*Journal of Computational Design and Engineering*, 2025, **12**, 201–222
DOI: 10.1093/jcde/qwaf107
Advance access publication date: 17 October 2025
Research Article

# Development of a lateral control system for autonomous vehicles by integrating quasi-static and dynamic control methods

Hong Seung Kim [1] and Yong-Gu Lee [2,*]

[1]Space R&D Centre, Hanwha Systems Co. Ltd, 491-23 Gyeonggidong-ro, Yongin-si 17121, Republic of Korea
[2]Artificial Intelligence Graduate School, Gwangju Institute of Science and Technology (GIST), 123 Cheomdangwagi-ro, Buk-gu, 61005 Gwangju, Republic of Korea
*Correspondence: lygu@gist.ac.kr

## Abstract

Recent advancements in lateral control systems for autonomous vehicles have focused on achieving high tracking precision and stable control performance. Building on these efforts, this study designs and implements an integrated lateral control system using quasi-static Stanley method and dynamic Proportional, Integral, Derivative (PID) controller.

The proposed system works in three stages: positioning, quasi-static control and dynamic control. In the positioning stage, precise positioning is achieved by applying Real-Time Kinematic enhanced positions and headings with velocity obtained from Inertial Measurement Unit. Additionally, a method is proposed to collect and process path coordinate data in environments where predefined path coordinates are unavailable. In the quasi-static control stage, an efficient path tracking control algorithm based on the Stanley method is implemented and applied to the collected ground truth path coordinate data. Finally, in the dynamic control stage, the PID controller is utilized to enhance control precision by minimizing the difference between the vehicle's actual heading and the target heading response time required to reach the target heading through adjusting the steering wheel angle.

The experiment was conducted by driving along two target paths within the GIST campus. On the first target path, Mean Absolute Error (MAE) was recorded at 0.0786 m, and Root Mean Square Error (RMSE) was 0.1140 m. Additionally, the average error between the vehicle's target heading and actual heading was recorded as 2.1411°. On the second target path, MAE was 0.0427 m, and RMSE was 0.0802 m. The average error between the target heading and actual heading was 2.5000°, demonstrating the performance of the lateral control system.

To prove the superiority of the proposed method, thorough collections of prior real-world lateral control systems were exercised, and direct comparisons were made. To the best of our knowledge, the proposed system outperforms previous literature on real-world experiments. We also provide the download link to the target path coordinate data so that future competitions on the best lateral control systems can be held. We believe our work can serve as a foundation for future lateral control systems in the autonomous driving community.

**Keywords:** GNSS, IMU, lateral control, path tracking, stanley method, PID controller, autonomous vehicle

## Nomenclature

### Acronyms

| | |
|---|---|
| GNSS: | Global Navigation Satellite System |
| IMU: | Inertial Measurement Unit |
| RTK: | Real-Time Kinematic |
| PID: | Proportional, Integral, Derivative |
| MAE: | Mean Absolute Error |
| RMSE: | Root Mean Square Error |
| DARPA: | Defense Advanced Research Project Agency |
| MPC: | Model Predictive Control |
| LiDAR: | Light Detection and Ranging |
| RADAR: | Radio Detection and Ranging |
| UTM: | Universal Transverse Mercator |
| VRS: | Virtual Reference Station |
| NTRIP: | Networked Transport of RTCM via Internet Protocol |
| NGII: | National Geographic Information Institute |
| NMEA: | National Marine Electronics Association |
| RTCMv3: | Radio Technical Commission for Maritime Services version 3 |
| RDDF: | Route Definition Data File |
| HD map: | High-Definition map |

### Symbols

| | |
|---|---|
| $\delta(t)$ : | Steering angle (°) |
| $\varphi(t)$ : | Heading error (°) |
| $e(t)$ : | Cross-track error ($m$) |
| $v(t)$ : | Vehicle velocity ($m/s$) |
| $k$ : | Control gain ($s^{-1}$) |
| $\emptyset(t)$ : | Vehicle heading (°) |
| $e_h(t)$ : | Feedback error for PID controller (°) |
| $\omega(t)$ : | Desired heading (°) |

**Received:** June 12, 2025. **Revised:** October 10, 2025. **Accepted:** October 11, 2025

# 1. Introduction

## 1.1. Research background

While numerous lateral control algorithms have shown promise in simulations, a critical gap remains in evaluating their robustness and repeatability under real-world conditions. Inconsistent test paths, vehicle set-ups, and sensor integration make direct comparison nearly impossible. Furthermore, the complex nature of the interactions between the road and the vehicle is not fully addressable in simulations. It remains unclear how these simulation-based methods translate to real-world performance. To address this challenge, we propose a lateral control system for autonomous vehicles on real roads. Another problem for real road experiments is the versatility of the used tracks. Previous works on lateral control systems were conducted on different road tracks and thus makes it difficult to conduct direct comparisons. As another contribution, we propose a normalized metric to enable more objective comparisons of lateral control systems tested on different paths.

We assume the following ground truth information is given. Those are (1) ground truth path with precise waypoints, (2) real-time access to the headings, positions, and velocity of the vehicle. Implementation of our work was done with a commercial position/heading/velocity measurement device. This device internally incorporates position measurement with dual antennas for Global Navigation Satellite System (GNSS) and achieves centimeter-level precision of position measurement through Real-Time Kinematic (RTK). Dual antennas were used to provide heading angles of the vehicle. It also provides time integration-based velocity measurements through Inertial Measurement Unit (IMU).

This study presents a lateral control system with three main stages: positioning, quasi-static control, and dynamic control. In the positioning stage, we rely on a commercially available position/heading/velocity measurement device that is assumed to provide precise ground truth positioning. The positioning capability also allowed us to create test paths for later experiments. In other words, collected positions were processed offline to create test paths.

In the quasi-static control stage, the Stanley method is used to determine the steering angle. The Stanley method enables efficient and intuitive path tracking based on the geometric relationship between the vehicle's current position, heading angle and velocity, and its performance is evaluated by applying it to the test path coordinate data previously collected. We name it quasi-static because, although static, instantaneous velocity is considered.

In the dynamic control stage, the PID controller is used to improve control precision by minimizing the discrepancy between the steering wheel angle and the actual vehicle behavior (i.e., the change in vehicle heading), while simultaneously minimizing the response time required to reach the target heading.

While both Stanley and PID are well-known, their integration in a field-validated, open-data lateral control platform—using real-world RTK-GNSS/IMU under reproducible test paths—offers unique value to the research community.

Overall, this study aims to demonstrate path tracking and lateral control through the known ground truth position, heading and velocity of the vehicle. This foundational system is intended to serve as a basis for the development of more sophisticated autonomous driving systems. Note, we do not aim to enhance the measured position, heading and velocity. This way, our approach is sensor-agnostic and compatible with alternative positioning and heading systems. The isolation of the sensor values (position, heading, and velocity) from the control problem allows more concentrated effort on the control issues, which is our primary contribution.

In summary, the main contributions of this study are as follows:

1. We propose a two-stage lateral control framework (quasi-static Stanley + dynamic PID) validated on public roads using a full-sized commercial vehicle.
2. We introduce a path-length-normalized tracking error metric to benchmark heterogeneous real-world studies fairly.
3. We release path coordinates, trajectory logs, and implementation details to support reproducibility.
4. We emphasize that the GNSS/IMU pipeline primarily supports reproducibility of the experiments and is not claimed as a control-theoretic novelty. The main contribution lies in the lateral control framework and the use of normalized benchmarking.
5. We emphasize a lightweight, deployment-ready architecture that complements optimization-heavy methods such as MPC.

## 1.2. Related work

### 1.2.1. *Lateral control of autonomous vehicles*

Lateral control of autonomous vehicles is a key task to control the vehicle so that it follows the path as accurately as possible. To achieve this, various path tracking algorithms (Liu *et al.*, 2017; Kouvaritakis & Cannon 2016; Park *et al.*, 2014; Thrun *et al.*, 2006) have been proposed. The most widely used algorithm is the Stanley method (Thrun *et al.*, 2006). It was first introduced by winning the 2005 DARPA Grand Challenge. These methods have their strengths and limitations (Samuel *et al.*, 2016; Vivek *et al.*, 2019). The Stanley method is widely used and minimizes both cross-track error and heading error to track the path simultaneously. Numerous studies (Amer *et al.*, 2018; Snider, 2009; Vivek *et al.*, 2019; Wang *et al.*, 2022) based on the Stanley method have demonstrated its path tracking performance in simulation environments. However, there is a lack of discussion regarding the implementation of the algorithm on actual vehicles driven on real roads. This study also uses the Stanley method (what we regard as quasi-static control) combined with dynamic control and aims to apply it to an actual vehicle driven on real roads. An additional benefit of our work is that our work can be also used to acquire test paths. These test paths or discretized path coordinates were converted and used as ground truth path coordinates.

The other geometric path tracking algorithm known as Pure Pursuit controls the vehicle by setting a look-ahead distance and searching for a target point on the path that is a given distance ahead of the vehicle. Pure Pursuit, along with studies that applied variations such as adaptive look-ahead distance (Park *et al.*, 2014), has been proposed in research that highlights its advantage of simplicity in implementation, making it suitable for actual vehicle control (Andersen *et al.*, 2016; Samuel *et al.*, 2016).

In addition to geometric path tracking algorithms represented by the Stanley method and Pure Pursuit, model-based control algorithms such as Model Predictive Control (Kouvaritakis & Cannon 2016; Liu *et al.*, 2017; Raffo *et al.*, 2009; Schwenzer *et al.*, 2021) have also been proposed. These model-based control algorithms predict the future state of the vehicle and calculate the optimal control inputs within given constraints.

However, recent advances in model MPC have significantly reduced computational burden, enabling real-time implementations not only in academic prototypes but also in industrial contexts. For instance, MPC-based slip ratio control for EVs considering road roughness was demonstrated (Ma *et al.*, 2019). A real-time nonlinear MPC for yaw motion optimization of distributed drive EVs was developed (Guo *et al.*, 2020). Most recently, a Lyapunov-based NMPC for handling stability of distributed drive EVs was proposed (Guo *et al.*, 2025). These studies clearly show that MPC can be computationally efficient and industrially applicable.

In contrast, the present study does not seek to replace advanced MPC solutions, but rather to provide a lightweight, deployment-ready two-stage controller that prioritizes simplicity, reproducibility, and ease of tuning for real-vehicle experiments. We believe this complementary perspective is valuable, particularly for benchmarking and experimenting in early-stage autonomous vehicle deployments.

Beyond classical and model-predictive approaches, recent studies have explored reinforcement-learning controllers for highway driving, reporting adaptive policies learned at scale (Kiran *et al.*, 2022; Sallab *et al.*, 2017). At the platoon level, both string-stability synthesis and information-flow topology have been analyzed to guarantee stability and scalability from single vehicles to coordinated fleets (Ploeg *et al.*, 2014; Zheng *et al.*, 2016). At the planning–control interface, spatiotemporal-restricted A∗ has been proposed to support lane-free intersection management under mixed flows (Chi *et al.*, 2024). These advances underscore the breadth of modern motion control. In contrast, our work provides a lightweight, deployment-ready baseline with real-world validation and open data, complementing these advanced approaches by enabling practical benchmarking and technology transfer.

In addition, several recent studies have addressed perception, safety, and planning-level topics that complement controller-centric research. Vision-based frameworks have been applied to detect minor-impact collisions in long video sequences (Hwang & Lee, 2024) and to estimate split liability in vehicle accidents using three-dimensional convolutional neural networks (Lee & Lee, 2023). At the mission-planning level, a D∗ Lite based approach has been proposed for unmanned combat vehicles operating under dynamically changing adversarial environments (Ahn *et al.*, 2023), while probabilistic risk-assessment frameworks integrating motion prediction and situational reasoning have been developed for urban autonomous driving (Noh & An, 2022). These studies collectively cover perception, safety, and planning aspects, complementing the present work at the control layer and situating our lightweight, deployment-ready lateral control system within the broader context of computational design and engineering research on autonomy.

Since previous works were evaluated under different experimental conditions including variations in used vehicles, targeted test paths, and evaluation methodologies performance comparison is inherently limited. To enable a more direct comparison, this study introduces a dimensionless performance metric, defined as the lateral error normalized by the total path length. This normalization provides a fair basis for evaluating and comparing path tracking performance across heterogeneous test environments.

When applying a path tracking algorithm, it is essential to calculate how much the vehicle's direction needs to change in order to follow the path. An equally important process is ensuring that the vehicle behaves accurately based on the calculated steering angle, thus changing its direction as required. PID controller (Johnson & Moradi, 2005) is one of the most straightforward yet powerful control technique for this purpose. It is a feedback control system that defines the error as the difference between the actual output and the desired output and uses this error to adjust the control input. Pure Pursuit with PID controller (Park *et al.*, 2014) has been proposed, and it successfully followed a predefined test path.

### *1.2.2. Autonomous vehicle sensors and positioning*

To replace human drivers, autonomous vehicles must achieve perception and decision abilities similar to those of human drivers. In this context, autonomous vehicles must continuously assess their own position and the surrounding environment. To achieve this, autonomous vehicles are utilizing various sensors to replace human cognitive capabilities (Van Brummelen *et al.*, 2018).

The sensors mounted on autonomous vehicles perform three main functions: self-sensing, positioning, and surrounding sensing (Vanholme *et al.*, 2012). Self-sensing refers to the use of proprioceptive sensors to monitor the vehicle's current state. Positioning also relies on proprioceptive sensors to determine the vehicle's current position. The proprioceptive sensors commonly used for this purpose are GNSS and IMU. For surrounding-sensing, exteroceptive sensors are used to collect data on external factors, providing information about the surrounding environment.

In summary, the vehicle's self-sensing and positioning utilize proprioceptive sensors, with GNSS and IMU being representative examples. For surrounding-sensing, exteroceptive sensors are used, including cameras, Light Detection and Ranging (LiDAR), and Radio Detection and Ranging (RADAR).

For an autonomous vehicle to accurately follow a path, it is crucial to perform the positioning as precisely as possible. As discussed earlier, the most common localization method is the use of GNSS (Cui & Ge, 2003; Zein *et al.*, 2018). By equipping the vehicle with GNSS, it can express its position using latitude and longitude coordinates (Rahiman & Zainal, 2013). Additionally, by adopting the Universal Transverse Mercator (UTM) coordinate system (Langley 1998), the latitude and longitude coordinates can be converted into meters, making it easier to represent distance, area, direction, and other relevant metrics.

GNSS, when used without correction, typically results in an error of several meters (Karaim *et al.*, 2018; Langley *et al.*, 2017). To achieve precise positioning, various methods have been proposed (Langley 1998; Rezaei & Sengupta, 2007; Teunissen & Khodabandeh, 2015). In this study, Real-Time Kinematic (Langley 1998), specifically network RTK through Virtual Reference Station (Landau *et al.*, 2009), was applied, and the procedure is discussed in detail in the Appendix.

## 2. Overall System and Positioning

### 2.1. System overview

The overall system was conducted on an autonomous vehicle modified for research purposes and for the objectives of the research. For the acquisition of position, heading and velocity data the vehicle is equipped with GNSS antenna and IMU. Moreover, to achieve best precision it is connected to the wireless internet to apply RTK.

Additionally, actuators and modules capable of controlling acceleration, deceleration, and steering were installed on various components of the vehicle. For this work only the steering module is utilized through CAN based commands. Figure 1shows the mounted positions and configurations of the sensors and control modules, while Table 1 provides the specifications of the utilized devices.

The developed system is designed with three main stages: positioning, path tracking via quasi-static control and dynamic control. Figure 2 shows a detailed flowchart of the overall system. The positioning stage is where the vehicle's position, heading and velocity data are acquired; the quasi-static control stage involves calculating the steering angle using a path tracking algorithm;

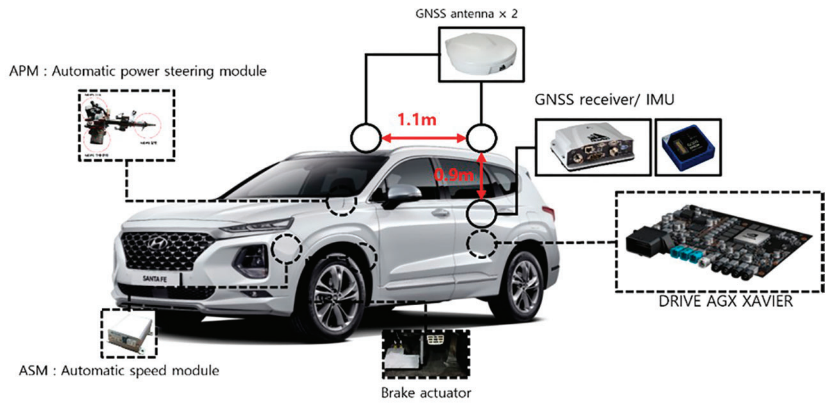


**Figure 1:** Configuration of onboard sensors and control modules.

**Table 1:** Device specifications.

| Vehicle model | Hyundai SantaFe TM |
|---|---|
| GNSS | Novatel PwrPak7D-E1 |
| GNSS antenna | Novatel VEXXIS® GNSS-500 |
| IMU | Epson G320N |
| Embedded board | NVIDIA Drive AGX Xavier |

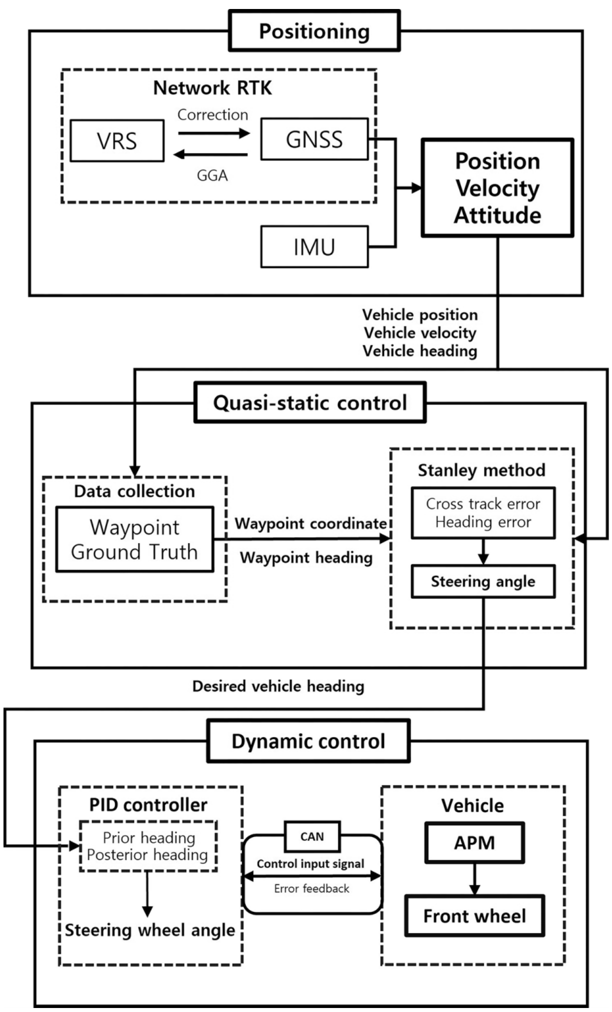


**Figure 2:** Overall system diagram.

**Table 2:** Input/output data for each system stage.

| | Input | Output |
|---|---|---|
| Positioning | Vehicle position<br>Vehicle velocity<br>Vehicle heading<br>RTK correction | Corrected vehicle position<br>Corrected vehicle velocity<br>Corrected vehicle heading |
| Quasi-static control | Corrected vehicle position<br>Corrected vehicle velocity<br>Corrected vehicle heading<br>Path coordinate, Path heading | Steering angle<br>Target vehicle heading |
| Dynamic control | Steering angle<br>Target vehicle heading | Error feedback<br>Vehicle steering wheel angle |

and finally, the dynamic control stage calculates time transient steering wheel angles to control the vehicle for best lateral error. Notice steering angle and steering wheel angles are different, which will be precisely defined in the Section 4.

The input and output data at each stage of the control system are summarized in Table 2. The data passed to the next stage after post-processing are highlighted in bold.

## 2.2. Waypoint ground truth data

Path tracking for an autonomous vehicle requires the provision of route information. In specific environments such as the DARPA Grand Challenge or autonomous driving test cities, path coordinates are provided through the Route Definition Data File or high-definition maps. However, in general road settings where such specific infrastructure is unavailable, obtaining corresponding path coordinate information is challenging. Similarly, in this study, the target path within the Gwangju Institute of Science and Technology campus roads does not have pre-defined path coordinate data.

To address this challenge, this study collected and processed path coordinates (ground truth) directly using the GNSS mounted on the vehicle. The acquired target path 1 and target path 2 are overlaid on satellite imagery, as shown in Figures 3A and B, respectively.

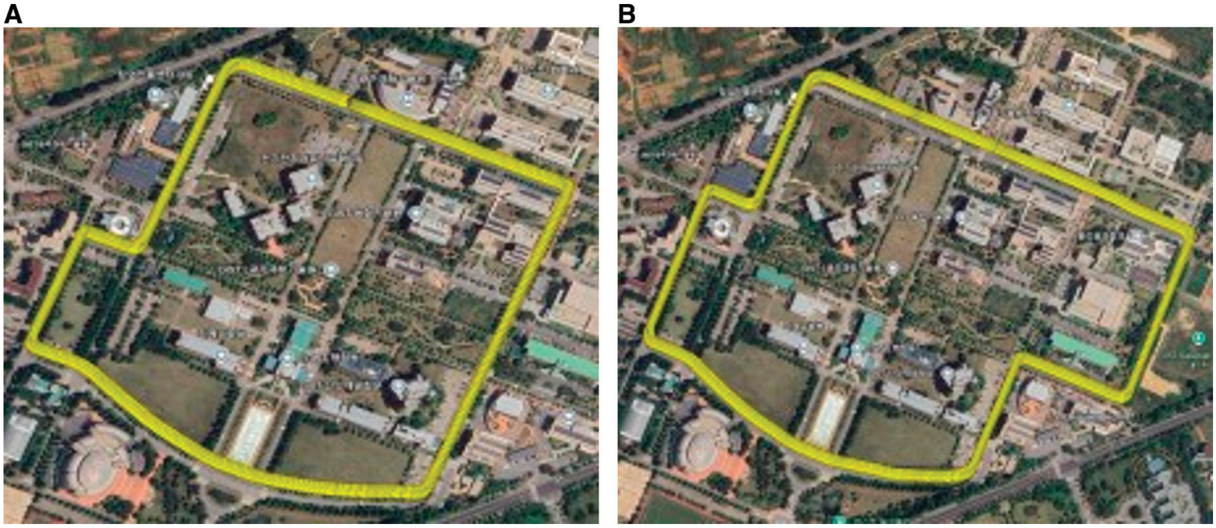


**Figure 3:** (A) Driving target path 1. (B) Driving target path 2.

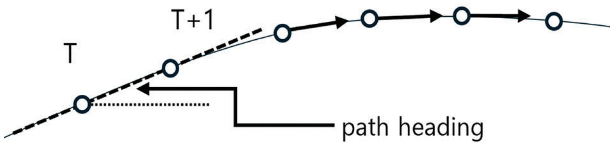


**Figure 4:** Path heading representation method of Campbell (2007).

The procedures for acquiring these target paths are as follows:

(1) Activate the GNSS and apply RTK correction.
(2) With the GNSS activated, manually drive along the desired path.
(3) Sampling the coordinates provided by the GNSS at uniform time intervals (e.g., every 0.1 seconds (10Hz)).
(4) Perform post-processing on the recorded coordinate data.

To obtain sufficiently dense coordinate data, the driving speed was limited so the vehicle was driven slowly along the path. The coordinate recording interval was also adjusted to 0.1 seconds (10Hz) and 0.05 seconds (20Hz), and multiple measurements were performed. The path coordinate data obtained in this manner consists of a set of discretized points.

### 2.2.1. *Path coordinates post-processing*

The raw data provided by the GNSS is expressed in latitude and longitude coordinates in degrees (°). Since the path tracking algorithm incorporates the distance error (m) from the path and the current vehicle speed (m/s), it is appropriate to use meters (m) rather than degrees (°). To address this, Universal Transverse Mercator conversion was performed. In South Korea, the 52nd UTM zone is used, with the reference origin of the coordinate system being at longitude 129° and latitude 0°.

The acquired path coordinates are a set of discretized points. An index was assigned to each point, and the path heading for each point was determined.

As illustrated in Figure 4, Campbell (2007) ees that to represent a path flexibly, it should be expressed in a piecewise linear manner. Each point is represented by linear segments, and each linear segment is expressed using sufficiently dense first or second-order polynomials to represent the path as a smooth trajectory.

When applying this path representation method, the path heading can be obtained by calculating the slope of each linear segment and applying the necessary operations to express the path heading in degrees (°). However, there are two issues with this approach. The system intermittently provides inaccurate path heading information. As shown in Figure 5, occasional outliers were observed.

When examining the cases where outliers occurred in Figure 5, it was observed that the movement along the x-axis was disproportionately smaller than the movement along the y-axis, or vice versa, with measurements below 0.01 m. As a result, the slope between the two path coordinates was calculated to be close to 0 or 1. This typically happens when the vehicle is stopped or moving at a very slow speed.

Secondly, the method of obtaining the slope between two adjacent path coordinates and calculating path heading differs from directly obtaining the vehicle heading through actual GNSS heading data, as summarized in Table 3. This difference requires unnecessary post-processing steps.

### 2.2.2. *Selected path coordinate data*

As we have previously stated the problems, the direction of the path is not obtained by calculating the slope and applying additional operations, but instead, the GNSS heading is measured simultaneously with the path coordinates. This approach eliminates the need for post-processing as soon as the path coordinates are acquired, allowing for direct application in the path tracking algorithm. In summary, the selected path consists of path coordinates measured at 0.1-second intervals, and no outliers were observed in the path heading measurements. Table 4 summarizes the information for the two selected target path coordinates with the two paths driven in opposite directions. The target test path coordinate data used in this study is available for free download at the provided GitHub repository (https://github.com/CallmeBOKE/Development-of-a-basic-GNSS-based-lateral-control-system-for-autonomous-vehicles). We anticipate this availability of the test paths will enable researchers to re-implement our work in their researches for direct comparisons and improvements.

Figures 6 and 7 visualize the trajectory and heading information of the first selected target test driving path, while Figure 8

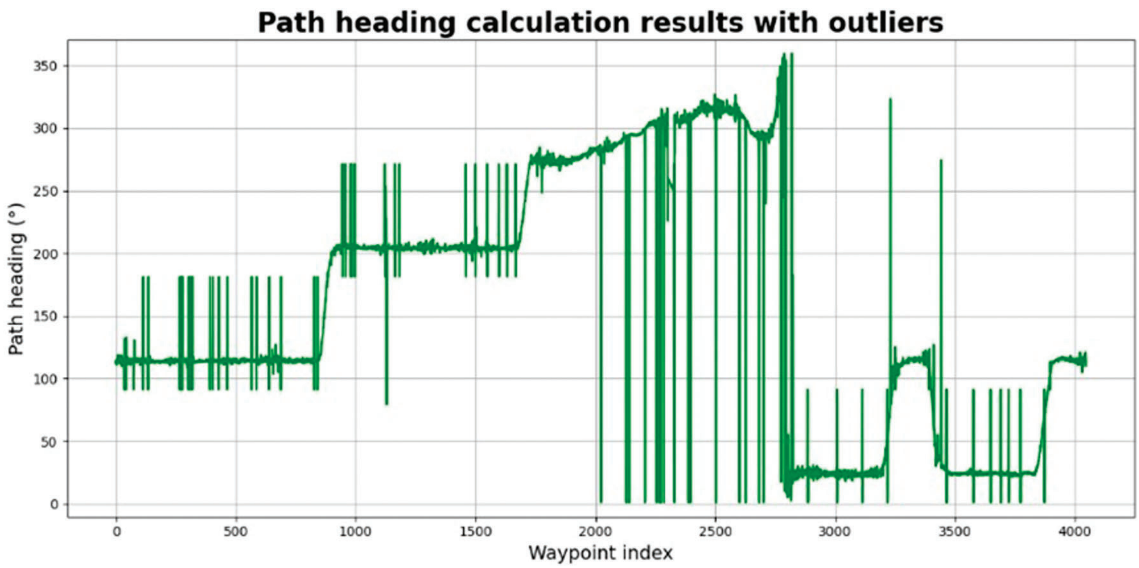


**Figure 5:** Path heading calculation results with outliers.

**Table 3:** Difference between GNSS heading and path heading.

| GNSS heading | Path heading |
|---|---|
| The **true north** direction is set as the origin (0°) | The **true east** direction is set as the origin (0°) |
| The angle increases in a **clockwise** direction | The angle increases in a **counterclockwise** direction |
| Conclusion: 90° phase difference with opposite signs | |

**Table 4:** Overview of the two selected target paths.

| | Target path 1 | Target path 2 |
|---|---|---|
| Total waypoints | 4 864 | 6 336 |
| Total number of straight sections | 5 | 8 |
| Total number of turning sections | 7 | 9 |
| Continuous turning section | 1 | |

and 9 provide additional information about the same driving path, specifically visualizing the pitch and altitude of the driving trajectory. Figures 10 and 11 visualize the trajectory and heading information of the second selected target driving path followed by Figures 12 and 13, which provide additional information about the second driving path by visualizing the pitch and altitude of the driving trajectory. The pitch information of the path shows abrupt changes in the slope at 17 and 19 points, respectively. These changes correspond to the number of speed bumps on each driving path and occur temporarily when passing over these bumps. Additionally, by examining the altitude information, it can be inferred indirectly that the driving path has a terrain profile that is “higher in the west, lower in the east.”

In Figures 8 and 12, the pitch information of the path shows abrupt changes in the slope at 17 and 19 points, respectively. These changes correspond to the number of speed bumps on each driving path and occur temporarily when passing over these bumps. Additionally, by examining the altitude information in Figures 9 and 13, it can be inferred indirectly that the driving path has a terrain profile that is “higher in the west, lower in the east.”

## 3. Quasi-static Control Stage

In the quasi-static control stage, the vehicle’s position, heading, and velocity data obtained in the positioning stage are used to minimize the deviation with respect to the given path coordinates and calculate the steering angle required for the vehicle to follow the target path.

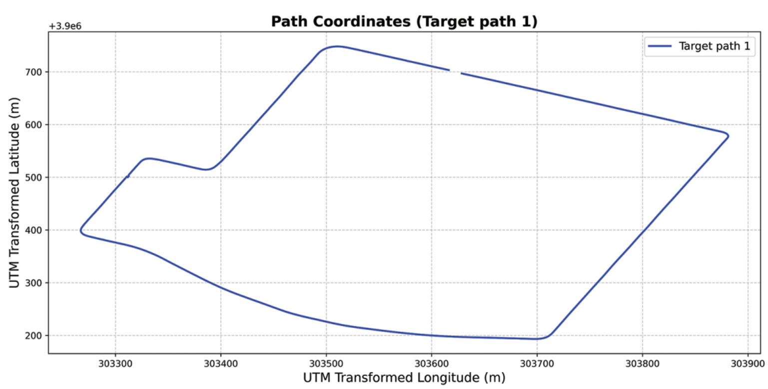


**Figure 6:** Trajectory of the selected target path 1.

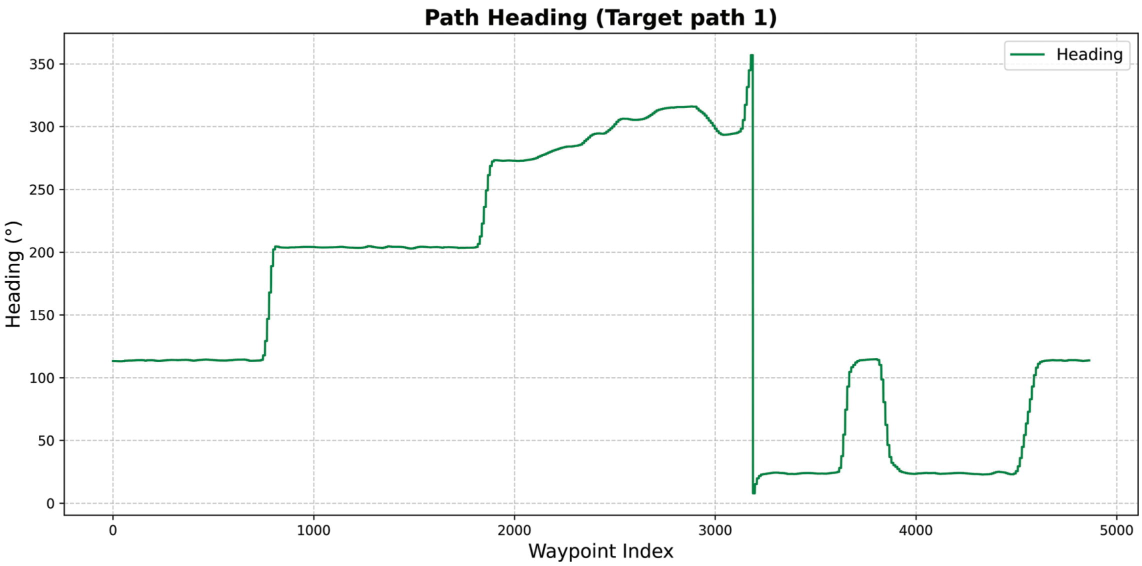


Figure 7: Waypoint heading of the selected target path 1.

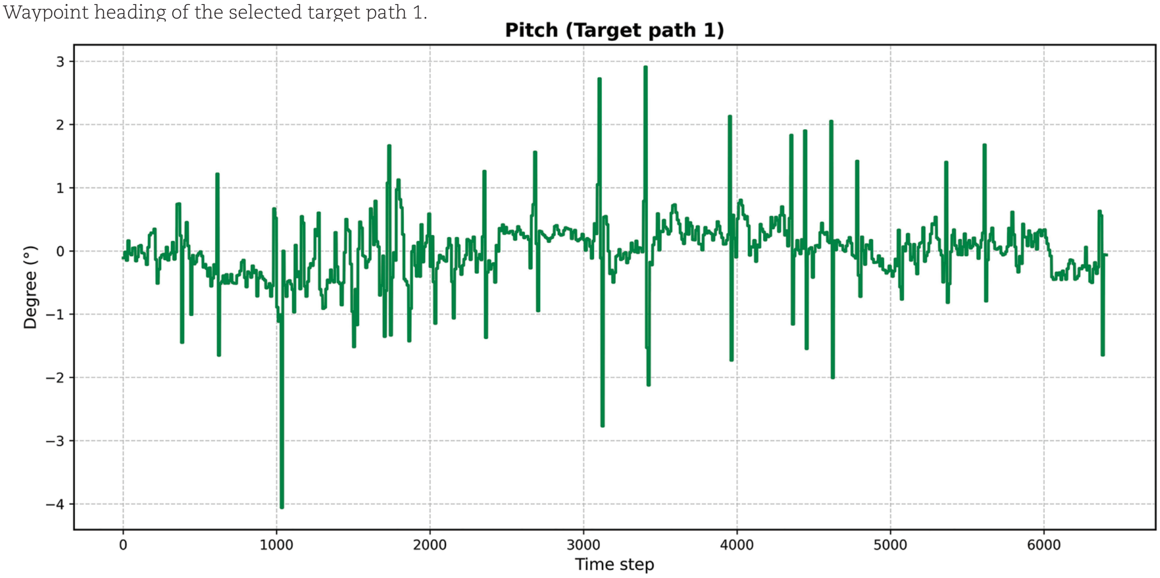


Figure 8: Pitch information of the selected target path 1.

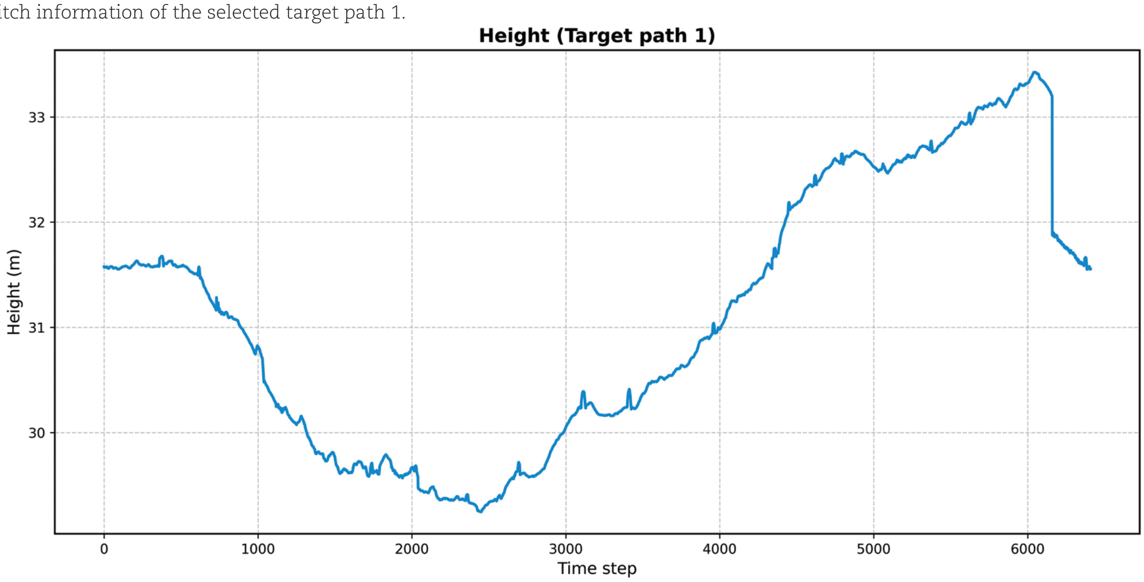


Figure 9: Height information of the selected target path 1.

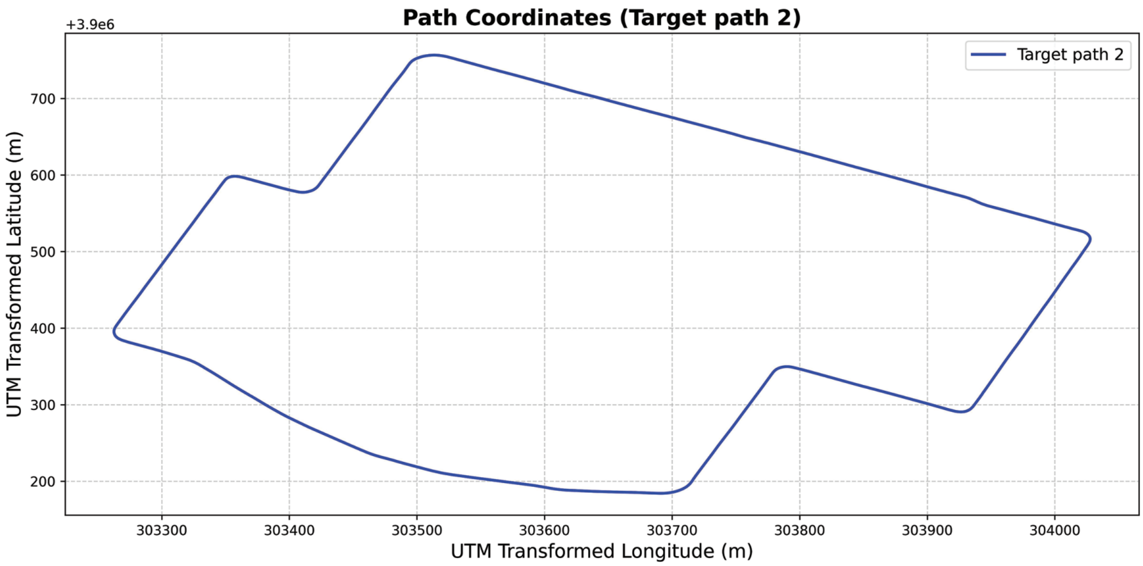


**Figure 10:** Trajectory of the selected target path 2.

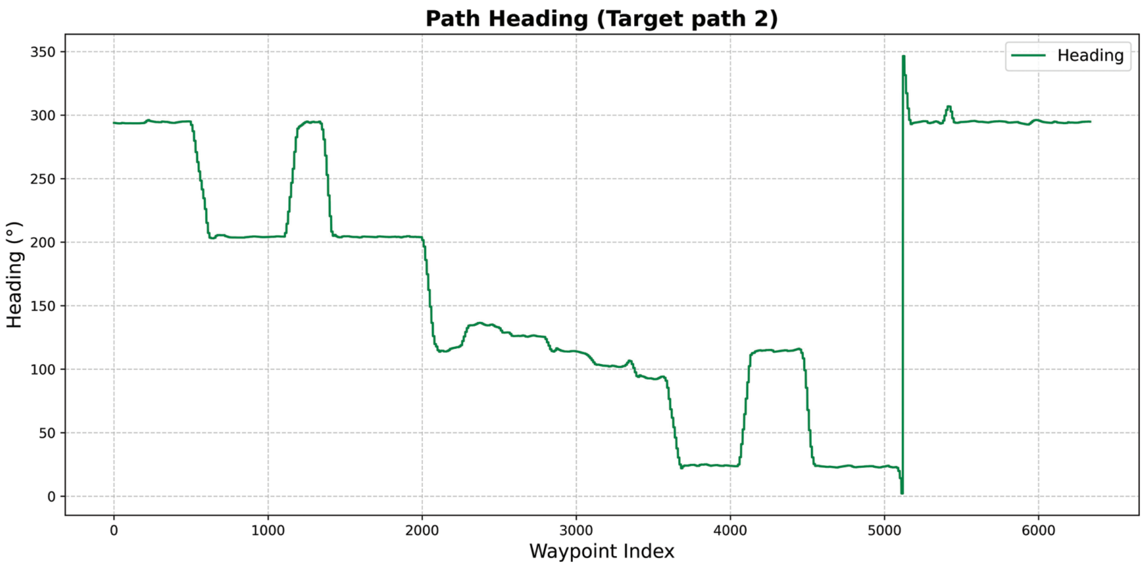


**Figure 11:** Waypoint heading of the selected target path 2.

## 3.1. Selection of a path tracking algorithm

In the quasi-static control stage, path tracking is the key component, enabling the system to compute steering angles that minimize deviation along the target path. The path tracking algorithm was implemented using the Stanley method. The rationale for adopting the Stanley method is as follows:

(1) The proposed system utilizes the vehicle's current position, velocity, and heading for path tracking. The Stanley method is a geometric path tracking algorithm that can effectively incorporate all of this information.
(2) The Stanley method, proposed in the research that won the 2005 DARPA Grand Challenge, is a proven control algorithm. It is widely used as a standard algorithm for vehicle lateral control.
(3) Unlike other geometric path tracking algorithms, the Stanley method determines the steering angle by incorporating both the heading error and the cross-track error (the perpendicular offset from the path).
(4) By calculating the heading error and cross-track error at regular intervals during the driving process, and computing the corresponding steering angle, the Stanley method ensures stable driving performance.

## 3.2. Detailed explanation of the stanley method

In the Stanley method, the vehicle is simplified and represented using the kinematic bicycle model. This model reduces the vehicle's four wheels to two virtual wheels and does not consider forces or moments when analyzing the vehicle's motion. The kinematic bicycle model assumes that the velocity vector of the wheels and the direction vector of the wheels are always aligned, neglecting the occurrence of slip angles and thus disregarding the lateral forces acting on the tyres. We assume operation in a low-slip regime where lateral tyre forces and slip angles are negligible, and thus the kinematic bicycle model is valid. Under these condi-

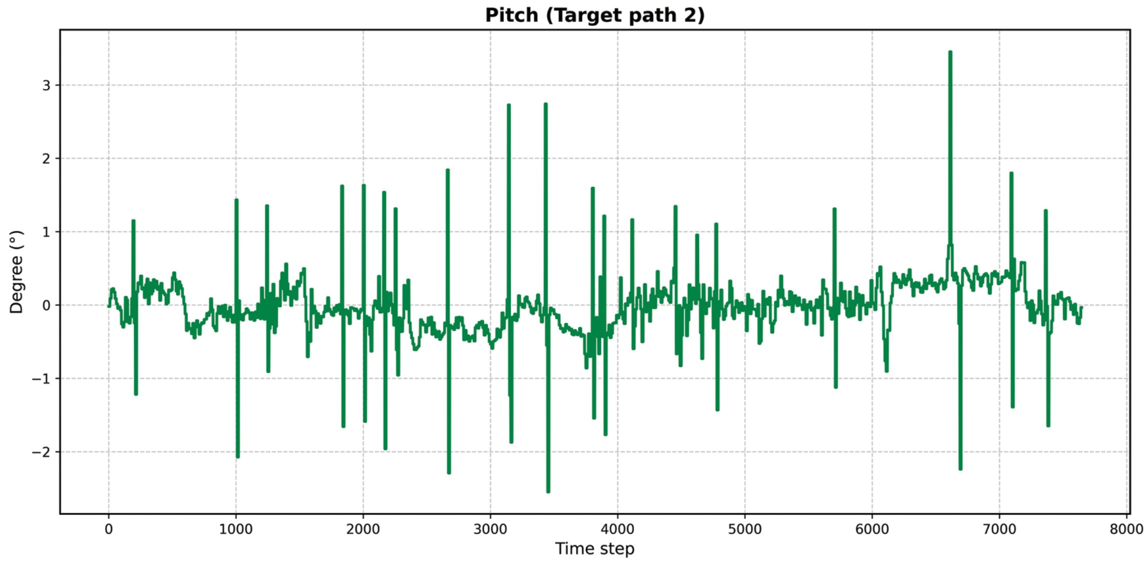


**Figure 12:** Pitch information of the selected target path 2.

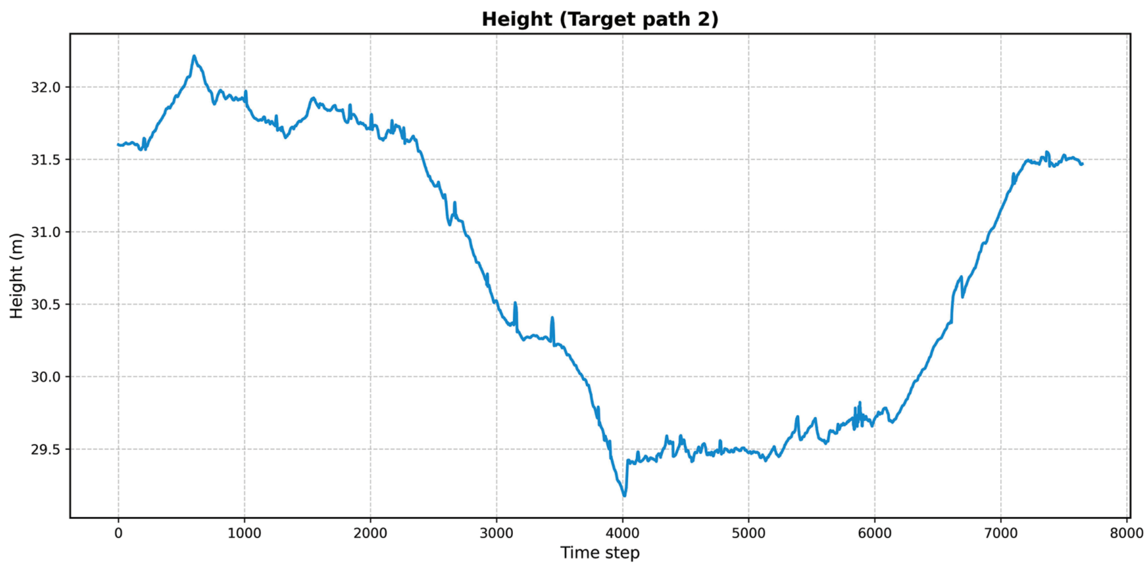


**Figure 13:** Height information of the selected target path 2.

tions, heading and cross-track error geometry dominate steering response, which justifies the use of the Stanley Method.

Therefore, this model is valid when the vehicle's speed is low, meaning that the lateral forces on the tyres are minimal.

As shown in Figure 14, the Stanley method determines the steering angle from the center of the vehicle front axle, and the steering angle is calculated by incorporating both the heading error and the cross-track error.

The heading error, as shown in Figure 14 represents the difference between the path heading and the vehicle's heading, indicated by the green dashed line. A negative value (–) is assigned when the vehicle is required to make a left turn, while a positive value (+) is used when the vehicle is required to make a right turn.

The cross-track error is defined as the perpendicular distance from the center of the vehicle's front axle to the closest point on the path. This distance is represented by the yellow solid line in the figure. The influence of cross-track error on the total steering angle can be adjusted by modifying the control gain. The control gain is predetermined, and in this study, it is set to a value of 0.55. This value was determined through iterative closed-loop testing across 100 trial runs at a speed of 10 km/h. Gain was selected based on minimum integral error across two path segments and verified against path-specific responsiveness.

### 3.2.1. *Application Strategy of the Stanley Method*

Considering that the path coordinates obtained in the preceding positioning stage are discretized, the Stanley method should be applied accordingly.

#### Target Point Searching

Since the vehicle's position and heading information are acquired through the positioning stage, the coordinates of the center of the vehicle's front axle can also be determined. In the vehicle used in this study, the center of the front axle is located 2.7 m in front of the GNSS primary antenna. Based on the coordinates

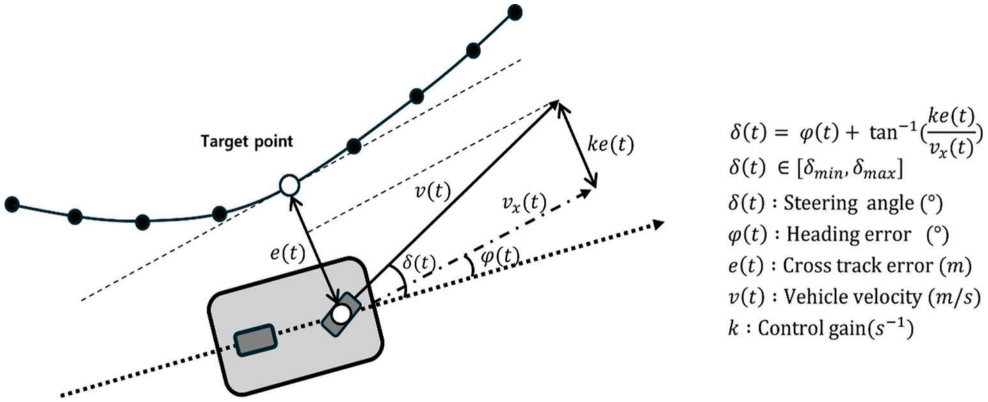


**Figure 14:** The Stanley method.

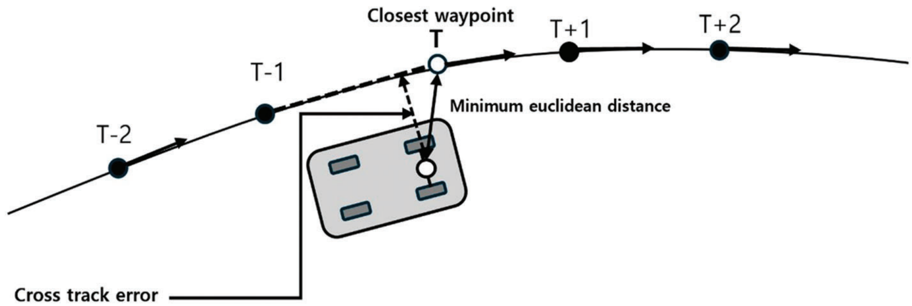


**Figure 15:** Application of the Stanley method on a discretized path.

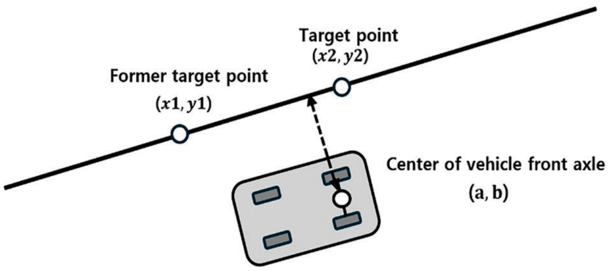


**Figure 16:** Cross-track error calculation.

of the center of the front axle, the closest path point, identified by the minimum Euclidean distance, was selected as the target point. Additionally, the index of the target point is recorded to ensure that any path coordinates with a lower index than the target point are not designated as the target point, thereby preventing previously passed path coordinates from being selected as the target point. The process described above is illustrated in Figure 15.

## Error Calculation

Each path coordinate includes the path heading information, so the difference between the vehicle's current heading and the path heading is defined as the heading error.

Unlike the calculation of heading error, calculating cross-track error requires more complex processes. As illustrated in Figure 16, the key is to obtain the equation of the straight line passing through target point and the immediately preceding path coordinate (former target point) and then calculate the distance between the center of the vehicle's front axle and this straight line.

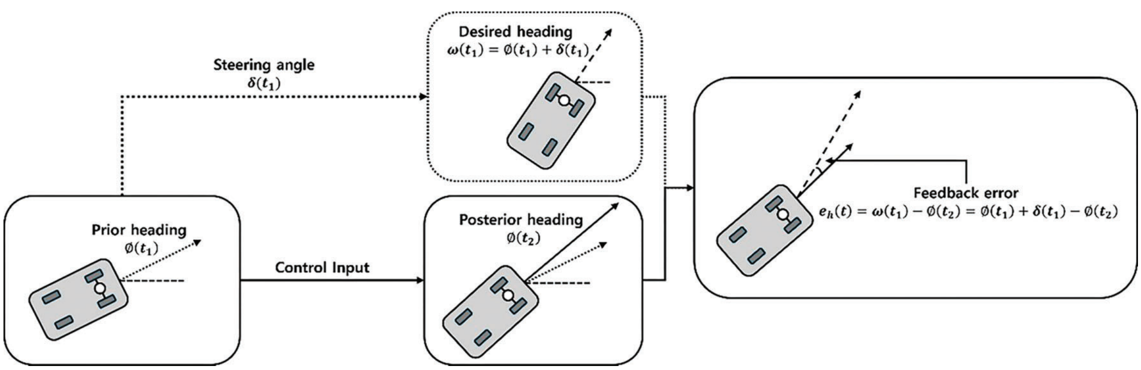


**Figure 17:** Overview of PID Controller.

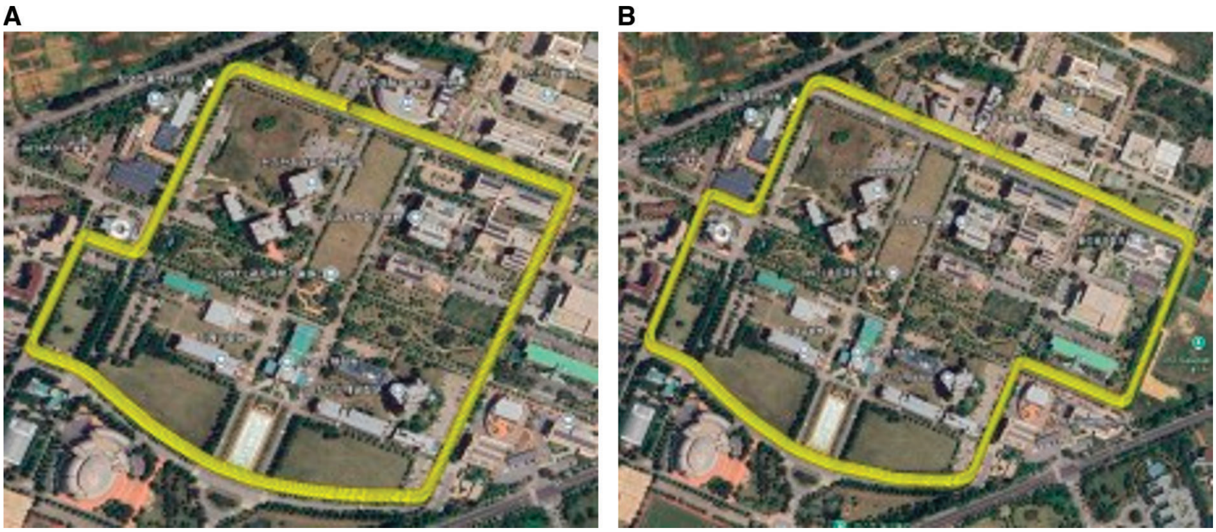


**Figure 18:** (A) Target path 1. (B) Target path 2.

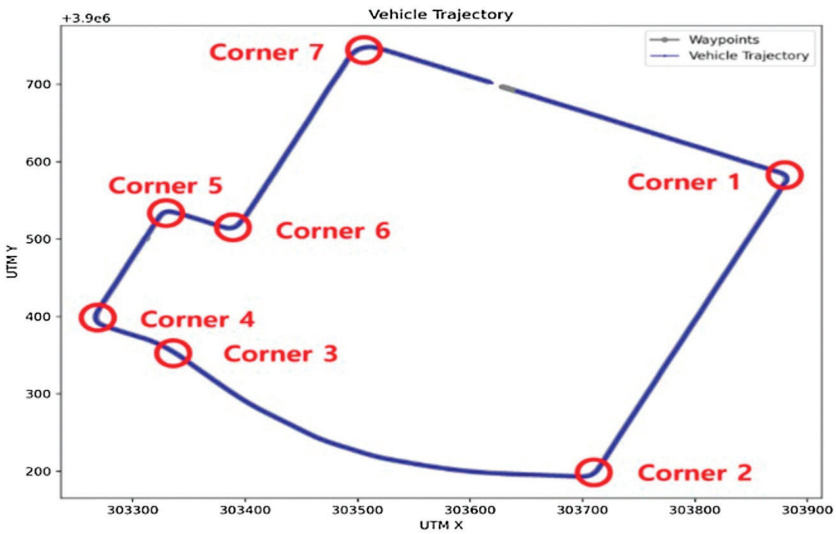


**Figure 19:** Vehicle trajectory.

(1) Obtain the equation of the straight line passing through the current target point and the former target point:

$$y = \frac{y_2 - y_1}{x_2 - x_1}(x - x_2) + y_2, \tag{1}$$

$$(y_2 - y_1)x + (x_1 - x_2)y + (y_1x_2 - x_1y_2) = 0. \tag{2}$$

(2) Calculate the distance between the coordinate of the center of vehicle front axle and the obtained equation of the straight line.

Distance between $P(a, b)$ and $(y_2 - y_1)x + (x_1 - x_2)y + (y_1x_2 - x_1y_2) = 0$

$$d = \frac{|(y_2 - y_1)a + (x_1 - x_2)b + (y_1x_2 - x_1y_2)|}{\sqrt{(y_2 - y_1)^2 - (x_1 - x_2)^2}}, \tag{3}$$

(3) Determine the sign by distinguishing between left and right turn situations.

Cross-track error (e(t))

$$= \begin{cases} +\dfrac{|(y_2 - y_1)a + (x_1 - x_2)b + (y_1x_2 - x_1y_2)|}{\sqrt{(y_2 - y_1)^2 - (x_1 - x_2)^2}} \\ -\dfrac{|(y_2 - y_1)a + (x_1 - x_2)b + (y_1x_2 - x_1y_2)|}{\sqrt{(y_2 - y_1)^2 - (x_1 - x_2)^2}} \end{cases}. \tag{4}$$

## 4. Dynamic Control Stage

The Stanley method determines the geometric path-following angle based on current heading and cross-track error. However, steering actuators may introduce latency or nonlinearity in their response. To address this, we layer a PID controller in the dynamic stage to minimize the error between desired and actual heading change, functioning as a feedback compensator.

In the dynamic control stage, the appropriate control input (steering wheel angle) is calculated, and the control input is transmitted to the vehicle through the Control Area Network. The steering control of the vehicle used in this research is managed by an

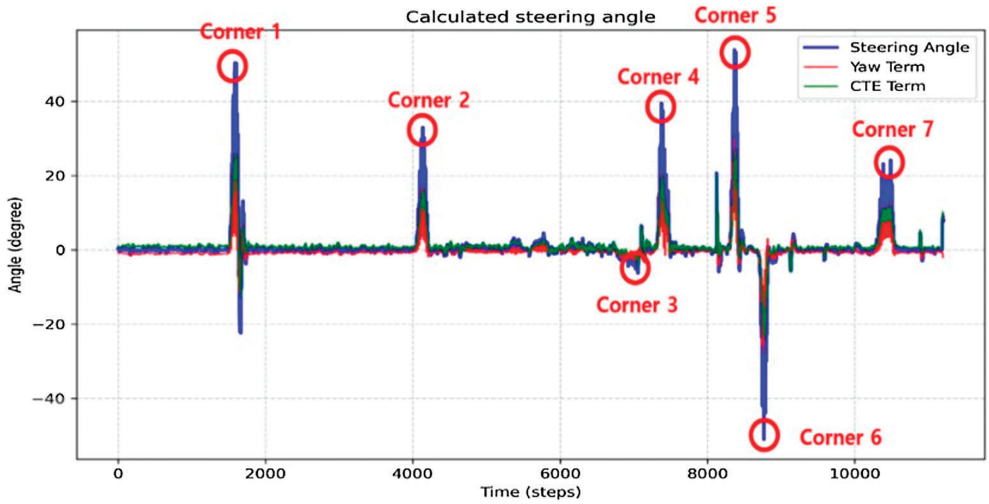


**Figure 20:** Steering angle calculated by the Stanley method.

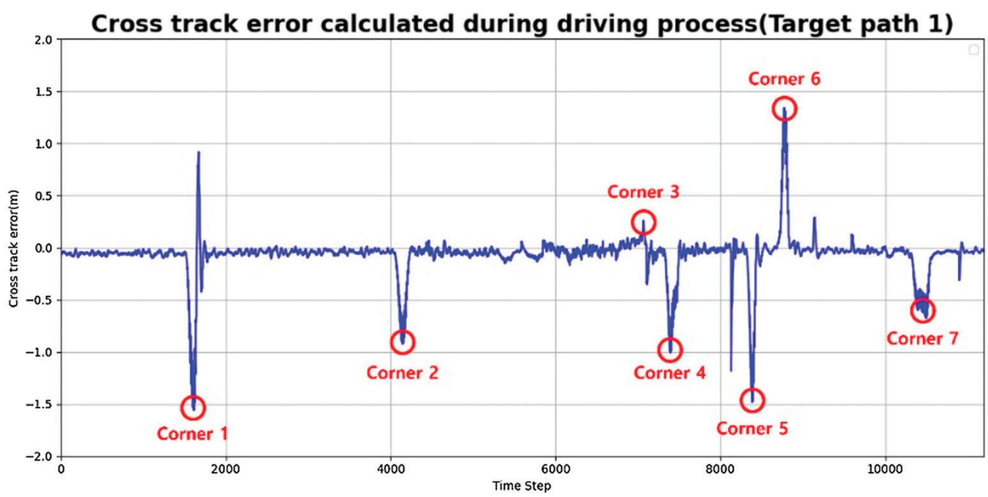


**Figure 21:** Cross-track error calculated during the driving process.

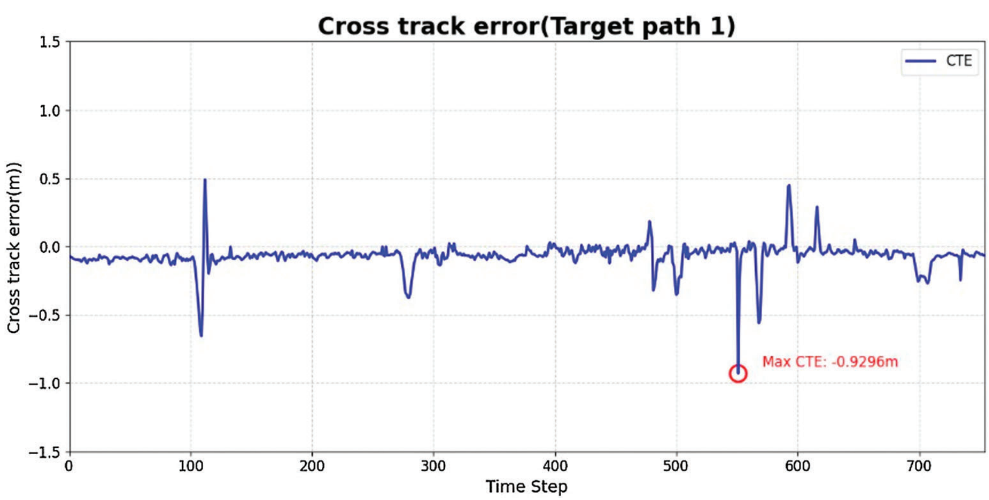


**Figure 22:** Cross-track error calculated after the driving process.

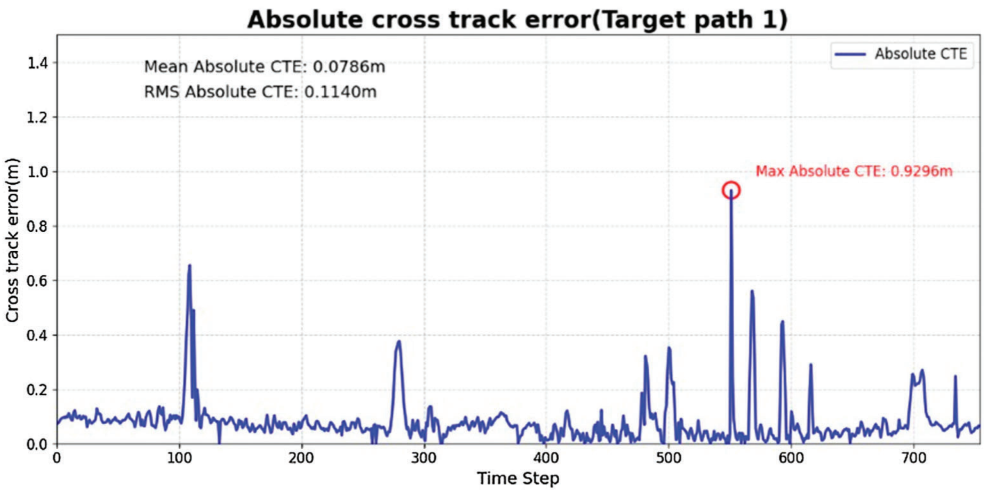


**Figure 23:** Absolute cross-track error calculated after the driving process.

**Table 5:** MAE and RMSE of target path 1.

| MAE (m) | RMSE (m) |
|---|---|
| 0.0786 | 0.1140 |

Automatic Power Steering Module, which manipulates the vehicle's steering wheel. The steering wheel angle is the control input, and the APM has a maximum steering wheel angle of ±450°. The computed steering wheel angle is received via CAN.

In the quasi-static control stage, the steering angle obtained through the Stanley method indicates that the vehicle's orientation ($\emptyset(t)$, vehicle heading) should rotate by the steering angle $\delta(t)$ such that the vehicle heading reaches the desired value $\emptyset(t) + \delta(t)$. However, the system controls the steering wheel, and it is not straightforward to determine the change in vehicle heading based on the steering wheel angle. One way to address this is by measuring the steering ratio to determine the vehicle's wheel rotation angle in relation to the steering wheel angle. However, this is typically done in a static environment (i.e., with the vehicle completely stationary), making it unsuitable for real road driving conditions. Moreover, the dynamic PID layer complements the geometric control by correcting for actuator delays and residual heading errors. It functions as a stabilizing feedback mechanism, ensuring smooth convergence even under real-world disturbances.

Therefore, in the dynamic control stage, a PID controller is applied to minimize the gap between the steering wheel angle and the actual vehicle behavior.

## 4.1. PID Controller

The PID controller (Johnson & Moradi, 2005) is widely adopted in practical systems for its simplicity and effectiveness. The PID controller compares the actual output of the controlled system with the desired output after the control input is applied, calculates the error ($e_h(t)$), and then feeds this error back into the control input for correction:

$$\text{Control Input} = K_p e_h(t) + K_i \int_0^t e_h(t)\,dt + K_d \frac{de_h(t)}{dt}. \tag{5}$$

In the proposed system, the initial steering wheel angle is set to the steering angle value obtained from the previous quasi-static stage. This initial steering wheel angle is then applied, and the error between the actual vehicle behavior and the desired behavior is fed back to determine the final steering wheel angle. The process is described in detail as follows and illustrated in Figure 17:

(1) In the path tracking stage, the steering angle ($\delta(t_1)$) calculated by the Stanley method is transmitted to the vehicle control node to initialize the steering wheel angle.
(2) Simultaneously with the calculation of the steering angle, the vehicle's direction is measured to define the prior heading ($\emptyset(t_1)$).
(3) By adding the prior heading and the calculated steering angle, the desired heading ($\omega(t_1)$) is defined.

$$\omega(t_1) = \emptyset(t_1) + \delta(t_1) \tag{6}$$

(4) By initializing the steering wheel angle with the calculated steering angle, the initial steering control is performed by transmitting the steering wheel angle to the APM via CAN.
(5) After the initial steering control is performed with the initialized steering wheel angle, the vehicle's heading is measured again to define the posterior heading ($\emptyset(t_2)$).
(6) The feedback error ($e_h(t)$) is the difference between the desired heading and the posterior heading.

$$e_h(t) = \omega(t_1) - \emptyset(t_2) = \omega(t_1) = \emptyset(t_1) + \delta(t_1) - \emptyset(t_2) \tag{7}$$

The final control input ($I_h(t)$) is as follows.

$$I_h(t) = K_p(\emptyset(t_1) + \delta(t_1) - \emptyset(t_2)) + K_i \int_{t_1}^{t_2} \emptyset(t_1) + \delta(t_1) - \emptyset(t_2)\,dt + K_d \frac{d(\emptyset(t_1) + \delta(t_1) - \emptyset(t_2))}{dt}. \tag{8}$$

PID gains were determined using an iterative closed-loop procedure inspired by the Ziegler–Nichols method. Critical oscillation was identified, and corresponding gain/period values were used to determine the initial settings. These values were then adjusted conservatively in real-world trials to ensure stability.

Gains were selected based on minimum integral error across two path segments and verified against path-specific responsiveness:

$$K_p = 6.5,\ K_i = 0.2,\ K_d = 0.7. \tag{9}$$

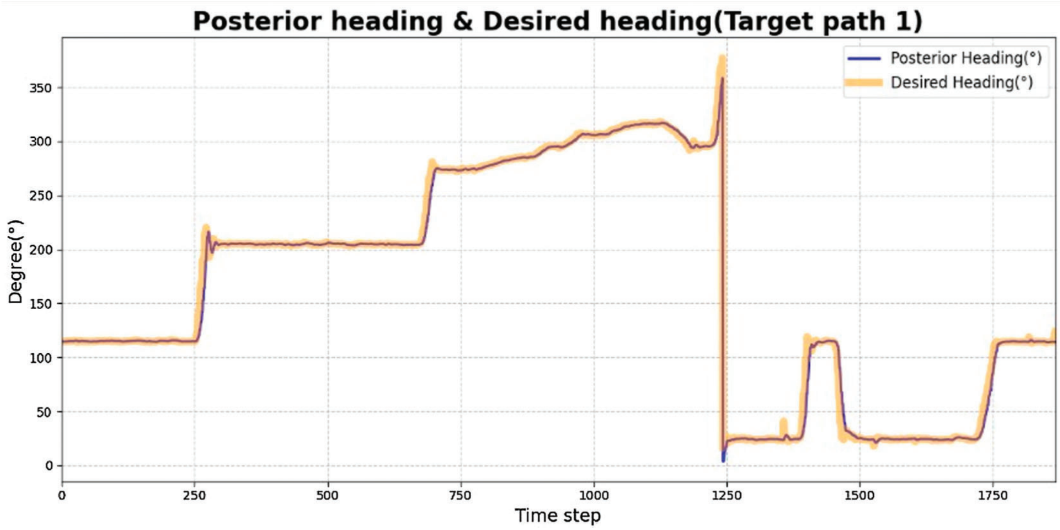


**Figure 24:** Desired heading and posterior heading.

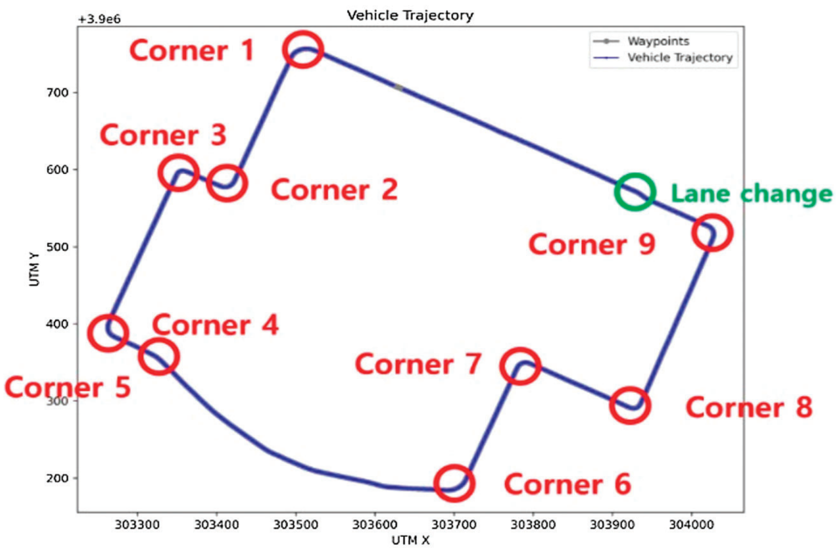


**Figure 25:** Vehicle trajectory.

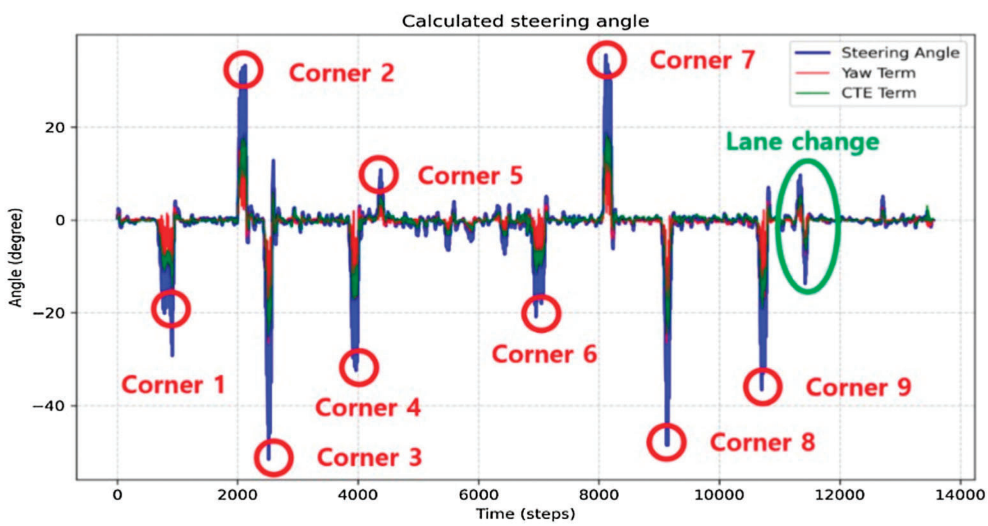


**Figure 26:** Steering angle calculated by the Stanley method.

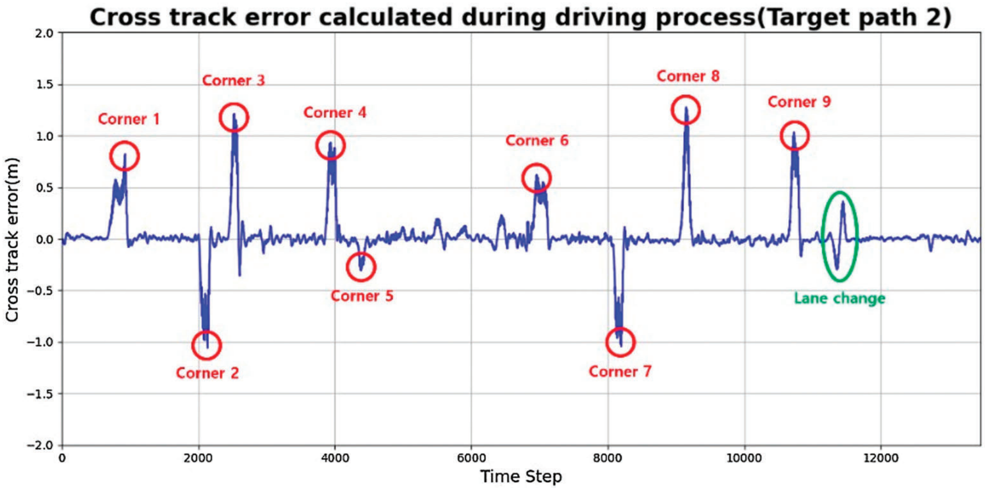


**Figure 27:** Cross-track error calculated during the driving process.

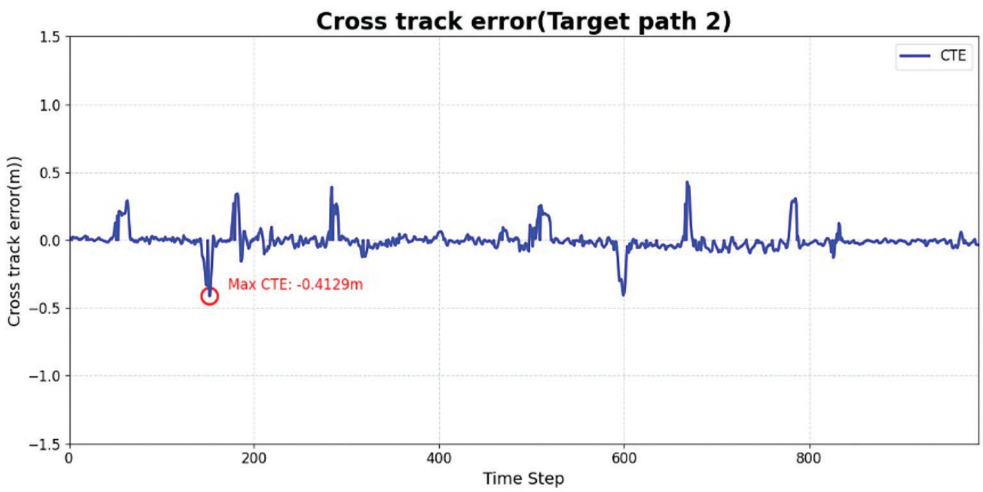


**Figure 28:** Cross-track error calculated after the driving process.

## 5. Experiments

The proposed system was validated by driving the vehicle along two different paths on the campus of Gwangju Institute of Science and Technology. The vehicle was driven along each path, with measurements taken at 0.1-second intervals. Target Path 1 and Target Path 2 are overlaid on satellite imagery, as shown in Figures 18A and B, respectively.

All experiments were performed on dry asphalt campus roads. The adhesion coefficient $\mu$ for dry asphalt typically ranges between 0.7 and 0.8, providing reliable tyre-road contact. Additionally, the vehicle's speed was limited to not exceed 10 km/h during the driving tests. For target path 1, the vehicle was driven in a clockwise direction when viewed from the top, whereas for target path 2, the vehicle followed a counterclockwise direction. The factors to be validated through the driving experiments are as follows:

(1) The completion of the target path:

The successful completion of the target path is used to validate the feasibility of the proposed system.

(2) Path tracking performance

During the driving process, the cross-track error and heading error are computed and minimized to verify whether the vehicle converges to the desired position and heading.

(3) To validate the appropriateness of processing the path heading and the vehicle heading information.

By driving each path in the opposite direction, the appropriateness of the method used in this study to obtain the path heading is verified. Additionally, it is confirmed whether the signs determined differently in the left and right turn situations of the path tracking algorithm are correct.

## 6. Result and Discussion

### 6.1. Target path 1

The proposed system completed target path 1, and the driving results are as follows.

Figure 19 shows the trajectory of the vehicle recorded during the driving process of target path 1, while Figure 20 displays the steering angle calculated by the Stanley method. The following is the cross-track error calculated in the quasi-static control stage.

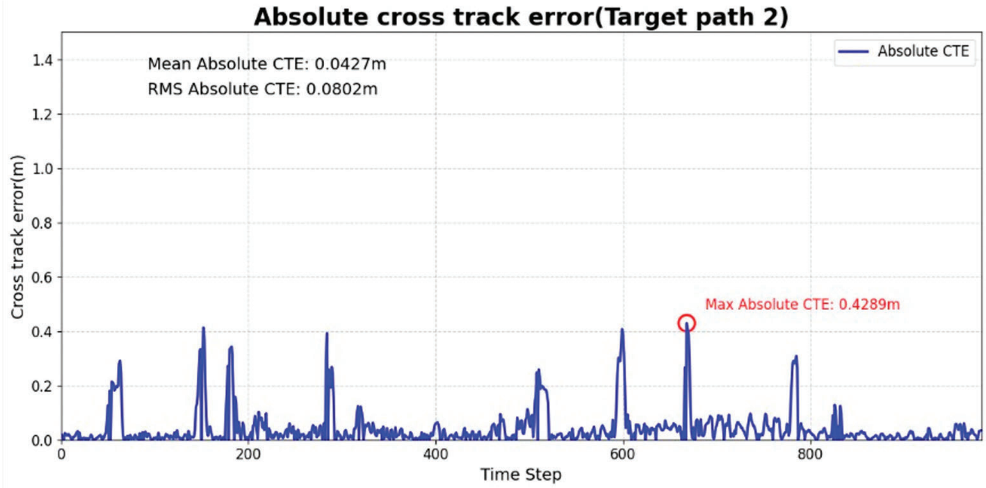


**Figure 29:** Absolute cross-track error calculated after the driving process.

**Table 6:** MAE and RMSE of target path 2.

| MAE (m) | RMSE (m) |
|---|---|
| 0.0427 | 0.0802 |

As illustrated in Figures 21–23 and summarized in Table 5, the cross-track error to the path was recalculated based on the coordinates of geometric center of the vehicle body, recorded during the real-world experiments. MAE was calculated as 0.0786 m, and RMSE was calculated as 0.1140 m. The maximum cross-track error was −0.9296 m.

In the dynamic control stage, the convergence between the desired heading and the posterior heading immediately after the control input was also evaluated, with an average error of 2.1411°. The difference between the desired heading and the posterior heading recorded during the driving experiment can be observed in Figure 24.

## 6.2. Target path 2

The proposed system completed target path 2, and the results of the driving are as follows.

Figure 25 shows the trajectory of the vehicle recorded during the driving process of target path 2, while Figure 26 displays the steering angle calculated by the Stanley method. The following is the cross-track error calculated in the quasi-static control stage.

For the case of driving target path 2, as illustrated in Figures 27–29 and summarized in Table 6, MAE was calculated as 0.0427 m, and RMSE was calculated as 0.0802 m. The maximum cross-track error was 0.4289 m.

In the dynamic control stage, the convergence between the desired heading and the posterior heading immediately after the control input was also evaluated, with an average error of 2.5000°. The difference between the desired heading and the posterior heading recorded during the driving experiment can be observed in Figure 30.

In the case of cross-track error, it was observed that the error temporarily increased at the starting point of the turning section and quickly minimized thereafter. Figure 31 illustrates the vehicle's trajectory along with a rectangle matching the actual size of the vehicle, overlaid on a satellite map image, to check for lane violations.


**Figure 30:** Desired heading and posterior heading.

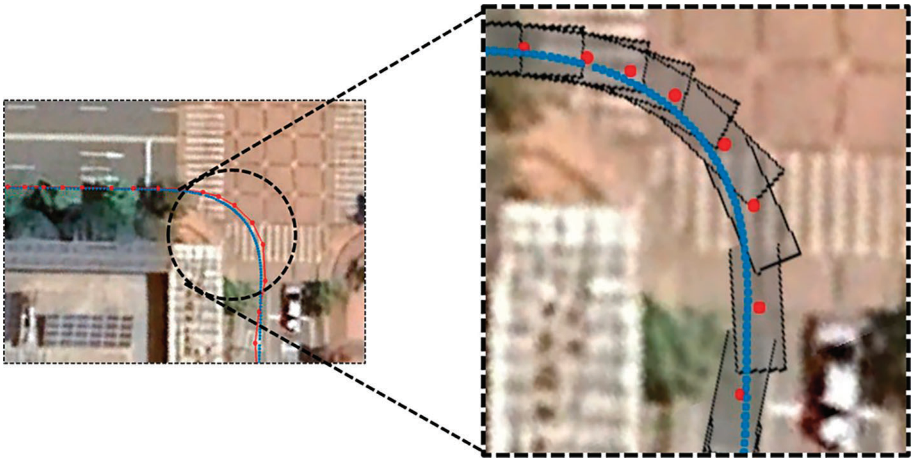

**Figure 31:** Temporary increase of cross-track error.

Before entering the turning section, the vehicle follows the straight path, and the cross-track error is minimized as 0 meter. However, as seen in Figure 31, the moment the vehicle enters the turning section, the slope of the path segment formed by the target path coordinates and the passed path coordinates changes drastically, causing both the heading error and cross-track error to rise sharply. This can also be observed in Figures 20 and 26, where the steering angle, heading error, and cross-track error all increase simultaneously upon entering the turning section.

Table 7 compares the lateral control system proposed in this study with those reported in previous research.

Only results obtained through actual real-world experiments were included. Cases lacking information on total travel distance were excluded from the analysis. As seen in Table 7, each study adopted different performance metrics such as maximum error, MAE, and RMSE. Furthermore, experiments were conducted on different trajectories and travel distances. These variations make direct performance comparison inherently challenging. To address the challenge of comparing across differing path lengths and shapes, we propose a normalized lateral error metric: error (m) divided by path length (m) × 1 000. Comparisons were limited to real-world studies that provide sufficient path information, including the total travel length. Because curvature information was rarely available, we report a dimensionless error normalized by path length. While this omits curvature dependence, it allows a conservative, reproducible comparison across heterogeneous routes. While this is not a standard metric, it provides a scale-invariant measure of controller performance, enabling approximate comparisons when trajectories differ in complexity. While limited, it enables conservative comparison. For identical routes, statistical tests such as paired t-tests or nonparametric Wilcoxon tests should be applied. Future benchmarking on our released datasets will incorporate such analyses.

Both the original and the normalized metrics are presented for comparison, and the best-performing results for each metric are highlighted in bold.

$$\text{Metric score} = \frac{\text{Former error metric (m)}}{\text{Total travel distance (m)}} \times 1000. \tag{10}$$

Based on this new formula, our method evaluated on path 2 exhibited the best on the normalized maximum error (0.1950), normalized MAE (0.0194), and normalized RMSE (0.0365)

## 7. Conclusions

This study was designed with a lateral control system consisting of three main stages: positioning, quasi-static control, and dynamic control. In the positioning stage, precise positioning was performed from RTK, and data was directly collected and processed to address the absence of the provided path coordinates. In the quasi-static stage, the Stanley Method was adopted as the path tracking algorithm and applied to the data collected. In the dynamic control stage, a PID controller was implemented to compensate for the mismatch between the steering wheel angle and the actual vehicle behavior, thereby performing lateral control.

Previous lateral control methods measured their maximum and mean lateral errors with different trajectory lengths. Our idea is that these errors should be normalized with respect to the total travel length. Based on this normalization we have shown that as evident from target path 2, our normalized lateral (cross-track) error shows the best among all compared past literature.

Future work includes extending tests to higher-speed and dynamic scenarios, integrating longitudinal control for complete vehicle stabilization, and formalizing curvature-aware error metrics.

Our experiments were limited to ∼10 km/h. Error spikes in turns stem from sharp curvature and actuator lag. We suggest mitigations such as gain scheduling and speed reduction near curves. Future work will address higher-speed environments and integrate longitudinal control.

1. Short-term driving performance

For the short-term driving performance, it is essential to assess how quickly the cross-track error and heading error are minimized. As observed in Section 6.1 and 6.2, it was confirmed that the cross-track error and heading error temporarily increased drastically at the moment of entering the turning section. There is a need to minimize these errors in the turning section to achieve more stable driving performance. Furthermore, it is necessary to determine the appropriate time for error minimization during the driving process. These are essential for driving safety.

2. Integration with longitudinal control system

**Table 7:** Performance comparison of lateral control systems.

| | Method | Maximum error: Original | Maximum error: Proposed metric² | MAE Original | MAE Proposed metric² | RMSE Original | RMSE Proposed metric² | Total travel distance | Remark |
|---|---|---|---|---|---|---|---|---|---|
| Dominguez et al.(2016) | Pure pursuit | 0.3600 m | 0.3600 | – | – | – | – | | |
| | Stanley method | 0.4000 m | 0.4000 | – | – | – | – | 1000 m | |
| | Sliding mode control | 0.4000 m | 0.4000 | – | – | – | – | | |
| | LatVel | **0.3000 m** | 0.3000 | – | – | – | | | |
| **Park et al. (2014)** | Adaptive Pure pursuit | 0.4000 m | 0.2857 | – | – | – | | 1400 m | |
| **Hossain et al. (2022)** | LQR + PID | – | – | 0.0953 m | 0.0226 | – | – | **4200 m** | |
| **Baksaas et al. (2021)** | Stanley | – | – | – | – | 0.1630 m | 0.2329 | 700.0 m | Summer |
| | Morin + PID | | | | | 0.2730 m | 0.3900 | | Winter |
| | Stanley + PID | −0.9296 m | 0.5160 | 0.0786 m | 0.0436 | 0.1140 m | 0.6333 | 1800 m | |
| **Ours** | | 0.4289 | **0.1950** | **0.0427 m** | **0.0194** | **0.0802 m** | **0.0365** | 2200 m | |

This table presents a comparative analysis of lateral control systems in terms of maximum error, mean absolute error (MAE), and root mean square error (RMSE). The value for each performance metric is highlighted in bold to indicate the best result. Dashes (–) indicate missing or unavailable data.

The lateral control system in this study and the longitudinal control system proposed can be integrated to control the vehicle's speed, thereby forming a comprehensive autonomous driving control system. We have not pursued the control of the longitudinal errors.

Moreover, because curvature information was generally absent in prior studies, we adopted path-length normalization as a conservative benchmark. Although this approach neglects curvature dependence, it enables reproducible comparisons. As future work, we plan curvature-weighted extensions and segment-wise reporting.

We also acknowledge the lack of testing in varied weather, dynamic traffic, or urban environments. The system is evaluated under low-speed, static conditions and may require modification for complex real-world deployment.

Despite the aforementioned limitations, this study has demonstrated the best quasi-static and dynamic lateral control. We also provided normalized error analysis so that comparison with other works can be made despite dealing with different paths. This foundational system provides a reproducible platform for benchmarking lateral control strategies and may assist future work in integrated autonomous driving systems.

## Conflicts of Interest

The authors declare no conflict of interest.

## Author Contributions

**Hong Seung Kim:** Conceptualization, Methodology, Software, Data curation, Writing—original draft. **Yong-Gu Lee:** Writing—review & editing, Supervision, Validation.

## Funding

This work was conducted by Korea Institute for Advancement of Technology (KIAT) grant funded by the Korea Government (MOTIE; P0020535, The Competency Development Program for Industry Specialist), and GIST Research Project grant funded by the GIST in 2024. And this research was supported by the Regional Innovation System & Education (RISE) program through the (Gwangju RISE Centre), funded by the Ministry of Education (MOE) and the (Gwangju Metropolitan City), Republic of Korea (2025-RISE-05-001).

## Data Availability

A downloadable link to the target test data is provided in Section 2.2.2.

## References

Ahmad, N., Ghazilla, R. A. R., Khairi, N. M., & Kasi, V. (2013). *Reviews on various inertial measurement unit (IMU) sensor applications*. *International Journal of Signal Processing Systems*, **1**, 256–262. https://doi.org/10.12720/ijsps.1.2.256-262

Ahn, J., Jung, S., Kim, H., Hwang, H.-J., & Jun, H.-B. (2023). A study on unmanned combat vehicle path planning for collision avoidance with enemy forces in dynamic situations. *Journal of Computational Design and Engineering*, **10**, 2251–2270. https://doi.org/10.1093/jcde/qwad099

Amer, N. H., Zamzuri, H., Hudha, K., Aparow, V. R., Kadir, Z. A., & Abidin, A. F. Z. (2018). Path tracking controller of an autonomous armoured vehicle using modified Stanley controller optimized with particle swarm optimisation. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, **40**, 1–17. https://doi.org/10.1007/s40430-017-0945-z

Andersen, H., Chong, Z. J., Eng, Y. H., Pendleton, S., & Ang, M. H. (2016). *Geometric path tracking algorithm for autonomous driving in pedestrian environment. In 2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*(pp. 1669–1674). IEEE. https://doi.org/10.1109/aim.2016.7577010

Baksaas, M., Olsen, L. E., & Mathiassen, K. (2021). *Evaluation of Two Path Following Controllers for an Ackermann Off-road Vehicle in Winter and Summer Conditions. In 2021 7th International Conference on Mechatronics and Robotics Engineering (ICMRE)*(pp. 121–130). IEEE. https://doi.org/10.1109/icmre51691.2021.9384830

Campbell, S. F. (2007). *Steering control of an autonomous ground vehicle with application to the DARPA urban challenge* (Doctoral dissertation, Massachusetts Institute of Technology).

Chi, H., Tang, C., Hou, D., Wu, W., & Huang, X. (2024). Spatiotemporal-restricted A∗ algorithm as a support for lane-free traffic at intersections with mixed flows. *Green Energy & Intelligent Transportation*, **3**, 100159. https://doi.org/10.1016/j.geits.2024.100159

Cui, Y., & Ge, S. S. (2003). *Autonomous vehicle positioning with GPS in urban canyon environments. IEEE Transactions on Robotics and Automation*, **19**, 15–25. https://doi.org/10.1109/tra.2002.807557

Dominguez, S., Ali, A., Garcia, G., & Martinet, P. (2016). *Comparison of lateral controllers for autonomous vehicle: Experimental results. In 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*(pp. 1418–1423). IEEE. https://doi.org/10.1109/itsc.2016.7795743

Guo, N., Lenzo, B., Zhang, X., Zou, Y., Zhai, R., & Zhang, T. (2020). A real-time nonlinear model predictive controller for yaw motion optimisation of distributed drive electric vehicles. *IEEE Transactions on Vehicular Technology*, **69**, 4935–4946. https://doi.org/10.1109/TVT.2020.2980169

Guo, N., Liu, J., Li, J., Chen, W., Zhang, Y., Lu, Q., & Chen, Z. (2025). Handling-stability control for distributed drive electric vehicles via Lyapunov-based nonlinear MPC algorithm. *IEEE Transactions on Transportation Electrification*, **11**, 6615–6628. https://doi.org/10.1109/TTE.2024.3513438

Hossain, T., Habibullah, H., & Islam, R. (2022). *Steering and speed control system design for autonomous vehicles by developing an optimal hybrid controller to track reference trajectory*. *Machines*, **10**, 420. https://doi.org/10.3390/machines10060420

Hwang, I., & Lee, Y.-G. (2024). The identification of minor impact collisions in a long video for detecting property damages caused by fleeing vehicles using three-dimensional convolutional neural network. *Journal of Computational Design and Engineering,* **11**, 106–121. https://doi.org/10.1093/jcde/qwae016

Johnson, M. A., & Moradi, M. H. (2005). *PID control*(pp. 47–107). Springer-Verlag London Limited. https://doi.org/10.1007/1-84628-148-2

Kaplan, E. D., & Hegarty, C. (Eds.). (2017). *Understanding GPS/GNSS: Principles and applications*. Artech house. https://doi.org/10.1017/aer.2019.98

Karaim, M., Elsheikh, M., & Noureldin, A. (2018). *GNSS error sources. In Multifunctional operation and application of GPS*. IntechOpen. https://doi.org/10.5772/intechopen.75493

Kiran, B. R., Sobh, I., Talpaert, V., Mannion, P., Sallab, A. A. A., Yogamani, S., & Pérez, P. (2022). Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, **23**, 4909–4926. https://doi.org/10.1109/TITS.2021.3054625

Kouvaritakis, B., & Cannon, M. (2016). *Model predictive control*. Springer: Springer International Publishing. https://doi.org/10.1007/978-3-319-24853-0

Landau, H., Vollath, U., & Chen, X. (2009). *Virtual reference station systems*. *Positioning*, **1**, 0–0. https://doi.org/10.5081/jgps.1.2.137

Langley, R. B. (1998). *RTK GPS*. *GPS World*, **9**, 70–76. https://gge.ext.unb.ca/Resources/gpsworld.september98.pdf. ( 15 October 2024, date last accessed).

Langley, R. B. (1998). *The UTM grid system*. *GPS World*, **9**, 46–50. https://doi.org/10.1007/978-3-642-72011-6_4

Langley, R. B., Teunissen, P. J., & Montenbruck, O. (2017). *Introduction to GNSS*. Springer Handbook of Global Navigation Satellite Systems, 3–23. https://doi.org/10.1007/978-3-319-42928-1_1

Lee, S., & Lee, Y.-G. (2023). Split liability assessment in car accident using 3D convolutional neural network. *Journal of Computational Design and Engineering*, **10**, 1579–1601. https://doi.org/10.1093/jcde/qwad063

Liu, C., Lee, S., Varnhagen, S., & Tseng, H. E. (2017). *Path planning for autonomous vehicles using model predictive control. In 2017 IEEE Intelligent Vehicles Symposium (IV)*(pp. 174–179). IEEE. https://doi.org/10.1109/IVS.2017.7995716

Ma, Y., Zhang, Z., Zhang, L., & Liu, C. (2019). MPC-based slip ratio control for electric vehicle considering road roughness. *IEEE Access*, **7**, 52405–52413. https://doi.org/10.1109/ACCESS.2019.2910891

Mekik, C., & Arslanoglu, M. (2009). *Investigation on accuracies of real time kinematic GPS for GIS applications*. *Remote Sensing*, **1**, 22–35. https://doi.org/10.3390/rs1010022

Ministry of Land, Infrastructure and Transport [Republic of Korea]. (2024). Regulations on Structure and Facilities of Roads [도로의 구조·시설 기준에 관한 규칙, MOLIT Ordinance No. 1360 ] . *Article*, **10**, 3. https://www.law.go.kr/법령/도로의구조·시설기준에관한규칙>. ( 5 September 2024, date last accessed).

Noh, S., & An, K. (2022). Reliable, robust, and comprehensive risk assessment framework for urban autonomous driving. *Journal of Computational Design and Engineering*, **9**, 1680–1698. https://doi.org/10.1093/jcde/qwac079

NovAtel. (n.d.). *Chapter 4: GNSS error sources*. NovAtel. https://novatel.com/an-introduction-to-gnss/gnss-error-sources(Accessed on September 5th, 2024)

Park, M. W., Lee, S. W., & Han, W. Y. (2014). *Development of lateral control system for autonomous vehicle based on adaptive pure pursuit algorithm. In 2014 14th International Conference on Control, Automation and Systems (ICCAS)*(pp. 1443–1447). IEEE. https://doi.org/10.1109/ICCAS.2014.6987787

Ploeg, J., Shukla, D. P., van de Wouw, N., & Nijmeijer, H. (2014). Controller synthesis for string stability of vehicle platoons. *IEEE Transactions on Intelligent Transportation Systems*, **15**, 854–865. https://doi.org/10.1109/TITS.2013.2291493

Raffo, G. V., Gomes, G. K., Normey-Rico, J. E., Kelber, C. R., & Becker, L. B. (2009). *A predictive controller for autonomous vehicle path tracking*. *IEEE Transactions on Intelligent Transportation Systems*, **10**, 92–102. https://doi.org/10.1109/tits.2008.2011697

Rahiman, W., & Zainal, Z. (2013). *An overview of development GPS navigation for autonomous car. In 2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA)*(pp. 1112–1118). IEEE. https://doi.org/10.1109/iciea.2013.6566533

Rezaei, S., & Sengupta, R. (2007). *Kalman filter-based integration of DGPS and vehicle sensors for localisation*. *IEEE Transactions on Control Systems Technology*, **15**, 1080–1088. https://doi.org/10.1109/tcst.2006.886439

Sallab, A. E., Abdou, M., Perot, E., & Yogamani, S. (2017). Deep reinforcement learning framework for autonomous driving. *Electronic Imaging*, **2017**, 70–76. https://doi.org/10.2352/ISSN.2470-1173.2017.19.AVM-023

Samuel, M., Hussein, M., & Mohamad, M. B. (2016). *A review of some pure-pursuit based path tracking techniques for control of autonomous vehicle. International Journal of Computer Applications*, **135**, 35–38. https://doi.org/10.5120/ijca2016908314

Schwenzer, M., Ay, M., Bergs, T., & Abel, D. (2021). *Review on model predictive control: An engineering perspective. The International Journal of Advanced Manufacturing Technology*, **117**, 1327–1349. https://doi.org/10.1007/s00170-021-07682-3

Snider, J. M. (2009). *Automatic steering methods for autonomous automobile path tracking*. Robotics Institute, Tech. Rep. CMU-RI-TR-09-08.

Teunissen, P. J. G., & Khodabandeh, A. (2015). *Review and principles of PPP-RTK methods. Journal of Geodesy*, **89**, 217–240. https://doi.org/10.1007/s00190-014-0771-3

Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., & Mahoney, P. (2006). *Stanley: The robot that won the DARPA Grand Challenge. Journal of Field Robotics*, **23**, 661–692. https://doi.org/10.1002/rob.20147

Van Brummelen, J., O'Brien, M., Gruyer, D., & Najjaran, H. (2018). *Autonomous vehicle perception: The technology of today and tomorrow. Transportation Research Part C: Emerging Technologies*, **89**, 384–406. https://doi.org/10.1016/j.trc.2018.02.012

Vanholme, B., Gruyer, D., Lusetti, B., Glaser, S., & Mammar, S. (2012). *Highly automated driving on highways based on legal safety. IEEE Transactions on Intelligent Transportation Systems*, **14**, 333–347. https://doi.org/10.1109/tits.2012.2225104

Vivek, K., Sheta, M. A., & Gumtapure, V. (2019). *A comparative study of Stanley, LQR and MPC controllers for path tracking application (ADAS/AD). In 2019 IEEE International Conference on Intelligent Systems and Green Technology (ICISGT)*(pp. 667–674). IEEE. https://doi.org/10.1109/icisgt44072.2019.00030

Wang, L., Zhai, Z., Zhu, Z., & Mao, E. (2022). *Path tracking control of an autonomous tractor using improved Stanley controller optimized with multiple-population genetic algorithm. Actuators*, **11**, 22. https://doi.org/10.3390/act11010022

Wanninger, L. (2004). *Introduction to network RTK. IAG Working Group*, **4**, 2003–2007.

Zein, Y., Darwiche, M., & Mokhiamar, O. (2018). *GPS tracking system for autonomous vehicles. Alexandria Engineering Journal*, **57**, 3127–3137. https://doi.org/10.1016/j.aej.2017.12.002

Zheng, Y., Li, S. E., Wang, J., Cao, D., & Li, K. (2016). Stability and scalability of homogeneous vehicular platoon: Study on the influence of information flow topologies. *IEEE Transactions on Intelligent Transportation Systems*, **17**, 14–26. https://doi.org/10.1109/TITS.2015.2402153(?PMU?)

# Appendix A. Positioning

Positioning refers to the process by which a moving object estimates its own position, speed, and orientation. When using GNSS equipment, the object's current position, represented by latitude and longitude, is determined by measuring the time it takes for signals transmitted from reference points—such as GNSS satellites and base stations—to reach the observation point, which in this case is the vehicle. By acquiring accurate position information, the vehicle can plan its driving path, calculate its deviation from the planned trajectory, and perform lateral control to minimize this deviation.

## A.1. GNSS

GNSS is a positioning technology that calculates the position and speed of a moving object using signals transmitted from satellites located in medium earth orbit at an altitude of 20 220 km above the Earth's surface (Kaplan & Hegarty, 2017). The basic positioning process of GNSS is as follows:

(1) Satellite signal reception

The GNSS receiver mounted on the vehicle receives signals transmitted from multiple satellites. The signals received contain information about the satellite's location and the time the signal was transmitted.

(2) Distance measurement

The GPS receiver calculates the distance to each satellite based on the time the signal was transmitted.

(3) Trilateration

The vehicle's position (latitude and longitude) is calculated using distance measurements from at least three satellites. To determine the position accurately and uniquely through trilateration, a minimum of three reference points is required.

However, the position information provided by GNSS can cause several meters errors in the absence of corrections. The main sources of these errors (Langley *et al.*, 2017) are summarized in Table A1.

(1) Satellite clock errors

Although the atomic clocks onboard the satellites provide highly accurate time information, even a small-time error can lead to significant positioning errors. The GPS receiver calculates the time difference between the transmitted and received signals and assumes the transfer speed of the signal to be the speed of light to compute the distance from the satellite. For instance, a 10ns error in satellite time would result in a positioning error of 3 m.

(2) Satellite orbit errors

The discrepancy between the predicted and actual orbits of GNSS satellites can lead to distance errors, resulting in positioning errors of up to 2.5 m.

(3) Ionospheric delay, tropospheric delay

The satellite signals travel through the ionosphere and troposphere before reaching the receiver on the Earth's surface, causing transfer time delays that result in positioning errors of up to 5 m.

(4) Receiver noise

It is caused by signal noise generated in the GNSS receiver, resulting in positioning errors of up to 0.3 m.

(5) Multipath error

It is the error caused when signals reflected or diffracted by buildings or other structures near the receiver arrive at the receiver along with the direct signal. The reflected or diffracted signals reach the receiver with a delay, resulting in positioning errors of up to 1 m.

In South Korea, the minimum lane width is 3.0 m for urban areas with a design speed of less than 60 km/h (Ministry of Land, Infrastructure and Transport [Republic of Korea] 2024). If GNSS po-

**Table A1:** Positioning errors of GNSS (NovAtel,n.d.).

| Source of errors | Error range |
|---|---|
| Satellite clock errors | ±2 m |
| Satellite orbit errors | ±2.5 m |
| Ionospheric delay | ±5 m |
| Tropospheric delay | ±0.5 m |
| Receiver noise | ±0.3 m |
| Multipath error | ±1 m |

sitioning data is used directly without correction, it implies that the vehicle would be driving with an error equivalent to one or more lanes, which makes it unsuitable for practical driving conditions. We acknowledge GNSS failures such as multipath and urban canyon effects. In practice, fallback strategies include conservative gains, smoothing filters, and safe disengagement thresholds. Because we wanted to make our paper focused on our merits, we have not discussed fallback strategies.

## A.2. RTK

RTK is a correction method applied in this study to minimize the positioning errors of GNSS, providing positioning results with an error range of a few centimeters (Langley 1998). RTK uses correction data from the carrier phase of the GNSS base station (with precise pre-existing position information) to correct the positioning of the GNSS rover in real time, ensuring an error range within a few centimeters. The process by which RTK corrects the positioning results using carrier phase correction values is as follows.

(1) A GNSS receiver is installed at a location with known latitude, longitude, and altitude (GNSS base), which serves as the reference point.
(2) The GNSS base calculates the errors by using the positioning data received from the satellites.
(3) The GNSS base transmits the error information to nearby GNSS receivers (GNSS rover) that are capable of receiving it.
(4) The GNSS rover receives error information via the network and uses it to correct its positioning results.

RTK has a limitation in that the correction accuracy decreases as the rover moves farther away from the base. This is because, as the distance from the base increases, the reception environment of the rover's signals changes due to factors like ionosphere and troposphere effects. To address this, network RTK(Wanninger 2004), typically known as a Virtual Reference Station, must be applied. In South Korea, the VRS service provided by the National Geographic Information Institute can be utilized. To apply Network RTK, NGII recommends using the NTRIP client, which connects to the VRS server of NGII to acquire correction data and transmit it to the rover (GNSS mounted on the vehicle).

The proposed system performs network RTK by installing the NTRIP client on a network-enabled PC, which communicates with the NGII VRS server. The correction data received from the server is then transmitted to the GNSS receiver via USB. Figure A1 presents the overall flow of RTK correction applied in the proposed system.

To apply RTK, the GNSS must be configured to transmit information such as current position, altitude, and time in the National Marine Electronics Association (NMEA) message format, specifically the GGA format, to the NTRIP client. The NTRIP client then communicates with the VRS server to acquire Radio Technical Commission for Maritime Services version 3 correction data. For RTK correction, a minimum of six satellites must be connected to receive signals, which allows the error to be reduced to within a few centimeters (Mekik & Arslanoglu, 2009).

## A.3 GNSS heading

GNSS heading refers to the estimation of the vehicle's attitude, particularly its direction (vehicle heading), based on positioning data. When additional antennas are installed, more accurate data can be obtained. Each antenna receives signals from satellites and acquires position and altitude information, which is then used to calculate the vehicle heading(yaw) and vehicle pitch. In the GNSS used in this system, when two or more antennas are installed with a baseline distance of at least 1 meter, the system exhibits an error of approximately $\pm 0.2^\circ$. In this study, the vehicle heading information obtained from the GNSS system is used to determine the vehicle's orientation, with the vehicle heading defined as $0^\circ$ at true north and increasing clockwise.

GNSS heading is obtained by calculating the relative position of the secondary antenna from the primary antenna, which serves as the reference point, as illustrated in Figure A2. Therefore, it is recommended to install the primary antenna behind the secondary antenna to improve the accuracy of the heading estimation. Additionally, the more accurate the position information obtained from each antenna, the more precise the attitude information will be. For optimal results, it is appropriate to use GNSS heading in a setup where RTK is applied, as this enhances the accuracy of the positioning data.

## A.4 IMU

Even when error correction is applied to GNSS, in places with intermittent signal interference, such as urban canyons, GNSS-based positioning can still be inaccurate (Cui & Ge, 2003). To handle this situation, an IMU is used. The IMU consists of an accelerometer and a gyroscope, which measure the vehicle's attitude and velocity at regular intervals. The IMU then integrates the measured attitude and velocity over time to determine the vehicle's final position and attitude (Ahmad *et al.*, 2013).

The IMU's positioning method provides relative position and attitude information from an initial state. So, it is essential to know the initial position and attitude. Therefore, the position information obtained from the GNSS's positioning result was set as the initial position.

In summary, the GNSS positioning results were primarily used in the positioning phase, and further refinement was achieved by using the IMU's positioning results to obtain more precise positioning data.

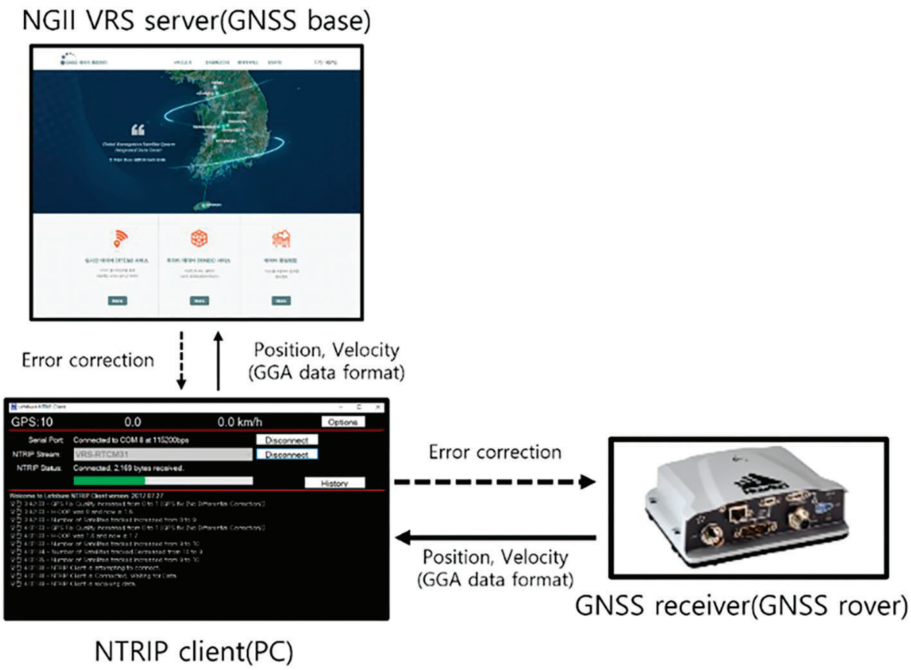


**Figure A1:** RTK application system flow diagram.

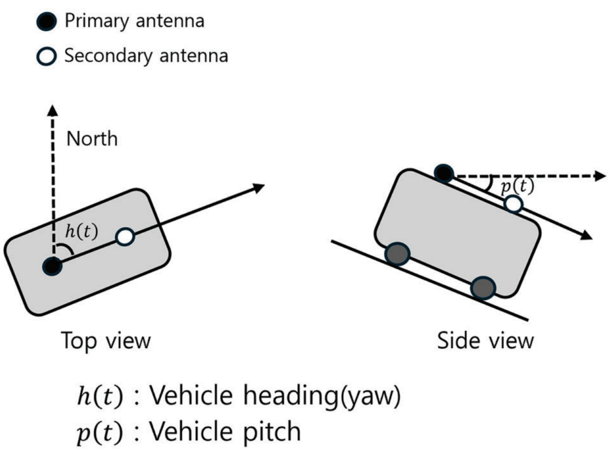


**Figure A2:** Vehicle attitude acquired via GNSS positioning.



**Received:** June 12, 2025. **Revised:** October 10, 2025. **Accepted:** October 11, 2025