

자동 주차 시스템의 경로 생성 및 추종 알고리즘 개발

신 희 석¹⁾ · 김 명 준¹⁾ · 김 정 하^{*2)}국민대학교 자동차전문대학원 자동차IT융합전공¹⁾ · 국민대학교 자동차IT융합학과²⁾

Path Generation and Tracking Algorithm of Auto Valet Parking System Development

Hee-Seok Shin¹⁾ · Myeong-Jun Kim¹⁾ · Jung-Ha Kim^{*2)}¹⁾Graduate School of Automotive Engineering, Kookmin University, Seoul 02707, Korea²⁾College of Automotive Engineering, Kookmin University, Seoul 02707, Korea

(Received 30 December 2019 / Revised 9 March 2020 / Accepted 23 April 2020)

Abstract : This paper proposes a method for creating a path for parking autonomous vehicles, regardless of the distance and the position of the parking space, by using a vehicle with sensor performance superior to the existing ultrasonic and camera-based sensors. Although conventional methods have a limited parking detection area, the proposed method has a greater free parking detection area. In addition, a new tracking algorithm was employed in order to follow the generated path. Meanwhile, a new algorithm was developed for the longitudinal controller to perform perfect parking in order to eliminate the ‘path following error’ caused by the difference between the steering speed and the vehicle speed. The path planning and path tracking method for the Auto Valet Parking System was then experimentally demonstrated.

Key words : Auto valet parking system(자동 주차 시스템), Autonomous vehicle(자율주행자동차), Path planning(경로계획), Path tracking(경로추종), Path generation(경로생성), Vehicle control(차량 제어)

Nomenclature

x_1, x_2, x_3, x_4 : x coordinate point, m
 y_1, y_2, y_3, y_4 : y coordinate point, m
 p_0, p_1, p_2, p_3 : parking point, m
 r_1, r_2, r_3, r_4, R : radius, m
 δ : ackerman angle, degree
 Φ : yaw angle, degree
 $v(t)$: vehicle speed, m/s
L : vertical length, m
 L_p : width, m
 O_1, O_2, O_3, O_4 : circle, m
 d_1, d_2, d_3, d_4 : distance from vehicle, m
 M_1, M_2, M_3, M_4 : extra space, m

1. 서론

최근 자율주행 차량이 증가됨에 따라, 차량에 장착되어지는 Sensor의 성능도 좋아지고 가격도 점차 감소하고 있다. 때문에 일반 상용화된 최근 출시되는 차량에도 좋은 성능의 Sensor가 점차 장착되어지는 추세이다. 최근 Bosch사와 Daimler사는 벤츠 박물관에서 Full auto valet parking을 선보인 바 있다. 운전자는 차량을 Parking 시작 위치에 두고 내리면, 차량은 스스로 Valet parking system을 시작한다. 건물 각 기둥에 있는 통신모듈 및 LiDAR (Light Detection And Ranging) Sensor와 V2X 통신을 통해 차량의 위치와 빈 주차공간의 위치를 인식해 주차공간까지 차량 스스로 이동하여 주차를 완료할 수 있게 된다. 하지만, 이러 한 현재 개발되어진 Valet parking은 건물과 기둥에 인프라 구축이 필수조건이라는 단점을 가지고 있다. 이처럼 인프라를 구축하여 Valet parking을 선보인 회사는 Continental, Audi 등이 있다. Daimler와 마찬가지로 주차장 기둥 및 주차공간에 인프라를 구축하여 시연을

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

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

선보였다. 또한 도로에는 많은 비 자율주행차량이 존재하는데, 미국 Stanley사에서는 이러한 비 자율주행차량을 위한 Parking system을 개발 중이다. 차량에 Auto parking system이 장착되어있지 않아도, 차량을 Valet zone에 위치해 놓게 되면, 바퀴가 달린 리프트형태의 Stanley robot이 Valet zone에 위치한 차량을 리프팅 시켜 Robot이 스스로 야외 주차장으로 Valet parking을 수행한다. 하지만 마찬가지로 인프라 구축이 필요하다는 단점을 가지고 있다. 이처럼 많은 자율주행회사에서 Auto valet parking system이 개발됨에 따라 Sensor의 높은 성능을 요구하게 되었고 자율주행 Sensor 수요가 증가되고 이로 인해 Sensor 업계의 경쟁이 격화되면서 ADAS(Advanced Driver Assistance Systems) Sensor의 가격이 감소가 되었다. 이렇게 Sensor의 개발은 점차 발전되어 높은 성능의 Sensor를 저렴한 가격으로 사용할 수 있지만, 이러한 발전하고 있는 Sensor의 기반을 둔 차량 Planning과 Control은 현재 제자리에 머무는 추세이다. 본 논문에서는 인프라 구축이 필요 없이 증가된 Sensor 성능에 기반을 둔 자율주차를 위한 새로운 Path planning과 Vehicle control을 제안한다.

2. 시스템 구조

본 논문에서 사용된 실험차량은 Kia사의 Sportage 차량을 자율주행으로 개조한 차량으로 사용하였다.

2.1 차량 Hardware

본 연구에서 제안하는 차량은 일반적인 주차장에서



Fig. 1 A platform of Kia - Sportage

Table 1 A spec table of Kia - Sportage vehicle platform

Size	4400×1855×1635 mm (Overall×Length×Height)
Weight	1548 kg
Displacement	1995 cc
Maximum power	184 HP
Maximum torque	41.0 kg·m
Steering angle	-520° ~ +540°
Steering speed	112.3 deg/s (Unloaden vehicle in Asphalt)
Steering ratio	16.4 : 1

Auto parking이 가능한 차량을 기반으로 연구되었기 때문에 일반적인 SUV차량인 Kia-Sportage를 개조한 차량으로 실험을 진행하였다.

Steering, Accelerator, Brake, Gear shift는 각 Board를 장착하여 제어하는데 사용하였으며, System 구동을 위해 12 V 배터리를 추가 장착하여 사용하였다.

2.2 차량 Software

본 논문의 실험에서는 제어를 위한 Windows 10 기반의 산업용 PC 1대를 구성하여 Lab view 프로그램을 사용하였고 Ubuntu 기반의 노트북 PC 1대를 ROS 프로그램을 사용하여 Sensor를 위한 PC를 구성하였다. 각 PC와 Sensor는 Ethernet을 통해 연결하였으며, 각 PC간 통신은 Ethernet을 사용한 UDP통신을 사용하였다.

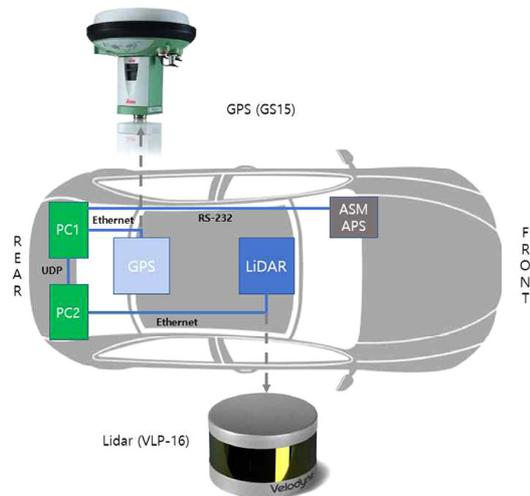


Fig. 2 A sensor configuration diagram

2.3 시스템 Architecture

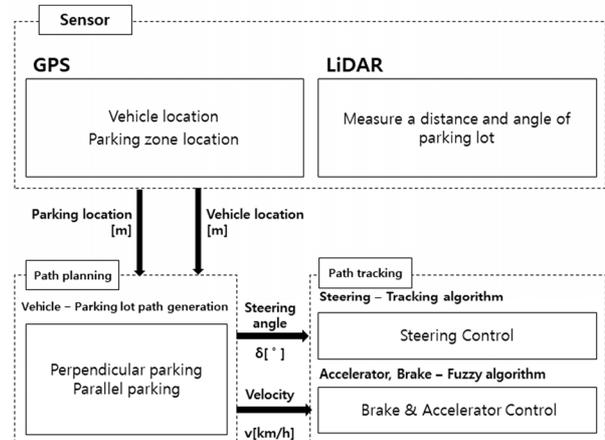


Fig. 3 A system architecture of autonomous vehicle

전체 시스템 Architecture의 구성과 순서는 Fig. 3과 같다. Sensor 시스템에서는 LiDAR와 GPS(Global Positioning System)의 정보를 기반으로 차량으로부터 주차공간의 상대 좌표 값을 받아온다. Path planning 시스템에서는 Sensor 시스템의 출력 값을 기반으로 직각주차 및 평행주차의 Path를 생성시켜 목표 조향 값과 목표 속도 값을 출력한다. Path tracking 시스템에서는 Steering 제어 값과 Brake 및 Accelerator 값을 연산하여 출력한다.

3. Sensor System

3.1 LiDAR (VLP-16)

차량 플랫폼을 제외한 H/W의 경우 Velodyne사의 VLP-16과 Leica 사의 GS15를 사용하여 인지 및 항법시스템을 구성하였으며, 제어기의 경우 산업용 PC 1 대를 사용하였다.

LiDAR는 Laser 광원을 이용하는 Range finder로써, 본 실험에서는 VLP-16과 PC간의 UDP 통신을 이용하여 Raw data 패킷을 받아오고 파싱 후 가공한 Point cloud 데이터들을 Sensor position과 Pose 값을 반영 하여 지면을 제거한다. 지면제거는 RANSAC(Random Sample Consensus) 알고리즘을 사용하였으며 샘플링 횟수는 100회씩 진행하였다. 지면이 제거되고 난 Point cloud는 유클리디언 클러스터링 알고리즘을 이용하여 군집화를 진행하고 이를 통해 내 차량 주변 콘의 크기 특징 점을 추출하여 측위를 진행한다.

Table 2 LiDAR sensor spec table

Model name	Velodyne VLP-16
Detect angle	360° Horizontal / ±15° Vertical
Resolution	0.1° Horizontal / 2° Vertical
Frequency	5-20 Hz
Detection distance	100 m

3.2 GPS (Leica - GSS15)

GPS는 정확한 자차위치와 주차 공간 위치 정보를 받기위해 RTK GPS를 사용하였다. RTK(Real Time Kinematic) 실시간 이동측위라는 개념으로 정밀한 위치정보를 가지고 있는 기준국의 반송파 위상에 대한 보정치를 이용하여 이동국에서 실시간으로 1~2 cm 정확한 측위 결과를 얻는 일련의 측량 과정을 말한다. 본 실험에서 사용된 RTK GPS의 오차범위는 수평에서 약 8 mm, 수직으로는 약 15 mm의 오차를 가지며, 최대 위성 수는 60개를 가질 수 있다. 반응시간 또한 1초미만이기 때문에 정확한 위치 정보를 신뢰할 수 있다.

Table 3 GPS sensor spec table

Model name	Leica GS15
Weight	1.34 kg
Error range	8 mm Horizontal, 15 mm Vertical
Number of satellites	Up to 60 Satellites
Reaction time	1 sec under

3.3 Sensor 구성

차량의 무게중심점인 뒤 차축 중심에 GPS를 장착하였으며, LiDAR는 전방의 주차공간의 상대좌표를 받기위해 전면 Roof 쪽에 장착하였다.

4. Path Planning

4.1 경로 생성 알고리즘 (평행주차)

4.1.1 기존 2개의 원호를 이용한 경로생성

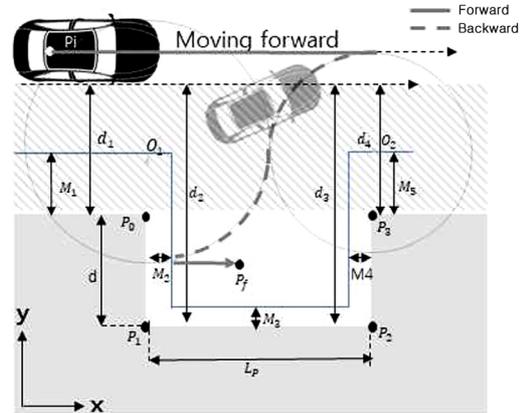


Fig. 4 Impossible of parking path using 2-Arcs

4.1.2 개선된 4개의 원호를 이용한 경로생성

$$M_1, M_2, M_3, M_4 : \text{extra space} \tag{1}$$

$$p_0, p_1, p_2, p_3 : \text{Parking point} \tag{2}$$

$$d_1, d_2, d_3, d_4 : \text{distance from the vehicle} \tag{3}$$

$$x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4 : \text{각 원의 } x, y \text{ 값} \tag{4}$$

$$y_4 = y(p_3) + r \text{ (최소회전반경)} \tag{5}$$

$$O_1 = P_0 \text{로부터 전폭의 } 1/2 \text{ 지점을 지나고 } O_2 \text{와 접점인 원} \tag{6}$$

$$O_2 = P_1 \text{와 같은 } y \text{ 선상에 위치해 있으며 } O_1 \text{과 접점인 원} \tag{7}$$

$$O_3 = P_i \text{를 지나고 } y(\text{parking point}) - r \text{을 중심으로 갖는 원} \tag{8}$$

$$O_4 = \text{반지름 } r \text{을 갖고 } O_3 \text{와 외접하는 원} \tag{9}$$

$$y(O_4) = y(\text{parking point}) + r_4 \tag{10}$$

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} = r_1 + r_2 \tag{11}$$

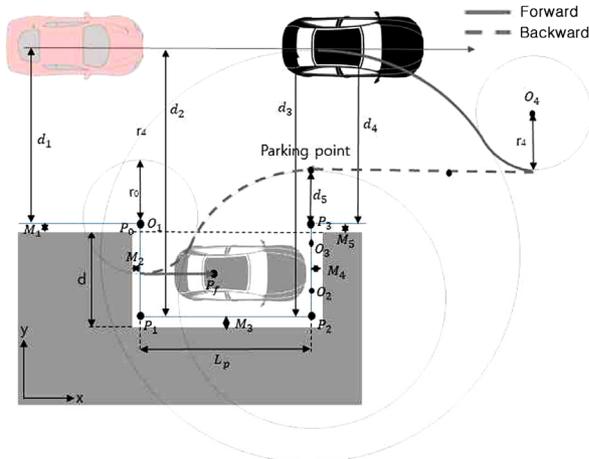


Fig. 5 Parking path planning using 4-Arcs

Fig. 4는 기존 주차 경로생성 방법으로써, 두 개의 최소 회전반경인 동심원을 사용하여 주차를 수행하는 방법이다. 차량이 주차 가능 공간 밖에 있게 되면 첫 번째 원이 커지게 되고 상대적으로 두 번째 원은 작아지면서 차량의 최소회전반경에 미치지 못하기 때문에 결국, 차량이 추종 할 수 없는 Path를 생성 시키게 되므로 주차를 수행할 수 없게 된다.¹⁻³⁾

Fig. 5는 기존에 사용하는 2개의 원호를 이용하는 방법이 아닌 4개의 원호를 이용하여 차량의 주차 공간 탐지 영역에 구애받지 않고 어떠한 위치에서도 차량이 주차 Path를 생성 시킬 수 있는 방법이다. 이는 Auto valet parking system에서 중요한 역할이라 할 수 있다. Path생성의 설명은 다음과 같이 설명할 수 있다.^{4,5)}

1. 차량이 차지 위치 및 Heading 값을 미리 인지하고 주차 가능 공간의 크기 및 위치를 Sensor를 통해 정보를 제공 받는다.
2. 원 O_1 은 차량의 최소회전반경을 고려하여, 주차 공간 꼭짓점인 P_0 을 중심으로 갖고 주차 공간 폭의 1/2점을 지나는 원을 생성한다.
3. 원 O_2 는 Parking point를 지나면서 원 O_1 을 접점으로 하는 원이다. Parking point는 차량의 주차 수행 기준점으로 두었다. 또한 O_2 의 중심점은 차량이 주차공간에 가장 가까이 붙어있을 경우 기준하여 최소회전반경 지점으로 지정하였다.
4. 원 O_4 는 차량의 최소회전반경을 갖는 원으로서, 차량이 Parking point 지점으로 이동하기 위한 원이다. 때문에 Parking point와 동일 선상에 두어 최소회전반경 만큼의 값을 Shift 시켜 원의 중심으로 두었다. 이렇게 4개의 원호를 이용하면 차량이 위치에 구애 받지 않고 주차 경로를 생성 시킬 수 있다.

4.2 경로 생성 알고리즘 (직각주차)

4.2.1 기존 동심 원호를 이용한 경로생성

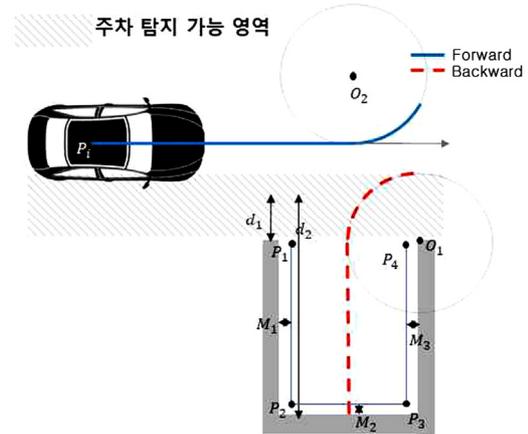


Fig. 6 Impossible of parking path using original method

4.2.2 개선된 가변 원호를 이용한 경로생성

$$x_1 = x(p_1) = x(p_4) \quad (12)$$

$$O_1 = \text{주차공간의 중심점을 지나고 최소회전반경을 갖는 원} \quad (13)$$

$$O_2 = \text{차량과 } x\text{축 동일선상에서 } O_1\text{과 접점을 갖는 원} \quad (14)$$

기존 직각 주차 방법은 차량이 주차 탐지 가능영역 내에 있으면 Path를 생성 시킬 수 있지만, 차량이 주차 탐지 가능 영역을 벗어나게 되면 원간의 접점이 생기지 않아 결국 주차 Path를 생성 시킬 수 없게 된다.

Fig. 7은 기존 최소회전반경을 갖는 동심원을 이용한 Path생성 방법이 아닌, 서로 다른 반경을 갖는 가변원호를 이용한 Path를 생성 시키는 방법이다. 이러한 방법을 사용하게 되면 주차 가능영역의 제한도 받지 않는다는 장점을 가질 수 있다. 설명은 다음과 같다.

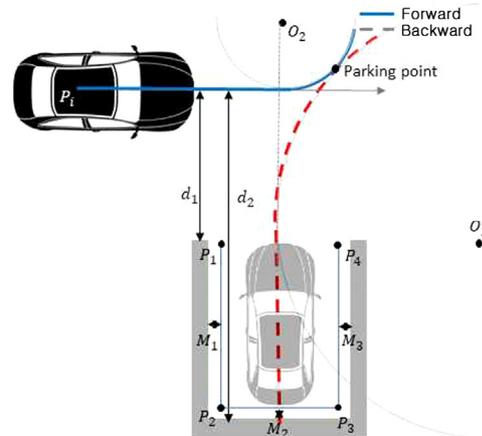


Fig. 7 Parking path using variable arcs

1. 차량의 자치위치 및 차량의 Heading 값을 인지한 후 목표 주차장의 Point 지점을 검출 및 인지한다.
2. 두 개의 가변원호를 이용한 Path를 생성시킨다.
3. 원 O_2 생성 : 차량의 위치 점과 동일한 x축 선상에 원 O_2 의 x축을 두고 최소회전반경 만큼 y축으로 방향으로 Shift 된 곳에 y축을 두어 중심점을 두었다.
4. 차량이 후진으로 주차구역에 진입 시 좌우측에 장애물이 있어도 충돌 없이 진입하기 위해서 장애물 P_1 , P_4 와 x축 동일선상에 O_1 의 y축 중심점을 두었다. 이는 원 O_1 과 후진 시 직선 Path와 접점을 갖기 위함이다.
5. 최종적으로 원 O_2 와 접하면서 x축을 가변적으로 이동시켜 원 O_1 을 생성시켰다.

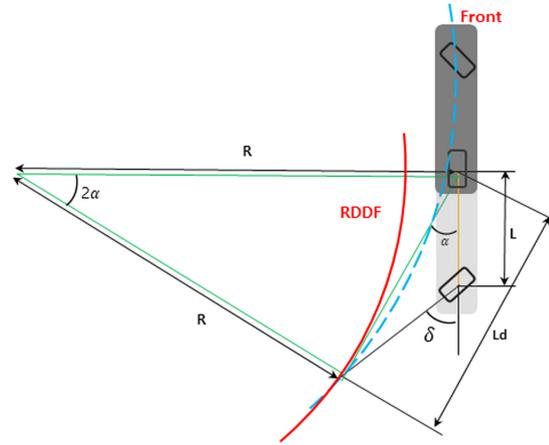


Fig. 9 Bicycle model of backward Pure pursuit

이를 이용한 후방 Pure pursuit 알고리즘이며, 가상의 차량을 차량 뒤에 두어 역 Steering angle을 산출하는 방법으로 적용하였다. Pure pursuit 특성상 LD 값이 차량 속도에 비해 너무 짧게 되면 차량이 경로를 추종하기 위해 좌우로 심하게 움직이면서 Oscillation이 발생하게 된다. 이는 식 (17), 식 (18)에서 볼 수 있듯이 LD값이 감소하면 Steering angle(δ)이 증가하게 되는데 증가하는 만큼 실시간으로 변화량($\Delta\delta$) 또한 증가하게 되는 이유이다. 반대로 LD값이 너무 길게 되면 Corner cut 현상이 발생하게 된다. 따라서 Pure pursuit을 이용하기 위해서는 적절한 LD 값 튜닝이 필요하다.

5. Path Tracking

5.1 Pure pursuit을 활용한 전/후방 추종 방법

기존 제어 분야에서 많은 추종알고리즘 중, 가장 많이 사용하는 기법은 Pure pursuit이나 Stanley method 방법을 사용한다. 본 논문에서는 Pure pursuit과 Stanley method의 단점을 보완한 새로운 Path 추종 알고리즘을 제안한다.⁶⁾

$$\frac{Ld}{\sin 2\alpha} = \frac{R}{\sin(90 - \alpha)} \quad (15)$$

$$\frac{Ld}{\sin \alpha} = 2R, k = \frac{1}{R} \quad (16)$$

$$\delta = \tan^{-1} \frac{2Ls \sin \alpha}{Ld} \quad (17)$$

$$\delta = - \tan^{-1} \frac{2Ls \sin \alpha}{Ld} \quad (18)$$

Fig. 8은 기존에 가장 많이 사용하고 있는 경로 추종기법 중 하나인 Pure pursuit 알고리즘이다. Pure pursuit은 차량의 Heading 각도와 LD(Look ahead Distance) 값의 각도 오차 값을 이용하여 Steering angle을 산출한다. Fig. 9는

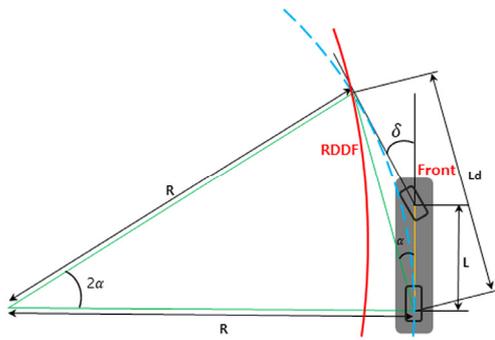


Fig. 8 Bicycle model of forward pure pursuit

5.2 Stanley method을 활용한 전/후방 추종 방법

$$X(t) : \text{앞차축과 가장 가까운 경로상의 수직거리} \quad (19)$$

$$\phi : \text{경로와 차량의 앞차축 Yaw 값 차이} \quad (20)$$

$$\delta : \text{Ackerman 조향각} \quad (21)$$

$$\delta(t) = \phi(t) + \tan^{-1} \left(\frac{kx(t)}{v(t)} \right) \quad (22)$$

$$\delta(t) = - \left[\phi(t) + \tan^{-1} \left(\frac{kx(t)}{v(t)} \right) \right] \quad (23)$$

Fig. 10은 Stanley method를 Bicycle model로 나타낸 그림이다. Stanley method 경우, 차량의 앞 차 축으로부터 수직선상의 가장 가까운 경로상의 각도와 차량 각도 오차를 이용하여 Steering angle을 산출하는 방법이다. 이 방법은 식 (22)와 식 (23)에서 볼 수 있듯이 차량의 차속이 증가하게 되면 전체적인 조향 각이 반비례로 감소하므로 비교적 Oscillation이 적게 발생된다는 특징을 가지고 있다. 이를 활용하여 Fig. 11과 같이 차량의 뒷부분에 가상의

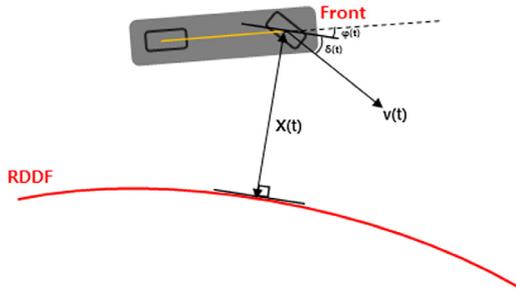


Fig. 10 Bicycle model of forward Stanley method

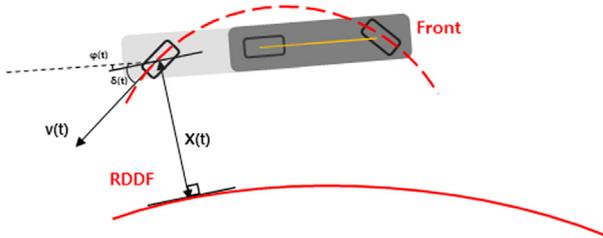


Fig. 11 Bicycle model of backward Stanley method

차량을 두어 역 Steering angle 산출을 통해 후방 Stanley method를 적용하였다. 하지만, Stanley method의 경우, 바퀴로 부터 기준점을 사용하여 추종하므로 Corner cut이 심하다는 단점을 가지고 있다.⁷⁻⁹⁾

5.3 개선된 경로 추종 알고리즘

위의 SSP의 앞서 설명한 Pure pursuit과 Stanley method는 각각 장, 단점이 존재한다. Pure pursuit의 경우 LD (Look ahead Distance)에 따라 차량이 Corner cut과 Oscillation이 발생 될 수 있다. 또한, Stanley method의 경우 앞 차축 기준으로 경로를 추종하므로 뒤따라오는 뒤 차축이 안쪽으로 파고들어 Corner cut 현상이 두드러지게 나타나게 된다. 따라서 두 가지의 장점과 단점을 보완하여 다음과 같은 새로운 경로추종 알고리즘인 SSP(Shin Stanley Pursuit)을 제안한다.

SSP는 Stanley method의 식 (23)과 같다. 다만, Stanley method의 단점인 Corner cut현 상을 방지하기 위해 앞 차축 중심이 아닌 뒤 차축 중심으로부터 Path와 차량의 Heading각의 오차인 $\phi(t)$ 를 산출하여 Steering angle을 계산한다. 이렇게 뒤 차축중심으로 Steering angle을 산출하면 Oscillation 발생이 심하게 나타나게 된다. 따라서 식 (23)과 같이 앞 차축으로부터 경로와 차량의 거리 $x(t)$ 를 고려하고 분모에 있는 속도 $v(t)$ 가 증가됨에 따라 반비례적으로 전체적인 Steering angle인 $\delta(t)$ 가 줄어들면서 Oscillation 현상을 줄일 수 있다. 추가적으로 Fig. 13과 같이 전방 SSP를 활용하여 차량의 뒷부분에 가상의 차량을 구성하여 역 조향 각 산출을 통해 후방 SSP를 구현하였다.

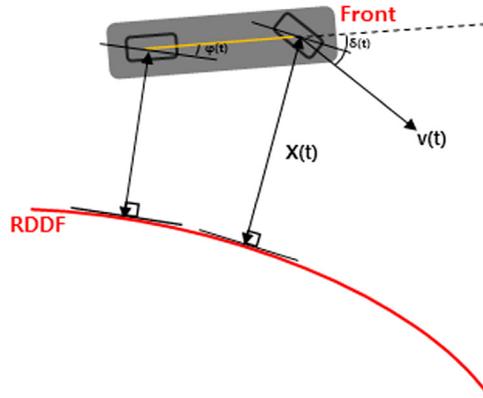


Fig. 12 Bicycle model of forward SSP (Shin Stanley Pursuit)

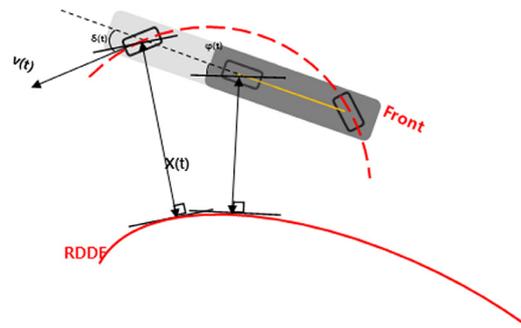


Fig. 13 Bicycle model of backward SSP (Shin Stanley Pursuit)

6. Longitudinal Control

직각주차와 평행주차실험을 진행하는데 있어, 원호가 비교적 많은 평행주차의 경우 Steering speed가 차량의 속도보다 상대적으로 느려 Error가 발생하는 현상이 발생되게 된다. Fig. 14와 같이 Steering speed가 차량의 속도보다 빠르게 되는 경우 문제없이 차량은 Path를 추종할 수 있지만, Fig. 15와 같이 Steering speed가 차량의 속도보다 느린 경우 Overshoot이 발생되면서 차량은 정확한 Path를 추종하기 어려워진다. 식 (26)은 Steering speed에 따른 Steering angle을 나타낸다. 식 (27)의 목표 Steering angle에 도달하기 위해서는 Steering speed가 Vehicle speed와 같거나 빨라야 한다. 따라서 Path의 곡률에 따라 Steering speed에 기반하여 정확하고 효율적으로 Path를 추종할 수 있는 종 방향 계획이 필요하다.¹⁰⁾

$$l = R \cdot \theta \quad (R: \text{각원의 반경}) \quad (24)$$

$$\tan \delta = \frac{L}{R}, \quad L = 2.54m \quad (25)$$

$$\theta = w_s \cdot \alpha \cdot t \quad (26)$$

(w_s : Steering angular speed, α : Steering ratio, θ : Steering \angle)

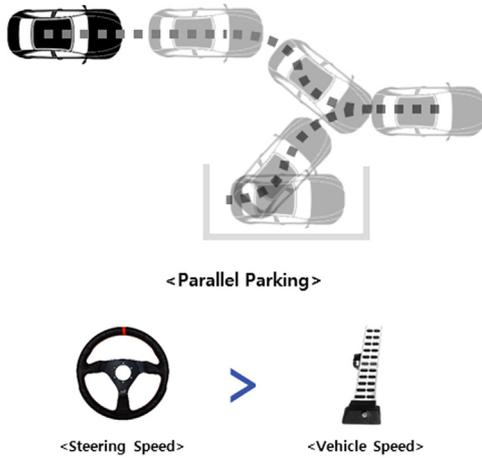


Fig. 14 Accurate following along the path

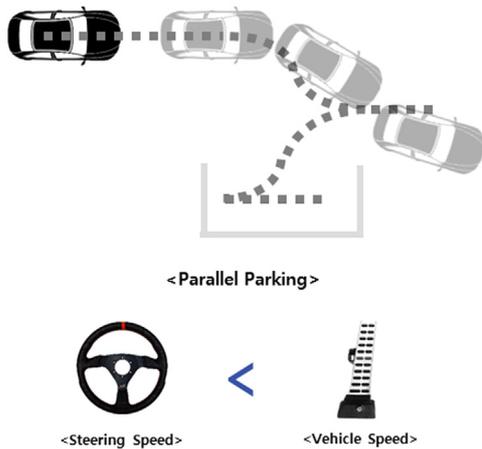


Fig. 15 Impossible of accurate following along the path

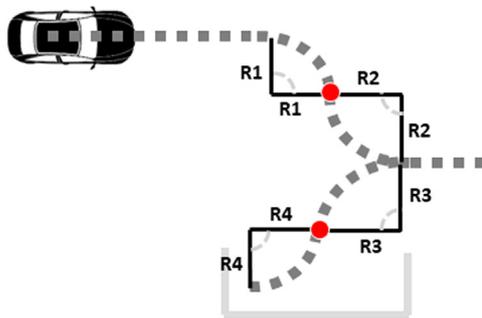


Fig. 16 Curvature calculation of each path using bearing angle

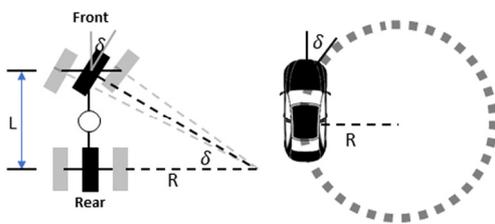


Fig. 17 Ackerman steering

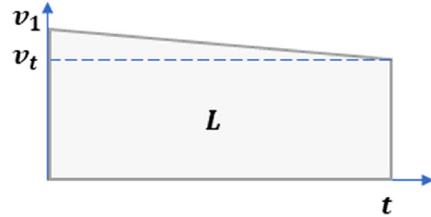


Fig. 18 Vehicle speed - Time graph

$$\delta = w_v \cdot t \quad (27)$$

$$(w_v = v : \text{Vehicle speed}, \alpha : \text{목표 Steering } \angle)$$

$$t_a(\text{목표 Steering angle 도달시간}) = \frac{\delta_t - \delta_1}{v_a} \quad (28)$$

$$v_1 = \text{현재 차량속도} \quad (29)$$

$$v_t = \text{목표 차량속도} \quad (30)$$

$$L = \text{차량 위치 - 목표 Path 원호길이} \quad (31)$$

$$t_t(\text{목표 Path 수렴 가능 시간}) = \frac{2L}{v_1 + v_t} \quad (32)$$

다음은 중 방향 제어를 하기 위한 방법 설명이다.

1. Fig. 16과 같이 곡률에 맞게 차량의 중 방향을 제어하는데 있어, Path의 직선과 곡률을 판단하여야 한다. 본 논문에서는 원호를 이용한 Path를 생성시키는 과정을 거쳤기 때문에 정확하게 직선과 곡선의 Path를 판단할 수 있다.
2. 실험에 사용된 차량은 일반 차량으로 Fig. 17에서와 같이 Ackerman 조향에 따른다. 이 Ackerman 조향에 기반하여 차량의 Steering ratio 적용에 따른 목표 곡률을 판단하고 그에 따른 Steering angle을 계산하게 된다.
3. 목표 Steering angle을 계산하면 현재 Steering angle까지의 원호를 계산한다.
4. 목표 Steering angle 도달시간과 목표 Path 수렴 가능 시간은 같으므로 같게 두고, Fig. 18과 같이 현재 차량속도에서 목표 Steering angle에 도달하기 위한 Steering speed에 따른 목표차량속도를 계산한다.

$$v_t \leq \frac{2v_a L}{\Delta \delta} - v_1, (\Delta \delta = |\delta_t - \delta_1|) \quad (33)$$

차량의 현재 위치에서부터 목표 Path 곡률 원호까지의 길이를 L로 두고 목표 Steering angle 도달시간이 목표 Path에 차량이 수렴 가능한 시간보다 높아야 비교적 적은 Error로 Path를 정확하게 추종할 수 있다. 따라서 최종적으로 식 (33)이 도출되게 된다.

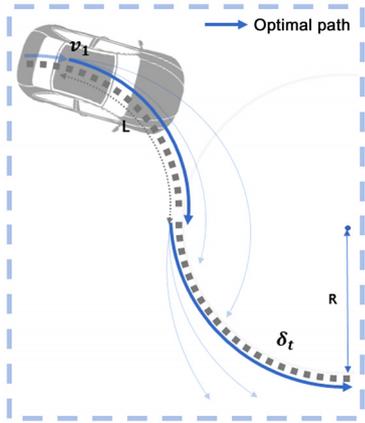


Fig. 19 Optimal path following curvature



Fig. 20 Place of experiment : Seoul grand park

7. 실험 환경

실험 장소는 과천에 위치하고 있는 서울대공원 주차장에서 진행하였다. Fig. 23과 같이 직각주차와 평행주차 모두 차량의 LiDAR로부터 각 주차 공간에 해당하는 꼬깔콘 위치의 상대좌표 값을 산출하여 Path를 생성시킨 뒤 Path를 추종하는 방법으로 실험을 진행하였다.¹¹⁾

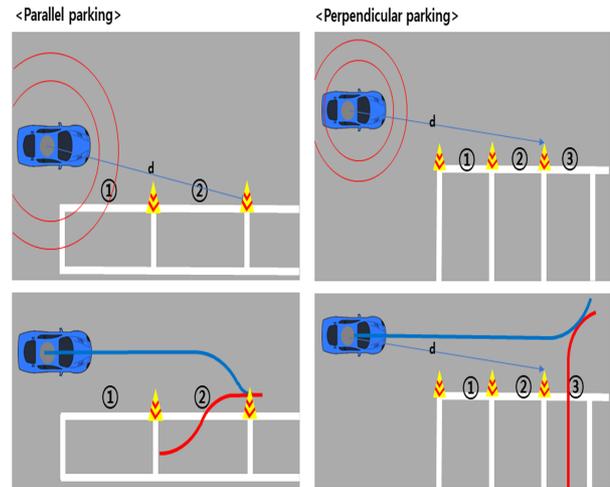


Fig. 23 Experiment of perpendicular and parallel parking

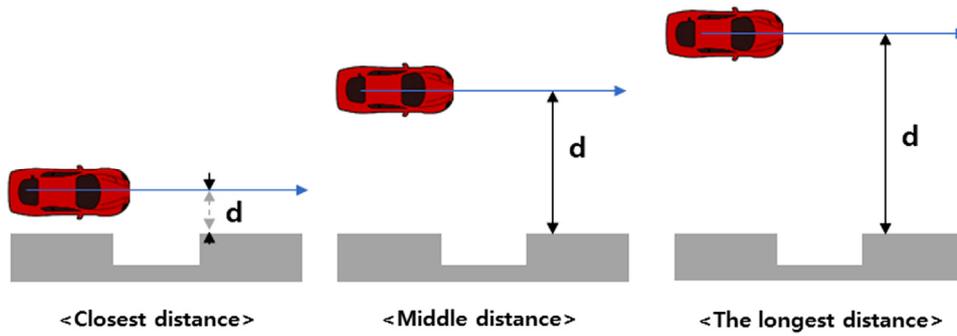


Fig. 21 Experiment : Path generate for parallel parking

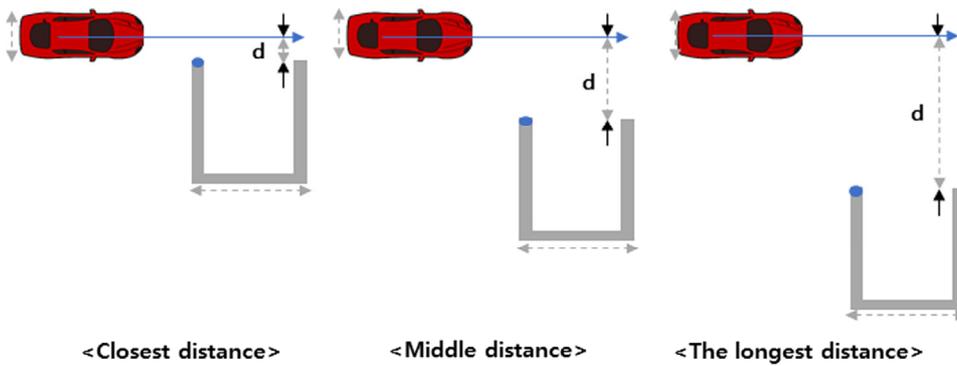
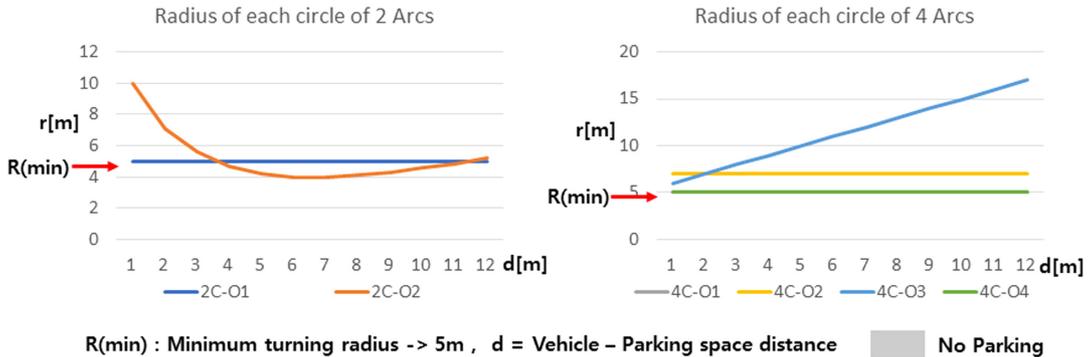


Fig. 22 Experiment : Path generate for perpendicular parking

8. 결과

8.1 개선된 평행주차 Path 생성 결과 값

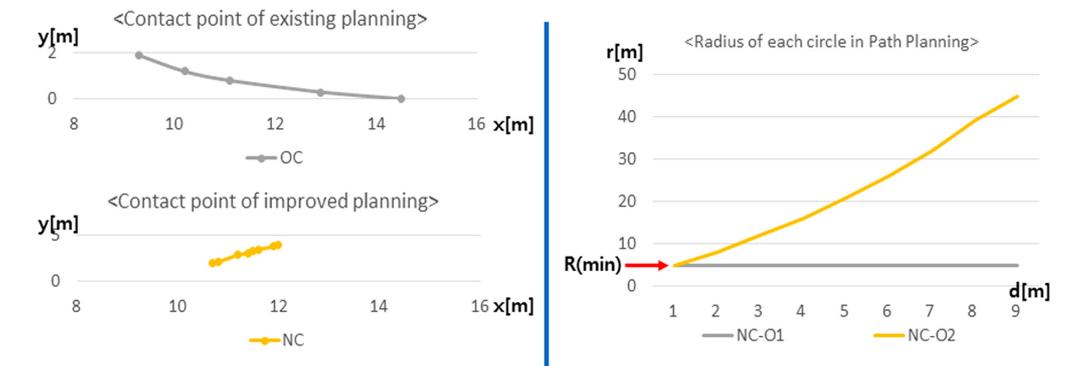


Radius of each circle using 2 Arcs (r) [m]												
D	1	2	3	4	5	6	7	8	9	10	11	12
2C-01	5	5	5	5	5	5	5	5	5	5	5	5
2C-02	10	7.1	5.6	4.7	4.2	4	4	4.1	4.3	4.6	4.8	5.2

Radius of each circle using 4 Arcs (r) [m]												
d	1	2	3	4	5	6	7	8	9	10	11	12
4C-01	5	5	5	5	5	5	5	5	5	5	5	5
4C-02	7	7	7	7	7	7	7	7	7	7	7	7
4C-03	6	7	8	9	10	11	12	13	14	15	16	17
4C-04	5	5	5	5	5	5	5	5	5	5	5	5

Fig. 24 Result of path generate for parallel parking

8.2 개선된 직각주차 Path 생성 결과 값



Contact point of each circle using existing planning [m]									
d	1	2	3	4	5	6	7	8	9
OC-X	9.3	10.2	11.1	12.9	14.5	X	X	X	X
OC-Y	1.9	1.2	0.8	0.3	0	X	X	X	X

Contact point of each circle using improved planning [m]									
d	1	2	3	4	5	6	7	8	9
NC-X	10.7	10.8	11.2	11.4	11.5	11.6	11.9	12	12.1
NC-Y	2	2.1	2.9	3.1	3.3	3.5	3.8	4	4.2

Radius of each circle using existing planning r [m]									
d	1	2	3	4	5	6	7	8	9
OC-01	5	5	5	5	5	X	X	X	X
OC-02	5	5	5	5	5	X	X	X	X

Radius of each circle using improved planning r [m]									
d	1	2	3	4	5	6	7	8	9
NC-01	5	5	5	5	5	5	5	5	5
NC-02	5	8.2	12	16.2	21	26.2	32	39.2	45

Fig. 25 Result of path generate for perpendicular parking

8.3 직각주차 및 평행주차 Path 추종 결과 값

실험은 Path 생성 부분과 Path 추종 부분으로 총 두 가지로 진행하였다. 먼저, Path 생성 부분은 Fig. 21과 Fig. 22와 같이 주차장과 차량의 거리를 증가시켜 실험을 진행하였다. 먼저 Fig. 24의 평행주차 Path 생성 결과 값을 보면, 기존 2개의 원호 방법을 이용한 결과는 주차장과 차량의 거리가 4 m 이상부터 2개의 원호 중 1개의 원호가 차량의 최소회전반경을 벗어난다는 것을 확인 할 수 있었다. 반면, 4개의 원호를 이용한 결과는 모든 거리에서 원호 모두 최소회전반경 이하의 값을 가졌으며 주차장과 차량 간 거리가 증가함에도 문제없이 Path를 생성시킬 수 있다는 것을 보여준다.

다음으로, Fig. 25의 직각주차 Path 생성 결과 값을 보면, 기존 방법을 사용하였을 때는 차량과 주차장간 거리가 6 m 이상부터 접점을 가지지 않으므로 Path를 생성시키지 못하는 것을 확인 할 수 있다. 반면, 개선된 방법을 사용하면, 모든 거리에서 접점을 가질 수 있을 뿐 아니라, 각 원의 반경 또한 최소회전반경을 만족시키는 값을 가지므로 문제없이 Path를 생성시킬 수 있다는 것을 보여준다.

Fig. 26은 직각주차와 평행주차의 각 Tracking 알고리즘에 대한 결과이다. 위에서 설명했던 내용을 바탕으로 생성된 경로에 대한 추종성능을 비교하였다. Pure pursuit의 경우, 차량의 뒤 차축으로부터 LD값을 산출하여 추종을 하므로 급격한 코너에서 Corner cut이 발생하는 것을 확인할 수 있었다. Stanley method의 경우, 차량의 앞 차축 중심으로 Path를 추종하기 때문에 마찬가지로 뒤따라오는 뒤 차축이 과교들면서 Corner cut이 발생하였다. 반면, 새로운 추종 기법인 SSP를 사용한 결과를 보게 되면 뒤

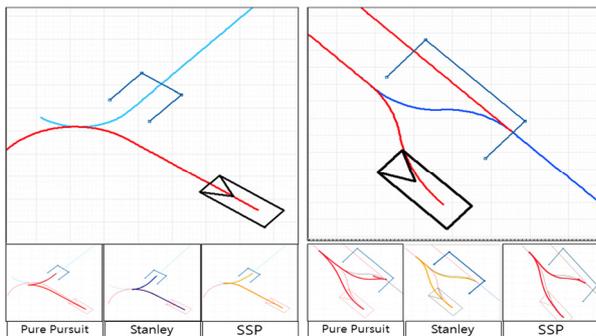


Fig. 26 Result of perpendicular and parallel parking

Table 4 RMSE(Root Mean Square Error) of each Parking

	Original	Pure Pursuit	Stanley	SSP
perpendicular	0%	37%	27%	11%
Parallel	0%	41%	31%	12%

차축 중심점으로부터 Path에 대한 상대 각도를 계산하여 조향 각을 산출하였기 때문에 비교적 적은 Corner cut이 발생된 것을 확인 할 수 있다.

Table 4는 수치적 Data를 분석하기 위해 차량의 뒤 차축 중심의 궤적이 Path를 추종하며 이동한 실제 차량 궤적과 생성된 Path의 차이를 계산하여 RMSE(Root Mean Square Error)를 분석하였다. 따라서 기준점인 Path는 RMSE가 0%이며, 직각주차의 결과는 Pure pursuit은 37%, Stanley method는 27%, SSP는 11%로 SSP의 RMSE 값이 가장 적은 오차를 갖는다는 것을 확인 할 수 있었고 이는 가장 높은 정확도를 갖는다는 것을 확인 할 수 있다. 평행주차의 결과는 Pure pursuit은 41%, Stanley method는 31%, SSP는 12%로 마찬가지로 SSP를 사용한 추종기법이 가장 높은 정확도를 갖는다는 것을 확인 할 수 있었다. 평행주차의 결과를 보면 직각주차의 결과보다 SSP가 다른 추종알고리즘에 비해 월등히 Corner cut 발생이 적다는 것을 확인 할 수 있다. 이는 평행주차 Path를 생성시키는 과정에서 더 많은 원호를 이용하여 Path를 생성시킴으로써 선회구간이 많기 때문에 이러한 결과가 나왔다는 것을 확인 할 수 있다. 따라서 SSP는 선회구간이 많고 주차와 같은 미세한 Path를 추종하는데 있어 좋은 성능을 갖는다는 것을 확인할 수 있다.

후 기

기존의 주차 경로 생성 알고리즘은 기존 초음파센서 기반으로 경로를 생성시키기 때문에 2개의 원호를 일반적으로 사용하였다. 본 논문에서는 4개의 원호 또 가변원호를 사용함으로써 증가된 Sensor의 성능에 따라 원거리에서도 탐지한 후 주차 경로를 생성시킬 수 있는 알고리즘으로 주차성능을 증가시켰을 뿐만 아니라, 기존의 확률기반인 Fuzzy알고리즘과 오차가 비교적 많이 발생하는 추종 알고리즘이 아닌 증가된 성능의 경로를 잘 추종하는 알고리즘을 구현함으로써 오차를 줄여 주차성능을 확보하였다.

References

- 1) K. S. Kim, Development of an Automatic Parallel Parking System Using Three Arcs, M. S. Thesis, KAIST, Daejeon, 2009.
- 2) H. S. Shin and J. H. Kim, "Develope of an Automatic Parallel Parking Path Planning Using four Arcs," KSAE Spring Conference Proceedings, pp.689-693, 2018.
- 3) J. H. Yoo and D. S. Kim, "Path-planning and Tracking-controller of Autonomous Parking Assis-

- tance System,” KSAE Annual Conference Proceedings, p.2385, 2008.
- 4) H. S. Shin and J. H. Kim, “Development of an Auto Valet Parking System using Variable Arcs,” Annual meeting of the Korean Institute of Electronics Engineers, pp.549-550, 2018.
 - 5) M. G. Kim H. S. Shin, K. I. Min and J. H. Kim, “AVM Camera Based Indoors Auto-parking System using Image Matching Method for Autonomous Vehicle,” Annual Meeting of the Korean Institute of Electronics Engineers, pp.547-548, 2018.
 - 6) J. M. Snider, “Automatic Steering Methods for Autonomous Automobile Path Tracking,” Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, 2009.
 - 7) T. Sebastian, M. Mike, D. Hendrik, S. David, A. Andrei, D. James, F. Philip, G. John, H. Morgan, H. Gabriel, L. Kenny, O. Celia, P. Mark, P. Vaughan, S. Pascal, S. Sven, D. Cedric, J. Lars-Erik, K. Christian, M. Charles, R. Carlo, N. Joe, J. Eric, A. Philippe, B. Gary, D. Bob, E. Scott, K. Adrian, N. Ara and M. Pamela, “Stanley: the Robot That Won the DARPA Grand Challenge,” Journal of Field Robotics, Vol.23, Issue 9, pp.661-692, 2006.
 - 8) J. M. Kim, K. I. Lim and J. H. Kim, “Development of Path Planning System of Autonomous Vehicle for Auto-Parking,” KSAE Fall Conference Proceedings, pp.690-695, 2015.
 - 9) S. W. Jang, H. W. Chang, S. K. Kim and W. J. Jung, “Backward Path-Tracking Motion Control Algorithm for a Semi-Auto Parking System,” KSAE Spring Conference Proceedings, pp.2116-2121, 2007.
 - 10) Y. B. Gil and J. H. Kim, Research on Driving Target Point and Velocity Planning Based on Global Path Curvature for Path Tracking of Autonomous Vehicle, M. S. Thesis, Kookmin University, Seoul, 2018.
 - 11) M. J. Kim, H. S. Shin and J. H. Kim, “The Perception System based on LIDAR Sensor for Auto Valet Parking Systems,” Journal of Institute of Control, Robotics and Systems, Vol.26, No.7, pp.617-624, 2019.