

Real-Time Path Planning for Trajectory Control in Autonomous Driving

Hye Young An, Won Seok Choi, Seong Gon Choi

Information & Communication Engineering, Chungbuk National University, Cheongju-si Chungcheongbuk-do, Korea
 youngflow@chungbuk.ac.kr, wschoi@chungbuk.ac.kr, choisg@chungbuk.ac.kr

Abstract— We propose an algorithm that enables real-time path planning and trajectory control of autonomous vehicles utilizing the detected lane, minimum turning radius, and driving radius. Path planning and trajectory control for autonomous driving are the key technologies in dynamic environments. For trajectory control, steering angle calculation is essential. This calculation mainly uses the next position information from GPS and predefined paths. For path planning and trajectory control, autonomous vehicles use sensors or communication systems to obtain information about their surroundings. However, it may be difficult to obtain the required path information for trajectory control because of environmental factors and surrounding conditions. To alleviate these uncertainties, we propose an algorithm that utilizes the next position generated in real time based on the detected lane information obtained from various sensors. The next position is generated by the intersection and midpoint based on the detected lane, minimum turning radius, and driving radius considering speed. Then, the steering angle is calculated based on the generated next position using a pure tracking algorithm. We simulated four different speeds at a roundabout. The obtained simulation results show that our algorithms work well in real-time driving at various speeds.

Keywords— Path trajectory, Roundabout, Pure pursuit algorithm, Bicycle model, Path tracking, Minimum turning circle

I. INTRODUCTION

Nowadays, advances in artificial intelligence, big data and information processing technologies have made autonomous vehicles (AVs) more attention. AVs have a huge potential for boosting road efficiency via safe increases in traffic density, overcoming human error, minimizing pollution, and energy waste. In particular, AVs technology has been advanced in various fields to improve driving safety, driving comfort, economy, and reduce traffic accidents [1].

In terms of software systems, autonomous driving is studied by classifying them into five categories, namely localization, perception, planning, control, and system management [2]. The localization identifies the location of the vehicle and the perception identifies driving environment elements such as other road users, traffic signals, and obstacles. The planning determines the travel paths and driving actions. The control transforms the behavior and actions indicated by the planning system to the vehicle manipulation commands (e.g., steering, accelerating, and

braking). The system management oversees the operation of all the systems and provides the human-machine interface.

We are focused on keeping the vehicle moving along a correct path. In order to achieve this task, the creation of an optimal route (planning) and good reference trajectory generation (control) are requirements. Effective path planning helps AVs complete tasks in the shortest time, saving energy and improving efficiency as well. Guo Qing et al. and A. Chambers et al. used the proposed algorithm to find the optimal path or plan a driving path based on energy efficiency [3], [4].

However, accurate and robust trajectory control is essential to maintain the desired performance (e.g., driving at the desired speed, maintaining passenger comfort) based on a path explored [8]. The trajectory tracking algorithm to control the vehicle has three classic techniques of lateral control based on the bicycle model: pure pursuit, Stanley, and simplified kinematic steering control. The trajectory tracking algorithms are calculated by choosing a point from path trajectory points generated in planning [7], [8]. Most studies using these algorithms generate trajectories along predefined paths and global positioning system (GPS) routes. The GPS is the most used localization system for vehicle applications, as they offer a cheap and easily accessible solution for global positioning. On the other hand, GPS is unreliable because it provides insufficient accuracy (~10m) for autonomous driving due to several factors (blocked by tall buildings, tunnels, mountain roads) [9]. In addition, in a real environment, it is difficult to use the predefined path, or the path can be changed for various reasons.

Therefore, we present a real-time path planning algorithm limited to close range based on detected lanes. For lane detection, various sensors (LiDAR, Radar, Camera, IMU and Odometry) are utilized. Various studies are shown that lane detection using sensors has a good accuracy level [10] ~ [12].

In our proposed method, it is possible to determine in real-time the next position for trajectory calculation using the detected lane by the sensors. In addition, the trajectory of the AVs is generated through the calculated steering angle based on the next position by applying a pure pursuit algorithm.

The remainder of this paper is structured as follows. Related work is discussed in Section II. The trajectory control using real time path planning for AVs is described in section III. In section IV, simulation results and discussions are

presented. Conclusion and a view on the future works are drawn in Section V.

II. RELATED WORK

Planning and control are the hotspots in the field of AVs in Intelligent Transportation Systems. Planning and control have been studied separately or integrated together.

Planning is responsible for path generation from an initial to a final point, focusing on saving energy and improving efficiency [2]. As is known, the two most famous competitions to promote autonomous vehicle techniques are the 2005 DARPA Grand Challenge and the 2007 Urban Challenge, after which the path planning of autonomous driving has been studied more actively [1]. Based on the improved Dijkstra algorithm, Guo Qing et al. show that it is possible to solve the path planning problem with multiple shortest paths and find the optimal path using both the shortest distance and travel time [3]. A. Schambers et al. present the use of an embedded graphical processing unit and a parallel implementation of Bellman-Ford on an embedded GPU system to plan driving routes based on energy efficiency [4].

Path planning is important but cannot be viewed in isolation from control due to the constraints of vehicle dynamics and the presence of surrounding vehicles in a dynamic environment. In particular, planning should include controlling the vehicle to maintain the desired performance in terms of precision, stability and passenger comfort. In other words, planning and control are separated into different layers, but they are intrinsically related.

Control for AV uses the information of vehicle dynamics, kinematics, tire dynamics, deviation of position, path, and heading and road curvature. Q. Yao et al. reviewed the current technologies on path tracking control strategies for AV [1]. Path tracking control methods include MPC, Pure pursuit & Stanley methods, etc. C. Liu et al. proposed a unified path planning approach using Model Predictive Control (MPC), which automatically decides the mode of maneuvers. However, the stability analysis of the path tracking system based on MPC is still a challenge [5]. Á. Domina et al. tested the geometrical trajectory following the controller through the implementation of the lateral dynamic bicycle model and the implementation of various controllers [7]. This trajectory is created with two curves, based on a tangent hyperbolic function. As a result, the geometrical trajectory following the controller is suitable for low-speed range tracking, and the simulation result is close to the real environment measurement result. S. Dominguez et al. evaluate three classical techniques of the lateral control (pure pursuit, Stanley and a simplified kinematic steering control) with a real car and a simulator. As a result, it is shown that it is sufficient to choose the pure pursuit in terms of passenger comfort and safety, and it is accurate enough for autonomous driving [8].

Most studies on control have focused only on the trajectory control using reference path and GPS [7], [8]. However, in real environments, it may be difficult to obtain the GPS route information and as a result, unexpected problems may occur in controlling the vehicle using the path.

Therefore, to overcome this problem, various sensors are utilized for autonomous driving. D. C. Andrade et al. presents a new strategy for lane detection and tracking using a monocular digital camera in a vehicle's windshield. The study shows good accuracy levels, including scenarios with shadows, curves, and road slopes [10]. A. Gupta et al. propose a novel, integrated, real-time framework for lane detection, tracking and road surface marking detection and recognition. Accurate and automated lane and road surface marking detection and recognition from cameras play an important role in intelligent vehicles. It helps increase driving safety in advanced driver assistant systems and enhance path planning and global localization in autonomous driving vehicles. The proposed method achieves a detection rate of over 96% in all challenging scenarios [11]. R. P. D. Vivacqua et al. present a novel low-cost high-accuracy localization method for outdoor vehicles capable of performing autonomous driving on narrow two-way roads using cameras or LIDAR [12]. In visual control systems, both circle and line detection are necessary for object detection and high-accuracy localization. C. Lu et al. shows a novel method for circle detection by analysing and refining arc-support line segments [13].

In most studies, it can be seen that object detection method utilizing sensors offers accurate image analysis in AV. Therefore, planning and control are performed on the premise that lane detection is possible based on sensors. It virtually draws the driving radius considering speed and minimum turning radius on the detected lane based on the sensor. Based on this, a real time path is generated through our proposed algorithm. In addition, the pure pursuit algorithm is applied to this path to control.

III. PROPOSED METHOD

The Paths of AVs must provide continuous position, heading, and curvature at all places along the proposed path in order to be feasible. The main aim of this study is to generate a real-time path for trajectory control so that the vehicle tracks the correct path. Our idea consists of two algorithms: one that determines the next position based on a detected lane with a sensor, and another that generates a trajectory.

Algorithm 1 shows a fragment of pseudocode used for generating the next positions and tracking trajectory. The next position generation algorithm proceeds with lane detection, drawing virtual lines, finding the intersection of lanes and minimum turning radius, calculating the vehicle's turning range, calculating the next point, and calculating of the rear axle location. Then, the steering angle is calculated through the trajectory tracking algorithm. Use this steering angle to move to the next position generated above. A detailed description of the algorithm is explained below.

A. The next position generation algorithm

The next point is selected by the intersection and midpoint of several line segments connecting the minimum turning radius, driving radius, and lane. The next position determination algorithm is graphically illustrated in Figure 1. The meanings of points and circles are explained in Table 1.

Algorithm 1: Path generation and tracking

input: sensor data (detected lane, current yaw angle, current position, destination position, speed)
output: steering angle and yaw angle (For next position)

```

1  while current position != destination position do
2       $R_{c\_dr} \leftarrow$  get driving radius considering speed
3       $R_{L\_min}, R_{R\_min}, R_{c\_dr} \leftarrow$  draw virtual circle
4       $P_{LI\_min\_lane}, P_{RI\_min\_lane} \leftarrow$  get intersection of lane and
        minimum turning radius
5       $P_{LI\_min\_dr}, P_{RI\_min\_dr} \leftarrow$  get limit of driving radius
6       $P_{C\_min\_lane} \leftarrow$  get midpoint ( $P_{LI\_min\_dr}$  and  $P_{RI\_min\_dr}$ )
7      heading_line  $\leftarrow$  make line ( $P_{C\_min\_lane}$  and  $P_{C\_rear}$ )
8       $P_{N\_head} \leftarrow$  get intersection ( $R_{c\_dr}$  and heading_line)
9      if  $P_{N\_head} > P_{LI\_min\_dr}$  then  $P_{N\_head} = P_{LI\_min\_dr}$ 
10     else if  $P_{N\_head} < P_{RI\_min\_dr}$  then  $P_{N\_head} = P_{RI\_min\_dr}$ 
11     end
12      $\ell_d \leftarrow$  get distance ( $P_{N\_head}, P_{C\_head}$ )
13      $\alpha \leftarrow$  get angle ( $P_{N\_head}$ )
14      $\delta \leftarrow$  get steering angle using pure pursuit ( $P_{N\_head}, L, \alpha, \ell_d$ )
15      $\psi \leftarrow$  get yaw angle of heading ( $P_{N\_head}, \delta$ )
16     move vehicle ( $\psi, \delta$ )
17      $P_{C\_rear} \leftarrow$  calculate rear axle position ( $\psi, P_{N\_head}, P_{C\_head}$ )
18     current position =  $\psi$ 
19 end

```

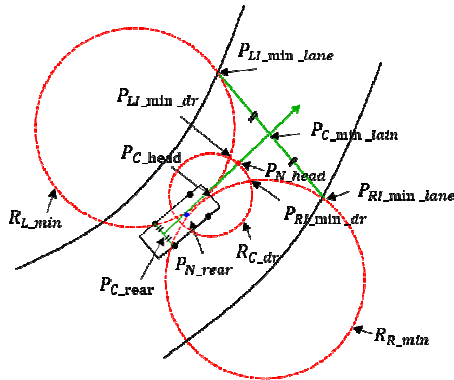


Figure 1. Calculating next position

Based on figure 1, a feasible next point is selected based on the sequence below:

1) Lane detection: The algorithm is performed on the premise that the sensor can detect straight lanes and road curves well [10] ~ [12].

2) Virtual Line drawing: A vehicle's minimum turning radius is the minimum radius measured by the outer wheels of the vehicle while making a complete turn. Generally, the minimum turning radius is drawn as the outer wheel in the direction of rotation [15]. As shown in Figure 1, we draw by scaling the size of the virtual minimum turning radius minus

TABLE 1. EXPLAIN OF POINTS AND CIRCLES

Item	Explain
R_{c_dr}	Driving radius according to the speed
R_{L_min}	Left minimum turning radius, a circle the size of the minimum turning radius minus 1/2 the width of the vehicle.
R_{R_min}	Right minimum turning radius, a circle the size of the minimum turning radius minus 1/2 the width of the vehicle.
$P_{LI_min_lane}$	Intersection of the left minimum turning radius (R_{L_min}) and the left lane of the road
$P_{RI_min_lane}$	Intersection of the right minimum turning radius (R_{R_min}) and the right lane of the road
$P_{C_min_lane}$	Midpoint of two intersections ($P_{LI_min_lane}$ and $P_{RI_min_lane}$) of minimum turning radius and lane
$P_{RI_min_dr}$	Intersection of right minimum turning radius (R_{R_min}) and driving radius (R_{c_dr})
$P_{LI_min_dr}$	Intersection of left minimum turning radius (R_{L_min}) and driving radius (R_{c_dr})
P_{N_head}	Intersection of line segments $P_{C_rear}, P_{C_min_lane}$ and driving radius (R_{c_dr}). The next heading center position, we have to change the steering angle of the vehicle in this direction.
P_{C_head}	Midpoint of the vehicle heading
P_{N_rear}	Next move position relative to the rear axle.
P_{C_rear}	Midpoint of the rear axle

1/2 of the vehicle's width, because the algorithm is computed based on the center of the vehicle's heading. Then, the driving radius according to the speed is drawn. This part corresponds to line 2 ~ 3 of Algorithm 1.

3) Intersections of lanes and minimum turning radius: Taking account of the road's consisting of curved and straight parts, an intersection point detection algorithm is used [14]. There are two equations used. The first is the intersection equation of a line and a circle. The second uses the formulas for finding the intersection points between two circles. Of the two detected intersection points, the coordinate in front of the vehicle heading is selected. The selected two intersection points are $P_{LI_min_lane}$ and $P_{RI_min_lane}$. However, several exceptions are existed to select intersection point with the lane. These various scenarios are explained in more detail in 4).

4) Various scenarios of intersection point selection between lanes and virtual circles of the vehicle: There are three scenarios for selecting the intersection of the lane and virtual circle (minimum turning radius or driving radius). In case Figure 2 (a), it is determined by the intersection of the detected lane and the minimum turning radius. (b) is a case where the intersection of the driving radius and the lane is ahead of the intersection of the minimum turning radius and

the lane. Because it must be moved by a constant distance according to the speed, the intersection of driving radius is selected. (c) is a special case. A virtual lane parallel to the right lane is created on the left side of the vehicle when the vehicle has an intersection between the right minimum turn radius and the curved road of the exit lane. A left lane intersection point is selected as the intersection of the virtual drawn lane and the left minimum turning radius. This part corresponds to lines 4 of Algorithm 1.

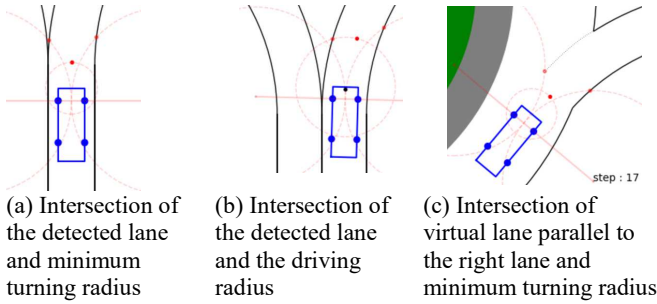


Figure 2. Various situations of intersection point between lanes and virtual circles of vehicles

5) Calculation of the vehicle turning range: The vehicle turning range is calculated based on the left and right minimum turning radius and driving radius. As shown in Figure 1, the vehicle turning range is an arc of the vehicle driving radius (R_{c_dr}) between $P_{LI_min_dr}$ and $P_{RI_min_dr}$. There are exceptions such as (b) and (c) in Figure 3. The driving radius of vehicle must be within its lane. Therefore, the intersection of the lane and the driving radius is selected, not the intersection of the minimum turning radius and the driving radius. This part corresponds to line 5 of Algorithm 1.

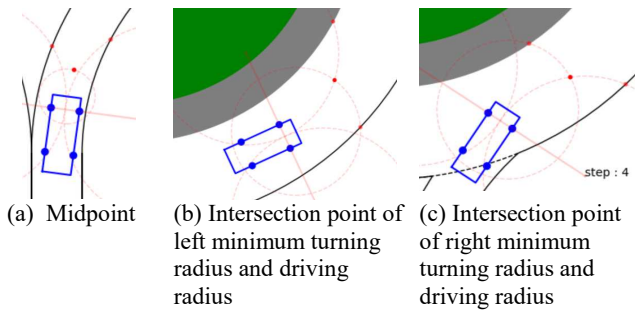


Figure 3. The next driving position selection

6) Calculation of the next point: The midpoint of lines connecting two intersection points $P_{LI_min_lane}$ and $P_{RI_min_lane}$ is $P_{C_min_lane}$. This part corresponds to line 6 of Algorithm 1. The intersection point (P_{N_head}) of line segments (P_{C_rear} and $P_{C_min_lane}$) and driving radius (R_{c_dr}) is selected as the next position as shown in Figure 3. As a result, P_{N_head} is the next driving position to move of the vehicle. The steering angle of the vehicle have to be changed in the direction of this next position. There are three selections depending on the

intersection position with the lane ($P_{LI_min_lane}$ and $P_{RI_min_lane}$). In Figure 3, In case (a), the next position is the midpoint between $P_{LI_min_dr}$ and $P_{RI_min_dr}$. In (b), if the midpoint ($P_{C_min_lane}$) of the lane is biased to the left due to the curvature of the lane, the intersection point with the driving radius may also be biased to the outside the range of the left minimum turning radius. In this case, since the next point is outside the range of the left minimum turning radius, the next position of the vehicle is selected as the intersection point ($P_{LI_min_dr}$) of the left minimum turning radius and driving radius. (c) is the opposite of (b). This part corresponds to line 7 ~ 11 of Algorithm 1.

7) Calculation of the rear axle location: When the vehicles move in the direction of P_{N_head} position, it is necessary to calculate the position of the rear axle. Because trajectory algorithm is calculated based on real axle. rear axle location (P_{N_head}) is calculated considering driving radius and yaw angle with respect to the global coordinates. This part corresponds to line 17 of Algorithm 1.

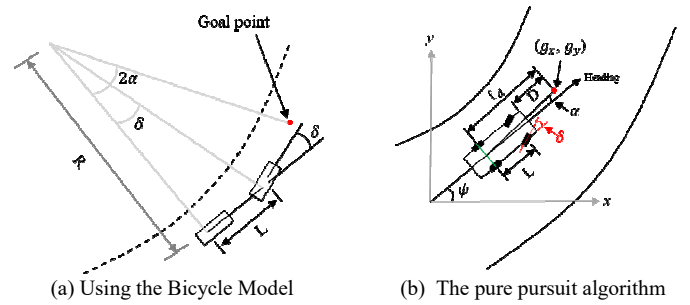


Figure 4. The trajectory algorithm

B. Trajectory tracking algorithm

The pure-pursuit algorithm used in our project is one of the most common and the most powerful trajectory tracking algorithm in vehicle control [8]. As shown in Figure 4 (a), this algorithm can be better visualized by using the Bicycle Model to represent the vehicle [6]. This simplification creates a simple geometric relationship between the front wheel steering angle and the rear axle. This algorithm approach geometrically determines the steering angle that will drive the vehicle from its current position to the next position on the path [6]. It uses vehicle position and odometry to control vehicle speed and steering to follow a reference path. The path tracking based on the pure pursuit algorithm can be demonstrated in Figure 4 (b). A goal point is written as (g_x, g_y) . The goal point is determined by a look-ahead distance (L) from the current rear axle position to the desired path. δ is the steering angle of the front wheel. The vehicle's steering angle (δ) can be determined using only the goal point location and the angle (α) between the vehicle's heading vector and the look-ahead vector. L is the distance between the front axle and rear axle (wheelbase). R is the radius of the circle that the rear axle will travel along at the given steering angle. Implementing path tracking control requires the vehicle's location in global

coordinates. ψ represents the yaw angle with respect to the global coordinates. This part corresponds to line 12 ~ 15 of Algorithm 1.

The Pure Pursuit control law can be expressed as.

$$\delta = \tan^{-1}\left(\frac{2L \sin(\alpha)}{\ell_d}\right) \quad (1)$$

By applying the law of sine and cosines, calculate the rear axle location.

$$(x, y) = (D \cos(\psi), D \sin(\psi)) \quad (2)$$

IV. SIMULATION

We simulated vehicle movement in a roundabout environment including both circular arcs and straight lines using python, matplotlib to validate the proposed algorithm.

A. Environment

1) Roundabout configuration: Our approach was tested considering the construction of the vehicle paths at a single-lane roundabout according to the design guidelines of the Ministry of Land, Infrastructure, and Transport (South Korea) [16]. The roundabout environment is defined in meters. The approach and departure width at the roundabout are 3 meters. The entry or exit radius are about 15 meters. The diameter of a center island is given as 25 meters. The circulatory roadway width is 5.5 m. The truck apron is 2.5 meters.

2) Specifications of the standard design vehicle: The basic proportions and overall dimensions of the vehicle is governed by the specifications of the standard design vehicle (SDV) and are shown in Figure 5. The minimum turning radius calculated from specifications of SDV is 6 m [16].

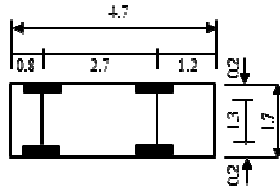


Figure 5. Specifications of the standard design vehicle(SDV)

B. Test result

In urban environments, the recommended design speed of the single entry lane and exit lane is 40-50 km/h, and the recommended design speed of the circulatory lane is 20-30 km/h [16]. Therefore, in a simulation environment, it was tested from 20km/h to 50km/h. Calculating the driving radius requires time and speed. Other time intervals can be used, but this simulation calculates the driving radius every 300ms. Figure 6. shows simulation result at a speed 30km/h. The movement of the vehicle is drawn every 3 steps. The obtained simulation works well at a speed of 30km/h. Figure 7. shows trajectory according to various speed. As a result, the vehicle follows a safe trajectory at various speeds. Figure 8. shows error from the center of the lane for each speed. The maximum error when entering the circulatory roadway is

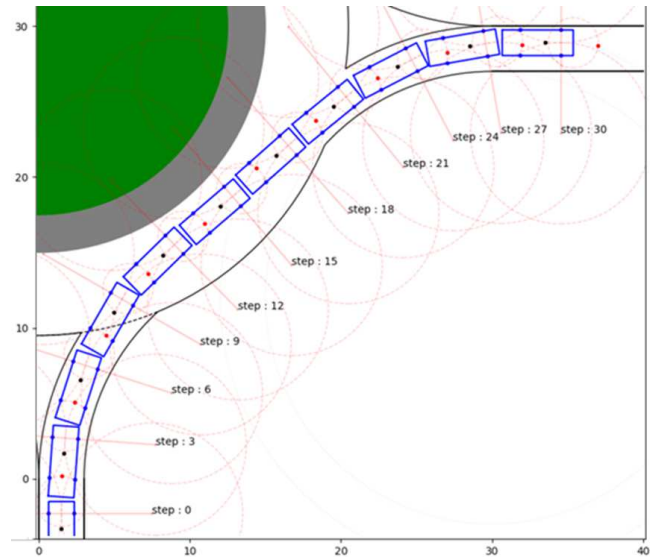


Figure 6. Simulation results at a speed of 30km/h

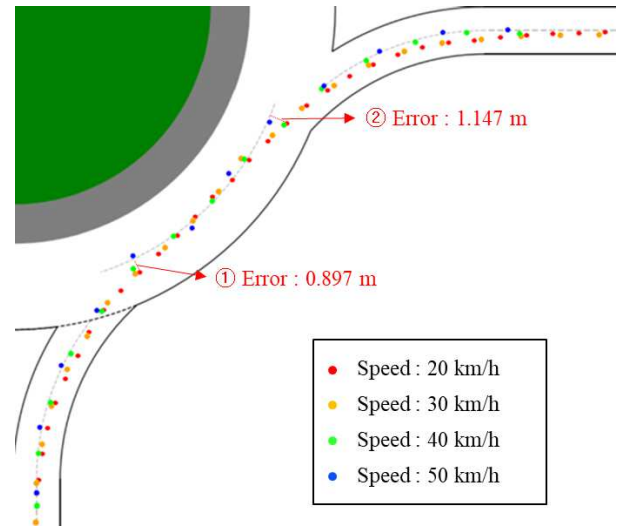


Figure 7. Trajectory according to speed change

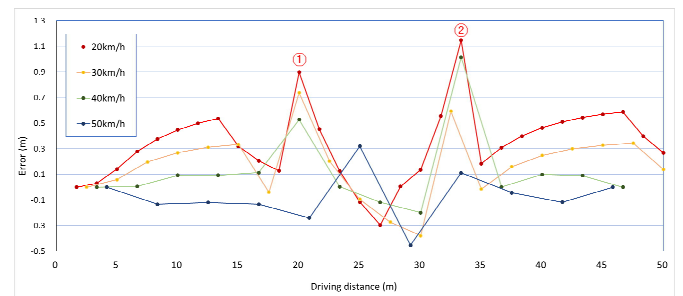


Figure 8. Error(m) from the center of the lane

0.897 m at a speed of 20km/h (part ① in Figure 7 and 8). When exiting the roundabout, an error of up to about 1.147m occurred (part ② in Figure 7 and 8). It is caused by the

curvature of the circulatory roadway when entering or exiting the roundabout.

V. CONCLUSIONS

This paper studied the trajectory control that generates paths in real time using sensors and tracking the path. We simulated in a roundabout including circular arcs and straight lines. As a result, the vehicle follows a safe trajectory, generating a path in real time from 20km/h to 50 km/h. When exiting the roundabout, an error of up to about 1.147m occurred due to the curvature of circulatory roadway. It has no effect on safe driving because the circulatory roadway is 5.5 m wide enough, and it is caused by the difference between the location of the vehicle and the center of circulatory roadway when vehicles enter or exit. In most cases, the error is less than 0.5m. The result shows that our algorithm has no problems with real time path generation and trajectory control on both straight and curved lanes.

Our algorithm only considers the single lane. Therefore, in further study, additional studies considering lane extension and lane change will be conducted.

ACKNOWLEDGMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education(2020R1A6A1A12047945).

*corresponding author is S.G. Choi (choisg@cbnu.ac.kr)

REFERENCES

- [1] Q. Yao, Y. Tian, Q. Wang and S. Wang, "Control Strategies on Path Tracking for Autonomous Vehicle: State of the Art and Future Challenges," in *IEEE Access*, vol. 8, pp. 161211-161222, 2020.
- [2] S. Kuutti, S. Fallah, K. Katsaros, M. Dianati, F. Mccullough and A. Mouzakitis, "A Survey of the State-of-the-Art Localization Techniques and Their Potentials for Autonomous Vehicle Applications," in *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 829-846, April 2018.
- [3] G. Qing, Z. Zheng and X. Yue, "Path-planning of automated guided vehicle based on improved Dijkstra algorithm," *2017 29th Chinese Control And Decision Conference (CCDC)*, pp. 7138-7143, 2017.
- [4] A. Chambers, M. Eavis-O'Quinn, V. Roberge and M. Tarbouchi, "Route planning for electric vehicle efficiency using the Bellman-Ford algorithm on an embedded GPU," *2018 4th International Conference on Optimization and Applications (ICOA)*, pp. 1-6, 2018.
- [5] C. Liu, S. Lee, S. Varnhagen and H. E. Tseng, "Path planning for autonomous vehicles using model predictive control," *2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 174-179, 2017.
- [6] J. M. Snider, "Automatic Steering Methods for Autonomous Automobile Path Tracking," *Technical Report CMU-RI-TR-09-08, Carnegie Mellon University Robotics Institute*, 2009.
- [7] A. Domina and V. Tihanyi, "Comparison of path following controllers for autonomous vehicles," *2019 IEEE 17th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, pp. 147-152, 2019.
- [8] S. Dominguez, A. Ali, G. Garcia and P. Martinet, "Comparison of lateral controllers for autonomous vehicle: Experimental results," *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1418-1423, 2016.
- [9] A. Eskandarian, C. Wu and C. Sun, "Research Advances and Challenges of Autonomous and Connected Ground Vehicles," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 2, pp. 683-711, Feb. 2021.
- [10] D. C. Andrade et al., "A Novel Strategy for Road Lane Detection and Tracking Based on a Vehicle's Forward Monocular Camera," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 4, pp. 1497-1507, April 2019.
- [11] A. Gupta and A. Choudhary, "A Framework for Camera-Based Real-Time Lane and Road Surface Marking Detection and Recognition," in *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 4, pp. 476-485, Dec. 2018.
- [12] R. P. D. Vivacqua, M. Bertozzi, P. Cerri, F. N. Martins and R. F. Vassallo, "Self-Localization Based on Visual Lane Markings: An Accurate Low-Cost Approach for Autonomous Driving," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 2, pp. 582-597, Feb. 2018.
- [13] C. Lu, S. Xia, W. Huang, M. Shao and Y. Fu, "Circle detection by arc-support line segments," *2017 IEEE International Conference on Image Processing (ICIP)*, pp. 76-80, 2017.
- [14] (2020) The Wikipedia website. [Online]. Available: [https://en.wikipedia.org/wiki/Intersection_\(Euclidean_geometry\)](https://en.wikipedia.org/wiki/Intersection_(Euclidean_geometry))
- [15] (2021) The Wikipedia website. [Online]. Available: https://en.wikipedia.org/wiki/Turning_radius
- [16] (2014) The MOLIT website. [Online]. Available: <https://www.codil.or.kr/filebank/mocroadguide/LS/CIKCLS126036/CIKCLS126036.pdf>



Hye Young An received B.S. degree in the College of Computer Science, Chungnam National University, Korea in 2001. She is currently a M.S. candidate in School of Electrical & Computer Engineering, Chungbuk National University. Her research interests include Autonomous vehicle, Optimization, AI, IoT.



Won Seok Choi received B.S. and Ph.D. degrees in the College of Electrical and Computer Engineering, Chungbuk National University, Korea in 2008 and 2015, respectively. He is currently a researcher at the Research Institute of Computer and Information Communication (RICIC), Chungbuk National University. His research interests include Vehicle network, AI, Energy saving network, SDN, NFV, and NGN.



Seong Gon Choi received his B.S. degree in electronics engineering from the Kyeongbuk National University, in 1990, as well as his M.S. and Ph.D. degrees from the Information Communications University, South Korea, in 1999 and 2004, respectively. He is currently a Professor at the College of Electrical and Computer Engineering, Chungbuk National University. His research interests include V2X, AI, smart grid, IoT, mobile communication, high-speed network architecture, and protocol.