

## 무인 자율 주행을 위한 최단 시간 경로계획 알고리즘 설계<sup>§</sup>

김동욱\* · 김학구\* · 이경수\*<sup>†</sup>

\* 서울대학교 기계항공공학부

### Design of Near-Minimum Time Path Planning Algorithm for Autonomous Driving

Dongwook Kim\*, Hakgu Kim\* and Kyongsu Yi\*<sup>†</sup>

\* School of Mechanical and Aerospace Engineering, Seoul Nat'l Univ.

(Received May 3, 2012 ; Revised October 23, 2012 ; Accepted February 4, 2013)

**Key Words:** Near-Minimum Time Path Planning(최소 시간 경로계획), Path Tracking(경로 추종), Autonomous Driving(무인 자율 주행), Path Optimization(경로 최적화), Optimal Preview Control(최적 예측 제어), Model Free Control(모델 독립식 제어)

**초록:** 본 논문은 무인 자율 주행을 위한 최소 시간 경로계획 알고리즘에 대해서 제안하였다. 최소 시간 경로계획 문제는 경로의 기하학적인 형상에 대한 고려뿐만이 아니라 차량 동역학까지 고려해야 하는 최적 문제이다. 경로계획은 후보 경로 생성 알고리즘과 속도 최적화 알고리즘으로 구성된다. 후보 경로 생성 알고리즘은 최단 거리 경로와 최고 속도 경로를 조합하여 후보경로를 생성한다. 속도 최적화 알고리즘은 차량의 주행성능 한계와 타이어 마찰 한계를 고려하여 각 후보 경로의 최고 속도를 계산한다. 이렇게 계산된 경로와 속도를 이용하여 각 후보 경로의 주행 시간을 계산하고 가장 작은 주행 시간의 경로를 최단시간 경로로 도출한다. 그리고 제안한 알고리즘은 CarSim 과 Matlab/Simulink 를 사용한 시뮬레이션을 통해 검증하였다.

**Abstract:** This paper presents a near-minimum time path planning algorithm for autonomous driving. The problem of near-minimum time path planning is an optimization problem in which it is necessary to take into account not only the geometry of the circuit but also the dynamics of the vehicle. The path planning algorithm consists of a candidate path generation and a velocity optimization algorithm. The candidate path generation algorithm calculates the compromises between the shortest path and the path that allows the highest speeds to be achieved. The velocity optimization algorithm calculates the lap time of each candidate considering the vehicle driving performance and tire friction limit. By using the calculated path and velocity of each candidate, we calculate the lap times and search for a near-minimum time path. The proposed algorithm was evaluated via computer simulation using CarSim and Matlab/Simulink.

- 기호설명 -

$x_i, y_i$	: 이산화한 경로 좌표	$\Gamma$	: 경로의 곡률
$x_l, y_l, x_r, y_r$	: 이산화한 경로 좌 우 경계 좌표	$\mu$	: 마찰 계수
$w_i$	: 전체 경로 폭에 대한 우측 경계로부터 경로까지의 거리 비율	$g$	: 중력 가속도
$L$	: 도록 폭	$\psi$	: 요각
$S$	: 경로의 길이	$\psi_d$	: 목표 요각
		$y_r$	: 거리 오차
		$e_\psi$	: 각도 오차
		$C_f, C_r$	: 코너링 강성
		$l_f, l_r$	: 차량의 질량 중심으로부터 전후륜 축까지의 거리
		$m$	: 차량 질량

<sup>§</sup> 이 논문은 2012 년도 대한기계학회 동역학 및 제어부분 춘계학술대회(2012. 4. 11.-13., 제주 KAL 호텔) 발표논문임.

<sup>†</sup> Corresponding Author, [kyi@snu.ac.kr](mailto:kyi@snu.ac.kr)

$I_z$	: 요 모멘트
$v_x$	: 종 방향 차량 속도
$a_x, a_y$	: 종 횡방향 가속도
$\alpha$	: 스로틀 앵글
$P_b$	: 브레이크 마스터 실린더 압력
$g_0$	: 자연 감가속도
$F_{th}, F_b$	: 스로틀과 브레이크에 대한 적응변수

## 1. 서 론

날로 늘어나는 차량으로 인해 도로 주행 환경은 점점 더 복잡해 지고 있다. 이로 말미암아 운전자의 실수로 인해 발생하는 사고 또한 증가하고 있다. 한 연구에 따르면 사고 원인 중 90%가 운전자의 판단 착오 또는 조작 실수에 의해서 일어난다고 한다. 이러한 사고를 방지하기 위한 하나의 방법으로 주목 받고 있는 것이 무인 자율 주행 기술이다. 무인 자율 주행 기술은 실수가 많은 운전자를 대신하여 차량을 제어하는 기술로 운전자에게 편의를 제공해 줄 뿐만 아니라 사고 예방과 에너지 절감 면에서도 많은 이점을 가지고 있다. 이런 이유로 여러 정부와 기업들에서는 무인 차량 대회 개최 등을 통해 무인 차량 기술 발전을 촉진시키고 있다.<sup>(1,2)</sup>

무인 자율 주행 차량이 사고를 방지하기 위해서는 일반적인 주행상황에서 주변 차량과 안전거리를 유지하고 위험 차량을 감지하는 기술도 필요하겠지만 사고 위험이 높은 주행상황에서도 숙련된 운전자와 같이 차량의 한계 성능까지 활용하여 사고 상황을 회피 할 수 있어야 한다. 숙련된 운전자는 제어력을 상실하지 않고 차량의 한계성능까지 제어하기 위해 필요한 세가지 기술을 가지고 있다.<sup>(3)</sup> 첫 째 그들은 도로와 타이어 사이의 마찰 한계를 인지 할 수 있다. 둘 째 그들은 모든 작동 장치를 정확히 제어하여 차량을 원하는 방향으로 제어 할 수 있다. 그리고 마지막으로 그들은 많은 주행 경험으로부터 주어진 경로를 최단시간으로 탈출하기 위한 최적의 경로와 속도를 알고 있다. 마찰계수를 측정하기 위한 방법은 이미 여러 연구에서 개발되었다.<sup>(4,5)</sup> 그리고 전동식 파워 스티어링(MDPS)과 전자식 차량 안정성 제어 장치(ESC)등이 개발되어 차량의 정밀한 제어가 가능해 졌다. 하지만 최적 경로와 속도는 아직 연구가 많이 진행되지 않았다. 그래서 본 연구에서는 주어진 도로를 최단시간으로 주행 할 수 있는 최단 시간 경로와 속도 설계에 초점을 맞추었다.

기존의 최단 시간 경로 계획에 대한 연구들은 차량 성능에 대한 고려가 부족하였다. 기존 연구에서는 경로와 속도를 분리하여 계획<sup>(6)</sup>하거나 차량의 가속 성능을 일정하게 고려하여 속도를 계획<sup>(7)</sup>하는 방법을 사용하였다. 이런 방법을 사용하면 시간을 단축 할 수는 있지만 최소화 할 수는 없다. 이를 위해서 본 연구에서는 차량의 최대 성능을 고려하기 위해 속도에 따른 가속도 최대 값을 차량 동역학 시뮬레이션 프로그램(CarSim)을 통해 측정하고 속도계획에 이를 반영하였다.

최단 시간 경로를 계획할 때 중요한 고려 사항은 주행 거리와 주행 속도이다. 그런 이유로 최단 시간 경로를 계획 할 때는 주로 거리를 최소화 하여 주행 시간을 단축 하거나 주행 속도를 최대화 하도록 경로의 곡률을 최소화 하여 시간을 단축하는 두 가지 방법을 사용한다. 이 두 가지 방법 중 어떤 경로가 최단 시간 경로가 되는지는 차량의 주행 성능에 달려있다. 그런 이유로 실제 최단 시간 경로는 최단 거리 경로와 최고 속도 경로가 절충된 경로에서 도출된다.<sup>(8)</sup> 본 논문에서는 절충된 경로를 찾기 위해 거리에 대한 비용함수와 곡률에 대한 비용함수를 선형 조합하여 새로운 비용함수를 구성하고 이를 최소화 하도록 최적 문제를 풀어 조합 비율에 따른 후보경로를 생성하였다. 각 후보 경로의 최대 주행 속도 계산을 위해서는 동적 계획법(Dynamic Programming)을 사용하였다. 이때 제한 조건은 차량의 구동,제동 장치의 성능 한계로 인한 종 방향 가속도 조건과 타이어의 마찰 한계로 인한 가속도 제한 조건으로 두었다. 이 두 제한 조건을 만족시키는 각 후보 경로의 최대 주행 속도를 계산하고 이를 통해 각 후보 경로의 주행 시간을 계산하여 최단 시간 경로를 도출하였다.

이렇게 도출된 경로가 제대로 계획된 것인지를 확인하기 위해서는 실제차량 모델을 이용한 시뮬레이션을 통해 검증 할 필요가 있다. 이를 위해서 본 논문에서는 선행 연구에서 개발한 조향, 구동, 제동 제어기를 사용하였다.<sup>(9,10)</sup> 그리고 시뮬레이션은 차량동역학 시뮬레이션 소프트웨어인 CARSIM과 Matlab / Simulink 를 이용하여 이루어졌다.

## 2. 최단 시간 후보 경로 생성

### 2.1 경로 이산화

최단 거리 경로와 최고 속도경로를 계산하는 알고리즘은 경로가 도로 내부에 존재하여야 하기 때문에 제약조건이 있는 최적화 문제를 풀어야 한다. 이 최적화 문제를 좀더 효율적으로 풀기 위해서

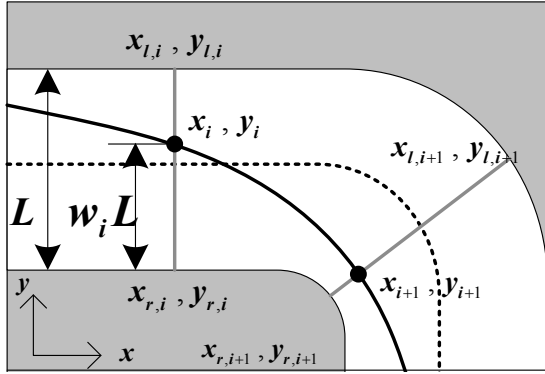


Fig. 1 Trajectory discretization

Fig. 1 과 같이 경로를  $n$  구간으로 나누어  $w$  라는 하나의 변수로 나타내었다.

위의 그림과 같이 정의한 경로를 수식으로 나타내면 식 (1)과 같다.

$$\begin{aligned} \bar{P}_i &= x_i \bar{i} + y_i \bar{j} \\ &= [x_{r,i} + w_i(x_{l,i} - x_{r,i})] \bar{i} + [y_{r,i} + w_i(y_{l,i} - y_{r,i})] \bar{j} \\ &= [x_{r,i} + w_i \Delta x_i] \bar{i} + [y_{r,i} + w_i \Delta y_i] \bar{j} \end{aligned} \quad (1)$$

## 2.2 최단 시간 경로

위와 같이 경로가 정의 될 때 각 구간의 길이는 다음 식과 같이 정의된다.

$$\bar{P}_{i+1} - \bar{P}_i = \Delta P_{x,i} \bar{i} + \Delta P_{y,i} \bar{j} \quad (2)$$

각 축의 거리 변화 량을  $w$  에 대해서 정리하면 다음과 같다.

$$\begin{aligned} \Delta P_{x,i} &= x_{r,i+1} - x_{r,i} + w_{i+1} \Delta x_{i+1} - w_i \Delta x_i \\ &= [\Delta x_{i+1} - \Delta x_i] \left\{ \begin{matrix} w_{i+1} \\ w_i \end{matrix} \right\} + \Delta x_{i,0} \\ \Delta P_{y,i} &= y_{r,i+1} - y_{r,i} + w_{i+1} \Delta y_{i+1} - w_i \Delta y_i \\ &= [\Delta y_{i+1} - \Delta y_i] \left\{ \begin{matrix} w_{i+1} \\ w_i \end{matrix} \right\} + \Delta y_{i,0} \end{aligned} \quad (3)$$

이 식을 이용하여 전체 경로 길이의 제곱을 구하면 다음과 같다.

$$\begin{aligned} S^2 &= \sum_{i=1}^n (\Delta P_{x,i}^T \Delta P_{x,i} + \Delta P_{y,i}^T \Delta P_{y,i}) \\ &= \sum_{i=1}^n \bar{w}_i^T [H_{S,i}] \bar{w}_i + \{B_{S,i}\} \bar{w}_i + \text{cost} \end{aligned} \quad (4)$$

여기서  $\bar{w}_i = \{w_{i+1}, w_i\}^T$  를 나타낸다. 식 (4)를 행렬로 변환하면 식 (5)와 같다.

$$S^2 = \bar{w}^T [H_S] \bar{w} + \{B_S\} \bar{w} + \text{cost} \quad (5)$$

식 (5)는 이차식 형태이므로 이차 계획법을 이용해 이 식을 최소화 하는  $\bar{w} = \{w_1, w_2, \dots, w_n\}^T$  을 구할 수 있다.

## 2.3 최고 속도 경로

최고 속도 경로는 이산화된 경로 각 지점에서의 곡률 합이 최소가 되어야 한다. 그리고 각 지점에서의 곡률이 최소화 되면서도 연속이어야 실제 차량이 주행 가능하기 때문에 각 구간 사이의 경로를 자연 3 차 곡선으로 가정하였다. 이를 식으로 나타내면 식 (6)과 같다.

$$\begin{cases} x_i(t) = a_{i,x} + b_{i,x}t + c_{i,x}t^2 + d_{i,x}t^3 \\ y_i(t) = a_{i,y} + b_{i,y}t + c_{i,y}t^2 + d_{i,y}t^3 \\ t(s) = \frac{s - s_{i0}}{ds} \end{cases} \quad (6)$$

식 (6)은 이산화된 경로 구간 사이를  $t$  라는 매개 변수를 이용하여 3 차식으로 나타내었다. 여기서  $t$  는  $i$  번째 구간경로를 곡선 가로좌표로 일반화한 값을 나타내고 0 에서 1 사이의 값을 가진다. 이산화된 구간 사이에 있는 한 지점의 곡률 ( $\hat{\Gamma}_i$ ) 은 식 (7)과 같이 정의된다.

$$\begin{aligned} \hat{\Gamma}_i^2 &= \left( \frac{d^2 x_i(t)}{ds^2} \right)^2 + \left( \frac{d^2 y_i(t)}{ds^2} \right)^2 \\ &= \left( \frac{d^2 x_i(t)}{dt^2} \left( \frac{dt(s)}{ds} \right)^2 \right)^2 + \left( \frac{d^2 y_i(t)}{dt^2} \left( \frac{dt(s)}{ds} \right)^2 \right)^2 \\ &= \left( \frac{dt(s)}{ds} \right)^4 \left[ \left( \frac{d^2 x_i(t)}{dt^2} \right)^2 + \left( \frac{d^2 y_i(t)}{dt^2} \right)^2 \right] \end{aligned} \quad (7)$$

모든 경로의 구간이 같은 길이( $ds^*$ )를 가진다면 식 (7)을 다음과 같이 단순화 할 수 있다.

$$\begin{aligned} \frac{dt(s)}{ds} &= \frac{1}{ds^*} \\ \Rightarrow \hat{\Gamma}^2 &= \left( \frac{1}{ds^*} \right)^4 \left[ \left( \frac{d^2 x_i(t)}{dt^2} \right)^2 + \left( \frac{d^2 y_i(t)}{dt^2} \right)^2 \right] \end{aligned} \quad (8)$$

식 (8)을 이용하여 전체 경로의 곡률의 합을 구하면 식 (9)와 같다.

$$\Gamma^2 = \sum_{i=1}^n \left[ \left( \frac{d^2 x_i(t)}{dt^2} \right)^2 + \left( \frac{d^2 y_i(t)}{dt^2} \right)^2 \right] \quad (9)$$

우리가 알고자 하는 곡률 합은 이산화된 경로

의 각 지점 ( $t=0$ )에서의 곡률의 합이다.  $t=0$  일 때 자연 3 차곡선의 2 차 미분 값을  $\bar{x}$  에 대해서 나타내면 다음과 같다.

$$\left. \frac{d^2 \bar{x}(s)}{dt^2} \right|_{s=0} = [D] \bar{x} \tag{10}$$

이때  $\bar{x}$  는 도로의 오른쪽 경계에 대해 식 (11)과 같이 나타낼 수 있다.

$$\begin{aligned} \bar{x} &= \bar{x}_r + [dx] \bar{w} \\ [dx] &= [x_{l,1} - x_{r,1} \quad x_{l,2} - x_{r,2} \quad \dots \quad x_{l,n} - x_{r,n}] \end{aligned} \tag{11}$$

식 (10) 과 (11)를 이용하여 이차 미분의 제곱 값을 계산하면 식 (12)와 같이 나타난다.

$$\begin{aligned} \left( \left. \frac{d^2 \bar{x}(s)}{dt^2} \right|_{s=0} \right)^2 &= \bar{w}^T ([dx]^T [D]^T [D] [dx]) \bar{w} \\ &+ 2(\bar{x}_r^T [D]^T [D] [dx]) \bar{w} + \bar{x}_r^T [D]^T [D] \bar{x}_r \end{aligned} \tag{12}$$

식 (12)을  $y$  의 2 차 미분에 대해서도 동일하게 적용하여 곡률의 제곱 값을 나타내면 최단 거리 경로 문제와 같이 이차식 형태인 식 (13)과 같이 나타낼 수 있게 된다.

$$\Gamma^2 = \bar{w}^T [H_r] \bar{w} + \{B_r\} \bar{w} + \text{cost} \tag{13}$$

### 2.4 후보 경로 생성

앞 절에서 최단 시간 경로는 최단 거리경로와 최고 속도경로의 조합 중에서도 발생한다고 가정하였다. 그러므로 최단 시간 경로를 찾기 위해서는 이 두 경로를 조합한 여러 후보 경로를 생성하여 주행 시간을 비교해 보아야 한다. 이를 위해서 식 (5)와 식 (13)를 선형 조합하여 식 (14)와 같은 새로운 비용 함수를 생성하였다. 후보 경로는 이 비용 함수를 최소화 하는 문제를 풀어 도출된다.

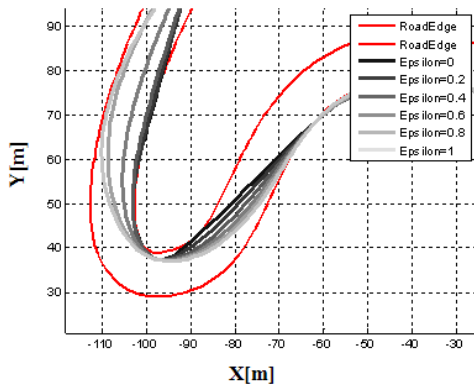


Fig. 2 Candidate trajectories

$$\begin{aligned} F^2 &= \epsilon \Gamma^2 + (1-\epsilon) S^2 \\ &= \bar{w}^T [H] \bar{w} + \{B\} S^2 \end{aligned} \tag{14}$$

$$\begin{aligned} [H] &= \epsilon [H_r] + (1-\epsilon) [H_s] \\ [B] &= \epsilon [B_r] + (1-\epsilon) [B_s] \end{aligned}$$

후보 경로는 비율  $\epsilon$  에 따라서 Fig. 2 와 같이 결정된다.  $\epsilon=0$  일 때는 최단 거리 경로로 코너의 안 쪽으로 경로가 생성되고  $\epsilon=1$  일 때는 최고 속도 경로로 곡률이 최소화 되도록 경로가 생성되는 것을 확인할 수 있다.

## 3. 속도 최적화

### 3.1 제한 조건

앞 장에서 생성한 후보 경로 중 최단 시간 경로를 선정하기 위해서는 각 경로의 최대 주행 가능 속도를 계산 하여야 한다. 이때 최대 속도는 도로의 형상뿐만이 아니라 차량의 주행 성능 또한 고려되어야 한다. 본 논문에서는 속도에 따른 최대 중방향 가·감속도와 타이어의 마찰 한계로 인한 최대 가속도를 제한 조건으로 두었다.

속도에 따른 최대 가·감속도 값을 얻기 위해서 차량 동역학 시뮬레이션 소프트웨어인 CarSim 의 차량 모델을 사용하였다. 풀 스로틀(Full Throttle), 풀 브레이크(Full Brake) 시뮬레이션을 통해 속도별 가·감속도 값을 수집하였다.

Fig. 3 은 2 리터 가솔린 엔진을 장착한 차량의 속도에 따른 가·감속도 데이터이다.

차량이 직선 주행 시에는 중 방향 가속도 제한 값의 영향만 받지만 곡선 주행 시에는 타이어의 마찰 한계로 인해 가속도의 제한을 받는다. 이를 수식으로 나타내면 식 (15)와 같다.

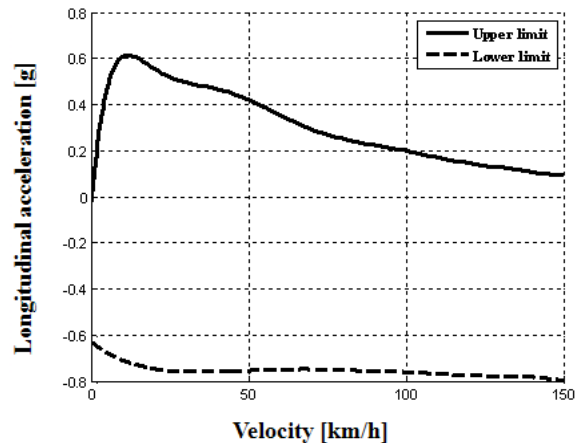


Fig. 3 Acceleration and deceleration data of 2 liter gasoline engine car

$$\sqrt{a_x(t)^2 + a_y(t)^2} \leq \mu g \times \text{safety factor} \quad (15)$$

여기서  $\mu$ 는 마찰 계수,  $g$ 는 중력 가속도를 의미한다. 실제 차량은 고속 선회 시 하중이동으로 인해 각 타이어에 다른 하중이 분배되어 모든 타이어의 마찰력을 최대로 사용할 수 없다. 그러므로 최대 허용 가속도에 안전계수를 곱하여 제한 값을 설정하였다.

### 3.2 동적 계획법

앞 절에서 설정한 제한 조건을 만족 시키면서 각 후보 경로의 최대 주행 가능 속도를 계산하기 위해서 동적 계획법을 사용하였다.

$$g_k(v_k) = t$$

$$J_k^*(v_k) = \min_{v_k \in I_k} \left\{ \begin{array}{l} g_k(v_k) + J_{k+1}(v_k) \\ + G_{ax}(v_k) + G_a(v_k) \end{array} \right\} \quad (16)$$

여기서  $g_k$ 는 시간에 대한 비용함수이고  $t$ 는  $k$  단계에서  $k+1$  단계까지의 주행 시간이다.  $J_k^*$ 는  $k$  단계의 최소 비용이고  $J_{k+1}$ 은  $n$  단계 부터  $k+1$  단계까지의 비용이다.  $G_{ax}$ ,  $G_a$ 는 각각 종방향 가속도 제한에 대한 비용함수와 타이어의 마찰한계로 인한 가속도 제한에 대한 비용함수이다.

각 지점에서의 종· 횡방향 가속도( $a_{x,k}$ ,  $a_{y,k}$ )는 도로의 곡률( $\Gamma_k$ )과 속도( $v_k$ )를 이용하여 식 (17)과 같이 계산한다.

$$a_{x,k} = \frac{v_{k+1}^2 - v_k^2}{2ds^*}, \quad a_{y,k} = \frac{v_k^2}{\Gamma_k} \quad (17)$$

그리고 3.1 절에서 설명한 제한 조건에 대한 비용함수는 식 (18)과 같이 정리되어 조건을 만족 못하는 속도는 동적 계획 알고리즘에 의해 제거된다.

$$G_{ax}(v_k) = \begin{cases} \infty & \text{if } a_{x,\min}(v_k) < a_{x,k} < a_{x,\max}(v_k) \\ 0 & \text{else} \end{cases} \quad (18)$$

$$G_a(v_k) = \begin{cases} \infty & \text{if } \sqrt{a_{x,k}^2 + a_{y,k}^2} < \mu g \times \text{safety factor} \\ 0 & \text{else} \end{cases}$$

속도 최적화 알고리즘을 검증하기 위해서 춘천 모터파크의 경로를 이용하여 최적화를 수행 하였다. 경로 정보는 DGPS 장비를 이용하여 직접 수집 하였고 수집한 경로는 Fig. 4 와 같다. 경로는 6개의 코너와 직선 구간으로 이루어져 있다.

Fig. 4 에서 파란색 실선은 도로의 경계를 나타내고 빨간색 실선은 도로 중심선을 나타낸다. 위의 경로 정보를 이용하여 속도 최적화를 수행한 결과는 Fig. 5 와 같다.

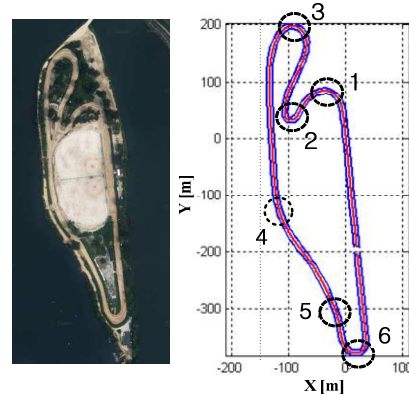


Fig. 4 Digital map of Chuncheon motor park

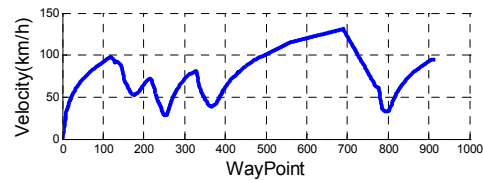


Fig. 5 Optimized velocity profile for minimum traveling time

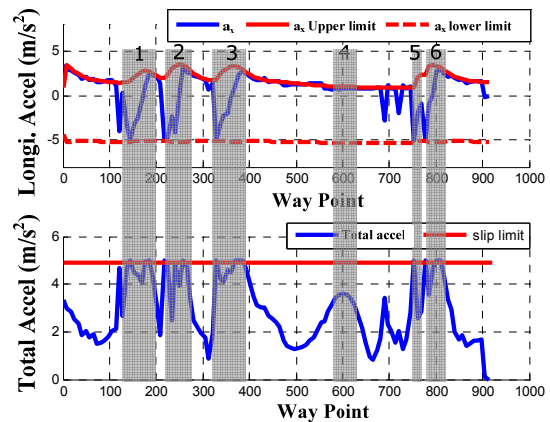


Fig. 6 Acceleration data of optimized velocity profile

위의 속도가 주행 시간을 최소화 하도록 최적화 되었는지 확인하기 위해서는 앞 절에서 제시한 제한 조건을 최대한 활용하였는지를 검토하여야 한다. 이를 위해서 Fig. 6 과 같이 구간 별 가속도를 나타내 보았다. Fig. 6 에서 위의 그래프는 종방향 가속도를 나타내고 아래의 그래프는 종 횡방향 가속도를 합한 총 가속도( $= \sqrt{a_x(t)^2 + a_y(t)^2}$ )를 나타낸다.

Fig. 6 에서 회색으로 표시된 영역은 Fig. 4 의 각 코너 구간과 일치한다. 직선 구간에서는 최대 종방향 가속도를 사용하는 것을 확인할 수 있고 코너 구간에서는 타이어 한계 범위 내에서 최대 가

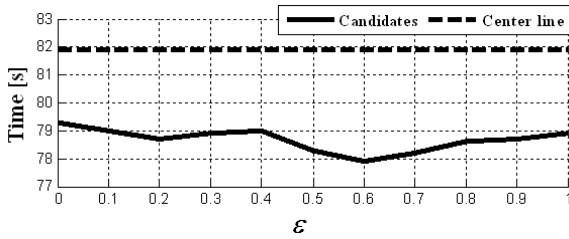


Fig. 7 Travel time of candidate trajectories

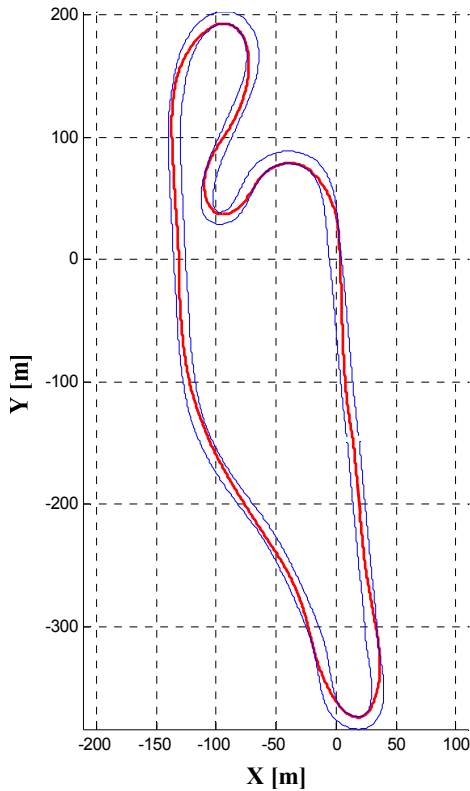


Fig. 8 Minimum time trajectory

속도를 사용하는 것을 사용하는 것을 확인할 수 있다.

3.3 경로 최적화

3.1 에서 생성한 후보 경로를 3.2 의 속도 최적화 방법으로 주행 시간을 계산하여 최단 시간 경로를 선택하였다.

주행 시간 계산 결과  $\epsilon = 0.6$  일 때 최단 시간이 되는 것을 확인할 수 있다. 이때 경로는 Fig. 8 과 같다.

4. 제어기 구조

앞장에서는 도출한 최단 시간 경로가 가속도를 제한 범위 내에서 속도를 최대화 시켜 최단 주행 시간을 가지는 것을 확인하였다. 하지만 이경로를 실제 차량이 계획된 속도대로 주행 할 수 없다면 경로계획이 제대로 되었다고 할 수 없다. 그래서

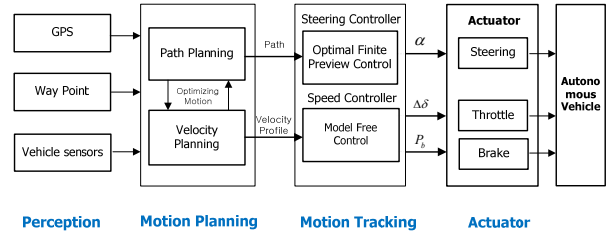


Fig. 9 Control structure

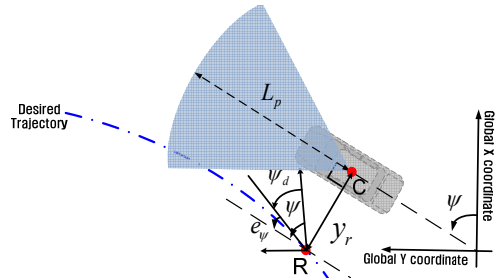


Fig. 10 Trajectory tracking controller

본 논문에서는 조향· 속도 제어기를 사용하여 계획된 경로와 속도대로 차량이 주행할 수 있는지 확인하였다. 제어기의 구조는 Fig. 9 와 같다. 본 논문에서 제안된 알고리즘은 Motion Planning 단계에서 경로와 속도를 비 실시간으로 계획하고 각 제어기는 Motion tracking 단계에서 실시간으로 제어량을 결정하여 차량을 제어하게 된다.

4.1 조향 제어기

조향 제어기를 위한 상태(X)는 횡 방향 위치 에러( $y_r$ ), 각도 에러( $e_\psi = \psi - \psi_d$ ) 그리고 이 두 값의 변화량으로 구성된다. 상태 방적식 (19)는 2 자유도 자전거 모델과 선형 타이어 모델의 횡력식에 의해 얻어진다.<sup>(9,10)</sup>

$$\dot{X} = AX + B\delta_f + F_d w_d \tag{19}$$

$$X = [y_r \quad \dot{y}_r \quad e_\psi \quad \dot{e}_\psi]$$

여기서

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{A_1}{v_x} & -A_1 & \frac{A_2}{v_x} \\ 0 & 0 & 0 & 1 \\ 0 & \frac{A_3}{v_x} & -A_3 & \frac{A_4}{v_x} \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ B_1 \\ 0 \\ B_2 \end{bmatrix} \quad F_d = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \quad w_d = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}$$

$$A_1 = \frac{-2 \cdot (C_f + C_r)}{m}$$

$$A_2 = \frac{2 \cdot (-l_f \cdot C_f + l_r \cdot C_r)}{m}$$

$$A_3 = \frac{2 \cdot (-l_f \cdot C_f + l_r \cdot C_r)}{I_z}$$

$$A_4 = \frac{-2 \cdot (l_f^2 \cdot C_f + l_r^2 \cdot C_r)}{I_z}$$

$$\dot{\psi}_d = \frac{v_x}{\rho}$$

$$B_1 = \frac{2 \cdot C_f}{m} \quad d_1 = -\frac{v_x^2}{\rho} + \frac{A_2}{v_x} \cdot \dot{\psi}_d$$

$$B_2 = \frac{2 \cdot C_f \cdot l_f}{I_z} \quad d_2 = \frac{A_4}{v_x} \cdot \dot{\psi}_d - \ddot{\psi}_d$$

여기서  $m$  은 차량의 질량,  $I_z$  는 요 모멘트 그리고  $l_f, l_r$  은 차량의 질량 중심으로부터 전후륜 축까지의 거리,  $C_b, C_f$  은 Cornering stiffness,  $\rho$  는 목표 경로의 회전 반경,  $\dot{\psi}_d, \ddot{\psi}_d$  는 현재 속도로부터 구해지는 목표 경로의 각속도와 각 가속도를 나타낸다.

조향 제어 입력 값은 다음 식과 같은 비유함수를 최소화하기 위하여 계산된다.

$$J = \lim_{t_f \rightarrow \infty} \frac{1}{t_f} E \left[ \int_0^{t_f} \frac{1}{2} (X^T Q X + \delta_f^T R \delta_f) dt \right] \quad (20)$$

여기서  $Q$  와  $R$  은 실행 오차 방법으로 속도 구간 별로 최적의 결과가 나오도록 조절 되었다.

최종적인 조향 제어 입력 값은 식 (21)와 같이 계산된다. 최적예견제어 방법을 이용하면 입력 값은 피드백 제어 입력( $-K_{opt}X(t)$ )과 피드포워드 제어 입력( $M(t)$ )의 합으로 계산 된다.

$$\begin{aligned} \delta_{desired}(t) &= -K_{opt} \cdot X(t) + M(t) \\ &= \delta_{fb}(t) + \delta_{ff}(t) \end{aligned} \quad (21)$$

피드백 게인( $K_{opt}$ )는 식 (22)와 같은 Riccati equation 으로부터 구할 수 있다.<sup>(11)</sup>

$$\begin{aligned} K_{opt}(t) &= -R^{-1} \cdot B^T \cdot P_{ss} \\ P_{ss} \cdot A + A^T \cdot P_{ss} + Q - P_{ss} \cdot B \cdot R^{-1} \cdot B^T \cdot P_{ss} &= 0 \end{aligned} \quad (22)$$

예견제어 이론을 이용한 피드포워드 제어 입력은 식 (23)으로부터 구할 수 있다.

$$\begin{aligned} M(t) &= -R^{-1} \cdot B^T \cdot \int_0^{T_p} F_1(\xi) \cdot w(t+\xi) \cdot d\xi \\ F_1(t) &= e^{A_c^T t} \cdot P_{ss} \cdot F_d \\ A_c &= A - B \cdot R^{-1} B^T \cdot P_{ss} \end{aligned} \quad (23)$$

#### 4.2 속도 제어기

속도 제어기는 목표가속도를 고려하는 상위제어기와 입력 값을 결정하는 하위제어기로 구분된다. 본 연구의 목적상 상위제어기가 운전자의 안락성 및 편의성을 고려할 필요가 없기 때문에 단순한 PID 제어기로 설계되었다. 하위 제어기는 대상 차량 모델에 대한 정보가 없기 때문에, 무인 차의 속도제어를 위하여 모델 독립형 속도제어기(MFCC)를 사용하였다. 이때 사용한 제어기의 모델은 식 (24)와 같다.

$$\dot{a}_x = -\frac{1}{\tau} a_x + F_{th} \alpha - F_b P_b - g_0 \quad (24)$$

여기서  $a_x$  는 차량의 가속도를 나타내고  $F_{th}, F_b$  는 각각 스로틀과 브레이크에 대한 적응변수를 나

타낸다. 입력은 차량의 스로틀 각도( $\alpha$ )와 브레이크 마스터 실린더 압력( $P_b$ )이고, 출력은 대상차량의 가속도이다. 전체적인 속도제어기의 구조는 Fig. 11 과 같다.

모델 독립형 속도제어기는 가/감속 입력을 가하는 시점을 판단하기 위한 스위칭 알고리즘과 적응 변수를 추정하기 위한 적응변수 adaptation 알고리즘, 페달입력 값을 결정하는 Throttle / Brake 제어기로 구성되어 있다. 스위칭 알고리즘은 차량의 자연감속을 기준으로 입력시점을 결정한다. 만일 목표가속도가 자연감속보다 클 경우 가속 페달을 가하며, 반대의 경우 브레이크를 밟게 된다. Throttle / Brake 제어기는 목표가속도 및 적응변수를 이용하여 입력 값을 결정하게 된다. 각각의 입력 값은 다음 식에 의하여 결정된다.

$$\begin{bmatrix} \alpha \\ P_b \end{bmatrix} = \begin{bmatrix} \frac{1}{\hat{F}_{th}} (a_{x,des} + \tau \dot{a}_{x,des} + \hat{g}_0) \\ \frac{1}{\hat{F}_b} (a_{x,des} + \tau \dot{a}_{x,des} + \hat{g}_0) \end{bmatrix} \quad (25)$$

적응변수 adaptation 알고리즘은 무인차 속도제어기의 가장 중요한 알고리즘이다. 이 알고리즘은 식 (26)과 같이 현재의 입력 값에 대한 차량의 가속도를 이용하여 적응 변수를 예측한다.

$$\begin{aligned} F_{th,m} &= \frac{a_x + \tau \dot{a}_{x,des} + \hat{g}_0}{\alpha} \\ F_{b,m} &= \frac{a_x + \tau \dot{a}_{x,des} + \hat{g}_0}{P_b} \end{aligned} \quad (26)$$

그러나, 예측된 적응변수는 직접적으로 사용될 수 없다. 왜냐하면, 적응 변수는 식 (24)에서 고려되지 못한 차량의 동특성에 크게 영향을 받을 수 있기 때문이다. 그러므로 예측된 적응 변수는 식 (27)과 같은 가중합(weighted sum)에 의해서 계산된다.

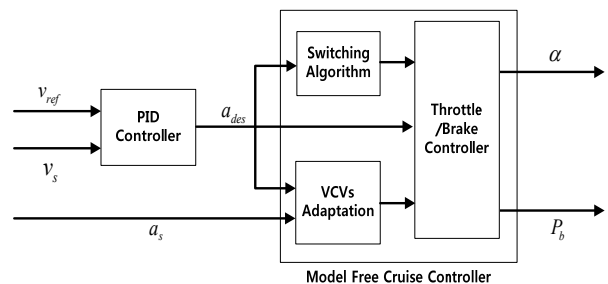


Fig. 11 Speed controller for autonomous vehicle

$$\begin{aligned}\hat{F}_{th}(k+1) &= \lambda \hat{F}_{th}(k) + (1-\lambda)F_{th,m} \\ \hat{F}_b(k+1) &= \lambda \hat{F}_b(k) + (1-\lambda)F_{b,m}\end{aligned}\quad (27)$$

여기서  $\lambda$ 는 현재 측정된 적응변수에 대한 현재 적응변수의 가중치를 나타낸다. 식 (27)를 연속 함수 형태로 표현하면 다음 식과 같다.

$$\dot{\hat{F}} = \kappa(-\hat{F}(k) + F_m(k)) \quad \frac{(1-\lambda)}{dt} = \kappa \quad (28)$$

차량의 가속도와 적응변수에 대한 오차 동역학 분석을 통해 적응변수에 대한 수렴 성을 확인 할 수 있다. 오차는 식 (29)과 같이 정의 할 수 있다.

$$\begin{aligned}e_1 &= a_{x,des} - a_x \\ e_2 &= F - \hat{F}\end{aligned}\quad (29)$$

만약  $\dot{F}$ 이 0에 가깝다고 가정하면 오차 동역학은 식 (24),(25),(28)을 이용하여 구 할 수 있다.

$$\begin{aligned}\dot{e}_1 &= \dot{a}_{x,des} - \dot{a}_x = -\frac{1}{\tau}e_1 - \frac{a_{x,des} + \tau\dot{a}_{x,des} + \hat{g}_0}{\tau\hat{F}}e_2 + \frac{1}{\tau}\tilde{g} \\ \dot{e}_2 &= \dot{F} - \dot{\hat{F}} = -\frac{\kappa\hat{F}}{a_{x,des} + \tau\dot{a}_{x,des} + \hat{g}_0}e_1\end{aligned}\quad (30)$$

여기서  $\tilde{g}$ 은 자연 감속도 측정 오차( $g - \hat{g}$ )를 나타낸다. 오차 동역학식이 비선형 항을 포함하므로 이 시스템은 비선형 시스템이다. 하지만 오차 동역학의 특성 방정식 극점은 제어 입력의 영향을 받지 않고 오직 시상수( $\tau$ )와 가중치( $\kappa$ ) 영향만 받는다. 그러므로 특성 방정식은 다음과 같다.

$$|sI - A| = s^2 + \frac{1}{\tau}s + \frac{\kappa}{\tau} = 0 \quad (31)$$

시상수 ( $\tau$ )는 정확히 알 수는 없지만 아주 좁은 영역 내에서 존재 할 것으로 예상된다. 그리고 가중치( $\kappa$ )를 특성 방정식의 감쇠 비를 고려해 설정하면 시스템은 수렴하게 된다.

대상차량의 자연감속도 역시 제어 입력 값 계산에 영향을 미치지만, 정확한 값이 없더라도, 적응변수 adaptation 알고리즘을 통해 보상이 가능하다. 그러므로 자연 감속도는 다음의 식을 통해 계산되었다.

$$\hat{g}(v_x) = k_1 v_x + k_2 \quad (31)$$

여기서  $k_1, k_2$ 는 자연 감속도를 계산하기 위한 계수이다. 이 값은 30m/s에서 자연 감속도가  $1\text{m/s}^2$ 이 나오도록 설정되었다.

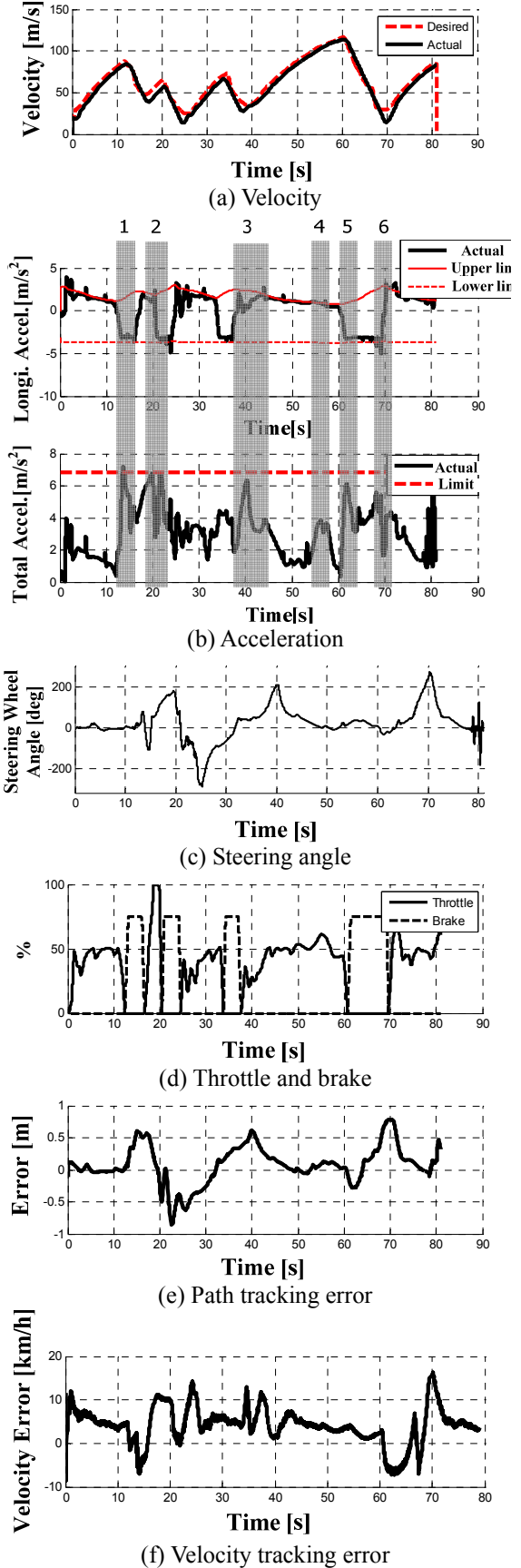


Fig. 12 Simulation Result



## 5. 시뮬레이션

개발된 알고리즘은 차량 동역학 시뮬레이션 소프트웨어 CARSIM 과 MATLAB/SIMULINK 를 활용하여 검증되었다. 앞 장에서 소개한 제어기가 활용되었으며, 이를 활용하여 페루프 시뮬레이션을 수행하였다.

경로와 속도는 3.3 절에서 최적화 하여 도출한 최단 시간 경로와 속도 궤적을 사용하였다. 결과는 Fig. 12 와 같다.

Fig. 12(e)를 보면 차량이 경로는 잘 추종 한 것을 확인 할 수 있다. 하지만 속도의 경우는 차량이 낼 수 있는 최대 가속도를 사용하도록 속도가 설계되었기 때문에 모델 독립형 제어기로 추종하는데 약간의 어려움이 있는 것을 Fig. 12(f)에서 확인 할 수 있다. 하지만 가속과 감속이 바뀌는 구간 외에서는 속도도 잘 추종 하는 것을 확인 할 수 있다. 그리고 Fig. 12(b) 위의 그래프에서 차량의 종방향 가속도가 제한 범위 내에서 최대로 사용한 것을 확인 할 수 있고 Fig. 12(b) 아래 그래프에서는 차량의 총 가속도 ( $=\sqrt{a_x(t)^2+a_y(t)^2}$ ) 또한 타이어 한계로 발생하는 제한 범위 내에서 최대로 사용한 것을 확인 할 수 있다.

## 6. 결론

본 논문은 도로의 경로와 폭 정보가 주어 졌을 때 최단 시간 경로와 속도를 설계 할 수 있는 알고리즘에 대하여 기술하였다. 최단시간 경로 문제는 거리에 대한 비용함수와 곡률에 대한 비용함수를 선형 조합한 새로운 비용함수를 최적화 하는 문제로 귀결 할 수 있었고 속도는 동적 계획법을 통하여 최적화가 가능하였다. 제안된 알고리즘은 춘천 모터파크 맵 데이터를 이용하여 검증되었고 차량 모델을 사용한 시뮬레이션을 통해 실 차량에서 구현 가능한 지를 검증해 보았다.

## 후 기

This work was supported by the BK21 program, SNU-IAMD, the Korea Research Foundation Grant funded by the Korean Government (MEST) (KRF-2009-200-

D00003), and Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology (20120000922)

## 참고문헌

- (1) Thrun, S., Montemerlo, M. and Dahlkamp, H., 2006, "Stanley, the Robot that Won the DARPA Grand Challenge," *Journal of Field Robotics*, Vol. 23, pp. 661~692..
- (2) Cerri, P., Soprani, G. and Choi, J., 2011, "Computer Vision at the Hyundai Autonomous Challenge," *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pp. 777~783
- (3) Yamakadoa, M., Takahashia, J. and Saitoa, S., "Improvement in Vehicle Agility and Stability by G-Vectoring Control," *International Journal of Vehicle Mechanics and Mobility*, Vol. 48, pp. 231~254
- (4) Gustafsson, F., 1997, "Slip-Based Tire-Road Friction Estimation," *Automatica*, Vol. 33, pp. 1087~1099
- (5) Cho, W., Yoon, J. and Yi, K., "Estimation of Tire Forces for Application to Vehicle Stability Control," *IEEE Transactions on Vehicular Technology*, Vol. 59, pp. 638~649
- (6) Villagra, J. and Milanés, V., 2012, "Path and Speed Planning for Smooth Autonomous Navigation" *IEEE Intelligent Vehicles Symposium*.
- (7) Lini, G. and Piazzzi, A., 2010, "Time-Optimal Dynamic Path Inversion for an Automatic Guided Vehicle," *49th IEEE Conference on Decision and Control*.
- (8) Braghin, F., Cheli, F., Melzi, S. and Sabbioni, E., 2008, "Race Driver Model," *Computers and Structure* 86, 1503~1516.
- (9) Kang, J. Y., 2007, Development of the Human Driver Model Based on the Human Factor, MS Thesis, Seoul National University.
- (10) Kang, J. and Yi, K., 2006, "Development of a Finite Optimal Preview Control-Based Human Driver Steering Model," KSAE
- (11) Burl, J. B., 1998, Linear Optimal Control, pp.179~226.
- (12) Kim, H. and Yi, K., 2012, "Design of a Model Reference Cruise Control Algorithm," SAE2012 World Congress