工學碩士 學位論文

신경회로망을 이용한 지능형 가공시스템 제어 및 모니터링에 관한 연구

A Study on Control and Monitoring of Artificial Intelligent Processing System using Neural Networks

指導敎授 李 尙 培

2001年 2月

韓國海洋大學校 大學院

電子通信工學科

趙 東 珉

목 차

Abstract

The purpose of this paper is realized the intelligent processing system using the neural networks and based on the IBM-PC interface. Generally, many of the factories have newer automation facilities. for example, the mobile factories, clothing manufactories, food processing factories, and the manufacturing process line of semiconductor etc.. In these places, the FA(Factory Automation) are activated. But the field of a primary industry is slowly going on automation. Specially, fish processing industry is required this change.

So in this paper, we propose the intelligent fish head and tail cutting processing system for the automation of fish industry.

Neural network is used to predict cutting point of fish head with data of fish length and width. And the output of neural networks is controlled precisely the position of fish cutter. Then this intelligent fish cutting processing system is inspected by IBM-PC interface I/O board, so the operating data (the position of cutter, cutting operation, conveyor of movement, detecting fish) of processing system are monitoring on IBM-PC monitor during the processing time.

This paper shows that the prediction result of cutting point of fish head by neural networks has good performance. And We proposes new monitoring method and control method by the neural networks. This system can be also controlled in remote area or control room by the operator through IBM-PC interface.

We conformed that our study had better performance than conventional one.

제 1 장 서 론

오늘날 공장 자동화가 진행되어 감에 따라 좀더 사용자 측면의 편리성이 강조되는 자동화 기기가 눈부신 속도로 개발 연구되고 있는 실정이다. 일례로 요즘 모든 자동차 생산라인, 의류 제조업, 식품제조, 반도체 공정등 실로 다양한 분야에서 생산 공정의 무인 자동화는 필연적 요소로 자리 잡아 가고 있다. 그러나 실제로 우리가 접하고 필수적으로 필요로 하는 1 차 산업 즉 농업 분야나 수산업 분야등에서는 아직도 많은 시간과 노동력을 들이는 노동 집약형 수공업형태를 못 벗어나고 있는 실정이다. 또한 사회적 분위기와 인식 부족이 맞물려 자동화 기기의 개발이 늦어지고 있거나 국산화 능력이 있음에도 불구하고 시장성의 단기적인 이윤만 내세워 많은 자동화 기기를 수입을 하고 있다. 따라서 1차 산업 분야 쪽의 자동화 기기 개발 연구에 악 영향을 미치고 있는 추세이다.

더구나 이런 추세로 가면 외국 자동화 기기에 대한 기술력 종속으로 더 큰 문제점을 유발할 수도 있을 것이다. 이에 본 연구는 지능형 실시간 가공 시스템을 IBM-PC인터페이스 기반으로 구현, 제어에 초점을 맞추었다. 가공 시스템의 대상으로는 아직도 열악한 1차 산업의 노동 집약적 수공업형태를 벗어나지 못하는 어류 전처리 과정의 아기미와 꼬리 부분의 실시간 어류 절단에 두었다.

어류의 크기는 비선형성을 많이 내포하고 있고 어족(魚族)에 따른 크기가 다양하여 여기에 일률적인 기계적 구조로써 아가미와 꼬리를 자른다면 수작업의 효과 보다 더 많은 손실을 가져오는 것은 당연하다. 따라서 본 논문은 이런 어류의 크기에 상관없이 가변적인 구조를 가지며 어류 살점 손실이 최소화되는 것을 주목적으로 실시간 가공 시스템에 신경망회로망을 적용하여 신경회로망의 우수성을 확인하였다.

그리고, 가변적인 구조를 제어할 수 있는 자동화 시스템은 인공지능적 기법(Artificial intelligence method)이 요구된다. 인공지능 기법들은 여러 가지가 있으며, 그 대표적인 기법은 퍼지논리(Fuzzy logic)와 신경회로망(Neural network)이다. 퍼지 논리는 인간의 추론 방법을 구현하기 때문에 인간의 지능을 표현하기가 매우 쉽고 복잡한 환경에서도 그 성능은 우수하다[1]. 그리고 신경회로망은 인간의 학습능력과 유사하기 때문에 패턴인식과 예측능력이 우수하다는 것을 여러 논문에서 입증하였다[2].

본 연구에서는 센서에서 받아들인 생선의 길이와 너비의 데이터를 바탕으로 신경회로망을 사용하여 절단할 생선의 아가미 부분을 예측하여 꼬리 부분과 함께 절단하는 실시간 가공시스템을 구현하였다.

그리고, 본 논문의 구성을 살펴보면 제 2장에서는 아가미 절단길이 예측을 위해 사용했던 신경망의 일반적인 개요 및 모델 알고리즘에 대해 논의하며 제 3장에서는 구현한 실시간 가공 시스템의 실제적인 하드웨어 구성 및 PC인터페이스 기반에서의 제어에 대해서 논의한다. 제 4장에서는 실시간 가공 시스템의 PC 인터페이스 I/O보드를 이용한 모니터링에 대해 논의하며, 5장에서는 이를 바탕으로 시뮬레이션한 결과를 제시하고 실험 및 고찰을 한다. 마지막으로 6장에서 결론을 맺는다.

제 2 장 신경회로망

2 .1 신경회로망의 모델

신경회로망은 인간의 학습능력을 모방하여 지식과 경험을 축척, 스스로 상황을 판단할 수 있는 능력을 가진 소프트웨어/하드웨어의 병렬 연산 시스템이라고 할 수 있다. 인간의 학습능력을 모방함으로써 인간의 사고 체계와 유사한 시스템을 구현하고자 하는 인공지능의 한 분야이다. 신경회로망의 발단은 뇌의 생물학적 신경세포(뉴런, neuron)와 그들의 연관관계를 단순화시키고 수학적으로 모델링하므로써 인간의 뇌가 나타내는 지능적 형태를 구현해 보는 것이었다. 신경회로망은 개념적으로 매우 단순하며 주어진 입력에 대해 자신의 내부구조를 스스로 조직화하여 학습해 나간다.

신경회로망에 대한 연구는 1943년 맥컬럭(McCulloch)과 피츠(Pitts)에 의해 처음으로 가능성을 제어되었다[2][3]. 이들은 인간의 두뇌를 수 많은 신경세포들로 이루어진 계산기라 생각했다. 그리고 단순한 논리적 임무를 수행하는 모델을 보여주었으며, 패턴분류가 인간의 지능적인 행위를 규명하는데 매우 중요하다는 것을 인식시켰다. 헵(hebb)은 두 뉴런사이의 웨이트(weight)를 조정할 수 있는 최초의 학습규칙을 제안하였으며, 이 학습에 관한 연구는 적응 신경회로망의 연구에 많은 영향을 끼쳤다. 1957년 로젠블렛은 퍼셉트론(perceptron)이라는 신경모델을 발표하므로서 실질적인 신경회로망에 관한 연구가 활성 되었다. 그후 민스키(Minsky)와 파퍼트(Papert)등이 퍼셉트론 모델을 수학적으로 철저히 분석하여, XOR 함수와 같이 단순한 비선형 문제를 풀 수 없다는 것을 밝혀낸 후 신경회로망에 관한 연구는 20년간 침체기를 맞게 되었다. 그러나 1980년대에 들어서 홉필드(Hopfield), 코호넨(Kohonen), 코스코(Kosko), 파커(Parker)에 의해 신경회로망이 새롭게 발전하게 되었다[3]. 오늘날 가장 일반적으로 사용하는 신경회로망의 학습알고리즘인 오류 역전파 알고리즘(Error Backpropagation Algorithm)은 웨

보스(Werbos)와 파커(Parker)에 의해 정립되었다. 이런 역사적인 배경으로 발전한 오늘날의 신경회로망은 패턴인식, 특징추출, 음성인식, 로봇이나 플랜트의 제어 시스템, 의료진단, 통신시스템 등 여러 분야에 널리 응용되고 있다.

신경회로망에 관한 연구는 두뇌와 신경세포 모델에 대한 연구에서 시작되었다. 뇌의 무게는 약 3 파운드를 조금 넘고, 100 140억개의 뉴런(Neuron)이라고 불리는 신경세포로 구성되어 있다. 뉴런(neuron)은 생체 속에서 정보 처리를 위해 특별한 분화로 이루어진 세포이다. 모든 인간의 행동은 최종적으로 작은 세포들의 활동에 기인하고 뉴런은 다른 뉴런들과 상호 연결되어 있어서 신경회로망이라고 하며, 뉴런들로 조밀하게 서로 상호 연결되어 있다. 이러한 광범위한 상호 연결은 대단히 큰 연산과 기억 능력을 제공해 준다.[4] 뉴런은 중추신경계의 기본 단위고, 생물학적 뉴런의 개략적인 모습은 그림 1에 나타내었다.

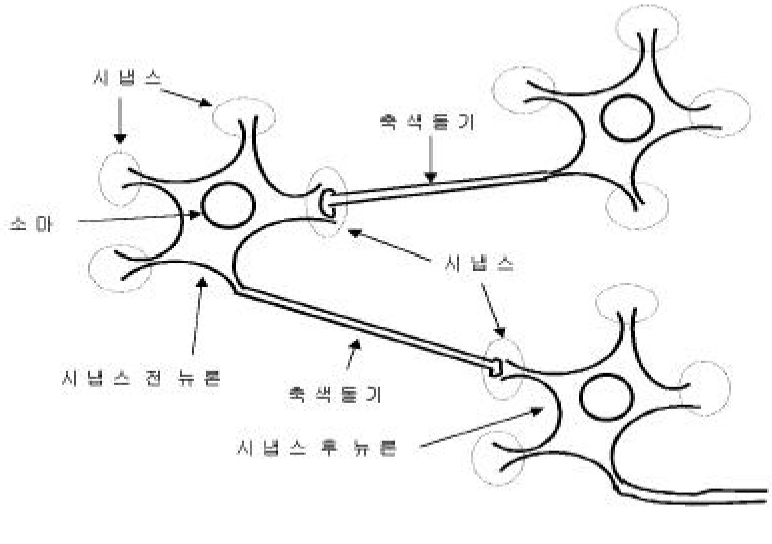


그림 1. 생물학적 뉴런의 구조.

Fig 1. Structure of a biological neuron.

그림 1에 나타난 것처럼 뉴런은 다른 뉴런으로부터 정보를 받아들이는 수상돌기(Dendrite), 수상돌기기 받은 정보를 수집 및 조합하는 세포체(Soma, Cell Body), 세포체의 결과를 다른 뉴런으로 전달하는 축색돌기(Axon)로 구성되어 있다. 그리고 한 뉴런의 수상돌기와 다른 뉴런의 축색돌기의 연결 지점을 시냅스(Synapse)하며 시냅스는 과거의 축적된 경험이나 지식 등을 기억한다[4].

크게 본다면 정보처리의 핵심이 되는 세포체, 또는 소마(soma)와 기억의 핵심이 되는 뉴런간의 연결사이에 존재하는 매우 좁은 간극 즉, 시냅스(synapse)의 두 부분으로 이루어져 있다. 정보의 처리와 전송은 시냅스 전(前) 뉴런의 활동전위 펄스가 시냅스 후(後) 뉴런의 막전위를 변화시킴으로써 정보가 수상돌기를 통해 세포체로 전달된다. 즉, 자신과 연결된 많은 다른 뉴런들로부터의 전기, 화학적 신호들을 시냅스를 통해 받아들여 종합한다. 이때 뉴런은 정보의 값이 뉴런 고유의 한계 값(임계치, Threshold)보다 커지면 뉴런은 발화(Fire)되어 다른 신경세포에 자신의 출력을 전달한다. 이러한 뉴런을 인공적인 뉴런으로 모델링하면 그림 2와 같이 된다[5][6]

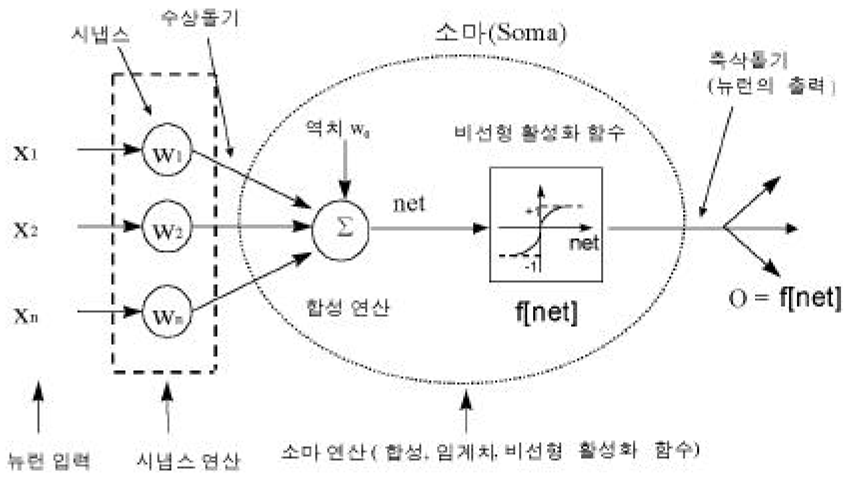


그림 2. 인공적인 뉴런의 구조

Fig 2. Structure of a artificial neuron

생물학적 뉴런의 성질이 복잡하고 다양하기 때문에 그 모델 특성을 요약하기가 어렵다. 뉴런은 다른 뉴런 또는 외부로부터 입력을 받으며 그 입력들의 가중된 합은 그림 2에서 나타낸 비선형 활성화 함수로 입력된다. 그리고 입력들의 가중된 합이 정해진 뉴런의 임계치를 넘었을 때 뉴런이 점화(firing)되어 다른 뉴런으로 출력된다. 이 뉴런 함수를 수학적으로 표현하면 식(1)과 같다.

$$O = f\left[\sum_{i=1}^{n} w_i x_i - w_0\right] \quad (1)$$

여기서 각각의 $x_1, \ldots, x_n$는 뉴런의 입력, $w_1, \ldots, w_n$는 각 뉴런 사이의 시냅스 가중치(웨이트), O는 뉴런의 출력, w_0는 임계치(역치), f 는 비선형 활성화 함수이다.

식(1)을 간략히 설명하자면 뉴런의 입력과 각 입력에 대응하는 시냅스 가중치(웨이트)를 곱하여 합한 값 $[\sum_{i=1}^{n} w_i x_i]$에 함수관계 f를 적용한 값을 출력한다. 여기서 w_0 의 임계치(역치)로써 $[\sum_{i=1}^{n} w_i x_i]$이 임계치 보다 작을 때는 뉴런이 활성화되지 않도록 역할을 한다.

신경회로망에서의 사용되는 활성화 함수는 그림 3에 나타내었고, 보편적으로 사용되는 대표적인 활성화 함수는 단 극성 선형함수(unipolar linear function), 양극성 선형함수(bipolar linear function), 양극성 계단함수(bipolar step function), 단 극성 시그모이드 함수(unipolar sigmoid function), 양극성 시그모이드 함수(bipolar sigmoid function)등이 있다[1].

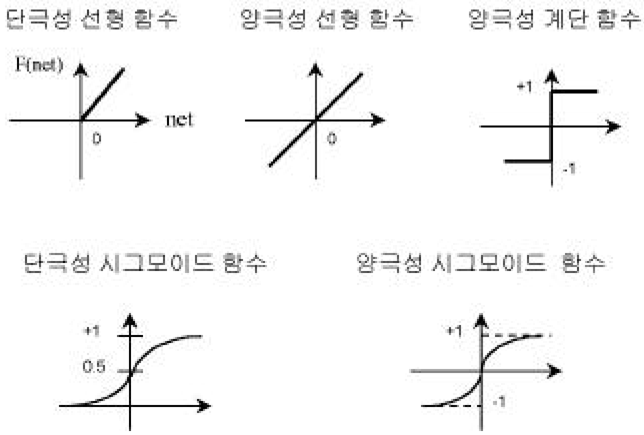


그림 3. 뉴런의 활성화 함수

Fig 3. Activation functions of neuron

대부분의 경우에는 제한(制限)된 단층신경망의 능력에 비해 다층 신경회로망의 사용 이점을 얻기 위해 비선형 활성화 함수(nonlinear activation function)가 사용된다. 본 연구에서 사용된 활성화 함수는 단 극성 시그모이드 함수로 수학적 수식은 (2)와 같다.

$$f(x) = sigmoid(x) = \frac{1}{1 + \exp(-\frac{x}{T})}, T > 0 \quad (2)$$

여기서, T→∞가 되면 활성화 함수는 선형 함수가 되며, 만약 T→0 으로 접근하면 sign 함수가 된다.[8]

2 .2 다층 신경회로망의 학습과 구조

인간의 뇌는 수많은 뉴런으로 서로 연결되어 있다. 따라서 인간의 뇌와 유사한 인공적인 신경회로망은 다층구조라 할 수 있다. 서로 연결된 뉴런에 의해서 더욱 더 좋은 성능으로 수행할 수 있다. 뉴런을 층에 배열하는 것은 뇌의 일부분인 계층화된 구조를 흉내낸 것이다. 시스템 인식과 예측, 제어와 같은 응용에서 가장 일반적으로 사용되는 신경회로망 구조는 오류 역전파 알고리즘을 갖는 다층 신

경회로망(Multlayered Neural Nework)이다. 전형적인 다층 신경회로망은 그림 4에 나타내었다.

그림 4에서 각각의 원(또는 node)은 그림 2에서 보여준 뉴런이다. 이 신경회로망은 x라는 입력 벡터를 갖는 입력 층과 y라는 출력 벡터 갖는 출력 층으로 이루어지며 입력 층과 출력층 사이의 층을 우리는 보통 은닉층(Hidden layer)이라 한다.

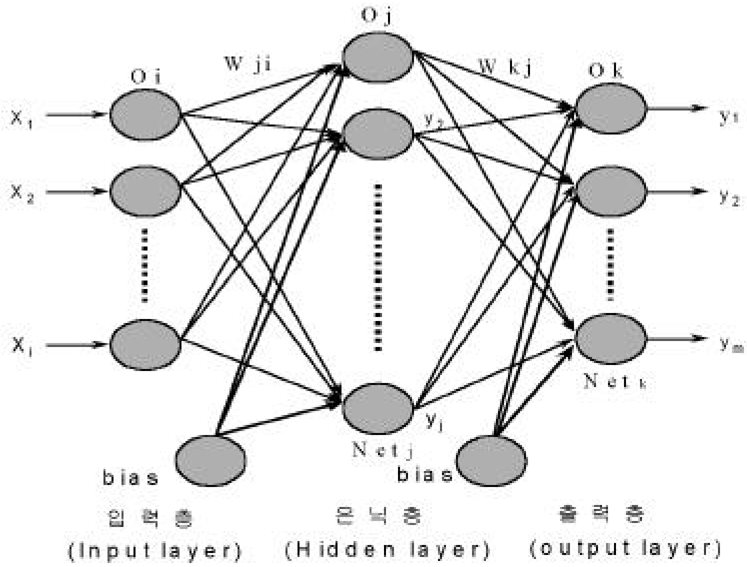


그림 4.일반적인 다층 신경회로망의 구조

Fig 4. Mutilayer Neural Networks

O_i, O_j, O_k는 입력층, 은닉층, 출력층들의 각 뉴런 출력이고, 입력 층과 은닉층 사이의 웨이트를 w_{ji}, 입력 층과 은닉층 사이의 웨이트를 w_{kj}로 표기한다. 모든 정보는 신경회로망의 웨이트에 저장되며, 학습과정 동안 웨이트 w_{ji}, w_{kj}의 성분은 계속적으로 새로운 정보로 바뀌어 진다. 일반적으로 웨이트의 새로운 정보를 변경하는 신경회로망의 대표적인 알고리즘은 오류 역전파 알고리즘이다. 오류 역전파 알고리즘은 신경회로망의 각 뉴런에 의해 계산된 최종 출력층 뉴런의 출력과 바라는 출력사이의 오차를 자승하여 최소화시키는 최소-평균 자승법(least-mean square)이다.

오류 역전파 학습 알고리즘의 원리는 먼저 입력 층에서는 신경회로망의 입력 x를 은닉 층으로 보낸다. 두 번째로 은닉 층의 뉴런들은 각각의 입력 층으로부터 입력된 값과 웨이트들의 곱을 합산함과 동시에 활성화함수를 통해 연산된 결과를 출력 층으로 보낸다.

출력 층은 은닉 층과 같은 뉴런 연산을 하여 출력한다. 이때 신경회로망의 출력 값이 바라는 목표 값과의 차이를 구하며, 이 차를 오차라고 말한다. 이 오차를 최소화 하기 위해 각 층에 있는 웨이트의 오차 벡터 항을 편 미분하여 웨이트를 조정한다.

다시 말해 출력 층의 출력과 바라는 목표치의 오차를 연산한 후 출력 층에서 은닉 층으로, 은닉 층에서 입력 층으로 역전파하여 오차에 따른 웨이트 변화량에 의해 웨이트들을 조정하며 이것을 오류 역전파라고 말한다.

오류 역전파 학습 알고리즘을 수식적으로 나타내면 다음과 같다.
먼저 입력 층, 은닉 층, 출력층의 뉴런 출력은 식(4), 식(6), 식(8)과 같다.

$$net_i = x_i \quad (i = 1, 2, 3, \ldots, n) \tag{3}$$

$$O_i = \lambda f(net_i) \tag{4}$$

$$net_j = \sum_j W_{ji} O_i \tag{5}$$

$$O_j = \lambda f(net_j) \tag{6}$$

$$net_k = \sum_k W_{kj} O_j \tag{7}$$

$$O_k = \lambda f(net_k) \tag{8}$$

여기서 사용한 f 는 활성화 함수는 단 극성 시그모이 함수를 사용한다.

$$f(x) = \frac{1}{1 + \exp(-\lambda x)} \tag{9}$$

net_i, net_j, net_k는 이전의 뉴런 출력과 현재 층에 있는 웨이트들과의 곱을 합산한 값이고 λ는 활성화 함수의 기울기이다 오류역전파 알고리즘을 설명하기 위해 편의상 λ는 1로 둔다. 신경회로망을

학습시키기 위해 신경회로망의 출력 값이 바라는 목표 값과의 차 즉, 오차를 구해야 하며 이 오차를 구하는 수식은 식(10)에 나타내었다.

$$E = \frac{1}{2}\sum_{k}(D_k - O_k)^2 \qquad (10)$$

학습의 목적은 웨이트를 조정하여 오차 E를 최소화하는 것이므로 웨이트의 조정에 대하여 살펴보면 오차를 최소화하기 위해 웨이트를 음의 경사방향(negative gradient direction)으로 변화시켜 한다. 따라서 웨이트 변화를 음의 경사 방향으로 오차에 대한 웨이트의 방향 벡터를 편 미분함으로써 웨이트 변화량을 구할 수 있다.
먼저 출력 층의 웨이트 변화량을 구하면 다음과 같다.

$$\triangle W_{kj} = -\eta\frac{\partial E}{\partial W_{kj}}, \qquad \eta > 0 \qquad (11)$$

여기서 η는 학습 속도를 나타내는 상수이며 이것을 학습률이라 한다. 따라서 일반화된 오차 신호는 다음과 같다.

$$\delta_k = -\frac{\partial E}{\partial Net_k} \qquad (12)$$

그리고 식(12)을 연쇄 규칙(chain rule)을 사용하여 아래와 같이 간단하게 쓸 수 있다.

$$\frac{\partial E}{\partial W_{kj}} = \frac{\partial E}{\partial net_k}\frac{\partial net_k}{\partial W_{kj}} = -\delta_k\frac{\partial net_k}{\partial W_{kj}}$$

$$\frac{\partial net_k}{\partial W_{kj}} = \frac{\partial(\Sigma W_{kj}O_j)}{\partial W_{kj}} = O_j \qquad (13)$$

(11)식에 의해 웨이트 변화량 $\triangle W_{kj}$는 다음의 식과 같다.

$$\triangle W_{kj} = \eta \delta_k O_j$$

$$\delta_k = O_k(1 - O_k)(D_k - O_k) \tag{14}$$

오차신호 δ_k**는 다음과 같이 계산된다.**

$$\delta_k = \frac{\partial E}{\partial net_k} = -\frac{\partial E}{\partial net_k} = -\frac{\partial E}{\partial O_k}\frac{\partial O_k}{\partial net_k} \tag{15}$$

식(9)**의 활성화 함수의 미분과 식**(10)**의 미분으로 다음과 같이 식** (15)**을 다시 쓸 수 있다.**

$$\delta_k = \frac{\partial E}{\partial net_k} = -\frac{\partial E}{\partial net_k} = -\frac{\partial E}{\partial O_k}\frac{\partial O_k}{\partial net_k}$$

$$\frac{\partial E}{\partial O_k} = -(D_k - O_k) \tag{16}$$

$$f'(net_k) = \frac{\partial O_k}{\partial net_k} = \frac{e^{-x}}{(1+e^{-x})^2}$$

$$= \frac{1}{(1+e^{-x})}\left(1 - \frac{1}{(1+e^{-x})}\right) \tag{17}$$

$$= f(net_k)(1 - f(net_k))$$

은닉 층에 대한 웨이트의 변화량도 오차를 최소화하기 위해 웨이트를 음의 경사 방향(negative gradient direction)**으로 변화시켜 준다.**

$$\triangle W_{ji} = -\eta\frac{\partial E}{\partial W_{ji}}, \quad \eta > 0 \tag{18}$$

식(18)**을 연쇄 규칙**(chain rule)**을 사용하여 아래와 같이 간단하게 쓸 수 있다.**

$$\frac{\partial E}{\partial W_{ji}} = \frac{\partial E}{\partial net_j}\frac{\partial net_j}{\partial W_{ji}} \tag{19}$$

$$\delta_j = -\frac{\partial E}{\partial net_j} \tag{20}$$

식(20)은 은닉 층에서의 오차 신호이다. 식(5)에 의해서

$$\frac{\partial net_j}{\partial W_{ji}} = O_i \qquad (20)$$

여기서, 입력 층의 출력값 O_i 는 신경망 입력값 x_i 와 같다. 즉, $O_i = x_i$ 이다. 그러므로 입력 층과 은닉층사이에 있는 웨이트의 변화량은 다음의 식으로 나타내어진다.

$$\triangle W_{ji} = \eta \delta_j O_i \qquad (21)$$

식(18)에서의 은닉 층의 오차신호 δ_j는 다음과 같이 계산된다.

$$\delta_j = -\frac{\partial E}{\partial net_j} = -\sum_k \frac{\partial E}{\partial net_k} \frac{\partial net_k}{\partial O_j} \frac{\partial O_j}{\partial net_j} \qquad (22)$$

다시 식(22)는 식(7), 식(8), 식(11)으로부터 다음과 같이 계산된다.

$$\begin{aligned} \delta_j &= -\sum_k \delta_k W_{kj} f'(net_j) \\ &= O_j(1-O_j)\sum_k \delta_k W_{kj} \end{aligned} \qquad (23)$$

따라서 웨이트의 변화는 다음과 같다.

$$W_{ji} = W_{ji} + \triangle W_{ji} \qquad (24)$$

$$W_{kj} = W_{kj} + \triangle W_{kj} \qquad (25)$$

다음의 그림 5는 지금까지 언급해 왔던 오류역전파 알고리즘의 순서도를 제시함으로써 오류역전파 알고리즘을 요약한다.

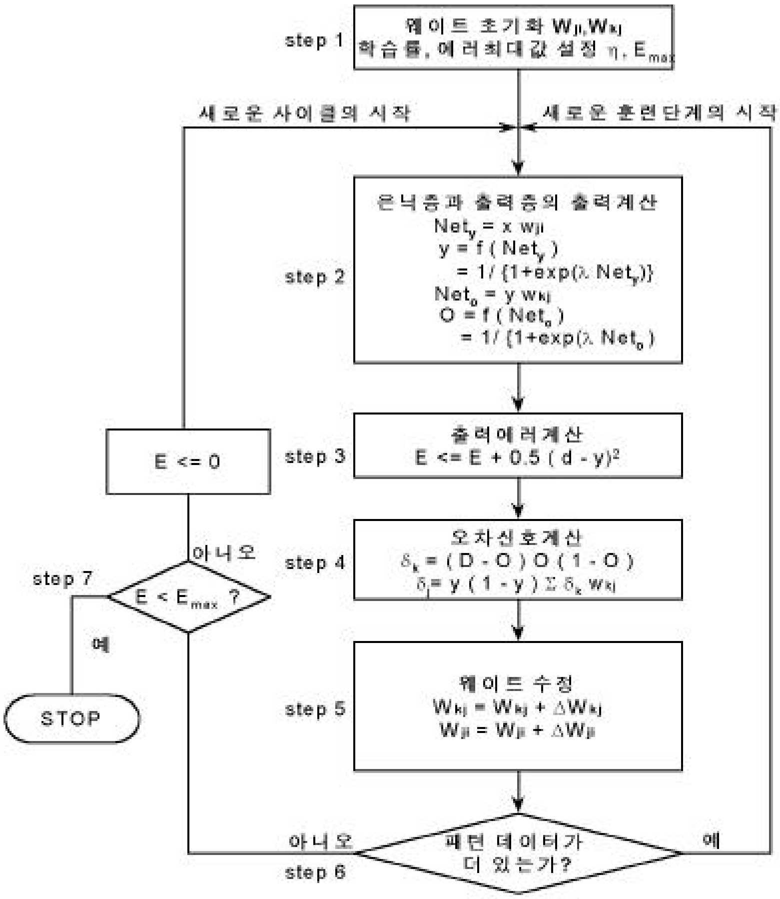


그림 5 오류 역전파 학습 알고리즘의 순서도

Fig 5. Error Back Propagation Training Algorithm

지금까지 설명했듯이, 오류 역전파 알고리즘은 오차 신호를 계산하고 신경회로망의 웨이트들을 조정하기 위해서 바라는 응답값이 필요하다. 이러한 초기의 학습 후에 신경회로망은 학습에 사용되지 않은 새로운 데이터의 집합을 입력할 수 있다.

학습된 데이터의 집합이 아닌 다른 데이터에서도 정확하게 응답하는 것을 일반화(generalization) **능력이라 한다. 그리고 이것은 곧 신경회로망의 신뢰도를 가리킨다. 학습과 신경회로망의 성능을 검사한 후에, 신경회로망은 패턴 분류기, 또는 잘 모르는 비선형 함수와**

복잡한 처리를 모델 화하는데 사용될 수 있다.

오류역전파 알고리즘의 학습의 요소에는 초기 웨이트, 누적웨이트 조정과 증분의 갱신, 활성화 함수의 기울기, 학습률 등을 들 수 있다. 신경회로망을 학습시킬 때 초기 웨이트는 작은 무작위 값으로 설정하는 것이 일반적이다. 그러나 초기화는 최종 출력에 영향을 많이 미친다. 초기 웨이트가 지역 최소 점에 빠졌다면 웨이트가 고정된 값에서 머물게 되므로 다른 지점에서 학습을 다시 시작해야 한다.

그림 6에서는 초기웨이트와 오차함수와의 관계를 그래프로 표현하고 있다. 전술한 바와 같이 오차함수가 지역최소점에 빠지는 경우는 a, b에서 시작했을 때에 해당되며 c지점에서 다시 학습을 시작하면 원하는 값을 얻을 수 있다. 보통 일반적으로 초기 웨이트는 -0.5에서 0.5사이의 값을 주로 사용하거나 설계자가 빠른 학습을 유도하기 위해서 임의의 랜덤한 범위의 초기웨이트를 임의로 설정하여 사용하기도 한다.

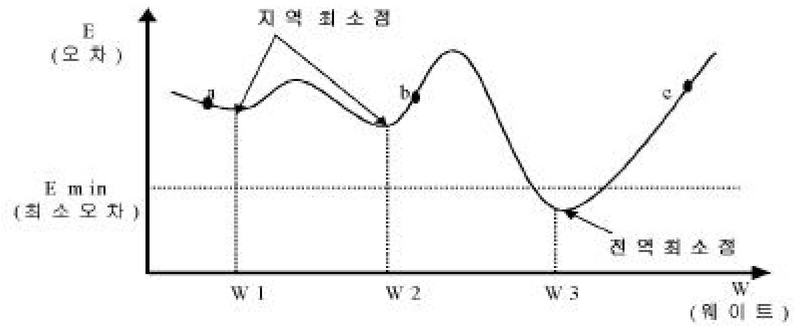


그림 6 웨이트와 오차함수의 관계

Fig 6. Relation between Weight and Error Function

또한 각 학습 패턴에 따라 웨이트를 조정하는 방식 즉, 증분갱신 (Incremental Updating) 으로 학습률이 충분히 작다는 가정하에 식 (10)는 모든학습 패턴이 종료한 에러의 변화만큼 웨이트를 조정한다.

$$E = \frac{1}{2}\sum_{k}(D_k - O_k)^2 \qquad (10)$$

그리고 출력의 수렴 정도는 학습률에 의해서도 달라질 수 있다. 학습률은 신경회로망의 구조와 응용 목적에 따라 다르고, 일정한 기준이 없으며 보통 0.001에서 10사이의 값을 사용한다[20].

급격하고 좁은 오차 최소 점을 갖는 응용에서는 큰 값의 학습률을 선택하면 빠르게 학습이 진행 될 수 있지만 무한대의 발산 즉, 오버슈트가 일어나 오차 최소 점(전역 최소 점)에 수렴하지 못해 학습이 안 되는 상황이 발생할 수 있다. 반면에 너무 작은 학습률을 선택하면 오차가 적어지는 형태로 학습이 이루어져서 최종적으로 오차 최소 점에 수렴하지만 각 학습단계에서의 연결강도 변화량이 아주 미세하여 전체 학습시간이 매우 길어지는 단점이 있다.

따라서 원하는 응용 목적에 여러 가지 값의 학습률을 선택하여 신경망을 여러 번 학습한 그 결과를 비교하여 최적의 성능을 내는 학습률을 선택하는 것이 바람직 하다.

마지막으로 신경회로망의 학습에 중요한 요소로 활성화 함수의 기울기를 들 수가 있다. 단 극성 시그모이드 함수 및 양극성 시그모이드 함수는 기울기 λ에 의해 특성이 결정되고 활성함수의 미분치는 위의 수식에서 보았듯이 에러신호 δ를 만드는 구성 요소이므로, 이 두 요소는 신경망의 학습속도에 영향을 미친다.

일반적으로 학습률과 활성함수의 기울기 λ값을 같이 조정하기보다는 λ는 1로 고정시켜 놓고 학습상수(학습률)로 학습속도를 조정하는 것이 신경회로망의 학습에 바람직 하다.[11][15]

제 3 장 지능형 가공시스템의 제어

3.1 지능형 가공 시스템 구성

현재 생선 전처리 공정에서의 시스템은 크게 두 부부으로 나뉜다. 어류 즉 명태의 크기가 일률적으로 특정한 값을 갖지 않고 다양한 크기를 가지므로 첫 번째 시스템은 사람의 손으로 아기미와 꼬리를 절단하는 수공업 형태의 시스템과 회전 톱날이 특정위치에 고정되어 컨베이어로 이송되어 오는 명태를 항상 같은 길이로 아기미를 자르는 기계화된 시스템이 국내(수산업 협동조합 물류센터 內 명태 가공공장)에 보급되어 있다.

첫 번째의 시스템의 경우는 완전히 사람의 노동력에만 의존한다. 노동집약형의 수공업 형태로 사람이 할 수 있는 시간당의 일에 대한 비효율성, 명태의 내장이나 찌꺼기 등의 배출 물로 인한 작업환경의 열악성, 노동집약형의 형태이므로 비싼 임금 등의 문제점이 제기된다. 두 번째 시스템의 경우 또한, 아가미의 편차를 고려하지 않고 고정위치에서 명태의 아기미를 절단하는 시스템의 경우 역시 어류의 크기를 무시한 채 일정하게 절단하기 때문에 고기의 살점 손실의 문제점을 야기 시킨다. 따라서, 본 논문에서 제시하는 실시간 가공 시스템은 생선 아가미 꼬리부분 절단 모듈 기능으로 어종의 종류와 크기에 전혀 구애를 받지 않고 인공지능적 기법 즉, 신경회로망을 사용하여 생선의 아가미와 꼬리를 알맞는 길이로 절단하는 것을 주(主) 기능으로 한다.

전반적인 시스템의 구성은 크게 기계적인 구조, 센서, IBM-PC인터페이스, 모니터링 부분으로 나뉜다. 그림 7은 지능형 가공시스템의 기계적인 구성으로 컨베이어 부분과 절단 실린더와 전면의 센서부를 나타내며 그림 8은 PC인터페이스 기반의 원거리 조작기능 즉, 운전자가 절단모듈의 공정을 직접 눈으로 확인하는 것이 아니라 사무실이나 중앙 제어실에서 모니터링하는 화면을 나타내고 있다.

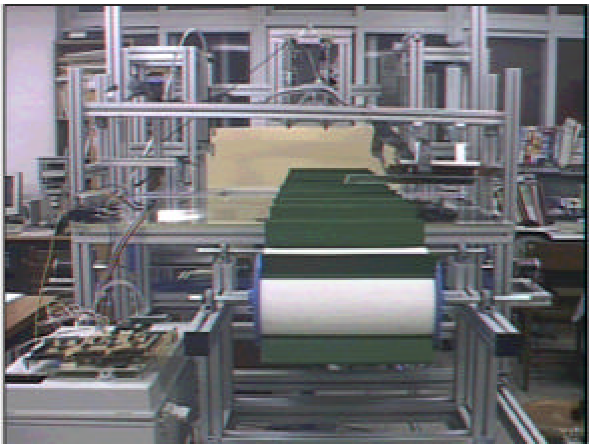


그림 7 **지능형 가공 시스템 기계적 구조**

Fig 7. Machine structure of a artificial processing system

그림 8. I/O**보드를 통한 모니터링 화면**

Fig 8. Monitoring through I/O board

3 .2 지능형 가공 시스템의 기능

3 .2 .1 센서 기능

전체 길이 검출 적외선 센서는 외부에 적외선 센서 드라이브단을 구성하여 자체적인 동작으로 어류의 전체 길이를 검출한다. 명태의 표본조사에 의한 꼬리지느러미의 절단 가변 길이는 35㎝정도의 범위를 보였다. 이에 수광소자(EL-7L)과 발광 소자(ST-7L)를 등간격(等間隔) 어레이형태로 배열하여 데이터 취득시 연산의 용이성을 부여했다. 동작순서를 열거하면 먼저, 이송 컨베이어의 동작신호를 PC-I/O보드에서 감지하면 센서 발진신호를 적외선 어레이센서 드라이브단에 인가하여 구동(즉, 꼬리지느러미부 센싱시작)하게 되며, 48개의 어레이가 각각 6개씩 한조로 하드웨어적으로 구성을 한다.

즉, 각조의 1번은 1번끼리, 2번은 2번 끼리 연결되어 있어 발광적외선 센서의 간섭을 피하도록 하였고, 공통으로 묶여 있는 센서를 쉬프트 하는 효과와 마찬가지로 8번을 번갈아 ON/OFF 하게되면 74LS574 버퍼쪽에서 데이터를 읽음으로써 한 라인의 데이터를 센싱하게 된다(센싱의 한 주기). 그때, 각각의 데이터를 정해진 메모리상에 저장하고 각각의 비트별 연산을 가하여 꼬리의 전체길이와 꼬리에서 몸통으로 이어지는 부분의 가장 잘록한 부분(칼날의 목표부분)을 계산하고 그 결과를 저장하게 된다. 센싱의 정지신호는 이송 컨베이어 신호가 정지신호를 발생할 때와 꼬리센싱중 데이터가 연속적(5센싱주기이상)으로 값이 없거나 전체 비트에 비해 한,두 비트정도의 값만 인식하는 등의 잡음이라고 생각되는 경우가 발생했을 경우로 구성했다.

너비 센서 또한 적외선 광센서를 이용하였다. 전체길이를 감지하는 센서와 거의 같은 동작을 한다. 컨베이어 우측에 적외선 광센서를 마찬가지로 등간격으로 달아 데이터의 값에 따른 너비에 대한 길이를 감지하도록 했다.(부록 5)

3 .2 .2 모터 위치 제어 기능

생선 아가미와 꼬리 절단실린더 이동용 모터의 위치제어는 생선이 컨베이어로 이송되어 올 때 컨베이어 측면 기준 점에 생선의 머리 부분이 닿아 오도록 설계되어 있어 그 점을 기준으로 절단용 실린더 운반 모터의 수평거리 즉, 아가미의 절단 길이 까지 이동하게 된다. 모터의 이동거리 계산은 3상 로터리 엔코더를 사용하여 이동거리 명령을 주었다. 모터가 이동하기 위해서 회전을 시작하면 엔코더는 회전에 따른 구형파 펄스를 발생시켜 이것을 I/O보드 내의 PIT (Programmable Interval Timer) 8253(intel) 범용 타이머 칩으로 펄스를 전달한다. 8253 카운터에는 1회전의 펄스당 모터가 얼마만큼 이동하는지에 대한 데이터를 계산하여 16bit 2진카운터 또는 BCD 10진수의 형태로 프로그램 된다. 그림 9은 PIT 8253의 카운트 동작에 관한 블록다이어 그램과 파형이다. 8253의 GATE신호가 HIGH로 고정된 상태에서 펄스가 들어오면 8253은 다운 카운팅에 들어가서 프로그램화되어 있는 값만큼 펄스를 계수하고 8253의 OUT단자를 HIGH로 세트 시키고, 이 신호에 의해서 절단용 실린더 운반 모터는 정지한다.

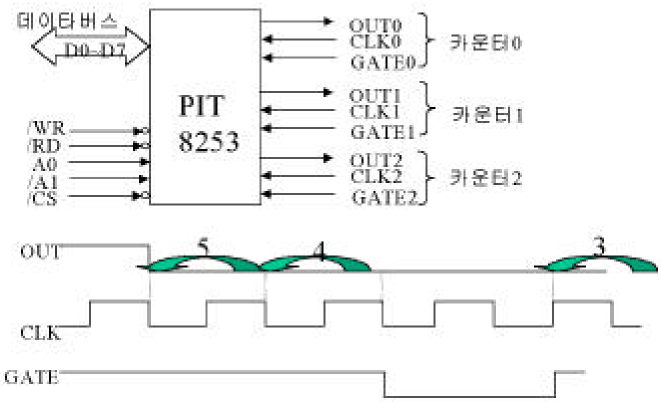


그림 9. 8253 0모드에서의 카운터 방식

Fig 9. Counter operation of 8253 in mode 0

PIT8253은 총 6개의 모드(모드 0 5)로 3개의 카운터를 독립적으로 사용할 수 있다. 여기서 모터의 펄스를 카운트하기 위해서 8253 모드 0을 사용했다. 그림 9에서 보면 알 수 있듯이 GATE신호가 HIGH일 때만 엔코더의 클럭을 계수하고 GATE가 LOW로 떨어지면 계수가 중단되었다가 다시 HIGH로 되어야 계수를 한다. 이후 PIT8253을 초기화시키지 않으면 OUT단자는 HIGH를 계속 유지하게 된다.[12][24]

엔코더는 Autonics사의 광학식 로타리 엔코더 ENB-25-3-1 (1회전당 25펄스)를 사용했다. PC-I/O보드 외부에 엔코더의 펄스를 2체배하는 회로를 74LS14(슈미트 트리거), 74LS74(D type flip-flop), 74LS86 (Exclusive OR)소자를 사용하여 1회전당 50펄스가 나오도록 회로를 구성했다.

모터 드라이브 단은 National Semiconductor 사의 LMD18200 전용칩을 사용하였다. 그림 10은 LMD 18200칩에 대한 동작과 내부구조에 관하여 언급하고 있다. 내부구조로는 DMOS타입 스위칭 소자를 H-Bridge형태의 회로로 구성되어 있고. 모터 드라이브단의 동작은 3개의 enable 단자 즉, Direction, PWM, Break 의 조합으로 모터를 정회전 또는 역회전. 정지를 시킨다. 8253의 OUT단자는 Break단자로 연결되어 모터의 정지신호를 만든다.

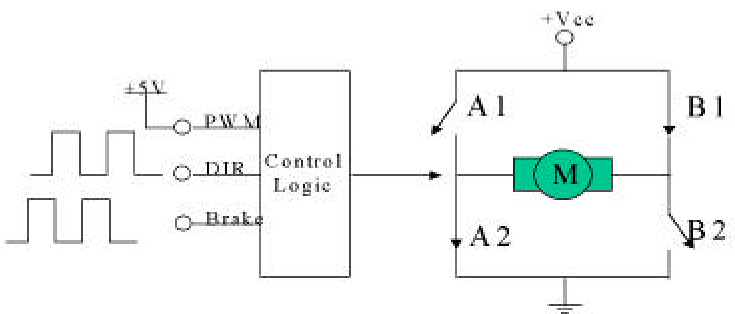


그림 10. LMD18200 의 동작과 내부구조

Fig 10. Operation and Inner structure of LMD18200

생선 절단용 실린더 이동용 모터는 수평방향으로 움직이기 위해 리드-스크류로 모터의 회전력을 직선운동으로 바꾼다. 시스템 구현상 모터 1회전당 리드-스크류는 2mm씩 움직이도록 되어 있다. 센

싱된 생선의 전체길이와 너비를 바탕으로 모터는 이동해야 할 위치를 8253에 기억을 시키고 엔코더에 의해 위치 값이 결정이 되면 PC-I/O인터페이스 보드의 8255의 병렬 입출력 포트를 통해서 모터의 전진방향을 결정한다. 또한 빠른 모터의 응답성을 얻기 위해 모터의 위치(변위)는 다음과 같이 계산한다. 그림 11는 모터의 위치제어에 대한 블록다이어 그램이다. 엔코더의 동작회로도와 모터의 구동회로는 부록(부록 1 2)에서 제시한다.

변위(Y)=Sensing데이터 - X(초기 모터 위치)
변위 만큼 이동 후 그 초기 모터위치 X =0
현재 모터 위치(X1) = Sensing데이터-(X+Y)

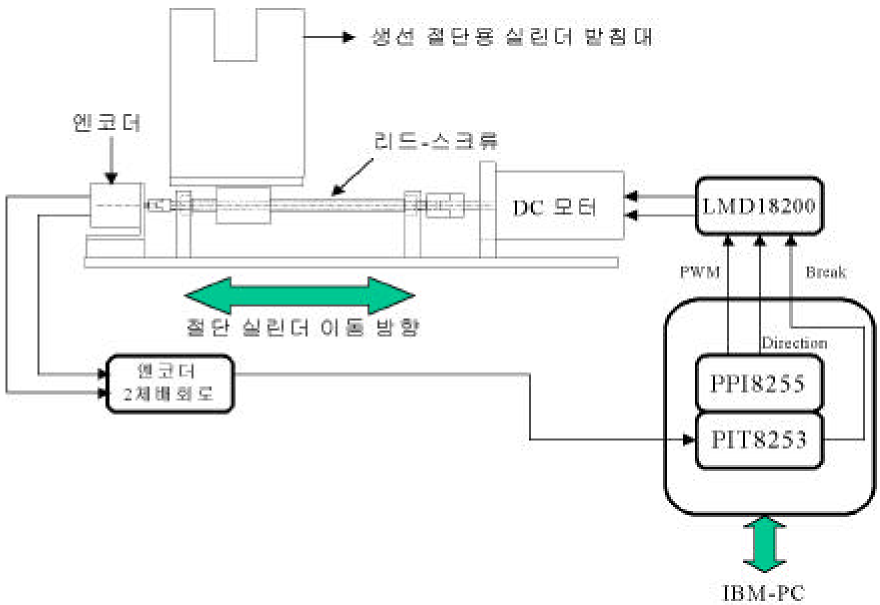


그림 11. 모터 위치제어 불럭다이어 그램
Fig 11. Block Diagram of a Motor Position Control

3 .2 .3 시퀜스 기반의 공압 실린더 제어 기능

실제적인 가공시스템의 작동 즉, 컨베이어의 정지, 생선고정, 생선 아가미 및 꼬리 절단의 일련의 과정은 시퀜스 제어 방식을 본 논문에서는 사용한다. 즉, IBM-PC인터페이스 I/O보드는 생선의 가변적인 크기에 대해 최적의 절단 위치를 찾기 위해 생선 절단용 실린더 이동용 모터 제어를 전담하게 하며 또한 공정의 효율성을 높히기 위해 최종적 처리는 I/O보드와는 독립적인 스퀜스 제어 회로를 구성하여 실시간 가공시스템의 현장성과 경제성을 높혔다.

생선 고정 실린더, 생선 절단 실린더는 공압식 솔레노이드 밸브를 사용하였고 밸브의 접점회로는 릴레이를 사용하여 솔레노이드 밸브를 구동시켜 실린더를 동작 시켰다. 그리고 생선 고정 실린더, 절단 실린더의 동작에는 스퀜스 타이머 회로를 사용하여 고기 고정 후 약간의 지연을 둔 후 칼날이 생선의 아가미와 꼬리를 자르도록 설계했다. 생선의 절단 과정에서 생선 검출 센서(광센서)를 사용하여 생선이 절단위치에 정확하게 위치하게 되면 이 검출 센서의 입력으로 시퀜스 동작이 이루어지게 설계했다.

또한 생선의 아가미 절단 부분의 칼날은 V자형으로 하여 일자형의 칼날보다 어육의 손실이 적도록 했다. 꼬리 부분은 일반적인 일자형 칼날을 사용했다. 그림 12에서는 실시간 가공 시스템의 전체적인 제어 유닛 구성을 나타내고 있다.

그림 12에서 보는 바와 같이 생선이 컨베이어 위에 올려진 상태에서 이송을 시작하게 되면 생선 전체길이 감지 센서와 너비감지 센서에 의해 절단하게 될 생선의 아가미 절단 지점과 꼬리 절단 지점을 획득하게 된다. 이때 아가미 절단 실린더 이송용 모터 및 꼬리 절단 실린더 이송용 모터는 그 센서에 의해 계산된 값만큼 전진 또는 후진하여 아가미 및 꼬리 부분의 절단 부분까지 실린더 이동을 완료하고 생선이 컨베이어에 의해 이송되어 오는 것을 생선감지 센서가 감지를 하게 된다.

즉, 아가미 및 꼬리 부분 절단용 실린더는 이미 이동 완료한 상태에서 생선이 감지가 되면 아가미 및 꼬리 절단 부분 실린더가 동작을 하여 생선 아가미 및 꼬리 부분을 절단을 하고 다음 생선의 전체길이 및 너비 데이터를 바탕으로 다시 계산된 절단 부분까지 모터에 의해 아가미 부분 및 꼬리 부분 절단 실린더가 이동되어 다음 동작을 연속적으로 처리하게 된다. 생선의 아가미 및 꼬리 부분의 절단동작은 먼저 고기고정 실린더가 동작을 하여 절단동작시 생선이 움직이는 것을 방지하며, 칼날이 아가미 부분 및 꼬리 부분을 절단하게 된다.

시퀜스 동작의 일련의 동작을 그림 13에 제어흐름 블럭선도로 간략하게 정리하여 나타내었다.

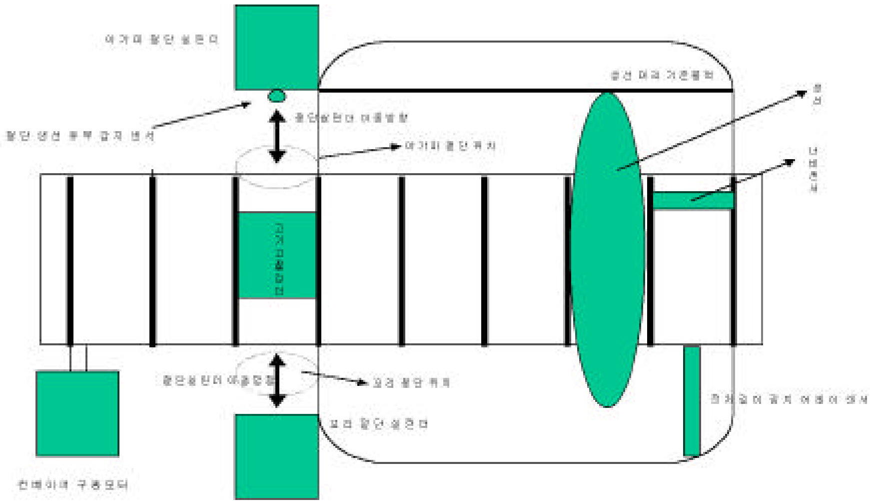


그림 12. 지능형 가공 시스템의 제어유닛 구성

Fig 12. Control-Components of a Artificial Processing System

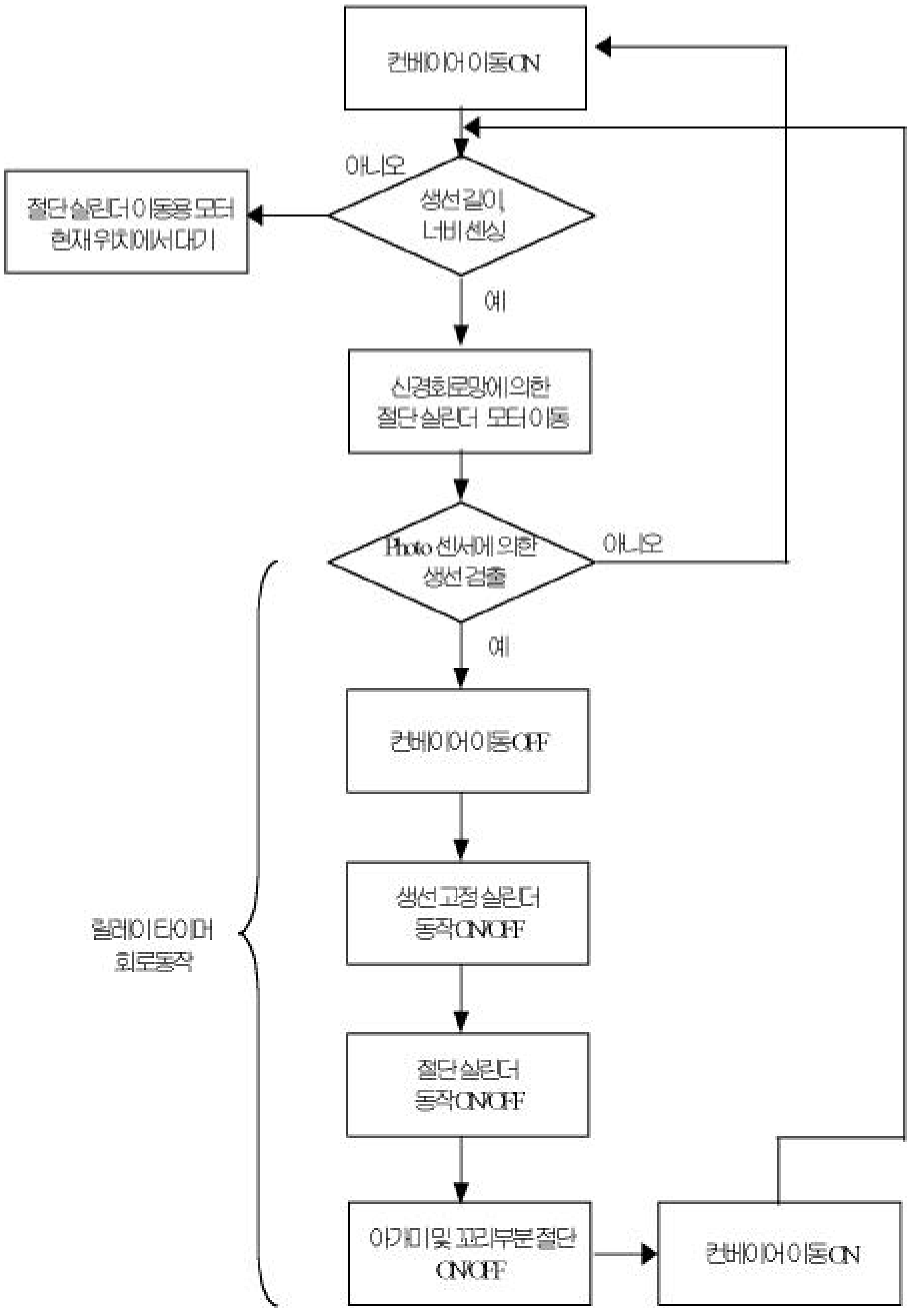


그림 13. **지능형 가공시스템의 전체적인 제어 흐름 블록 선도**

Fig 13. Entire Control Flow Chat of a Processing System

제 4 장 지능형 가공시스템의 실시간 모니터링

4 .1 IBM-PC 인터페이스 기반의 지능형 가공 시스템의 모니터링 구조

본 연구에서는 일반적인 가공 시스템의 결점을 보완하며 최소한의 노동력으로 생선의 아가미와 꼬리를 절단하는 실시간 가공 시스템 공정 관리하기 위해 IBM-PC인터페이스 기반 시스템을 구축했고 생선아가미 및 꼬리 자동 절단 가공 시스템에 모니터링 기능을 부가하였다.

관리자 또는 운전자가 각 제어 모듈기기 별로 진행 공정상황을 감지하고 적절한 조치를 취할 수 있게 하도록 PC-I/O인터페이스 보드를 PC외부에 구성하여 모니터링 기능과 함께 각 제어 유닛에 적절한 출력 값을 주거나 각각의 제어유닛에서 입력을 받는 I/O인터페이스 기반 시스템을 갖추어 운전자 또는 관리자는 효율적인 공정관리를 할 수 있게 된다.

PC-I/O인터페이스 기반의 실시간 가공 시스템의 생선 아가미와 꼬리 절단 모듈은 효율적인 공정관리와 좀더 정밀한 절단 시스템과의 인터페이스(Interface) 즉, 디지털 영상처리 카메라를 센서로 사용하는 시스템 기반의 절단모듈과의 연결이 다른 부가장치 없이 IBM-PC와 바로 맞물릴 수 있는 인터페이스의 장점이 있다.

그림 14에서는 전반적인 지능형 가공 시스템의 I/O구조를 나타내고 있다. 살펴보면 알 수 있듯이 실시간 가공 시스템의 전반전인 진행 사항을 on-line상에서 PC 모니터로 한눈에 관측할 수 있다.

본 연구에서 제안한 지능형 가공 시스템은 생선크기에 차이가 있어도 지능적으로 예측, 편차가 나는 만큼 보정한 다음 생선 아가미와 꼬리 절단용 실린더 운반 모터를 움직여 운전자가 원하는 생선 아가미 와 꼬리 절단부분에 정확히 절단용 칼날이 도달하게 하여 아가미와 꼬리를 자르는 지능형 가공 시스템이며, IBM-PC 모니터 상에는 각 생선 절단 모듈의 각각의 제어 동작을 수행하는 실린더 동작부 및 생선 이송용 컨베이어의 상태가 디스플레이가 된다.

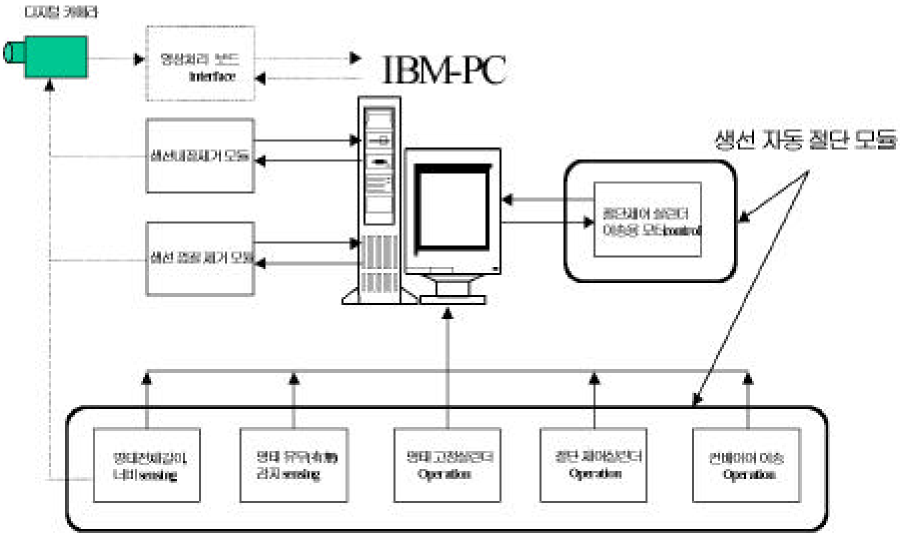


그림 14. **전반적인 지능형 가공 시스템의 모니터링** I/O**구조**

Fig 14. Monitoring I/O Structure of a General Processing System

기본적인 I/O**는 그림** 15**에서 보는 바와 같이 컨베이어 이송**(**移送**) **여부**, **생선 고정**(**固定**) **제어 실린더 동작여부**, **생선 절단 제어 실린더 동작여부**, **절단위치에서의 생선의 유무**(**有無**) **상태를 확인하는 생선 유무 감지센서 동작여부의 입력** 4**개 부분과 신경망 출력에 의한 모터 위치제어 출력부분으로 구분할 수 있다**. **각 동작은 인텔사의** PPI8255 **병렬 입출력 인터페이스 칩을 사용하여** I/O**의 동작을 모니터링 하게 된다**.

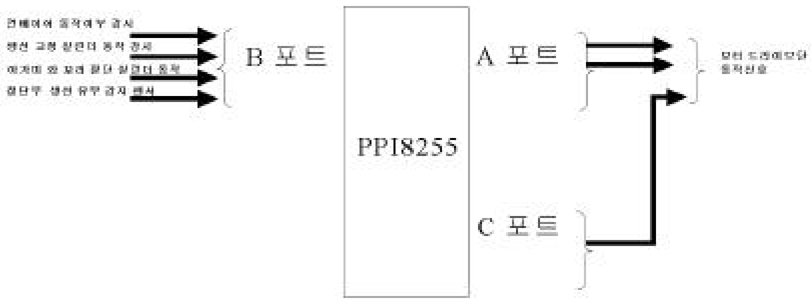


그림 15. PPI8255 **인터페이스 구조 및 흐름**

Fig 15. The Structure and Data Flows of PPI 8255

4 .2 IBM-PC 인터페이스 기반의 가공 시스템 모니터링 기능

그림 16**은 이제까지 제시했던** IBM-PC**인터페이스 기반의** I/O**보드를 이용하여 구성한 모니터링 기능을 요약한 것이며 이것을 바탕으로 그림** 17**에서는 구현된 지능형 가공 시스템에서의 모니터링 화면을 제시한다**.

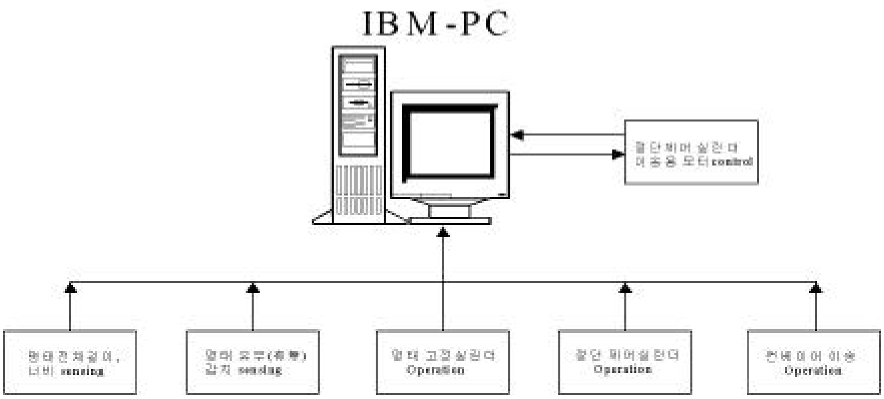


그림 16. **본 논문에서 제안된** IBM-PC **인터페이스 모니터링 기능**

Fig 16. Proposed Monitoring through IBM-PC Interface Function in this paper

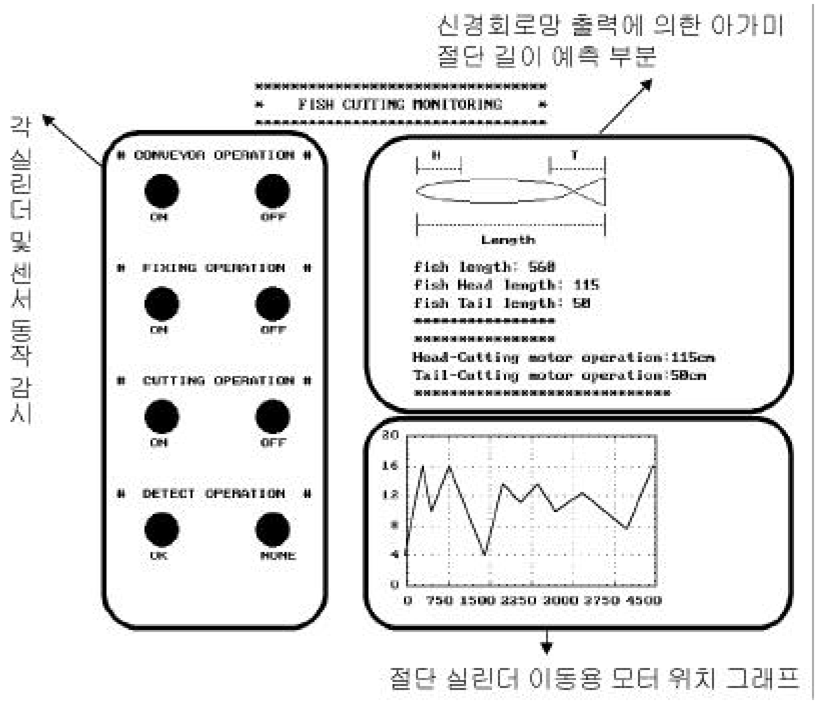


그림 17. **지능형 가공시스템의 모니터링 화면**

Fig 17. A Picture of Monitoring of the Artificial Processing System

모니터링의 기능은 그림 17**에서 보는 바와 같이 다음과 같다**. **각각의 제어 유닛 즉**, **컨베이어 이송**(Conveyor Operation), **생선 고정 실린더 작동**(Fixing Operation), **절단 실린더 기능**(Cutting Operation),**절단 부에서의 고기 위치 검출**(Detecting Operation)**의** 4**개 부분의 작동여부를 확인할 수 있다**(**그림** 17**의 좌측 부분**).

이 부분의 제어는 시퀜스 제어이므로 컨베이어 이송(ON←OFF)→**생선 검출**(OK/NONE)→**컨베이어 정지**(ON→OFF)→**생선 고정**(ON←OFF)→**생선 아가미 및 꼬리 절단**(ON←OFF)**의 순서로 모니터 상에서 색깔이 바뀌면서 점등된다**(**괄호 안의 순서는 모니터링 순서**).

오른쪽 상단에 표시되는 것은 생선(**명태**)**의 전체길이와 너비의 데이터가 어레이 센서에 의해서 검출되어 숫자로 표시된다**. **우측 하부**

의 그래프는 생선아가미와 꼬리 부분 절단 실린더 이동용 모터의 이동거리가 표시되어 운전자가 생선의 아가미 및 꼬리가 잘 짤렸는지 아니면 이상이 있는지를 확인해 볼 수가 있다.

프로그램의 간편성과 조작이 용이한 것에 초점을 맞추고, 구현한 I/O보드는 산업 현장에서 많이 쓰고 있는 ISA방식으로 범용 인터페이스 카드를 사용한다. 경제적으로 저렴하고 초고속의 데이터 통신을 원하는 것이 아니므로 ISA방식의 인터페이스 I/O를 선택하였으며 윈도우 프로그램보다는 산업현장에서 많이 사용되는 C 언어로 구성하여 DOS 기반에서도 운전할 수 있도록 모니터링화면을 단순하게 구성하였다.

제 5 장 시뮬레이션 및 실험결과

5 .1 시뮬레이션

많은 어종(魚種)중에서 현재 우리 나라에서 꾸준한 수요를 얻고 있는 명태를 시뮬레이션의 대상으로 설정했다. 현재 명태를 이용한 생선가공공장에서는 원양어선을 통해서 잡히는 북태평양 어장이나 러시아 일본 등에서 잡히는 명태를 주원료를 사용하고 있다.

명태의 크기는 3통 6통(통: 명태의 크기를 일컫는 단위, 약 56cm 100cm) 정도의 크기를 주원료로 하여 가공제품을 만들므로 여기에 초점을 맞추어 무작위로 명태의 크기를 조사하였다.

조사방법은 국립 수산진흥원의 명태의 체장(體長) 및 너비에 대한 데이터를 협조 받았으며, 또한 실제 명태 가공공장인 부산 사상 무지개 공단 內 유림수산유통 주식회사의 도움을 받아 일주일간 명태에 대한 체장, 아가미 길이, 너비, 꼬리길이 등의 데이터를 조사, 수집하였다.

그림 18은 명태의 실제적인 측정방법을 제시하고 있다. 이것을 바탕으로 명태에 대한 전체 길이(1) 및 너비(6) 데이터를 표로 정리하여 신경회로망의 학습 데이터로 사용하였다.

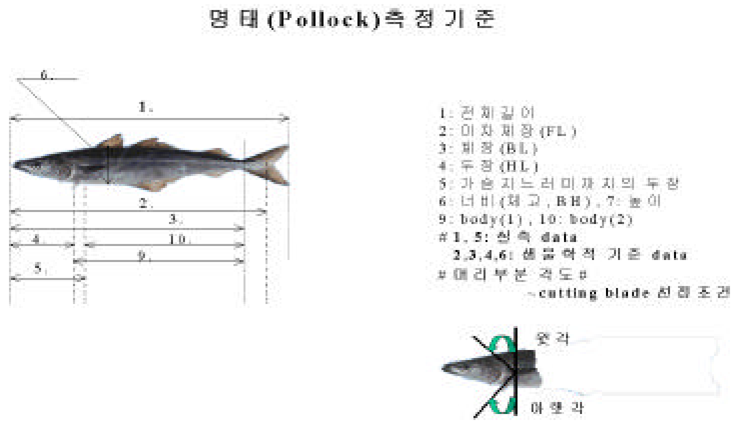


그림 18. 생선(명태) 측정 기준

Fig 18. The Standard of Measurement of Fish(Pollock)

아가미 절단부분의 길이를 예측하기 위해서 오류역전파 알고리즘을 사용한 신경회로망을 사용하였다 [1][3].

부록의 표1 5까지의 데이터를 바탕으로 전체길이와 너비의 2개의 파라메타가 있을 때 아가미 길이를 예측하는 신경회로망을 시뮬레이션 한다. 이 경우의 신경회로망의 구조는 입력 뉴런 2개, 은닉층 뉴런 30개, 출력 층 뉴런 1개로 구성되며 각각 바이어스 항을 은닉층과 출력 층에 두었다.

학습률은 0.7 바이어스 값은 -0.6으로 하였고 학습된 결과는 각각 파일로 저장하여 신경회로망의 feedfoward함수가 빠른 응답을 할 수 있도록 웨이트를 고정시켜 실제적으로 생선절단 모듈의 절단 실린더 이동용 모터의 이동거리 즉, 아가미가 잘려지는 부분까지의 거리를 출력으로 낼 수 있도록 프로그램 하였다.

그림 19에서는 아가미 절단 예측 부분의 신경회로망 출력 학습진행을 나타낸다. 학습횟수는 1925회이며 오차는 지역 최소 점을 감안하여 최대오차로 0.0039에서 학습을 종료하였다.

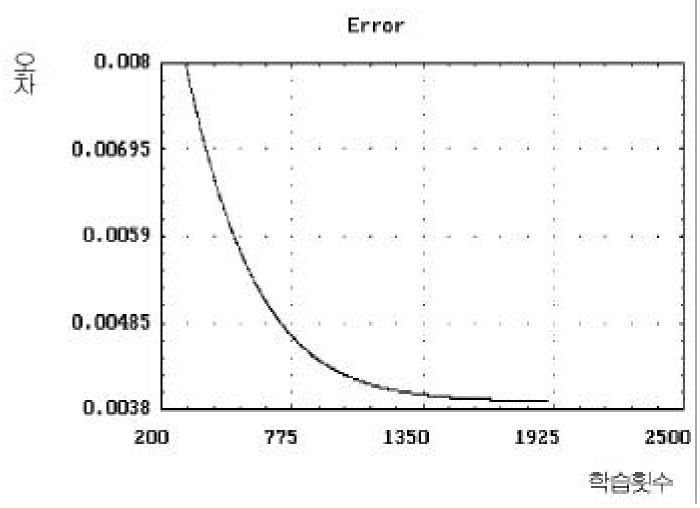


그림 19. 신경회로망의 학습진행 상황

Fig 19. The Learning Processing of Neural Networks

그림 20는 생선의 전체길이와 너비를 바탕으로 하여 신경망으로 예측한 아가미 길이 출력 값(실선)과 실제 생선의 아가미 길이(점선)를 비교한 것이다. 생선(명태)은 본래의 크기보다 1/100으로 스케이링 해서 학습을 시켰다. 실제로 모터를 이동시켜 생선의 아가미를 절단했을 때의 경우의 오차는 그림 21에서 나타낸다.

즉, 그림22는 아가미 절단시 오차 범위에 대한 것이다. 오차의 최대 편차는 -0.01 +0.01을 보이고 있으며 cm로 환산을 하게 되면 아가미 길이의 오차는 0.2 0.6사이에서 가장 많이 분포하게 된다.

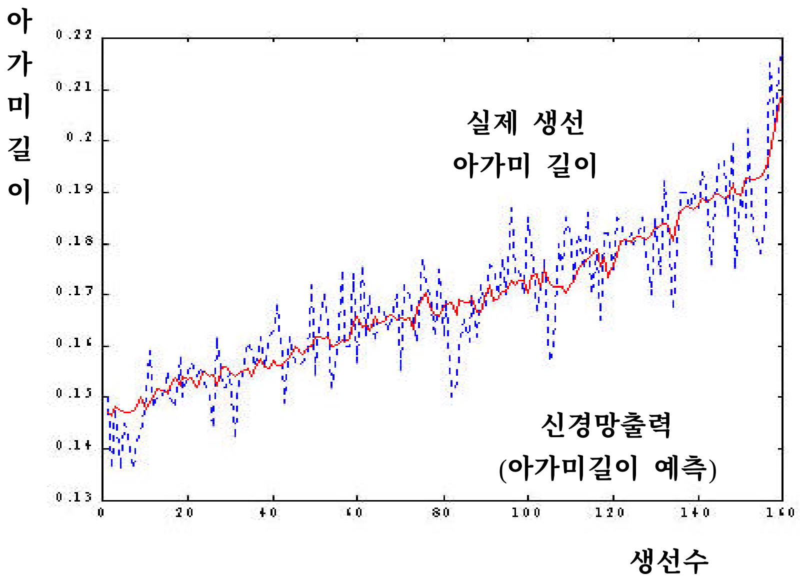


그림 20. 신경회로망 출력과 실제 생선 아가미 길이 비교

Fig 20. Compare the output of Neural Netwoks with the Gills of a Fish

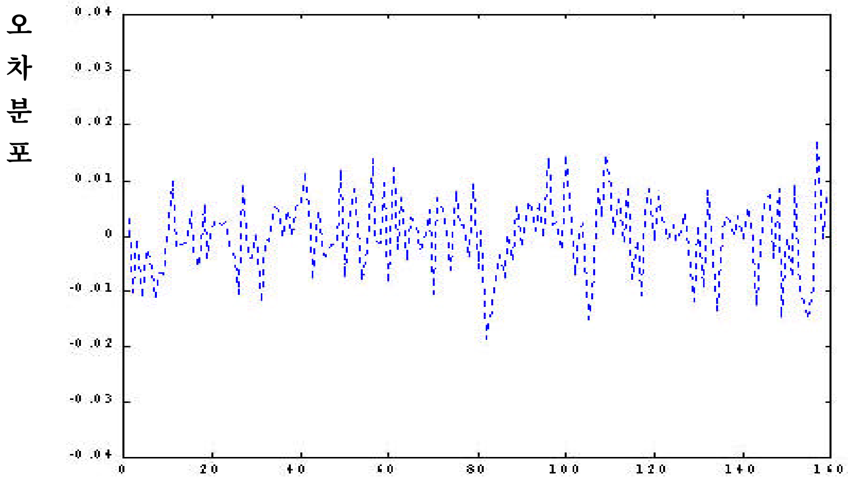


그림 21. **신경회로망 출력과 실제 생선 아가미 길이의 오차 분포**

Fig 21. Error Distribution of the Output of Neural Networks and real Gill-length of a Fish

다음은 32**개의 랜덤한 생선(명태) 패턴을 사용하여 길이와 너비가 센싱 되었을 때의 신경회로망의 출력을 시뮬레이션 결과로 제시한 것이다**. **학습률은** 0.7 **최대 학습오차는** 0.001**로 하였고 활성화함수는 단 극성 시그모이드 함수를 사용했을 때**, **그림** 22**은 신경회로망의 학습 진행 상황을 나타낸다**.

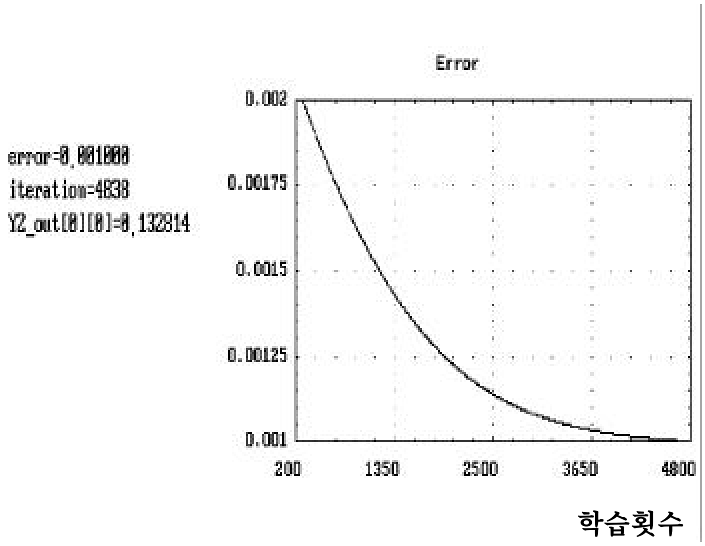


그림 22. 신경회로망의 학습진행 상황
(32 마리에 대해서)
Fig 22. The Learning Processing of Neural Networks(for 32 fish)

그림 23는 랜덤한 32마리 생선의 전체길이와 너비를 바탕으로 하여 신경회로망의 출력 값과 실제 아가미 길이를 비교한 것이며 그림 24은 그림 23의 데이터를 바탕으로 실제적인 아가미 길이와 아가미 절단 예측 길이(신경망 출력)의 오차편차를 나타낸다. 단 전체 생선 160마리 중에서 랜덤 하게 32 마리를 골라 학습시켰을 때의 학습상황과 오차편차를 나타낸다.

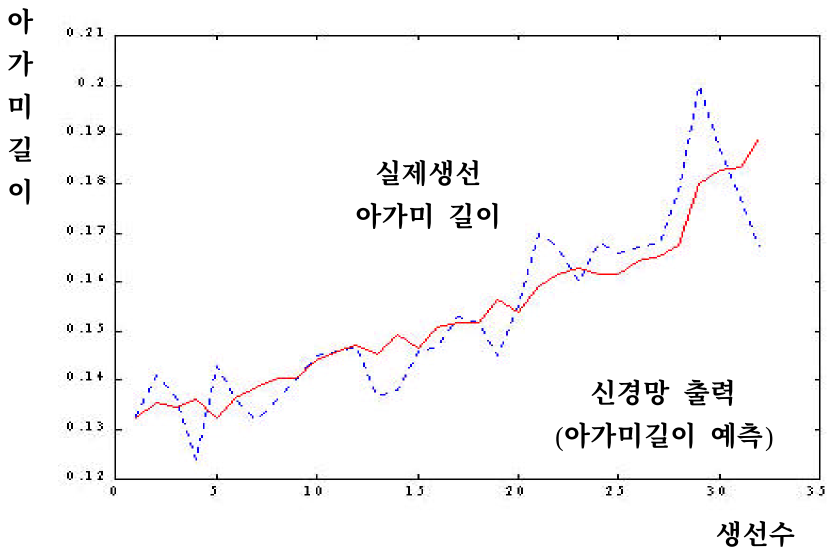


그림 23. 신경회로망의 아가미 예측 출력과 실제 생선 아가미 길이 비교

Fig 23. Compare the output of Neural Netwoks with the Gills of a Fish

그림 24에서 보듯이 너비와 길이의 데이터로 아가미의 길이를 예측하였을 때의 출력오차 범위는 +/-0.005 0.1사이의 값에서 오차 분포를 보이고 있다. 센싱되는 생선의 전체길이와 너비는 1/100으로 스케일링을 한다. 즉, cm로 환산한다면 +/-0.5 0.9cm정도의 편차를 보이며 평균적인 에러(error)는 0.5cm내외의 분포를 보인다.

따라서 너비와 전체길이의 센싱은 좀더 정확한 아가미 절단 부분을 예측할 때 좋은 가능을 발휘할 수 있다는 결론을 얻었다.

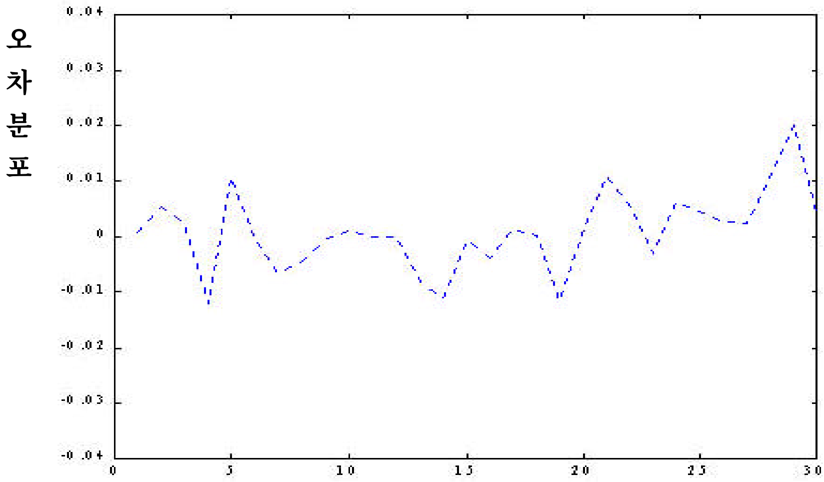


그림 24. 신경회로망 출력과 실제 생선 아가미 길이의 오차 분포

Fig 24. Error Distribution of the Output of Neural Networks and real Gill-length of a Fish

5 .2 실험 결과 및 고찰

시뮬레이션 결과 신경망의 학습능력으로 아가미의 절단 위치를 예측하는 능력이 충분한 역할을 할 수 있음을 확인할 수 있었다. 명태의 랜덤한 획득을 가만 한다면 어느 정도 신빙성을 갖는 자료임을 알 수 있었다. 최소한의 센서로 고기의 불확실한 아가미 길이를 찾아내는 것은 너무나 비 선형적이고 어떠한 법칙이 있는 함수로 정의 할 수도 없다.

따라서 이러한 비선형적이고 불확실한 데이터를 바탕으로 결과를 산출할 수 있는 능력은 신경망이의 학습능력으로 볼 때 충분한 가능성이 있다고 사료된다.

본 논문에서는 약간은 제한적인 조건을 두고 예측을 했지만 실제로 가공 시스템에 적용시켜 보았을 때는 오차 범위를 0,5cm내외에서 아가미를 절단하고 아물러 전체길이에 대한 꼬리를 비율적으로 절단함으로써 좋은 결과를 얻을 수 있었다. 또한 유기적인 모니터링이 이루어져 가공 시스템의 동작을 완전히 파악 할 수 있었으며 실제로 산업현장 적용시 좋은 효과를 기대할 수도 있을 것이다. 또한 신경망과 감도가 월등한 센서가 역할을 분담하게 된다면 정확성이 급격히 상승할 것이라 사료된다. 센서의 역할을 신경망이 대신 수행하는 것을 볼 때 생선의 아가미 예측 뿐만 아니라 예측 제어 하고 다시 그정보를 신경망을 통해 학습시키는 세미 페루프를 형성시키면 더욱더 강인한 제어로 여러 종류의 플랜트에 적용될 수 있으리라 사료된다.

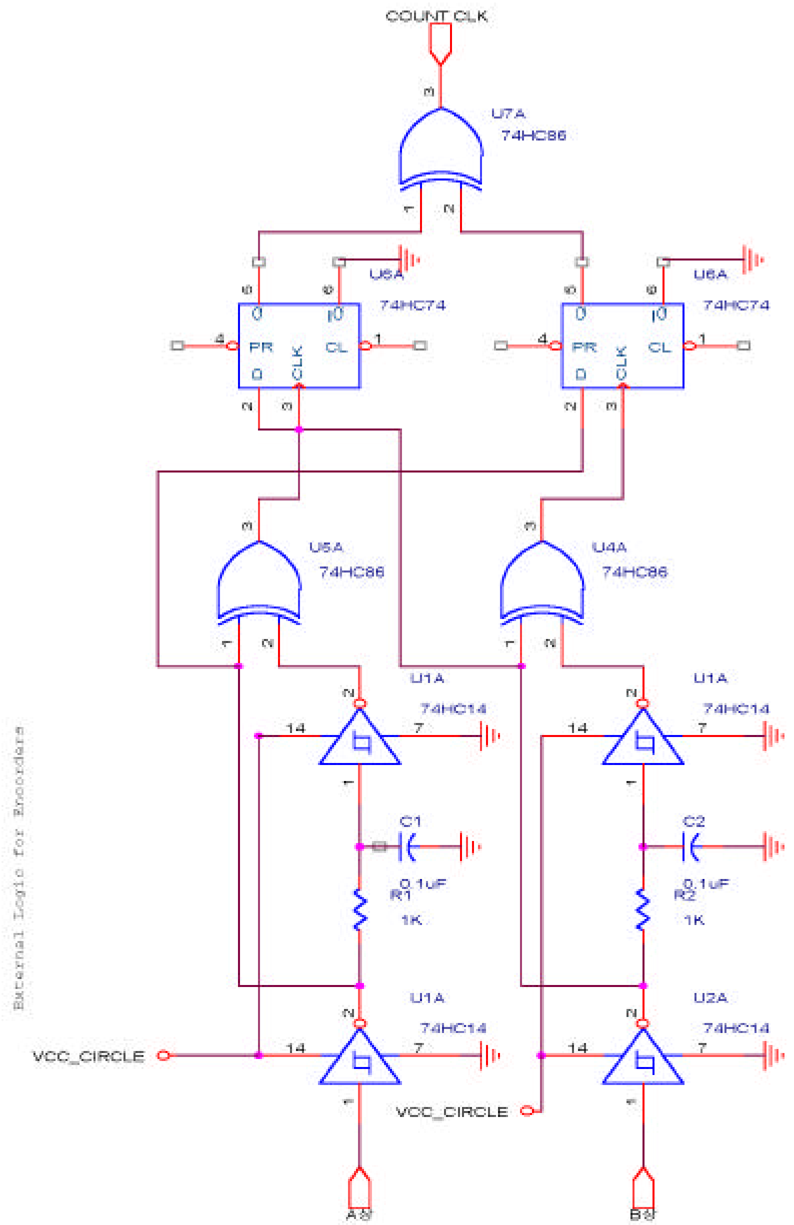
제 6 장 결 론

본 논문에서는 인공지능 기법의 한 방법인 신경회로망을 이용하여 현재 자동화가 더디게 이루어지는 1차 산업의 수산 가공에서 꼭 필요한 생선 전처리 지능형 가공시스템을 구현하였다. 생선의 가변적인 요소 즉, 크기가 일률적이지 못한 대상물을 포착하고 초소한의 센서(본 논문에서는 광센서를 사용)를 사용하여 센서의 능력대신 신경망의 학습능력을 이용하여 적절한 생선의 절단 부분을 예측하고 그것에 합당한 제어 요소를 control 함과 동시에 전 시스템의 효율적인 동작관리와 이상여부를 쉽게 감지 할 수 있는 모니터링 요소를 IBM-PC 인터페이스 기반으로 구축했다.

단순한 구조의 신경회로망은 짧은 시간 안에 빠른 연산을 할 수 있으므로 광범위한 비선형 데이터를 예측하는 시스템에 적합하다는 것과 우수성을 시뮬레이션 결과와 실험을 통해 확인할 수가 있었다.

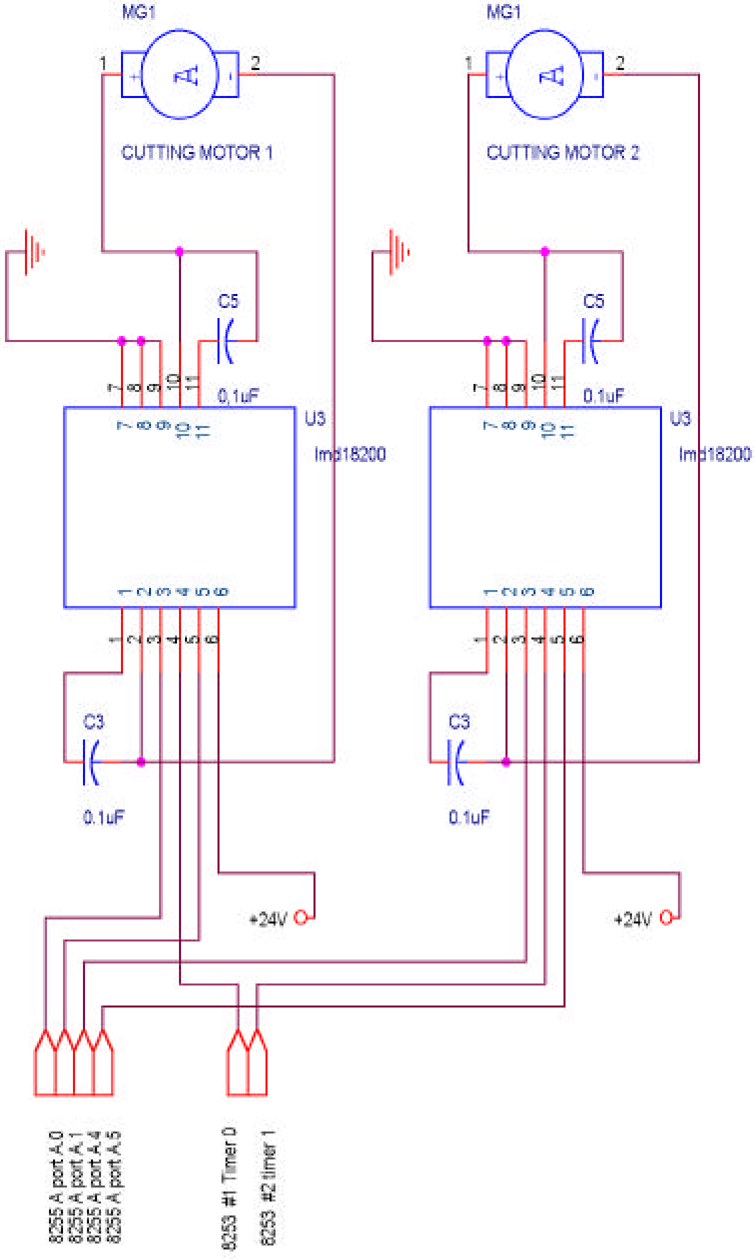
본 논문에서 제한적인 범위 즉 신경회로망이 예측한 절단 아가미 부분까지 정확하게 절단 실린더을 움직이는 모터제어에 주기능을 부여하여 신경회로망을 적용하고 IBM-PC I/O기반의 모니터링 기법을 연구했다. 향후 연구 방향은 가공시스템 공정 전반적인 제어요소를 파라메타로 하여 실시간 공정시 예기치 못한 상황에 좀더 유연하게 시스템이 대처할 수 있고 모니터링 기능이 지역적 범위를 벗어나 웹(web) 기반에서의 모니터링 기능을 할 수 있는 시스템을 구축하는 것과 관리자의 노하우 및 경험을 바탕으로 퍼지제어(Fuzzy Logic Control) 기법이 가지는 추론능력을 신경망과 함께 도입 하는 것이다. 이렇게 함으로써 완전한 지능형 실시간 가공시스템이 구현될 것으로 사료된다.

부 록(Appendix)



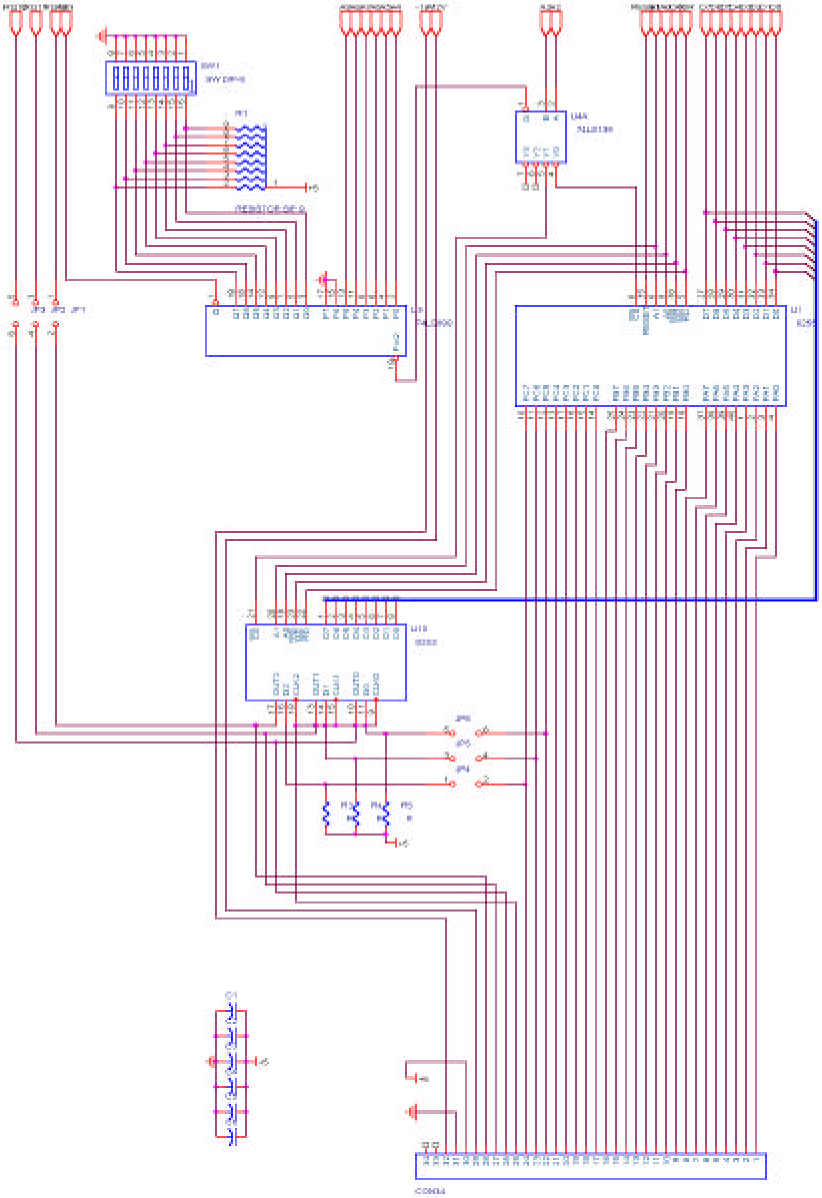
부록 1. **외부 엔코더** 2**체배 회로**

Appendix 1. External logic for Encoder Circuit



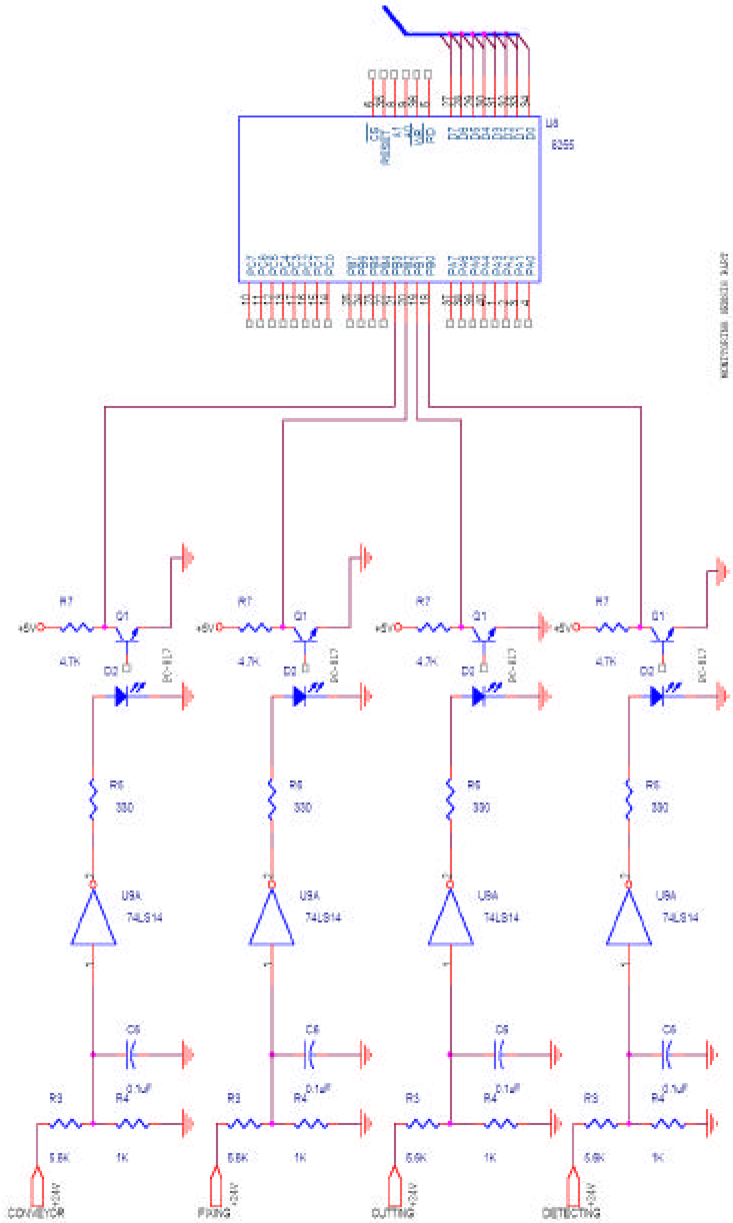
부록 2. 모터 구동 회로도

Appendix 2. Driver Circuit of DC-Motor

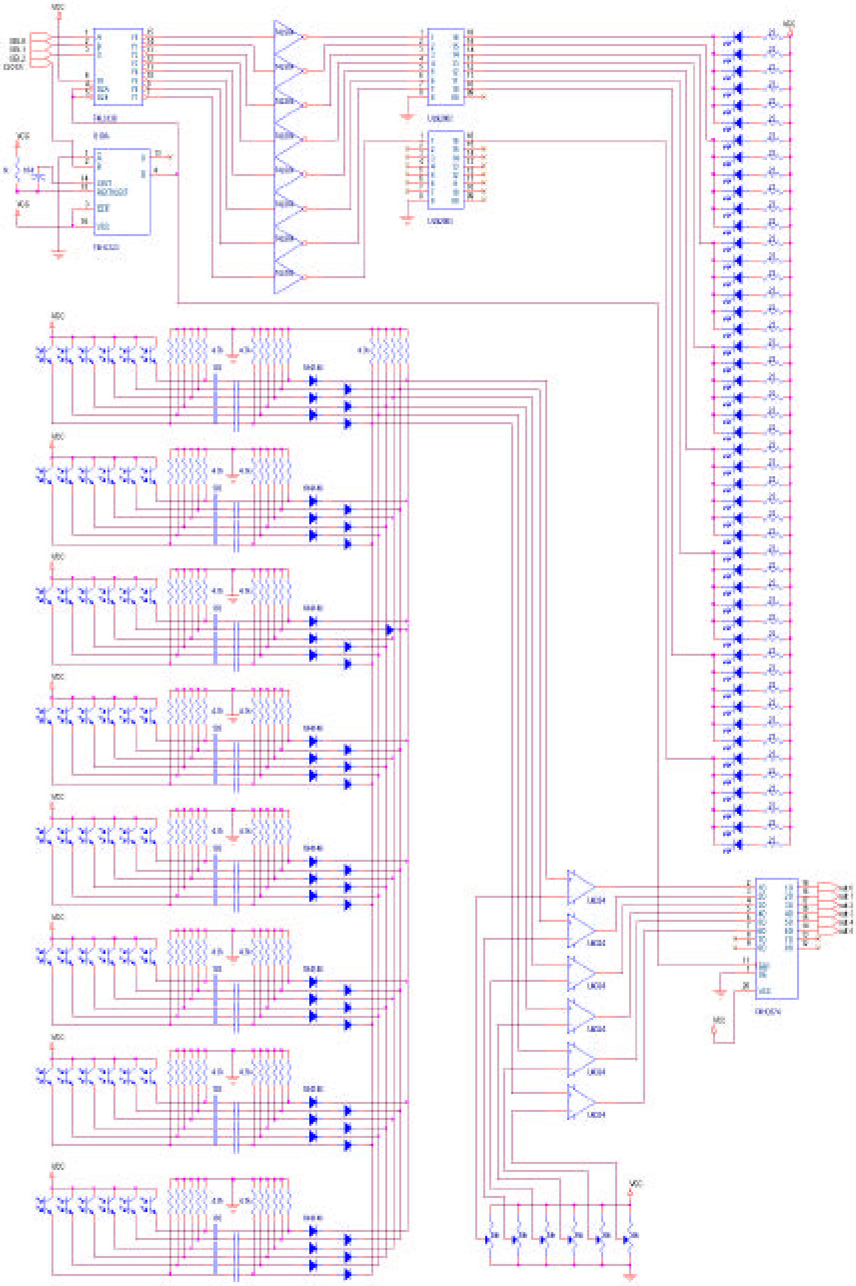


부록 3. IBM-PC I/O **인터페이스 보드 회로도**

Appendix 3. Circuit of IBM-PC I/O interface board



부록 4. 컨베이어, 생선고정, 생선절단, 고기검출 I/O회로도



부록 5. 적외선 센서 회로도

Appendix 5. The Circuit of Infrared Rays Sensor

표 1 명태 길이 및 너비 (단위 cm)
(자료출처:국립 수산 진흥원 "원양자원과")

미차체장	너비	미차체장	너비
39	6.3	41.6	6.2
39	6.4	41.7	6.2
39.1	6.2	41.7	7
39.2	5.9	41.7	6.9
39.2	6.7	41.8	7.7
39.3	6.5	41.9	6.3
39.4	7.5	41.9	6.7
39.4	6.3	42	8.2
39.4	6	42	6.4
39.4	6.8	42	7.4
39.5	6.1	42.1	6.4
39.6	7.2	42.2	7.4
39.6	6.8	42.3	7.8
39.8	6	42.4	6.5
40	6.4	42.5	6.8
40	6.8	42.6	7
40	6	42.7	7.6
40	6.4	42.9	6.5
40	5.8	43	8.8
40	6.6	43.2	6.4
40.1	6.2	43.4	7.2
40.1	6.8	43.6	6.7
40.2	6.6	43.7	6.8
40.2	7.8	43.8	7
40.3	7.9	44.2	6.8
40.3	7.6	44.3	7.8
40.3	6.3	44.3	7.4
40.4	6.4	44.4	7.7
40.4	5.7	44.4	8
40.4	9.6	44.4	7.4
40.5	6.2	44.4	6.8
40.5	6.4	44.6	6.4
40.5	7	44.7	8
40.6	6.8	44.8	6.5
40.6	7	45.4	7.8
40.6	6.2	45.4	7.1
40.6	6.9	45.7	8
40.7	6.4	45.8	7.2
40.7	6.5	46.4	6.9
40.8	6.8	46.6	8.6
40.8	6.8	47.2	6.8
40.8	6.6	47.6	8.7
40.9	6	47.8	10
41	6.4	47.8	8.2
41.1	7.2	48.6	8
41.2	6.6	49.7	7.6
41.2	6.6	49.8	8.3
41.3	7.2	50.3	7.6
41.4	6.6	50.5	6.8
41.4	6.4	51.2	8.3
41.4	6.6	51.4	8.3
41.4	7	51.7	9
41.4	6.2	52.2	7.8
41.6	7	52.8	8.9
41.6	6.4	53.2	8.8
41.6	6.2	53.6	8.2
41.6	6.8	54.2	11.5
41.6	6.3	54.5	8.7
		57.8	9

표2 명태 길이 너비 데이터 (단위 m)
(자료 출처: 실제측정 데이터 (주)유림수산유통)

전체길이	너비		아가미
0.57	0.085		0.15
0.572	0.079		0.136
0.574	0.088		0.148
0.574	0.083		0.136
0.575	0.081		0.145
0.576	0.08		0.143
0.578	0.08		0.136
0.582	0.082		0.142
0.583	0.088		0.143
0.585	0.075		0.15
0.587	0.08		0.159
0.587	0.085		0.148
0.588	0.091		0.15
0.588	0.09		0.15
0.589	0.086		0.155
0.59	0.092		0.15
0.595	0.094		0.148
0.595	0.088		0.158
0.595	0.096		0.15
0.595	0.093		0.155
0.596	0.094		0.157
0.598	0.083		0.154
0.599	0.09		0.156
0.6	0.096		0.153
0.602	0.089		0.15
0.602	0.091		0.144
0.604	0.08		0.162
0.605	0.096		0.152
0.605	0.096		0.152
0.605	0.091		0.155
0.605	0.087		0.142
0.606	0.09		0.154
0.607	0.09		0.154
0.607	0.093		0.161
0.607	0.086		0.159
0.61	0.092		0.156
0.61	0.098		0.162
0.61	0.09		0.156
0.61	0.089		0.162
0.61	0.096		0.163
0.613	0.09		0.168
0.615	0.087		0.158
0.617	0.088		0.149
0.618	0.091		0.162
0.62	0.1		0.157
0.62	0.098		0.156
0.62	0.093		0.157
0.625	0.093		0.156
0.625	0.095		0.172
0.625	0.103		0.154
0.628	0.1		0.164
0.63	0.097		0.17
0.63	0.098		0.164
0.631	0.089		0.152
0.632	0.09		0.158
0.632	0.093		0.175
0.633	0.091		0.16
0.635	0.091		0.16
0.637	0.104		0.174

표3 명태 길이 너비 데이터 (단위 m)
(자료 출처: 실제측정 데이터 (주)유림수산유통)

0.638	0.108		0.157
0.64	0.097		0.176
0.64	0.1		0.162
0.641	0.093		0.17
0.644	0.102		0.161
0.644	0.097		0.166
0.644	0.098		0.167
0.645	0.101		0.163
0.645	0.103		0.165
0.645	0.099		0.17
0.645	0.1		0.155
0.647	0.097		0.172
0.647	0.098		0.17
0.647	0.088		0.165
0.648	0.105		0.161
0.65	0.112		0.177
0.653	0.115		0.172
0.654	0.103		0.17
0.655	0.093		0.162
0.655	0.091		0.175
0.655	0.1		0.162
0.655	0.103		0.169
0.657	0.103		0.15
0.659	0.09		0.153
0.66	0.103		0.16
0.66	0.101		0.165
0.661	0.101		0.161
0.663	0.104		0.17
0.663	0.09		0.162
0.665	0.088		0.172
0.665	0.104		0.168
0.665	0.113		0.176
0.665	0.098		0.175
0.666	0.099		0.17
0.667	0.105		0.177
0.667	0.104		0.17
0.668	0.113		0.187
0.668	0.109		0.174
0.67	0.11		0.175
0.67	0.11		0.17
0.67	0.099		0.185
0.672	0.113		0.18
0.672	0.114		0.167
0.672	0.103		0.172
0.673	0.117		0.177
0.673	0.104		0.157
0.674	0.102		0.16
0.674	0.101		0.18
0.675	0.1		0.175
0.675	0.094		0.185
0.678	0.099		0.181
0.679	0.1		0.173
0.685	0.107		0.182
0.685	0.111		0.175
0.687	0.113		0.186
0.688	0.115		0.17
0.69	0.118		0.178
0.69	0.103		0.165
0.693	0.11		0.182
0.695	0.088		0.182
0.696	0.099		0.175

표4 명태 길이 너비 데이터 (단위 m)
(자료 출처: 실제측정 데이터 (주)유림수산유통)

0.7	0.104		0.185
0.7	0.116		0.184
0.702	0.115		0.18
0.703	0.111		0.182
0.705	0.114		0.18
0.705	0.115		0.182
0.705	0.113		0.185
0.708	0.109		0.178
0.71	0.112		0.17
0.713	0.115		0.185
0.714	0.117		0.174
0.715	0.118		0.192
0.715	0.116		0.184
0.715	0.101		0.167
0.72	0.114		0.18
0.725	0.12		0.19
0.725	0.124		0.19
0.725	0.122		0.187
0.725	0.12		0.19
0.73	0.12		0.197
0.733	0.123		0.194
0.735	0.118		0.191
0.735	0.12		0.176
0.737	0.124		0.188
0.738	0.123		0.196
0.739	0.117		0.196
0.743	0.115		0.185
0.745	0.124		0.2
0.745	0.116		0.175
0.749	0.11		0.189
0.75	0.124		0.185
0.75	0.126		0.202
0.754	0.119		0.182
0.755	0.12		0.18
0.755	0.123		0.178
0.76	0.125		0.185
0.765	0.116		0.215
0.8	0.124		0.202
0.82	0.125		0.214
0.825	0.13		0.217

표5 명태 길이 너비 데이터 (단위 m)
(자료 출처: 실제측정 데이터 (주)유림수산유통)

전체길이	너비		아가미
0.568	0.081		0.133
0.58	0.091		0.141
0.581	0.082		0.137
0.584	0.088		0.124
0.585	0.065		0.143
0.592	0.084		0.136
0.594	0.095		0.132
0.605	0.096		0.136
0.613	0.09		0.14
0.635	0.091		0.145
0.635	0.104		0.146
0.637	0.108		0.147
0.637	0.097		0.137
0.657	103		0.139
0.658	0.088		0.146
0.667	0.105		0.147
0.675	0.104		0.153
0.676	0.103		0.152
0.683	0.124		0.145
0.686	0.107		0.155
0.71	0.115		0.17
0.714	0.123		0.167
0.72	0.124		0.16
0.722	0.116		0.158
0.728	0.107		0.166
0.73	0.12		0.167
0.744	0.113		0.168
0.745	0.122		0.178
0.814	0.122		0.2
0.817	0.13		0.187
0.822	0.126		0.177
0.86	0.123		0.167

참고 문헌

[1] Lefteri H. Tsoukalas, Robert E. Uhrig. "Fuzzy and Neural Approaches in Engineering", John Wiley & Sons,Inc, pp. 1-7, 1992.

[2] Jacek M. Zurada, "Introduction to Artifical Neural System", west Publishing Company, pp. 26-235, 1992

[3] M. Minsky and S. Paper, *Perceptrons*, MIT Press, 1969.

[4] C. Koch, T. Poggio, "Multiplying with Synapses and Neurons", in Single Neuron Computation, T. Mckenna, J. Davis, and S. F. Zonnetzer[eds.], pp. 3165-3455, 1992.

[5] Jacek M. Zurada, "introduction to Artificial Neural System" Systems." West Publishing Company, pp. 26-235, 1992.

[6] Y. Chen and F. Bastani, "ANN with Two Dendrite Neurons and Weight Initialization", Proc. IJCNN, Baltimore, Vol.Ⅲ, pp. 139-146, 1992.

[7] Sigeru Omatu, Marzuki Khalid and Rubiyah Yusof, "Neuro-Control and its Applications", pp. 1-27, 1995

[9] Hopfield, J,J. and D.W.Tank, "Neural computation of decision in Optimizition problems", Biological Cybermetics Vol.52. pp. 141-155 , 1985

[10] Hopfield, J.J, "Neural networks and physical systems with emergent computational abilities", Proc. of National Academy of Secinces", Vol.79, pp. 2554-2558, 1982

[11] **이상배**, "**퍼지-뉴로 제어 시스템**", **교학사**, pp. 111-152 , 1999

[12] **신정환**, "C**가 미는 로봇트 Ⅰ**", Ohm**사**, pp. 183-280, 1998

[13] **신정환**, "C**가 미는 로봇트 Ⅱ**", Ohm**사**, pp. Ⅰ-3 Ⅱ-57, pp.Ⅴ-4 Ⅴ-17, 1999

[14] **신대섭**, **정봉성**, "8051+C**언어를 이용한 초보자가 만드는 로봇**", **도서출판 세화**, pp.20-218, 1999

[15] Fausett, " Fundamentals of neural networks-acrchitectures ,algorithms, and application", Prentice Hall, pp. 246-331,

[16] **이재현**, **강성인**, **이상배**, "**신경회로망을 이용한 동적 시스템의 상태공간 인식 모델에 관한 연구**", 1997**년 한국퍼지 및 지능시스템 학회 논문집**, Vol 7, pp. 115-129, 1997

[17] **차영배**, "IBMPC **인터페이스 응용**", **동일 출판사**, pp .227-425, 1999

[18] J.-S.R Jang, C.-T.Sun, E.Mizutani, "Neuro-Fuzzy and Soft Computing", Prentice Hall International Editions, pp. 129-257, 1997

[19] Hagan, Demuth, Beale, "Neural Network Design", PWS Plushing Company, pp Ⅱ-11 Ⅺ-43

[20] **오창석**, "**뉴로 컴퓨팅**", **지성 출판사**, pp. 121-275, 1996

[21] S. Chen et al, "Non-linear System Identification Using Neural Networks", Int.J.control, vol 6, pp, 1191-1214, 1990

[22] Simon Haykin, "Neural Networks", Macmillan Company, 1994

[23] Lin, Lee,"Neural Fuzzy systems-A Neuro-Fuzzy Synergism to Intelligent Systems", Prentice Hall, pp. 205-468, 1996

[24] **차영배**, "Microcontroller 80c196-**기초에서 응용까지**", **다다미디어**, pp. 417-480, 1998

[25] C.W. de Silva, " Intelligent Control-fuzzy logic applications", CRC press, pp. 235-268, 1995

[26] **김태영**, **김관형**, **이상배**, **김일**,"**신경회로망의 패턴분류를 이용한 용접 갭 검출에 관한 연구**", **한국퍼지 및 지능시스템학회 논문집**, Vol.8, No.2, pp. 527-531, 1988

[27] **강성인**, **김관형**, **이상배**, **탁한호**,"**신경회로망을 이용한 용접 갭 검출과 실시간 모니터링에 관한 연구**", **한국퍼지 및 지능시스템학회 논문집**, Vol.8, No.2, pp. 539-544, 1988

[28] **이주원**,"**인공지능형 전자식 배전반 시스템의 구현에 관한 연**

구”, **석사학위논문**, **한국해양대학교**, pp. 5-21, 1999