

차량과 인프라 간 통신 기반 자동 발렛 파킹 시스템 개발 및 검증

차량 간 통신 또는 차량과 인프라 간의 통신을 기반으로 자동차를 스마트화 또는 무인화 하려는 기술들이 지속적으로 개발되고 있다. 이러한 시스템의 안전성을 보장할 수 있는 신뢰도를 검증할 수 있는 방법들이 활발하게 개발되고 있다. 특히 제어기의 강인성과 고장감지의 신뢰도와 같은 관점에서 개발되는 알고리즘의 성능을 검증하는 방법에 대해서 살펴보고자 한다.

■ 송봉섭*, 최정단

(*아주대학교 기계공학과, 한국전자통신연구원)

I. 서론

의료, 항공, 자동차, 조선 등과 같은 기존 산업과 IT 기술이 융합되는 시도들이 최근에 더욱 관심을 받고 있다. 특히 자동차의 경우 스마트 또는 커넥티드(connected) 자동차라는 이름으로 IT 기술이 접목되고 있으며 그 중에서도 차량 간(V2V) 또는 차량과 인프라(V2I) 간 통신 기술과의 융합을 통하여 운전의 안전성이나 편의성을 제고할 수 있는 기술들이 주목 받고 있다[1].

이러한 통신 기술을 차량에 적용하고자 하는 시도는 해외에서는 이미 십여 년 전부터 시도되고 있다. 대표적으로 미국의 경우 2003년에 VII (Vehicle Infrastructure Integration)라는 프로젝트를 시작하여 DSRC (Dedicated Short-Range Communications) 기반 V2V와 V2I 통신을 개발하였으며 2008년과 2009년에 이러한 통신 기술의 실증 테스트가 미시건 주와 캘리포니아 주에서 각각 수행되었다[2]. 일본과 유럽에서도 Smartway 21 프로젝트나 i2020 Intelligent Car Initiative라는 이름으로 자동차와 IT 기술을 융합하는 연구가 활발히 진행되고 있다[3].

국내의 경우 IT, 자동차, 도로 기술을 융합하여 실시간으로 정보를 공유하고 전천후, 주야간 안전주행 지원기술 구현을 목표로 2007년부터 스마트 하이웨이 사업이 시작하여 현재까지 미래의 지능형 고속도로 개발 연구를 진행하고 있다[4]. 또한 쇼핑

몰이나 백화점과 같이 주차를 위해 많은 시간이 소비되는 교통 혼잡지역에서 이러한 문제를 해결하기 위하여 기존에 주차 가능 지역으로 차량을 유도하는 차량 유도 시스템(그림 1 참조)과 이미 상용화되어 차량 주차 시 횡방향 제어를 자동으로 해 주는 주차지원시스템을 통합하고 한 걸음 더 진보시켜 V2I 통신을 기반으로 자동 발렛 파킹을 할 수 있는 스마트 주차 시스템 개발이 2010년부터 진행되고 있다[5].

또 다른 스마트 자동차의 진화 방향은 무인 자율 주행이 가능한 무인 자동차 개발이다. 이러한 무인 자동차의 실현 가능성을 보여준 대표적인 사례로 2007년 미국 DARPA에서 개최한 Urban Challenge가 있으며 도심 교통 환경에서 주행 및 주차까지 가능함을 선보였다[6,7]. 더 나아가 2010년에는 미국 Google사에서 그림 2에서 보이는 차량을 사용하여 1,600 Km 이상을 무인으로 주행하였고 이후 225,000 Km 이상을 무인으로 주행했다고 보고하고 있다. 또한 2012년 네바다 주에서는 미국에서 처음으로 무인 주행이 가능하도록 차량 등록을 허가하였다.

이렇게 IT 기술이 접목된 차세대 지능형 자동차 또는 시스템이 다양한 형태로 활발하게 개발되고 있으나 주행 시 안전성을 확보하는 문제는 그 무엇보다 중요한 필수조건이다. 본 논문에서는 인지, 판단 및 계획, 제어와 같이 여러 관점에서 필요한 기술들 중 제어기 설계 및 검증 문제로 집중해서 살펴보고

자 한다. 특히 센서 노이즈, 통신 지연, 외란이 존재하는 실제 환경에서 제어기 성능을 보장하기 위해서 제어기의 강인성을 어떻게 평가하고 센서나 통신 시스템의 고장 발생 시 이를 감지해내는 고장 감지를 위한 개발 및 검증 환경에 대해서 논의하고자 한다.

II. 자동 발렛 파킹 시스템

1. 시스템 구성

현재 개발되고 있는 무인 자율 주행 자동차는 다양한 주변 환경을 인지해야 하기에 추가적으로 많은 고성능 센서들이 장착되어야 하며 이러한 센서들이 장착되어 있더라도 일반 주차장 환경에서 최적의 주차공간을 탐색하는 것은 쉽지 않다. 반면에 최적의 주차공간을 탐색하여 알려주는 차량 주차 유도 시스템은 이미 상용화가 되었고 좀 더 자세한 주행 경로를 핸드폰과 같은 단말기를 통하여 운전자에게 제공하려는 기술이 지속적으로 개발되고 있다. 이러한 무인 자율 주행 자동차의 한계점을 보완하기 위한 방법 중 하나로 제한된 영역에서 자율주행과 자



그림 1. 차량 유도 시스템.



그림 2. 미국 Google사의 무인자동차.

동주차를 지원하는 V2I 통신 기반 자동 발렛 파킹(Automatic Valet Parking, AVP) 시스템 또는 스마트 주차 시스템 개발이 연구되고 있다.

그림 3에서 보는 바와 같이 AVP 시스템은 하드웨어적으로 크게 네 부분으로 나누어 생각해 볼 수 있다. 운전자에게 자동 발렛 파킹에 대한 정보를 실시간으로 전달할 수 있는 Nomadic device, AVP를 하려는 차량을 등록하고 최적의 주차공간을 탐색하여 해당 경로를 생성해주는 Global Situation-Aware Server (또는 AVP server), 분산되어 있는 인프라 기반 센서로부터의 측정값을 이용하여 차량의 위치나 heading 각을 측정하고 예측하는 Local Spatial-Aware Server (또는 perception server), 마지막으로 중/횡방향으로 차량을 제어하기 위한 차량 제어기(vehicle controller)로 구성이 되어 있다. 좀 더 구체적인 주요 기능들이 그림 4에 정리되어 있다.

2. 주행 시나리오

AVP를 위한 주행 시나리오는 그림 3에서 보는 바와 같이 A지점부터 B지점까지 자율주행과 B지점부터 E지점까지 자동주차로 나눌 수 있다. Waypoint라 불리는 주행 경로는 그림 3에서의 AVP 서버에서 계획되어 무선 통신을 통하여 차량에 전달된다. A지점에서 B지점 사이 자율주행 구간에서 곡선구간을 포함하

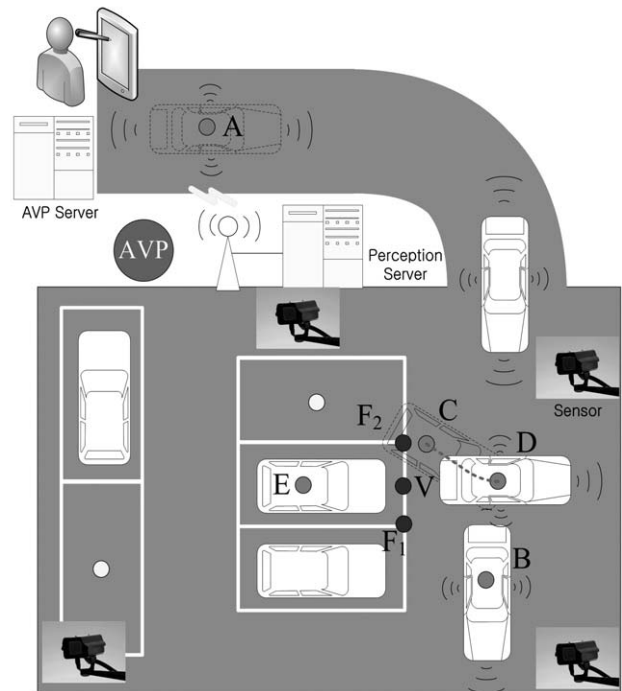


그림 3. 자동 발렛 파킹 시스템의 개념도.

고 있는 경우 조밀한 waypoint를 생성하여 곡선로 주행을 위한 주행 궤적을 생성할 수 있도록 해야 한다. B지점에서 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 3에서 보는 바와 같이 B점에서 C점으로 후진 주차 후 다시 D점으로 전진 후 E점으로 후진 주차를 하여 주차를 완료하게 된다.

3. 적용 차량의 하드웨어 구성

그림 3와 4에 명시된 차량 제어기는 실제 차량에 장착이 되어 있으며 중·횡 방향 차량 제어를 위한 센서 및 액추에이터를 포함하고 있다. 좀 더 구체적으로 그림 5에서 보는 바와 같이 전기 자동차를 기반으로 프로토타입(prototype)이 개발되어 사용되고 있다. 개발의 효율성을 위해서 다중 프로세서를 사용하며 CAN (Controller Area Network) 통신을 사용한 분산 시스템으로 개발하고 있다. 또한 인프라와 차량간 통신은 IEEE 802.11a 기반 무선 통신을 사용하여 정보를 주고 받는다. 추가적으로 다양한 차종의 차량이 자동 발렛 파킹 시스템을 사용하여 주차될 수 있음을 보여주기 위해서 Hatchback이나 SUV 형태의 자동차에 대한 프로토타입도 동시에 개발을 진행하고 있다.

Ⅲ. 제어기 설계

1. 제어기 설계를 위한 고려사항

기존의 횡방향 차량 제어는 단일 차종에 대하여 주로 한번에 주차하기 위한 경로를 생성하는 차량중심의 주차지원시스템을 위한 횡방향 제어 연구가 주로 이루어진 반면에[8-9], 인프라와의 연동을 통한 자동 발렛 파킹을 수행하기 위해서는 다음과 같은 문제를 고려하여 횡방향 제어를 설계해야 한다.

- i. 다양한 차종에 대한 자동 발렛 파킹이 가능해야 한다.
- ii. AVP를 위해 다양한 형태의 주차가 가능해야 한다. 예를 들

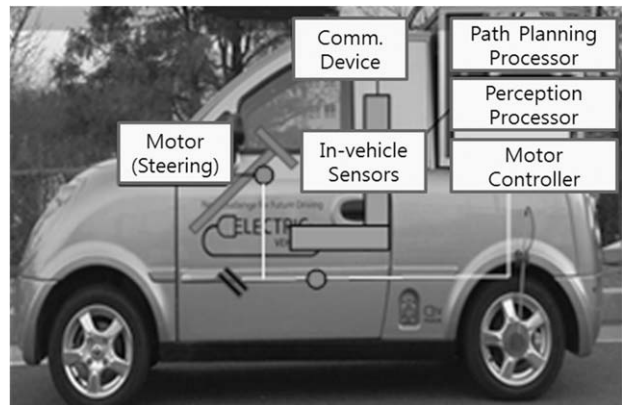


그림 5. 차량 제어를 포함하고 있는 개발 플랫폼.

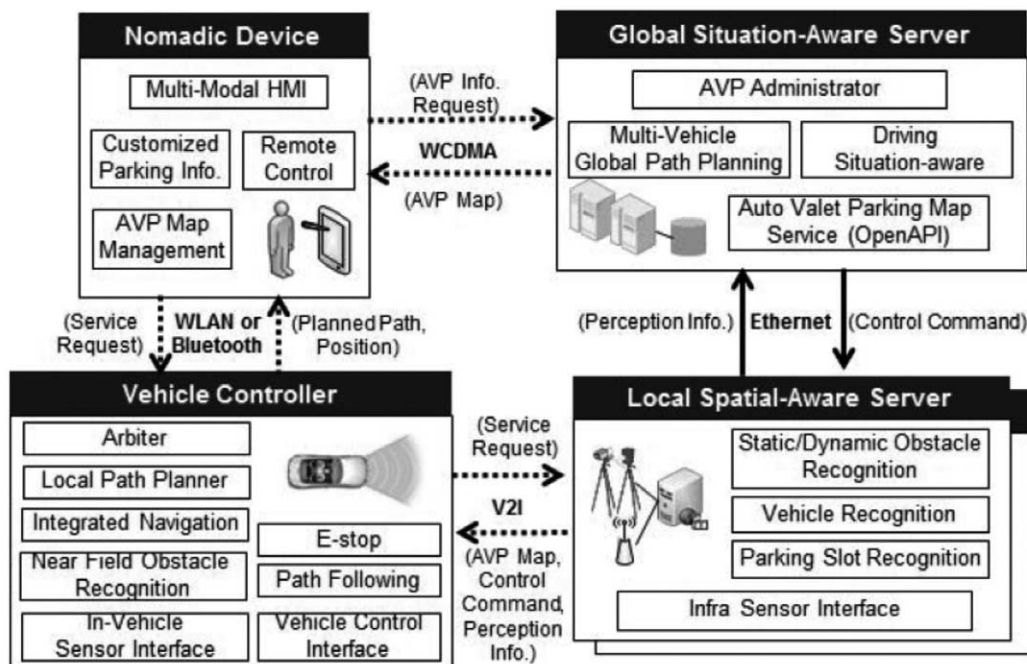


그림 4. 자동 발렛 파킹 시스템의 구조 (5).

어서, 주차 형태로 전방 직각 주차, 후방 직각 주차, 후방 평행 주차, 사선 주차 등이 있다.

iii. 위치 오차 또는 통신 지연에 강인해야 하며 이로 인한 제어기의 성능저하가 적어야 한다.

iv. 센서 및 액추에이터의 고장, 통신 두절과 같은 상황을 감지할 수 있는 고장 감지 알고리즘이 포함되어야 한다.

v. 일반적으로 한 번에 주차 가능한 지점으로 이동하여 주차를 시작하지만 통신지연이나 센서 노이즈 등에 의해서 이러한 지점에 위치하지 못했을 경우 다단계(multi-step) 주차를 지원해야 한다. 또한 주행 중 경로상에 정지되어 있는 자동차가 있는 경우 회피해서 주행할 수 있어야 한다.

vi. 실차를 이용하여 제안되는 알고리즘을 검증하기에는 많은 비용과 노력이 예상되며 위에서 언급된 모든 고려사항을 반영하여 실차에 적용되기 전에 제어기를 검증할 수 있는 검증 플랫폼이 필요하다.

2. 횡방향 제어기 설계

모델을 기반으로 제어기를 설계하는 방법이 주로 사용되고 있으며 일반적으로 고속 주행의 경우 bicycle 모델을 기반으로, 저속 주행 시 기구학적(kinematic) 모델을 사용하여 횡방향 제어가 설계되고 있다[9]. 하지만 위에서 언급된 첫 번째 고려사항을 생각하면서 횡방향 제어기를 개발하기 위해서는 우선적으로 다양한 차량의 거동을 모델링 하기 위한 차량 모델 개발이 필요하다. 예를 들어서, 차량이 10 Km/s 내외로 저속 주행을 하며 이에 따라 횡방향 미끄럼(slip angle)이 거의 없다고 가정한다면 다음과 같은 기구학적 차량 모델을 사용할 수 있다.

$$\begin{aligned} \dot{x} &= v \cos \psi \\ \dot{y} &= v \sin \psi \\ \dot{\psi} &= \frac{v}{L_j} \tan \delta \end{aligned} \quad (1)$$

여기서 x 와 y 는 차량의 위치, ψ 와 $\dot{\psi}$ 는 차량의 heading 각과 요레이트(yaw rate), δ 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 나타낸다.

제어기 설계를 위해서 차량 모델이 식 (1)과 같이 주어진다 면 로봇제어, 특히 모바일 로봇 제어를 위한 제안된 많은 제어기 기법이 적용될 수 있다. 예를 들어서 비선형 제어 기법을 사용하여 속도 변화 및 차종 변화에 따라 제어기가 쉽게 재설

계되고 실차 주행에서도 강인한 횡방향 제어기가 소개되고 있다[10].

다음으로 횡방향 미끄럼(β)을 무시할 수 없지만 각 바퀴의 속도방향이 바퀴의 각도와 일치한다고 가정한다면 다음과 같이 차량 모델을 정의할 수 있다[9].

$$\begin{aligned} \dot{x} &= v \cos(\psi + \beta) \\ \dot{y} &= v \sin(\psi + \beta) \\ \dot{\psi} &= \frac{v}{L_j} \cos \beta \tan \delta \end{aligned} \quad (2)$$

더 나아가 조향 액추에이터의 동역학적 특성을 고려한다면, 즉 차량의 조향 각이 기구학적으로 제한되어 있고 조향 액추에이터의 특성에 따라 조향 각의 회전 속도가 제한되어 있다면 다음과 같이 표현될 수 있다.

$$|\delta| \leq \delta_{\max}, |\dot{\delta}| \leq \omega_{\max} \quad (3)$$

여기서 δ_{\max} 는 최대 조향 각이며 ω_{\max} 는 최대 조향 회전 속도를 뜻한다.

본 장에서는 특정 제어기 설계 기법에 대한 설명을 하기 보다는 상대적으로 간단한 기구학적 모델을 사용하는 경우도 식(1)에 식(2)로, 그리고 식(3)을 동시에 고려하면서 수학적으로 좀 더 복잡한 제어기가 설계될 수 있다는 점을 보여주고자 한다. 또한 bicycle 모델을 기반으로 횡방향 제어기를 설계하는 방법들도 제안되고 있다.

3. 강인성 및 고장감지

노면의 상태가 정상적인 상태 (예를 들어서 건조한 아스팔트)나 차량 주행 속도가 일정 수준 이상이 되면 식(2)에서 주어진 가정을 더 이상 사용할 수 없는 경우가 발생할 수 있다. 이럴 경우 식(2)의 β 와 δ 에 변수 불확실성을 추가하여 $\beta + \Delta\beta$ 와 $\delta + \Delta\delta$ 로 각각 표현할 수 있다. 또한 식(1)이나 식(2)를 기반으로 제어기를 설계하기 위해서는 차량의 위치와 heading 각을 측정해야 하며 센서 노이즈를 포함하고 있고 통신을 통하여 정보를 전달하는 경우 다음과 같이 표현할 수 있다.

$$y_m(t - \Delta t) = [x(t) \ y(t) \ \psi(t)]^T + [\Delta x \ \Delta y \ \Delta \psi]^T \quad (4)$$

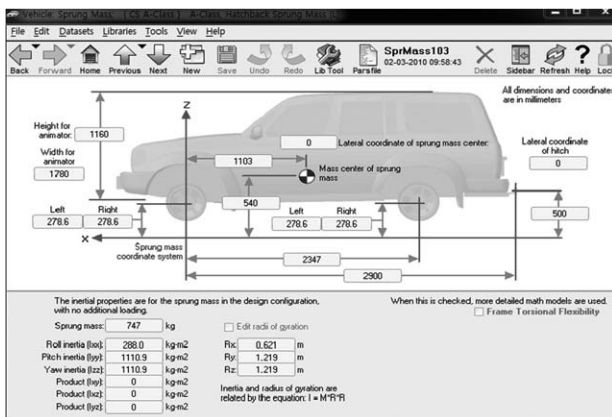
여기서 Δt 는 시간지연, $[\Delta x \ \Delta y \ \Delta \psi]^T$ 는 센서 노이즈를 뜻한다.

마지막으로 네 번째 고려사항을 생각하는 경우 식(4)에 additive 또는 multiplicative fault 형태로 센서의 고장 및 통신 두절 상황을 수학적으로 모델링을 해서 고장 증상을 모사할 수 있다 [11-12]. 이러한 수학적 모델링을 통하여 고장 감지 알고리즘을 다양한 기법을 사용하여 개발할 수 있다 [11-13]. 개별 차량을 중심으로 주행이나 주차를 위한 횡방향 제어를 설계하는 연구가 많이 진행되어 있는 반면 AVP 시스템에서 요구하는 다양한 차종에 대해 여러 주행모드를 실행할 수 있는 횡방향 제어기 설계에 대한 연구가 진행되고 있으며 강인성과 고장 감지를 고려하여 설계하는 연구가 최근에 시작되고 있다.

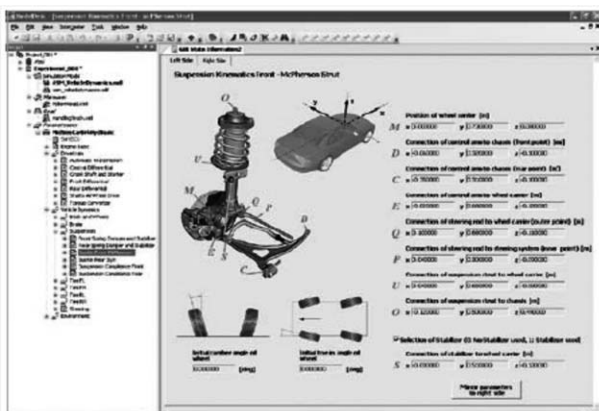
IV. 알고리즘 검증 플랫폼

1. 차량 모델 기반 제어기 검증

개발된 제어기를 검증하기 위해서는 우선 크게 세 가지의 구성요소를 가지고 있는 시뮬레이션 환경이 필요하다. 첫 번째는



(a) CarSim의 차량 모델링 파라미터 구성 예 (14)



(b) dSPACE사의 McPherson Suspension 모델링 (15)

그림 6. 차량 시뮬레이션을 위한 차량 모델.

경차, 세단, SUV와 같은 다양한 차량을 모사할 수 있는 차량 모델과 좀 더 구체적으로 차량 내부 구성 요소, 예를 들어서 서스펜션이나 타이어 모델을 교체할 수 있는 환경이 필요하다. 두 번째로는 직선 및 곡선로, 또는 일반도로와 주차장과 같은 다양한 도로환경을 구성하고 다중 차량을 시뮬레이션 할 수 있는 환경이 필요하다. 마지막으로는 센서 및 액추에이터의 측정오차나 고장을 인위적으로 만들 수 있는 모델이 필요하다.

위에서 언급된 두 가지의 경우 자체적으로 시뮬레이션 환경을 만들 수도 있으나 일반적으로 상용화된 차량 시뮬레이터를 많이 사용한다. 대표적으로 그림 6에서 보는 바와 같이 Mechanical Simulation Corporation의 CarSim이나 dSPACE사의 Automotive Simulation Models (ASM)을 들 수 있다 [14, 15]. 그림 5에서 보는 바와 같이 다양한 도로 환경을 구성할 수 있고 차량의 내부 모델을 파라미터화하여 다양한 차량 모델을 구성할 수 있다.

다음으로 예를 들어서 자동 발렛 파킹을 위한 횡방향 제어를 하기 위한 알고리즘을 개발하는 경우 이러한 차량 시뮬레이터가 제어기를 설계하는 과정과 검증하는 과정에 사용될 수 있다. 우선 모델기반 제어기법이 사용되었다고 가정한다면 제어기 설계를 위한 차량 모델링과 이를 검증하는 단계가 선행되어야 한다. 좀 더 구체적으로 CarSim이라는 상용 차량 시뮬레이터의 경우 해치백, 세단, SUV 등 다양한 차량 모델을 제공하며 예를 들어서 식 (1)의 모델이 제어기를 설계하기 위한 모델로 사용되었다면 주어진 속도 (v)와 조향 각 (δ)에 대해서 CarSim에서 출력되는 값과 식 (1)의 출력 값을 비교하여 제어기 설계를 위한 모델의 타당성을 확인할 수 있다. 예를 들어서 그림 7에서 보는

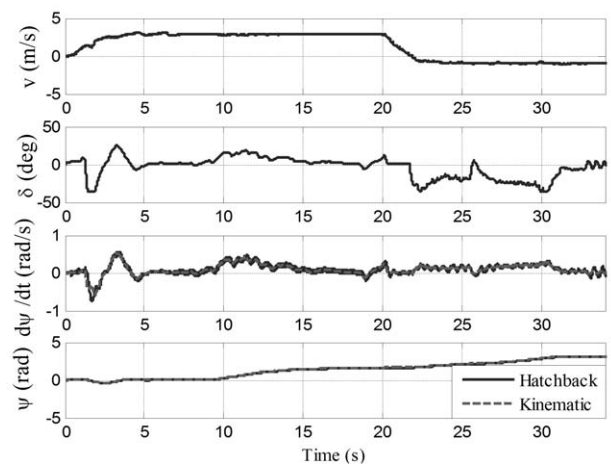


그림 7. CarSim의 해치백형 차량 모델과 운동학적 차량 모델의 주행 및 주차 시 요레이트와 heading 각 비교 (10).

바와 같이 주어진 속도와 조향각에 대해서 차량 시뮬레이터에서 출력되는 값과 (1)의 출력값을 요레이트와 요각을 기준으로 비교하였다. 차종을 바꾸어 가면서 요레이트($\dot{\psi}$)와 heading 각(ψ)

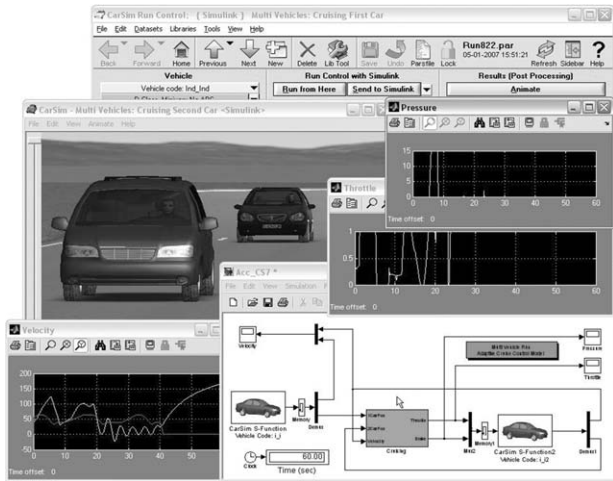
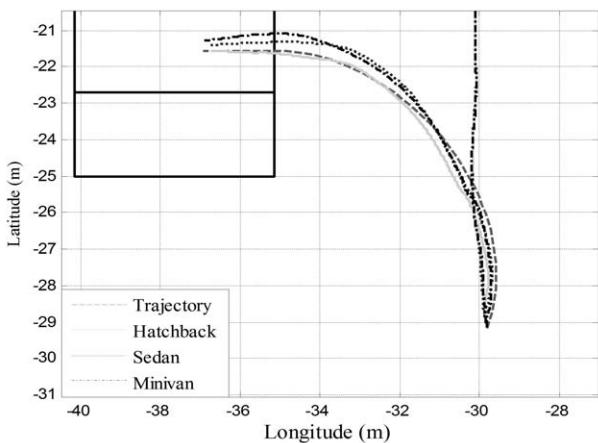


그림 8. 차량 시뮬레이터 기반 제어기 검증 (15).



(a) 직각 주차 시 주차 궤적 및 차량 위치 (9)



(b) 제어기 성능 비교 애니메이션

그림 9. 다양한 차종에 대한 횡방향 제어기 성능 분석.

을 기준으로 제어기 설계를 위한 차량모델에 대한 적합성을 확인할 수 있다.

검증된 차량 모델을 기반으로 횡방향 제어기가 설계되었다면 오프라인 시뮬레이션 환경에서 제어기의 성능을 검증할 수 있다. 제어기가 Matlab/Simulink나 Labview로 프로그래밍이 되었을 경우 그림 8과 같이 다중 차량을 생성시킬 수 있는 상용 시뮬레이터와 연동하여 제어기의 성능을 검증할 수 있다. 더 나아가 제어기의 강인성을 확인하기 위해서 위치나 heading 각에 대한 측정 값에 노이즈를 추가하거나 다양한 주행 및 주차 시나리오(전방 및 후방 직각, 평행, 사선 주차)를 대상으로 제어기의 성능을 검증할 수 있다.

예를 들어서, 차량 시뮬레이터인 CarSim에서 제공하는 Hatchback, Sedan, Minivan와 같이 3종류의 차량을 대상으로 횡방향 제어기의 성능을 검증하였다. 주행 궤적이 그림 9(a)과 같이 주어졌을 때 각 차종에 대한 횡방향 오차를 그림 9(a)에서 확인할 수 있다. 주행시작부터 주행 완료지점까지 세 종류의 차량 모두 횡방향의 최대 오차가 0.1m를 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 m 였다. 더 나아가 그림 9(b)와 같이 동시에 각기 다른 차종에 대한 제어기 성능을 비교할 수도 있다.

2. HILS 기반 고장 검지 검증

오프라인 시뮬레이션을 통해 검증된 제어기는 다음으로 실차에 적용하기 전에 실시간으로 잘 작동될 수 있는지 또는 실시간으로 발생하는 고장을 검지해 낼 수 있는지를 확인해야 한다. 이를 위해서는 우선적으로 차량 모델을 실시간으로 구동시킬 수가 있어야 한다. 그림 10에서 보는 바와 같이 일반적으로 상용 시뮬레이터의 경우 실시간 OS를 기반으로 실시간으로 구동할 수 있는 환경을 제공한다.

자동차의 자동화 정도에 따라 다양한 Hardware-in-the-loop simulation (HILS) 환경 구성이 가능하다. 예를 들어서 그림 11과 같이 종방향은 운전자가 수동으로 운전하고 횡방향은 자동으로 운전자를 지원하는 반자동 발렛 파킹을 위한 횡방향 제어기의 성능 검증을 위한 HILS 환경을 구축하고자 한다. 이를 위해서는 실시간 OS 중 하나인 QNX를 기반으로 CarSim을 실시간으로 구동시키는 차량 시뮬레이터(vehicle simulator), 차량의 스톱틀이나 브레이크 패달, 또는 기어박스의 신호를 읽고 차량 시뮬레이터와 연동할 수 있는 하드웨어 및 인터페이스 프로세서가

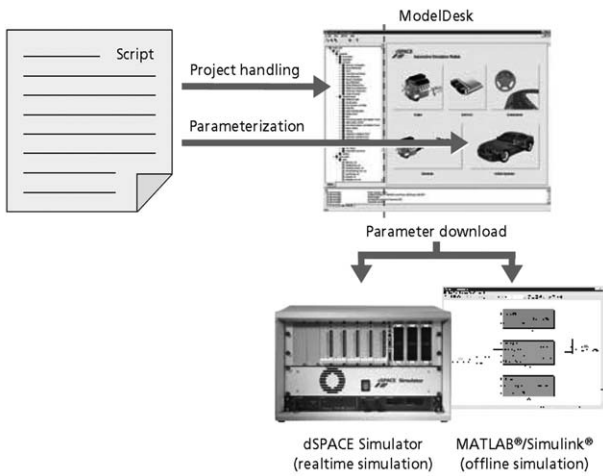


그림 10. dSPACE사의 HILS구성을 위한 실시간 차량 시뮬레이션 환경 (16).

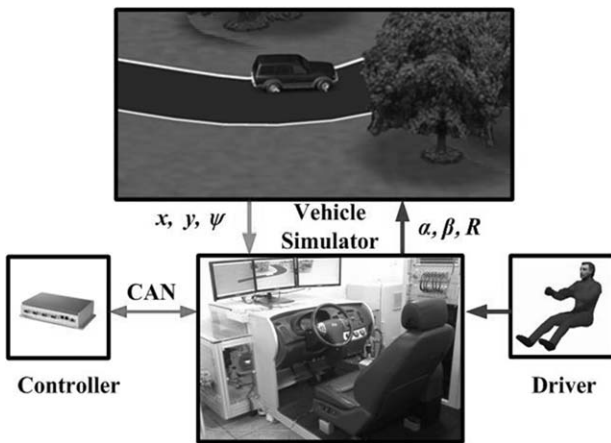


그림 11. HILS 기반 횡방향 제어기 검증을 위한 구성도.

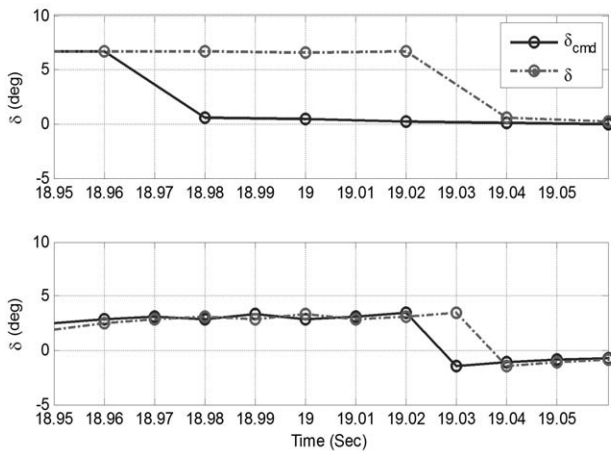


그림 12. HILS를 통한 통신 시간지연 (위: HILS, 아래: 오프라인).

필요하다[16]. 또한 설계된 제어기는 단일 프로세서로 임베디드화되고 CAN (Controller Area Network) 통신을 통하여 정보를 주고 받으며 제어기의 실시간 구동 성능을 검증할 수 있다.

좀 더 구체적으로 횡방향 제어기는 차량의 위치 및 heading 각, waypoint 정보를 차량 시뮬레이터로부터 제공받고 그때 필요한 조향 각을 계산해서 다시 차량 시뮬레이터에 CAN 통신을 통하여 전달하는 방식이다. 또한 종방향 제어의 경우 운전자가 실제로 스로틀 및 브레이크 페달을 조작하여 주행 속도를 실시간으로 조절 가능한 상태로 제어기 성능을 검증하였다. 또한 통신지연이 존재하는 상황에서 실시간으로 다양한 차량에 적용될 수 있음을 HILS 기반으로 검증할 수 있다. 예를 들어서, 그림 12에서 보는 바와 같이 오프라인 시뮬레이션보다 CAN통신으로 정보를 주고 받음으로써 추가적으로 50 (ms)의 시간 지연이 발생하게 할 수 있음을 확인 할 수 있다.

중·횡 방향을 모두 자동화하는 경우 그림 13과 같이 구성할 수 있으며 추가적으로 고장 생성기(fault emulator)를 사용하여 무선 통신 시 발생할 수 있는 packet loss나 disconnect 현상 재현하거나 위치추정 시 발생할 수 있는 고장 현상을 모사하여 실시간으로 고장 생성 시나리오를 구현할 수 있다. 따라서, 이러한 환경을 사용하여 고장감지 알고리즘의 실시간 성능을 확인할 수 있다.

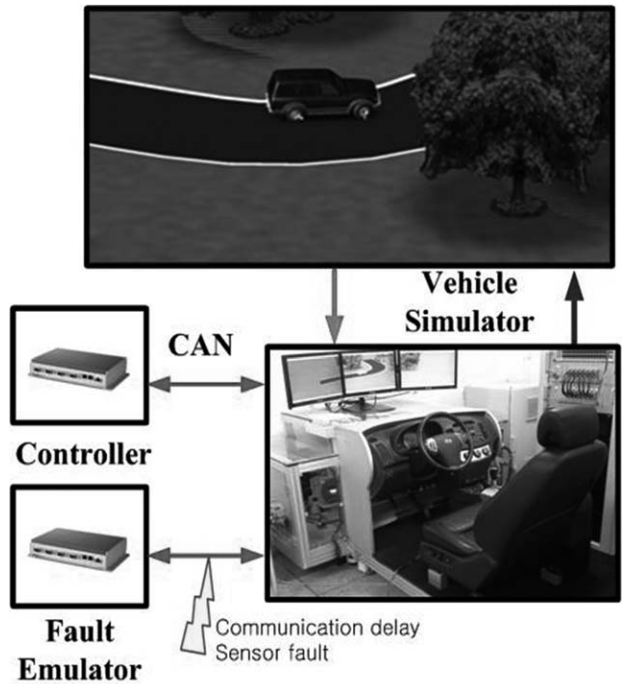


그림 13. 고장 감지를 위한 HILS 테스트 구성도.

3. 실차 기반 검증

마지막으로 HILS를 통하여 검증된 알고리즘을 그림 14와 같이 실차를 이용하여 검증하는 단계를 수행되어야 한다. 우선 횡방향 제어기 알고리즘을 평가하기 위해 그림 14와 같이 자율 주행 테스트 차량을 사용하여 설계된 제어기가 전진 주행과 후진 주차 시 적절한 조향을 자동으로 할 수 있는지를 검증할 수 있다. 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 15에서 보는 바와 같이 주행 시나리오를 정해 놓고 실

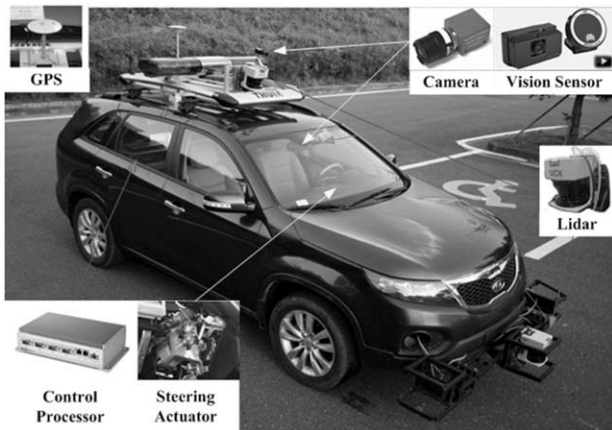


그림 14. 제어기 성능 검증을 위한 테스트 차량.

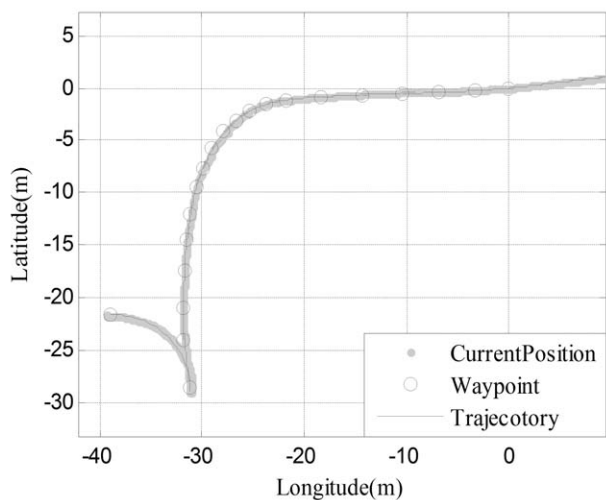


그림 15. 실차 테스트를 통한 자동 주행 및 직각 주차 시 주행 궤적 및 차량의 위치 (10).

험을 수행하였다. 전진 주행 및 후진 직각 주차를 포함하는 주행 시나리오에 대해서 횡방향 제어기의 성능을 각각 보여주고 있다.

마지막으로 운전자에게 자동 발렛 파킹에 대한 정보를 실시간으로 전달할 수 있는 모바일 단말기(nomadic device)와 차량과의 연계를 확인하는 단계를 그림 16에서 보는 바와 같이 진행하였다. 그림 16(a)에서 보는 바와 같이 단말기를 통하여 자동 발렛 파킹을 명령하고 주차가 완료된 후 단말기를 통하여 주차가 완료된 영상을 송출하고 있음을 보여준다.

Ⅶ. 결론

통신 기반 자동 발렛 파킹 시스템 개발을 위해서는 차종 및 주행 시나리오의 다양성, 센서 노이즈 및 통신 지연, 그리고 고장 발생을 종합적으로 고려하여 개발하여야 한다. 이렇게 다양한 환경 변화에 대한 제어기의 강인성과 고장 검지 알고리즘의 신뢰성을 검증하기 위해서는 개발 및 검증 플랫폼 구축이 필수적이다. 차량 시뮬레이터, HILS (Hardware-In-the-Loop Simulator), 프



(a) 모바일 단말기와 연계된 자동 발렛 파킹 요청



(b) 자동 발렛 파킹 완료 후 모바일 단말기를 통한 확인

그림 16. 모바일 단말기와 자동 발렛 파킹 시스템의 연계.

로타입 실차를 활용한 순차적 개발 환경을 구축하여 효율적으로 제어기를 검증할 수 있음을 보여주었다.

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] D. Jung, "Intelligent vehicle development and application projects in USA," *KSAE Auto Journal (in Korean)*, vol. 31, no. 4, pp. 66-71, 2009.

[4] Smart Highway R&D Center, a Proposal of Smart Highway Project (*in Korean*), Korea Expressway Co. 2009.

[5] 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.

[6] C. Urmson, J. Anhalt, and D. Bagnell, "Autonomous driving in urban environments," *J. of Field Robotics*, vol. 25, no. 8, pp. 425-466, 2008.

[7] 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.

[8] 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.

[9] R. Rajamani, *Vehicle Dynamics and Control*, Springer, 2003.

[10] H. Choi and B. Song, "Design of Lateral controller for automatic valet parking and its performance analysis with respect to vehicle types," *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 18, no. 11, pp. 1051-1058, 2012.

[11] B. Song and J. K. Hedrick, *Dynamic Surface Control for Uncertain Nonlinear Systems: An LMI Approach*, Springer, 2011.

[12] B. Song and J. K. Hedrick, "Fault tolerant nonlinear control with applications to an automated transit bus," *Vehicle System Dynamics*, vol. 43, no. 5, pp. 331-350, 2005.

[13] S. Kim, B. Song and H. Song, "Integrated fault detection and diagnosis system for longitudinal control of an autonomous all-terrain vehicle," *International Journal of Automotive Technology*, vol. 10, no. 4, pp. 505-512, 2009.

[14] CarSim, available at <http://www.carsim.com>.

[15] ASM Vehicle Dynamics Simulation Package, available at http://www.dspaceinc.com/en/inc/home/products/sw/automotive_simulation_models.cfm.

[16] QNX, available at <http://www.qnx.com>.

저자약력



송봉선

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



최정단

- 1993년 중앙대학교 컴퓨터공학과 졸업.
- 1995년 중앙대학교 컴퓨터공학과 석사.
- 2005년 충남대학교 컴퓨터공학 박사.
- 1995년~현재 한국전자통신연구원 책임연구원.
- 관심분야 : 컴퓨터 그래픽스, 영상처리, 자동차-IT, GIS