autonomous driving functions relying solely on environmental sensors via V2V communication. In this paper, a cooperative autonomous driving algorithm was developed by using V2V communication to allow a safe and comfortable convoy driving with the preceding vehicles. The algorithm implements CACC when the convoy is maintained with the preceding vehicle. In addition, the algorithm has been designed to handle situations that can lead to dangerous conditions when handled only by a CACC logic, such as when the convoy changes its formation by a merging or leaving vehicle in front. For validation, simulation was carried out on different scenarios, and the results confirmed that the proposed algorithm can significantly improve string stability during convoy driving compared to ACC and CACC algorithms. It was also confirmed that the proposed algorithm can provide enhanced comfort by completing transient convoy maneuvers with minimum control effort.

Key words : Cooperative autonomous driving(자율협력주행), V2V communication(차량 간 통신), Convoy driving(대열주행), Autonomous vehicle(자율주행자동차), Sensor fusion (센서 융합)

1. 서 론

자율주행자동차의 글로벌 시장 규모가 2020년 71억 달러에서 2030년에는 1조 1,204억 달러까지 연평균 41% 가량 성장할 것으로 예상되고 있다.¹⁾ 이처럼 자율주행 기술이 발달하면서 기존 자율주행 시스템의 성능 개선 및 안전성 확보에 대한 필요성이 증가하고 있다. 기존의 ACC(Adaptive Cruise Control), LKAS(Lane Keeping Assist System), AEB(Autonomous Emergency Braking)와 같은 ADAS 기능들은 운전자를 보조하여 편의성과 안전성을 향상시켜준다. 하지만 이들 시스템은 레이더, 카메라와 같은 환경 센서를 기반으로 작동하므로 인지 범위 및 획득할 수 있는 정보의 종류가 제한적이며,²⁾ 이로 인해 이들 ADAS 기능이 작동될 수 있는 범위 즉 ODD

(Operational Design Domain)가 제한적이라는 문제를 안고 있다.

이를 극복하기 위해 주변 차량들과의 근거리 무선 통신으로 정보를 공유할 수 있는 V2V 통신을 활용하는 방안이 대안으로 떠오르고 있다. V2V 통신을 통해 주변 차량의 상태 및 각종 정보들을 수신하므로 시야가 확보되지 않는 부분에 대한 정보도 획득할 수 있어 인지 범위를 확대할 수 있고, 환경 센서로는 계측할 수 없었던 다양한 정보들을 이용해 차량을 제어할 수 있다.

V2V 통신을 이용한 자율주행 제어 알고리즘 중 가장 대표적인 알고리즘인 CACC(Cooperative Adaptive Cruise Control)는 전방 차량의 가속도 정보를 V2V 통신으로 획득하고 이를 자차량의 종방향 제어에 사용함으로써 기

*Corresponding author, E-mail: kpark@kookmin.ac.kr

This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium provided the original work is properly cited.

존의 ACC보다 짧은 차간 거리에도 안정적인 주행을 할 수 있다.³⁾ Song 등⁴⁾은 V2V 통신을 통해 차차량과 같은 차선에 있는 여러 선행 차량의 정보를 수신하여 전방에서 발생한 에러가 뒤로 누적되며 전파되지 않도록 하는 중방향 제어 기술도 개발되었다. 하지만 이러한 CACC는 ACC와 마찬가지로 차차선에 존재하는 전방 차량과의 거리 유지만을 목표로 개발되어, 전방으로 새로운 차량이 합류/이탈하는 과도 상황에 대해서는 거동이 불안정해지는 단점이 있다.

이러한 단점을 해결하고자 Xu 등⁵⁾은 고속도로의 병합 구간과 같이 합류 지점이 명확한 경우, 합류 차량은 합류 지점에서 대기하면서 합류 신호를 송출하고 신호를 수신한 주변 차량은 감속을 통해 합류할 거리를 확보해 차량이 안전하게 합류할 수 있도록 하는 연구를 수행하였다. 하지만 이 알고리즘은 합류 지점이 명확하게 존재해야 하며, 합류 차량이 정차 후 거리가 확보될 때까지 대기해야 하는 한계점이 있다.

Milanés 등⁷⁾은 PID 제어와 Feed forward를 위한 가속도 게인 설정을 이용해 합류/이탈 상황을 모두 커버할 수 있는 알고리즘에 대한 연구를 수행하였다. 하지만 합류를 위한 거리를 미리 확보한 상태에서 합류하는 방식이 아니라 대열에 합류 후 Time gap 거리 제어를 하는 방식을 채택했기 때문에 짧은 거리에서 합류 시 가속도 값이 크게 증가하는 문제가 있었다.

이에, 고원식과 장동희⁶⁾는 합류 신호를 수신하면 합류 차량과 동일한 중방향 위치를 가지는 가상 차량을 차차선 전방에 생성하고, 가상 차량과의 거리를 제어하는 알고리즘을 통해 합류 거리를 미리 확보하여 한계를 극복하고자 하였다. 하지만 이 알고리즘은 합류 상황에서는 적용이 가능하지만 이탈 상황에 대해서는 대응하지 못하는 한계가 존재한다.

이처럼 많은 연구들이 V2V 통신과 CACC 알고리즘을 이용하여 다양한 주행 상황에 적용할 수 있는 알고리즘을 개발하고자 하였으나, 각각 제약조건이나 한계점들이 존재해 추가적인 연구가 필요한 상황이다.

CADS(Cooperative Autonomous Driving System)로 정의되는 자율협력주행 기술은 V2V 통신과 CACC 알고리즘을 기반으로 주변 차량들과 대열을 이루어 주행하면서 차선당 유효 용량을 증가시키고, 교통 흐름을 개선해 효율적인 주행을 가능하게 한다. 긴급 상황 시에도 신속하고 안전하게 정차하거나 다양한 상황에서도 대열을 유지하는 능력이 뛰어나 안전성 측면에서도 장점이 있다. 본 논문에서는 V2V통신 및 환경 센서와의 센서퓨전을 통해 주변 차량의 정보를 획득하고, 이를 바탕으로 주변 차량과의 대열을 형성하고 동시에 차량의 합

류 이탈 등의 상황에서도 대열을 안정적으로 유지하기 위한 자율협력주행 알고리즘을 개발하고자 한다.

제안한 알고리즘의 개발 및 성능 검증을 위해 시뮬레이션 환경을 구축하였으며, 모든 알고리즘 로직은 Matlab/Simulink를 이용해 구현하였으며, 차량 모델과 주행 환경은 CarMaker를 이용해 구현하였다. 다양한 시나리오 상황에 대해 제안한 자율협력주행 알고리즘에 대한 기존 ACC 알고리즘과 CACC 알고리즘과의 비교 평가를 통한 성능 검증을 수행하였다. 그 결과 본 연구에서 개발한 자율협력주행 알고리즘이 선행 차량 추종 성능과 안정성을 개선하였으며, 작은 가속도 변화로도 우수한 요구 거리 제어 성능을 확보하는 것을 확인할 수 있었다.

2. 시뮬레이션 환경 구성

본 논문에서는 MATLAB/Simulink와 CarMaker를 이용해 Fig. 1과 같이 자율협력주행 시스템의 아키텍처를 설계하고, 시뮬레이션 환경을 구축하였다. CarMaker는 차량 모델과 센서 시뮬레이션 환경을 제공하고 Simulink는 CarMaker에서 제공되는 정보들을 바탕으로 알고리즘을 구현할 때 이용되었다.

먼저, CarMaker에서 제공되는 Traffic vehicle에 대한 정보를 이용하여 V2V 통신 모듈을 모사하였으며, 차량 모델의 Chassis CAN 정보와 Radar 센서 정보를 이용하여 센서퓨전 모듈을 구현하였다. 이때, V2V 통신모듈에 적용된 메시지는 V2V통신에 활용되는 메시지셋을 규정한 SAE J2735-2009 표준의 Basic safety message set⁸⁾의 일부와 CACC 알고리즘 구현을 위해 필요한 Convoy gap error를 포함해서 Table 1과 같이 구성하였다. Convoy gap error는 대열의 가장 앞에 존재하는 차량의 속도를 기준으로 Time gap 거리 제어를 위한 요구 상대 거리를 설정했을 때 대한 오차 값을 의미한다. 요구 상대 거리는 대열주행 전략에 따라 다르게 설정할 수 있지만, FSRA(Full Speed Range ACC)의 기준을 정의한

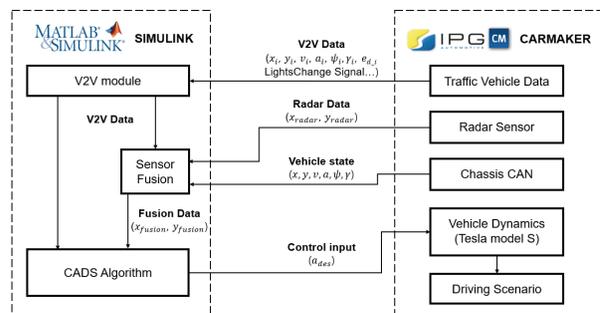


Fig. 1 Simulation environment of CADS

Table 1 V2V message set for CADs

V2V message set	
Vehicle ID	
Position	Longitude [m]
	Latitude [m]
Motion	Velocity [m/s]
	Yaw angle [rad]
	Longitudinal acceleration [m/s ²]
	Yaw rate [rad/s]
Convoy	Convoy gap error [m]
	Turn signal [-]

ISO 22179에 따르자면 선행 차량의 급정거 시에도 2 m 이상이 될 수 있도록 해야 한다.⁹⁾

3. 전방 차량 위치 추정 알고리즘

자율협력주행 기술을 구현하려면 자차량의 위치뿐만 아니라 주변 차량의 위치도 정확하게 알 수 있어야 한다. 글로벌 좌표계에 대한 물체의 위치는 GPS를 사용하면 측정할 수 있는데, 현재 양산 차량에 사용되고 있는 GPS는 오차가 수 미터까지 발생할 수 있어 이를 그대로 사용될 경우 잘못된 위치 정보로 인해 큰 사고로 이어질 수 있다.¹⁰⁾ 이번 장에서는 이러한 문제를 해결하기 위해 본 연구에서 개발한 전방 차량의 위치 추정 알고리즘에 대해 설명하고자 한다.

3.1 오차 가중치를 이용한 센서퓨전

본 연구에서는 V2V 통신을 통해 획득할 수 있는 전방 차량의 GPS 좌표 정보와 전방 차량의 속도 정보를, 자차량의 GPS 좌표 정보와 레이더 센서 정보와 융합하여 전방 차량의 좌표를 추정하는 알고리즘을 개발하였다.

앞서 설명한 것과 같이 GPS로부터 얻는 좌표값은 최대 수 미터의 거리오차를 포함할 수 있기 때문에 이를 제어 로직에 그대로 사용할 경우 큰 사고로 이어질 수 있다. 본 연구에서는 이러한 문제를 해결하기 위해 GPS의 위치 좌표값에 비해 상대적으로 오차가 적은 차속 정보를 활용하여 위치 좌표값의 정밀도를 높이고자 하였다.

먼저, 전방 차량의 속도는 위치 좌표값을 이용한 2가지 다른 방법으로 계산할 수 있다. 첫째, V2V를 통해 받은 전방 차량의 GPS 좌표값(x_g, y_g)의 한 샘플 동안의 변화량으로 계산할 수 있다. 둘째, 자차가 레이더로 측정된 전방 차량의 좌표값(x_r, y_r)에 대해 같은 방법으로 계산할 수 있다. 식 (1)과 (2)는 이 관계식을 보여준다.

$$v_{g,k} = \frac{\sqrt{(\Delta x_{g,k} - \Delta x_{g,(k-1)})^2 + (\Delta y_{g,k} - \Delta y_{g,(k-1)})^2}}{dt} \quad (1)$$

$$v_{r,k} = \frac{\sqrt{(x_{r,k} - x_{r,(k-1)})^2 + (y_{r,k} - y_{r,(k-1)})^2}}{dt} \quad (2)$$

위 식에서 아래 첨자 (\cdot)_k와 (\cdot)_{k-1}은 샘플 순간을 나타내며, dt는 샘플링 주기를 나타낸다. ($\Delta x_g, \Delta y_g$)는 전방 차량의 GPS 좌표값(x_g, y_g)에서 자차량의 GPS 좌표값을 뺀 값으로, 전방 차량의 위치를 자차량의 로컬 좌표계 기준으로 변환하기 위한 값이다. 이렇게 한 이유는 추후 센서퓨전에서 전방 차량의 좌표값을 센서퓨전을 통해 계산할 때 자차량의 로컬 좌표계를 기준으로 계측된 전방 차량의 좌표값(x_r, y_r)과 가중치를 이용한 덧셈을 가능하게 하기 위함이다.

식 (1)과 (2)은 이들 각각의 한 샘플 동안의 변화량을 구하고 있기 때문에, 두 좌표값에 오차가 존재하지 않는다고 가정하면 $v_{g,k}$ 와 $v_{r,k}$ 는 동일한 값이어야 함을 알 수 있다. 또한 주목할 것은 ($\Delta x_g, \Delta y_g$)와 (x_r, y_r)은 방향성을 가지는 벡터 물리량이지만 $v_{g,k}$ 와 $v_{r,k}$ 는 크기만을 가지는 스칼라 물리량이라는 점이다.

전방 차량의 차속을 구하는 방법은 위에서 본 $v_{g,k}$ 와 $v_{r,k}$ 이외에 한가지가 더 있는데, 그것은 전방 차량의 사시 CAN에서 출력되는 전방 차량의 차속 정보를 V2V 통신을 통해 수신하는 것이다(Table 1 참조). 이 값을 $v_{c,k}$ 라고 하면, 현재 샘플 k에서 전방 차량의 차속을 나타내는 값으로 총 3개 즉 v_g, v_r, v_c 를 얻을 수 있다. 본 연구에서는 이들 가운데서 v_c 를 상대적으로 가장 신뢰할 수 있는 값으로 판단하여 기준값으로 삼았으며 그 근거는 다음과 같다.

먼저, v_g 의 경우 GPS가 출력하는 좌표계 값 ($\Delta x_g, \Delta y_g$)을 이용해 식 (1)을 통해 얻어지는데, 현재 양산 차량에 장착되고 있는 GPS의 경우 위치 오차가 최대 수 미터까지 나는 것으로 알려져 있다.¹⁰⁾ v_r 은 레이더가 출력하는 좌표계 값 (x_r, y_r)을 이용해 식 (2)를 통해 얻어지는데, 레이더 센서의 경우 여러 대의 차량이 밀집해 있는 구간에서는 간섭 현상이 발생해 허위 목표물이 생성되거나 목표물을 측정하지 못하는 경우가 발생할 수 있다.¹¹⁾ 또한, 레이더 센서의 특성상 종방향 좌표에 대해서는 비교적 정확하게 계측하지만, 횡방향 좌표는 좌우로 오차가 발생한다.¹²⁾ 반면 사시 CAN을 통해 얻어지는 차속 v_c 의 경우 본 연구에서 고려하는 경우와 같이 직진 상황이나 횡방향 속도 성분이 작은 경우에는 신뢰성이 높은 것으로 알려져 있으며,¹³⁾ In-Wheel을 적용한 전기차에 제한적으로 장착이 되고 있는 비접촉 광학식

속도 센서를 이용할 경우 오차범위를 0.1 % 이내로 줄일 수 있다는 연구도 보고된 바 있다.^{14,15)}

이제 세 가지 다른 방법을 통해 구한 선행 차량의 속도 v_g, v_r, v_c 를 이용하여, 선행 차량의 위치 좌표계 값의 오차를 줄이기 위해 본 연구에서 제안하는 방법을 설명한다. 먼저 위 세 값 가운데 v_c 를 기준값으로 삼은 다음, 나머지 두 값의 오차를 정의한다.

$$e_g = |v_c - v_g| \tag{3}$$

$$e_r = |v_c - v_r| \tag{4}$$

이렇게 계산된 오차와 $(\Delta x_g, \Delta y_g)$ 와 (x_r, y_r) 로부터 아래 식을 이용해 전방 차량의 좌표값을 계산한다.

$$x_{fs} = \frac{e_g}{e_g + e_r} x_r + \frac{e_r}{e_g + e_r} \Delta x_g \tag{5}$$

$$y_{fs} = \frac{e_g}{e_g + e_r} y_r + \frac{e_r}{e_g + e_r} \Delta y_g \tag{6}$$

위 식을 보면, 두 센서 즉 GPS와 레이더 신호로부터 계산한 오차 가중치와 좌표값이 서로 교차되어 곱해지는 것을 볼 수 있는데, 이는 오차가 작은 센서 데이터에 더 큰 가중치를 곱해주기 위함이다.

3.2 확장 칼만필터

대열주행의 합류/이탈 상황과 같이 차속이 빠르게 변하는 상황에서도 신뢰성 높은 전방 차량의 좌표값을 얻기 위해, 본 연구에서는, 확장 칼만필터를 이용해 센서 퓨전을 통해 추정된 좌표값을 추가적으로 개선하는 작업을 수행하였다. 먼저 칼만필터를 위한 상태벡터는 아래와 같이 정의하였다.

$$\chi_k = [x_k, y_k, v_{x,k}, v_{y,k}]^T \tag{7}$$

위 식에서 는 전방 차량의 좌표이며, 는 전방 차량의 종/횡방향 속도이며, Fig. 2는 이들 값에 대한 좌표계를 보여준다.

칼만필터를 위한 측정벡터는 식 (8)과 같이 정의하였다.

$$\xi_k = [x_{fs,k}, y_{fs,k}, v_{c,k}]^T \tag{8}$$

칼만필터의 상태방정식은 식 (9)로 표현할 수 있다. 자율협력주행 로직의 샘플 주기는 자율주행 제어기의

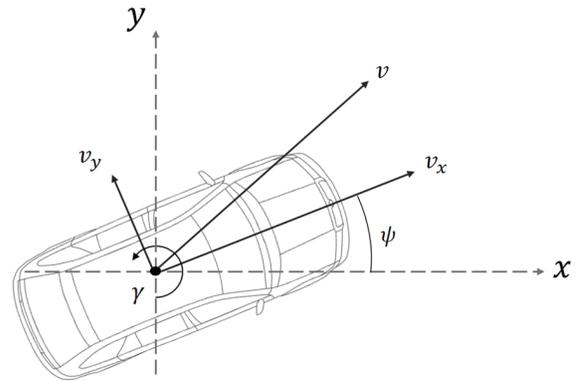


Fig. 2 States for Kalman filter

샘플 주기와 유사한 0.01초를 갖게 되는데, 이렇게 짧은 시간 동안에는 차량의 가속도의 미분인 저크를 0으로 가정하는 것은 무리가 없으며 상태방정식의 천이행렬 A는 식 (10)과 같이 된다.

$$\chi_k = A \chi_{k-1} \tag{9}$$

$$A = \begin{bmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{10}$$

측정벡터와 상태벡터 사이에는 식 (11)과 같은 비선형 관계가 성립한다.

$$\xi_k = h(\chi_k) = \begin{bmatrix} x_k \\ y_k \\ \sqrt{v_{x,k}^2 + v_{y,k}^2} \end{bmatrix} \tag{11}$$

확장 칼만필터에서는 매 샘플마다 이러한 비선형식을 Jacobian을 통해 선형화시켜 사용하게 되는데 선형화된 측정방정식은 식 (12)와 같다.

$$\xi_k = H_k \chi_k \tag{12a}$$

$$\text{where } H_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{v_{x,k}}{\sqrt{v_{x,k}^2 + v_{y,k}^2}} & \frac{v_{y,k}}{\sqrt{v_{x,k}^2 + v_{y,k}^2}} \end{bmatrix} \tag{12b}$$

식 (12b)를 보면 H_k 를 구성하기 위해서 $v_{x,k}$ 와 $v_{y,k}$ 가 필요하지만, 이들 값은 현재 샘플에서 칼만필터 알고리즘의 수행이 완료되고서야 그 결과물로 나오는 값들임을 알 수 있다. 이런 문제는 측정방정식의 행렬 H_k 를 구성

할 때, 한 샘플 이전에서 계산된 상태변수 추정값을 사용함으로써 해결할 수 있다.¹⁶⁾

칼만필터는 아래와 같은 식 (13)~(15)와 같은 단계로 진행된다. 먼저, 상태방정식으로부터 상태벡터 \hat{x}_k 를 추정하고 오차 공분산 P_k 를 예측한 후, 이를 바탕으로 칼만이득 K_k 를 구한다. 이때, Q 와 R 은 각각 시스템과 측정값에 대한 잡음 공분산 행렬이다.

$$\hat{\chi}_k = A \hat{\chi}_{k-1} \quad (13)$$

$$P_k = A P_{k-1} A^T + Q \quad (14)$$

$$K_k = P_k H_{k-1}^T (H_k P_k H_k^T + R)^{-1} \quad (15)$$

그 후, 칼만필터의 상태방정식 (9)와 측정방정식 (12)을 이용해 식 (16)과 같이 상태변수를 예측한다.

$$\hat{\chi}_k = \hat{\chi}_k + K_k (\xi_k^T - H_k \hat{\chi}_k) \quad (16)$$

마지막으로 오차 공분산은 다음 식 (17)을 통해 업데이트되며, 다시 식 (13)의 과정으로 돌아가 다음 상태를 예측한다.

$$P_k = P_k - K_k H_k P_k \quad (17)$$

3.3 알고리즘 성능 검증

이번 절에서는 앞의 두 절에서 소개한 전방 차량 위치 추정 알고리즘의 성능 검증을 위해 수행한 시뮬레이션 결과를 보여준다. 보다 실제적인 검증을 위해 각 센서 데이터에 Table 2와 같이 사인 함수와 가우시안 노이즈를 이용한 Non-gaussian noise를 추가하였다.¹⁷⁾ Table 2에서 $\mathcal{N}(\mu, \sigma)$ 는 평균이 μ 이고 표준편차가 σ 인 가우시안 확률분포를 의미한다.

Table 3은, 자차량이 전방 차량과 동일한 속도로 주행할 때 CarMaker에서 획득할 수 있는 전방 차량의 정확한 좌표를 기준으로, 본 논문에서 제안된 알고리즘을 통해 추정된 좌표값의 RMSE(Root Mean Square Error)를 보여주며, 이와 함께 알고리즘의 입력으로 들어가는 GPS 좌표값과 레이더 좌표값의 RMSE를 함께 보여준다.

Table 3을 보면 차량의 속도가 증가하면서 제안된 알고리즘의 RMSE가 증가하는 것을 볼 수 있다. 이는 GPS 오차는 차속이 증가해도 거의 변하지 않는 반면, 레이더의 오차는 차속에 따라 증가하기 때문인 것으로 판단된다. 하지만, 제안된 알고리즘은 레이더 센서의 측정값에 비해 정확도를 약 0.1 m가량 향상시키고 있는 것을 볼

Table 2 Noise added to measurement signals

Signal	Noise added to signal
Δx_g	$3.5\sin(\mathcal{N}(0,0.1) + \mathcal{N}(1.5,1) + 5)$
Δy_g	$3.5\sin(\mathcal{N}(0,0.1) + \mathcal{N}(1.5,1) + 5)$
x_r	$\sin(\mathcal{N}(0,0.02))$
y_r	$\sin(\mathcal{N}(0,0.05))$
v_c	$\sin(\mathcal{N}(0,0.1))$

Table 3 RMSE in estimation of proposed algorithm (unit: m)

Velocity (m/s)	RMSE GPS	Radar RMSE	RMSE Proposed
17	2.225	0.187	0.085
23	2.224	0.207	0.101
28	2.230	0.225	0.115

수 있다. 이를 통해 전방 차량에 대한 GPS 좌표값과 레이더 좌표값이 오차를 포함하더라도 본 연구에서 개발한 추정 알고리즘을 통해 전방 차량의 좌표값의 오차를 효과적으로 보정 할 수 있다는 것을 확인할 수 있었다.

4. 자율협력주행을 위한 종방향 알고리즘 개발

자율협력주행은 V2V 통신을 기반으로 데이터를 획득하기 때문에 환경 센서에 비해 넓은 인지 범위와 다양한 정보를 활용할 수 있어, 환경 센서 기반의 ACC에 비해 선행 차량의 급정지 및 Cut-in 상황 같은 주행 환경의 급격한 변화에도 빠르게 대응할 수 있는 장점이 있다. 4장에서는 본 연구에서 개발한 대열주행을 위한 V2V 통신 기반의 자율협력주행 알고리즘에 대해 설명하고자 한다.

본 연구에서 제안하는 자율협력주행 알고리즘은 CACC mode, Merging mode, Leaving mode의 총 3가지 모드로 구성된다. Fig. 3은 제안하는 알고리즘의 Finite state machine 형태의 모델을 나타낸다.

위 그림에서 δ 는 전방 차량과의 상대거리, δ_{des} 는 δ 의 요구값을 나타낸다. Merging mode flag와 Leaving mode flag로는 Table 1의 V2V 메시지셋 중 방향지시등을 나타내는 Turn signal 신호를 이용하였다. 전방 차량이 자차량과 동일 차선에 있는지를 판단하는 Same lane 논리값은 현재 자차량이 주행 중인 차선의 중심선 좌표와 전방 차량과의 거리를 통해 판단하였다. 또한, 자차량의 위치 및 주변 차량의 좌표는 3장에서 개발한 센서퓨전 알고리즘과 V2V를 이용한 협력측위를 통해 보정된 값을 수신받는다는 가정하에 연구를 진행하였다.

차량이 CACC mode로 대열을 형성하면서 주행 중,

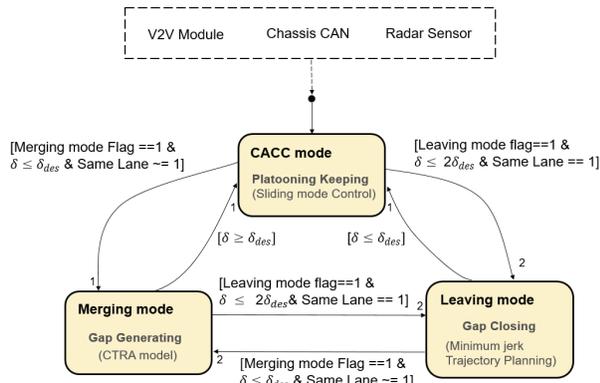


Fig. 3 Finite state machine model of proposed CADs

V2V 통신으로 합류/이탈 신호를 수신하게 되면 먼저 설정된 거리 내에 위치한 차량인지 확인하고 차선 정보를 확인해 해당하는 모드로 전환하게 된다. 이후, 본 연구의 알고리즘이 실행되면서 요구 상대거리를 달성하게 되면 다시 CACC mode로 돌아와 주행한다.

본 연구의 자율협력주행 알고리즘은, CACC mode에서는 Sliding mode control 기법을 이용해 기존의 대열을 유지하기 위한 목표 가속도를 생성한다. Merging mode에서는 전방에 대열 합류 차량 발생 시, 등선회율-등가속도 모델을 이용해 합류 차량의 경로를 예측하고 거리 제어를 위한 목표 가속도를 생성한다. Leaving mode에서는 전방 차량이 대열을 이탈하는 경우 Minimum jerk trajectory planning 기법을 통해 상대거리를 줄이기 위한 목표 가속도를 생성한다. 아래에서는 각각의 모드에서 수행되는 알고리즘의 자세한 내용을 살펴본다.

4.1 CACC Mode 알고리즘

본 논문에서 제안하는 CACC 알고리즘은 기존의 군집주행 알고리즘과는 다르게 정해진 선두차량이 없이 같은 차선 내의 차량들이 대열을 형성하여 주행하기 위해 사용된다. 이렇게 하는 것은 V2V를 이용한 자율협력주행 시스템에서 어디까지를 하나의 대열로 선정하는지, 어떤 차량을 기준으로 좌표 일원화를 진행하는지에 대한 기준을 선정하기 어려울 뿐만 아니라 이에 따른 장점도 크게 없기 때문이다. 본 논문에서는 V2V를 통해 획득하는 모든 정보를 자차량 기준의 좌표계로 변환하여 제어에 활용하고 있어 제어 측면에서 효율적이며, 레이더 정보뿐만 아니라 다양한 정보를 바탕으로 차량을 제어하는 것을 통해 기존의 ACC에 비해 안전하고 효율적인 제어 수행을 한다.

Fig. 3에서 본 것과 같이 본 논문의 자율협력주행 시스템은 대열주행 유지 시 CACC mode를 수행하게 되는

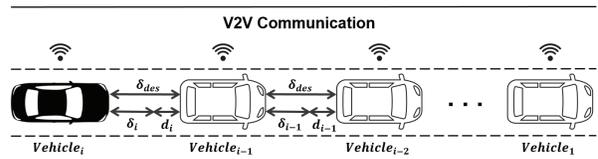


Fig. 4 Schematics of CACC mode algorithm

데, 본 연구에서는 이를 위해 V2V 통신 기반의 SMC (Sliding Mode Control) 제어를 개발하였다. V2V를 통해 획득한 선행 차량과의 가속도, 속도, 상대거리를 기반으로 대열 거리 오차가 0이 될 수 있도록 제어를 설계하였고, 연산되는 요구 가/감속도를 통해 차량의 종방향 제어를 할 수 있도록 하였다. Fig. 4는 CACC 알고리즘의 개념도를 보여준다. 차량 번호는 자차량이 i 이고 앞으로 가면서 번호가 하나씩 줄어들도록 정의하였다.

SMC 제어기의 구현을 위해 먼저 종방향 제어 오차의 Sliding surface를 아래 식과 같이 정의하였다.

$$S = k_1 e_\delta + k_2 \dot{e}_\delta + k_3 e_v + k_4 e_a + k_5 e_d \quad (18a)$$

$$e_\delta = d_i - \delta_{des} - \delta_i \quad (18b)$$

$$e_v = v_i - v_{i-1} \quad (18c)$$

$$e_a = a_i - a_{i-1} \quad (18d)$$

위 식에서 e_δ 는 차간거리 오차를, e_v 와 e_a 는 자차와 선행차량 간의 속도 및 가속도 오차를 나타낸다.

e_d 는 V2V 통신을 통해 수신받는 Convoy gap error를 나타내며 이 값은 식 (19)에서 보는 것과 같이 자차의 전방에 존재하는 대열 참여 차량들에 대한 e_δ 의 누적 합을 나타낸다.

$$e_d = \sum_{n=2}^{i-1} e_{\delta_n} = \sum_{n=2}^{i-1} d_n \quad (19)$$

식 (18)에서 δ_{des} 는 아래 식과 같이 Time gap τ , 선행 차량 속도 v_c , 안전 제동거리 δ_0 로 구성된다.

$$\delta_{des} = \tau v_c + \delta_0 \quad (20)$$

선행 연구에 따르면 일반적인 CACC 알고리즘은 Time gap 0.5초 이상에서 작동하는 것을 볼 수 있다.¹⁸⁾ 안전 제동거리는 정차 시 앞차와의 간격을 의미하며, FSRA에 대한 ISO 표준을 참고하여 2 m 이상으로 설정했다.⁹⁾

다음으로는 식 (18)에서 정의한 Sliding surface에 도달하기 위한 조건을 구하기 위해 S 에 대한 Lyapunov 함수 V 를 식 (21)과 같이 정의하였다.

$$V = \frac{1}{2} S^2 \quad (21)$$

$$\dot{V} = S \dot{S} \leq 0 \quad (22)$$

Lyapunov 함수의 S 에 대한 미분은 식 (22)와 같은 Negative semidefinite 조건을 만족해야 하는데, 이를 만족시키기 위한 식을 도출하면 식 (23)과 같다.

$$S \dot{S} \leq -\lambda S \text{sign}(S), \lambda > 0 \quad (23)$$

이제 위의 Sliding condition을 만족하는 제어 입력을 구하면 다음과 같다.

$$u = u_{eq} - \lambda \text{sign}(S) \quad (24)$$

본 논문의 CACC 모드에서 제어 입력 u 는 목표 가속도를 나타내며 u_{eq} 는 등가 제어 입력값을 나타내고, 제어이득 λ 는 설계 변수이다. 일반적으로 λ 를 작게 하면 채터링 현상이 감소한다. 따라서 시스템의 궤적이 Sliding surface에 거의 도달하면 채터링을 줄이기 위해 가능한 한 λ 를 작게 한다.

S 가 Sliding surface에 도달하게 되면 $S=0$ 이 되므로 $u = u_{eq}$ 가 되는데 CACC 시스템의 등가 제어 입력값은 0이어야 하므로 u 는 식 (25)가 된다.

$$u = -\lambda \text{sign}(S) \quad (25)$$

함수는 $S=0$ 인 지점에서 불연속이기 때문에 채터링 현상을 방지하고자 $\text{sign}()$ 대신 $\text{sat}()$ 함수를 사용하였다.

$$u = -\lambda \text{sat}(S) \quad (26)$$

4.2 Merging Mode 알고리즘

대열주행 중 주변 차량이 차선 변경을 통해 전방으로 합류하는 경우, 합류 차량을 제대로 인지하지 못하거나 충분한 감속을 하지 못한다면 충돌 사고로 이어질 수 있다. 합류 차량에 대한 경로 예측을 통해 합류 지점을 파악하고 적절한 요구 감속도를 계산해 충분한 거리를 선제적으로 확보한다면, 충돌 사고를 방지할 뿐만 아니라 급격한 감속을 방지하는 것을 통해 대열 유지 측면에서도 도움이 된다.

차량의 경로를 예측하는 방법에는 주로 물리 기반 모델과 거동 기반 모델이 활용되어 왔다. 물리 기반 모델은 등속도 모델(Constant velocity model), 등가속도 모델(Constant acceleration model) 등이 있으나 차선변경과 같이 차량의 Yaw 거동이 발생하는 상황에서는 오차가 크게 발생해 짧은 시간 예측만 가능하다는 한계가 있다.¹⁹⁾

반면, 거동 기반 모델은 등선회율-등속도 모델(Constant turning rate and Velocity model)이나 등선회율-등가속도 모델(Constant turning rate and Acceleration model)²⁰⁾ 같이 차량의 Yaw 정보를 이용함으로써 선회 상황에서 물리 기반 모델의 경로 예측방법이 가지는 한계를 보완할 수 있다. 그럼에도 이들 방법은 환경 센서만으로는 주변 차량의 정확한 Yaw angle, Yaw rate의 산출이 어려운 문제가 있어 그동안 자차량의 경로 예측 및 위치 추정에만 주로 사용되어 왔다. 하지만 V2V 통신을 이용하면 주변 차량의 Yaw 관련 정보를 수신받을 수 있어 이러한 한계를 극복하고 주변 차량의 경로를 예측할 수 있다.

이번 절에서는 등선회율-가속도 모델을 이용해 대열에 합류하고자 하는 차량의 경로 및 합류 지점을 예측하고, 합류 지점을 기준으로 Time gap 제어를 위한 거리를 확보하기 위한 요구 감속도 값을 계산하는 본 연구의 Merging mode 알고리즘을 소개한다. Fig. 5는 Merging mode 알고리즘의 개념도를 보여준다.

좌/우측 차선의 측전방에 위치하고 있는 차량이 차선의 대열에 합류하고자 V2V 통신을 통해 Turn signal을 보내면, 합류 차량의 좌/우측, 속도, 가속도, Yaw angle, Yaw rate 값을 수신하고, 식 (27)~(32)에 정의된 등선회율-등가속도(CTRA) 모델을 통해 합류 차량의 주행 경로를 예측한다.

$$\tilde{x}_{k+1} = \tilde{x}_k + \frac{\tilde{v}_{k+1}}{\tilde{\gamma}_{k+1}} (-\sin(\tilde{\psi}_k) + \sin(\tilde{\gamma} \Delta t_k + \tilde{\psi}_k)) \quad (27)$$

$$\tilde{y}_{k+1} = \tilde{y}_k + \frac{\tilde{v}_{k+1}}{\tilde{\gamma}_{k+1}} (\cos(\tilde{\psi}_k) - \cos(\tilde{\gamma} \Delta t_k + \tilde{\psi}_k)) \quad (28)$$

$$\tilde{\psi}_{k+1} = \tilde{\psi}_k + \tilde{\gamma}_k \Delta t \quad (29)$$

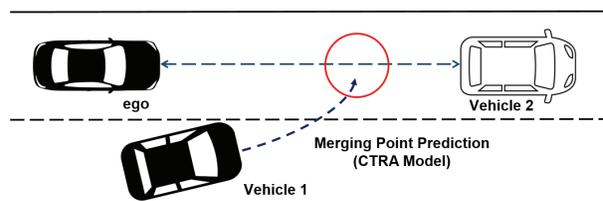


Fig. 5 Schematics of merging mode algorithm

$$\tilde{v}_{k+1} = \tilde{v}_k + \tilde{a}_k \Delta t \quad (30)$$

$$\tilde{\gamma}_{k+1} = \tilde{\gamma}_k \quad (31)$$

$$\tilde{a}_{k+1} = \tilde{a}_k \quad (32)$$

위 식에서 $\tilde{\psi}$ 와 $\tilde{\gamma}$ 는 류 차량의 Yaw angle과 Yaw rate의 값을 나타내며, \tilde{v} 와 \tilde{a} 는 합류 차량의 속도, 가속도 값을 나타낸다. CTRA 모델을 이용해 합류 차량의 거동을 매 샘플 Δt 초 마다 예측하며, Fig. 6은 시간에 따른 자차량의 위치와 합류 차량의 예측된 위치를 보여준다.

이렇게 예측한 경로를 바탕으로 합류 차량이 내 차선에 진입하는 순간의 좌표를 합류 지점으로 선정하고, 이때의 좌푯값을 기준으로 Time gap 제어를 위한 거리를 확보할 수 있는 요구 감속도 값을 생성한다. 요구 감속도를 구하기 위한 식은 다음과 같다.

$$D = v_0 t_m + \frac{1}{2} a t_m^2 \quad (33)$$

$$a = 2 \left(\frac{D}{t_m^2} - \frac{(v_c - v_{ego})}{t_m} \right) \quad (34)$$

위 식에서 t_m 은 차선 변경 시간을 나타낸다. 기존의 연구에 의하면 차선 변경에 사용하는 평균 시간은 5~6 초라고 한다.²¹⁾ 자율협력주행이 되더라도 HMI 관점에서 탑승자가 불안감을 덜 느낄 수 있는 익숙한 시간에 맞춰 차선 변경이 이루어질 것으로 가정하고 본 연구에서는 t_m 을 5초로 설정하였다. D 는 CTRA 모델로부터 예측된 합류 지점으로부터 CACC time gap 제어를 만족하는 거리이며 v_0 는 자차량의 초기 속도, v_c 는 합류 차량의 속도, v_{ego} 는 자차량의 속도를 의미한다.

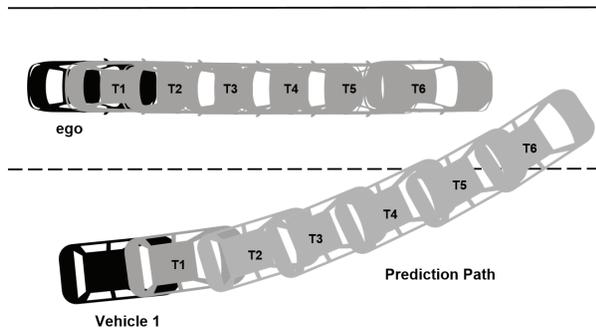


Fig. 6 Position prediction by CTRA model

4.3 Leaving Mode 알고리즘

환경 센서를 활용한 ACC의 경우, 선행 차량을 추종하며 주행하던 중 선행 차량이 차선 변경으로 이탈하게 되면, 그 앞에 다른 자동차가 가까이 있더라도 이를 제대로 인지하지 못하고 설정된 속도로 주행하는 등속 모드로 전환되었다가 설정거리에 도달했을 때 다시 차간거리 유지 모드로 전환하게 된다. 이때, 등속 모드에 설정된 속도와 선행 차량의 속도 차이가 큰 경우, 급격한 가/감속이 발생하게 되어 차량의 종방향 안정성과 승차감에 악영향을 주게 된다. 특히 자율협력주행을 이용한 대열주행의 경우, 이러한 현상이 대열의 어느 한 차량에 발생하면 대열 전체의 종방향 안정성인 String stability에 심각한 영향을 줄 수 있다. 하지만 이러한 문제점에도 불구하고 관련된 연구는 부족한 상황이다.

이번 절에서는 전방 차량이 대열에서 이탈하는 경우 이탈하는 차량 앞에 있는 새로운 선행 차량의 정보를 수신해 이탈로 발생한 간격을 좁히는 목표지점을 설정하고, Minimum jerk trajectory planning 기법을 통해 목표지점까지의 주행 거리를 가속구간 감속구간으로 나누어 급격한 가/감속 없이 부드러운 주행을 지원하는 Leaving mode 알고리즘을 소개한다. Fig. 7은 Leaving mode 알고리즘의 개념도를 보여준다.

본 연구에서는, 자율협력주행을 이용한 대열주행 중 자차선의 선행 차량으로부터 이탈 신호를 수신하게 되면 V2V 통신을 통해 선행 차량 앞의 설정된 영역 이내에 새롭게 추종할 수 있는 차량이 있는지 탐색하고, 조건에 부합하는 차량이 존재하는 경우 새로운 선행 차량으로 선정한다. 그 후, 새로운 선행 차량의 좌표와 속도 값을 바탕으로 Time gap 제어를 위한 목표 지점을 산출하고, 이에 도달하기 위한 구간별 속도 값을 Minimum jerk trajectory planning 기법을 이용해 계산한다.

Minimum jerk trajectory planning과 관련된 기존 연구에서, 5차 이상의 다항식을 사용하였을 때 가장 효율적인 성능을 준다는 것을 보인 바 있다.^{22,23)} 이에 따라서

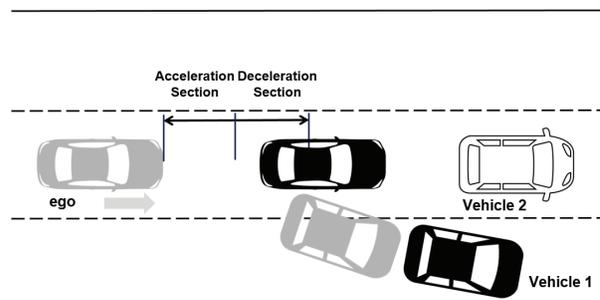


Fig. 7 Schematics of leaving mode algorithm

본 논문에서는 5차 다항식을 이용하여 최적의 속도, 가속도를 계산하였다. 초기 조건과 시작 및 끝 지점에서의 속도, 가속도 값이 0인 점을 이용하여 식 (35)~(37)의 식을 유도하였다.

$$x(t) = x_i + (x_f - x_i) \left(10 \left(\frac{t_m}{t_t} \right)^3 - 15 \left(\frac{t_m}{t_t} \right)^4 + 6 \left(\frac{t_m}{t_t} \right)^5 \right) \quad (35)$$

$$\dot{x}(t) = \frac{1}{t_t} (x_f - x_i) \left(30 \left(\frac{t_m}{t_t} \right)^2 - 60 \left(\frac{t_m}{t_t} \right)^3 + 30 \left(\frac{t_m}{t_t} \right)^4 \right) \quad (36)$$

$$\ddot{x}(t) = \frac{1}{t_t^2} (x_f - x_i) \left(60 \left(\frac{t_m}{t_t} \right) - 180 \left(\frac{t_m}{t_t} \right)^2 + 120 \left(\frac{t_m}{t_t} \right)^3 \right) \quad (37)$$

위 식에서 x_f, x_i 는 각각 목표 지점, 시작 지점의 좌표이다. 목표지점은 새로운 선행 차량인 Vehicle 2와의 CACC time gap 제어를 만족하는 위치로, Leaving mode 종료 시 CACC mode로 전환되어 거리 제어가 유지될 수 있도록 하였다. t_m 은 이동 시간, t_t 는 시작 위치에서 목표 위치까지 도달하는데 걸리는 시간을 의미한다. 이때, t_t 는 (목표 지점까지의 거리)/평균 속도로 구할 수 있다. 본 논문에서는 목표지점에 도달하기 위한 평균 속도를 선행 차량 속도의 105%로 설정하고 진행하였다.

5. 성능 검증 및 평가

이번 장에서는 본 논문에서 개발된 V2V 기반 자율협력주행 시스템의 성능을 검증하고, 이를 ACC 및 CACC의 성능과 비교 분석한 결과를 보여준다. 목표 Time gap을 모두 0.5초로 설정하였으며, 각 모드에 대해 대열주행 속도가 8 m/s인 저속 상황과 30 m/s인 고속 상황을 고려하였다. 또한, 본 논문에서는 종방향 정보를 기반으로 차량을 제어하고 있으므로 종방향 상대거리, 종방향 속도, 종방향 가속도에 대한 결과 분석을 통해 제안하는 알고리즘의 성능을 평가하고자 한다.

5.1 CACC Mode 알고리즘 검증

이번 절에서는 CACC 모드 알고리즘의 종방향 거리 제어 성능을 검증하고, 선행 차량의 급격한 변화에 대한 반응성을 ACC와 비교 평가하고자 한다.

비교에 사용된 ACC 알고리즘²⁴⁾은 실제 ACC와 같이 레이더로 측정된 차간 거리와 속도 오차를 이용해 선행 차량을 추종한다. CACC 알고리즘의 주행 성능 평가에 대한 표준이 따로 없는 관계로, FSRA(Full Speed Range ACC) 표준인 ISO 22179_2009의 Automatic stop capability test를 참고하여 검증을 진행하였다.⁹⁾

Fig. 8과 같이 자차선에서 대열을 유지하며 주행하던 중 Vehicle 2 차량이 0.25 g의 감속도로 급정차할 때, 자차량과 선행차량 사이의 거리가 ISO요구사항(2m 이상)을 만족하는지 평가를 진행하였으며, Fig. 9는 결과 그래프를 보여준다.

Fig. 9의 상대거리 그래프를 보면, CACC, ACC 모두 ISO 요구사항인 2 m를 만족하고 있어 성능을 충족하고 있다는 점을 확인할 수 있다.

CACC의 경우, ACC와 비교해 빠른 속도로 목표 상대 거리를 추종하는 것을 확인할 수 있으며, 상대 차량이 감속을 시작한 시점인 30초 부근에서 ACC보다 더 큰 상대 거리를 가지는 것을 볼 수 있다. 이는 CACC가 V2V 통신을 통해 Convoy gap error 값과 선행 차량의 가속도 값을 수신하여 제어에 반영하기 때문에 ACC에 비해 선행 차량의 급격한 변화에 빠르게 반응하여 감속을 진행할 수 있었기 때문이다.

Fig. 9의 가속도 그래프를 보면, ACC는 31초 부근에

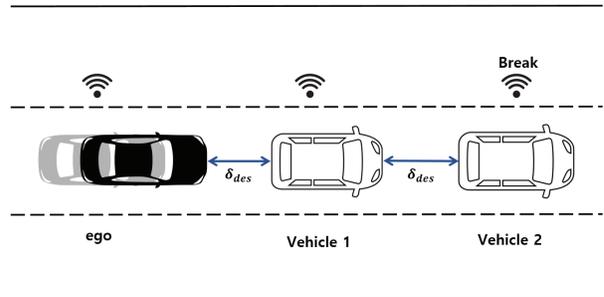


Fig. 8 Test scenario for CACC Mode

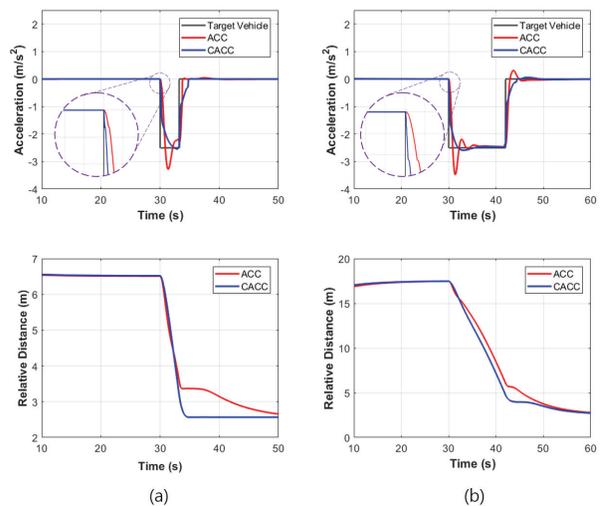


Fig. 9 Simulation results of CACC Mode algorithm (a) low velocity (b) high velocity

서 저속의 경우 28 %, 고속의 경우 39 %로 큰 오버슈트가 발생하는 것을 볼 수 있다. 하지만 CACC는 저속 상황에서 오버슈트가 발생하지 않았으며, 고속의 경우에도 4 %의 수준의 작은 오버슈트가 발생하는 것을 확인할 수 있다. 선행 차량이 급정지하는 상황에서 고속 상황이 저속에 비해 더 많은 감속을 해야 하기 때문에 가속도의 오버슈트가 더 크게 발생하는 것으로 보인다. 식 (18)에서 보는 것과 같이 5가지 오차항을 모두 고려하면서 가속도 제어를 하므로, ACC에 비해 작은 가속도 변화량으로도 목표 거리를 유지하는 것을 확인할 수 있다.

이를 통해, 본 연구의 V2V 통신을 이용한 CACC mode 알고리즘이 기존 환경센서 기반의 ACC에 비해, 선행 차량을 보다 안전하고 부드럽게 추종하는 성능을 가진다는 것을 확인할 수 있었다.

5.2 Merging Mode 알고리즘 검증

본 연구의 Merging mode 알고리즘을 평가하기 위해, 대열주행 중 Fig. 10과 같이 오른쪽 차선 0.5 m 전방에 있는 차량이 자차량의 차선으로 5초 동안 차선변경을 통해 합류하는 상황에 대한 시뮬레이션을 수행하였다.

앞 절에서 ACC보다 대열주행 성능이 뛰어난 것이 확인된 CACC와 비교 평가를 진행하였고, 그 결과는 Fig. 11과 같다.

Fig. 11의 결과를 보면 CACC의 경우, 차선에 합류하는 차량의 유무 및 합류 지점을 미리 예측하지 못하기 때문에 가속도 값이 불안정하고 큰 폭으로 변하는 것을 볼 수 있다. 특히, 저속 상황에서 약 -2 m/s^2 로 감속한 것과 비교해 고속 상황에서는 -9 m/s^2 정도로 감속도가 크게 증가한다. 이는 고속 상황에서는 저속에 비해 Time gap 제어를 위한 요구 상대 거리가 늘어나기 때문이다. 또한, 과도하게 감속을 진행했기 때문에 25초 부근에서 선행 차량과의 추종 속도를 맞추기 위해 다시 급격하게 가속을 진행하고 있음을 확인할 수 있다.

하지만 본 연구의 Merging mode 알고리즘을 적용했을 때에는 합류 신호를 기점으로 합류 차량의 거동을 예측하여 선제적인 감속과 더불어 안정적인 가속도 및 속

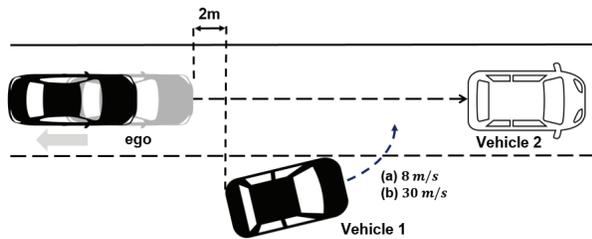


Fig. 10 Test scenario for merging mode

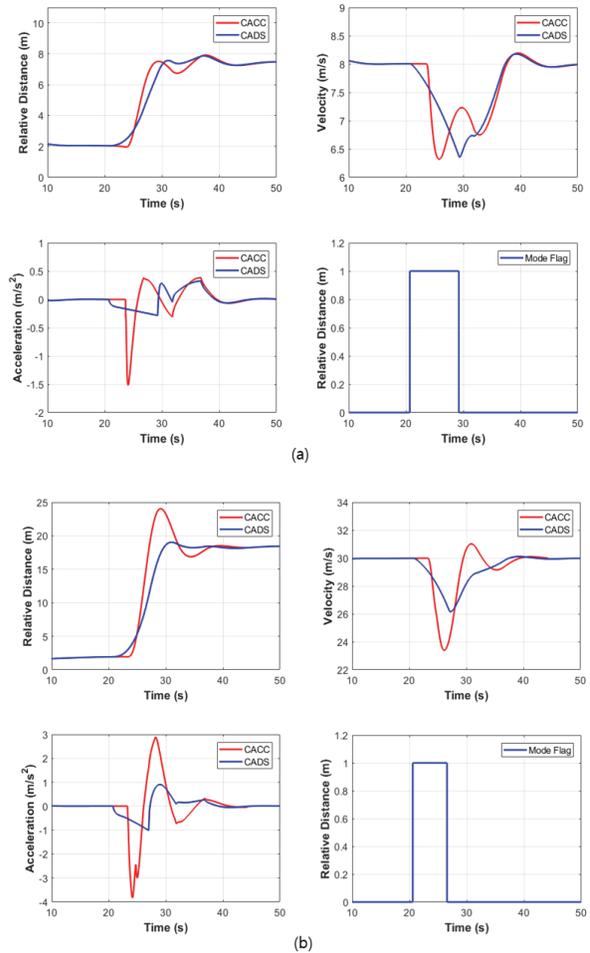


Fig. 11 Simulation results of merging mode algorithm (a) low velocity (b) high velocity

도 변화를 통해 합류 차량과의 충분한 거리를 미리 확보하며 안정적으로 주행하는 것을 확인할 수 있다. 이러한 경향은 속보다 저속에서 두드러지게 나타나는데, 저속 상황에서는 CACC 대비 약 20 % 크기의 감속도 값으로 성능을 충족시킬 수 있었으나 고속 상황에서는 약 25 % 크기의 감속도 값으로 목표 성능을 충족시킬 수 있었다. 이는 미리 합류 지점을 예측했다고 하더라도 고속 상황에서의 거리 제어를 하기 위해서는 더 큰 감속도가 필요하기 때문이다.

5.3 Leaving Mode 알고리즘 검증

본 연구의 Leaving Mode 알고리즘을 평가하기 위해, 대열주행 중 Fig. 12와 같이 전방 차량이 오른쪽 옆 차선으로 대열을 이탈하게 되었을 때 새로운 전방의 차량과 다시 대열을 형성하는 상황에 대한 시뮬레이션을 수행하였다.

앞 절과 마찬가지로 CACC와 비교 평가를 진행하였

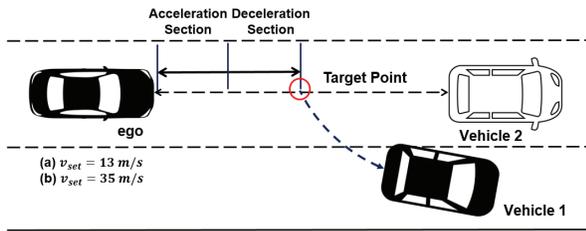


Fig. 12 Test scenario for leaving mode

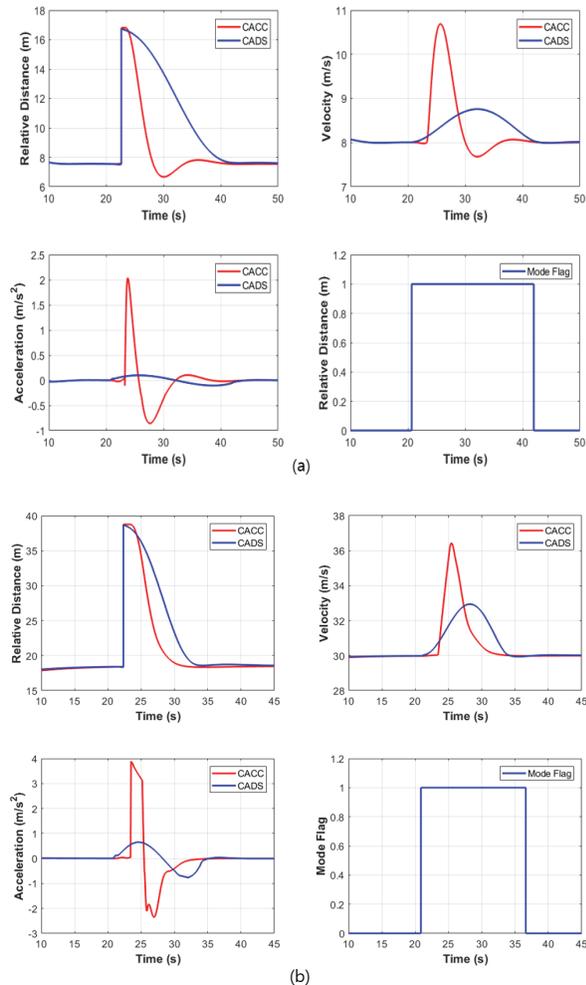


Fig. 13 Simulation results of leaving mode algorithm (a) low velocity (b) high velocity

으며 그 결과는 Fig. 13과 같다. 그림에서 상대거리는 자차량과 동일한 차선에 위치한 전방 차량과의 거리를 나타낸다.

Fig. 13의 가속도 그래프를 보면 CACC는 선행 차량이 이탈 시, 가속도의 크기가 저속의 경우 8 m/s^2 에서 -6.5 m/s^2 , 고속의 경우 4 m/s^2 에서 -2.5 m/s^2 까지 급격하게 변하는 것을 볼 수 있다. 이는 새롭게 선행 차량으

로 설정되는 Vehicle 2와의 거리가 설정된 Time gap 거리보다 멀리 있어 대열을 형성하지 못하고 설정된 속도에도달하기 위해 가속을 했다가 이후, Vehicle 2를 발견하고 Time gap 제어를 하기 위해 다시 감속을 하면서 급격한 가속도 변화가 발생하기 때문이다. 또한, 저속 상황의 경우 고속 상황과 비교하여 더 큰 가속도 변화가 발생하는데 이는 저속 상황일 때 Time gap 거리가 더 짧아 Vehicle 2와의 Time gap 제어를 위한 거리에 더 빨리 도달하여 거리 제어를 하기 위해 가/감속하기 때문인 것으로 보인다.

Leaving Mode 알고리즘을 적용한 경우에는 가속도 변화가 저속 상황의 경우 0.2 m/s^2 에서 -0.1 m/s^2 , 고속의 경우 0.7 m/s^2 에서 -0.8 m/s^2 정도로 CACC에 비해 최대 40배의 차이를 나타내는 것을 볼 수 있다. 이처럼 Leaving mode 통해 가속도의 변화가 줄어 안정적인 주행이 가능함을 확인할 수 있다. 특히, 이탈 신호를 수신하는 것을 통해 Vehicle 1이 차선을 벗어난 시점부터 가속을 하는 CACC와는 다르게 미리 가속을 진행해 상대 거리가 줄어드는 모습을 볼 수 있다. 이는 V2V 통신을 통해 Vehicle 1의 이탈 여부와 Vehicle 2의 존재를 미리 인지하고 목표지점 및 최적 속도를 설정해서 주행하기 때문이다.

6. 결론

기존의 ADAS나 자율주행 기능들은 레이더, 카메라와 같은 환경 센서를 기반으로 작동하기 때문에 인지 범위 및 획득할 수 있는 정보의 종류가 제한적이며 이로 인해 기능이 작동될 수 있는 범위가 제한적이라는 문제를 안고 있다. 이를 극복하기 위해 주변 차량들과의 근거리 무선 통신으로 정보를 공유할 수 있는 V2V 통신을 활용하는 방안이 떠오르고 있다. 본 논문에서는 V2V 통신 및 환경 센서와의 센서퓨전을 통해 주변 차량의 정보를 획득하고, 이를 바탕으로 주변 차량과의 안정된 대열주행을 가능하게 하는 자율협력주행 알고리즘을 개발하였다.

자율협력주행 기술을 구현하려면 자차량의 위치와 함께 주변 차량의 위치도 알 수 있어야 하지만, 현재 양산 차량에 사용되고 있는 GPS는 오차가 수 미터까지 발생할 수 있다는 문제를 안고 있다. 이를 해결하기 위해 본 연구에서는 전방 차량의 위치를 보다 높은 정밀도로 추정할 수 있는 알고리즘을 개발하였다. V2V 통신을 통해 획득할 수 있는 전방 차량의 정보와 자차량의 센서 정보를 융합한 센서퓨전 로직을 구현하였으며 여기서 나오는 신호를 다시 확장 칼만 필터를 통과시켜, 오차 10

cm 수준의 전방 차량 좌푯값을 자율협력주행 알고리즘에 전달할 수 있게 하였다.

본 연구에서 제안하는 자율협력주행 알고리즘은 CACC mode, Merging mode, Leaving mode의 총 3가지 모드로 구성된다. CACC mode에서는 Sliding mode control 기법을 이용해 대열주행을 안정하게 유지하기 위한 제어를 구현하였다. Merging mode에서는 대열주행 도중 전방에 대열 합류 차량 발생 시, 합류 차량의 경로를 예측하고 거리 제어를 위한 목표 가속도를 생성하여 합류 차량과 자차량이 안전하게 새로운 대열을 형성할 수 있게 하였다. Leaving mode에서는 전방 차량이 대열을 이탈하는 경우 선행하는 차량과 새로운 대열을 형성할 때까지 안정성을 유지함과 동시에 과도한 가감속으로 승차감이 저하되지 않도록 알고리즘을 구현하였다.

성능 검증을 위해 다양한 시나리오 상황에 대해 제안한 자율협력주행 알고리즘에 대한 기존 ACC 알고리즘과 CACC 알고리즘과의 비교 평가를 수행하였다. 그 결과 본 연구에서 개발한 자율협력주행 알고리즘이 기존의 ACC나 CACC에 비해 선행 차량 추종 성능과 안정성을 크게 개선하였으며, 작은 가속도 변화로도 요구 거리 제어 성능을 확보하여 승차감 개선에도 크게 기여하는 것을 확인할 수 있었다.

후 기

본 연구는 산업통상자원부와 한국산업기술진흥원의 “새만금지역 상용차 자율주행 테스트베드 구축사업”(과제번호 P0013841)의 지원을 받아 수행된 연구임.

References

- 1) Samjong KPMG INSIGHT, New Changes Made by Autonomous Driving, <https://home.kpmg/kr/ko/home/insights/2020/02/insight69.html>, Vol.69, 2020.
- 2) J. H. Kwak and W. Y. Kim, “Virtual Contamination Lane Image and Video Generation Method for the Performance Evaluation of the Lane Departure Warning System,” Transactions of KSAE, Vol.24, No.6, pp.627-634, 2016.
- 3) Q. Deng, Heavy-Duty Vehicle Platooning: Modeling and Analysis, Ph. D. Dissertation, KTH Royal Institute of Technology, Stockholm, 2016.
- 4) X. Song, L. Chen, K. Wang and D. He, “Robust Time-Delay Feedback Control of Vehicular CACC Systems with Uncertain Dynamics,” Sensors, Vol.20, No.6, 2020.
- 5) Q. Xu, K. Hedrick, R. Sengupta and J. VanderWerf, “Effects of Vehicle-vehicle/Roadside-vehicle Communication on Adaptive Cruise Controlled Highway Systems,” Proceeding of IEEE 56th Vehicular Technology Conference, Vol.2, pp.1249-1253, 2002.
- 6) W. Ko and D. E. Chang, “Cooperative Adaptive Cruise Control Using Turn Signal for Smooth and Safe Cut-in,” Proceeding of 18th International Conference on Control, Automation and Systems (ICCAS), pp.807-812, 2018.
- 7) V. Milanés, S. E. Shladover, J. Spring, C. Nowakowski, H. Kawazoe and M. Nakamura, “Cooperative Adaptive Cruise Control in Real Traffic Situations,” IEEE Transactions on Intelligent Transportation Systems, Vol.15, No.1, pp.296-305, 2013.
- 8) M. McGurrin, “Vehicle Information Exchange Needs for Mobility Applications,” United States. Joint Program Office for Intelligent Transportation Systems, No. FHWA-JPO-12-021, 2012.
- 9) International Standard Organization, ISO 21717: 2018, <https://www.iso.org/standard/71471.html>, 2018.
- 10) C. S. Kim and K. S. Huh, “Development of a Pose and Position Estimation Algorithm with Considering the Error and Latency from the GPS and In-vehicle Sensors,” Transactions of KSAE, Vol.26, No.5, pp. 620-629, 2018.
- 11) R. Prophet, J. Martinez, J. C. F. Michel, R. Ebel, I. Weber and M. Vossiek, “Instantaneous Ghost Detection Identification in Automotive Scenarios,” 2019 IEEE Radar Conference (RadarConf), 2019.
- 12) S. R. Choi, Y. H. Jeong, M. S. Lee and K. S. Yi, “Radar, Vision, Lidar Fusion-based Environment Sensor Fault Detection Algorithm for Automated Vehicles,” Korean Auto-vehicle Safety Association (KASA), Vol.9, No.4, pp.32-37, 2017.
- 13) S. W. Kim, S. H. Park and J. Y. Ryu, “Fabrication and Characteristic of Active Wheel Speed Sensor,” The Journal of Korean Institute of Information Technology, Vol.10, No.10, pp.157-164, 2012.
- 14) A. Dadashnialehi, A. Bab-Hadiashar, Z. Cao and A. Kapoor, “Accurate Wheel Speed Measurement for Sensorless ABS in Electric Vehicle,” 2012 IEEE International Conference on Vehicular Electronics and Safety (ICVES 2012), IEEE, 2012.
- 15) K. N. Kim, J. H. Lee, J. K. Ok, W. S. Yoo and J. Y. Park, “Friction Coefficient of Emergency Braking on ABS and Non-ABS Car,” Transactions of KSAE, Vol.16, No.3, pp.52-59, 2008.
- 16) S. P. Kim, Understanding Kalman Filter Using MATLAB in Korean Sub, A-JIN, Seoul, pp.39-208, 2010.

- 17) M. Lin, J. Yoon and B. Kim, "Self-Driving Car Location Estimation Based on a Particle-Aided Unscented Kalman Filter," *Sensors*, Vol.20, No.9, 2020.
- 18) I. K. Nikolos, A. I. Delis and M. Papageorgiou, "Macroscopic Modelling and Simulation of ACC and CACC Traffic," 2015 IEEE 18th International Conference on Intelligent Transportation Systems, 2015.
- 19) R. Schubert, C. Adam, M. Obst, N. Mattern, V. Leonhardt and G. Wanielik, "Empirical Evaluation of Vehicular Models for Ego Motion Estimation," In 2011 IEEE Intelligent Vehicles Symposium (IV), pp.534-539, 2011.
- 20) R. Schubert, E. Richter and G. Wanielik, "Comparison and Evaluation of Advanced Motion Models for Vehicle Tracking," In 2008 11th International Conference on Information Fusion, pp.1-6, 2008.
- 21) T. Toledo and D. Zohar, "Modeling Duration of Lane Changes," *Transportation Research Record*, Vol.1999, No.1, pp.71-78, 2007.
- 22) R. Shadmehr and S. P. Wise, *The Computational Neurobiology of Reaching and Pointing: A Foundation for Motor Learning*, MIT Press, Cambridge, 2005.
- 23) S. M. Joo, J. H. Lee and J. K. Seo, "Minimum-Jerk Trajectory Generation for Control Applications," *Transactions of the Korean Nuclear Society Autumn Meeting*, 2018.
- 24) W. B. Na, J. W. Kim and H. C. Lee, "Study on TTC-based Optimal Lane Change Algorithm in Adaptive Cruise Control," *Transactions of KSAE*, Vol.27, No.8, pp.627-636, 2019.