

# 자동 발렛 파킹을 위한 횡방향 제어기 설계 및 차종변화에 대한 제어 성능 분석

## Design of Lateral Controller for Automatic Valet Parking and Its Performance Analysis with Respect to Vehicle Types

최 희 재, 송 봉 섭\*  
(Heejae Choi<sup>1</sup> and Bongsob Song<sup>1</sup>)  
<sup>1</sup>Ajou University

**Abstract:** The unified lateral control algorithm for automatic valet parking for various types of vehicles is presented and its feasibility is shown experimentally via field tests for the given parking scenario. First, a trajectory generation algorithm for forward driving and backward multi-step parking maneuvers is developed. Then, with consideration of different types of vehicles and operating conditions, a kinematic vehicle model is used and validated using field test data. Using the nonlinear vehicle model, the lateral controller is designed based on dynamic surface control. Finally the proposed lateral control law is validated via hardware-in-the-loop simulations for different types of vehicles and experimentally using a test vehicle through field tests.

**Keywords:** lateral control, automatic valet parking, multi-step perpendicular parking, dynamic surface control

### I. 서론

IT 기술이 급속도로 빠르게 발전함에 따라 스마트 폰, 스마트 TV와 같은 새로운 제품이 지속적으로 개발되고 있으며 이러한 기술은 의료, 농기계, 자동차, 조선 등과 같은 기존 산업과 융합하고자 하는 시도들이 최근에 급증하고 있다. 특히 자동차의 경우 스마트 카 또는 커넥티드(connected) 자동차라는 이름으로 IT 기술이 접목되고 있으며 그 중에서도 통신과 DVI (Driver-Vehicle Interface) 기술과의 융합이 특히 주목 받고 있다[1]. DVI 기술의 경우 음성 인식이나 모션 인식을 통하여 운전자의 의도를 차량에 전달하기 위한 기술 개발이 주를 이루고 있으며 주로 스마트 폰, DMB, MP3와 같은 IT 기기를 차량 내에서도 쉽게 사용하고 운전 방해가 되지 않도록 하기 위한 기술로 요약될 수 있다.

통신 기술의 경우 이미 오래 전부터 다양한 형태로 차량 또는 도로 인프라와 접목되어 개발되고 있다. 미국의 경우 2003년에 VII (Vehicle Infrastructure Integration)라는 프로젝트를 시작하여 DSRC (Dedicated Short-Range Communications) 기반 V2V와 V2I 통신을 개발하였으며 2008년과 2009년에 이러한 기술들이 얼마나 실질적으로 잘 동작하는지를 검증하기 위한 실증(Proof-of-Concept) 테스트가 미시건 Oakland County와 캘리포니아 Palo Alto에서 각각 수행되었다[2]. 더 나아가 2009년도 12월에 발표한 미국 교통국 ITS (Intelligent Transport Systems) Strategic Research Plan (2010~2014)에 의하면 연구 개발 전략이 크게 차량, 인프라, 그리고 보행자의 IT 기기를 연결

하여 안전성, 이동성, 그리고 친환경성의 효율을 최대화 하기 위한 기술로 요약될 수 있다[3]. 일본과 유럽에서도 Smartway 21 프로젝트나 i2020 Intelligent Car Initiative라는 이름으로 자동차와 IT 기술을 융합하는 연구가 활발히 진행되고 있다[4].

우리나라의 경우 2007년부터 스마트 하이웨이라는 사업명으로 IT, 자동차, 도로 기술을 융합하여 실시간으로 정보를 공유하고 전천후, 주야간 안전주행 지원기술 구현을 목표로 미래의 지능형 고속도로 개발 연구를 진행하고 있다[5]. 또한 2010년부터 도심지역에서 특히 공항이나 백화점과 같이 주차를 위해 많은 시간을 소비하고 이로 인해 발생하는 환경적 문제를 효율적으로 해결하기 위하여 기존의 차량중심으로 상용화되고 있는 주차지원시스템(parking assist system)을 한 걸음 더 진보시켜 자동으로 발렛 파킹을 할 수 있는 스마트 주차 시스템 개발이 진행되고 있다[6].

주차에 관련된 연구는 이미 상용화되어 운전자에게 후방 비전센서를 통하여 후면 주차 공간을 보여주고 초음파센서를 통하여 충돌경고를 해주는 주차지원시스템부터 운전자의 개입 없이 자동주차를 해 주는 지능형 주차지원시스템으로 진화하고 있다. 더 나아가 자동주차 관련 기술은 무인 자율 주행 기술과 통합되어 자율 주행 및 자동 주차를 수행하는 자동차가 개발되고 있다. 2007년 미국의 DARPA에서 개최한 Urban Challenge라는 대회를 통해 도시의 교통환경에서 주행 및 주차까지 가능함을 선보였으며[7,8], 2010년에는 미국 Google사에서 1,600 Km 이상을 무인으로 주행하였고 이후 225,000 Km 이상을 무인으로 주행했다고 보고하고 있다. 하지만, 현재 개발되고 있는 무인 자율 주행 자동차는 다양한 주변 환경을 인지해야 하기에 추가적으로 많은 고성능 센서들이 장착되어야 하며 이러한 센서들을 가지고 있더라도 일반 주차장 환경에서 최적의 주차공간을 탐색하는 것은 쉽지 않은 문제이다.

\* 책임저자(Corresponding Author)

논문접수: 2012. 8. 27., 수정: 2012. 9. 14., 채택확정: 2012. 9. 25.

최희재, 송봉섭: 아주대학교 기계공학과

(chj372@ajou.ac.kr/bsong@ajou.ac.kr)

※ 본 사업은 지식경제부의 지식경제 기술혁신사업(10035250, 자동발렛파킹을 위한 센서기반 공간인지 및 자동주행 기술개발)의 지원을 받아 수행되었음.

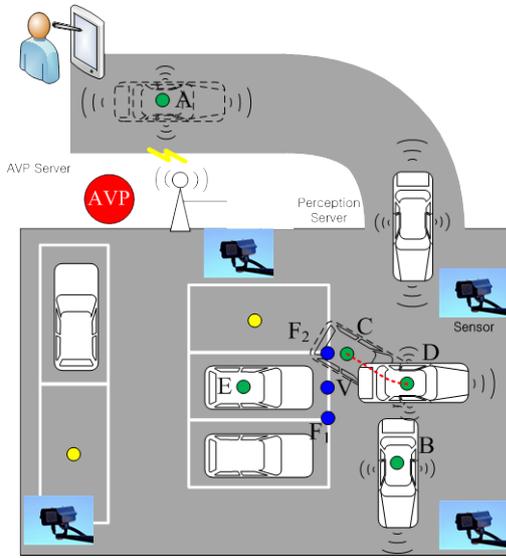


그림 1. 자동 발렛 파킹 시스템의 개념도.  
Fig. 1. Schematic of AVP.

이러한 한계점을 보완하기 위한 방법 중 하나로 제한된 영역에서 자율주행과 자동주차를 지원하는 인프라 기반 자동 발렛 파킹(AVP: Automatic Valet Parking) 시스템 (또는 스마트 주차 시스템) 개발이 연구되고 있다. 그림 1에서 보는 바와 같은 AVP 시스템을 개발하기 위해서는 통신, 인지, 판단, 제어 등 다양한 IT 융합 기술들이 접목되어야 하며 본 논문에서는 인프라에서 제공하는 통신을 기반으로 횡방향 차량 제어를 위한 제어기 설계에 대해서 중점적으로 살펴보려 한다. 기존의 횡방향 차량 제어는 단일 차종에 대하여 주로 한번에 주차하기 위한 경로를 생성하는 차량중심의 주차지원 시스템을 위한 횡방향 제어 연구가 주로 이루어진 반면에[9-11], 인프라와의 연동을 통한 자동 발렛 파킹을 수행하기 위해서는 다음과 같은 문제를 고려하여 횡방향 제어기를 설계해야 하며 본 논문에서 이러한 문제점에 대한 접근방법 및 해결방안을 제안하고자 한다.

- i. 인프라기반으로 다양한 차종에 대한 주차지원이 가능해야 한다.
- ii. AVP를 위해 필요한 자동 주행 및 다단계 주차(multi-step parking)를 모두 지원해야 한다.
- iii. 위치 오차 또는 외란에 강인해야 하며 이로 인한 제어기의 성능저하가 적어야 한다.
- iv. 마지막으로 실제 차량을 이용하여 제안되는 알고리즘을 검증하기에는 많은 비용과 노력이 예상되며 다양한 차종을 고려하여 제어기를 검증할 수 있는 개발 플랫폼이 필요하다.

**II. AVP 주행 시나리오**

AVP를 위한 주행 시나리오는 그림 1에서 보는 바와 같이 A지점부터 B지점까지 자율주행과 B지점부터 E지점까지 자동주차로 나눌 수 있다. Waypoint라 불리는 주행 경로는 그림 1에서의 AVP 서버에서 계획되어 무선 통신을 통하여 차량에 전달된다. A지점에서 B지점 사이 자율주행 구간에서 곡선구간을 포함하고 있는 경우 조밀한 waypoint를 생성하여 곡선

로 주행을 위한 주행 궤적을 생성할 수 있도록 해야 한다. B 지점에서 E지점으로 직각주차를 수행하는 경우 주변 환경, 예를 들어서 한번에 주차를 수행할 수 있는 경우와 주차공간이 협소하거나 장애물이 있어 여러 번에 걸쳐서 다단계(multi-step) 후진 주차를 수행해야 하는 상황이 발생할 수 있다. 다단계 후진 주차를 수행해야 하는 경우 그림 1에서 보는 바와 같이 B점에서 C점으로 후진 주차 후 다시 D점으로 전진 후 E점으로 후진 주차를 하여 주차를 완료하게 된다.

다음으로 AVP 시스템은 그림 2에서 보는 바와 같이 크게 네 부분으로 나누어 생각해 볼 수 있다[6]. 운전자에게 자동 발렛 파킹에 대한 정보를 실시간으로 전달할 수 있는 Nomadic device, AVP를 하려는 차량을 등록하고 최적의 주차 공간을 탐색하여 해당 경로를 생성해주는 Global Situation-Aware Server, 분산되어 있는 인프라 기반 센서로부터의 측정값을 이용하여 차량의 위치나 요(yaw)를 측정하고 예측하는 Local Spatial-Aware Server, 마지막으로 종/횡방향으로 차량을 제어하기 위한 차량 제어기(vehicle controller)로 구성이 되어 있다. 좀 더 구체적인 주요 기능들이 그림 2에 정리되어 있다.

차량 제어기의 경우 그림 2와 3에서 보는 바와 같이 차량에 장착이 되어 있으며 종/횡방향 차량 제어를 위한 센서 및 액추에이터를 포함하고 있다. 그림 3의 경우 프로토타입(prototype)으로 개발되고 있는 전기자동차이며 개발의 효율성을 위해서 다중 프로세서를 사용하여 분산 시스템으로 개발하고 있다. 또한 Hatchback 타입이나 SUV 타입의 자동차에 대한 프로토타입도 동시에 개발을 진행하고 있다.

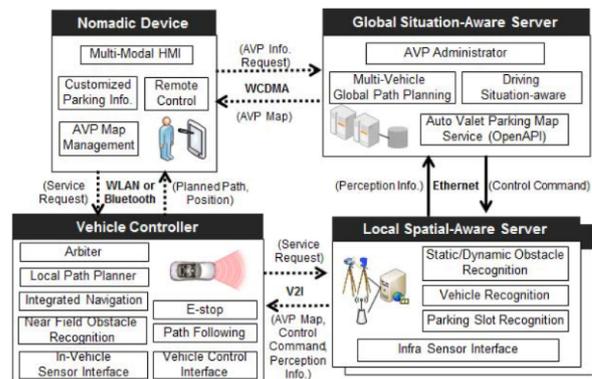


그림 2. 자동 발렛 파킹 시스템의 구조.  
Fig. 2. Architecture of AVP system (refer to [6]).

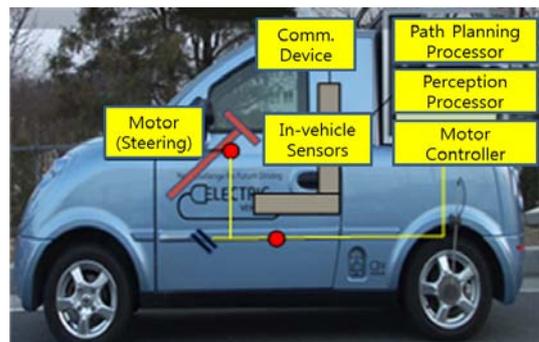


그림 3. 차량 제어기를 포함하고 있는 개발 플랫폼.  
Fig. 3. Hardware platform for vehicle controller of AVP system.

III. 궤적 생성 알고리즘

궤적 생성을 위한 방법은 크게 원과 직선을 이용하는 방법 [9,11-13], curve-fitting 방법[10], 그리고 clothoid 기반으로 생성하는 방법[14,15] 등이 제안되고 있다. 본 장에서는 전진 주행과 직각 주차를 위하여 curve-fitting 기법을 이용한 궤적 생성 알고리즘을 사용하고자 한다. 전진 주행을 위한 궤적 생성은 주어진 두 waypoint 사이를 cubic Bezier curve interpolation을 이용하여 주행 가능한 궤적을 생성하는 방법을 사용하였다[10]. 후진 주차의 경우 원과 직선을 사용하여 평행 주차를 위한 궤적을 생성한 방법[11,12]과 달리 curve-fitting 기법을 사용하여 후진 직각 주차 또는 다단계 직각 주차를 위한 궤적 생성 알고리즘을 제안하고자 한다.

후진 직각 주차를 위한 궤적 생성 알고리즘은 세 단계로 구성되어 있다. 첫 번째 단계는 주차 전략의 구성(configuration of parking strategy)이다. 즉, 궤적을 생성하기 전에 주어진 주차 공간의 구조와 차량의 위치를 따라 직각 주차가 한 번에 이루어질 수 있는지를 판단하는 단계이다. 그림 4에서 보는 바와 같이 Ackermann angle을 고려한 선회 반경을 가진 원을 그릴 수 있고 그 원의 반지름은 다음과 같이 계산된다.

$$R_{\min} = \frac{L}{\sin(\delta_{\max})} \tag{1}$$

여기서  $\delta_{\max}$  최대 조향각이며  $L$  은 휠베이스(wheelbase)의 길이이다. 또한, 이 원의 중심 점( $c_0$ )은 다음과 같다.

$$c_0 = \begin{bmatrix} l_c \cos\left(\psi + \frac{\pi}{2}\right) + x_r \\ l_c \sin\left(\psi + \frac{\pi}{2}\right) + y_r \end{bmatrix} \tag{2}$$

여기서  $P(x_r, y_r)$ 은 후방 차축 중심점을 뜻하며,  $\psi$  는 요(yaw), 그리고  $l_c$  는  $P(x_r, y_r)$ 와  $c_0$  사이의 거리이다(그림 4 참조).

다음으로  $c_0$ 에서 실제 차량의 무게 중심,  $P(x, y)$ , 까지의 거리를 실제 차량의 선회반경( $R_0$ )이 정의되고  $\delta_{\max}$ 인 경우 차량의 선회반경을  $R_{a,\min}$ 으로 정의할 수 있다. 따라서 정의된  $R_{a,\min}$  이 그림 1이나 5에 보는 바와 같이 현재 차량위치에서 주차 입구지점(그림 1과 5에서 지점  $V$  참조)과의 거리보다 작다면 한번에 후진 직각 주차가 가능하다고 판단할 수 있다.

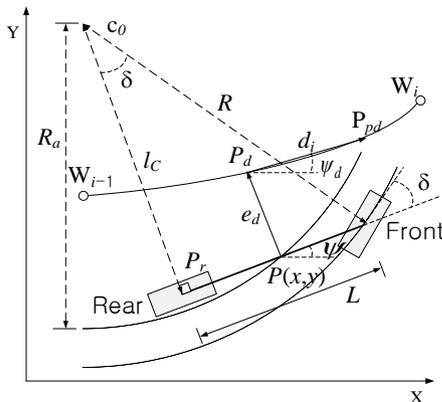


그림 4. 차량 모델 변수 및 오차의 정의.  
Fig. 4. Variables of vehicle model and definition of errors.

즉, 한번에 후진 직각 주차가 가능한지를 판단하는 조건은 다음과 같다.

$$R_{a,\min} \leq \|P(x, y) - V\| \tag{3}$$

식 (3)을 만족한다면, 즉 한 번에 후진 직각 주차가 가능하다고 판단된다면 두 번째는 주차 궤적 후보 생성(trjectory candidate generation) 단계로 cubic Bezier curve interpolation 기법을 사용하여 다음과 같이  $n$  개의 궤적 후보를 생성할 수 있다.

$$p_{di}(u) = \begin{cases} a_i u^3 + b_i u^2 + c_i u + p_0 & \text{if } y \geq V_y \\ \begin{bmatrix} x_d \\ y_d \end{bmatrix} & \text{otherwise} \end{cases} \tag{4}$$

여기서  $u \in [0,1]$ ,  $p_0$ 는 주차 시작 위치 (그림 5에서  $B$  위치), 그리고 벡터  $a_i, b_i, c_i$  는 다음과 같이 정의된다.

$$c_i = 3(p_{1i} - p_0) \in \mathbb{R}^2, \quad b_i = 3(p_{2i} - p_{1i}) - c_i \in \mathbb{R}^2, \\ a_i = p_3 - p_0 - b_i - c_i \in \mathbb{R}^2$$

여기서  $p_3$  는 주차 구역의 입구 (그림 5에서  $V$  지점),  $p_{1i}$ 와  $p_{2i}$  는 다음과 같이 정의된다.

$$p_{1i} = \begin{bmatrix} p_0 + l_i \cos(\varphi_1) \\ p_0 + l_i \sin(\varphi_1) \end{bmatrix}, \quad p_{2i} = \begin{bmatrix} p_3 + l_i \cos(\varphi_2) \\ p_3 + l_i \sin(\varphi_2) \end{bmatrix} \tag{5}$$

다양한 완만함을 갖는 주차 궤적은 (5)의  $l_i$ 을  $n$ 개로 차등적으로 선택함으로써 생성되며 다음과 같이 선정될 수 있다.

$$l_i = \gamma_i \|p_0 - p_3\| \text{ for } i = 1, \dots, n$$

여기서  $\gamma_i \in (0,1)$  는 차등적으로  $l_i$ 을 생성하기 위한 상수이다. 또한,  $\varphi_1$ 과  $\varphi_2$ 는

$$\varphi_1 = \tan^{-1}\left(\frac{y - c_{0,y}}{x - c_{0,x}}\right) + \frac{\pi}{2}, \quad \varphi_2 = \beta$$

여기서  $\beta$  는 그림 5에서 보는 바와 같이 점  $E$ 와  $V$ 를 이은 선분의 각도이다.

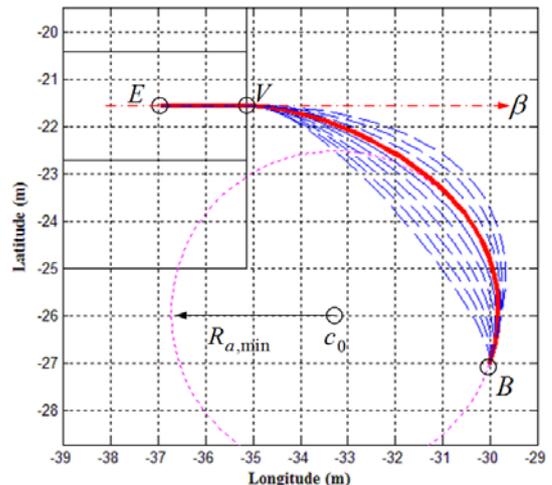


그림 5. 직각주차를 위한 궤적 생성.  
Fig. 5. Trajectory generation for perpendicular parking.

마지막 단계는 위에서 생성한 궤적 후보들 중에 최적의 궤적을 선택하는 궤적 선택(trajjectory selection)이다. 주차 궤적 후보 중에서 하나의 궤적을 선택하기 위해서는 두 가지의 제한 조건을 만족해야 한다. 즉 차량의 기하학적 구조를 고려하여 주행 가능한 궤적이어야 하며 주차 입구 진입 시 조향각의 변화가 가장 작은 궤적이어야 한다. 따라서, 먼저 생성된 궤적의 초기 위치와 선회 반경의 중심( $c_0$ )와의 거리가 최소 선회반경( $R_{a,min}$ )보다는 작아야 한다는 조건을 만족하는 궤적 후보들 중에서 그 거리가 가장 큰 것을 선택한다. 이것을 수식으로 요약하면 다음과 같다.

$$\max_{\substack{1 \leq i \leq n \\ 0 \leq u \leq \alpha}} \|p_{di}(u) - c_0\| \quad \text{subject to } \|p_{di}(u) - c_0\| \leq R_{a,min} \quad (6)$$

여기서  $\alpha$  는 그림 5의 B점 주변의 궤적 일부를 선택하기 위한 상수이다. 그림 5는 차량의 위치가 B로 주어졌을 경우 10개의 궤적 후보가 점선으로 생성되었고 그 중 (6)에 의해 최종 궤적이 선정되었음을 보여준다(그림 5의 실선 참조).

식 (3)의 조건을 만족하지 않는 경우는 다단계 직각 주차가 필요하다. 다단계 직각 주차를 위한 궤적을 생성하기 위해서는 다음과 같은 세 단계로 구성되어 있다. 첫 번째 단계는 그림 6에서 보는 바와 같이 차량의 위치와 주차 구역의 입구(V)와의 거리가  $R_{a,min}$  보다 작을 경우 추가적인 waypoint(그림 6의 I 참조)를 생성한다. 여기서 추가적인 waypoint의 위치는 주차 구역의 진입입구 선을 포함하는 직선의 방정식과  $c_0$ 를 중심으로  $R_{a,min}$  반경을 갖는 원과의 교점으로 결정된다. 다음으로 위에서 설명한 바와 같이 (4)를 이용하여 궤적 후보가 생성된다. 그림 6에서 보는 바와 같이 B에서 I로 궤적이 여러 개의 궤적 후보가 생성되고 (6)에 의해 파란색 실선으로 최종 주행 궤적으로 선택된다.

두 번째 단계는 차량이 한번에 E지점으로 후진 주차가 가능한 위치로 이동하는 단계이다. 차량이 후진 주행 중에 장애물이나 직선 EV를 지나가게 되면(C), 두 번째 궤적이 C에서 G 까지 cubic Bezier curve interpolation 기법을 기반으로 생성된다. 여기서, 제어 점 중 하나 인  $p_0$ 는 차량의 현재 위치(C)이며 나머지 제어 점은 다음과 같이 정의된다.

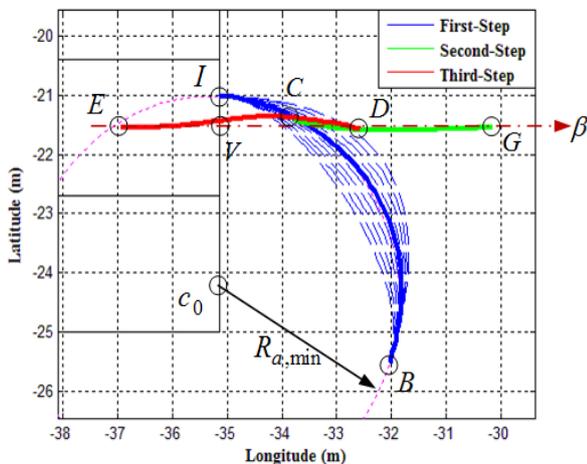


그림 6. 다단계 직각주차를 위한 궤적 생성.  
Fig. 6. Trajectory generation for multi-step perpendicular parking.

$$p_1 = \begin{bmatrix} p_0 + l \cos(\varphi_1) \\ p_0 + l \sin(\varphi_1) \end{bmatrix}, \quad p_2 = \begin{bmatrix} p_3 - l \cos(\varphi_2) \\ p_3 - l \sin(\varphi_2) \end{bmatrix}, \quad p_3 = \begin{bmatrix} G_x \\ G_y \end{bmatrix}$$

여기서

$$\varphi_1 = \psi, \quad \varphi_2 = \tan^{-1}(\beta), \quad G = \begin{bmatrix} V_x + \varepsilon \cos \beta \\ V_y + \varepsilon \sin \beta \end{bmatrix}$$

여기서  $\varepsilon$  과  $l$  은 설계상수이며 각각 5 m, 0.25가 사용되었다. 마지막으로 차량의 요(yaw)가 주차 구역 입구의 수직각( $\beta$ )과 가까워 진다면, 차량은 정지(D)하고 세 번째 궤적이 D에서 E 까지 두 번째와 유사한 방법으로 생성된다. 여기서  $p_0$ 는 현재 위치(D)이고, 남은 제어 점은 다음과 같다.

$$p_1 = \begin{bmatrix} p_0 + l \cos(\varphi_1) \\ p_0 + l \sin(\varphi_1) \end{bmatrix}, \quad p_2 = \begin{bmatrix} p_3 - l \cos(\varphi_2) \\ p_3 - l \sin(\varphi_2) \end{bmatrix}, \quad p_3 = \begin{bmatrix} E_x \\ E_y \end{bmatrix}$$

여기서,  $\varphi_1 = \psi + \pi, \varphi_2 = \beta + \pi$ .

#### IV. 횡방향 제어기 설계

고속 주행의 경우 Bicycle 모델을 기반으로 주로 제어기를 설계하는 반면 저속 주행 시 기구학적(kinematic) 모델을 사용하여 횡방향 제어기가 설계되고 있다. 본 연구에서는 비선형제어기법을 사용하여[10,11] 다양한 차종 변화에 따라 제어기가 쉽게 재설계되고 실차 주행에서도 강인한 횡방향 제어기임을 보이고자 한다.

##### 1. 차량 모델링

개별 차량을 중심으로 주행이나 주차를 위한 횡방향 제어기를 설계하는 연구가 활발하게 진행되고 있는 반면 AVP 시스템에서 요구하는 다양한 차종에 대해 여러 주행모드를 실행할 수 있는 횡방향 제어기 설계에 대한 연구는 거의 찾아보기 어렵다. 이러한 횡방향 제어기를 개발하기 앞서 우선적으로 다양한 차량의 거동을 모델링 하기 위한 차량 모델 개발이 필요하다. 차량이 10 Km/s 내외로 저속 주행을 하며 이에 따라 횡방향 미끄럼(slip angle)이 거의 없다고 가정한다면 다음과 같은 운동학적 차량 모델을 사용할 수 있다[15].

$$\begin{aligned} \dot{x} &= v \cos \psi \\ \dot{y} &= v \sin \psi \\ \dot{\psi} &= \frac{v}{L_j} \tan \delta \end{aligned} \quad (7)$$

여기서  $x$  와  $y$ 는 차량의 위치,  $\psi$  와  $\dot{\psi}$  은 차량의 요각(yaw)과 요레이트(yaw rate),  $\delta$  는 바퀴의 조향각(wheel angle)을 의미한다. 또한  $v$  는 차량 속도이며 후진 시 음수 값으로 정의되며  $L_j$  은 다양한 차종 별로 실제 축거( $L$ , length of wheelbase)를 의미한다. 다시 말해서, 아래 첨자  $j$  는 차량 종류를 나타내며 예를 들어서 CarSim이라는 상용 차량 시뮬레이터의 경우 사용하는 값을 표 1에 요약하였다[16].

표 1. 횡방향 차량 모델을 위한 상수.  
Table 1. Parameter for lateral vehicle model.

	SUV	Hatchback	Sedan	Minivan
$L_j$ (m)	2.71	2.31	2.78	3

식 (7)의 차량모델이 다양한 차종에 적용될 수 있다는 것을 보이기 위해 CarSim에서 제공하는 차종들의 주행 데이터와 실제 차량을 이용하여 획득한 주행 데이터를 사용하여 검증하였다. 첫 번째로 그림 7에서 보는 바와 같이 주어진 속도와 조향각에 대해서 차량 시뮬레이터에서 출력되는 값과 (7)의 출력값을 비교하였다. 요레이트와 요각을 기준으로 최대오차가 0.04 rad/s과 0.1 rad 임을 각각 확인할 수 있다. 표 1에 요약되어 있는 다른 차종에 대해서도 요레이트와 요각을 기준으로 차량모델에 대한 적합성을 확인하였다.

다음으로 그림 8은 중형 SUV (그림 9 참조)의 전진 주행 및 후진 직각 주차 시 데이터를 각각 나타내고 있다. 실험 시 측정된 차량 속도와 조향각을 입력으로 (7)의 차량 모델 출력 값을 요레이트와 요각을 기준으로 실제 측정 값과 비교하였다. 요레이트와 요각을 기준으로 최대오차가 0.03 rad/s과 0.09 rad 임을 각각 확인할 수 있다.

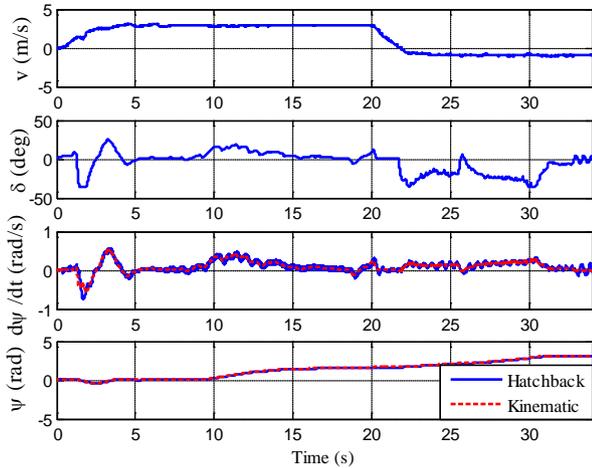


그림 7. 차량 모델과 해치백 타입의 차량의 주행 및 주차 시 요레이트와 요 값의 비교.

Fig. 7. Comparison between kinematic model and hatchback type of vehicle.

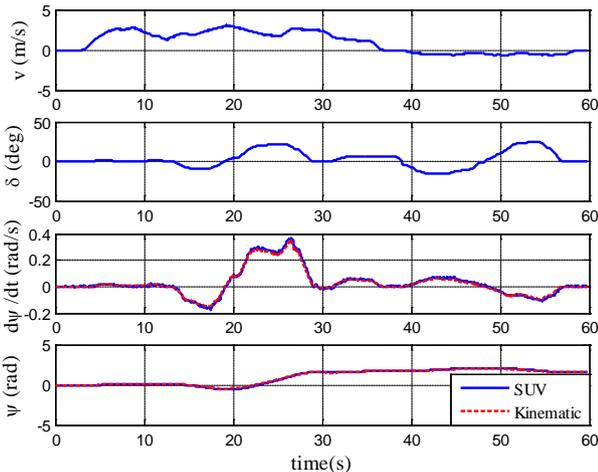


그림 8. 차량 모델과 SUV 타입의 차량의 주행 및 주차 시 요레이트와 요 값의 비교.

Fig. 8. Comparison between kinematic model and medium size SUV.

## 2. 횡방향 제어기 설계

식 (7)의 비선형 차량 모델에 대해서 동적 표면 제어 기법을 사용하여 횡방향 제어기를 설계하고자 한다[12,17,18]. 먼저 횡방향 오차에 대한 정의를 다음과 같이 정의한다.

$$S_i(t) = e_{bi} + d_i(\psi_d - \psi) \quad (8)$$

여기서

$$e_{bi} = \begin{cases} 0 & \text{if } c = 0 \\ -\text{sign}(c)\|e_d\| & \text{otherwise,} \end{cases} \quad e_d = \begin{bmatrix} x_d - x \\ y_d - y \end{bmatrix}$$

$$c = \det \begin{bmatrix} a^T \\ b^T \end{bmatrix}, \quad a = \begin{bmatrix} x_{pd} - x_d \\ y_{pd} - y_d \end{bmatrix}, \quad b = \begin{bmatrix} x - x_d \\ y - y_d \end{bmatrix}$$

아래 첨자  $i$ 는 전진 주행( $i = f$ )과 후진 주차( $i = b$ )와 같은 운전 모드를 나타내며, 좀 더 자세한 정의 및 추가적인 변수는 그림 4에 정의되어 있다. 마지막으로  $d_i$ 와  $\psi_d$ 는 다음과 같이 계산된다.

$$d_i = \|a\|, \quad \psi_d = \angle a$$

다음으로 (8)의  $S_i$ 를 미분한 후 (7)을 적용을 하면 다음과 같다.

$$\dot{S}_i = \dot{e}_{bi} + d_i(\dot{\psi}_d - \dot{\psi}) = \dot{e}_{bi} + d_i\dot{\psi}_d - \frac{d_i v}{L_j} \tan \delta \quad (9)$$

$S_i$ 를 0으로 수렴시키기 위해  $\dot{S}_i = -K_i S_i$ 을 적용하면 원하는 조향각은 다음과 같다.

$$\delta_{des} = \tan^{-1} \left[ \frac{L_j}{d_i v} (\dot{e}_{bi} + d_i \dot{\psi}_d + K_i S_i) \right] \quad (10)$$

여기서  $K_i$ 는 제어기 변수이다.

마지막으로 속도가 작은 경우 식 (7)의 분모가 작은 값이 되므로 다음과 같이 스위칭을 하는 횡방향 제어기를 최종적으로 사용한다.

$$\delta_{des} = \begin{cases} \tan^{-1} \left[ \frac{L_j}{d_i v} (\dot{e}_{bi} + d_i \dot{\psi}_d + K_i S_i) \right] & \text{if } |v| \geq \varepsilon \\ K_p S_i & \text{otherwise} \end{cases} \quad (11)$$

여기서  $K_p$ 는 비례제어기의 제어상수이며  $\varepsilon$ 는 스위칭 조건을 위한 상수(threshold)이다.

## V. 알고리즘 검증

자동 발렛 파킹(AVP)를 위해서 개발된 횡방향 제어기의 성능 평가를 위해 두 가지 방법을 사용하였다. 우선 모델의 불확실성이나 측정 노이즈에 대해서 얼마나 강인한지를 검증하기 위하여 실제 차량을 기반으로 주행 실험을 통하여 제어기의 성능 검증을 수행하였다. 다음으로는 통신지연이 존재하는 상황 및 다양한 차량에 적용될 수 있음을 보이기 위해서 Hardware-in-the-loop 시뮬레이터를 기반으로 검증하였다.

우선 횡방향 제어기 알고리즘을 평가하기 위해 그림 9와 같이 자율주행 테스트 차량을 사용하여 (11)에서 제안된 제어기가 전진 주행과 후진 주차 시 적절한 조향을 자동으로

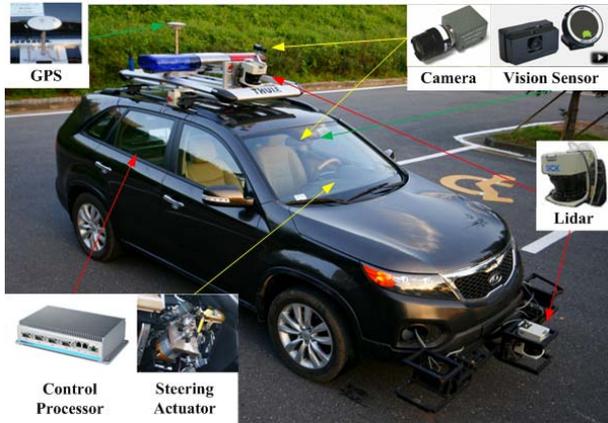


그림 9. 횡방향 제어기 성능 검증을 위한 테스트 차량.

Fig. 9. Test vehicle for lateral control.

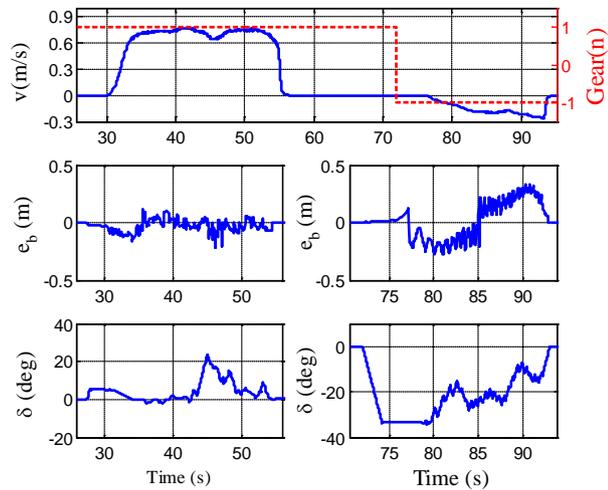


그림 10. 실차 테스트를 통한 자동 주행 및 직각 주차 시 횡방향 제어기 성능.

Fig. 10. Performance of lateral controller for perpendicular parking by test vehicle.

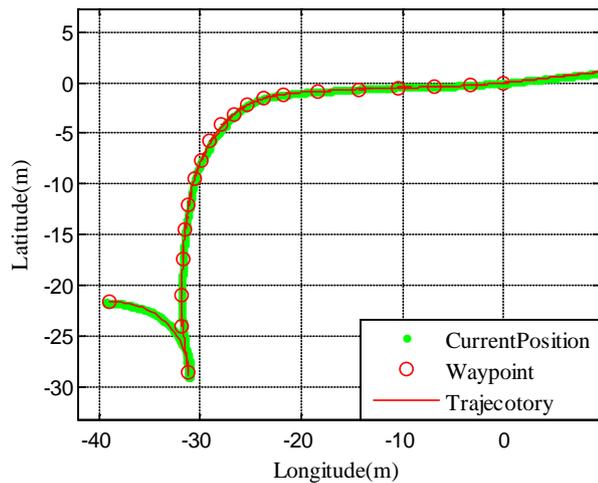


그림 11. 실차 테스트를 통한 자동 주행 및 직각 주차 시 주행 궤적 및 차량의 위치.

Fig. 11. Vehicle position of lateral controller for perpendicular parking.

할 수 있는지를 검증하였다. 참고로 장착된 DGPS의 최대 위치 오차는 0.2 m이며 조향 제어를 위해 추가적으로 모터를 장착하였다. 원래 AVP시스템에서는 주행을 위한 해당 주행 모드와 waypoint를 서버에서 차량 쪽으로 차량과 인프라간 통신을 통하여 정보를 전달하지만 본 실험에서는 횡방향 제어기의 성능을 검증하기 위한 것이어서 그림 11에서 보는 바와 같이 주행 시나리오를 정해 놓고 실험을 수행하였다.

그림 10에서는 전진 주행 및 후진 직각 주차를 포함하는 주행 시나리오에 대해서 횡방향 제어기의 성능을 각각 보여주고 있다. 전진 주행 시 횡방향 최대오차는 0.11 m이고 후진 직각 주차 주행 시 최대오차는 0.33 m이다. 또한 그림 11에서는 후진 평행 주차 시나리오에 대한 차량 위치와 주행 궤적을 비교하여 보여주고 있다.

다음으로 표 1에서 소개를 한 바와 같이 다양한 차종에 대한 제어기의 성능을 비교 분석하여 하였다. 그림 12에서 보는 바와 같이 제어기가 실차에 적용되었을 때의 실시간성을 고려하기 위하여 제어기를 단일 프로세서로 임베디드화 시키고 실시간으로 구동되는 차량 시뮬레이터와 CAN (Controller Area Network)을 통하여 정보를 주고 받으며 횡방향 제어를 검증하였다. 다시 말해서, 차량의 위치 및 요(yaw), waypoint 정보를 차량 시뮬레이터에서 제공하며 그 때 필요한 조향각을 횡방향 제어프로세서에서 계산해서 시뮬레이터에 전달하는 방식이다. 또한 종방향 제어의 경우 운전자가 실제로 스로틀 및 브레이크 페달을 조작하여 주행 속도를 실시간으로 조절가능한 상태로 제어기 성능을 검증하였다.

차량 시뮬레이터인 CarSim에서 제공하는 Hatchback, Sedan, SUV와 같이 3종류의 차량을 대상으로 (11)의 횡방향 제어기의 성능을 검증하였다. 주행 궤적이 그림 11과 같이 주어졌을 때 각 차종에 대한 횡방향 오차를 그림 13에서 확인할 수 있다. 주행시작부터 주행 완료지점까지 세 종류의 차량 모두 횡방향의 최대 오차가 0.1 m를 넘지 않았고 주차 시작 지점 22초에서 후진 직각 주차완료 될 때까지 Hatchback 타입의 차량의 경우 횡방향 최대오차는 0.24 m, Sedan타입 차량의 오차는 0.26 m, 마지막으로 Minivan 차량은 0.36 m였다. 따라서 차종의 변화에 따라 큰 성능 변화 없이 자동 발렛 파킹을 위한 조향제어가 이루어졌음을 확인할 수 있다. 또한 각각의 경우에 대한 차량의 위치를 그림 14에 중합하여 나타냈다.

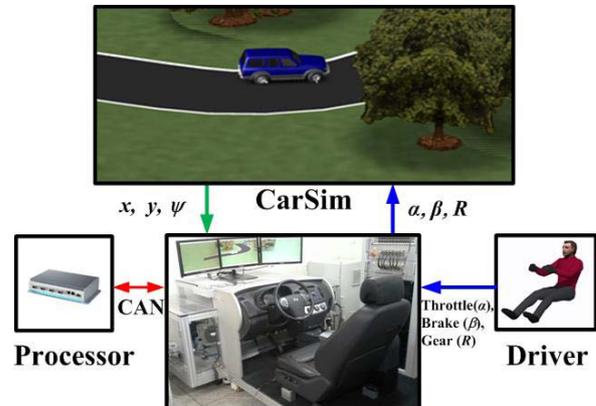


그림 12. HILS 테스트를 위한 구성도.

Fig. 12. Layout of HILS.

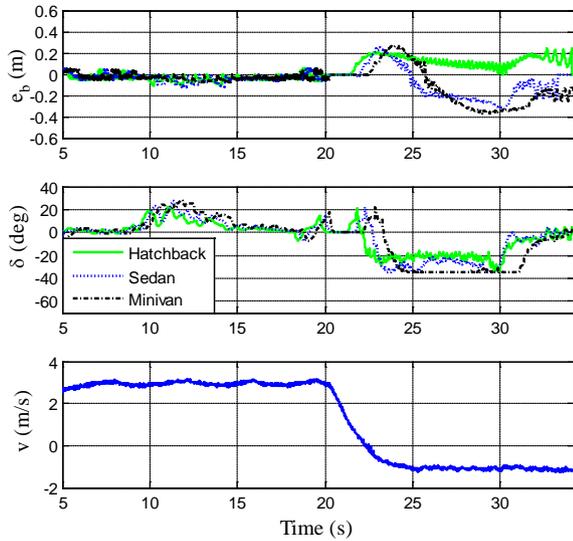


그림 13. 다양한 차종에 대한 횡방향 제어기 성능.  
Fig. 13. Performance of lateral controller with respect to various vehicle types.

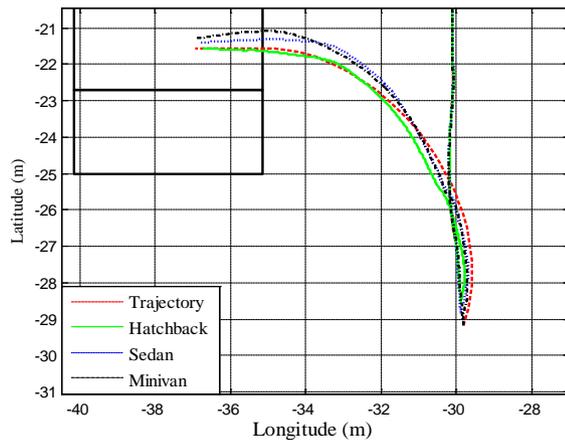


그림 14. 다양한 차종에 대한 횡방향 제어 시 주차 궤적 및 차량 위치.  
Fig. 14. Vehicle position of lateral controller for one-step perpendicular parking.

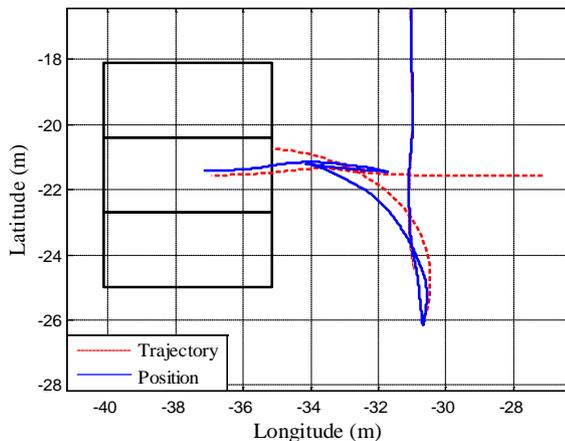


그림 15. 다단계 후진 직각 주차 시 궤적 및 차량 위치.  
Fig. 15. Vehicle position of lateral controller for multi-step perpendicular parking.

마지막으로 그림 15에서 보는 바와 같이 주차입구 선을 기준으로 4 m가 떨어진 상태에서 주차를 시작해야 하는 경우 한번에 후진주차가 어렵다고 판단되어 다단계 주차를 수행하게 된다. 그림 15는 다단계 후진 주차를 수행하기 위한 궤적을 생성하고 이를 추종하기 위한 횡방향 제어기가 적절히 작동되었음을 확인할 수 있다.

### VI. 결론

본 논문에서는 운동학적 차량 모델과 비선형 제어기를 기반으로 횡방향 제어기를 설계하고 실험적으로 검증하였다. 즉, 실차를 이용하여 검증함으로써 모델의 불확실성이나 측정 노이즈에 대해서도 강인함을 확인하였다. 또한 AVP 시스템에 적용하기 위해 필요한 요건으로 다양한 차종과 주행 모드에 변화에 대해 좀 더 정확한 차량 모델을 구하고 preview 제어를 위한 제어상수를 적절히 선택함으로써 제어기의 성능개선을 가져올 수 있음을 확인할 수 있다. 마지막으로 HILS (Hardware-In-the-Loop Simulation)을 활용한 개발 환경을 구축하여 효율적으로 제어기를 검증할 수 있음을 보여주었다. 향후에는 좀 더 다양한 주행 및 주차모드에 대하여 횡방향 제어기의 적용 가능성을 제시하고자 한다.

### 참고문헌

- [1] B. Song, "Perception technology for intelligent safety vehicle," *KSAE Auto Journal (in Korean)*, vol. 34, no. 6, pp. 35-40, 2012.
- [2] California PATH, *Intellimotion*, vol. 16, no. 1, 2010.
- [3] ITS Strategic Research Plan 2010-2014, available at <http://www.its.dot.gov/research.htm>.
- [4] D. Jung, "Intelligent vehicle development and application projects in USA," *KSAE Auto Journal (in Korean)*, vol. 31, no. 4, pp. 66-71, 2009.
- [5] Smart Highway R&D Center, a Proposal of Smart Highway Project (*in Korean*), Korea Expressway Co. 2009.
- [6] K. An, J. Choi, and D. Kwak, "Automatic valet parking system incorporating a nomadic device and parking servers," *Proc. of IEEE Int. Conf. on Consumer Electronics*, pp. 111-112, 2011.
- [7] C. Urmson, J. Anhalt, and D. Bagnell, "Autonomous driving in urban environments," *J. of Field Robotics*, vol. 25, no. 8, pp. 425-466, 2008.
- [8] A. Bacha, et al., "Odin: team victor tango's entry in the DARPA urban challenge," *Journal of Field Robotics*, vol. 25, no. 8, pp. 467-492, 2008.
- [9] Y. K. Lo, A. B. Rad, C. W. L. Wong, and M. L. Ho, "Automatic parallel parking," *Proc. of IEEE Conference on Intelligent Transportation Systems*, pp. 1190-1193, 2003.
- [10] D. Kim and B. Song, "Lateral vehicle control for semi-autonomous valet parking with consideration of actuator dynamics," *International Journal of Control Science and Engineering*, To be published.
- [11] R. Rajamani, C. Zhu, and L. Alexander, "Lateral control of a backward driven front-steering vehicle," *Control Engineering Practice*, vol. 11, no. 5, pp. 531-540, 2003.
- [12] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of Mathematics*, vol. 79, no. 3, pp. 497-516, 1957.

- [13] J. A. Reeds and L. A. Shepp, "Optimal paths for a car that goes both forwards and backwards," *Pacific Journal of Mathematics*, vol. 145, no. 2, pp. 367-393, 1990.
- [14] B. Muller, J. Deutcher, and S. Groddle, "Continuous curvature trajectory design and feedforward control for parking a car," *IEEE Transactions on Control System Technology*, vol. 15, no. 3, pp. 541-553, 2007.
- [15] S. Fleury, P. Soueres, J.-P. Laumond, and R. Chatila, "Primitives for smoothing mobile robot trajectories," *IEEE Transactions on Robotics And Automation*, vol. 11, no. 3, pp. 441-448, 1995.
- [16] R. Rajamani, *Vehicle Dynamics and Control*, Springer, 2003.
- [17] B. Song and J. K. Hedrick, *Dynamic Surface Control for Uncertain Nonlinear Systems: An LMI Approach*, Springer, 2011.
- [18] CarSim, available at <http://www.carsim.com>.



### 최 희 재

2012년 아주대학교 기계공학부 졸업.  
2002년~현재 아주대학교 대학원 기계공학과 석사과정 재학중. 관심분야는 횡방향 차량제어, 주행 궤적생성.



### 송 봉 섭

1996년 한양대학교 정밀기계공학과 졸업. 1999년 U. C. Berkeley 기계공학부 석사. 2002년 동 대학원 박사. 2004년~현재 아주대학교 기계공학과 부교수. 관심분야는 센서융합, 차량제어, 비선형제어, 고장감지.