

자율주행 차량을 위한 경로 계획-추적 통합 제어 시스템 설계 및 성능 평가

김지형*, 김양중**

한국공학대학교 소프트웨어융합공학과

{*morisatokiki, **zeroplus}@tukorea.ac.kr

Design and Performance Evaluation of an Integrated Path Planning–Tracking Control System for Autonomous Vehicles

Kim Ji Hyoung*, Yangjung Kim**

Tech University of Korea

요약

자율주행 제어에 있어 기존 방식은 커맨드형 액추에이터 제어에 의존하여 경로는 정확하게 따라가더라도 승차감 저하, 조향 진동과 같은 실용성 문제가 있었다. 이에 차량의 전체 궤적 정보를 기반으로 부드럽고 연속적인 제어를 수행할 수 있는 경로 기반 제어기를 설계할 필요가 있다. 따라서 본 논문은 주행 제어기에서 나타나는 승차감 저하, 급가속과 급감속, 핸들 떨림 문제 등을 해결하기 위해, 경로 계획과 추적을 통합한 새로운 제어 알고리즘을 제안하며, 제안하는 제어기는 Kanayama Control을 기반으로 알고리즘을 개발을 통해, 매트랩 시뮬링크를 이용한 모델을 구현 및 성능평가를 진행하였으며, 추가적으로 임베디드 보드에 탑재해 실제 시스템하고 실험을 통해 검증하였다. 결국, 제안된 제어 시스템 방식이 기존 방식보다 경로 추종과 속도 안정성을 제공할 수 있음을 확인함은 물론, 실시간 제어 서비스를 저가형 시스템에서도 적용 가능함을 확인하였다.

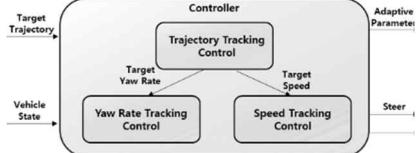
I. 서 론

자율주행 제어 시스템의 핵심은 주어진 경로를 얼마나 정확하고 안정적으로 추종할 수 있는가에 있다. 일반적으로, 기존의 제어 시스템은 종방향 속도제어와 횡방향 조향제어를 각각 독립적으로 수행함으로, 목표한 위치로의 수렴성은 확보할 수 있지만, 전체 궤적에 대한 분석없이 단순히 변화에 따른 오차만을 보정하는 방식이기 때문에 다음과 같은 문제점을 수반한다.

- 급감속 또는 급가속 시 승차감을 저하
- 조향각의 불연속 제어로 인해 핸들 떨림 현상이 발생에 따른 고속 주행 시 차량 불안정성
- 저가형 하드웨어의 오차 보정을 실시간으로 실행하지 못함에 따른 응답지연

제기된 문제를 해결하기 위해서, 최근에는 경로 계획(Path Planning)과 추적(Path Tracking)을 통합하여 제어기를 설계하는 연구가 활발히 진행되고 있다. 예를 들어, MPC 기반 제어 구조를 활용하여 경로 계획과 추적을 통합하는 시도가 있었으나[2][3], 이들은 대부분 고성능 연산 자원을 요구하는 대신 실시간 지원이 제한되는 한계를 가진다. APF(Artificial Potential Field) 기반 접근이나 Stanley 제어기 기반 통합 모델[4][5]은 외부 복잡도에 민감하거나 곡선 구간에서 제어 성능이 급격히 저하되는 문제를 갖고 있다. 승차감 향상과 연속 제어에 초점을 맞춘 연구는 주로 'Jerk 방식'이나 '조향 진동' 특성을 중심으로 연구되고 있으며, 주행 안정성 평가에 대한 다양한 정량 지표들이 제안되고 있다[6][7].

본 논문은 이러한 기존 연구의 한계를 고려하여, 고속 주행 시 승차감과 주행 안정성을 동시에 확보할 수 있는 경로 기반 제어기를 설계하고, 저비용 임베디드 시스템에서도 실시간 제어가 가능함을 검증하는 것을 목표로 한다. 이에, 제안하는 전체 제어 시스템의 계층적 구조는 <그림 1>에서 확인할 수 있다.

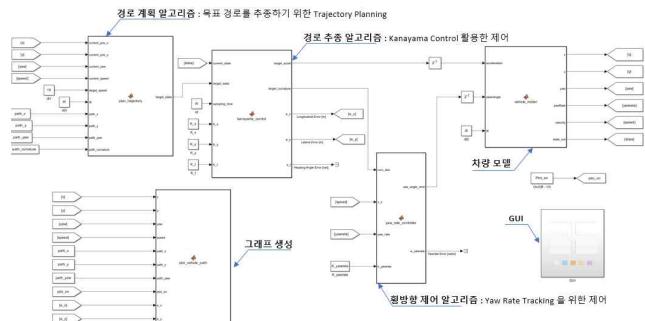


<그림 1. 경로 기반 자율주행 제어기의 구성도>

II. 본론

1. 경로 계획 및 추적의 통합

기존 제어기는 경로 추종 오차 보상이 미흡하여 고속 주행 시 안정성 저하 문제가 발생하므로 경로 계획과 추적을 독립적으로 처리하는 기존 방식과 달리, 두 기능을 통합함으로써, 차량은 현재 상태뿐만 아니라 예측된 미래 경로를 기반으로 제어가 가능하다. 이는 목표 경로와 현재 상태를 기반으로 최적의 경로를 계획하고, 실시간으로 보정할 수 있는 기능을 제공한다. 본 논문에서는 Kanayama Control[1]을 활용하여 경로 계획과 추적을 결합한 자율주행 제어 알고리즘을 제안한다. <그림 2>는 제안하는 시스템 내부 시스템 구성 및 세부 알고리즘 흐름을 보여준다.



<그림 2. 시스템 구성 및 알고리즘 설계>

2. 경로 계획 및 추적 알고리즘

경로 계획 알고리즘은 차량이 목표 궤적을 따라 이동하도록 매 순간 최적의 상태를 계산을 통해 판단하는 역할을 한다. 차량의 상태와 목표 경로 간의 오차를 제안된 알고리즘을 통해 확인하고, 이를 기반으로 제어 신호를 생성한다.

2.1. 상태 변수 정의

상태 변수는 위치, 방향(yaw), 속도, 목표 경로 좌표 및 곡률로 구성된다.

2.2. 경로 추적 오차

차량의 현재 상태와 목표 상태 간의 오차는 다음과 같이 정의된다:

- 종방향오차:

$$e_x = \cos(\theta) \cdot (x_d - x) + \sin(\theta) \cdot (y_d - y) \quad \text{식(1)}$$

- 횡방향오차:

$$e_y = -\sin(\theta) \cdot (x_d - x) + \cos(\theta) \cdot (y_d - y) \quad \text{식(2)}$$

- 헤딩 오차: $e_\theta = \theta_d - \theta$

$$\text{식(3)}$$

(오차 e_θ 는 $[-\pi, \pi]$ 범위로 제한)

2.3. 목표 곡률과 속도 계산

경로 계획 알고리즘에서 생성된 주행 궤적은 구간별로 곡률 K_d 및 속도 v_{target} 의 형태로 제공된다. 이러한 목표 값은 차량이 현재 주행 상황에 따라 매시간 갱신되며, 차량 제어기의 기준 값으로 사용된다.

목표 곡률 K_d : 주행 경로의 기하학적 곡률을 나타내며, 궤적 곡선의 회전 반경에 기반하여 결정된다.

목표 속도 v_{target} : 곡률 크기에 따라 안전한 주행을 위해 제한되며, 커브에서는 속도를 낮추고 직선 구간에서는 높은 속도를 유지하도록 조정된다. 이러한 목표 값들은 이후 제어 단계에서 차량 상태와의 오차에 따라 실시간으로 보정되어 최종 제어 명령으로 변환된다.

2.4. 위치 제어 알고리즘

위치 제어는 차량이 목표 경로를 정확히 추종하도록 하기 위한 제어 방식으로, 본 연구에서는 Kanayama 제어기를 기반으로 제어 알고리즘을 구성하였다.

Kanayama 제어기는 차량의 현재 위치와 방향이 목표 궤적과 일치하도록 유도하는 데 효과적이며, 실시간 위치 오차를 기반으로 목표 곡률 κ 를 계산하여 차량이 자연스럽게 경로를 따르도록 한다. 차량의 좌표계 기준으로 정의된 횡방향 오차 e_y 및 헤딩오차 e_θ 를 보상형으로 활용하여, 제어 곡률 κ 는 다음과 같이 계산된다:

$$\kappa = \kappa_d + K_y + e_y + K_\theta \cdot \sin(e_\theta) \quad \text{식(4)}$$

여기서 κ_d 는 경로 계획 알고리즘에서 주어진 목표 곡률이며, e_y 와 e_θ 는 각각 횡방향 오차 및 방향 오차이다. 해당 식은 목표 곡률을 중심으로 피드백 보정을 적용한 것으로, 차량이 경로를 따라 정확히 회전이 가능하도록 유도한다.

목표 속도 v_d 는 다음과 같이 계산된다:

$$v_d = v \cdot \cos(e_\theta) + K_x \cdot e_x \quad \text{식(5)}$$

2.5. 가속도

계획된 속도에 차량이 부드럽게 수렴하도록 하기 위해서, 초기 속도 계획은 Jerk 기반으로 아래와 같이 표현할 수 있다.

$$a_{target} = 0.07 \cdot (v_{target} - v) - 0.5 \cdot a \quad \text{식(6)}$$

여기서 a 는 현재 가속도, 그리고 v_{target} 은 경로의 곡률에 따라 결정된 목표 속도이다. 이 방식은 목표

속도에 빠르게 도달하면서도 급격한 Jerk를 억제하여, 속도 감소를 고려한 부드러운 응답을 달성할 수 있다.

3. 속도 제어 알고리즘

속도 제어는 차량이 목표 속도에 도달하도록 가속도 및 감속도를 조정한다. 이를 위해 PID 제어를 사용하며, 제어 입력은 다음과 같이 정의된다.

$$v_d = v_{target} \cdot \cos(e_\theta) + K_x \cdot e_x \quad \text{식(7)}$$

$$e_v^{\text{int}} = \int (v_d - v) dt \quad \text{식(8)}$$

$$v_d^{\text{new}} = v_d^{\text{old}} + K_p (v_d^{\text{old}} - v) + K_i \cdot e_v^{\text{int}} \quad \text{식(9)}$$

$$a_{target} = \frac{v_d - v}{T_s} \quad \text{식(10)}$$

이 식은 차량의 종방향 오차 및 속도 상태를 고려한 제어 입력을 실시간으로 보정하며, 실제 적용 환경에서도 구현이 간단하다는 장점을 가진다.

4. 횡방향 제어 알고리즘 (Yaw Rate 기반)

Yaw Rate 제어는 차량이 목표 방향으로 정확하게 회전하도록 유도하는 기능을 갖으며, Yaw Rate 오차 자체를 직접 제어에 사용하는 방식은 다음과 같은 제한점을 확인할 수 있다.

- 차량이 정지 상태일 때 Yaw Rate은 0이기 때문에 제어가 무의미함

- 정지 상태에서 미리 조향각이 설정되어 있어야 하며, 그렇지 않으면 출발 시 제어 오류 발생

이에 해결하기 위해서, 본 논문에서는 Curvature 기반 제어 방식을 적용하였으며, 차량의 곡률 오차는 다음과 같이 정의된다.

$$e_{curv} = \kappa_d - \frac{\omega}{v} \quad \text{식(11)}$$

Curvature 오차는 차량의 조향각 델타를 조정하여 다음과 같은 제어 법칙으로 보상된다.

$$\delta = K_\omega \cdot e_{curv} + (gearratio) \cdot \kappa_d \quad \text{식(12)}$$

5. 차량 모델

본 논문에서는 제안한 제어기의 유효성을 검증하기 위해 단순한 차량의 운동학적 모델을 구성하였다. 해당 모델은 가속도 및 조향각 입력에 따라 다음과 같은 방식으로 상태를 갱신한다:

$$\begin{aligned} x_{k+1} &= x_k + v_k \cdot \cos(\theta_k) \cdot dt \\ y_{k+1} &= y_k + v_k \cdot \sin(\theta_k) \cdot dt \\ \theta_{k+1} &= \theta_k + \omega_k \cdot dt \\ v_{k+1} &= v_k + a \cdot dt \\ \omega_{k+1} &= \frac{\delta \cdot v_k}{L} \end{aligned} \quad \text{식(13)}$$

x_k, y_k : 차량의 현재 위치,

θ_k : 차량의 방향 (yaw angle),

v_k : 현재 속도,

ω_k : Yaw Rate,

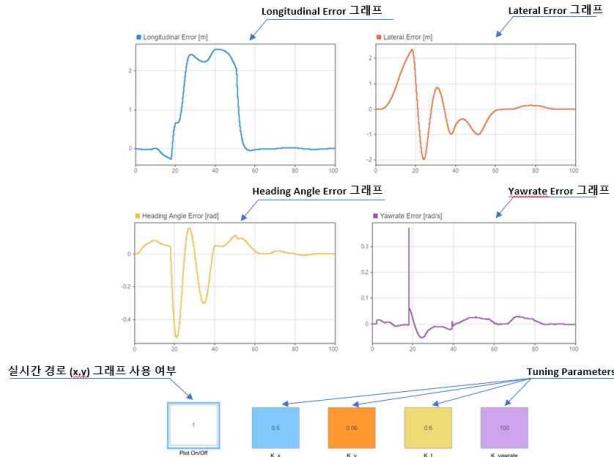
δ : 조향각 (steering angle),

L : 차량의 휠베이스

6. 시뮬레이션 환경

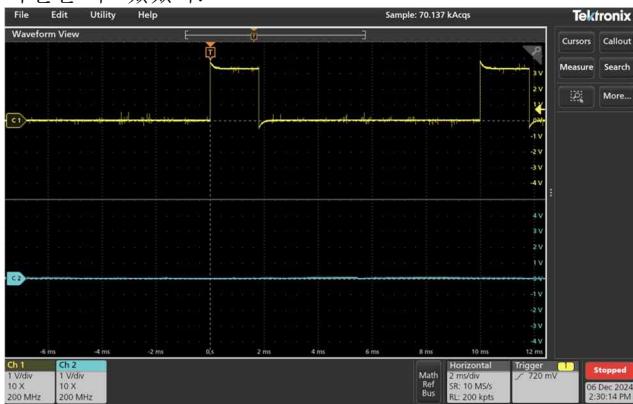
실제 주행환경에 맞춘 시뮬레이션 환경은 매트랩 시뮬링크를 이용하였으며, 초기 차량의 위치와 속도, 그리고 목표 궤적을 실험 목적에 맞게 설정되며, 이를 통해 경로 추적 및 속도 제어 성능을 평가하였다. 차량이

목표 궤적을 얼마나 정확하게 추종하는지, 또 목표 속도를 얼마나 안정적으로 유지하는지 확인 및 분석한다.



<그림 3. 시뮬레이션 실험 환경 및 방법>

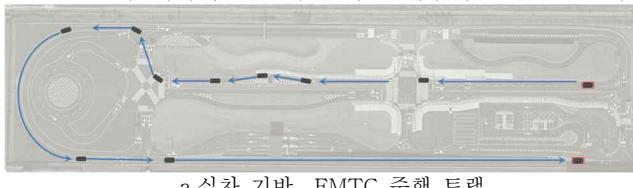
매트랩 시뮬링크 기반의 경로 추종 알고리즘은 C 코드로 구현하여 저비용 환경(NXP MPC5748G[8] 보드)에 이식함으로써 실시간 제어 성능을 검증한다. <그림 4>에서 볼 수 있듯이, 1.8ms 의 제어 실행시간과 차량이 기본 제어 주기와 동일한 10ms 의 제어 주기를 확인할 수 있었다.



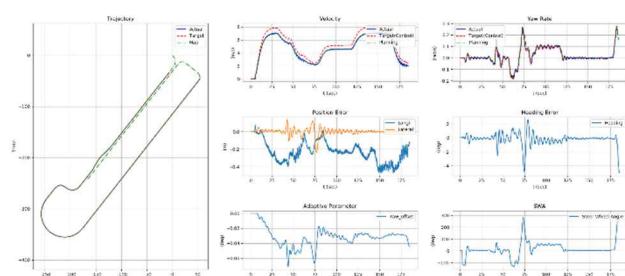
<그림 4. 1 Cycle 실행 시간 및 주기>

7. 실차 실험

실제 주행 시험장에서 제안된 알고리즘의 성능을 검증하기 위한 주행 시나리오를 구성하였으며, 결국 도출된 결과 데이터를 통해 분석을 세부적으로 진행한다.



a. 실차 기반 FMTC 주행 트랙



b. 실차 기반 알고리즘 성능 검증

<그림 5. 실차 기반 FMTC 주행 트랙(a) 및 실차 기반 알고리즘 성능 검증(b)>

실차 환경은 FMTC 트랙에서 구성되었으며, 주행 코스 및 구간과 전체 실험 결과는 <그림 5>로 확인할 수 있다.

III. 결 론

본 논문에서는 경로 계획과 추적을 결합한 자율주행 제어 알고리즘을 제안하였으며, 이를 통해 기존 제어기의 한계인 핸들 떨림 및 급정거 문제를 해결할 수 있었으며, 저비용 임베디드 보드를 활용한 제어 시스템에서 안정적인 성능을 확보할 수 있음을 시뮬레이션을 통해 검증하고 실차 실험을 진행하였다. 경로 계획 및 추적 알고리즘의 성능 개선을 통해 더욱 현실적인 상황을 적용하기 위한 추가 방안을 연구할 것이며 이에 세부적인 성능도 실험을 통해 평가하고자 한다.

ACKNOWLEDGMENT

본 연구는 고용노동부 및 한국산업인력관리공단의 ‘고속련 마이스터 사업(2025)’의 지원을 받음.

참 고 문 헌

- [1] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, "A stable tracking control method for an autonomous mobile robot," in Proc. IEEE Int. Conf. Robotics and Automation (ICRA), 1990, pp. 384–389.
- [2] T. Yang, L. Liu, and Z. Xu, "Integrated Trajectory Planning and Tracking Control for Autonomous Vehicles Based on Pseudo-time-to-Collision Risk Assessment Model," SAE Technical Paper 2024-01-5046, 2024. doi: 10.4271/2024-01-5046
- [3] C. Huang, B. Li, and M. Kishida, "Model predictive approach to integrated path planning and tracking for autonomous vehicles," in 2019 IEEE Intelligent Transportation Systems Conference (ITSC), 2019, pp. 1448–1453, doi: 10.1109/ITSC.2019.8916898
- [4] M. H. Al-Jumaili and Y. E. Özok, "New control model for autonomous vehicles using integration of Model Predictive and Stanley based controllers," Scientific Reports, vol. 14, no. 1, p. 13516, 2024. doi: 10.1038/s41598-024-69858-7.
- [5] Y. Zhang, K. Liu, F. Gao, and F. Zhao, "Research on Path Planning and Path Tracking Control of Autonomous Vehicles Based on Improved APF and SMC," Sensors, vol. 23, no. 18, p. 7918, 2023. doi: 10.3390/s23187918
- [6] Z. Chen, Y. Liu, W. Ni, H. Hai, C. Huang, B. Xu, Z. Ling, Y. Shen, W. Yu, H. Wang, and J. Li, "Predicting driving comfort in autonomous vehicles using road information and multi-head attention models," Nature Communications, vol. 16, no. 1, p. 2709, 2025. doi: 10.1038/s41467-025-57845-z.
- [7] D. G. Sharma, I. Taney, and K. Shimohara, "Detecting driver-induced steering oscillations through adaptive thresholding of the power spectrum of vehicle's lateral acceleration," IEEJ Transactions on Electronics, Information and Systems, vol. 138, no. 3, pp. 254–262, 2018. doi: 10.1541/ieejeiss.138.254.
- [8] NXP Semiconductors, "MPC5748G Microcontroller Data Sheet," 2018.