
주변 상황에 적응하는 LVQ 지능 시스템

엄 기 환*

Adaptive LVQ Intelligent System for Perimeter Condition

Ki-Hwan Eom*

본 연구는 동국대학교 전문학술지 논문개재연구비 지원으로 수행되었습니다.

Abstract

In this paper, the system with an artificial intelligent that is able itself to adjust the perimeter condition of the plant is presented. The proposed intelligent system is composed of two learning vector quantization(LVQ) networks, which are used mostly in the field of the pattern recognition and signal processing.

From the external condition of the plant, the first LVQ network recognizes the pattern of the sensed signal and the second LVQ network judges synthetically user's characteristics and performs learning. The controller controls the plant using the reference value, which is the output value of the synthetic judgement part.

In order to verify the usefulness of the proposed method, we simulated the two LVQs are implemented for the artificial intelligent illuminator as well as being carried out computer simulations. We implemented the proposed artificial intelligent illuminator and perform the experiment

1. 서론

20세기 중반부터 시작된 인공지능에 대한 연구는 21세기를 앞둔 현재까지도 많은 연구가 이루어지고 있다. 특히 1980년대 중반부터는 그 연구가 대단히 활발해져 인공지능이 적용되는 범위가 확장되어, 여러 분야에 인공지능이 응용되고 있다.¹⁾
1) [3], [6]

인공지능의 종류로서는 전문가 시스템(Expert System), 퍼지 시스템(Fuzzy System), 신경 회로망(Artificial Neural Network), 유전자 알고리즘(Genetic Algorithm), 카오스(chaos) 등이 있으며, 각각의 특성에 따라 장점은 살리고 단점을 극복하기 위하여 혼합된 인공 지능 형태도 있다. 이들 중에서도 신경 회로망은 외부로부터 주어지는 정보에 따라서 뉴런 사이의 연결 강도와 바이어스

* 동국대학교 전자공학과

접수일자 : 1999년 7월 17일

를 조정하고, 외부의 정보를 신경 회로망 내부에 반영하는 시스템으로서 그 학습 방법에 따라 다음과 같이 크게 세 가지로 구분된다. 목적 패턴이 주어져서 신경회로망의 출력 패턴이 목적 패턴과 근사화 되도록 학습하는 지도 학습(supervised learning), 목적 패턴이 없이 자기 조직을 지도함으로써 입력 패턴을 인식하는 비지도 학습(unsupervised learning) 그리고, 외계로부터 입력을 얻어 이것을 기초로 출력을 내는 강화학습(reinforcement learning) 등이다.^{[1]-[5]}

본 논문에서는 제어 대상, 즉 플랜트를 사용하는 사용자가 여러 가지 변화하는 상황에 맞게 플랜트의 목표값을 지정하여야 하는 기존의 제어 방식과는 달리 각 상황에서 발생할 수 있는 모든 출력의 경우들 중에서 사용자가 가장 선호하는 출력을 만들 수 있도록 목표값을 자동으로 지정해주는 지능 알고리즘을 제안한다. 제안한 지능 알고리즘은 신경 회로망을 이용하여 구성하며 각 상황마다 사용자가 원하는 출력을 학습함으로써 학습이 종료되면 이후 각 상황마다 사용자가 원하는 출력을 자동으로 만들어낼 수 있는 기능을 갖도록 한 것이다. 신경 회로망은 두 개의 학습 벡터 양자화(Learning Vector Quantization), 즉 LVQ 네트워크로 구성한다.

플랜트에 주어진 상황과 사용자가 원하는 출력을 예측하여 예측된 신호가 제안한 지능형 알고리즘의 입력 패턴으로 들어가면 첫 번째 LVQ 네트워크에서는 패턴을 인식하여 두 번째 LVQ 네트워크를 학습하도록 한다. 첫 번째 LVQ 네트워크는 패턴 인식을 위한 전처리기로써 군집화(Clustering)된 데이터 베이스를 구성하고 두 번째 LVQ 네트워크는 학습 기간동안 첫 번째 LVQ 네트워크에 의하여 학습을 한 후 학습이 종료되면 각 상황에 맞도록 사용자가 원하는 플랜트의 목표값을 자동으로 지정해 준다. 여기서, 학습 기간은 플랜트를 사용하는 사용자의 특성에 적응하는 기간으로서 사용자가 정의할 수 있으며, 특별한 경우에는 학습 기간이 길수록 제안한 방식의 성능의 정확성이 높아질 것이다. 학습 기간은 사용자의 특성을 판단하는 기간이므로 사용자의 특성을 판단할 수 있는 데이터를 더 많이 확보할 수 있는 것이다.

제안한 지능형 제어 알고리즘의 유용성을 확인하기 위하여 플랜트로서 광도가 조절 가능한 조명기에 적용하여, 조명기의 주위의 임의의 상황에 대하여 사용자가 원하는 출력을 학습한 후 같은 상황에 주어진 여러 가지 목표값들 중에서 사용자가 원하는 출력을 자동으로 발생시키는지를 시뮬레이션 및 실험을 통하여 검토한다.

II. 학습 벡터 양자화 네트워크

2-1. LVQ 네트워크의 구조와 동작

LVQ 네트워크는 다층 신경 회로망으로서 입력층, 은닉층(중간층), 출력층 등의 3개 층으로 구성된다. 입력층은 입력 벡터가 되고 은닉층은 경쟁층으로서 경쟁 네트워크를 사용하며 출력층은 선형층으로서 전방향 신경망으로 구성된다. 경쟁층에서 사용되는 활성화 함수는 경쟁 함수(Competitive Function), 출력층에서 사용되는 활성화 함수는 선형 함수(Linear Function)이다.^{[1]-[3]}

LVQ 네트워크의 구조는 그림2-1과 같다.

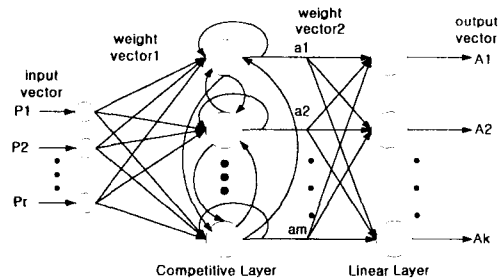


그림 2-1. LVQ 네트워크
Figure 2-1. LVQ Network

LVQ 네트워크에서 경쟁층은 입력 패턴과 원형 패턴의 거리를 직접 계산하여 그 결과를 입력 벡터로 사용한다. 따라서 경쟁 네트워크만을 구성하여 사용하면 내적 계산을 위하여 벡터의 일반화(Normalizing) 과정이 필요하지만 LVQ 네트워크에서는 이러한 과정이 필요 없이 직접 계산할 수 있는 장점이 있다. 경쟁층의 입력 벡터 net 를 표현하면 식(2-1)과 같다.

$${}^1net = \begin{bmatrix} \|{}^1W_1 - P\| \\ \|{}^1W_2 - P\| \\ \vdots \\ \|{}^1W_m - P\| \end{bmatrix} \dots\dots\dots(2-1)$$

여기서, 1W_i 는 i 번째 뉴런의 연결 강도 벡터, $P' = \{P_1, P_2, \dots, P_n\}$ 는 입력벡터이다. 경쟁층의 출력 벡터 1a 는 경쟁 함수의 출력으로서 식(2-2)와 같다.

$${}^1a = \text{compet}({}^1net) \dots\dots\dots(2-2)$$

그러므로 원형 벡터를 나타내는 경쟁층의 입력 벡터와 가장 가까운 연결 강도 벡터를 포함한 뉴런의 출력은 1이 되고, 나머지 뉴런들은 출력이 0이 된다. 이 출력의 의미를 살펴보면, 경쟁 네트워크에서 승리한 뉴런은 입력 벡터를 포함한 클래스를 나타내지만 LVQ 네트워크에서 승리한 뉴런은 클래스에 포함된 부분 클래스를 나타낸다. 예를 들어 경쟁층의 j 번째 뉴런의 출력이 1을 갖으면 j 번째 뉴런에 연결된 연결 강도가 나타내는 원형 벡터에서 가까운 입력 벡터를 포함하는 부분 클래스를 인식하게 되는 것이다.^{[1],[3],[4]}

이러한 동작원리를 패턴인식그림으로 표현하면 그림2-2와 같다.

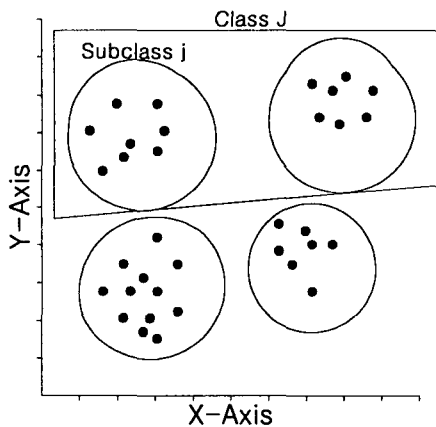


그림 2-2. 군집 및 클래스를 이용한 패턴인식
Figure 2-2. Pattern Recognition using Cluster and Class

LVQ 네트워크의 이러한 동작 원리를 통하여 군집화를 할 수 있으며 군집화와 함께 패턴인식이 가능하게 된다. 기존의 군집화 및 패턴인식기의 구조보다 비교적 간단하고 부분 클래스와 클래스의 표현이 용이하여 많은 부분에서 사용되고 있다.

2-2. LVQ의 학습

LVQ 네트워크의 학습 법칙은 감독 학습을 갖는 경쟁 학습 법칙이다. 모든 감독 학습은 입력 벡터와 목적 벡터(Target Vector)가 동시에 존재한다. LVQ 네트워크에서는 목적 벡터가 부분 클래스를 구분하기 위한 하나의 표시가 되며 이들의 원소는 하나는 1의 값을 갖고 나머지는 0의 값을 갖도록 함으로서 경쟁층의 각 뉴런이 부분 클래스를 나타내도록 지시해주며 표현식은 식(2-3)과 같다.

$$\{P_1, t_1\}, \{P_2, t_2\}, \dots, \{P_Q, P_Q\} \dots\dots\dots(2-3)$$

여기서, P_i 는 입력벡터이고, t_i 는 목적벡터이다. LVQ 네트워크의 중간층을 개선하기 위하여 코호넨 규칙이 사용되며 만약 입력 벡터 P 가 정확하게 분리되면 그때는 중간층의 승리한 뉴런과 연결된 연결 강도가 입력 벡터를 향해 움직이도록 하게 된다. 이러한 관계를 식으로 표현하면 식(2-4)와 같다.

$${}^1W(t) = {}^1W(t-1) + \alpha(P(t) - {}^1W(t-1)) \dots\dots\dots(2-4)$$

그러나, 만약 입력 벡터 P 가 정확하게 분류되면 중간층의 승리한 뉴런이 잘못된 것이므로 입력 벡터로부터 뉴런과 연결된 연결 강도 벡터를 멀리 떨어지도록 연결 강도를 조정해 주어야 한다. 이때의 연결 강도의 학습식은 식(2-5)과 같다.^{[1],[3],[4]}

$${}^1W(t) = {}^1W(t-1) - \alpha(P(t) - {}^1W(t-1)) \dots\dots\dots(2-5)$$

Ⅲ. 제안한 지능 제어 시스템

본 논문에서는 사용자가 이와 같은 시스템을 사용할 때 영구적이면서 지속적으로 변화하는 주변

상황마다 입력을 설정할 필요 없이, 일정한 학습만 시켜주면 각 상황에 적응하는 지능 제어 시스템을 제안한다. 제안한 지능형 제어 알고리즘을 구현하기 위해서는 하나의 가정이 필요하다. 이 가정은 제어대상의 출력과 그에 대응되는 제어기의 입력은 정확하게 사상(Mapping)이 이루어진 상태라고 가정한다. 이것은 제어대상의 사용자가 원하는 출력이 설정되면 그에 해당하는 제어기의 입력이 자동으로 결정됨을 의미한다. 예를 들어, 제어 시스템의 입력과 출력 중에서 원하는 출력이 선택되면 이것과 사상된 입력도 함께 선택되는 것이다. 이와 같은 가정 하에, 본 논문에서 제안한 지능형 제어 시스템의 구조는 그림3-1과 같다.

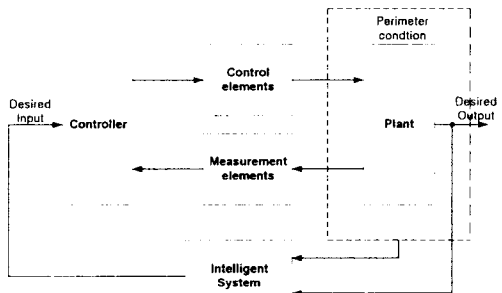


그림 3-1. 제안한 지능 제어 시스템의 전체구조
Figure 3-1. Block Diagram of the Proposed Intelligent Control System

그림 3-1에서 출력과 주변상황은 센서(Sensor)에 의하여 검출되고 이 신호가 지능 시스템(Intelligent System)의 입력이 되면 주어진 주변상황과 그때의 사용자가 원하는 출력을 인식하고 학습한다. 이후 학습이 종료되면 제어 대상에 임의의 상황이 주어졌을 때 학습된 지능 시스템은 그 상황에 맞는 제어기의 입력값을 만들어준다.

제안한 지능 시스템은 신경 회로망으로 구성된다. 패턴 인식을 위한 신경 회로망으로서 두 개의 LVQ를 사용하고, 첫 번째 LVQ네트워크를 'LVQ-I', 두 번째 LVQ 네트워크를 'LVQ-II'라 명칭하며 그 구성은 그림3-2과 같다.

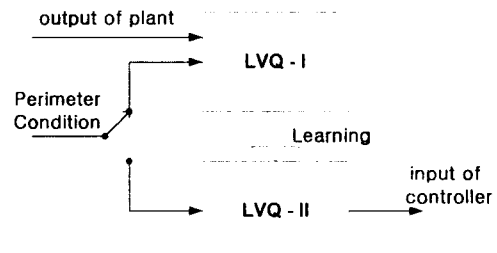
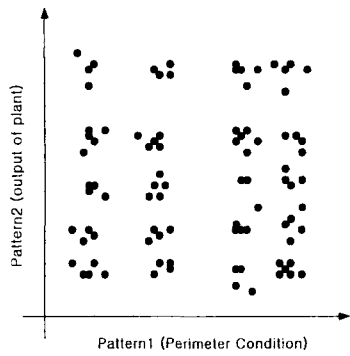


그림 3-2. 제안한 지능 시스템의 구조
Figure 3-2. Structure of the Proposed Intelligent System

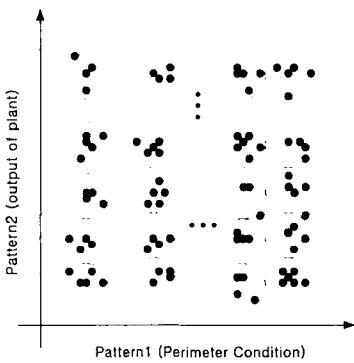
여기서, LVQ-I은 제어대상의 주변상황과 제어대상의 출력을 입력 패턴으로서 인식하여 LVQ-II를 학습시키고, 학습이 종료되면 LVQ-II에 주변상황 신호가 입력으로 들어가 주변상황에 대하여 학습된 결과를 패턴들을 기반으로 각 상황에 맞게 제어기의 입력을 만들게 된다. 주변상황을 검출하여 얻은 신호가 연결된 선은 학습 기간과 관련되어 스위치 역할을 하며 학습 후에는 제어대상의 출력 신호는 검출하지 않고 주변상황만 검출하여 LVQ-II의 입력으로 들어간다.

주변상황에 대한 데이터들과 그에 대응하는 제어대상의 출력 데이터가 그림3-3과 같이 주어진다. 패턴인식을 위하여 제어대상의 특성에 맞게 군집화를 할 수 있다. 또한 군집들 중에서도 같은 특성을 갖는 것끼리 서로 묶어서 하나의 클래스를 형성할 수 있다. 군집을 만들거나 클래스를 만드는 것은 패턴 인식기를 설계하는 사람이 제어대상과 주변상황의 성질에 따라 임의로 선택할 수 있으며 제안한 지능 시스템을 적용하기 위해서도 이 과정은 반드시 선행되어야 한다. 그림4-4는 발생할 수 있는 기본적인 데이터들을 가정하여 나타내고 있다.

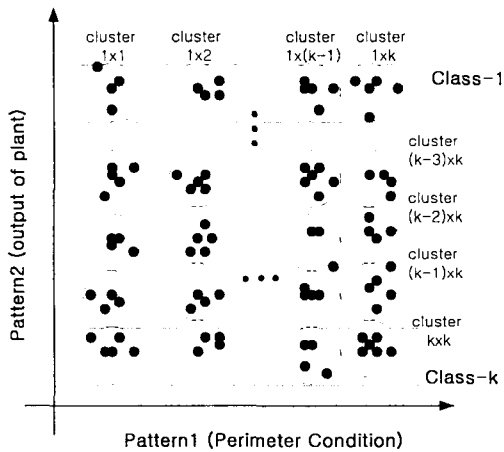
여기서, 그림3-3(a)는 R차원의 주변상황 패턴벡터와 R'차원의 플랜트의 출력 패턴벡터들을 2차원 평면상에서 나타낸 그림이다. 그림3-3(b)는 그림 3-3(a)의 데이터들에서 k x k개의 점들을 중앙점(Center Point)으로 지정하여 중앙점에서 가까운 거리의 데이터들끼리 군집화를 한 것이다. 그림3-3(c)는 그림3-3(b)의 군집들을 플랜트의 출력 패턴이 유사한 것들끼리 클래스로 묶어준 것이다. 이처럼



(a) 주변상황과 제어대상의 출력 데이터



(b) 군집화



(c) 군집화된 데이터의 패턴인식

그림 3-3. 주변상황과 출력데이터 및 패턴인식
Figure 3-3. Perimeter Condition and Output Data, Pattern Recognition

데이터들을 특징에 따라 데이터-베이스(Data-Base)화하여 패턴 인식의 전처리 과정을 수행하면 패턴인식기를 설계할 수 있다.

그림3-3을 기반으로 패턴인식기를 설계한다. 제안한 알고리즘에서 사용된 LVQ 네트워크를 이용하여 패턴인식기를 만들고 다시 이것을 기반으로 학습용 패턴인식기를 설계한다. 그림3-3의 데이터-베이스를 기반으로 설계한 LVQ 패턴인식 네트워크는 그림3-4와 같다.

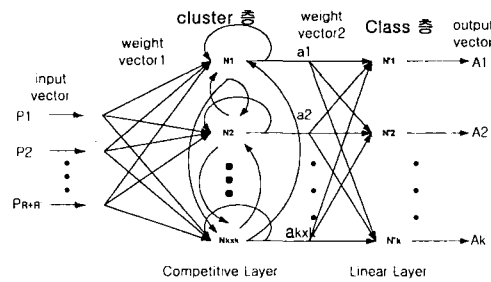


그림 3-4. 지능 시스템의 LVQ-I 네트워크
Figure 3-4. LVQ-I Network of the Intelligent System

여기서, 입력 벡터는 $R+R'$ 차원이며, 이 $R+R'$ 차원은 연결 강도를 결정하는 원형 벡터의 차원도 결정해준다. 입력 벡터와 입력층과 중간층의 연결 강도를 나타내는 연결강도 벡터는 식(3-1), 식(3-2)와 같다.

$$P = \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_R \\ P_{R+1} \\ \vdots \\ P_{R+R'} \end{bmatrix} \dots \dots \dots (3-1)$$

$$W = \begin{bmatrix} W_{11} & W_{12} & \dots & W_{1R+R'} \\ W_{21} & W_{22} & \dots & W_{2R+R'} \\ \vdots & \vdots & \ddots & \vdots \\ W_{k1} & W_{k2} & \dots & W_{kR+R'} \end{bmatrix} \dots \dots \dots (3-2)$$

3차원 이상의 고차원 벡터로서 두 종류의 벡터를 나타내기 위하여 본 논문에서는 각 벡터를 직교 좌표의 가로축과 세로축에 위치하게 함으로서 $R+R'$ 차원의 입력 벡터를 이차원적으로 해석한다. 중간층의 뉴런들은 그림3-3(c)의 각 클러스터와

사상(Mapping)되며 첫 번째 뉴런 N1은 Cluster 1×1, 두 번째 뉴런 N2는 Cluster 1×2, ..., 마지막 번째 뉴런 N k×k는 Cluster k×k를 나타낸다. 출력층의 뉴런들은 각각 그림3-3(c)의 클래스를 나타낸다. 각 클래스들은 소속클래스(Subclass), 즉 클러스터를 갖고 있으며 이러한 원리를 통하여 중간층과 출력층의 연결 강도가 결정된다. 중간층의 뉴런 N1부터 Nk는 클래스-1에 소속되어있으므로 뉴런 N1부터 Nk과 뉴런 N'1과의 연결 강도만이 '1'의 값을 갖고 N'1과 중간층의 나머지 뉴런들과의 연결 강도는 '0'의 값을 갖게 된다. 결국 이와 같은 원리에 의하여 중간층과 출력층의 연결 강도가 결정되어 식(4-3)과 같이 연결 강도 벡터가 결정된다.

$${}^2W = \begin{bmatrix} 1 & 1 & \dots & 1 & 0 & 0 & \dots & \dots & \dots & 0 \\ 0 & 0 & \dots & 0 & 1 & \dots & 1 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & \dots & \dots & 0 & 1 & 1 & \dots & 1 \end{bmatrix} \dots \dots \dots (3-3)$$

여기서, 연결 강도 벡터 2W 은 $k \times k^2$ 차원이 되며, LVQ-I의 동작은 다음과 같다.

먼저, 입력 벡터 P 가 들어오면 원형 벡터 1W 과의 거리를 계산하여 중간층의 뉴런들의 입력을 만들고 식(3-4)와 같다.

$${}^1net = norm(P - {}^1W) = \|P - {}^1W\| \dots \dots (3-4)$$

이 중간층은 경쟁 함수를 사용하므로 중간층의 출력은 식(3-5)와 같다.

$${}^1a = compet({}^1net) = \begin{bmatrix} {}^1a_1 \\ {}^1a_2 \\ \vdots \\ {}^1a_k \end{bmatrix} \dots \dots \dots (3-5)$$

1a 는 다시 2W 와 곱해져 식(3-6)과 같이 되어 출력층의 입력으로 들어가며 양의 선형 함수를 사용하는 출력층의 출력은 식(3-7)과 같다.

$${}^2net = {}^2W \cdot {}^1a \dots \dots \dots (3-6)$$

$$A = poslin({}^2net) = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_k \end{bmatrix} \dots \dots \dots (3-7)$$

위 식에 의하여 LVQ-I의 출력이 결정되며, 결국 제어대상의 주변상황 벡터와 출력 벡터를 패턴 인식하게 된다. 그림4-5와 같은 LVQ-I 네트워크는 패턴을 인식하면서 또 다른 LVQ 네트워크, 즉 LVQ-II를 학습시키며 LVQ-II 네트워크는 그림3-5과 같이 설계된다.

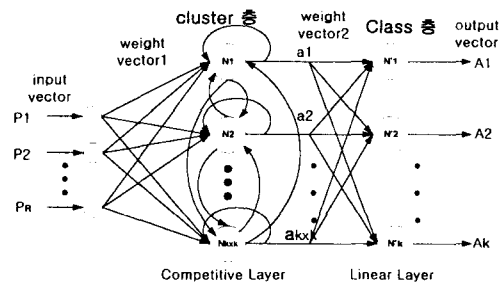


그림 3-5. 지능 시스템의 LVQ-II 네트워크
Figure 3-5. LVQ-II Network of the Intelligent Network

LVQ-II에서 입력 벡터의 차원은 LVQ-I의 입력 벡터 $R+R$ 차원과 달리 LVQ-II R 차원이다. LVQ-II는 입력으로 제어대상의 주변상황만을 입력으로 받아들이고 LVQ-I에 의하여 학습된 결과를 바탕으로 사용자가 원하는 클래스의 출력을 만들기 때문에 입력 벡터가 R 차원이 된다. 그러나 LVQ-I 구조와의 차이는 입력 벡터에 의하여 입력층과 중간층 사이의 연결 강도 이외에는 존재하지 않기 때문에 중간층의 뉴런수와 출력층의 뉴런수, 그리고 중간층과 출력층 사이의 연결 강도는 같다. LVQ-I과 차이나는 LVQ-II의 입력 벡터와 연결 강도는 식(3-8), 식(3-9)와 같다.

$${}^1P = \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_R \end{bmatrix} \dots \dots \dots (3-8)$$

$${}^1W = \begin{bmatrix} W_{11} & W_{12} & \dots & W_{1R} \\ W_{21} & W_{22} & \dots & W_{2R} \\ \vdots & \vdots & & \vdots \\ W_{k1} & W_{k2} & \dots & W_{kR} \end{bmatrix} \dots \dots \dots (3-9)$$

1P 가 LVQ-II의 입력이 되어 원형 벡터 2W 과의 거리를 계산하는 것은 LVQ-I의 동작 원리와 같음

며 그 표현은 식(4-10)과 같다.

$$|net = norm(P - W) = \|P - W\| \dots\dots\dots (3-10)$$

제안한 방식은 LVQ-I에서 어떤 원형 벡터와 거리가 가까운 입력 벡터가 들어왔을 경우 이 원형 벡터와 연결된 뉴런은 활성화하고 LVQ-II의 뉴런을 학습시킨다. 즉, LVQ-II의 중간층의 모든 뉴런들이 LVQ-I의 모든 뉴런들과 각각 하나씩 연결되어, LVQ-I의 중간층의 뉴런중에서 j번째 뉴런이 활성화하면 LVQ-II의 중간층의 j번째 뉴런은 활성화도 높아지도록 한다. 그래서, 학습을 마치면 LVQ-II의 중간층의 뉴런들은 각각 자기만의 활성도를 갖게 되어 LVQ-I의 입력 벡터들 중에서 원소 P_{R+1} 부터 P_{R+R} 까지의 원소가 없어도 활성도가 높은 뉴런이 활성화할 수 있도록 하는 것이다. 이것은 식 (3-11)에 의하여 표현될 수 있다.

$$|a = compet(net / \delta) \dots\dots\dots (3-11)$$

여기서 δ 는 활성도를 나타내고 이것은 제안한 방식을 위하여 식(3-12)와 같이 정의된다.

$$\delta = 1 + \alpha \cdot L \dots\dots\dots (3-12)$$

α , 또한 제안한 방식을 위하여 새로 정의되며 학습률을 나타낸다. α 는 $0 < \alpha \leq 0.01$ 의 범위를 갖아야 한다. L 은 LVQ-I의 중간층의 각 뉴런에 대한 활성 횟수를 나타낸다. 결국, LVQ-I의 중간층의 j번째 뉴런이 l번 활성화했다면 L 은 l값이 되고 LVQ-II의 중간층의 j번째 뉴런의 활성도는 같은 클래스에 속하는 다른 뉴런들 보다 상대적으로 높아지게 된다. 이와 같은 원리를 통하여 하나의 클래스 내에 활성도가 높은 뉴런이 승리하게 됨으로서 임의의 주변상황이 주어지면 그 상황에 맞는 뉴런이 승리하여 지능 시스템의 출력을 만들게 된다.

LVQ-I과 LVQ-II를 포함하여 제안한 지능 시스템은 그림3-6과 같다.

그림에서 상단의 LVQ 네트워크는 LVQ-I을 나타내며 하단은 LVQ-II를 나타낸다. 그리고 각 LVQ의 중간층 내 뉴런들간에 연결된 굵은 선은 LVQ-I이 LVQ-II를 학습시킴을 표현한 것이다.

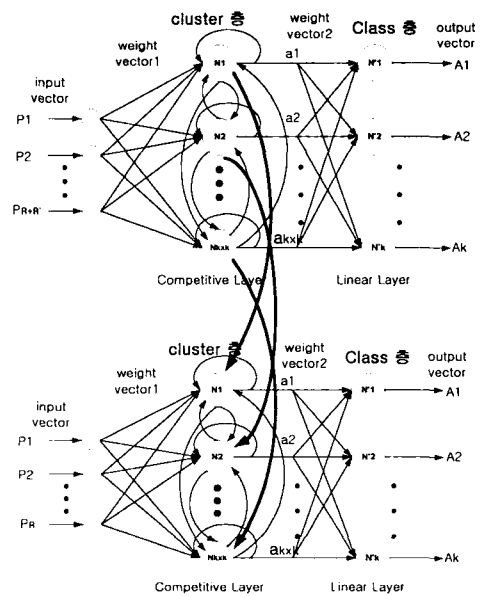


그림 3-6. 제안한 지능 시스템
Figure 3-6. Proposed Intelligent System

LVQ-I의 출력은 지능 시스템의 외부와 연결되지 않고 단지 패턴만 인식하며 지능 시스템의 출력은 LVQ-II의 출력이 된다. 그러면 가정에 의하여 각 클래스와 사상된 제어대상의 출력이 결정되고, 다시 제어대상의 출력과 사상된 제어기의 입력이 설정되어 제어기의 입력으로 들어가는 것이다.

V. 시뮬레이션

제안한 지능 제어 시스템의 성능을 확인하기 위하여 제어대상을 조절 가능한 조명기에 적용하여 시뮬레이션 한다.

제어대상인 조명기는 주변상황이 주위밝기가 되며 출력이 조명기의 밝기가 된다. 주위밝기를 S1이라 하고 조명기의 출력을 S2라고 하여 군집화와 클래스를 그림4-1과 같이 임의로 설정하여 데이터-베이스를 만든다.

데이터 그림에서 주위밝기는 6개의 단계로 구분하여 군집을 설정하고 조명기의 출력 조도는 8개의 클래스로 설정하였다. 총 48개의 군집과 6개의 군집을 갖는 8개의 클래스가 되도록 한 후, 제안한 지능 시스템의 LVQ-I를 설계하면 입력 벡터의

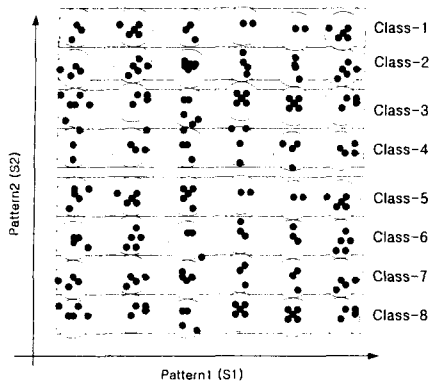


그림 4-1. 조명기의 주위밝기와 출력 데이터
Figure 4-1. Illuminance of the Illuminator and Output Data

원소는 2개, 중간층의 뉴런수는 48개, 출력층의 뉴런수는 8개가 된다. 각 군집의 중심점의 좌표는 다음과 같다.

- (1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (1, 8)
- (2, 1), (2, 2), (2, 3), (2, 4), (2, 5), (2, 6), (2, 7), (2, 8)
- (3, 1), (3, 2), (3, 3), (3, 4), (3, 5), (3, 6), (3, 7), (3, 8)
- (4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (4, 6), (4, 7), (4, 8)
- (5, 1), (5, 2), (5, 3), (5, 4), (5, 5), (5, 6), (5, 7), (5, 8)
- (6, 1), (6, 2), (6, 3), (6, 4), (6, 5), (6, 6), (6, 7), (6, 8)

이 중심점들과 군집을 포함한 클래스를 바탕으로 LVQ-I의 연결 강도를 결정하여 패턴인식을 하면 그림4-2와 같다.

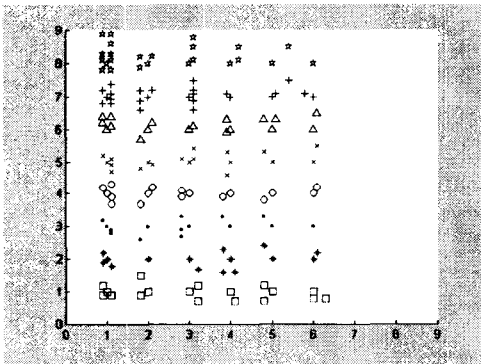


그림 4-2. LVQ-I를 이용한 데이터-베이스와 패턴 인식
Figure 4-2. Data-Base and Pattern Recognition using LVQ-I

그림4-2에서 LVQ-I의 패턴 인식 성능을 확인할 수 있다. 많은 데이터를 입력으로 주었을 때 정확히 패턴 인식한 후 그림에서 각 클래스에 해당하는 기호를 통하여 출력을 만드는 것을 확인할 수 있다. 이러한 패턴 인식의 골격이 되는 LVQ-I의 원형 벡터만을 시뮬레이션 하면 그림4-3과 같다.

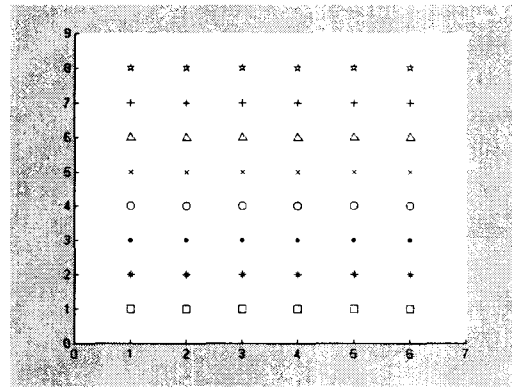


그림 4-3. LVQ-I의 원형 벡터
Figure 4-3. Prototype Vector of LVQ-I

그림4-3을 기반으로 지능 시스템을 구성하여 제어 대상인 조명기에 적용시켜 20번의 학습을 시킨다. 여기서 LVQ-I에 의해 학습되는 LVQ-II와 관련된 식3-12에서 학습률 α '는 0.01로 한다. 학습하는 LVQ-I의 출력은 그림4-4와 같으며 학습된 LVQ-II의 출력과 조명기의 출력은 그림4-5와 같다.

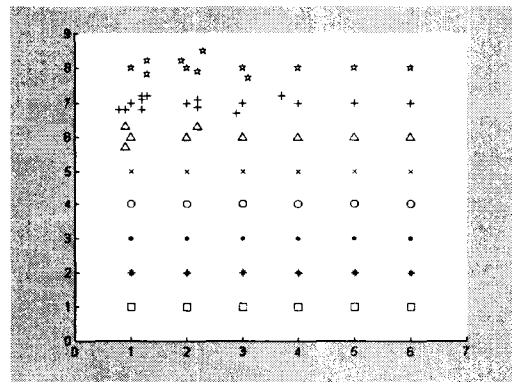


그림 4-4. 제안한 지능 시스템의 학습
Figure 4-4. Learning of the Proposed Intelligent System

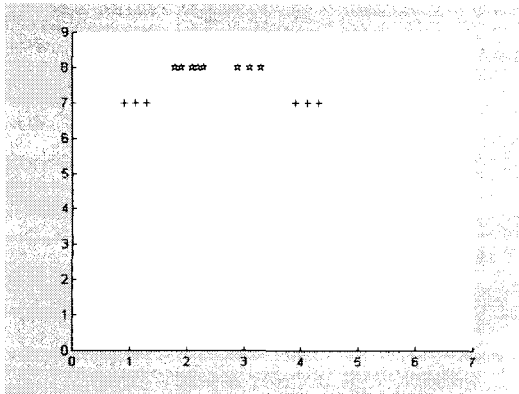


그림 4-5. 제안한 지능 시스템의 출력
Figure 4-5. Output of the Proposed Intelligent System

VI. 실험 및 검토

시뮬레이션에 의한 결과를 실험상으로 확인하기 위한 블록선도는 그림5-1과 같다.

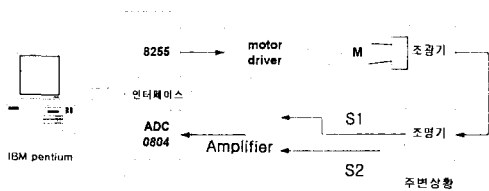


그림 5-1. 실험 장치의 전체 구성도
Figure 5-1. Block Diagram of the Experiment Device

조명기는 220V, 100W용 백열 전구를 사용하고, 조명기의 조도를 조절하는 조광기는 자동 조절을 위하여 스텝핑 모터와 기어를 연결하여 사용하였다. 스텝핑 모터를 사용한 이유는 제어가 쉽고 오차가 적으며 기어와의 연결을 통하여 위치 제어에 적합하기 때문이다. 조도를 감지하기 위한 포토 센서는 코덴시(Kodenshi)사의 HP-3F를 사용하여 주변 조도용 S1, 조명기 조도용 S2로서 사용하였다. 조도계는 히어기(HIOKI) 3422 디지털 룩스 테스터(Digital Lux Tester)기를 사용하여 S1, S2에 해당하는 조도를 측정하였다.

조광기 및 실험 장치 사진은 그림5-2과 같다.

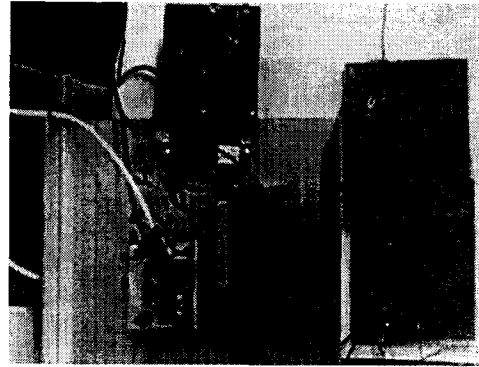


그림 5-2. 실험 장치 사진
Figure 5-2. Picture of the Experiment Device

조명기에서 30cm 떨어진 곳에 S2에 해당하는 포토 센서를 두고, 스텝 모터의 한 스텝당 발생하는 조도를 감지한 센서의 출력 전압을 오실로스코프를 이용하여 측정하였다.

주변조도 S1은 조명기의 영향권밖에 존재하도록 하였고 조명기가 작동하는 환경의 조도를 6단계로 구분하여 패턴인식을 하기 위하여 최소 조도와 최대 조도를 검출하여 6으로 나누어 S1 패턴 입력을 설정하였다. 실험실의 최소 조도는 0 [lux]이고 낮에 측정한 최대 조도는 190 [lux]이다.

LVQ-I의 설계를 위하여 설정된 패턴들의 데이터 베이스를 정리하면 그림5-3과 같다.

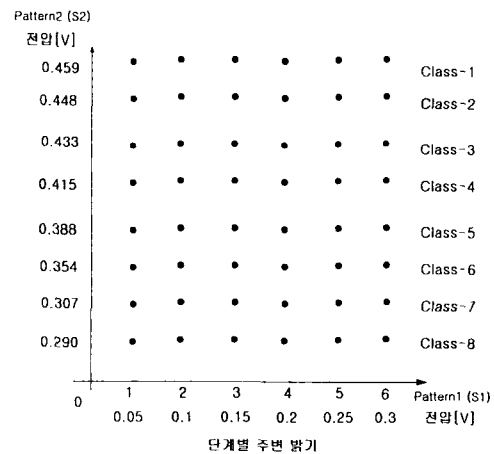


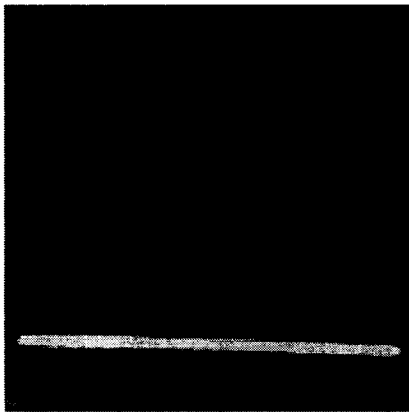
그림 5-3. 실험을 위한 패턴인식
Figure 5-3. Pattern Recognition for the Experiment

조도 센서의 출력을 오실로스코프로 측정하면 그림5-4와 같다.

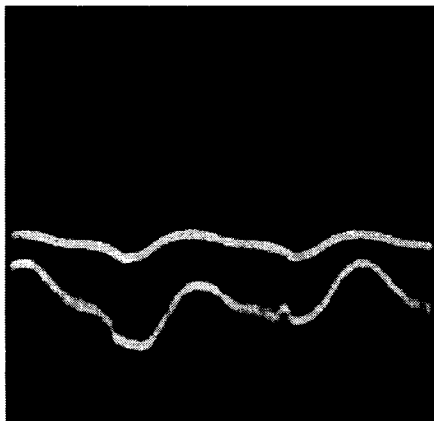
오실로스코프의 두 채널(Channel)은 각각 S1과 S2를 측정하며 0.1 VOLTS/DIV를 기준으로 측정하였다. 그림5-4(a)는 두 채널을 모두 Ground 상태로 고정하였을 때를 나타내며, 이것을 기준으로 조도의 변화를 측정하면 그림5-4(b)와 같다.

그림5-4(b)의 그림에서 S1의 출력이 위 부분의 파형이고 아래 부분의 파형은 S2를 나타내는데, S1의 출력 전압이 S2의 출력 전압보다 높은 이유는 조명기의 안쪽에 S2가 장치되어 조명기의 박스

내부에 존재하므로 외부의 조도에 영향을 받지 않기 때문이다. 제안한 지능 조명 시스템을 작동시키면 센서 S1과 S2에 입력이 그림5-4(b)와 같은 입력을 받아들이며 주어진 조도에 따라 패턴을 인식하고 학습하게 된다. 사용자가 주변조도에 따라 조광기의 조절부를 조절함으로써 LVQ-I이 패턴인식을 하여 LVQ-II를 학습시키고 나면, 이후 주변 조도에 따라 사용자가 많이 사용하던 조명기의 조도를 발생시키도록 스텝모터가 제어하여 그림5-5와 같은 출력을 얻게 된다.

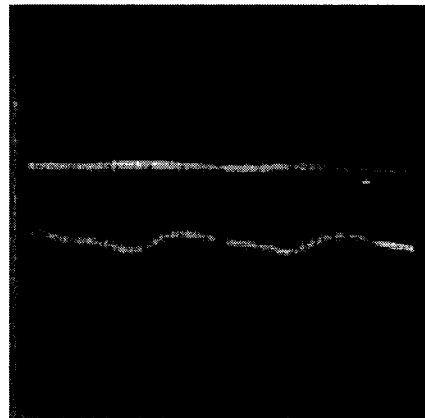


(a) GND 상태의 출력

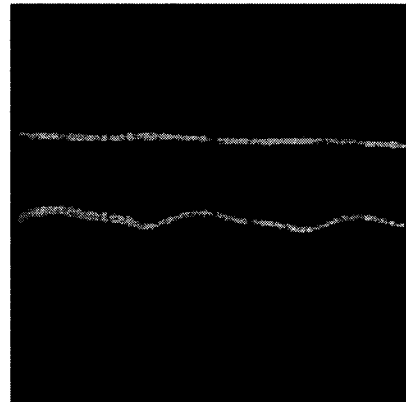


(b) S1과 S2의 출력

그림 5-4. 조명기의 주변조도(S1)와 출력조도(S2)
Figure 5-4. Output Illuminance and Perimeter of the Illuminator



(a) 학습 후의 제안한 지능 조명 시스템의 출력I



(b) 학습 후의 제안한 지능 조명기의 출력II

그림 5-5. 제안한 지능 시스템에 의한 조명기의 출력과 주변조도

Figure 5-5. Illuminance's Output and Perimeter Illuminance by Proposed Intelligent System

지능 조명기 시스템이 주변조도 130 [lux]에서 0.23 [V]의 S1전압을 검출하면 조명기의 출력은 160 [lux]가 되도록 학습을 시키고, 새로운 실험에서 주변조도 130 [lux]에서 조명기의 출력이 300[lux]가 되도록 학습하였다.

그림5-5(a)는 20번의 학습을 한 후 제안한 지능 조명기 시스템의 출력을 나타낸다. 스킵의 아래 부분 전압은 주변조도 S1이 0.24 [V]의 입력으로 들어왔을 때를 나타내며 그때의 조명기의 출력조도 S2는 0.41 [V]로서 스킵의 위 부분 전압이 된다. 같은 방법으로 새로운 데이터들로 다시 20번의 학습시킨 후 지능 조명기 시스템의 출력은 그림5-5(b)에 나타나며 아래 부분의 전압 S1이 0.24 [V], 위 부분의 전압 S2가 0.43 [V] 되므로 학습 후 제안한 지능 시스템은 사용자에게 적응하여 출력을 만들어 냄을 확인할 수 있다.

Ⅶ. 결 론

본 논문에서는 제어 대상, 즉 플랜트를 사용하는 사용자가 여러 가지 변화하는 상황에 맞게 플랜트의 목표값을 지정하여야 하는 기존의 제어 방식과는 달리 각 상황에서 발생할 수 있는 모든 출력의 경우들 중에서 사용자가 가장 선호하는 출력을 만들 수 있도록 목표값을 자동으로 지정해주는 지능 알고리즘을 제안하였다. 제안한 지능 알고리즘은 신경 회로망을 이용하여 구성하며 각 상황마다 사용자가 원하는 출력을 학습함으로써 학습 기간이 종료되면 이후 각 상황마다 사용자가 원하던 출력을 자동으로 만들어 낼 수 있는 기능을 갖도록 한 것이다. 신경 회로망은 두 개의 학습 벡터 양자화 (Learning Vector Quantization), 즉 LVQ 네트워크로 구성하며 LVQ-I은 검출된 신호의 패턴을 인식하여 LVQ-II를 학습시키고 LVQ-II는 학습된 결과를 토대로 상황을 판단하는 역할을 한다.

제안한 지능형 제어 알고리즘의 유용성을 확인하기 위하여 플랜트로서 조도가 조절 가능한 조명기에 적용하여, 조명기의 주위의 임의의 상황에 대하여 사용자가 원하는 출력을 학습한 후 같은 상

황에 주어진 여러 가지 목표값들 중에서 사용자가 원하는 출력을 자동으로 발생시키는지를 시뮬레이션 및 실험을 통하여 검토하였다.

검토 결과 제안한 지능 시스템은 학습한대로 주어진 상황에 적응하여 조명기의 출력을 자동으로 만들어 냄을 확인할 수 있었다.

향후 연구 방향은 조명기 뿐만 아니라, 주어진 상황이 달라지면 새로운 입력을 설정해주어야 하는 기타 다른 장치에도 제안한 지능 시스템을 적용해 보는 것이다.

참고문헌

- [1] Hagan, Demuth, Beale, "Neural Network Design", PWS Publishing Company, 1995.
- [2] Yong-Zai Lu, "Industrial Intelligent Control" Fundamentals and Applications, JOHN WILEY & SONS, 1996.
- [3] B. D. Ripley, "Pattern Recognition and Neural Networks", CAMBRIDGE UNIVERSITY PRESS, 1996.
- [4] N. K. Bose, P.Liang "Neural Network Fundamentals With Graphs, Algorithms, and Applications", McGraw-Hill, 1996.
- [5] Madan M. Gupta, Naresh K. Sinha "Intelligent Control Systems", IEEE PRESS, 1995
- [6] Xu Young, Yan Guangqun, Chen Hexin, Dai Yisong, "A New Competitive Learning Algorithm for Vector Quantization Based on the Neuron Winning Probability", IEEE International Conference on Intelligent Processing Systems, October 28-31, Beijing, China, 1997.
- [7] Atsushi SATO, Keiji YAMADA, "A Formulation of Learning Vector Quantization Using a New Misclassification Measure", 1998, IEEE.



엄 기 환(Ki-Hwan Eom)

1972년 2월 동국대학교 전자공
학과(공학사)

1986년 2월 동국대학교 전자공
학과(공학박사)

1978년 3월~1994년 유한대학
전기공학과 교수

1994년 3월~1999년 현재 동국대학교 전자공학과
교수

* 관심분야 : 인공지능 및 자동화 시스템 설계