

동적 및 정적 물체 회피를 위한 정밀 도로지도 기반 지역 경로 계획

High-Definition Map-based Local Path Planning for Dynamic and Static Obstacle Avoidance

정의곤¹·송원호²·명현[†]
Euigon Jung¹, Wonho Song², Hyun Myung[†]

Abstract: Unlike a typical small-sized robot navigating in a free space, an autonomous vehicle has to travel in a designated road which has lanes to follow and traffic rules to obey. High-Definition (HD) maps, which include road markings, traffic signs, and traffic lights with high location accuracy, can help an autonomous vehicle avoid the need to detect such challenging road surroundings. With space constraints and a pre-built HD map, a new type of path planning algorithm can be conceived as a substitute for conventional grid-based path planning algorithms, which require substantial planning time to cover large-scale free space. In this paper, we propose an obstacle-avoiding, cost-based planning algorithm in a continuous space that aims to pursue a globally-planned path with the help of HD map information. Experimentally, the proposed algorithm is shown to outperform other state-of-the-art path planning algorithms in terms of computation complexity in a typical urban road setting, thereby achieving real-time performance and safe avoidance of obstacles.

Keywords: Autonomous Driving, Path Planning, High-Definition (HD) Map

1. 서 론

자율주행을 위해서는 주변 환경을 인지하여 차량을 제어해 출발지부터 목적지까지 안전하게 주행하여야 한다. 모바일 로봇 공학에서 경로 계획은 장애물을 피할 수 있도록 모바일 로봇이 목표를 안전하게 달성하는 방법을 결정하는 전략을 말한다¹⁾. 경로 계획 및 모션 제어를 성공적으로 달성하려면 자율주행 로봇이 피해야 할 장애물을 감지하여 안전하게 목표에 도달할 수 있어야 한다. 이러한 모바일 로봇과 다르게 자율주행 차량은 차

선과 도로 환경을 고려하며 도로 교통법규를 위반하지 않는 경로 계획이 필요하다. 이를 위하여 자율주행차에서는 일반적으로 센티미터 단위의 정확도를 가지는 정밀 지도를 기반으로 차선 정보 및 차량 주변 정보를 이용하여 경로 생성을 하고자 한다.

경로 생성에는 전역 경로 계획과 지역 경로 계획이 있다²⁾. 전역 경로 계획은 정밀지도에서 사용 가능한 모든 정보를 사용하는 반면, 지역 경로 계획은 차량 주변만을 고려한다. 지역 경로 계획은 예기치 못한 동적 및 정적 장애물을 회피하면서 전역 경로 계획을 최대한 따라가도록 하는 경로를 찾아내는 것을 목표로 한다. 정밀지도를 활용한 자율주행 시스템은 목적지까지의 전역 경로가 있는 상황에서 해당 경로를 최대한 따라가면서 차량 주변의 인식된 장애물들을 회피하는 것을 목표로 한다. 본 논문에서는 지역 경로 계획 부분에 대한 연구를 다루고자 한다.

지역 경로 계획 기술은 오랜 시간 동안 연구된 분야이다. 대표적인 지역 경로 계획 알고리즘으로는 Dynamic Window Approach (DWA)³⁾와 Vector Field Histogram (VFH)⁴⁾ 알고리즘 등이 있다.

DWA는 속도 기반 지역 경로 계획 알고리즘으로서 목적지까지 장애물이 없는 최적의 속도를 계산한다. 특징으로는 (x, y)

Received : Feb. 10. 2021; Revised : Mar. 18. 2021; Accepted : Mar. 19. 2021

※ This work was supported by the research project "Development of A.I. based recognition, judgement and control solution for autonomous vehicle corresponding to an atypical driving environment," which is financed from the Institute for Information and Communications Technology Planning & Evaluation (Republic of Korea) and the Ministry of Science and ICT (Republic of Korea) Contract No. 2019-0-00399. The students are supported by the BK21 FOUR from the Ministry of Education (Republic of Korea)

1. Master's Candidate, KAIST, Daejeon, Korea (egjung94@kaist.ac.kr)

2. PhD Candidate, KAIST, Daejeon, Korea (sw4613@kaist.ac.kr)

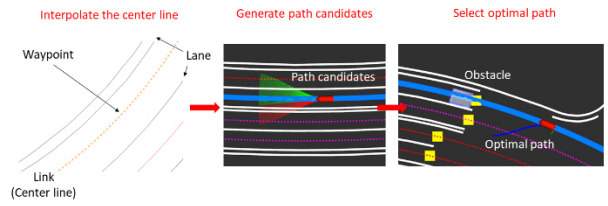
† Professor, Corresponding author: School of Electrical Engineering, KI-AI, KI-R, KAIST, Daejeon, Korea (hmyung@kaist.ac.kr)

직교좌표에서 (v, w) 속도계 좌표로 변환시켜 계산을 한다는 것이다. v 는 직진 속도(Translational velocity)이며, w 는 회전 속도(Angular velocity)이다. 로봇의 최대 가속도, 최소 가속도, 최대 각속도, 최소 각속도를 고려한 윈도우 내에 들어오면서 장애물과 충돌하지 않는 (v, w) 쌍 중 최적의 평가 함수를 가지는 쌍을 선택한다. Global Dynamic Window Approach (GDWA)는 각 점유 격자(Occupancy grid)로부터 목적지까지의 거리를 고려하여 경로 계획을 실행한다. 하지만, 전역 경로를 따라가기 위해 이산화된(Discretized) 모든 그리드에 대해 계산을 하는 것은 비효율적이다. DWA와 GDWA 모두 경로의 부드러움(Smoothness)을 반영하지 않았기 때문에 운전자 및 승객의 승차감을 감소시킬 수 있다. VFH는 극좌표 히스토그램(Polar histogram)을 만들어 주변 환경을 각 섹터별로 분할하여 나타내고, 섹터별 장애물의 분포도에 따라 각도, 직진 속도와 조향(Steering)을 결정한다. VFH+^[5]는 경로의 부드러움과 안정성을 개선했으며, 로봇의 크기를 고려하여 비용 함수 기반 경로 생성하였다. VFH*^[6]는 A* 알고리즘을 병합하여 기존 VFH와 VFH+가 가진 지역적인 경로 생성에 따른 문제를 개선하였다. VFH와 DWA에서 파생된 알고리즘들은 최종 목적지에 대한 조향을 조정하지만, 전역 경로 계획을 추종하는 요소가 평가 함수에 포함되지 않았기에 장애물 회피 이후 전역 경로를 벗어나 버릴 수 있다. 특히 교통 법규와 차선 등을 지켜야 하는 자동차의 경우에는 장애물 회피 후 전역 경로로 돌아오도록 운행해야 한다.

본 논문은 자율 자동차의 특성을 고려하여 전역 경로 계획을 추종하는 지역 경로를 생성하는 알고리즘을 제안한다. 본 논문의 기여는 다음과 같다. 첫째, 전역 계획 추종에 관련된 비용 함수를 정의하여 목적지 지향적이 아닌 전역 경로 지향적으로 수행하도록 하였다. 둘째, 동적 물체에 대해 끼어들기(Cut-in)와 따라가기(Following)를 하는 경로를 계획하여 현실적인 주행을 하도록 하였다. 셋째, 다양한 상황에서 장애물 회피가 가능한 경로를 생성하도록 하여 안정성을 높였다.

2. 지역경로 생성

본 논문에서 제안하는 방법은 전역 계획 경로를 추종하며 정밀 도로지도를 바탕으로 안전하고 부드러운 지역 경로를 생성하고 그에 해당하는 적절한 가속도를 계산한다. 경로 후보군 생성과 비용 함수 기반 최적 경로 및 속도를 결정하는 방법은 기존 논문^[7,8]을 참고하였으나, 전역 경로를 추종하는 비용 함수를 추가하였다. 또한, 동적 물체에 대한 비용 함수를 변경하여 끼어들기와 따라가기에 대해 반영하였다. 마지막으로, 장애물 회피 시의 예외적인 상황이 없도록 경로를 생성하여 안정성을 높였다. 전역 경로는 정밀 도로지도의 차선과 중앙



[Fig. 1] Overview of cost-based spline path planning algorithm

선의 경로점(Waypoint)의 구조적 관계를 바탕으로 이미 생성이 되었다는 가정으로 실험을 하였다.

[Fig. 1]은 비용함수 기반 3차식 곡선 경로를 생성하는 방법을 크게 3단계로 나눠 보여준다. 1) 전역 경로를 호 길이(Arc length)로 매개변수화(Parametrization)한다. 2) 경로 후보군을 생성한다. 3) 경로 후보군 중 최적의 경로를 선택한다.

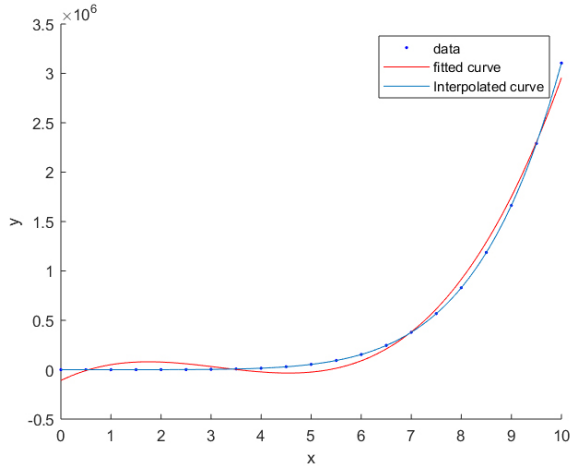
맨 처음으로 전역 경로의 경로점들을 3차식 스플라인 곡선(Cubic spline curve)으로 보간(Interpolate)한다. 경로 후보군은 현재 차량의 위치, 속도, 방향, 그리고 전역 경로에서부터 횡방향 차이를 두어 여러 개의 3차식 스플라인 곡선들을 생성한다. 그 후, 후보군 중 정적·동적 물체와의 충돌, 운행의 부드러움, 전역 경로의 추종 여부를 고려하여 선택한다. 마지막으로 후보군 결정 후 해당 경로에 대한 속도를 결정한다.

2.1 전역 경로 호 길이 매개변수화

전역 경로의 경로점들은 정밀 도로지도를 기반으로 생성되었다고 가정하였다. 정밀 도로지도의 경로점은 촘촘하고 높은 정확도로 표현되고 있기 때문에 곡선 피팅(Fitting)보다는 곡선 보간법(Interpolation)을 사용하였다. 곡선 보간법을 사용할 시 [Fig. 2]에서와 같이 추종 오차를 줄일 수 있다. 도로 환경에서의 위치는 호의 길이에 대해 매개변수화하여 쉽게 표현할 수 있다^[8]. 따라서, 전역 경로에 해당하는 각 경로점 w 의 x 와 y 좌표를 식 (1)으로 표현할 수 있다^[9,10].

$$\begin{aligned} w(s) &= (x(s), y(s)), \quad 0 \leq s \leq L & (1) \\ x(s) &= a_{x_i}(s-s_i)^3 + b_{x_i}(s-s_i)^2 + c_{x_i}(s-s_i) + d_{x_i} \\ y(s) &= a_{y_i}(s-s_i)^3 + b_{y_i}(s-s_i)^2 + c_{y_i}(s-s_i) + d_{y_i} \end{aligned}$$

s 는 호의 길이이며, L 은 전역 경로의 전체 호의 길이이다. $x(s)$ 와 $y(s)$ 는 $\{s_0, s_1, \dots, s_n\}$ 까지의 호 마디(Arc segment)를 표현하는 3차식 스플라인 방정식이다. $s_0 = 0$ 이고, $s_n = L$ 이다. d_{x_i} 와 d_{y_i} 는 각 호 마디의 시작점인 s_i 에서의 x 와 y 좌표이다. 따라서, 식 (1)을 통해 전역 경로 위의 모든 위치에 대해 호의 길이를 식에 대입하여 보간된 x 와 y 좌표를 얻을 수 있다. 또한, 그 전역 경로점에서의 접선의 각도와 곡률(Curvature)은 각각 식 (2) 및 식 (3)과 같이 계산할 수 있다^[11].



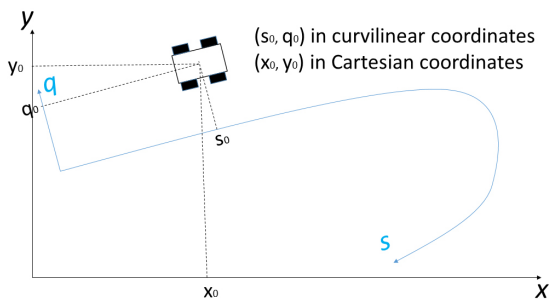
[Fig. 2] Comparison between the interpolated curve and the fitted curve based on sample data points

$$\theta = \tan^{-1}\left(\frac{dy}{dx}\right) \quad (2)$$

$$K_{global} = \frac{\left(\frac{dx}{ds}\right)\left(\frac{d^2y}{ds^2}\right) - \left(\frac{d^2x}{ds^2}\right)\left(\frac{dy}{ds}\right)}{\left(\frac{dx}{ds} + \frac{dy}{ds}\right)^{1.5}} \quad (3)$$

2.2 전역 경로 상에서의 위치 파악

현재 차량의 위치를 전역 경로에 대한 호 길이와 횡방향 차이로 나타내면 도로 상에서의 위치를 좀더 쉽게 표현할 수 있다. 이를 위해 모든 위치를 직교 좌표계(The Cartesian coordinates)에서 곡선 좌표계(Curvilinear coordinates)로 [Fig. 3]과 같이 변환한다. 차량의 위치에서부터 중앙선까지의 가장 가까운 거리 q_0 와 해당 호의 길이 s_0 를 2차 최소화(quadratic minimization)와 뉴턴 방법(Newton's method)를 사용해서 구할 수 있다⁹⁾. 하지만, 제안된 방법은 거리만 계산함으로 중앙선으로부터의 위치를 표현하기에 충분하지 않다. 따라서, 차량의 진행방향 기준



[Fig. 3] Conversion from the Cartesian coordinates to curvilinear coordinates

[Algorithm 1] Signed lateral distance from center line

1. **Input:** arc length on the center line s_0 , closest distance from a vehicle to center line q_0 , vehicle position in the Cartesian coordinates (x_0, y_0)
2. Compute $x(s_0), y(s_0)$ from equation (1) where $s_i \leq s_c \leq s_f$
3. Compute $x(s_0 + \Delta s), y(s_0 + \Delta s)$ from equation (1) where $\Delta s \ll (s_{i+1} - s_i)$
4. Compute
 $\Delta x = x(s_0 + \Delta s) - x(s_0),$
 $\Delta y = y(s_0 + \Delta s) - y(s_0),$
 $m_0 = \frac{\Delta y}{\Delta x},$
5. **if** $\Delta x > 0 \ \&\& \ \Delta y > 0$ or $\Delta x > 0 \ \&\& \ \Delta y < 0$, i.e. first or fourth quadrant **then**
6. **if** $y_0 - y(s_0) < m_0(x_0 - x(s_0))$ **then**
7. return negative distance $-q_0$
8. **else**
9. return positive distance $+q_0$
10. **else**, i.e. second or third quadrant
11. **if** $y_0 - y(s_0) < m_0(x_0 - x(s_0))$ **then**
12. return positive distance $+q_0$
13. **else**
14. return negative distance $-q_0$

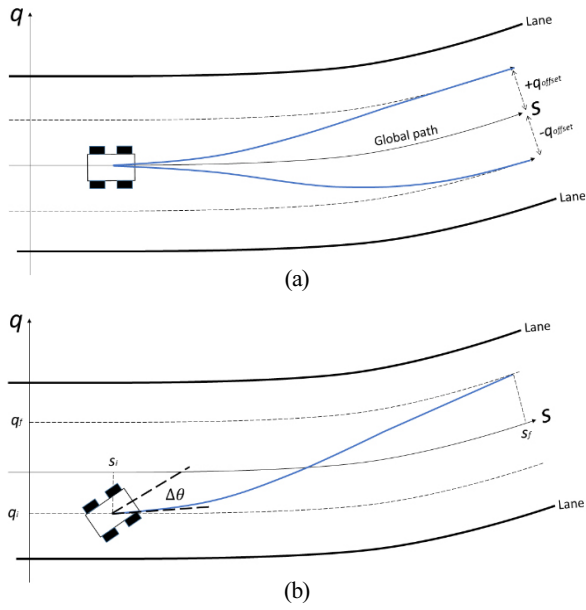
으로 현재 중앙선으로부터 우측에 있을 때는 음수 횡방향 차이($-q_0$)로, 좌측에 있을 때는 양수($+q_0$)로 [Algorithm 1]을 통해 판단할 수 있다.

2.3 경로 후보군 생성

경로 후보군은 전역 경로를 바탕으로 횡방향 차이(Lateral offset)을 주어 생성한다. [Fig. 4(a)]에서와 같이 경로의 끝 점이 각 $+q_{offset}$ 과 $-q_{offset}$ 의 횡방향 차이를 가진 연속적인 경로 후보군을 생성할 수 있다. 경로 후보군은 3차 스플라인 곡선으로 s 에 대한 q 의 식으로 식 (4)와 같이 나타낼 수 있다.

$$q(s) = \begin{cases} a(s - s_i)^3 + b(s - s_i)^2 + c(s - s_i) + q_i, & \text{if } s_i \leq s < s_f \\ q_f, & \text{if } s = s_f \end{cases} \quad (4)$$

식 (4)에서 3차식의 계수 a, b, c 에 따라 후보군을 생성하기 위해서 경계 조건(Boundary conditions)을 설정할 수 있다. [Fig. 4(b)]에서 현재 차량의 s_i 와 q_i 은 2.2절을 참고하여 계산할 수 있다. $\Delta\theta$ 는 차량의 방향과 차량 위치에 해당하는 전역 경로의 접선의 각과의 차이이다. $\Delta s = s_f - s_i$ 로 경로 후보군의 호 길이를, q_f 와 q_i 로 전역 횡방향 차이를 결정할 수 있다. 경계 조건은 식 (5)로 나타낼 수 있다.



[Fig. 4] Path candidate generation with respect to the global path. (a) Generated paths for different lateral offsets, (b) boundary conditions for path generation

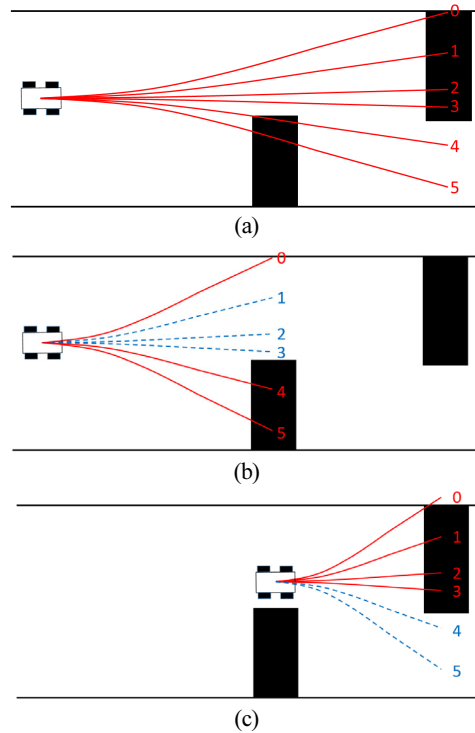
$$\begin{cases} q(s_i) = q_i, \\ q(s_f) = q_f, \\ \frac{dq}{ds}(s_i) = \tan(\Delta\theta), \\ \frac{dq}{ds}(s_f) = 0 \end{cases} \quad (5)$$

식 (5)에서의 경계 조건식을 바탕으로 경로 후보군의 3차식 계수들을 결정할 수 있다. 각 경로 후보군은 식 (2)와 식 (3)을 바탕으로 곡선 좌표계에서 직교좌표계로 변환하여 차량 제어에 적용할 수 있다¹²⁾.

경로를 생성할 때 안정성은 중요한 고려사항이다. 안정적인 주행을 위해서는 차량의 주행 속도가 빠를 때는 경로가 완만해야 하며, 주행 속도가 느릴 때는 경로가 급격하더라도 안정성에 영향이 적다. 경로 후보군은 같은 횡방향 차이를 가정했을 때 경로의 호 길이 Δs 에 따라 경로의 가파름이 결정된다. 따라서, 식 (6)과 같이 차량의 속도에 따라 경로의 호 길이가 결정되어야 한다.

$$\Delta s_{vel} = \begin{cases} \Delta s_{min} + \frac{v^2}{|a_{min}|}, & \text{if } \Delta s_{min} + \frac{v^2}{|a_{min}|} < \Delta s_{max} \\ \Delta s_{max}, & \text{otherwise} \end{cases} \quad (6)$$

Δs_{min} , Δs_{max} 은 횡방향 차이를 위한 최소 및 최대 호 길이이다. v 는 차량의 현재 속도이며, a_{min} 은 차량의 최소 가속도이다. Δs_{min} 과 Δs_{max} 는 실험하며 조정이 필요한 변수이다.



[Fig. 5] Avoidance problems and solutions to an exceptional occasion. (a) No available paths (solid), (b) available paths (dashed) for avoiding the first obstacle, (c) available paths (dashed) for avoiding the second obstacle

하지만, 차량의 속도로만 횡방향의 차이를 결정하게 되면 물체 회피에서 심각한 문제를 직면할 수 있다. 경로 상에 정적 물체가 [Fig. 5(a)]와 같이 놓여 있고 차량이 빠른 속도로 주행 시 모든 경로 후보군은 장애물과 충돌한다고 판단된다. 이런 예외적인 상황에서만 경로의 호 길이를 줄이는 방법을 사용할 수 있으나⁸⁾, 휴리스틱(Heuristic)하며 일반화가 힘든 방법론이다. 따라서, 안정성이 감소하더라도 장애물을 피할 수 있도록 호 길이를 조정해야 한다.

[Algorithm 2]는 장애물의 여부에 따라 경로의 호 길이 값을 결정하는 과정을 나타낸다. 센서를 통해 판단된 물체 Obj 중 차선, 도로경계석 등 정밀 도로 지도에 맵핑된 물체가 아닌 것을 장애물이라 판단한다. 현재 차량으로부터 각 장애물까지의 호 길이의 차이를 구한다. 만약 호 길이의 차이가 양수라면, 장애물이 차량의 진행 방향에 위치하여 피해야 할 대상이므로 그 값을 δ 에 넣는다. δ 중 최소 장애물과의 호 길이 차이 Δs_{obs} 을 찾아내고, 경로 생성의 호의 길이를 Δs_{obs} 과 Δs_{min} 중 작은 것으로 결정한다. 만약 δ 가 비어있다면, Δs_{vel} 를 따른다.

[Fig. 5(b)]와 [Fig. 5(c)]는 개선된 알고리즘으로 경로 생성 시 [Fig. 5(a)]와 같은 상황을 피할 수 있는 것을 보여준다. 물체를 감지하는 순간부터 회피하는 경로를 생성할 수 있으므로 회피의 안정성이 증가한다.

[Algorithm 2] Arc-length determination for obstacle avoidance

1. **Input:** $\text{obj} = (\text{obj}_1, \text{obj}_2, \dots, \text{obj}_m) \in \text{Obj}$,
 $\delta = \{\}$
2. **Output:** arc length Δs for path generation
3. **Condition:**
 $\text{Avoid}(\text{obj}_i) = \begin{cases} 1, & \text{if } \text{obj}_i \text{ is an obstacle} \\ 0, & \text{otherwise} \end{cases}$
4. **for each** $\text{obj}_i \in \text{Obj}$
5. **if** $\text{Avoid}(\text{obj}_i) = 1$ **then**
6. Compute $\Delta s_{\text{obs}_i} = s_{\text{obj}_i} - s_{\text{veh}}$
7. **if** $\Delta s_{\text{obs}_i} > 0$ **then**
8. Insert Δs_{obs_i} into δ
9. **if** δ is empty **then**
10. Return Δs_{vel}
11. **else**
12. Find minimum arc-length-distanced obstacle
 $s_{\text{obs}} = \min(\delta)$
13. Return $\Delta s = \min\{\Delta s_{\text{obs}}, \Delta s_{\text{min}}\}$

2.4 최적 경로 선택

경로 후보군 중 비용 함수가 가장 작은 경로를 최적의 경로로 선택한다. 비용 함수는 식 (7)과 같이 정적 물체 비용(Static object cost)^[7] C_s , 부드러움 비용(Smoothness cost)^[8] C_{sm} , 전역 경로 추종 비용(Global-path-following cost) C_g , 동적 물체 비용(Dynamic cost) C_d 로 구성되어 있으며, w_s, w_{sm}, w_g, w_d 로 각각 비용의 중요도를 조정할 수 있다. i 는 각 경로 후보에 해당하는 인덱스이다.

$$C_{\text{total}}[i] = w_s C_s[i] + w_{sm} C_{sm}[i] + w_g C_g[i] + w_d C_d[i] \quad (7)$$

정적 물체 비용은 각 경로 후보와 장애물과의 충돌 여부를 0과 1로 표현한 후, 가우시안 곱셈합(Gaussian convolution)을 하여 장애물과 멀리 있을수록 비용을 낮게 결정한다^[8]. 차선, 도로경계선, 가드레일 등도 장애물로 간주하기 때문에 정밀 도로지도의 정보를 사용하여 비용을 계산한다.

부드러움 비용은 횡방향 가속도를 최소화하는 경로 후보를 고르기 위해 사용된다. 차량의 속도가 일정하다고 가정했을 때, 곡률은 순간 구심 가속도에 비례한다. 따라서, 곡률의 제곱을 경로 후보에 대해 적분을 하여 부드러움 비용을 정의하며, 식 (8)과 같이 나타낼 수 있다^[8].

$$C_{sm}[i] = \int k_i^2(s) ds \quad (8)$$

전역 경로 추종 비용은 전역 경로로부터의 횡방향 차이가 가장 작은 후보를 고르기 위해 사용된다. 경로 후보군들의 횡방향 차이의 합을 각각 후보의 횡방향 차이에서 나눠주어 비용 함수를 0과 1 사이의 값으로 정규화할 수 있으며, 식 (9)와 같이 나타낼 수 있다.

$$C_g[i] = \frac{|q(s_{\text{veh}} + \Delta s)[i]|}{\sum_j |q(s_{\text{veh}} + \Delta s)[j]|} \quad (9)$$

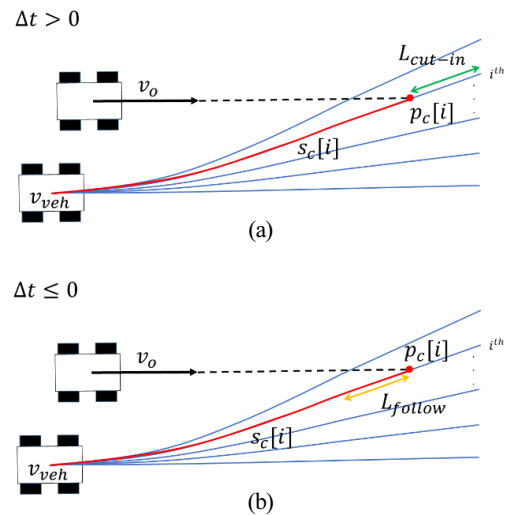
$q(\cdot)[i]$ 는 i 번째 경로 후보의 식 (4)와 같은 3차 스플라인 곡선이다. $q(s_{\text{veh}} + \Delta s)[i]$ 는 해당 경로 후보의 횡방향 차이로서, 식 (4)에서 s_i 을 전역 경로상에서의 차량의 현재 호 위치 s_{veh} 로 변경하여 대입한 값이다.

동적 물체 비용은 동적 물체의 속도와 진행 방향을 고려하여 최적의 경로를 선택하기 위해 사용된다^[7]. [Fig. 6]에서 동적 물체는 \vec{v}_o 속도 벡터로 주행 중이며, $t_{\text{obs}}[i]$ 초 후에 i 번째 경로 상의 $p_c[i]$ 위치에 도달한다. 현재 차량의 위치에서부터 도달 지점 $p_c[i]$ 까지의 경로 거리가 $s_c[i]$ 이고, 현재 차량의 속도가 v_{veh} 라면 식 (10)으로 충돌까지 걸리는 시간 $t_{\text{veh}}[i]$ 를 계산할 수 있다.

$$t_{\text{veh}}[i] = \frac{s_c[i]}{v_{\text{veh}}} \quad (10)$$

$$\Delta t[i] = t_{\text{obs}}[i] - t_{\text{veh}}[i] \quad (11)$$

경로와의 충돌 시간과 차량의 충돌 지점까지 주행하기까지 걸리는 시간차 $\Delta t[i]$ 를 바탕으로 끼어들지 따라갈지를 결정



[Fig. 6] Example of a dynamic obstacle collision with a generated path. (a) Cutting in front of the moving obstacle by $L_{\text{cut-in}}$, (b) following the moving obstacle by L_{follow}

한다. 차량이 현재 속도를 유지할 때, Δt 가 양수이면 끼어들기가 가능하고, Δt 가 음수이면 따라가기가 가능하다. $\Delta t = 0$ 일 때는 따라가도록 하였다. [Fig. 6(a)]는 i 번째 경로로 끼어들기가 가능한 상황에 대한 예시이다. i 번째 경로로 L_{cut-in} 의 거리를 앞서 끼어들기로 하였을 때의 필요한 가속도는 식 (12)로 구할 수 있다. 해당 경로의 동적 물체 비용은 끼어들기 위해 필요한 가속도와 주행 거리에 대한 식 (13)으로 계산할 수 있다. 식 (12)에서 $s_c[i] + L_{cut-in} - v_{veh} t_{obs}[i] \leq 0$ 일 경우 현재 속도로 충분히 끼어들기가 가능하므로 가속도는 0이다. 그렇지 않을 경우에는 기본적인 운동학 방정식을 통해 가속도를 구할 수 있다.

$$a[i] = \begin{cases} 0, & \text{if } s_c[i] + L_{cut-in} - v_{veh} t_{obs}[i] \leq 0 \\ 2 \left(\frac{s_c[i] + L_{cut-in} - v_{veh} t_{obs}[i]}{t_{obs}[i]^2} \right), & \text{otherwise} \end{cases} \quad (12)$$

$$C_d[i] = |a[i]| (s_c[i] + L_{cut-in}) \quad (13)$$

[Fig. 6(b)]는 i 번째 경로로 따라가기가 가능한 상황에 대한 예시이다. i 번째 경로로 L_o 의 거리를 두고 따라가기로 하였을 때, $s_c[i]$ 를 고려한 L_{follow} 를 식 (14)로 구할 수 있다. L_{follow} 거리 차를 두기 위한 가속도는 식 (15)로 구할 수 있다. 해당 경로의 동적 물체 비용은 식 (16)으로 나타낼 수 있다.

$$L_{follow} = \begin{cases} L_o, & \text{if } L_o \leq s_c[i] \\ s_c[i], & \text{otherwise} \end{cases} \quad (14)$$

$$a[i] = 2 \left(\frac{s_c[i] - L_{follow} - v_{veh} t_{obs}[i]}{t_{obs}[i]^2} \right) \quad (15)$$

$$C_d[i] = |a[i]| (s_c[i] - L_{follow}) \quad (16)$$

2.5 목표 속도 설정

목표 속도는 도로에서 지정한 최고 속도 v_{limit} , 선택된 경로의 곡률에 따른 속도 v_κ , 정적 물체 비용에 따른 속도 v_s 중 가장 느린 속도로 정한다⁷⁾. 곡률에 대한 속도는 식 (17), 동적 물체에 대한 속도는 식 (18)과 같이 나타낼 수 있다.

$$v_\kappa = \sqrt{\frac{|a_{lat}|_{max}}{\max \kappa(\text{path})}} \quad (17)$$

$$v_s = (1 - k_s C_s^2(\text{path})) v_{ref} \quad (18)$$

$|a_{lat}|_{max}$ 는 최대 횡방향 가속도로서 주행 스타일에 따라 변 가능한 변수이며, $\max \kappa(\text{path})$ 는 선정된 경로의 최대 곡률이다. $C_s(\text{path})$ 는 선정된 경로의 정적 물체 비용이며, 장애

물에 둘러싸인 좁은 공간일 경우 느린 속도로 주행하기 위해 속도 선정에 사용된다. k_s 는 정적 물체 비용에 대한 증폭값이고, v_{ref} 는 선정된 경로에 대한 기준 속도로서, 둘 다 실험을 통해 설정할 수 있는 변수이다.

3. 실험 및 결과

본 논문의 실험은 ROS (Robot Operation System) 환경에서 진행하였고, 시뮬레이션 결과는 ROS의 RVIZ 툴을 사용하여 시각화하였다. 차량 및 동적 물체 주행은 MPC (Model Predictive Control)¹³⁾를 사용하여 경로 추종을 하도록 하였다. MPC는 모델링된 시스템을 기반으로 미래의 특정 구간 동안의 출력을 예측하여 최적화하는 방법이다¹⁴⁾. MPC를 이용한 경로 추종은 제약 조건 내에서 전방 이동할 경로점에 대한 비용 함수를 줄이는 제어 값을 계산하여 경로를 추종한다¹⁵⁾. 비용 함수로는 각 전방 경로점에서의 속도와 기준 속도(Reference velocity)의 차이, 조향각, 가속도, 그리고 연속되는 두 경로점의 조향각과 가속도의 변화를 기반으로 정의되고, 비용 함수를 최소화하는 각 경로점의 조향각과 가속도를 산출한다¹⁶⁾. 제어기의 제약 조건으로는 2.4절에서 설명한 따라가기-끼어들기를 위한 가속도와 2.5절에서 설명한 목표 속도를 사용하였다.

해당 실험은 국토정보플랫폼에서 제공하는 2019년도 대구 자율주행특화지구 정밀 도로지도를 바탕으로 진행하였다. 시뮬레이션에서 빨간색 직사각형은 경로 계획을 시행하는 차량이며, 빨간 점선은 차량의 궤도를 나타낸다. 검정색 선은 정밀 도로지도 기반의 차선이다. 차선 위 초록색 점선은 이미 계획된 전역 경로이며, 이에 따라 경로 후보군을 생성한다. 경로 후보군 중 빨간색 선은 충돌 위험에 의해 제외된 경로이며, 초록색 선은 주행 가능한 경로이다. 그중 파란색 선은 선택된 최적의 경로로서 MPC로 추종하는 경로이다. 정적 및 동적 장애물은 짙은 회색 타원으로 나타냈으며, 차량의 크기를 고려하여 옅은 회색의 버퍼(Buffer)를 만들었다. 차량의 최고 속도는 정밀 도로지도 기반으로 현재 차량이 위치하고 있는 도로에 따라 설정하였으며, 실험한 도로의 최고 속도는 모두 50 km/h이다. 차량의 최대 가속도와 최소 가속도는 각각 1 m/s²와 -3 m/s²으로 설정하였다. [Table 1]은 실험에 사용된 파라미터 값이다.

[Table 1] Parameter values for simulation

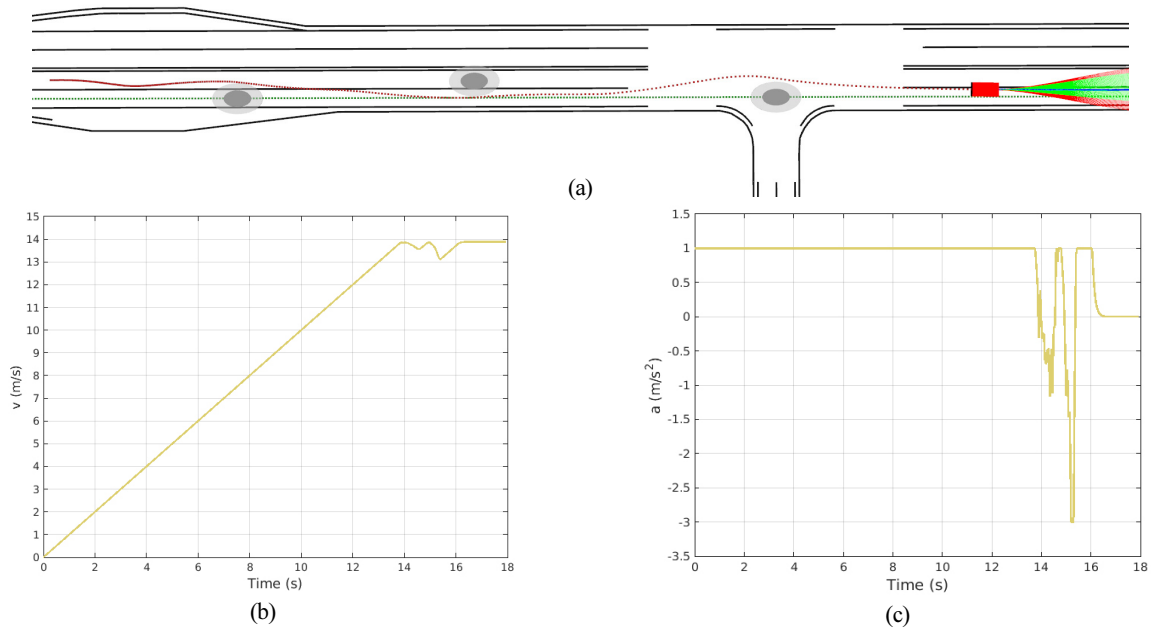
$ a_{lat} _{max}$	5.0 m/s ²
k_s	0.8
Δs_{min}	10.0 m
Δs_{max}	50.0 m
v_{ref}	50 km/h

3.1 정적 물체 회피

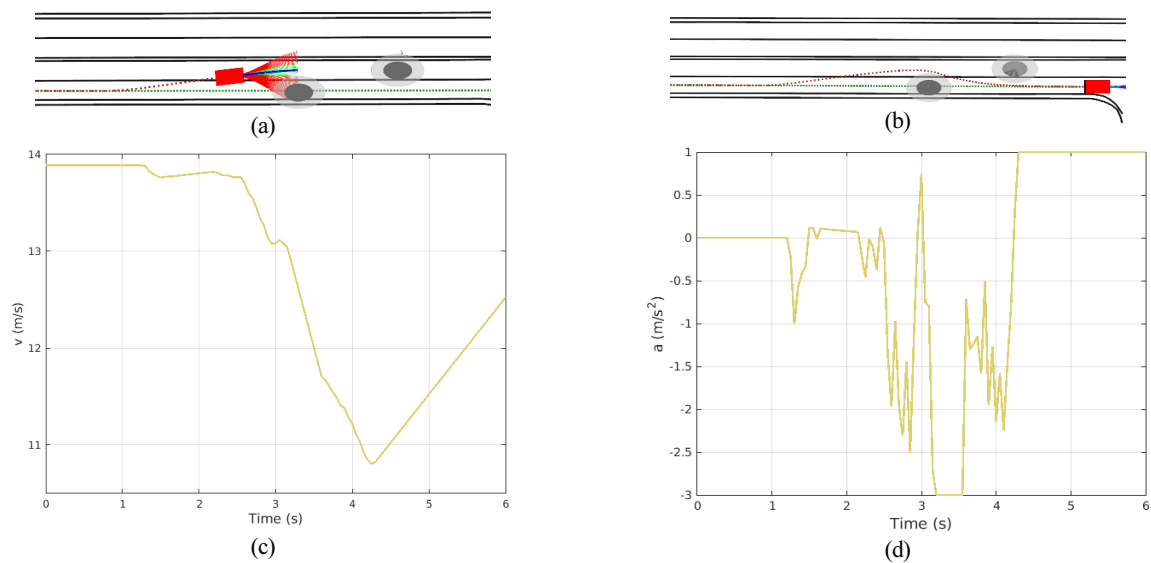
[Fig. 7]은 여러 개의 정적 물체를 회피하며 전역 경로를 추종한 결과이다. 시작 속도는 0 m/s에서 시작하기 때문에 최대 가속도로 주행을 시작한다. 전역 경로 추종 비용에 의해 전역 경로를 추종하는 경로로 주행 중 전역 경로 상에 장애물을 확인하고 회피하는 것을 볼 수 있다. 세 번째 장애물을 회피하기 전까지는 차량이 목표 속도에 도달하지 않아 최대 가속도로 회피를 하였다. 하지만, 세 번째 장애물 회피 시에는 전역 경로를 벗어날 때와

회피 후 전역 경로로 돌아올 때 감속하는 것을 보여준다.

[Fig. 8]은 [Fig. 5(a)]에서 제기한 문제를 해결한 회피 경로 생성을 보여준다. [Fig. 8(a)]에서 [Algorithm 2]에 따라 생성된 경로 후보군의 길이는 가장 가까운 장애물까지의 거리로 설정되어 두 번째 장애물에 의해 모든 경로가 막히는 문제를 피할 수 있다^[8]. 또한, 장애물을 미리 피할 수 있도록 경로가 생성되므로 급격한 회피를 방지할 수 있다. -1 m/s^2 의 감속으로도 첫 번째 장애물은 안전하게 피할 수 있고, 두 번째 장애물의 갑작스러운 출현도 대처하는 것을 보여준다.



[Fig. 7] Static obstacle avoidance. (a) Trajectory of the ego vehicle avoiding three static obstacles, (b) the speed of the ego vehicle, (c) the acceleration of the ego vehicle



[Fig. 8] Avoidance of two closely-located obstacles. (a) The moment of avoidance that addresses the problem shown in [Fig. 5], (b) trajectory of avoidance, (c) the speed of the ego vehicle, (d) the acceleration of the ego vehicle

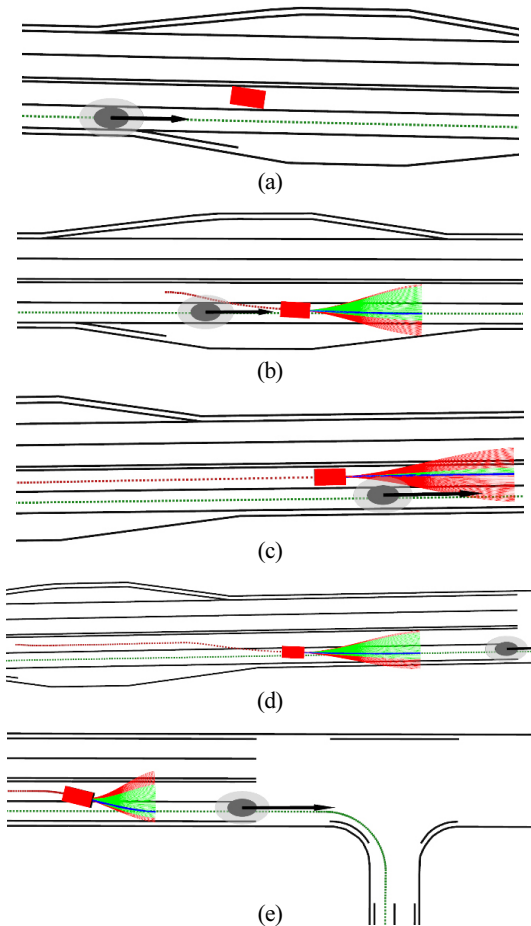
3.2 동적 물체 회피

전역 경로를 추종하기 위해 차선을 변경할 때 동적 물체가 주행을 방해하는 시나리오에서 회피 여부를 실험 하였다. [Table 2]는 실험에 사용된 변수값이다.

[Fig. 9]는 동적 물체의 속도에 따른 차량의 궤적을 보여준다. [Fig. 9(b)]에서 동적 물체의 속도(10 m/s)가 차량의 속도(7 m/s)보다 빠르나, 차량보다 뒤에 위치하고 있어 끼어들기가 가능하다고 판단한다. [Fig. 9(d)]는 차량과 동적 물체의 속도 차이가 많이 나기 때문에 동적 물체를 보내고 따라가는 것을 보여준다.

[Table 2] Parameter values for dynamic obstacle avoidance

L_{follow}	5.0 m
L_{cut-in}	5.0 m

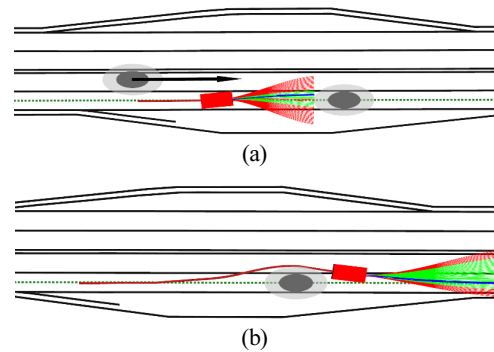


[Fig. 9] The result of following and cutting in front of a dynamic object. (a) The simulation setting in which the ego vehicle, travelling at 7 m/s initially, is located astray from the global path, and a dynamic object is approaching towards the ego vehicle on the global path, (b) cutting in front of the obstacle travelling at 10 m/s, (c) travelling alongside the obstacle moving at 13.889 m/s, (d) following the obstacle travelling at 20 m/s, (e) following the obstacle travelling at 13.889 m/s to make a turn

[Fig. 9(c)]는 차량의 속도와 동적 물체의 속도가 비슷할 때 전역 경로로 차선을 변경하지 못하는 현상을 보여준다. 이런 현상은 실제 도로에서 전역 경로로부터 크게 벗어나는 문제를 야기할 수 있다. 예로 [Fig. 9(e)]와 같이 우회전을 해야할 때 전역 경로로 차선 변경을 하지 못하고 지나쳐버릴 수 있다. 이는 차량의 전방 전역 경로의 곡률을 바탕으로 차량의 목표 속도를 정함으로써 해결 가능하다. [Fig. 9(e)]는 회전 구간의 곡률을 고려해 감속함으로써 동적 물체를 보낸 후 따라가기를 실행하는 것을 보여준다.

3.3 정적·동적 물체 회피

정적·동적 물체가 동시에 진로를 방해하는 경우에 대한 실험을 진행하였다. [Fig. 10]은 정적 물체가 전역 경로 위에 있고 동적 물체는 왼쪽 차선에서 접근 할 때의 결과이다. 차량의 최초 속도는 2.0 m/s이며, 동적 물체는 15 m/s로 주행하고 있다. [Fig. 10(a)]에서 차량은 전역 경로를 추종하는 경로 후보가 정적 물체와 충돌함을 확인하고 회피하려 한다. 왼쪽 차선으로 회피하는 경로 중 최소 가속도로 감속할 수 없는 경로 후보군은 제외된다. 남은 후보군 중 최소 가속도로 감속 가능하고, 정적 물체를 가장 안전하게 회피하며 전역 경로를 최대한 추종하는 경로가 선택된다. [Fig. 10(b)]에서 동적 물체를 보낸 후 정적 물체를 안전하게 회피하는 것을 볼 수 있다.



[Fig. 10] Static and dynamic obstacle avoidance simultaneously. (a) The moment at which the ego vehicle selects a local path that avoids the static obstacle and follows the dynamic obstacle with the given minimum acceleration value, (b) the result of successful avoidance

3.4 경로 계획 시간

본 실험에서는 시뮬레이션 차량을 왕복 2차선 도로에서 약 400 미터 길이의 계획된 전역 경로를 따라 주행하도록 하였다. 차량 사양과 환경 설정은 [Table 3]과 같다.

이하 모든 실험은 Intel Core i7-8700 CPU(12 코어, 3.2 GHz)와 16GB RAM 환경에서 진행하였다. 제안한 경로 계획 알고리즘은 각 경로 후보에 대해 비용을 계산해야 하므로 경로 후보군의 개

수와 길이에 따라 복잡도가 결정된다¹⁸⁾. 본 실험에서는 위의 설정에 따라 70개의 경로 후보군을 생성하였다. 후보군의 길이에 따른 경로 생성과 선택에 걸리는 시간은 [Table 4]에서 비교하였다. 각 결과는 100초간의 시뮬레이션에 대한 평균값이다. [Table 4]는 경로의 길이가 늘어날수록 연산량이 증가함을 보여준다. 이는 장애물과 충돌 여부를 확인해야 할 경로점들이 늘어나 생기는 현상이다. 충돌 여부를 비교해야 하는 장애물의 개수가 지역 경로 생성 알고리즘의 성능 저하를 일으키는 주요한 원인이 된다는 것을 발견했다. 실제 도로 환경에서 주변 차량의 부재 시 중앙선과 보도블록들이 주요 고려사항이 된다. 중앙선과 보도블록은 도로 정밀지도에서 촘촘한 점들로 표현되기 때문에 각 경로와의 충돌 여부를 확인할 때 많은 연산량을 요구하게 된다.

[Table 5]는 같은 실험 환경에서 차선(Lane)과 보도블록(Curb) 등의 정적 장애물을 지도에 포함했을 때와 하지 않았을 때 제안된 알고리즘과 타 알고리즘과의 경로 계획 시간을 비교한 결과이다. 차량의 주변 중앙선과 보도블록에 해당하는 점의 개수는 약 1,400개 정도이며, 100초간 주행했을 때의 평균 계산 시간을 비교하였다. 각각 알고리즘의 변수는 상이하지만 같은 개수와 같은 해상도로 경로 후보군을 만들도록 하여 동일한 조건에서 진행하였다. DWA는 전역 경로 추종이 아닌 도착점에 대해서만 고려하고, 경로 후보군을 윈도우 방식으로 생성하기 때문에 많은 계산량을 요구하지 않는다. 따라서, 정적 장애물이 없을 시 가장 빠른 계산량을 보여준다.

하지만, 정적 장애물이 존재했을 때는 가장 무거운 알고리즘으로 판단된다. DWA는 최대한 정적 장애물로부터 멀어지려는 알고리즘의 특징으로 인해 편도 2차선 도로에서 차선을 밟으며 중앙에서 주행하기 때문에 정밀 도로지도 기반 자율주행에 부적합한 성능을 보여준다. 프레넬 좌표 경로(Frenét frame trajectory) 생성 방법¹⁷⁾은 자율주행 차량에 직접 탑재해서 테스트될 정도로 자율주행에 매우 적합한 알고리즘으로 평가된다. 프레넬 좌표에서 횡방향과 종방향의 속도, 가속도, 가가속도를 고려하여 4차식 다항식(Quartic polynomial)과 5차식 다항식(Quintic polynomial)으로 표현되는 경로를 생성한 후 가장 적은 비용 함수를 가진 경로를 택한다. 프레넬 알고리즘은 정적 장애물이 존재했을 때 DWA보다 빠른 계산 속도를 보여주며, 전역 경로를 추종한다는 점에서 자율주행에 더 적합한 알고리즘이다. 하지만, 정적 장애물의 충돌 여부를 계산하는 부분에서 많은 연산량을 요구하는 것은 실시간성의 한계를 보여준다. 제안된 알고리즘은 프레넬 알고리즘과 비슷하게 횡방향의 차이로 경로 후보군을 생성하지만, 현재 차량의 속도, 차후 경로의 곡률, 회피할 장애물까지의 거리를 고려하여 하나의 종방향을 결정한다. 계산이 가벼운 비용함수를 정의하여 부드럽고 전역 경로 추종하는 지역 경로를 선택하기 때문에 프레넬 알고리즘보다 빠른 계산량을 갖는다. 또한, 경로 후반부에 차선 및 보도블록과의 충돌이 자주 일어나는 도로의 특성을 이용하여 충돌 여부 계산량을 크게 줄였다.

[Table 3] Vehicle specification and parameters

Parameter	Definition	Value
$ a_{lat} _{max}$	Maximum lateral acceleration	5.0 m/s ²
$a_{long_{max}}$	Maximum longitudinal acceleration	1.0 m/s ²
$a_{long_{min}}$	Minimum longitudinal acceleration	-3.0 m/s ²
v_{max}	Maximum speed	50 km/h

[Table 4] Computation time for the proposed algorithm

Path length (m)	Path generation (ms)	Path selection (ms)	Execution time (ms)
10	46	4	50
30	83	11	94
50	129	39	168

[Table 5] Computation time comparison among the proposed algorithm and other state-of-the-art algorithms with and without lanes and curbs, namely static obstacles

	Computation time (ms)
DWA ¹⁷⁾ (w/o)	1
Frenét ¹⁷⁾ (w/o)	51
Proposed (w/o)	43
DWA (w/)	130
Frenét (w/)	104
Proposed (w/)	56

4. 결론

본 논문에서는 동적 및 정적 물체를 회피하며 전역 경로를 추종하는 지역 경로 생성 알고리즘을 제안하였다. 이미 생성된 정밀 도로지도를 사용하므로 지도 상에 없는 물체를 식별하기 쉽고 지도 상에 있는 물체 또한 위치 오차가 적기 때문에 빠르고 정확한 지역 경로 생성이 가능하다. 본 논문에서 제안한 알고리즘은 전역 경로에 대한 좌표계 변환, 경로 후보군 생성, 비용 기반 경로 선택의 순서로 진행된다. 정의한 비용 함수를 통해 정적 장애물은 최대한 여유로운 공간을 두고 회피하고, 동적 장애물은 따라가기와 끼어들기로 회피하도록 경로를 선택한다. 알고리즘의 계산 속도 또한 실시간 경로 생성의 적합하다는 것을 보여준다. 논문에서 소개한 파라미터는 실험을 하며 설정하였으나, 향후 강화 학습을 통해 여러 시나리오에서 강인한 최적의 파라미터 값을 찾을 수 있도록 알고리즘을 향상할 수 있을 것이다. 본 논문에서는 동적 물체가 현재 속도와 경로를 그대로 유지한다는 가정으로 실험을 하였지만 제안한 알고리즘은 동적 물체의 방향과 속도가 변하여도 강인하기 때문에 추후 이를 검증하는 실험을 할 예정이다. 마지막으로 본 알고리즘을 실제 차량 실험을 통해 검증할 예정이다.

References

[1] S. G. Tzafestas, "Mobile Robot Path, Motion and Task Planning," *Introduction to Mobile Robot Control*, Elsevier Inc, 2014, pp. 429-478, DOI: 10.1016/B978-0-12-417049-0.00011-0.

[2] K. H. Sedighi, K. Ashenayi, T. W. Manikas, R. L. Wainwright, and H.-M. Tai, "Autonomous local path planning for a mobile robot using a genetic algorithm," *2004 Congress on Evolutionary Computation*, Portland, OR, USA, pp. 1338-1345, 2004, DOI: 10.1109/CEC.2004.1331052.

[3] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics and Automation Magazine*, vol. 4, no. 1, pp. 23-33, Mar., 1997, DOI: 10.1109/100.580977.

[4] J. Borenstein and Y. Koren, "The Vector Field Histogram-Fast Obstacle Avoidance for Mobile Robots," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 278-288, Jun., 1991, DOI: 10.1109/70.88137.

[5] I. Ulrich and J. Borenstein, "VFH+: reliable obstacle avoidance for fast mobile robots," *1998 IEEE International Conference on Robotics and Automation*, Leuven, Belgium, pp. 1572-1577, 1998, DOI: 10.1109/ROBOT.1998.677362.

[6] I. Ulrich and J. Borenstein, "VFH*: local obstacle avoidance with Lookahead Verification," *2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia*, San Francisco, CA, USA, 2000, DOI: 10.1109/ROBOT.2000.846405.

[7] X. Hu, L. Chen, B. Tang, D. Cao, and H. He, "Dynamic path planning for autonomous driving on various roads with avoidance of static and moving obstacles," *Mechanical Systems and Signal Processing*, vol. 100, no. 1, pp. 482-500, Feb. 2018, DOI: 10.1016/j.ymssp.2017.07.019.

[8] K. Chu, M. Lee, and M. Sunwoo, "Local path planning for off-road autonomous driving with avoidance of static obstacles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1599-1616, Dec. 2012, DOI: 10.1109/TITS.2012.2198214.

[9] H. Wang, J. Kearney, and K. Atkinson, "Arc-length parameterized spline curves for real-time simulation," *5th Int. Conf. Curves Surf.*, pp. 387-396, 2002, [Online], <https://psychology.uiowa.edu/sites/psychology.uiowa.edu/files/groups/hank/files/ArcLength03.pdf>.

[10] H. Wang, J. Kearney, and K. Atkinson, "Robust and efficient computation of the closest point on a spline curve," *5th Int. Conf. Curves Surf.*, pp. 397-406, 2002, [Online], <https://homepage.cs.uiowa.edu/~kearney/pubs/CurvesAndSurfacesClosestPoint.pdf>.

[11] I. N. Bronshtein, K. A. Semendiyayev, G. Musiol, and H. Mühlig, *Handbook of Mathematics*, 3rd ed. Springer-Verlag, 2015, [Online], <https://www.amazon.com/Handbook-Mathematics-I-N-Bronshtein/dp/0442211716>.

[12] T. Barfoot and C. Clark, "Motion planning for formations of mobile robots," *Robotics and Autonomous Systems*, vol. 46, no. 2, pp. 65-78, Feb., 2004, DOI: 10.1016/j.robot.2003.11.004.

[13] E. F. Camacho and C. B. Alba, *Model Predictive Control*, Springer Science & Business Media, 2013, [Online], https://books.google.co.kr/books/about/Model_Predictive_Control.html?id=tXZDAAAAQBAJ&redir_esc=y.

[14] S.-M. Lee, H. Kim, and H. Myung, "Cooperative Particle Swarm

Optimization-based Model Predictive Control for Multi-Robot Formation," *Journal of Institute of Control, Robotics and Systems*, vol. 19, no. 5, pp. 429-434, May, 2013, DOI: 10.5302/J.ICROS.2013.13.9007.

[15] M. Kim, G. Im, and J. Park, "A Comparative Study of Parking Path Following Methods for Autonomous Parking System," *Journal of Korea Robotics Society*, vol. 15, no. 2, pp. 147-159, Jun., 2020, DOI: 10.7746/jkros.2020.15.2.147.

[16] J. Choi, G. Lee, and C. Lee, "Path Tracking with Nonlinear Model Predictive Control for Differential Drive Wheeled Robot," *Journal of Korea Robotics Society*, vol. 15, no. 3, pp. 277-285, Sept., 2020, DOI: 10.7746/jkros.2020.15.3.277.

[17] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a Frenet frame," *2010 IEEE International Conference on Robotics and Automation*, pp. 987-993, 2010, DOI: 10.1109/ROBOT.2010.5509799.



정의곤

2020 한국과학기술원(KAIST) 전기및전자공학부(학사)
2020~현재 한국과학기술원(KAIST) 전기및전자공학부(석사)

관심분야: Robotics, Autonomous Driving, Sensor Fusion



송원호

2015 연세대학교 토목환경공학과(학사)
2018 Carnegie Mellon Univ., Civil and Environmental Eng. (석사)
2019~현재 한국과학기술원 전기 및 전자공학부(박사)

관심분야: Autonomous Driving, Localization, Sensor Fusion



명현

1992 한국과학기술원 전자전산학부(공학사)
1994 한국과학기술원 전기및전자공학과(공학석사)
1998 한국과학기술원 전기및전자공학과(공학박사)
1998~2002 한국전자통신연구원(ETRI) 선임연구원
2002~2003 (주) 이머시스 연구소장/기술이사

2003~2008 삼성전자종합기술원 전문연구원
2008~2018 한국과학기술원 건설및환경공학과 교수
2018~현재 한국과학기술원 로봇공학학제 책임교수
2019~현재 한국과학기술원 전기및전자공학부 교수

관심분야: Autonomous Robot/Drone/Car Navigation, SLAM, Deep Learning, Machine Learning