

## 자율주행 차량 제어

### Navigation Control for an Autonomous Road Vehicle

한 민 홍, 문 순 환  
M. H. Han, S. H. Moon



한 민 홍  
· 1942년 3월생  
· 고려대학교 산업공학과 교수  
· 지능 로봇, 생산자동화



문 순 환  
· 1966년 10월생  
· 고려대학교 산업공학과 박사  
· 과정  
· 지능 로봇, 컴퓨터 제어

량의 개념을 도입함으로써 물류 자동화를 실현 시킬 수 있어 산업적인 큰 의의를 갖게 될 것이다. 이 밖에도 차량의 내구시험 또는 기타 시험을 행할 때 운전자를 탑승시키지 않고 이를 수행함으로써 열악한 주행조건으로부터 운전자를 보호할 수도 있으며, 군사 용 목적으로도 사용될 수도 있다.

성공적인 자율주행을 위해서는 차량은 스스로 주어진 주행로의 경계를 인식하고 상황에 맞게 핸들을 제어하면서 주행해야 한다. 또한 설정된 경로에 따라 주행 중 장애물의 인식과 회피, 도로 교통표식 등을 인식할 수

#### 1. 서 론

자율주행 차량이란 목적지를 입력하면 운전자의 도움없이 자율적으로 주행환경을 인식하면서 주어진 목적지까지 안전하고 신속하게 스스로 주행할 수 있는 차량을 말한다. 이러한 차량개발로 운전자가 전혀 필요없는 무인주행 차량의 출현과 그 기능을 부분적으로 이용한 운전보조 시스템으로 우선 활용되어 피로나 운전 미숙으로 인한 교통사고를 예방할 수 있는 기대효과를 얻을 수 있다. 또한 산업체에서의 옥내외 물류 운반에 본 차



Fig.1 KARV-2

있어야 한다. Fig.1은 저자들이 개발하여 실험차량으로 사용되고 있는 자율주행 차량 KARV-2(Korea University Autonomous Road Vehicle-2)를 보인다.

차량이 자율주행을 하기 위한 가장 중요한 요소는 주행로의 경계를 인식하고 정확한 조향 각도를 구하는 일이다. 이의 잘못된 인식과 잘못된 계산은 차량으로 하여금 주행로를 이탈하게 하고 이로 말미암아 많은 물적, 인적손실을 가져올 수 있다.

본 연구에서는 주위환경을 인식하는데 차량의 전방에 CCD 카메라를 부착하여, 이를 통해 얻어진 도로의 영상으로부터 도로 경계의 인식 방법과 핸들의 조향 각도를 계산하는 방법론을 제시하고자 한다. 이러한 자율주행을 위한 연구는 그동안 많이 행해졌다. 그중에서 Pomerleau, D. A.<sup>6)</sup>는 ALVINN을 개발하여 자율주행에 대한 연구들을 수행하였다. 그것은 역전파(back-propagation)방법을 사용하였는데, 입력층(input layer)에는 960개의 유니트(unit)들과 은닉층(hidden layer)에는 4개의 유니트들, 출력층(output layer)은 30개의 유니트들로 구성되어 있다. 신경망의 입력으로 카메라로부터의 30×32 크기의 영상을, 출력으로는 핸들 각도로하여 시스템을 구축하였다. Tsinas, L., Graefe, V.<sup>7)</sup>는 역전파방법인 신경망 이론을 사용하여 도로경계의 인식과 장애물의 인식에 대한 연구를 하였다. 실험 대상인 주행선은 고속도로상에서 큰 곡률반경을 가진 포장 이 잘된 주행로(well-marked lanes)를 대상으로 하였다. Isomoto, K., Nibe, T., Suetomi T., and Butsuen T.<sup>4)</sup>들은 주행선 이탈 방지를 위한 차선 유지 시스템(lane-keeping system)을 개발하였다. 이 시스템은 주행선 인식(lane-marker recognition), 이탈 예측 방법(departure prediction), 자동 조향(automatic steering control actuator)들을 포

함한다.

차량의 고속 자율주행이 성공하기 위해 필수적인 사항은 주행선의 인식과 핸들 조향 각도의 실시간적 산출이다. 이를 위하여는 주행로 경계가 실시간에 추출될 수 있어야 하고 이로부터 핸들의 조향 각도를 구하기 위해 정확하면서도 간단한 수학적인 모델링이 필요하다. 그러나 정확하고 간단한 모델 구축은 결코 쉬운 일이 아니며 모델링을 했다해도 그 구성요소들을 조율(tuning)하는데 많은 시간을 소비하게 된다. 따라서 본 연구에서는 이러한 모델링의 과정을 거치지 않고도 조향 각도를 구하기 위하여 인공 신경망의 일종인 경쟁 네트워크(competitive network)를 적용하는 방법을 기술하고자 한다. 이는 기존에 많이 사용된 역전파(back-propagation)방법보다 학습을 위해 많은 시간을 절약할 수 있고, 새로운 환경을 학습을 시킬 경우 과거에 학습된 내용들은 잃어버리지 않고 곧바로 추가학습이 가능하게 된다. 또한 알고리즘을 구현하기가 용이하며 인간 운전자로부터 실시간 학습 직후 바로 컴퓨터에 의해 자동 조향을 할 수 있는 신속성의 특징을 가진다.

자율주행 차량이 주어진 경로의 주행을 위

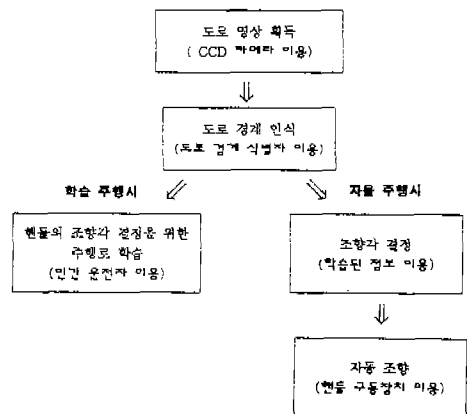


Fig.2 자동조향을 위한 절차

해 환경에 적절한 조향 각도를 산출하고, 자동 조향을 하기 위한 절차는 Fig.2와 같다.

## 2. 주행선 추출

### 2.1 식별자

본 연구에서는 차량이 주행하는 주행로의 주행선을 올바르게 인식하기 위해서 경계 연산자를 이용하였으며, 그 경계 식별자의 gradient는 다음과 같다.

$$\text{gradient} = (Z_3 + 2Z_6 + Z_9) - (Z_1 + 2Z_4 + Z_7) \quad (1)$$

Z <sub>1</sub>	Z <sub>2</sub>	Z <sub>3</sub>
Z <sub>4</sub>	Z <sub>5</sub>	Z <sub>6</sub>
Z <sub>7</sub>	Z <sub>8</sub>	Z <sub>9</sub>

여기서, Z는 해당 픽셀의 화소값(gray level)이다.

만약 gradient 값이 어느 주어진 임계값보다 클때에는 해당위치를 주행선으로 식별한다. 그 임계값의 결정은 항상 고정된 값을 사용하는 것이 아니라 주행로의 사정에 알맞게 유동적인 값을 사용함으로써 주행선 추출에 융통성을 갖도록 하였다.

### 2.2 주행선 인식 방법

도로의 영상은 차량의 앞에 부착된 CCD 카메라를 통해 실시간으로 계속해서 얻어진다. 차량의 올바른 주행을 위해 실시간으로 얻어지는 영상으로부터 도로의 주행선을 정확하게 파악해야 한다. 그러나 도로의 영상은 주위 환경의 변화로 인하여 수시로 변하게 된다.

실제 주행하면서 도로 주행선을 추출하다 보면 많은 어려움이 존재한다. 가령 건물이나 도로변의 가로수 등의 그림자로 인하여 도로 주행선의 식별이 어려운 경우, 또는 주행선이 불연속적이기 때문에 이를 추출하지

못하는 경우, 주행로의 노화로 인해 주행선이 지워져 식별이 어려운 경우 등이 있다. 이외에도 주위 조도의 변화로 인해 카메라로 들어오는 도로의 영상이 불완전하여 주행선의 추출을 어렵게 하는 경우도 있다. 또한 전방 또는 측방 등의 차량들로 인하여 주행선의 인식이 어려운 경우도 있다.

그외에도 주행을 하다보면 도로의 보수 또는 재포장의 관계로 주행선이 그려져 있지 않은 경우에 주행을 계속해야할지 혹은 차량을 정지시켜야 할지 결정하는 것도 중요한 문제이다. 즉, 차선을 인식하지 못하고 주행을 계속하는 것은 많은 위험을 가지지만 차선의 인식을 못하는 것이 어느 순간인 경우 차량을 정지시키는 것 또한 불필요한 행동이기 때문이다.

도로 경계의 인식은 카메라로부터 입력된 영상으로부터 주행로의 주행선을 전부 추출하는 것이 아니라 주행선의 개략적인 윤곽만 추출할 수 있도록 몇 개의 지점만 추출한다. 이는 주행도로들의 형태가 몇 개의 점들만 추출하고도 충분히 그 형태를 추정될 수 있기 때문이다. 도로 주행선의 추출을 위해 영상화면상에서 수직으로 세 지점을 선택하여 그 각각의 지점으로부터 좌측과 우측방향으로 탐색하여 도로의 주행로를 이루는 6개의 점들을 식별하도록 하였다. 또한 주행로의 경계를 찾는데 탐색영역을 설정하여 그 영역내에서만 경계를 추출하도록함으로써 불필요한 탐색시간을 줄임은 물론 영역내에서 잡음에 의한 주행로 경계의 인식 오차 가능성을 제거하였다. 이는 실시간적인 주행로 경계의 추출을 가능하게 하여 고속도로에서 고속주행이 가능하게 되었다. 주행로 경계를 인식하는 방법을 간단히 살펴보면 다음과 같다. 여기서 y좌표는 영상화면에서 수직 좌표이고, x좌표는 수평 좌표를 나타낸다.

Step 1 : 영상 화면상의 3점의 y좌표를

찾는다.

이때 y좌표의 위치가 영상화면에서 아래부분에 위치할수록 주행로 경계를 추출하는데 수월하지만 그 위치와 차량과의 거리가 너무 가깝다면 원활한 자율주행을 하기가 어렵게 된다.

Step 2 : 탐색영역을 정한다.

바로 직전에 추출한 주행로 경계를 포함하는 6개의 직사각형의 탐색 영역을 지정한다. (Fig 4. 참조)

Step 3 : 도로 경계 식별자를 이용하여 주행로 경계를 인식한다.

탐색 영역내에서 주어진 주행로의 양측 경계를 찾는 단계이다. 그 경계를 찾기 위해서는 주행로 경계 식별자를 좌측과 우측으로 수평이동하면서 식(1)에 의해 gradient 값을 구하고, 주어진 임계값과 비교함으로써 구할 수 있다. 이 때 추출된 경계들을 잠정적으로 주행로의 경계라 가정한다. 이 단계에서는 다음의 세 가지의 경우가 나타날 수 있다.

case 1 : 양측의 경계 모두를 찾은 경우

왼쪽 경계점이나 오른쪽 경계점이 한 개 이상 발견된 경우에는 이전 구해진 지수 평활화된 값과 비교해 가장 가까운 지점을 주행로의 경계라고 결정한다. 좌측과 우측 경계의 지수 평활화된 값은 아래의 식에 의해 계산된다.

$$\begin{aligned} X_{l\_avg}(t+1) &= \alpha \times X_{l\_avg}(t) + (1-\alpha) \times X_l, \\ X_{r\_avg}(t+1) &= \alpha \times X_{r\_avg}(t) + (1-\alpha) \times X_r, \\ 0 \leq \alpha \leq 1 \end{aligned} \quad (2)$$

여기서,  $X_l(t)$ ,  $X_r(t)$  : t 시점에서의 좌측과 우측 경계의 x 좌표 위치,

case 2 : 왼쪽이나 오른쪽 중 어느 한쪽의 경계를 인식하지 못한 경우  
구하지 못한 경계에 대해서는 이전에 구한

$X_{l\_avg}$ 나  $X_{r\_avg}$ 의 정보와 도로 폭의 넓이의 자료를 이용하여 경계를 추정하도록 한다.

case 3 : 양측의 경계를 모두 인식하지 못한 경우

불연속적인 도로의 여건이나 기타의 여러 가지 이유로 도로의 양측 경계 모두를 인식하지 못한 경우이다. 도로의 경계의 추정은 과거에 구한 자료들이 사용된다. 이 상황이 계속될수록 오차가 증가하게 되고, 일정 시간 이상인 경우는 주행의 계속여부를 결정한다.

Step 4 : 주행로의 경계를 검증한다.

주행로의 경계가 제대로 인식되었는지 또는 잡음으로 인한 잘못된 경계인식인지를 판정한다. 다음의 조건을 모두 만족한다면 주행로의 경계를 정확히 인식한 것으로 간주한다.

1. 구해진 주행로 경계의 폭을 주어진 임계값과 비교하고 일정 범위내에 있는지를 검토
2.  $d_1 < d_2 < d_3$ ,
3.  $c_1 \geq c_2 \geq c_3$ , 또는  $c_1 < c_2 < c_3$

여기서,  $d_1$  : 각각의 y 좌표내에서 도로 경계의 폭

$c_1$  : 각각의 y좌표내에서 추출한 도로 경계 중심의 x 좌표

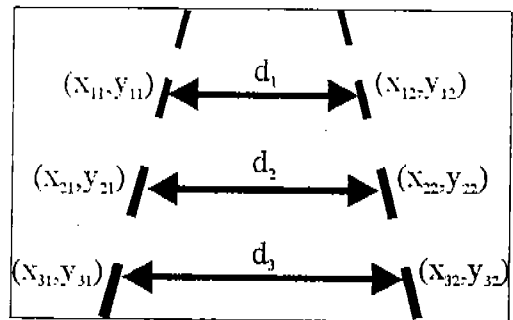


Fig.3

Step 5 : Step 1로 간다.

Fig.4는 위의 절차들을 이용하여 차량이 실제 도심도로를 주행하면서 주행로의 경계

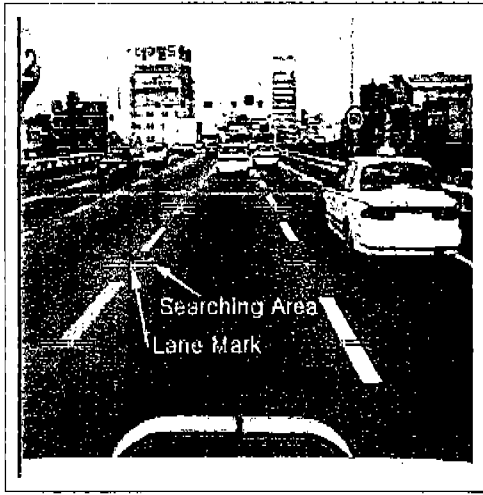


Fig.4 주행 중 주행로 경계의 인식

를 추출하고 있는 장면을 나타낸다.

### 3. 핸들 조향 방법

본 연구에서는 수학적 모델링을 위해 필요한 요소들을 전혀 고려하지 않고도 알고리즘의 구현이 용이하며 실시간 주행을 원활하게 할 수 있는 인공 신경망 방법을 제시한다. 도로의 주행하는 상황을 학습시키기 위해서 입력층과 경쟁층의 두 개의 층으로 구성된 경쟁 네트워크(competitive network)를 이용하였다. 이는 학습하고자 하는 주행로를 먼저 인간 운전자가 직접 주행 한 직후 바로 컴퓨터가 자율주행을 수행할 수 있어 실시간 학습에 의한 자율주행이 가능하다. 또한 새로운 상황을 학습시키고자할 때 기존에 학습된 내용들을 손실하지 않고도 추가학습이 가능하다.

제안된 모델의 입력층은 앞절에서 구한 주행로 경계의 추출된 점들의 위치와 차량의 속도로 구성된다. 이 입력된 벡터들은 경쟁층에 있는 모든 뉴런 사이에서 서로 경쟁하여 가장 적합한 출력 뉴런을 찾게 되고 그에

해당하는 핸들의 각도를 핸들 구동장치에 보냄으로써 주행이 시작된다. 학습을 위해 경쟁층에 있는 뉴런의 수를 미리 지정하지 않고 동적인 특성을 갖도록하여 상황에 알맞게 적절히 생성되도록 하였다. 이렇게 구성된 네트워크(network)에서 가장 적절하다고 선정된 출력 뉴런은 그 자신 고유의 핸들 각도를 나타내는 class를 갖게 되는데, 경쟁층에 있는 뉴런의 수를 고정된 값을 사용하지 않고 동적인 값을 사용한 것은 다음과 같은 이유에서 였다. 즉, 추출한 주행로 경계점들의  $x$  좌표들과 차량의 속도를 입력벡터로하여 네트워크가 구성되는데 동일한 핸들의 각도를 갖는다고 입력벡터가 항상 동일하지 않고 여러 개의 입력벡터가 존재할 수 있다. 이는 비록 같은 상황 즉 동일한 입력 벡터를 가질지라도 인간 운전자가 동일한 핸들 각도를 출력하지 않을 수 있기 때문이다. 또는 현재 얻어진 도로 경계의 위치에 무관하게 인간 운전자는 앞으로의 방향을 미리 예측하여 핸들을 제어함으로써 비록 도로 경계들의 위치는 틀릴지라도 같은 핸들의 각도로 주행을 할 수도 있기 때문이다.

#### 3.1 조향 각도 결정 방법

핸들각도를 계산하기 위해 제시하는 방법은 다음과 같이 두 개의 단계로 구성되었다. 첫 번째로 주행 환경에 적절한 조향각도를 계산하기 위해 인간 운전자로부터 학습을 시키는 단계이다. 주행 환경의 학습을 위해 카메라로 부터의 영상을 전체 이미지를 대상으로 하지 않고 단지 그 영상에서 주행로의 경계점들이 이용되었다. 만약 전체 이미지를 대상으로 하는 경우에는 주행 도로에 무관한 정보들 가령 차량, 건물, 가로수 등의 처리들을 포함할 뿐아니라 이들의 처리들로 인하여 많은 처리시간을 필요로 한다. 단지 주행로

경계의 특징들을 이용함으로써 학습시 주행 도로의 곡률 반경의 변화를 제외한 다른 환경의 변화가 있을 지라도 다시 학습을 시킬 필요가 없게 된다. 두 번째로는 학습후의 단계로써 도로의 주행환경이 주어지면 학습된 자료로부터 적합한 조향 각도를 계산해 주행 하도록하는 단계이다.

학습을 위해 사용된 특징점들은 주행로 경계점들의 위치와 차량의 속도가 사용되었다. 차량의 속도는 차량의 바퀴에 부착된 홀소자가 한회전당 10개의 펄스를 발생시키는데, 발생된 펄스로부터 간단한 산술식을 이용하여 현재 차량의 속도를 감지하도록 하였다. 그리고 핸들의 각도는 핸들축에 부착된 absolute encoder에 의해 계산하였다.

### 3.1.1 학습 방법

차량이 처음 주행로를 주행하기 위해서는 주행로에 대한 학습을 필요로 하게된다. 주어진 주행로를 학습하기 위한 절차는 다음과 같다.

Step 1 : 신경망의 입력층에 입력 벡터를 제시한다.

입력벡터는 세 지점의 각기 다른 y 좌표들에서 추출한 주행로 경계들의 x 좌표들과 차량의 속도로 구성된다.

Step 2 : 입력 벡터와 모든 뉴런들과의 거리를 계산한다.

입력과 출력 뉴런 j 사이의 거리  $d_j$ 는 다음 식으로 계산된다.

$$d_j = \sum_{i=1}^N [X_i(t) - W_{ij}(t)]^2 \quad (3)$$

여기서,  $X_i(t)$ 는 t 시점에서의 i번째 입력 벡터의 원소이고,  $W_{ij}(t)$ 는 t 시점에서의 i번째 입력벡터의 원소와 j번째 출력 뉴런 사이의 연결강도이다. N은 입력 벡터 원소의 수이다.

Step 3 : 최소 거리에 있는 뉴런  $j^*$ 를 선택한다.

1. 입력패턴과 그 뉴런과의 거리가 주어진 임계거리 이상인 경우에는

- ① 새로운 뉴런을 생성하고,
- ②  $W_{ij}(t) = X_i(t)$ , for  $i=1, 2, 3, 4$
- ③ Step 1로 간다.

2. 그렇지 않은 경우 Step 4로 간다.

Step 4 : 뉴런  $j^*$ 의 연결강도를 다시 조정한다.

먼저, 뉴런  $j^*$ 의 class와 입력벡터의 class를 비교한 뒤 식(4), 식(5)를 이용하여 연결강도를 재조정한다. 여기서 class는 encoder의 값을 나타낸다.

1. 동일한 class를 가지는 경우

$$W_{ij}(t+1) = W_{ij}(t) + \alpha(t)[X_i(t) - W_{ij}(t)], 0 \leq \alpha(t) \leq 1 \quad (4)$$

2. 다른 class를 가지는 경우

$$W_{ij}(t+1) = W_{ij}(t) - \alpha(t)[X_i(t) - W_{ij}(t)], 0 \leq \alpha(t) \leq 1 \quad (5)$$

그리고 Step 1로 간다.

위 절차에서  $\alpha(t)$ 와 새로운 뉴런을 생성하기 위해 그 기준이 되는 임계값의 결정은 중요한 요소가 된다. 임계치를 크게 설정하면 할 수록 생성뉴런의 수가 적고 처리시간은 짧아지는 반면, 핸들의 조향 각도를 결정하는데 분별력을 약화시켜 보다 많은 오차를 야기하게 된다. 그와 반대로 임계치를 낮추게 되면 생성 뉴런의 수를 증가하여 보다 정밀한 제어를 할 수 있지만 불필요하게 많은 뉴런을 생성하여 실시간 처리를 저해할 수 있다. 이의 결정은 실험을 통해서 오차와 처리시간을 고려하여  $\alpha$ 의 값을 0.1로 하였고,

그 임계값은 10으로하여 모델링을 하였다.

### 3.1.2 조향 각도 결정

차량이 자율주행하고자 하는 주행도로를 먼저 학습시킨 후 이번에는 학습시킨 자료들을 이용하여 주어진 주행로를 주행하기 위해 조향 각도를 구하는 방법에 대해 살펴본다. 그 조향각 결정 절차는 다음과 같다.

Step 1 : 계산된 차량의 속도와 추출된 주행로 경계로부터 세 지점의 x좌표들을 계산해 인공 신경망에 제시한다.

Step 2 : 신경망의 입력층에 입력된 값들과

모든 뉴런간의 거리를 계산한다.

Step 3 : 최소 거리에 있는 출력 뉴런을 선택하고, 그에 해당하는 핸들의 조향 각도를 찾는다.

## 4. 실험

주행 실험을 위하여 제작된 차량의 제어시스템 구조를 Fig.5에 나타내고 있다. 이 차량은 핸들의 자동 조향, 장애물 감지 및 회피 기능, 자동 속도 조절 기능, 자동 변속 기능, 원격 조정 기능 등의 기능들을 포함한다. 이

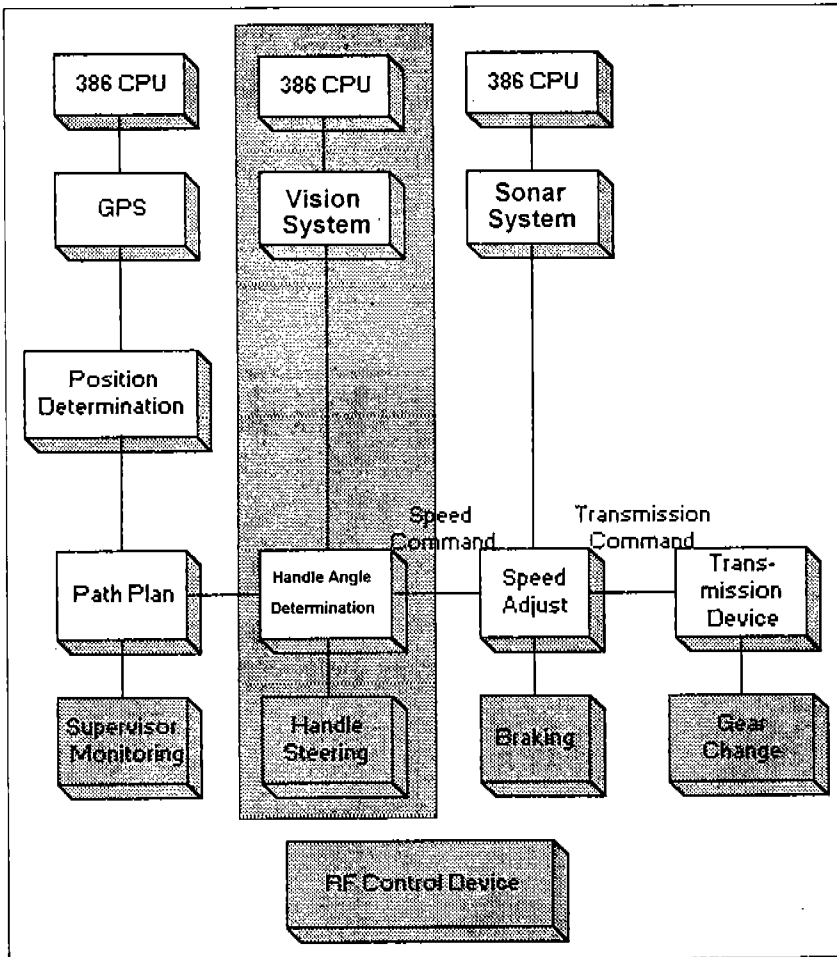


Fig.5 시스템 구조도

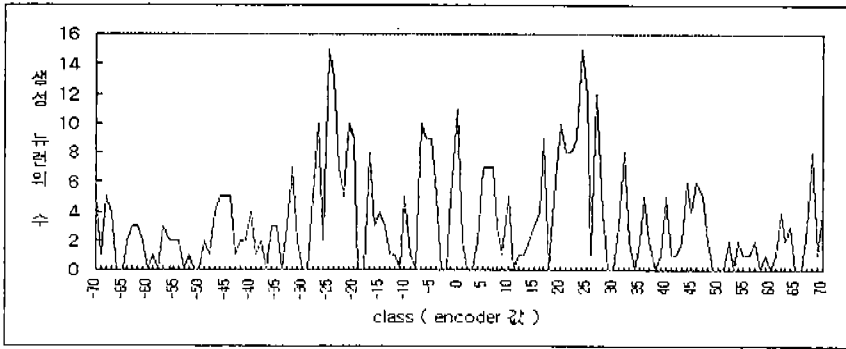
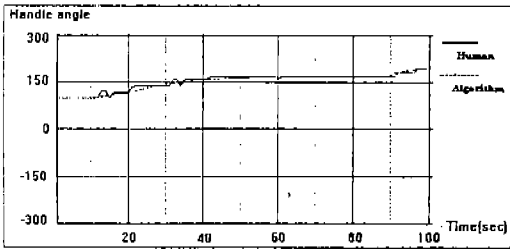
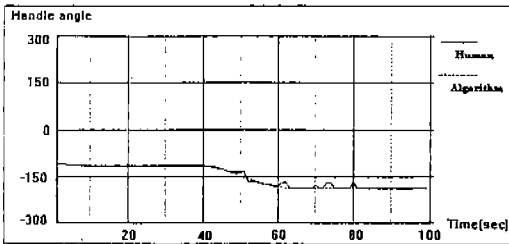


Fig.6 핸들의 각도에 따른 형성된 뉴런들의 수



(a) 좌측 주행



(b) 우측 주행

Fig.7 핸들 조향 각도의 비교

러한 기능들이 서로 상호 유기적으로 작동함으로써 주어진 목적을 보다 안전하고 신속하게 처리되도록 되어 있다. 그러나 본 논문에서는 자율주행에 필요한 기능 가운데 자율조향에 초점을 맞추어 기술하고자 한다.

주행 실험은 곡률 반경 10m의 곡선 주행로에서부터 직선도로까지 다양한 형태의 도로조건하에서 행해졌다. 도로 경계의 인식을

위한 Vision System으로는 해상도 256×256를 갖는 영상처리 보드(vision board)로 구성되었고, 이는 초당 30frame을 처리할 수 있다. 주행로 경계의 인식과 조향 각도의 계산을 위해 40MHz IBM호환 386PC가 이용되었다. 주행로 경계의 추출과 조향각도를 계산하는 속도는 평균 초당 25.5번이며, 이는 차량이 100km/hr의 고속주행을 하더라도 실시간의 처리에 충분한 계산속도이다.

Fig.6은 곡률반경 10m인 주행로를 인간 운전자에 의해 학습을 시킨 후 차량 핸들의 회전각도에 대해서 각 class에서 형성된 뉴런들의 수들을 도식화한 것이다. 수평축은 class에 해당하는 encoder의 값들을 나타내고, 수직축은 각각의 class에 해당하는 형성된 뉴런들의 수를 나타낸다. 이는 encoder의 값이 -70에서 70의 범위내로 주행할 수 있는 모든 주행로에 대해서 학습이 이루어진 것을 의미하고, 생성된 뉴런의 수는 좌측회전의 경우 229개, 우측회전의 경우 220개, 중심에 있는 경우가 11개 그래서 전체 생성된 뉴런의 수는 460개이다.

주어진 경로를 주행 중 정상적인 인간 운전자와 구현된 알고리즘에서 구한 핸들의 각도를 비교한 결과를 Fig.7에 보여준다. 여기에서 점선은 인간 운전자의 핸들의 각도, 직



선은 컴퓨터에 의해 계산된 핸들의 각도를 나타낸다. Fig.7(a)는 차량이 직선에서 좌측으로 주행하는 경우이고, (b)는 우측으로 주행하는 경우를 나타낸다.

인간 운전자와 구현한 알고리즘에 의해 계산된 핸들의 각도를 비교해본 결과 그 오차의 평균은 12.3도 표준편차는 11.2도였다. 인간 운전자의 운전 형태는 곡선도로에서나 직선도로에 관계없이 부드러운 반면 컴퓨터에 의한 주행은 다소 변화가 있음을 알 수 있다.

## 5. 결 론

본 연구에서는 자율주행을 위해 무엇보다도 중요한 주행할 주행로의 주행선 인식과 핸들의 조향각을 결정하는 방법론에 대해 살펴 보았다.

먼저, 통계적인 기법을 이용하여 실시간적인 주행로 경계의 추출방법을 제안하였다. 주행로 경계의 인식을 위해 전체 영상 화면으로부터 주행로 경계를 전부 추출하지 않고, 몇 개의 지점만을 이용하여 주행로의 주행선을 식별하도록 함으로써 보다 실시간 처리를 가능하게 하였다. 또한 조향각의 결정을 위해서 실시간 학습이 가능하고 알고리즘 구현이 용이한 경쟁 네트워크(competitive network)를 사용하였다. 이는 학습하고자 하는 주행로를 먼저 인간 운전자에 의해 운전한 다음 바로 직후 자동으로 자율주행이 가능할 뿐만 아니라 과거에 학습된 정보들을 잃어버리지 않고 새로운 환경에서의 추가적인 재학습이 가능하다는 장점을 갖고 있기 때문이다. 또한 수학적인 모델링을 사용하는 경우에 비하여 조향각도를 구하는 과정이 단순하여 보다 실시간적인 처리를 할 수 있어 차량의 주행이 저속이나 고속에 관계없

이 원활한 주행이 가능하였다.

본 논문에 제시된 방법을 활용하여 자율조향을 실시할 때 시속 100-120km/hr의 속도 주행이 가능하였다.

## 참 고 문 헌

1. Fausett, L., Fundamentals of Neural Networks-Architectures, Algorithms, and Applications, Prentice Hall International, Inc., U.S.A., 1994.
2. Freeman, J. A., and Skapura, D. M., Neural Networks-Algorithms, Applications, and Programming Techniques, 1991.
3. Gonzalez, R. C., and Wools, R. E., Digital Image Processing, Addison-Wesley Publishing Company, U. S. A., 1992.
4. Isomoto, K., Nibe, T., Suetomi, T., and Butsuen, T., "Development of a Lane-Keeping System for Lane Departure Avoidance", The Second World Congress on Intelligent Transport Systems -'95 Yokohama, Nov. 9~11, pp. 1266~1271, 1995.
5. Kohonen, T., Self-Organizing Maps, Spinger, Germany, 1995.
6. Pomerleau, D. A., Neural Network Perception For Mobile Robot Guidance, Kluwer Academic Publishers, U. S. A., 1993.
7. Tsinas, L. and Graefe, V., "Coupled Neural Networks for Read-time Road and Obstacle Recognition by Intelligent Road Vehicles", Proc. of 1993 International Joint Conference on Neural Networks, pp. 2081~2084, 1993.