論文98-35C-11-9

강화 학습을 이용한 자율주행 차량의 횡 방향 제어

(Lateral Control of An Autonomous Vehicle Using Reinforcement Learning)

李 政 勳 *, 吳 世 泳 *, 崔 斗 鉉 **

(Jeong-Hoon Lee, Se-Young Oh, and Doo-Hyun Choi)

요 약

강화 학습은 이산적인 공간을 가상하여 많은 연구가 행해졌지만, 많은 실제적인 제어 문제는 연속적인 공간에서 이루어진다. 평가 함수와 행동 함수를 연속함수로 하면 강화 학습 구조를 연속 공간에서 사용할 수 있다. 그러나 이 경우 두가지 고려해야 할 점이 있다. 하나는 어떤 종류의 함수 표현 법을 사용할 것인가 하는 문제고, 다른 하나는 첨가하는 잡음의 양을 결정하는 것이다. 평가 함수와 정책 함수(제어기)에는 신경회로를 사용하였다. 강화 예측기로 다음 순간의 강화 신호를 예측하고, 아울러 첨가하는 잡음의 양도 결정하였다. 제안된 강화 학습 구조를 사용하여 차량의 횡 방향 제어 모의 실험에서 온라인 학습의 특성을 확인하였다. 제안된 구조를 실차 실험에도 적용하여 유용성과 타당성을 검증하였다.

Abstract

While most of the research on reinforcement learning assumed a discrete control space, many of the real world control problems need to have continuous output. This can be achieved by using continuous mapping functions for the value and action functions of the reinforcement learning architecture. Two questions arise here however. One is what sort of function representation to use and the other is how to determine the amount of noise for search in action space. The ubiquitous neural network is used here to learn the value and policy functions. Next, the reinforcement predictor that is intended to predict the next reinforcement is introduced that also determines the amount of noise to add to the controller output. The proposed reinforcement learning architecture is found to have a sound on-line learning control performance especially at high-speed road following of high curvature road. Both computer simulation and actual experiments on a test vehicle have been performed and their efficiency and effectiveness has been verified.

Ⅰ. 서 론

인간의 노동을 대신하는 자동화는 공장 뿐만 아니라 가정에까지 파고 들어 많은 자동화 기술과 제품을 개발/생산하고 있다. 최근 자동화가 시도되고 있는 재미있는 분야 중에 자동차 도로 주행의 자동화 즉, 자율 주행 차량의 개발이 있다. 이 자율 주행 차량의 개발은 인간을 운전이라는 단순 노동에서 해방시켜 줄 뿐만 아니라 인간의 부주의나 실수로 인한 사고를 방지하여 인간의 생명과 재산을 보호해 줄 수 있게 된다.

횡 방향 제어 시스템에 대한 연구는 세계 각국에서 여러 해 동안 많은 연구가 이루어졌다. 카네기 멜론 대학에서는 신경 회로를 이용한 ALVINN[1] 과 컴퓨터 비전을 이용한 RALPH[2]를 개발하였다. 버클리

* 正會員, 浦港工科大學校 電子電氣工學科
(Dept. of Electronic and Electrical Engineering, Pohang University)
** 正會員, 慶北大學校 電子電氣工學部
(School of Electronic and Electrical Engineering, Kyungpook National University)

接受日字:1998年4月20日, 수정완료일:1998年10月21日

대학의 Partners for Advanced Transit and Highways(PATH)센터는 영상 센서를 사용한 CMU와는 달리 도로상의 자기(magnetic)정보를 이용한 자율주행 시스템을 개발하였다[3]. 유럽에서도 독일 뮌헨대학에서 입력 센서로 스테레오 카메라와 칼만 필터링(Kalman filtering)기법을 사용하여 차량의 자율주행 시스템을 개발하였다[4]. 국내에서도 몇몇 대학과 자동차 회사를 중심으로 이에 대한 연구가 진행되고 있다[5]. 그러나, 위에 열거한 시스템들은 환경이 급박하게 변하거나 차량의 동역학 모델이 변화하였을 경우 다시 새롭게 제어기의 인자들을 바꾸어 주어야 한다. 일반적으로 강화 학습(Reinforcement Learning; RL)은 사람이 학습에 관여하지 않아도, 제어기의 행위에 대한 환경으로부터의 평가신호를 이용하여 올바른 제어 신호를 제어기가 스스로 배우게 되고 환경이 변하더라도 사람처럼 제어기가 스스로 적응해 간다. 그러므로 다양한 모델과 환경이 존재하는 자동차의 횡방향 제어 알고리듬에 적용할 수 있다. 본 논문에서는 시간과 환경에 따라 변하는 차량의 동역학 모델에 적응할 수 있는 새로운 제어 구조를 제안한다. 제안된 제어기는 신경회로로 구성되어 있고, 강화 학습을 이용한다. 신경망을 사용할 경우 학습이라는 과정이 필요한데, Yu[6]는 외부에서 별도로 학습 신호를 제시할 필요도 없고, 연속적인 공간에서 적용할 수 있는 RL 알고리듬을 제안하였다.

강화 학습은 크게 두개의 구성요소가 있다. 하나는 행위자(agent)이고 다른 하나는 외계(environment)이다. 행위자는 외계에서 주어진 상태에 따라 적절한 행위(action)를 외계에 행하며, 외계는 주어진 행위에 따라 변화된 상태와 행위가 적절했는가에 대한 판단인 강화 신호를 행위자에게 보내는데, 이와 같은 과정을 반복하며 학습이 이루어지게 된다. 여기서 중요한 것이 외계의 상태와 행위자의 행위간의 사상이다. 행위자는 평가 함수(value function)와 정책 함수(policy function)를 가지고 있다. 행위자는 평가 함수를 통하여 상태에 따른 정책 함수를 평가하여 정책을 변경하고 또한 외계에서 주어진 강화 신호에 따라 상태의 평가 함수를 적절하게 다시 변경하여야 한다. 평가 함수의 학습은 다음과 같은 TD(Temporal Difference) 방법을 사용하게 된다[7].

$$TD(\lambda):\quad \Delta w_t = \alpha(V_{t+1}-V_t)\sum_{k=1}^{t}\lambda^{t-k}\nabla_w V_k \qquad (1)$$

$$TD(1):\quad \Delta w_t = \alpha(V_{t+1}-V_t)\sum_{k=1}^{t}\nabla_w V_k \qquad (2)$$

$$TD(0):\quad \Delta w_t = \alpha(V_{t+1}-V_t)\nabla_w V_k \qquad (3)$$

여기서 α와 λ는 상수이다.

강화 학습은 초기에 Barto[8]에 의해 Pole-balancing 문제에 적용된 후 많은 문제에 적용되고 있다. 하지만 대부분이 제어 대상의 상태나 제어기의 출력을 이산적으로 표현하였다. 그러나 이러한 방법은 출력으로 연속적인 제어 신호를 필요로 하는 제어 문제에 쓰일 수 없고 연속 공간(continuous space)을 이산적으로 만들어서 사용하더라도 미세한 레벨의 제어가 필요한 곳에는 적용되지 못하는 단점이 있다. 또한 최적해를 탐색하기 위해서 제어기의 출력에 잡음(noise)을 추가하게 되는데 이 추가되는 잡음이 시스템의 성능에 대한 함수가 아니라 일정한 상수 값의 확률 밀도 함수를 가지고 있다. 따라서 최적해를 찾은 후에도 항상 잡음이 추가되어 제어 대상이 우리가 원하는 목표에 수렴하지 않고 진동하게 된다. 이와는 달리 Gullapalli[9]는 신경 회로를 이용하여 제어대상 및 제어기의 출력이 연속적인 공간일 경우에 대해 적용하였다. 그러나 그는 기존의 강화 학습 방법이 장기적인 기대치를 최대화하는 방향으로 학습을 시키는 것과는 달리 순간적인 강화 신호를 최대화하는 방향으로 제어기를 학습시켰다. 따라서 제어기가 장기적으로 좋은 안정적인 방법보다는 순간의 강화 신호를 최대화하기 위해서 불안하게 학습이 되거나 성공적인 학습이 어렵게 된다. 그러므로, 차량의 횡 방향 제어에 이용하기 위해서는 연속 공간에 대해 강화 학습을 적용하되 순간적인 강화 신호가 아닌 장기적인 기대치를 최대화하도록 학습시켜야 하고, 원하는 목표 값에 수렴하기 위해서 첨가되는 잡음을 시스템의 성능에 대한 함수로써 자동적으로 조절하여야 한다.

본 논문의 구성은 다음과 같다. 제2장에서 연속 공간에서 사용 가능한 새로운 강화 학습 구조를 제안하고 널리 알려진 Ball-and-Beam 시스템의 제어를 통해 제안된 구조의 타당성을 검증한다. 제 3장과 4장에서는 실제 차량을 이용하여 강화 학습 구조의 타당성을 시험한다. 제 3장에서는 실제 차량의 제어 시스템의 구조와 차량의 횡적 제어를 위해 선행되어야 하는 영상처리 알고리듬에 대해 설명한다. 제 4장에서 컴퓨터 모의 실험과 실차 실험을 통하여 제안된 강화 학습

제어기를 이용한 성능을 시험하고, 마지막으로 제 5 장에서 결론 및 향후 연구에 대해서 설명한다.

Ⅱ. 제안된 강화 학습 구조

1. 제안된 강화 학습 구조

그림 1은 장기적 효용인 가치 예측기(Value Predictor)와 단기적 효용(short-term utility)인 강화 신호 예측기(Reinforcement Predictor)를 가진 강화 학습 구조이다. 여기서 단기적 효용인 예측된 강화 신호는 잡음의 크기를 조절하는 역할을 하고, 장기적 효용인 가치는 제어기의 학습에 이용하게 된다. 단기적 효용을 이용하여 잡음의 크기를 조절하는 이유는 예측된 강화 신호를 사용하여 제어기가 학습해야 할 기대치의 범위를 알 수 있기 때문이다.

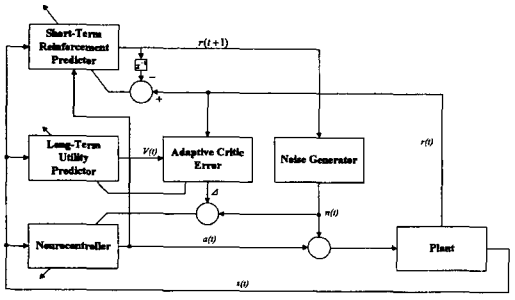


그림 1. 제안된 강화 학습 구조
Fig. 1. Proposed reinforcement learning architecture.

강화 신호 예측기는 현재의 상태 $s(t)$와 제어기에서 나온 $a(t)$를 입력으로 하여 다음 순간의 강화 신호를 예측한다. 이 예측된 강화 신호를 이용하여 잡음 생성기(Noise Generator)는 예측된 강화 신호가 높으면 작은 잡음을, 예측된 강화 신호가 낮으면 큰 잡음을 출력하도록 구성된다. 강화 신호 예측기는 플랜트의 상태와 그 상태에서 플랜트에 실제로 전해진 제어 신호, $n(t)+a(t)$가 주어졌을 때 실제로 플랜트가 내는 $r(t+1)$을 목표값이 되도록 학습시킨다. 나머지 기대값 예측기, 제어기 그리고 평가 모듈(Critic)은 연속적인 값을 출력할 수 있도록 연속적인 사상 함수인 신경 회로를 이용하였고 학습을 할 때는 TD 학습을 사용한 Barto의 방법을 사용하였다.

그림 2에 본 논문에 적용된 강화 학습의 알고리듬을 나타내었다.

Algorithm

Given state, s(0):

① Calculate $V(0)$, $a(0)$.
② Calculate the predicted reinforcement $r(1)$ using $s(0)$ and $a(0)$.
③ Calculate $n(0)$ using $r(1)$.
④ Control the plant using $a(0)+n(0)$.

Repeat:

⑤ Learn Reinforcement Predictor using error as $r(t) - (t)$.
⑥ Calculate AHC error:
$\Delta := r(t) + \gamma V(t) - V(t-1)$.
⑦ Learn Value Predictor using AHC error.
⑧ Learn Controller using $\Delta \cdot n(t)$ as error.
⑨ Calculate $a(t)$.
⑩ Calculate $(t+1)$ using $s(t)$ and $a(t)$.
⑪ Calculate $n(t)$ using $(t+1)$.
⑫ Control the plant using $a(t)+n(t)$.

End Repeat.

그림 2. 제안된 강화 학습 알고리듬
Fig. 2. Proposed reinforcement learning algorithm.

①-④번까지는 시작 상태 $s(0)$가 주어졌을 때 제어기와 잡음 생성기로부터 각각 제어 행위 $a(0)$와 잡음 $n(0)$을 플랜트에 적용한다. 여기서 잡음 $n(0)$은 시스템의 성능 정도를 나타내는 예측된 강화 신호, $r(1)$을 이용하여 결정한다. ⑤번부터 마지막까지는 매 시간 스텝 t마다 반복되어진다. ⑤번은 단기적 효용인 강화 신호 예측기를 학습시키는 과정으로 실제로 받은 강화 신호 값을 원하는 값으로 하여 학습시킨다. Gullapalli는 강화 신호를 예측하기 위하여 상태 $s(t)$만을 입력으로 하였다. 그러나 같은 상태이더라도 플랜트에 들어가는 제어값에 의해서 강화 신호가 달라질 수 있으므로 학습이 더 잘되기 위해서 여기서는 플랜트에 들어가는 제어값도 강화 신호 예측기의 입력으로 추가하였다. ⑥번은 기대값 함수와 제어기를 학습할 때 쓰이는 AHC error를 TD 학습을 이용하여 구하는 것이다. 시간 t에서 예측된 기대값 $V(t)$와 다음 시간 스텝, $t+1$에서 실제로 얻어진 강화 신호, $r(t+1)+\gamma V(t+1)$의 차이를 구한다. ⑦번에서는 기대값 예측기를 학습시키는 과정으로 ⑥에서 계산된 AHC error를 이용하여 학습시킨다. ⑧에서 $\Delta \cdot n(t)$를 오차로 사용하여 제어기를 학습시킨다. 즉, 기대값 예측기가 어느 정도 학습된 후, AHC error Δ가 양수라면 이것은 첨가된 잡음 $n(t)$에 의해 기존의 제어기가 내던 $a(t)$에 의한 것

보다 좋은 결과를 냈다는 것을 의미한다. 따라서 제어기는 첨가된 $n(t)$ 만큼을 더 내도록 학습이 된다. 여기서 Δ가 큰 값이라면 $n(t)$에 의한 시스템의 성능 향상도가 높은 것이므로 제어기가 $n(t)$를 더 잘 쫓아 갈 수 있도록 가중이 된다. 반대로 가 음수라면 첨가된 잡음에 의해서 기존의 제어기가 내던 $a(t)$보다 시스템의 성능이 나빠진 것이므로 제어기는 첨가된 잡음 $n(t)$ 만큼 값을 줄여주게 학습이 된다. Δ가 양수일 때와 마찬가지로 Δ가 절대값이 큰 음수의 값이면 시스템의 성능이 크게 나빠짐을 의미하므로 잡음 $n(t)$의 부호의 반대 방향으로 제어기의 출력을 강하게 이끌어 주는 성질을 갖는다. 마찬가지로 Δ가 절대값이 작은 음수의 값일 때에는 첨가된 잡음 n(t)에 의해서 시스템의 성능이 크게 나빠진 것이 아니기 때문에 제어기의 출력 $a(t)$를 $-n(t)$방향으로 크게 이끌어 주지 않게 된다. ⑨번에서부터 마지막까지는 앞의 ①-④와 같이 계산된다.

2. 제안된 강화 학습 구조를 이용한 모의 실험

그림 3은 제안된 알고리듬을 실험하기 위해 널리 알려진 불안정 시스템인 Ball-and-Beam 시스템을 나타내었다.

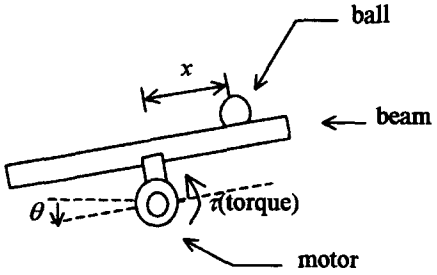


그림 3. Ball-and-beam 시스템
Fig. 3. Ball-and-beam system.

여기서 사용된 변수들은 다음과 같다.

M = 0.05 kg

J_b = 2 × 10-6 kg · m²

R = 0.01 m

G = 9.81 m/s² .

이 시스템은 빔(Beam)의 기울기를 제어하여 볼(Ball)을 빔 위의 원하는 위치에서 정지하도록 하는 것이다. 여기서 빔의 기울기의 제어는 그림 3에서 보듯이 모터를 이용한다. 컴퓨터 모의 실험을 위한 위 시스템의 모션 방정식은 다음과 같이 구해진다.

$$0=\left(\frac{J_b}{R^2}+M\right)\ddot{x}+MG\sin\theta-Mx\dot{\theta}^2 \tag{4}$$

J_b: moment of inertia of the beam
M: mass of the ball
R: radius of the ball
G: acceleration of gravity
x: distance from the center of the beam
θ: declined angle of the beam

상태 변수는 이며 적분방법은 Euler Finite Difference Approximation이 사용되었다. 샘플링 시간은 0.02초로 하였다.

기대치 예측기와 제어기는 3-10-1로 구성된 MLP를 사용하였다. 강화 신호 예측기는 4-10-1로 구성되었고 학습률과 관성항은 각각 0.03과 0.9로 하였다. 볼을 빔의 가운데에 두는 것을 목표로 한다고 가정하여, 다음과 같이 연속적인 강화 신호를 정의하였다.

$$r(t)=\begin{cases}1-2\cdot|x(t)|, & |x(t)|<0.5\\ -1, & |x(t)|\geq 0.5\end{cases} \tag{5}$$

이 경우, 볼이 빔의 가운데에 있을 때 1의 강화 신호를 받고 멀어지면 멀어질수록 강화 신호를 적게 받으며 빔의 길이가 1인 경우 빔의 끝에서 0의 강화 신호를 받게 된다. 볼이 빔의 끝에 도달했을 때 더욱 더 강한 벌을 가하기 위해 실패라고 정의 하고 1의 강화 신호를 받는다.

잡음 생성기는 가우시안 랜덤 잡음 (Gaussian random noise) 을 발생시킨다. 여기서 잡음의 평균은 0이고 분산은 예측된 강화 신호와 반비례가 되도록 다음과 같이 정의한다.

$$\sigma(t)=\min(0.2,\ 1-r(t)) \tag{6}$$

예측된 강화 신호가 1이라면 지금의 제어 신호에 의해 볼이 빔의 가운데로 가는 것을 의미한다. 따라서 이때는 최적 해를 찾아 탐색을 할 필요가 없다. 그러므로 σ는 0이 되어 잡음이 섞이지 않게 된다. 그리고 예측된 강화 신호가 ±8 이면, 즉 다음의 볼의 예측되는 위치가 ±0.1m 이내이면, 잡음의 분산을 0.2 보다 적게 하며 ±0.1 의 밖에 볼의 위치가 예측되면 최대 분산 값인 0.2를 이용하여 잡음을 생성한다. 최대 분산 값은 제어하고자 하는 시스템과 제어기의 특성을 고려하여 정해야 하며 여기서 사용된 Ball-and-

Beam 시스템에서는 0.2로 정하였다. 학습은 실패 후 랜덤 위치에서 다시 학습하는 방법으로 했다. 그림 4는 학습과정을 그래프로 나타낸 것이다. 약1300회의 시도와 학습으로 적어도 200초 이상 볼의 제어에 성공함을 알 수 있다.

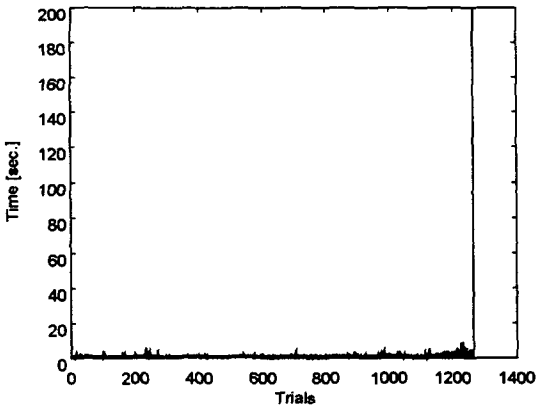


그림 4. Ball-and-Beam 시스템의 학습 곡선
Fig. 4. Learning curve for the ball-and-beam system.

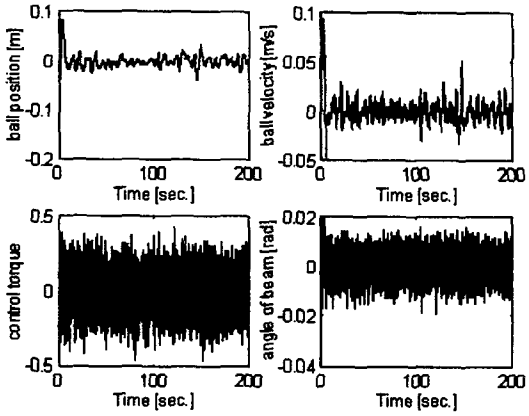


그림 5. 잡음의 분산이 고정된 기존의 제어기의 성능
Fig. 5. Control performance of the conventional controller with fixed noise variance.

그림 5에서와 같이 잡음이 고정되어 있는 기존의 제어기의 경우 목표하는 값에 수렴하지 않고 계속되는 잡음의 영향으로 볼의 위치와 빔의 각도가 진동함을 알 수 있다. 제안된 제어기의 경우 그림 6에서와 같이 목표에 가까이 갈수록 잡음의 크기가 줄어 들어 목표하는 값에서 볼의 위치와 빔의 각도가 수렴함을 알 수 있다. 볼의 위치만으로 강화 신호를 주었지만, 장기적 효용인 기대치(value)를 사용하는 강화 학습 구조로 인해 볼을 0의 위치로 보내기 위해서 볼의 속도와 빔의 각도를 0으로 보내게 학습이 되는 것이다. 제어 신호를 비교하면 그림 5에서는 항상 일정한 분산에 의해서 들어오는 잡음에 의해 학습이 되어서 제어 출력 신호가 크지만 그림 6에서 보듯이 강화 신호 예측기를 이용하여 잡음의 분산을 조절할 때 훨씬 적은 값으로 효율적으로 제어되는 것을 알 수 있다.

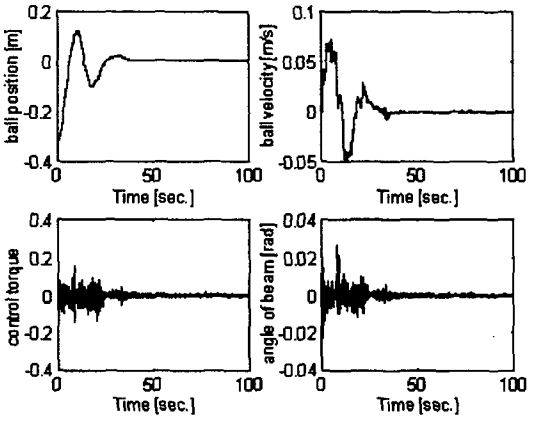


그림 6. 잡음의 분산이 성능에 따라 변화하는 제안된 제어기의 성능
Fig. 6. Control performance of the proposed controller with adaptive noise variance.

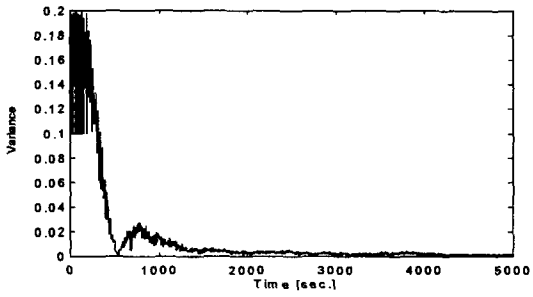


그림 7. 학습이 진행됨에 따른 분산의 변화
Fig. 7. Evolution of noise variance as learning is progressed.

그림7은 학습이 되어 볼이 원하는 위치에 가까이 갈수록 첨가되는 잡음의 분산이 작아짐을 보여주고 있다. Ball-and- Beam 시스템을 Gullapalli의 알고리듬에 적용하였을 때에는 학습이 잘 되지 않았다. 이는 순간적인 강화 신호만을 최대화하기 위해 제어기가 학습이 되기 때문이다. 즉, 볼이 우리가 목표하는 지점인 0에 가까이 있고 속도가 무척 높을 때에도 위치가 0에 가까이 있기 때문에 제어기는 높은 보상을 받게 된

다. 하지만 속도가 크기 때문에 곧 빔의 끝에 다다라 실패하게 된다. 그러나 제안된 구조에서는 장기적 효용인 기대치 값을 이용하므로 볼이 0에 가까이 있을 때에라도 속도가 높으면 장기적인 효용도를 나타내는 기대치 값이 낮기 때문에 순간적인 강화 신호 값을 사용하여 학습하는 것보다 안정하고, 학습도 훨씬 잘 된다.

Ⅲ. 시스템 구조 및 영상처리 알고리듬

1. 시스템 구조

그림 8은 전체적인 시스템 구조를 나타내었다. CCD 카메라는 룸미러(room mirror)의 위치에 설치되었고, 산업용 컴퓨터의 PCI 슬롯(slot)에 장착된 이미지 그래버 보드 (image grabber board)에 연결되어 있다. 차량의 뒤쪽에 실린 산업용 컴퓨터는 133 MHz의 펜티엄 CPU를 장착하고 있는데, 이미지 그래버 보드로부터 이미지를 전송받아 영상처리하고 제어 출력값을 계산하여 ISA 슬롯에 설치된 제어 보드에 전송하고 이 제어보드는 차량의 조향축에 물려 있는 모터를 제어하여 원하는 조향각을 낼 수 있게 하였다.

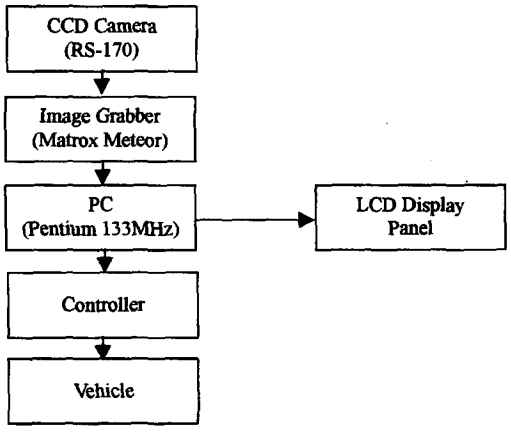


그림 8. 전체 시스템 구조
Fig. 8. Overall system architecture.

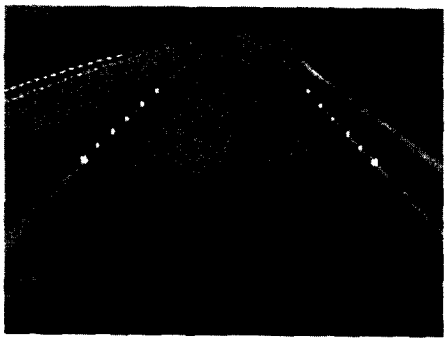
2. 차선 정보 추출을 위한 영상 처리

강화 학습 구조를 사용하기에 앞서, 차선 정보 추출을 위한 영상처리 알고리듬을 개발하였다. 그림 9는 이 논문에서 사용된 영상처리 알고리듬이다. 전체적인 영상에서 바로 차선 정보를 추출하는 것은 시간도 많이 걸리고 쉽지 않기 때문에 지역 영상에서 차선 정보를 추출하여 전체적으로 다시 검증하는 방법을 채택하였다. 첫 순간에는 전 영역이 영상처리 영역이 되며 전체 영역에서 차선 정보를 찾은 후 다음부터는 이전 시간의 영상에서 찾아진 차선의 위치를 중심으로 일정 넓이만을 영상처리 영역으로 한다. 그러나 찾는 영역의 중심을 이전 영상의 차선 위치로 하게 되면 잡음에 민감하게 되므로 차선이 시간적으로 이웃한 영상간에는 크게 변하지 않는다고 가정하여 다음과 같은 방법으로 찾는 영역의 중심을 구한다.

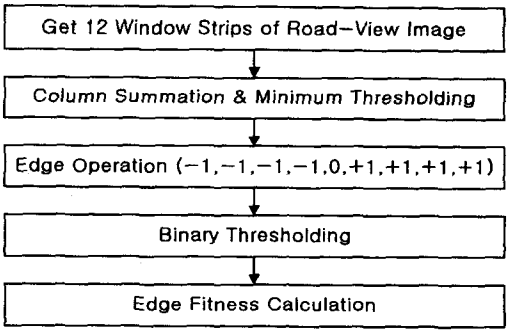
$$s(t+1) = \alpha \cdot c(t) + (1-\alpha) \cdot s(t) \tag{7}$$

$s(t)$: 시간 t 에서 찾는 영역의 중앙 위치
$s(t)$: 시간 t 에서 찾아진 차선의 위치.

여기서 사용된 값은 0.05이고 찾는 영역의 넓이는 중앙 위치를 중심으로 ±30 화소(Pixel)이다.



(a)



(b)

그림 9. (a) 실제 영상 처리에 사용한 12개의 탐색 윈도우, (b) 차선을 찾기 위한 영상 처리 과정
Fig. 9. (a) Twelve search windows used for real-time processing. (b) Image processing procedure for lane detection.

그림 10에서처럼 잡음 성분을 제거한다. 차선 정보와 관계 없는 길은 그림자나 타이어 자국에 영향을 받

지 않기 위해서 일정한 값 이하의 열들은 같은 값들을 갖게 한다. 이 과정을 Minimum Thresholding이라 한다. 여기서 사용된 각 영역의 줄(row)의 개수는 그림 9 (a)의 각 스트립(strip)에 대해 위의 영역부터 차례대로 2, 2, 2, 3, 3, 4이고, Minimum Threshold 값은 찾는 지역 영상 화소값들의 평균값을 이용하였다. 차량이 주행하는 도로는 가까이 있는 차선을 볼 경우 부분적으로 선형적인 성질을 가진다는 것을 이용하여 line fitting[10]방법을 이용한다. 왼쪽 차선에서 지역적인 정보를 이용하여 얻은 6개의 차선 위치를 경유점으로 하여 가장 가까운 직선을 구한 후 그 직선으로부터 6개의 점의 위치까지의 거리를 구한다. 점과 직선과의 거리가 일정값 이상일 경우, 이전 영상에서 얻어진 차선 위치와 많은 차이가 날 경우, 구해진 직선이 도로의 일반적인 모양과 일치하지 않을 경우, 잡음 정보라고 간주하고 제거한 후 다시 나머지 차선에 대해 위의 과정을 반복한다. 잘못된 차선 정보가 없다고 판정될 경우 남아 있는 차선의 위치를 최종 찾아진 차선이라고 간주한다.

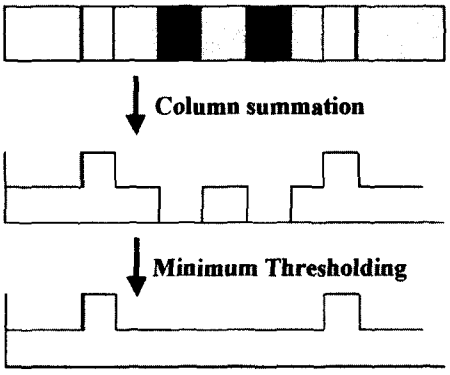


그림 10. 잡음 제거를 위한 열 합과 문턱 처리 과정
Fig. 10. Column summation and minimum thresholding procedure for noise removal.

3. 차량의 모델링

제안된 알고리듬을 실차 실험하기 전에 안정성 및 성능 테스트를 위하여 컴퓨터를 이용한 모의 실험이 필요하며, 이 모의 실험을 통하여 차량의 역학적인 성질을 알아 볼 수가 있다. 정확한 차량의 모델을 구하기 위해서는 고려할 사항이 많아지고 복잡해지기 때문에 간단하게 정리된 식들이 있다. Ozguner[11]는 비선형 두바퀴 모델 (Nonlinear Bicycle Model) 에 기반하여 간단한 식을 유도했으며, Hedrick[3]은 그림 11과 같이 차량의 모델을 간단하게 표현한 후 전륜 구동형 차량에 대한 다음과 같은 시스템 식을 구하였다.

$$\frac{d}{dt}\begin{bmatrix} y_r \\ \dot{y}_r \\ \Delta\varepsilon \\ \dot{\Delta\varepsilon} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & \frac{A_1}{V} & -A_1 & \frac{A_2}{V} \\ 0 & 0 & 0 & 0 \\ 0 & \frac{A_3}{V} & -A_3 & \frac{A_4}{V} \end{bmatrix} \begin{bmatrix} y_r \\ \dot{y}_r \\ \Delta\varepsilon \\ \dot{\Delta\varepsilon} \end{bmatrix} + \begin{bmatrix} 0 \\ B_1 \\ 0 \\ B_2 \end{bmatrix} \delta + \begin{bmatrix} 0 \\ A_1 - V^2 \\ 0 \\ A_4 \end{bmatrix} \frac{1}{\rho} \tag{8}$$

$$\Delta\varepsilon = \varepsilon - \varepsilon_a \tag{9}$$

y_r: 차량의 무게중심과 도로의 중심선과의 거리
ε: 차량의 편주각 (yaw angle)
ε_a: 목표 편주각
δ: 앞 바퀴의 조향각
ρ: 도로의 곡률 반경
V: 차량의 속도

$A_1, A_2, A_3, A_4, B_1, B_2$ 는 차량의 매개변수(parameter)로 다음과 같이 정의되어 있다.

$$A_1 = \frac{-2(C_{sf} + C_{sr})}{m}; \quad A_2 = \frac{-2(C_{sr}l_2 - C_{sf}l_1)}{m};$$
$$A_3 = \frac{-2(C_{sr}l_2 - C_{sf}l_1)}{I_z}; \quad A_4 = \frac{-2(C_{sr}l_1^2 + C_{sr}l_2^2)}{I_z};$$
$$B_1 = \frac{2C_{sf}}{m}; \quad B_2 = \frac{2l_1C_{sf}}{I_z}. \tag{10}$$

m: 차량의 질량
I_z: 차량의 회전 관성력 (rotational inertia)
l_1: 앞 바퀴축과 차량의 무게 중심과의 거리
l_2: 뒤 바퀴축과 차량의 무게 중심과의 거리
C_{sf}: 앞 바퀴 타이어의 cornering stiffness
C_{sr}: 뒤 바퀴 타이어의 cornering stiffness

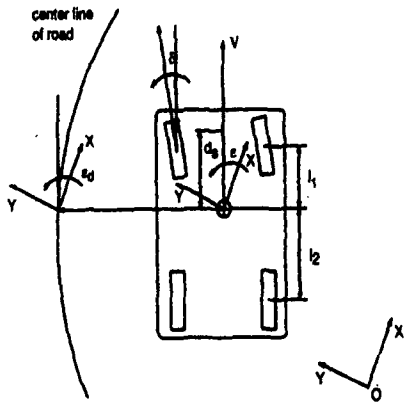


그림 11. 차량의 선형 모델[3]
Fig. 11. A linear model of a vehicle.

여기서 타이어의cornering stiffness는 타이어와 도로와의 상관관계를 표현하는 매개변수이다. 본 논문의 실험에 사용한 현대 그레이스 밴의 변수 매개 변수 값은 표 1과 같다.

표 1. 현대 그레이스 밴의 차량 매개변수 값
Table 1. Parameter values of a Hyundais Grace van.

Cornering Stiffness(N/rad)	68327
질량(kg)	1750
회전 관성력(kg-m^2)	3464
l_1 (m)	1.08
l_2 (m)	1.36

Ⅳ. 강화 학습을 이용한 차량의 횡적 제어 구조

실제 차량의 제어를 위해 2장에서 제안되었던 강화 학습 구조를 사용하였다. 강화 학습 예측기, 기대치 예측기, 신경 제어기 (Neuro-Controller) 로는 2장에서 Ball-and-Beam 시스템의 제어에 사용하였던 은닉층이 하나인 MLP 신경 회로 대신에 처리 속도가 빨라서 실시간 응용이 가능한 CMAC을 사용하였다.

1. 컴퓨터 모의 실험 시 차량의 횡적 제어 구조

오프라인(off-line)학습이 가능한 대부분의 교시 학습과 다른 점으로 강화 학습은 주로 온라인(on-line) 학습에 의존해야 하기 때문에 가능한 학습에 걸리게 되는 시간을 줄여야 한다. 더군다나 실제 차량으로 실험하는 경우 착오에 의해서 계속 차선을 벗어나게 하는 것은 무척 위험하기 때문에 몇 가지 학습 속도를 개선시키고 학습 시 안전을 보장할 수 있는 방법이 필요하다. 학습 속도가 빠른 CMAC 신경 회로를 사용하여 학습에 걸리는 시간을 단축시켰고, 안전을 위해서 실패의 정의를 차선을 벗어나는 시점이 아니라 차선 안에 일정한 경계선을 설정하여 차량이 이 선을 벗어날 경우 실패라 간주하고 다시 재시도 하였다. 표 2에서 보듯이 차선의 실제 폭은 약 3.5 ~ 4 m 정도이지만, 차로의 중심에서 0.5m 밖으로 벗어났을 때 실패라고 간주하고 다시 임의의 위치에서 학습이 되도록 하였다.

강화 신호와 잡음의 분산은 다음과 같이 정의하였다.

$$r(t)=\begin{cases}1-2\cdot|y(t)|, & |y(t)|<0.5\\ -1, & |y(t)|\geq 0.5\end{cases} \tag{11}$$

$$\sigma(t) = \min(0.01, 0.01\times(1-r(t))) \tag{12}$$

여기서 $y(t)$는 차량의 횡거리이고 $r(t)$는 예측된 강화 신호이며 CMAC의 학습률은 0.3이고, 강화 학습의 기대치 학습에서 감소율은 0.95로 하였다.

표 2. 모의 실험 시 CMAC의 상태 변수의 범위
Table 2. The range of CMAC state variables for simulation.

입력의 상태변수	입력 레벨	양자화 수준 (강화 신호 예측기)	일반화 상수 (강화 신호 예측기)
횡적 거리(m)	[-0.5, 0.5]	128(32)	64(32)
횡적 속도(m)	[-0.5, 0.5]	128(32)	64(32)
핸들 상태(rad.)	[-0.05, 0.05]	128(32)	64(32)

2. 컴퓨터 모의 실험

컴퓨터 모의 실험을 위해서 우선 그림 12와 같은 모의 도로를 만들었다.

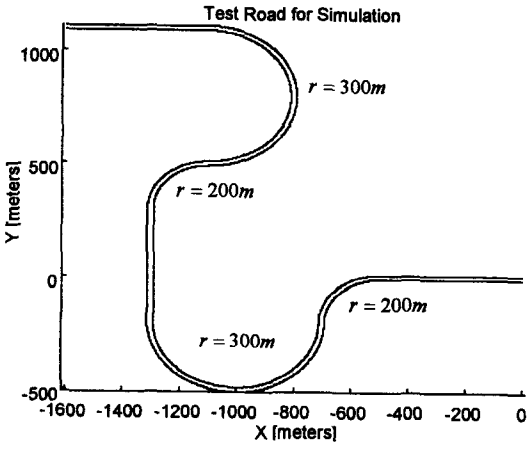


그림 12. 컴퓨터 모의 실험용 도로
Fig. 12. Test road for computer simulation.

학습은 처음에 출발점에서 시작하여 실패하면 다시 출발점에서 실험하는 방법을 취했다. 차량의 속도는 40km/h로 설정하였고 그림 13은 이때의 학습 곡선을 나타낸 것이다. CMAC의 빠른 학습 능력으로 70번이 조금 넘어서 모의 도로를 완주하였다. 이때의 결과는 그림 14 (b)에 나타내었다. 그림에서 보듯이 처음으로 학습에 성공하였을 때에는 성능이 그리 좋지 않음을

알 수 있다. 곡률이 반대로 바뀌는 지점에서 오차가 순간적으로 무척 커짐을 쉽게 확인할 수 있다.

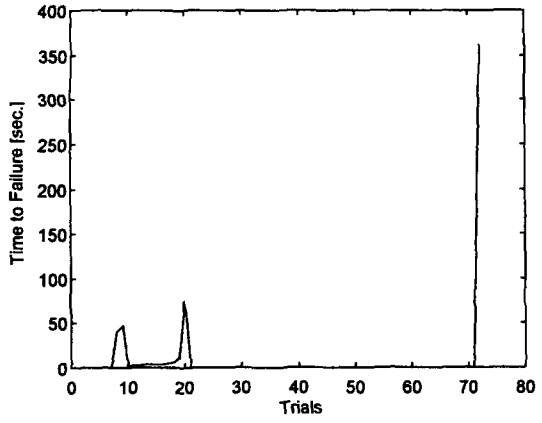
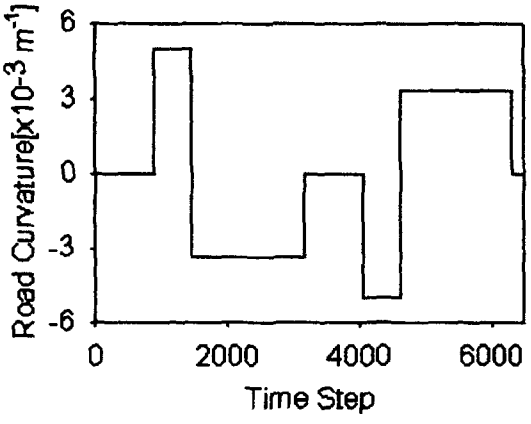
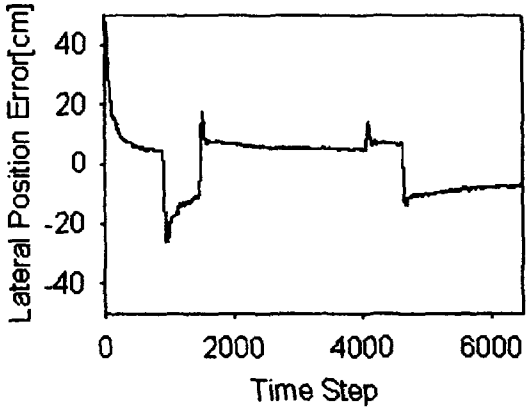


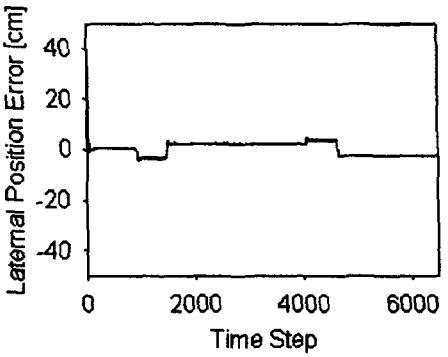
그림 13. 컴퓨터 모의 실험 시 학습 곡선
Fig. 13. Learning curve at the time of computer simulation.



(a)



(b)



(c)

그림 14. 40km/h에서의 강화 학습 제어기의 성능. (a) 도로의 곡률, (b) 처음 주행에 성공하였을 때, (c) 처음 주행 성공 후 300번 반복 학습하였을 때

Fig. 14. RL performances at 40 km/h. (a) Road curvature. (bc) RL control performance at 40 km/h, (b) after the first successful drive-through, (c) after 300 iterations of further training after the first success.

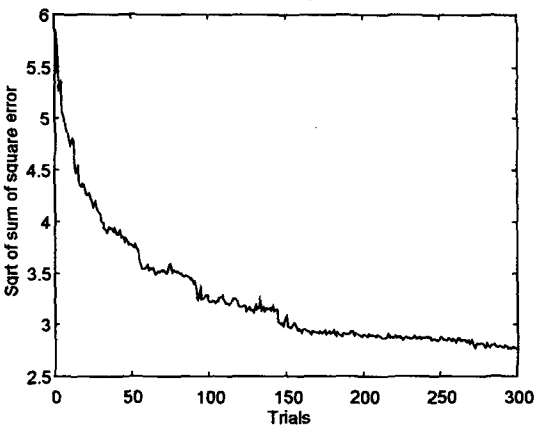


그림 15. 처음 주행 성공 이후의 강화 학습 곡선
Fig. 15. Reinforcement learning curve after the first success.

이는 사람의 경우와 일치하는 특성이다. 속도를 빠르게 하여 80km/h로 하였을 때의 성능은 그림 16 (a)에 나타내었다. 그림 16 (a)에서 곡률이 바뀔 때 순간 오차 최대치는 약 10.6cm이나 정상상태에서는 속도가 증가하여도 별 무리 없이 적응해 가는 것을 볼 수 있다. 그림 16 (b)에 40km/h의 속도에서 학습된 신경회로 제어기를 80km/h의 속도로 다시 50번 재학습시켰을 때 곡률이 바뀌는 지점에서 최대 오차가 줄어드는 것을 보였으며, 그림 16 (c)에는 다시 이 신경

회로 제어기로 40km/h의 속도로 달리는 차량을 제어할 때 과거에 학습한 것을 잃어버리지 않는다는 것을 보였다. 속도의 증가로 인하여 곡률이 바뀌는 지점에서 오차가 커지는 것을 볼 수 있다.

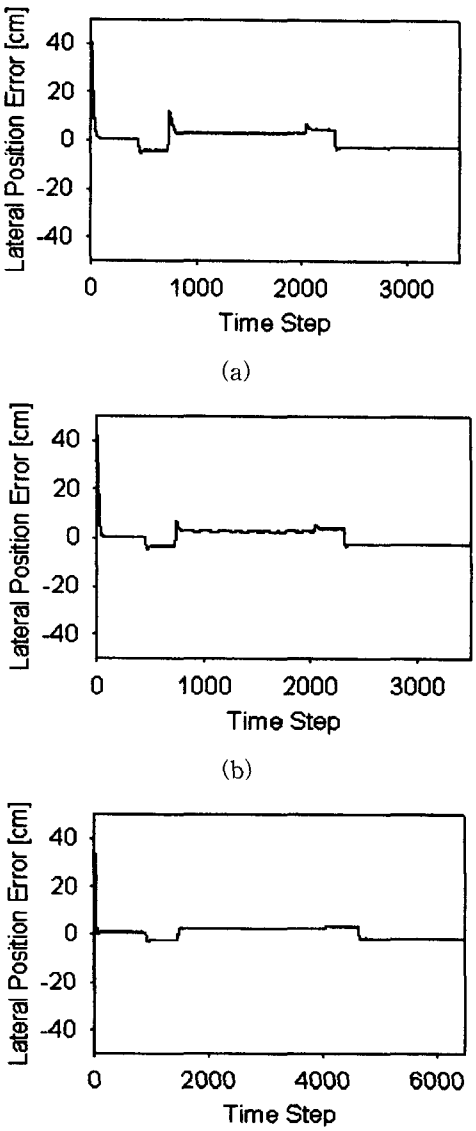


그림 16. 다른 속도에서의 강화 학습 제어기의 성능. (a) 40 km/h의 속도로 학습한 후 80km/h로 주행했을 때, (b) 80km/h로 50회 반복 학습했을 때, (c) 다시 40km/h의 속도로 주행했을 때

Fig. 16. RL control performances at various vehicle speeds. (a) Controller trained at 40 km/h but tested at 80 km/h. (b) Controller further trained for 50 iterations at 80 km/h. (c) Same controller tested back at 40 km/h.

그러나 80km/h의 속도로 학습이 되면 다시 곡률이 변할 때의 오차의 크기가 줄어들며 또 한가지 특기할 점은 다시 40km/h의 속력으로 줄였을 때도 원래 40km/h로만 학습된 경우보다 성능이 더 좋아진다는 점이다.

비교를 위해서 그림 17 (a), (b)는 각각 40km/h와 80km/h의 속력으로 달리는 차량을 PD제어기를 이용하여 제어한 결과를 나타낸 것이다. 속도가 증가할수록 제어 성능이 떨어짐을 알 수 있다. 또한 PD제어기는 샘플링 시간이 줄어들면 줄어들수록 제어 성능도 더 나빠진다.

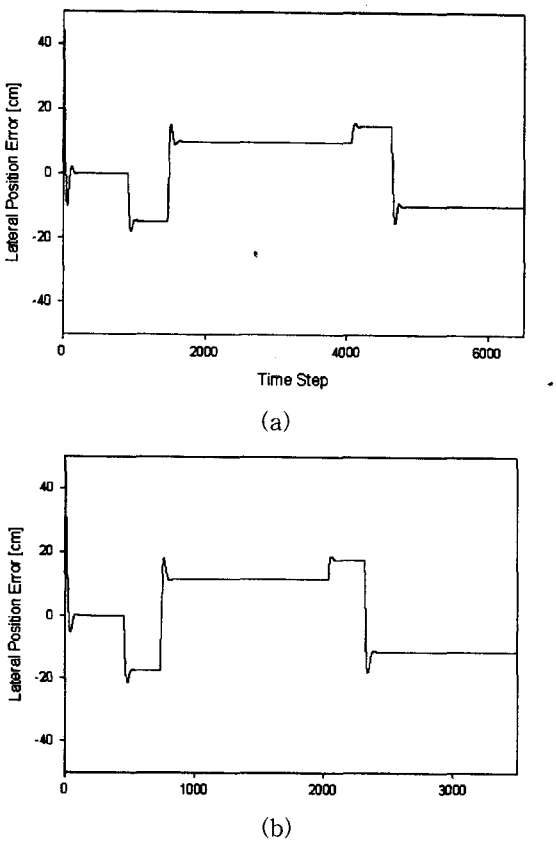


그림 17. PD (P = 0.1, D = 0.05) 제어기의 성능 (a) 40 km/h, (b) 80 km/h

Fig. 17. PD (P = 0.1, D = 0.05) control performance at (a) 40 km/h, (b) 80 km/h.

3. 실험 시 차량의 횡적 제어 구조

기본적으로 컴퓨터 모의 실험에서 사용되었던 구조를 그대로 사용하였다. 다만 컴퓨터 모의 실험에서 차량의 상태 변수를 차량의 모델식에서 얻을 수 있는 것과는 달리 3장에서 설명한 영상 처리 알고리듬을 이용

하여 얻어야 하며 영상에서 얻은 차선의 위치를 영상 좌표를 이용하여 얻기 때문에 차의 중심선과 도로의 중심 사이의 횡적 거리가 앞의 컴퓨터 모의 실험에서 사용된 단위와 틀려진다. 식 (13)은 전방 도로의 중심을 구하는 식이다.

$$y(t)=\frac{w_1(y_1^l(t)+y_1^r(t))+\cdots+w_6(y_6^l(t)+y_6^r(t))}{(w_1+\cdots+w_6)\times 2} \quad (13)$$

여기서 $y_n^{l,r}(t)$는 t 시간의 영상의 위로부터 n번째 지역 영상에서의 왼쪽, 오른쪽 차선의 위치이고 $w_1+\cdots+w_6$는 영상의 위쪽으로부터 아래쪽으로 각각의 지역 영상에 대한 가중치를 나타낸다. w_1 방향의 가중치가 크면 영상에서 위쪽에 가중치를 두어 먼 곳을 보는 효과를 얻을 수 있고, 반대로 w_6 방향의 가중치가 크면 영상의 아래쪽에 가중치를 두어 가까운 곳을 보는 효과를 얻는다. 따라서 속도에 따라 가중치를 변하게 할 경우 안정적인 제어가 가능하다. 여기서는 모두 1로 같게 놓아 영상의 중간에서 차량의 횡적 거리를 얻었다. 이와 같이 달라진 입력에 대하여 표 3은 CMAC의 입력 상태 변수의 구성을 나타내었다.

표 3. 실험 시 CMAC의 상태 변수의 범위
Table 3. Range of CMAC state variables for real experiment.

입력의 상태 변수	입력 레벨	양자화 수준	일반화 상수
횡적 거리(Pixel)	[-25, 25]	50	3
횡적 속도(Pixel)	[-5, 5]	10	3
핸들 상태(Pulse)	[-500, 500]	50	3

또한 강화 신호와 잡음의 분산 신호도 다음과 같이 새롭게 정의하였다.

$$r(t)=\begin{cases}1-|y(t)|/25, & |y(t)|<25\\ -1, & |y(t)|\geq 25\end{cases} \quad (14)$$

$$\sigma(t)=\min(0.01,\ 0.01\times(1-r(t))) \quad (15)$$

위 식에서 차량의 횡적 거리가 25 화소를 넘어서면 실패로 간주하고 강화 신호가 1이 된다. 그 외의 변수들은 컴퓨터 모의 실험에 사용되었던 값과 같다.

4. 실차 실험 결과

실험 장소로는 캠퍼스 근처의 도로이다. 도로의 길이가 짧기 때문에 처음부터 제어기를 학습시키지 않고 우선 P 제어기를 이용하여 CMAC 제어기가 실패하지 않고 차량을 제어할 수 있도록 학습한 후에 강화 학습을 이용하였다.

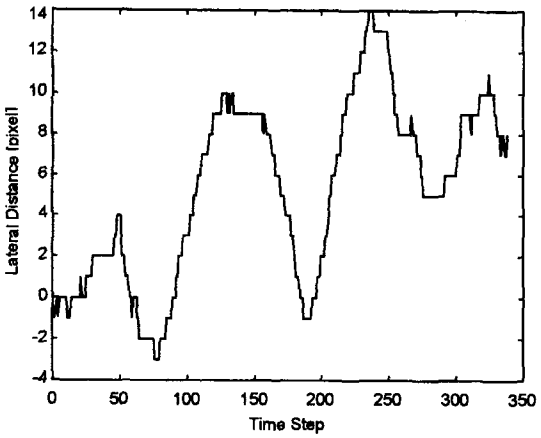


그림 18. 40km/h의 실차 실험 시 P 제어기의 성능
Fig. 18. Control performance of the P controller for real experiment at the speed of 40km/h.

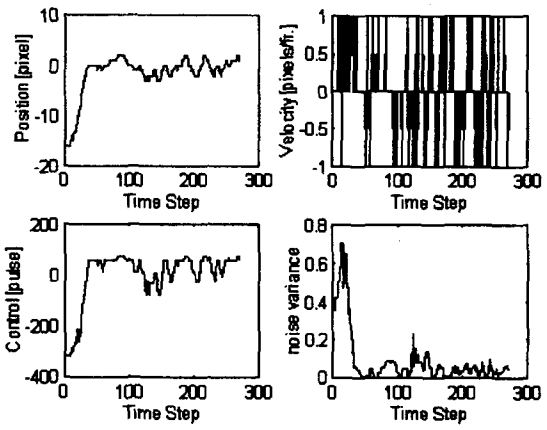


그림 19. 40km/h에서의 제안된 제어기의 성능
Fig. 19. Control performance of the proposed controller at the speed of 40km/h.

그림 18은 기존의 P 제어기를 이용하여 약 40km/h의 속력으로 달리는 차량의 제어할 때 성능을 나타내었고 그림 19는 P 제어기로 학습을 시킨 후 강화 학습을 이용하면 성능이 개선됨을 보인 것이다. 모의 실험에서와 같이 P 제어기보다는 개선된 성능을 보여주고 있다. 최적 제어를 찾기 위해 첨가하는 잡음의 크기가 차량이 도로의 가운데로 감에 따라 감소하는 것을 볼 수 있다. 횡 방향의 단위인 화소 하나의 크기는

약 2.5cm이고 1초에 약 15장의 영상을 처리하였다.

V. 결론 및 향후 연구

본 논문에서는 연속 공간에서 사용할 수 있는 새로운 강화 학습 구조를 제안하였다. 제안된 강화 학습 구조를 사용한 Ball-and-Beam 시스템과 모의 차량의 제어로 알고리듬의 유용성과 타당성을 확인하였다. 실제 차량에 적용하기 위해서는 환경의 인식이 필요하므로 영상 센서인 카메라를 사용하였다. 획득한 영상으로부터 차선 정보를 추출하는 알고리듬을 개발하였다. 제안된 강화 학습 구조는 사람이 운전할 때와 유사하게 학습이 반복될수록 우수한 성능을 보였고, 무엇보다도 온라인 학습 및 계속적인 성능 개선이라는 특성을 확인할 수 있었다.

실제로 주행 실험을 할 경우 너무나 다양한 환경이 존재하기 때문에, 아직까지 모든 환경에서 완벽하게 차량의 횡 방향을 제어하기에는 충분치 않다. 영상 처리 부분에서 일반적인 도로에서는 충분히 안정적으로 작동하였으나 나무의 그림자와 같이 얼룩진 부분을 처리할 때는 문제점이 노출되기도 했다. 또 속도에 따라 어느 정도 멀리 있는 영상을 봐야 하는가도 해결해야 할 문제이다. 영상 처리 부분에서 너무 제한된 정보인 횡 방향 거리와 횡 방향 속도만을 강화 학습 제어기에 전해주기 때문에 영상 처리 부분과 제어기가 분리되어 신경회로의 효과를 크게 살리지 못하는 문제점도 있다. 신경 회로가 제어기에 주로 사용되었지만, 향후에는 다양한 형태의 영상에서 사람이 차선의 위치를 찾아내듯이 차선의 위치를 찾을 수 있도록 영상 처리 부분에 대한 적용도 연구되어야겠다.

감사의 글

※ 본 연구는 1998년 서울대학교 제어 계측 신기술 연구 센터의 연구기금에 의해 연구되었음.

참 고 문 헌

[1] D. Pomerleau, "Neural Network Perception for Mobile Robot Guidance," Ph. D. thesis, School of Computer Science, Carnegie-Mellon University, 1992.

[2] D. Pomerleau and T. Jochem, "A Rapidly Adapting Machine Vision for Automated Vehicle Steering," IEEE Expert, pp. 19-27, 1996.

[3] J. K. Hedrick, M. Tomizuka, and P. Varaiya, "Control Issues in Automated Highway Systems," IEEE Trans. on Control Systems, pp. 21-32, 1994.

[4] R. Behringer and M. Maurer, "Results on Visual Road Recognition for Road Vehicle Guidance," Proc. of Intelligence Vehicles 96, Tokyo, pp. 415-420, 1996.

[5] D. H. Choi, S. Y. Oh, and K. I. Kim, "Connectionist-Nonconnectionist Fusion Architecture for High Speed Road Following," Neural, Parallel & Scientific Computations, vol. 4, No. 3, pp. 367-386, 1996.

[6] G. Yu and I. K. Sethi, "Road Following with Continuous Learning," Proc. of Intelligent Vehicle 95, Detroit, pp. 412-417, 1995.

[7] R. S. Sutton, "Learning to Predict by the Methods of Temporal Difference," Machine Learning, nol. 3, pp. 9-44, 1988.

[8] A. G. Barto, R. S. Sutton, and C. W. Anderson, "Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problems," IEEE Trans. on SMC, vol. SMC-13, no. 5, 1983.

[9] V. Gullapalli, "A Stochastic Reinforcement Learning Algorithm for Learning Real-valued Functions," Neural Networks, vol. 3, no. 6, pp. 671-692, 1990.

[10] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*, Vol. 1, pp. 588-597, Addison-Wesley Pub., 1992.

[11] U. Ozguner, K. A. Unyelioglu, C. Hatipoglu, and F. Kautz, "Design of a Lateral Controller for Cooperative Vehicle Systems," IVHS and Advanced Transportation Systems, pp. 27-34, Detroit, 1995.

저 자 소 개

李 政 勳(正會員)
1996년 2월 포항공과 대학교 전자전기공학과 졸업(공학사). 1998년 2월 포항공과 대학교 전자전기공학과 대학원 졸업 (공학석사). 주관심 분야는 신경망, 퍼지논리, 진화 알고리듬, 로보틱스, 자율 주행 로봇, 감성 공학

崔 斗 鉉(正會員) 第 35卷 C編 第 9號

吳 世 泳(正會員) 第 35卷 C編 第 9號