SOM을 이용한 자율주행로봇의 횡 방향 제어에 관한 연구

A Study on the Steering Control of an Autonomous Robot Using SOM Algorithms

김 영 욱*, 김 종 철*, 이 경 복*, 한 민 홍*
Young-Wook Kim, Jong-Chul Kim, Kyong-Bok Lee, Min-Hong Han

요 약

기존의 횡 방향제어 알고리즘은 도로에서 발생할 수 있는 변수를 고려하여 알고리즘을 작성해야 했다. 이러한 제어 알고리즘을 작성하기 위해서는 주행해야 하는 도로에 따라 파라미터를 재조정해야 하는 문제와 대량의 계산이 요구되는 모델링 문제가 있었다. 본 논문에서는 지능적 횡 방향제어가 가능한 학습알고리즘에 관해 연구하였다. 학습알고리즘은 인공지능 알고리즘 중 자기구성 알고리즘을 사용하였으며 학습데이터는 도로의 특징점을 이용하였다. 컴퓨터를 이용한 시뮬레이션 결과 본 논문의 학습알고리즘에 의한 조향제어가 가능한 것을 알 수 있었고 실제로 주행이 가능한 자율이동로봇에 적용하여 학습에 의한 횡 방향제어가 가능한 것을 확인하였다.

Abstract

This paper studies a steering control method using a neural network algorithm for an intelligent autonomous driving robot. Previous horizontal steering control methods were made by various possible situation on the road. However, it isn't possible to make out algorithms that consider all sudden variances on the road. In this paper, an intelligent steering control algorithm for an autonomous driving robot system is presented. The algorithm is based on Self Organizing Maps(SOM) and the feature points on the road are used as training datum. In a simulation test, it is available to handle a steering control using SOM for an autonomous steering control. The algorithm is evaluated on an autonomous driving robot. The algorithm is available to control a steering for an autonomous driving robot with better performance at the experiments.

Key Words : Artificial Intelligence, Image Processing, Autonomous Driving, Neural Network, Autonomous Robot

I. 서 론

과거에는 로봇의 횡 방향 제어를 구현하기 위해 초음파센서나 적외선센서를 사용하였다. 하지만 이러한 센서는 사용하기는 편리하지만 도로의 정보를 종합적으로 파악할 수 없는 단점이 있었다. 영상처리 하드웨어의 발전으로 인하여 이러한 단점을 극복하기 위한 방법이 모색되며 초음파센서 대신 이미지센서를 사용한 로봇의 횡 방향 제어연구에 관심을 갖기 시작하였다. 그러나 이미지센서를 사용한 로봇은 기존의 초음파센서 로봇보다 도로의 정보를 종합적으로 파악할 수 있는 장점이 있지만 로봇의 주행을 위한 횡 방향 제어알고리즘을 작성하는 것은 초음파센서 로봇보다 복잡하다는 단점을 가지고 있다. 이미지센서에 의한 로봇의 횡 방향 제어를 구현하기 위해서는 로봇이 주행할 때 발생할 수 있는 모든 도로 상황을 감안할 수 있는 알고리즘을 작성해야 하는데 도로의 특성상 모든 상황을 고려할 수 있는 알고리즘을 작성하는 것은 실질적으로 불가능하다. 최근에는 이러한 문제점을 보완하기위해 신경망 알고리즘을 로봇의 횡 방향제어 알고리즘으로 이용하는 연구가 이루어지고 있다.[1] 로봇의 방향을 제어하는데 특정 제어 알고리즘 대신 신경망알고리즘을 사용하게 되면 영상으로부터 들어오는 도로정보를 이용해 로봇을 학습시키는 것이 가능하게 되기 때문에 특별한 제어 알고리즘 없

*고려대학교 산업시스템정보공학과
접수 일자 : 2003. 7. 29 수정 완료 : 2003. 8. 29
논문 번호 : 2003-3-10

이도 횡 방향제어가 가능하게 된다. 또한 학습되는 영상 속에는 도로에서 발생하는 모든 상황이 포함되어 있기 때문에 특정 상황에서의 횡 방향제어도 학습 후에는 가능하게 된다.

본 연구에서는 신경망알고리즘 중 자기구성알고리즘을 사용하여 학습에 의한 횡 방향제어를 시도하였다. 학습데이터를 습득하기 위해 영상을 검색하는 방식을 전체 영역이 아닌 부분영역 검색으로 전환하여 영상검색으로 인한 성능저하를 방지하였으며, 검색된 영역에 대해서도 전에 검색된 주행선 검색영역을 기준으로 주행선을 판단하도록 작성하여 도로의 일부분이 보이지 않아도 도로의 윤곽을 파악할 수 있도록 하였다. 검색된 도로의 특징점에 대해서도 실제 도로의 특징점이 맞는지를 판단하도록 하여 잘못된 검색영역으로 인한 학습이나 학습조향제어에 문제가 발생하지 않도록 했다. 학습은 도로의 특징점에 따른 목표 조향각을 연결시키는 작업을 통해 이루어지며 학습이 완료된 후에는 학습데이터를 이용하여 로봇 스스로 횡 방향제어를 하게 된다. 학습알고리즘에 의한 횡 방향제어의 타당성을 검토하기 위해 먼저 컴퓨터로 시뮬레이션 하였으며, 알고리즘의 타당성을 확인한 후 제안된 알고리즘을 실험용 주행 로봇에 적용하여 실험을 하였다. 실험을 통해 일정한 규격의 트랙을 설치하여 학습한 주행 로봇의 주행능력을 알아보았다.

II. 관련연구

학습이 가능한 지능로봇을 개발하기 위하여 산업체에서는 신경망 알고리즘을 로봇의 제어알고리즘으로 사용하기 시작하였다. 초기 학습에 사용한 로봇은 주행 중에 지엽적인 정보만을 수집할 수 있는 LABMATE 형식의 로봇이었다.[2][3] 이러한 로봇은 초음파센서나 적외선센서 시스템을 갖추고 있기 때문에 도로의 상황을 종합적으로 파악할 수는 없지만 로봇의 현재위치나 전방물체에 대한 계산이 센서를 통해 신속하게 이루어지는 장점이 있다. 이러한 센서를 이용한 학습은 로봇에 부착한 컴퓨터를 이용해 주로 이루어졌으며 로봇의 제어를 위해 사용된 학습알고리즘은 역전파 알고리즘이었다.[4] K. J. LEE는 적외선센서가 장착된 로봇에 자기구성신경망 알고리즘을 이용하여 학습에 의한 로봇의 주행이 가능하다는 것을 보여 주었다.[5] 하지만 이러한 지엽적인 정보를 이용하는 학습 로봇은 주행 중에 발생할 수 있는 오차를 주행 중에 확인할 수 없어 장거리 주행을 할 경우 오차의 누적으로 인해 원하는 목적지까지 도착하지 못할 가능성이 있다. 이러한 단점과 컴퓨터 처리속도의 발전으로 영상을 고속 처리할 수 있게 됨에 따라 이미지센서를 로봇의 학습주행연구에 이용하기 시작하였다.[6] Carnegie Mellon University의 NAVLAB(NAVigation LABoratory)에서는 이미지센서를 사용하여 인간의 운전방법을 학습 할 수 있는 신경회로망을 이용하여 학습주행이 가능한 실제 자동차를 개발하였다.[7] 학습주행에 사용된 신경회로망 ALVINN(Autonomous Land Vehicle In a Neural Network)의 입력층은 차량의 카메라에서 촬영된 도로의 전체 영상을 30×32 화소 영상으로 변환하여 사용하였으며 사용한 알고리즘은 역전파 알고리즘이다. Y. C. Lim의 논문에서는 도로의 특징점을 학습데이터로 이용하여 로봇의 조향뿐만 아니라 속도제어까지 가능한 것을 보여주었다.[8] 위의 학습은 역전파 알고리즘을 사용하여 도로의 특징점과 조향각을 연결시키는 작업을 통해 이루어 졌다.

III. 주행선 인식 알고리즘

학습에 필요한 주행선의 특징점을 추출하기위해 전방의 도로를 검색할 수 있는 검색영역을 정하고 그 영역 내에서 도로의 형태를 추출하도록 하였다. 실제 학습을 위해 도로주행을 하는 경우 도로의 형태는 연속적인 주행선을 갖는 경우도 있지만 불연속적인 주행선을 갖는 경우도 발생하게 된다. 따라서 학습을 위해 도로의 특징점을 추출하거나 주행하기 위해서는 주행선이 사라지는 경우에 대해서도 학습이 가능해야 한다. 본 연구에서는 위의 두 가지 형태의 도로에서 학습이 가능하도록 아래와 같은 방법을 제시하였다. 여기서 제시하는 방법은 주행선 추출을 실시간 처리에 중점을 두어 영상의 전체 영역을 처리하지 않고 부분영상을 이용하는 것과 경계를 검지하는데 경계 인식 연산자를 사용한다는 것이다.

1. 경계 인식 연산자

주행학습에 자기구성알고리즘을 사용하기 위해서는 도로영상에서 학습 데이터를 실시간으로 추출할 수 있어야 한다. 본 연구에서는 학습데이터로 도로의 특징점을 사용하기 위해 영역검출에 사용되는 소벨연산자(Sobel operator)를 이용하였다.

2	1	0
1	0	-1
0	-1	-2

(a)

0	1	2
-1	0	1
-2	-1	0

(b)

그림 1. 소벨 연산자
Fig 1. Sobel operator

도로에서 영역 검출을 위해 그림 1 (a)를 이용해서는 영상의 왼쪽에 나타나는 차선을 찾는데 적용하고, (b)를 이용해서는 영상의 오른쪽에 나타나는 차선을 검색하는데 적용하였다.

2. 특징점 추출 방법

주행 중 학습이 가능하기 위해서는 실시간으로 입력되는 영상으로부터 도로의 주행선을 정확히 추출할 수 있어야 한다. 그러나 도로의 영상은 주위환경의 변화로 인하여 수시로 변하므로 주행선의 정확한 추출이 어렵다. 실제로

학습하면서 도로에서 주행선을 추출하다보면 많은 어려움이 존재하게 되는데, 특히 주위 조도의 변화로 인해 카메라로 들어오는 도로의 영상이 불완전하여 주행선의 추출을 어렵게 하는 경우가 그 예이다. 따라서 도로의 특징점을 조도 변화에 상관없이 추출하기 위해 본 논문에서는 경계를 결정하는 임계값을 고정시키지 않고 조도에 따라 변화가 가능하도록 하였다. 도로의 특징점 추출은 입력된 영상에서 전부 추출하는 것이 아니라 도로의 개략적인 윤곽만 알 수 있도록 몇 개의 지점만을 추출하였다. 이는 도로의 형태가 몇 개의 점으로도 충분히 그 형태를 추정할 수 있기 때문이다. 도로의 주행선 추출을 위해 전체 화면상에서 수직으로 네 지점을 선택하여 그 각각의 지점으로부터 좌측과 우측 양방향으로 탐색하여 도로를 이루는 8개의 주행선을 식별하도록 하였다. 또한 주행로의 경계를 찾는데 탐색영역을 설정하여 그 영역 내에서만 주행선을 추출하도록 함으로써 불필요한 탐색시간을 줄임은 물론 영역 내에서 잡음에 의한 도로 주행선인식 오차의 가능성을 제거하였다. 추출된 주행선은 그림 2와 같이 좌표계를 설정하여 도로 주행선의 중간 값에 해당하는 x값을 특징점 데이터로 사용한다. 도로의 주행선을 추출하는 방법을 간단히 살펴보면 다음과 같다. 여기서 y 좌표는 영상화면에서의 수직 좌표이고, x 좌표는 수평좌표를 나타낸다.

3. 주행선 인식 절차

Step 1 : 영상 화면상에 설정된 4 점의 y 좌표로 이동해 탐색영역을 설정한다.
이 때 y 좌표의 위치가 영상화면에서 아랫부분에 위치할수록 주행로 경계를 추출하는데 수월하지만 그 위치와 로봇의 거리가 너무 가깝다면 원활한 횡 방향 제어를 하기가 어렵게 된다. 본 실험에서는 중간에 1 점의 좌표를 기준으로 위쪽으로 1개, 아래쪽으로 2개의 점을 설정하였다. 설정된 y 좌표를 기준으로는 8개의 직사각형 모양의 탐색영역을 지정하였다. (그림 3 (c) 참조)

Step 2 : 도로 경계 연산자를 이용하여 주행로 경계를 인식한다.
탐색 영역 내에서 주어진 주행로의 양측 경계를 찾는 단계이다. 경계를 찾기 위해서는 주행로 경계 연산자를 좌측과 우측으로 수평이동하면서 그림 1의 소벨 연산자를 적용하여 값을 계산한 후 그 값이 경계인지 아닌지를 결정해야 한다. (그림 3 (b) 참조) 경계를 결정하기 위해서는 임계값과 비교를 해야 하는데 임계값을 고정시켜 놓고 사용하게 되면 조명과 같은 주위환경에 따라 경계인식에 많은 차이를 보일 수 있다. 따라서 도로 영상의 전체적인 명도 변화에 따라 임계값도 함께 변할 수 있도록 입력된 영상 중 검색영역내의 명도값들을 고려하여 임계값을 주기적으로 변화시켜준다. 이 때 추출된 경계들을 잠정적으로 도로의 경계점이라 가정한다. 주행로의 경계를 탐색하는 단계에서는 다음의 세 가지 경우가 나타날 수 있다.

case 1 : 양측의 경계 모두를 찾은 경우
왼쪽 경계점이나 오른쪽 경계점이 한 개 이상이 발견된 경우에는 이전에 구해진 지수 평활화된 값과 비교해 가장 가까운 지점을 주행로의 경계라고 결정한다. 좌측과 우측 경계의 지수 평활화된 값은 아래의 식 1에 의해 계산된다.

$$X_{L_{avg}}(t+1) = \alpha \times X_{L_{avg}}(t) + (1-\alpha) \times X_L(t)$$
$$X_{R_{avg}}(t+1) = \alpha \times X_{R_{avg}}(t) + (1-\alpha) \times X_R(t) \qquad (1)$$

여기서, $0 \le \alpha \le 1$,
$X_L(t)$, $X_R(t)$: 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 : 양쪽의 경계를 모두 인식하지 못한 경우
불연속적인 도로의 여건이나 기타 여러 가지 이유로 도로의 양측 경계 모두를 인식하지 못한 경우이다. 도로 경계의 추정은 과거에 구한 자료들이 사용된다.

Step 3 : 주행로의 경계를 검증한다.
주행로의 경계가 제대로 인식되었는지 또는 잡음으로 인한 잘못된 경계인식인지를 판정한다.

다음의 조건을 모두 만족한다면 주행로의 경계를 정확히 인식한 것으로 간주한다. (그림 2 참조)
1. 구해진 주행로 경계의 폭을 주어진 임계값과 비교하고 일정 범위 내에 있는지를 검토
2. $D_1 > D_2 > D_3 > D_4$
3. $C_1 \le C_2 \le C_3 \le C_4$ 또는 $C_1 > C_2 > C_3 > C_4$

여기서, D_i : 각각의 y 좌표내에서 도로 경계의 폭
C_i : 각각의 y 좌표내에서 추출한 도로 경계 중심의 x 좌표

Step 4 : Step 1로 간다.

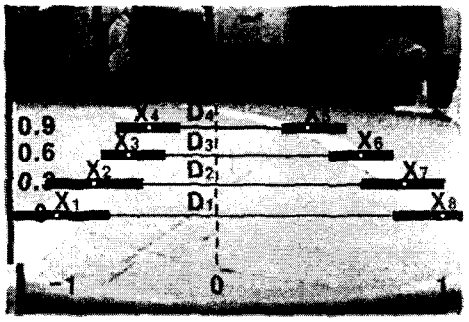
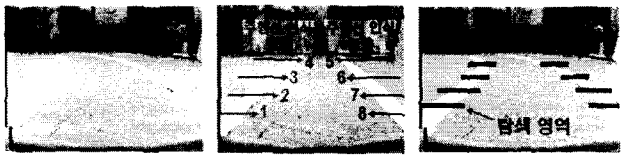


그림 2. 주행선 좌표
Fig 2. Coordinates of Driving Lane

4. 주행선 인식 결과

그림 3은 위의 주행선 인식 방법을 이용하여 로봇이 주행로의 경계를 추출한 결과 화면이다.



(a) 입력 영상 (b) 주행선 인식방향 (c) 영상처리 결과

그림 3. 입력 영상의 특징점 추출
Fig 3. Extracting feature points of input image

5. 자기 구성 신경망(Self Organization Method)을 이용한 학습

카메라와 로봇의 기하학적인 관계를 이용하면 영상으로부터 도로의 곡률, 로봇의 위치, 로봇의 진행방향을 실시간으로 구하는 것이 가능하다. 하지만 기하학적인 관계를 이용하는 방법은 모델링을 위해 로봇에 장착된 카메라의 위치와 카메라의 초점거리등을 포함한 많은 요소들을 고려해야할 뿐만 아니라 이들 요소를 조율(*tuning*)하는데 많은 시간을 필요로 하게 된다. 따라서 본 연구에서는 이러한 모델링 과정을 거치지 않고 횡 방향 제어를 하기 위하여 인공신경망의 일종인 경쟁네트워크(competitive network) 알고리즘인 SOM을 사용하였다.[9][10] 이 방법은 기존에 많이 사용된 역전파(back-propagation) 알고리즘보다 학습을 위한 시간을 절약할 수 있고, 새로운 환경을 학습시킬 경우 과거에 학습된 내용을 잃어버리지 않고 곧바로 추가 학습이 가능한 장점이 있다. 또한 알고리즘을 구현하기가 용이하고 학습 직후 곧바로 자율주행을 할 수 있는 신속성의 특징이 있다. 여기서 출력층을 이루는 출력뉴런의 수는 고정되어 있는 것이 아니라 입력층을 형성하는 입력벡터에 따라서 증가하거나 감소할 수 있게 하였다.

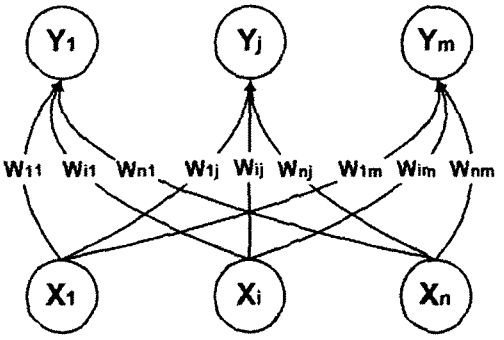


그림 4. 자기 구성 지도 구조
Fig 4. Overview of self_organizing map

제안된 모델의 입력층은 앞 절에서 구한 주행로경계 특징점의 위치들로 구성되어 있다. 입력된 벡터들은 경쟁층에 있는 모든 뉴런사이에서 경쟁하여 가장 적합한 출력뉴런을 찾아 그에 해당하는 조향각도를 조향제어장치에 보내어 횡 방향제어를 하게 된다. 본 연구에서는 학습을 위해 경쟁층에 있는 뉴런의 수를 미리 지정하지 않고 동적인 특성을 갖도록 하여 상황에 맞게 적절히 생성되도록 하였다. 경쟁층에 있는 뉴런의 수를 고정된 값을 사용하지 않고 동적인 값을 사용한 것은 동일한 조향각도를 갖는다고 입력벡터가 항상 동일하지는 않기 때문이다. 그 예로 현재 얻어진 도로경계의 위치와 무관하게 사람이 앞으로의 방향을 미리 예측하여 방향을 제어함으로써 비록 도로 경계들의 위치는 틀리더라도 같은 조향각도로 주행 할 수 있는 것을 들 수 있다. 로봇이 주어진 경로의 주행을 위해 환경에 적절한 조향 각도를 산출하고, 횡 방향 제어를 하기 위한 절차는 그림 5와 같다.

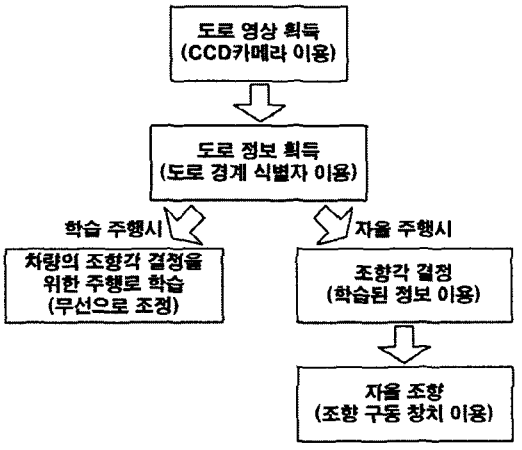


그림 5. 횡 방향 제어를 위한 절차
Fig 5. A block diagram of the system

6. 주행로 학습을 위한 절차

로봇이 주행로를 주행하기 위해서는 주행로에 대한 학습을 필요로 하게 된다. 주어진 주행로를 학습하기 위한

절차는 다음과 같다.

Step 1 : 신경망의 입력 층에 입력 벡터를 제시한다.
입력벡터는 주행선 추출알고리즘으로부터 얻어진 도로의 특징점($X_1, X_2, ..., X_8$)으로 구성된다.

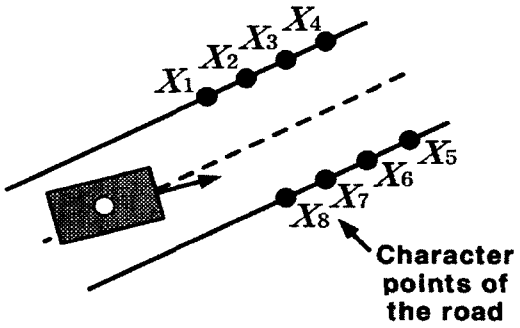


그림 6. 학습에 사용되는 입력 변수
Fig 6. An input variance for learning

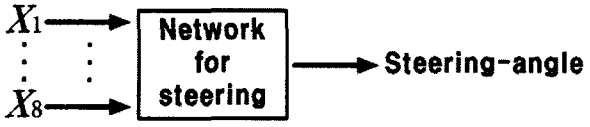


그림 7. 횡 방향 제어를 위한 입력 관계
Fig 7. An input connection for steering control

Step 2 : 입력 벡터와 모든 뉴런들과의 거리를 계산한다.
입력과 출력 뉴런 j 사이의 거리 d_j는 다음 식(2)와 같이 계산된다.

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

여기서, $X_i(t)$는 t 시점에서의 i 번째 입력 벡터의 원소이고, $W_{ij}(t)$는 t 시점에서의 i 번째 입력 벡터의 원소와 j 번째 출력 뉴런 사이의 연결강도이다. N 은 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 $i = 1,2,3,4$
j : 새로 생성된 뉴런
③ Step 1 로 간다.
2. 그렇지 않은 경우 Step 4 로 간다.

Step 4 : 뉴런 j^*의 연결강도를 다시 조정한다.
먼저, 뉴런 j^*의 class 와 입력벡터의 class를 비교한 뒤 식(3), 식(4)를 이용하여 연결강도를 재조정한다. 여기서 class는 조향각을 나타낸다.
1. 동일한 class를 가지는 경우

$$W_{ij}(t+1) = W_{ij}(t) + \alpha(t)[X_i(t) - W_{ij}(t)] \quad (3)$$

$$0 \le \alpha(t) \le 1$$

2. 다른 class를 가지는 경우

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

$$0 \le \alpha(t) \le 1$$

그리고 Step 1 로 간다.

위 절차에서 $\alpha(t)$와 새로운 뉴런을 생성하기 위해 그 기준이 되는 임계값의 결정은 중요한 요소가 된다. 임계값을 크게 설정하면 할수록 생성뉴런의 수가 적고 처리시간은 짧아지는 반면, 로봇의 조향각을 결정하는데 분별력을 약화시켜 보다 많은 오차를 야기하게 된다. 그와 반대로 임계값을 낮추게 되면 생성 뉴런의 수가 증가하여 임계값이 높은 경우보다 정밀한 제어를 할 수 있지만 불필요하게 많은 뉴런을 생성하여 실시간 처리를 저해할 수 있다. 이의 결정은 실험을 통해서 오차와 처리시간을 고려하여 α의 값을 0.1로 하였고, 그 임계값은 10으로 하여 모델링 하였다.

7. 학습 후 주행 시 조향각 결정

로봇이 주행하고자 하는 주행도로에서 먼저 학습시킨 후 학습시킨 자료들을 이용하여 주행로를 주행하기 위한 조향각 산출방법은 다음과 같다.

Step 1 : 영상으로부터 추출된 주행로 경계로부터 8개의 좌표를 계산해 인공신경망에 제시한다.
Step 2 : 신경망의 입력 층에 입력된 값들과 모든 뉴런간의 거리를 계산한다.
Step 3 : 최소 거리에 있는 출력 뉴런을 선택하고, 그에 해당하는 조향각을 찾는다.

8. 자율이동로봇의 운동

연구에서 사용한 로봇의 회전 운동은 Servo motor 2개를 사용하는 마이크로 마우스와 달리 일반 차량의 회전 방식과 유사 하다.[11] 따라서 실험에 사용한 로봇은 조향을 위해 모터의 회전 속도를 고려해 줄 필요가 없이 간단히 조향 모터의 방향만 전환시킬 수 있는 장점이 있다. 실제로는 원심력도 작용하고 차량이 일정한 반지름을 가지고 선회하기 위해 원심력에 평행되는 힘인 코너링 포스(cornering force)도 작용하게 되지만 실험에서 사용한 로봇의 회전운동은 저속에서 일어나므로 코너링 포스 같은 외부 힘은 무시하였다. 일반적인 이동로봇의 회전 궤적은 그림 8과 같다.[12]

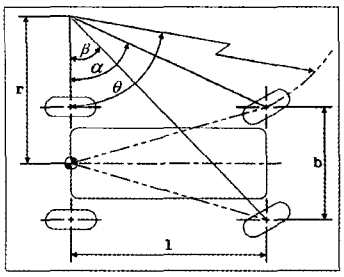


그림 8. 이동로봇의 회전 궤적
Fig 8. Wheel track of driving robot

Ⅳ. 실 험

1. 시뮬레이션

본 연구 알고리즘의 타당성을 검증하기 위해 고정된 속도에서 조향에 의한 주행제어가 가능한 프로그램을 작성한 후 컴퓨터에서 시뮬레이션을 하였다. 그림 9와 같이 입력된 길을 사람이 키보드를 사용하여 학습한 후 자율 주행 버튼을 누르면 그전까지의 학습한 데이터를 바탕으로 자율주행을 시작하게 된다.

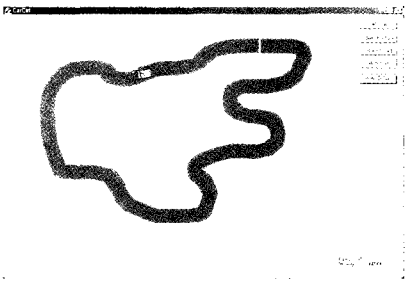


그림 9. 자율 주행 시뮬레이션
Fig 9. A simulation test for autonomous driving

그림 10은 주어진 길을 자기 구성 알고리즘을 사용해 주행로 학습을 시킨 후 로봇을 자율 주행시킨 결과이다.

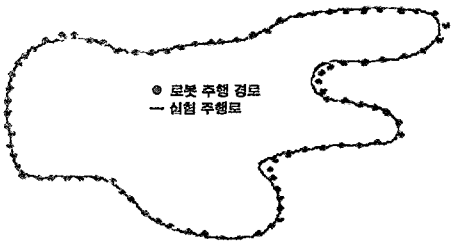
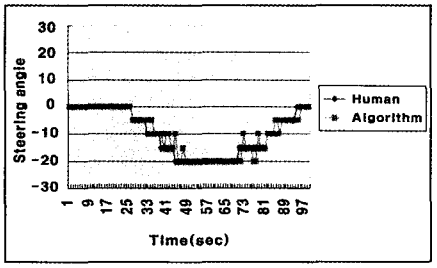
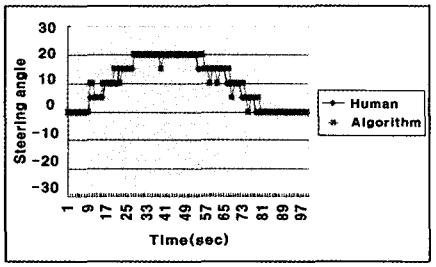


그림 10. 시뮬레이션 결과
Fig 10. A result of simulation test

그림 10에서 볼 수 있듯이 급격한 회전이 존재 하는 부분에서는 도로 중심에서 조금 벗어난 움직임을 보였으나 직선이나 완만한 굴곡에서는 큰 오차 없이 동작하였다. 도로중심을 벗어난 움직임이라고 해도 도로를 이탈하지 않고 조향동작을 하였기 때문에 도로의 특징점을 이용한 자기구성 알고리즘 학습이 가능하다는 것을 알 수 있었다. 주어진 경로를 주행 중 실제 운전자와 구현된 알고리즘에서 구한 조향각도를 비교한 결과를 그림 11에서 보여준다. 그림 11 에서 (a)는 차량이 직선에서 좌측으로 주행하는 경우이고, (b)는 우측으로 주행하는 경우를 나타낸다. 운전자와 구현한 알고리즘에 의해 계산된 핸들의 각도를 비교해본 결과 그 오차의 평균은 왼쪽 0.95°, 오른쪽 1°이고 표준편차는 왼쪽 2.09°, 오른쪽 2° 였다. 운전자의 운전 형태는 곡선도로에서나 직선도로에 관계없이 부드러운 반면 학습에 의한 주행은 다소 변화가 있음을 알 수 있었다.



(a) 좌측 주행



(b) 우측 주행

그림 11. 조향 각도 비교
Fig 11. A comparison of steering angle

2. 시스템 구성

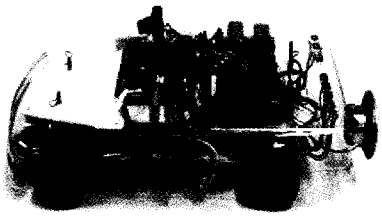


그림 12. 실험에 사용된 이동 로봇
Fig 12. Experimental driving robot

본 시스템에서는 컴퓨터 대신 고속연산이 가능한 DSP를 사용하였다. 학습 로봇의 구성은 크게 주제어부와 부제어부로 나누어진다. 주제어부에서는 Texas Instrument사의 DSP 칩인 TMS320C6711을 이용하여 영상처리와 인공지능 알고리즘에 의한 학습이 가능하도록 구성하였다. 영상처리는 CCD카메라로부터 NTSC 신호를 받아, 그 신호를 Digital 신호로 변환하여 이루어 졌다. 영상처리 능력은

회색음영으로 360*240의 크기를 영상처리 할 경우 15프레임의 속도로 처리하였다. 영상처리에 의한 입력 데이터 외에 로봇의 조향데이터를 얻기 위해 본 연구에서는 Potential meter를 사용하였다. Potential meter의 값을 입력받기 위해 DSP에 A/D 변환 기능을 가진 ADG508칩을 사용하였으며 무선 조정 시에는 특징점에 따른 입력데이터로 사용하고 학습 후 횡 방향제어 시에는 로봇의 현재 진행방향으로 사용하였다. 학습이 완료된 데이터는 삼성전자의 CMOS SD RAM인 K4S641632D-TC/L60에 저장하였다. 제어나 데이터 교환을 위한 2개의 시리얼 통신이 가능하도록 설계하여 하나는 현재 진행 상황을 서버 컴퓨터로 보내는 RF Module 통신부와 연결되어 있고 나머지 하나는 동작을 제어하는 부제어부와 연결하는데 사용하였다.

부제어부는 Intel사의 16bit 프로세서인 80C196KC를 사용하여 DSP에서 지시한 명령을 시리얼 통신으로 받아들여 로봇의 움직임을 제어할 수 있도록 구성하였다. 동작제어부는 무선으로 로봇을 조정하여 데이터 습득 시에는 사용되지 않으며 학습에 의한 횡 방향제어 시에만 사용된다. 학습에 의한 횡 방향제어 시에는 RC Motor(Radio Controlled Servo Motor)의 조향 각 분해능력을 -30°~30° 사이로 왼쪽과 오른쪽 각각 5°씩 13개가 되도록 설정하여 정밀하면서 신속한 조향을 시도 하였다.

연산을 담당하는 로봇과 컴퓨터와의 무선 통신이 가능하도록 본 시스템에서는 양방향 통신이 가능한 RF Module인 Radiometrix사의 Bim418F를 사용하였다. 자기구성지도 알고리즘의 상황감시용으로 사용된 컴퓨터의 사양은 펜티엄 Ⅲ 933Mhz이며 사용 OS와 Language는 windows98과 C++이다.

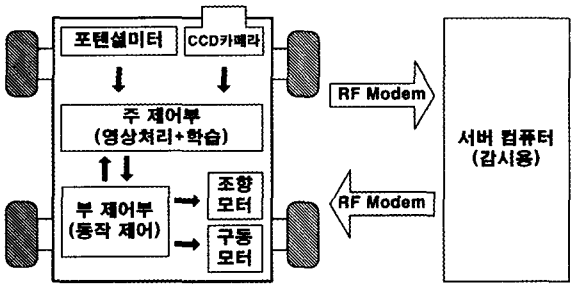


그림 13. 전체 시스템 구성도

Fig 13. A block diagram of full system

3. 실험

본 실험에서는 도로의 곡률에 의해 속도를 제어하는 부분에서 주행 도로의 형태와 제어 로봇의 빠른 속도 때문에 속도의 단계를 산출하는 것이 어려워 일정한 속도에서 조향각도를 산출하는 것만을 실험하였다. 로봇의 속도는 0 또는 0.3m/s로 제한하여 주행 중에 기민한 제어를 하지 못해 발생하는 차선이탈을 방지하였다. 실험의 목적인 학습 알고리즘에 의한 로봇의 횡 방향제어를 구현하기 위하여 차선의 폭이 50~100m^{-3} 트랙의 폭 250~300m^{-3} 총길이가 20m인 트랙을 준비하였다. 실험은 사람이 무선 조종으로 로봇을 움직여서 로봇에게 횡 방향제어 학습을 시킨 후 학습된 로봇이 트랙을 얼마나 정확히 돌 수 있는지를 판단하였다. 로봇을 학습 시킨 횟수는 총 10회이며 각 횟수마다 트랙을 60번 횡 방향제어 주행을 시켜 각 횟수에 따른 로봇의 차선이탈 여부를 판단하였다. 아래 표 1은 10번의 학습 주행에서 각도에 따라 생성된 뉴런의 수를 나타낸 것이다. 생성된 뉴런의 개수는 실험에 따라 차이는 있지만 90~97 사이에서 각각 생성되어 개수에서는 큰 차이가 없었다.

표 1. 생성된 뉴런의 수

Table 1. The quantities of generating neuron

각도		-30°	-25°	-20°	-15°	-10°	-5°	0°	5°	10°	15°	20°	25°	30°
실험 1	생성된 뉴런 수	9	3	12	6	2	3	18	2	3	8	14	3	7
실험 2		4	7	15	8	6	8	11	8	7	5	11	2	0
실험 3		11	1	9	8	6	4	14	5	6	7	5	12	9
실험 4		15	5	7	4	5	7	16	5	8	1	13	5	4
실험 5		8	3	12	3	8	6	16	6	8	3	11	6	6
실험 6		5	8	11	1	4	5	18	8	4	6	16	4	7
실험 7		3	13	9	6	3	4	18	7	6	2	15	3	8
실험 8		12	8	7	5	3	6	16	6	1	2	9	8	13
실험 9		7	6	9	17	2	5	9	4	14	3	8	4	7
실험10		6	9	8	6	6	3	14	3	6	4	12	9	4

각 실험에서 조향을 어떤 방식으로 진행했느냐에 따라 각도에 대한 뉴런의 수가 차이는 있었지만 표 1에서 보듯이 각도에 따라 생성된 뉴런의 수가 각도 0°를 기준으로 좌우에 하나씩 크게 세 분류로 모이는 것을 알 수 있었다. 이는 실험에서 사용하는 조향 각도가 사람마다 다르기는 하지만 주로 사용하는 각도는 직진, 왼쪽 오른쪽 크게 세 가지로 분류 된다는 것을 보여주고 있다. 표 2는 각 실험에서의 자율 주행 성공률을 보여주고 있다.

표 2. 실험에 따른 성공 횟수

Table 2. The frequency of success per test

	횟수	성공 횟수
실험 1	60번	55
실험 2	60번	51
실험 3	60번	55
실험 4	60번	45
실험 5	60번	51
실험 6	60번	55
실험 7	60번	55
실험 8	60번	48
실험 9	60번	47
실험10	60번	50

자기 구성 신경망 학습에 의한 자율 주행 성공률은 60회 중 평균적으로 51.20번을 성공함으로써 85%의 성공률을 보였다. 본 실험을 통해 논문에서 제시한 학습알고리즘이 가능한 것을 알 수 있었다.

V. 결론 및 향후 연구계획

본 논문에서는 인공지능 알고리즘의 일종인 자기구성 알고리즘을 사용하여 횡 방향 제어가 가능한 학습 알고리즘을 제안 하였다. 자기구성 알고리즘의 특성상 학습을 위한 시간을 절약할 수 있었으며 추가 학습이 필요할 경우 과거에 학습된 내용을 잃어버리지 않고 곧바로 추가학습이 가능하였다. 또한 주위 조도가 변하거나 주행선이 짧은

시간 사라지는 경우에도 도로의 특징점을 추출하여 학습하거나 주행하는데 문제가 발생하지 않았다. 하지만 본 연구에서는 속도 제어를 고려하지 않아 일정한 속도에서 학습하고 제어하는 형태로 시스템을 구현하였다. 향후 연구에서는 속도 제어가 가능하도록 실험 모델을 개선하여 속도를 고려한 실험을 하도록 한다.

참고 문헌

[1] Seung-Hak Yang, Fumio Harashima Young-Cheol Lim, Kyeung-Young Cho, Young-Jae Ryoo,"Steering control system for autonomous road vehicle using fuzzy neural network," Trans IEE of Japan, Vol. 115D, No. 11, pp.1343-1379, Nov. 1995

[2] LABMATE User Manual Version 5.21L-f.

[3] Floreano,D. and Mondada, F. 1996 "Evolution of homing navigation in real mobile robot", IEEE Transactions on System, Man and Cybernetics, Vol. 26, No. 3 ,pp. 396-407, Jun. 1996

[4] Oussama Khatib, "Real-time Obstacle Avoidance for Manipulators and Mobile Robots", Intl. J. of Robotics Research, Vol. 5, No. 1, pp. 90-98, spring, 1986.

[5] 이광주 "신경망 학습 및 유전자 프로그램의 진화를 이용한 자율이동로봇의 온라인 행동 학습" 서울대학교 석사논문, 2000

[6] Tomohiro Shibata, Yoshio Matsumoto, Taichi Kuwahara, Masayuki Inaba, Hirochika inoue, "Hyper Scooter : a Mobile Robot Sharing Visual Information with a human", Proceedings of the 1995 IEEE International Conference on Robotics and Automation,pp. ICRA'95, 1074-1079 ,1995

[7] Thorpe, Charles E., "Neural Network Perception for Mobile Robot Guidance Vision and Navigation" - The CMU Navlab, Boston: Kluwer Academic Publishers, U.S.A., 1990.

[8] 임영철 "신경회로망을 이용한 자율주행차량의 속도 및 조향제어" 한국퍼지 및 지능시스템학회 1998년도 추계학술대회 학술발표논문집 pp.274-281 1225-1127

[9] S. Haykin, "Neural Networks: A Comprehensive Foundation", Prentice Hall, 1999

[10] T.Kohnen, "Self-Organization and Associative Memory" Springer-Verlag, New York, NY, 1984

[11] Jean-Claude Latombe."Robot Motion Planing" Kluwer Academic Publishers, pp. 26-29, 1991

[12] 원윤재 "영상기반 자율이동로봇의 주행제어기술 개발" 고려대학교 석사논문, 2001

김 영 욱(Young-Wook Kim)
準會員
2002년 한국외국어대학교 산업공학과졸업
2002~현재 고려대학교 산업시스템정보공학과 석사과정

관심분야 : 이미지 프로세싱, 로보틱스, 뉴럴 네트웍,

김 종 철(Jong-Chul Kim)
準會員
2002년 중앙대학교 기계공학과 졸업
2002~현재 고려대학교 산업시스템정보공학과 석사과정

관심분야 : 임베디드 시스템, 무인 자동차, 이미지 프로세싱

이 경 복(Kyong-Bok Lee)
準會員
2001년 남서울대학교 산업공학과 졸업
2003년 고려대학교 산업공학 석사
2003~현재 고려대학교 산업시스템정보공학과 박사과정

관심분야 : 무인자동차, 로보틱스, 뉴럴 네트웍, 임베디드 시스템, DSP

한 민 홍(Min-Hong Han)
正會員
1964년 서울대학교 기계공학과 졸업
1973년 University of Minnesota 산업공학 석사
1981년~1982년 숭실대학교 산업공학과 조교수
1985년 Georgia Institute of Tech 산업공학과 박사
1985년~1988년 Texas A&M University 산업공학과 조교수
1988년~1990년 포항공과대학 산업공학과 부교수
1991~현재 고려대학교 산업시스템정보공학과 교수
관심분야 : 지능형 자동차, 지능 로봇, ITS, 컴퓨터비전