

TCP의 트래픽 제어를 위한 동적 베이시안 네트워크 기반 지능형 PID 제어기

An Intelligent PID Controller based on Dynamic Bayesian Networks for Traffic Control of TCP

조 현 철, 이 영 진, 이 진 우, 이 권 순*

(Hyun Choel Cho, Young Jin Lee, Jin Woo Lee, and Kwon Soon Lee)

Abstract : This paper presents an intelligent PID control for stochastic systems with nonstationary nature. We optimally determine parameters of a PID controller through learning algorithm and propose an online PID control to compensate system errors possibly occurred in realtime implementations. A dynamic Bayesian network (DBN) model for system errors is additionally explored for making decision about whether an online control is carried out or not in practice. We apply our control approach to traffic control of Transmission Control Protocol (TCP) networks and demonstrate its superior performance comparing to a fixed PID from computer simulations.

Keywords : intelligent PID, online learning, DBN model, TCP traffic

1. 서론

PID 제어방식이 산업현장에서 표준화 제어기로 널리 사용되고 있지만, 여전히 그 성능 향상을 위하여 많은 연구가 되고 있으며, 이는 최근의 시스템은 그 구조가 복잡하고 다양한 동특성을 지니므로 기존의 PID 설계법이나 해석으로는 다소 어려운 점이 있기 때문이다. 이런 문제들을 극복하기 위한 PID 연구는 수년 동안 이어져오고 있으며, 최근엔 소프트웨어 연산 알고리즘을 접목한 지능형 PID 제어시스템이 각광을 받고 있다. 이런 시스템은 주로 시변(time-varying) 또는 아주 복잡한 비선형성(nonlinearity) 그리고 시스템 섭동(perturbation)과 같은 동특성을 지니고 있으며, 주로 소프트웨어 연산을 통해 PID 제어기를 설계하고 있다[1,2]. 일반적으로 소프트웨어 연산 알고리즘은 인간의 의사결정, 정보처리 메커니즘 또는 생체 시스템을 해석적 모델로 변환하여 기존의 난해한 공학적 문제를 다루는데 이용된다. 주로 인공신경회로망[3], 퍼지이론[4], 유전자 알고리즘[5] 그리고 면역 알고리즘[6] 등이 이에 속하며, 이러한 알고리즘 등을 이용한 지능형 PID 제어시스템 또한 최근 활발히 진행되고 있는 연구 분야이기도 하다.

퍼지이론을 접목한 PID 제어는 제어기의 파라미터를 선정하는데 퍼지추론이 주로 이용된다. Yu 등은 실시간 PID 제어시스템을 구성하기 위하여 동적 퍼지기법을 이용하였고[7], Guzelkaya 등은 시변 또는 시간지연이 매우 강한 시스템에 퍼지 PID 제어기를 적용하였으며[8], Kukulj 등은 퍼

지 최적화 알고리즘을 이용하여 PID 파라미터 조정법을 개발하여 유도전동기 제어에 적용하였다[9]. 그 외 퍼지 PID는 산업현장의 폭 넓은 부분에 응용이 되고 있으며, 특히 로봇 시스템 등과 같은 정교한 제어성능을 요구하는 곳에 사용되고 있다. 문헌 [10]에서 Tian은 인공근육 및 관절을 제어하기 위한 퍼지 PID를 개발하였으며, Khoury 등은 5 자유도를 갖는 복잡한 로봇 팔의 제어에 퍼지 PID를 적용하였다[11]. 또한, Tan 등은 지능형 의족에 위치한 DC 모터의 정밀제어를 위하여 온라인 퍼지 PID를 개발하였으며 [12], 대전력 발전기나 대용량 보일러의 연소 시스템 같은 곳에도 퍼지이론을 적용한 PID 제어기도 적용되고 있는 실정이다[13,14].

신경회로망을 이용한 PID 제어는 주로 학습을 통하여 제어기의 파라미터를 결정하는데 사용될 수 있으며, 제어대상이 불확실성이나 모델링 오차 등과 같은 특징을 갖는 시스템에 주로 효과적으로 활용된다. 문헌 [15]에서 Zayed 등은 모델링의 불확실성으로 인한 시스템 오차를 보상하기 위해 비선형 PID제어기를 신경회로망을 통해 구성하였으며, Shu와 Pi는 시간지연 특성을 갖는 시스템을 위한 PID 신경회로망을 개발하였다[16]. Garg와 Gulati는 실시간에서 모델링 오차가 심한 비선형 자기부상시스템을 위한 신경회로망 PID제어기를 설계하였으며[17], 열교환기[18], 빌딩의 에너지 관리시스템[19], 자동차 주행[20], 로봇 보행[21] 등 다양한 시스템에 신경회로망 PID 제어가 연구되고 있다. 그 외 유전자 알고리즘이나 면역 알고리즘 등과 같은 탐색기법들도 최적의 파라미터를 선정하기 위하여 PID 제어에 적용되고 있으며, 특히 신경회로망이나 퍼지기법과는 달리 대역해(global solution)를 갖는 장점이 있다[22-24].

산업현장의 다양한 동적 시스템들이 지능형 알고리즘을 접목한 PID 제어시스템과 함께 응용되었으며, 기존의 방법으로는 힘들었던 제어성능을 얻을 수 있었다. 하지만 대부

* 책임저자(Corresponding Author)

논문접수 : 2007. 1. 25., 채택확정 : 2007. 2. 17.

조현철, 이진우, 이권순 : 동아대학교 전기공학과

(hyuncho@gmail.com/jinwoo@donga.ac.kr/kslee@donga.ac.kr)

이영진 : 한국폴리텍 항공대학 항공전기과(airlee@kopo.or.kr)

※ 이 논문은 정부(과학기술부)의 재원으로 한국과학재단의 국가 지정연구실사업으로 수행된 연구임 (No. M1030000030306J0000 30310).

본의 연구는 시스템을 결정론적(deterministic) 방법으로 모델링 또는 해석하여 PID 제어를 설계하고 있다. 앞서 설명한 바와 같이 최근의 시스템은 확률특성을 많이 내포하고 있으며, 특히 비가우시안(non-Gaussian)이나 비정상(nonstationary)와 같은 복잡한 특성을 가지므로 기존의 결정론적 방법으로 제어를 설계하는 것은 바람직하지 못하다. 특히 이런 특징을 갖는 시스템은 실시간 제어 시 예상치 못한 동특성으로 만족스러운 제어성능을 얻기 힘들며, 따라서 실시간 시스템 오차를 보상하는 방법이 요구된다.

본 논문에서는 비정상 특성을 갖는 확률시스템에 대한 지능형 PID 제어를 개발하였다. 오프라인 학습알고리즘을 통해 제어기의 파라미터를 결정하고 실시간에서 발생하는 시스템 오차를 위한 온라인 학습을 제안하였다. 개발한 PID 제어기는 비선형 및 확률특성이 매우 강한 Transmission Control Protocol(TCP) 네트워크의 트래픽(traffic) 제어를 위하여 적용되었으며, 컴퓨터 시뮬레이션을 통해 오프라인 학습에서 구성된 고정 PID와 제안한 온라인 PID 제어기의 성능을 비교 분석하였다. 본 논문의 구성으로는 2장에서 이산 PID 제어기의 파라미터 학습을 서술하며, 3장에서 온라인 PID 제어기법을 제안하였다. 4장과 5장에서는 시뮬레이션 결과 및 고찰을 통하여 본 논문의 결론을 기술하였다.

II. 이산 PID 제어를 위한 오프라인 학습 알고리즘

본 장에서 이산 PID 제어를 서술하였으며, 제어기의 파라미터 선정을 위한 학습 알고리즘을 제안한다.

1. 이산 PID 제어기

일반적인 연속시간 t 에 대한 PID 제어식은 다음과 같이 나타낼 수 있다.

$$u(t) = K \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{d}{dt} e(t) \right) \quad (1)$$

여기서 e 는 시스템 오차, PID의 파라미터로 주어지는 $K > 0, T_i > 0, T_d > 0$ 는 비례계수, 적분시간, 그리고 미분시간을 각각 나타낸다. 또는 $K_i = K/T_i, K_d = KT_d$ 로 정의하여 각각 적분계수 그리고 미분계수로도 나타낸다. 산업현장에서 PID 제어시스템은 디지털 컴퓨터와 함께 주로 구현되고 있으므로 (1)의 연속시간 PID를 이산형태로 전환해서 설계하여 한다. 이산시간 PID 제어식으로 나타내기 위해 미분연산을 (1)에 적용하면

$$\frac{d}{dt} u(t) = K \left(\frac{d}{dt} e(t) + \frac{1}{T_i} e(t) + T_d \frac{d^2}{dt^2} e(t) \right) \quad (2)$$

이고 Euler 근사법을 적용하여 다음과 같은 이산시간 k 에 대한 PID 제어식을 구할 수 있다.

$$u(k) = u(k-1) + K \left[\left(1 + \frac{h}{T_i} + \frac{T_d}{h} \right) e(k) - \left(1 + \frac{2T_d}{h} \right) e(k-1) + \frac{T_d}{h} e(k-2) \right] \quad (3)$$

여기서 h 는 샘플링 시간을 나타낸다. (3)의 PID 입력은 이

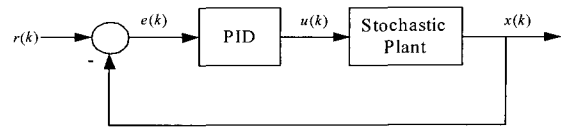


그림 1. 시변 확률시스템의 PID 제어.

Fig. 1. PID control for time-varying stochastic systems.

전 시간 $k-1$ 에서의 입력과 현재와 이전의 오차값과 그에 따른 계수들과 함께 선형 조합된다.

2. PID 파라미터 학습

PID 제어기의 설계는 주어진 플랜트에 대하여 원하는 제어성능을 갖는 최적의 파라미터를 결정하는 것이다. 일반적으로 선형방정식으로 주어지는 플랜트에 대하여 선형시스템 이론을 적용하여 원하는 제어사양에 맞는 PID 파라미터를 해석적인 방법을 통해 간단히 구할 수 있다 [25]. 하지만 비선형 시스템에 대한 선형화를 한 후 선형시스템 이론을 적용하여 구할 수 있지만, 선형화는 항상 모델링 오차가 동반되는 단점이 있다. 실시간 실험을 통해 획득된 시스템 응답을 이용하여 PID 제어를 설계하는 기법도 널리 사용되고 있다. 이런 방법은 복잡한 해석적 과정을 거치지 않는 장점은 있으나 실험 데이터가 시스템에 대한 모든 동특성 정보를 반영하기 어려운 점이 있으며, 특히 시스템 변수 및 환경의 변화로 인해 동특성이 달라지는 경우 파라미터를 재조정해야 하는 문제점을 가지고 있다. 최근의 플랜트는 복잡한 구조나 비선형성이 강한 특징이 있으며 특히 확률성분을 갖는 시스템에 대하여 기존의 방법으로는 PID 제어를 구현하는데 많은 어려움이 대두되고 있다. 이런 문제점 등은 학습을 통해 어느 정도 해결할 수 있다. 다시 말해, 수학적 모델로 주어지는 플랜트에 예상할 수 있는 시스템 특성을 모두 반영하여 최적화 기법을 적용하여 원하는 제어성능을 획득하기까지 반복적으로 시뮬레이션을 실시하여 제어기의 파라미터를 결정한다. 본 논문은 그림 1에서 보는 바와 같이 시변 또는 비정상 확률특성을 갖는 시스템에 PID 제어를 적용하였으며, 최적의 파라미터를 결정하기 위한 학습알고리즘을 제안하였다.

우선, (3)의 이산 PID 제어식을 가중치 $w_i, i=1,2,3$ 으로 대신해서 나타내면

$$u(k) = u(k-1) + w_1 e(k) - w_2 e(k-1) + w_3 e(k-2) \quad (4)$$

이 되며, 여기서 w_1, w_2 및 w_3 는 다음과 같다.

$$w_1 = K \left(1 + \frac{h}{T_i} + \frac{T_d}{h} \right) > K$$

$$w_2 = K \left(1 + \frac{2T_d}{h} \right) > K$$

$$w_3 = \frac{T_d}{h} > 0$$

따라서 PID 제어기 파라미터 학습은 (3)의 w_i 의 최적값을 선정하는 것이다. 그림 1의 플랜트 출력은 확률성분을 가지므로 이동평균(moving averaging) 알고리즘을 적용하여 평균

값으로 나타내면

$$\bar{x}(k) = \gamma \bar{x}(k-1) + (1-\gamma)x(k) \quad (5)$$

이 되고, 여기서 가중치 $\gamma \in (0,1)$ 이다. 학습 알고리즘을 산출하기 위하여 목적함수는 다음과 같이 주어질 수 있다.

$$J = \arg \min \left(\frac{1}{2} e^2 \right) \quad (6)$$

여기서 시스템오차 e 는 기준값과 (5)의 평균 출력에 대한 편차로서, 즉 $e = r - \bar{x}$. 이 목적함수를 최소화하는 파라미터 w_i 의 학습규칙을 산출하기 위하여, 최급강하(the steepest gradient descent) 알고리즘 [3]을 적용하면 (4)의 파라미터의 학습규칙은 (7)과 같이 나타낼 수 있다.

$$w_i(k) = w_i(k-1) - \eta \frac{\partial J}{\partial w_i}, \quad i = 1, 2, 3 \quad (7)$$

여기서 학습파라미터 $\eta \in (0,1)$ 이며, 이전의 파라미터 값과 파라미터에 대한 목적함수의 편미분 항으로 구성되며, 편미분항은 미분의 연쇄법칙(chain rule)을 이용하여 전개하면 다음과 같다.

$$\frac{\partial J}{\partial w_i} = \frac{\partial J}{\partial e} \frac{\partial e}{\partial \bar{x}} \frac{\partial \bar{x}}{\partial x} \frac{\partial x}{\partial u} \frac{\partial u}{\partial w_i}, \quad i = 1, 2, 3 \quad (8)$$

(8)의 우변항의 미분을 풀면 각각 다음과 같다.

$$\frac{\partial J}{\partial w_1} = -e^2(k)(1-\gamma) \frac{\partial x}{\partial u} \quad (9)$$

$$\frac{\partial J}{\partial w_2} = -e(k)e(k-1)(1-\gamma) \frac{\partial x}{\partial u} \quad (10)$$

$$\frac{\partial J}{\partial w_3} = -e(k)e(k-2)(1-\gamma) \frac{\partial x}{\partial u} \quad (11)$$

여기서 $\partial x / \partial u$ 은 시스템 자코비안으로서 확률시스템에 대하여 그 해를 구하기가 어려우므로 다음과 같은 근사식으로 두번 (12)와 같이 얻을 수 있다[26].

$$\frac{\partial x}{\partial u} = \frac{x(k) - x(k-1)}{u(k) - u(k-1)} \quad (12)$$

따라서 (9)-(11)을 대입하면 (4)의 파라미터 학습규칙은 최종적으로 다음과 같다.

$$w_1(k) = w_1(k-1) + \eta' e^2(k) \left(\frac{x(k) - x(k-1)}{u(k) - u(k-1)} \right) \quad (13)$$

$$w_2(k) = w_2(k-1) + \eta' e(k)e(k-1) \left(\frac{x(k) - x(k-1)}{u(k) - u(k-1)} \right) \quad (14)$$

$$w_3(k) = w_3(k-1) + \eta' e(k)e(k-2) \left(\frac{x(k) - x(k-1)}{u(k) - u(k-1)} \right) \quad (15)$$

여기서 $\eta' = \eta(1-\gamma) < \eta$ 이다.

III. DBN을 이용한 온라인 PID 제어

본 장에서는 온라인 PID 제어방식을 제안하였으며, 그 구현을 위하여 오차 신호에 대한 DBN 모델링과 온라인 제

어의 실행여부를 결정하는 가설검정법도 제시하였다.

1. 온라인 파라미터 학습

2장에서 제안한 학습법은 오프라인 또는 일괄처리(batch) 알고리즘으로서 플랜트의 해석적 모델을 이용하여 학습을 실시한다. 오프라인 학습은 많은 데이터에 대하여 최적의 값을 구할 수 있는 특징이 있지만, 학습 시 경험하지 못한 환경이 실시간에서 발생할 경우 제어성능이 저하될 수 있다. 이러한 문제점은 시스템을 아주 복잡하게 구성하거나 네트워크 기반 제어시스템, 또는 확률 특성이 변화는 시스템에 대하여 종종 발생하고 있다. 실시간에서 발생하는 시스템 오차에 대한 연구는 계속 진행되고 있으며, 특히 온라인 학습에 대한 연구는 최근 신경회로망 제어공학자들에게 많은 각광을 받고 있다[27]. 다시 말해, 제어기의 온라인 학습은 실시간에서 제어오차가 허용범위를 넘을 경우 오차를 줄이는 방향으로 파라미터 값을 수정한다. 온라인 학습은 기존의 오프라인 학습과 달리 확고한 설계지침이 없으며, 현재까지도 많은 연구가 이루어지고 있는 연구테마이다. 동적 시스템에 대한 온라인 학습은 정상상태에서 발생하는 오차의 보상으로써, 파라미터 학습규칙에 대하여 적응형 학습 파라미터를 적용하는 방법과 파라미터의 일부분만을 수정하는 방법이 널리 연구되고 있다[27]. 첫 번째 방법은 오프라인 학습에서 학습 파라미터는 고정값이었던 반면, 실시간에서는 제어오차의 크기에 따라 적응식으로 그 값을 변화시켜 학습속도를 향상시키는 방법이다. 두 번째의 경우는 모든 파라미터의 값을 수정하는 대신 시스템의 안정성에 영향을 미치지 않는 등 여러 가지 상황을 고려하여 어느 파라미터의 일부분만을 수정하는 방법으로 연산량과 시간을 감소할 수 있는 특징이 있다. 예를 들어, (4)의 제어식으로부터 w_1 은 현재 오차에 관한 파라미터로서 온라인 학습을 적용하여 실시간 오차의 보상을 위하여 수정규칙을 적용하며 나머지 두 파라미터 $w_i, i=1,2,3$ 은 오프라인 학습시 결정된 값을 유지한다. 그림 2는 이 온라인 학습에 대한 블록선도를 나타낸다. 따라서 온라인 학습시의 PID 제어입력은 다음과 같다.

$$u(k) = u(k-1) + w_1(k)e(k) - w_2^*e(k-1) + w_3^*e(k-2) \quad (16)$$

여기서 w_2^*, w_3^* 은 오프라인 학습에서 결정된 최적의 파라미터 값을 나타낸다.

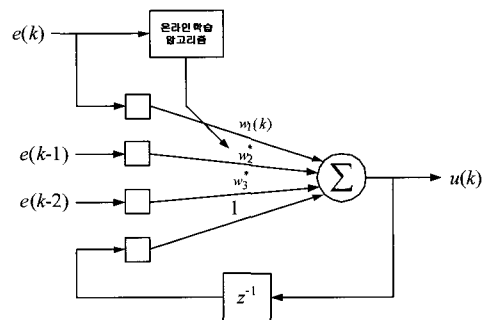


그림 2. PID 파라미터의 온라인 학습.

Fig. 2. Online learning for PID control parameters.

2. 온라인 학습을 위한 의사결정법

앞서 설명한 바와 같이, 제어기의 온라인 파라미터 학습은 정상상태에서 발생하는 시스템 오차를 보상하는 것으로 시스템 오차에 대하여 학습의 실행여부를 결정해야 한다. 다시 말해, 실시간 오차가 미리 정해진 허용범위를 넘을 경우에 온라인 학습이 이루어지며, 그렇지 않은 경우는 온라인 학습이 이루어지지 않고 이전의 파라미터 값을 그대로 적용해 나간다. 이것은 다음의 이진 가설검정(binary hypothesis testing)에 의해 표현될 수 있다.

$$\begin{aligned} H_0 : |e(k)| < \epsilon \\ H_1 : |e(k)| \geq \epsilon \end{aligned} \quad (17)$$

여기서 $\epsilon > 0$ 이다. (17)에서 첫 번째 가설의 경우 시스템 오차의 절대값이 ϵ 보다 작을 경우이며, 이 때 온라인 학습이 이루어지지 않는다. 두 번째의 경우는 ϵ 보다 크거나 같은 경우로서 실시간 학습이 이루어짐을 의미한다. 확률 시스템의 경우 시스템 출력은 확률특성을 가지며, 이 경우 시스템 오차값을 직접 적용하는 것보다 확률값으로 변환하여 가설검정에 적용하는 것이 의사결정법 관점에서 보다 효과적이다[28]. 연속신호인 오차 $e(k)$ 를 $n < \infty$ 등분으로 이산화하면 (18)과 같이 나타낼 수 있으며,

$$e(k) = \{e_1(k), e_2(k), \dots, e_n(k)\} \quad (18)$$

이에 대한 확률은 다음과 같다.

$$p(e(k)) = \{p(e_1(k)), p(e_2(k)), \dots, p(e_n(k))\} \quad (19)$$

여기서 확률공리(probability axiom)에 의해 (20)과 같이 나타낼 수 있다.

$$\sum_{i=1}^n p(e_i(k)) = 1 \quad (20)$$

(19)의 확률에 대하여 최대 확률값과 대응되는 변수를 결정할 수 있다. 즉,

$$i = \max \{p(e_1(k)), p(e_2(k)), \dots, p(e_n(k))\} \quad (21)$$

여기서 $i \in [1, n]$ 이 된다. 따라서 (21)을 적용하여 가설검정으로 나타내면 다음과 같다.

$$\begin{aligned} H_0 : i < \delta, \delta \in [1, n] \\ H_1 : i \geq \delta \end{aligned} \quad (22)$$

여기서 δ 는 기준값으로서 최대 확률을 갖는 변수가 δ 보다 낮은 위치의 변수일 때 H_0 의 의사결정을 따르며, 즉 온라인 학습이 이루어지지 않으며, δ 보다 같거나 높은 위치의 변수일 경우 학습이 이루어진다. 따라서 이 방법은 이산화된 오차변수에 대하여 각각의 확률값을 계산하여 의사결정이 이루어진다.

3. 시스템 오차의 확률 모델링

주어진 확률변수의 확률밀도함수(probability density function)를 결정하는 것은 쉬운 일이 아니며, 문제를 단순화하기 위하여, 일반적으로 가우시안 확률분포로 가정하여

시스템을 설계 및 해석하는 경우가 많다. 이것은 대부분 시스템이 실시간에서 가우시안 특성을 주로 가지기 때문이다. 하지만 시변 또는 비선형 시스템인 경우 이 가정이 늘 만족스러운 성능을 가져오는 것은 아니다. 따라서 주어진 플랜트에 대한 확률변수의 정확한 확률밀도 추정이 필요하며, 특히 온라인 추정을 위해서는 계산량이 적고 보다 효율적인 알고리즘이 요구된다. 본 논문은 [29]에서 제안한 온라인 확률밀도 추정알고리즘을 이용한다. 이 추정법은 동적 베이지안 네트워크를 이용하여 이산 확률변수를 모델링하였으며, 시스템 관측값을 기반으로 확률변수의 밀도함수를 실시간으로 계산하였다.

(19)의 이산 오차변수는 그림 3의 간단한 DBN 모델인 마코브 체인(Markov chain) [30]을 이용하여 나타낼 수 있다. 그림 3에서 네트워크 파라미터로 주어지는 조건부 확률 a_{ij} 는 다음과 같이 정의되며

$$a_{ij} = p(e(k) = i | e(k-1) = j), \quad i, j = 1, \dots, n \quad (23)$$

마찬가지로 다음과 같은 제약조건을 갖는다.

$$\sum_{i=1}^n a_{ij} = 1, \quad j = 1, \dots, n \quad (24)$$

그림 3의 입출력 관계를 벡터형태로 나타내면 다음과 같다.

$$p(e(k)) = Ap(e(k-1)) \quad (25)$$

여기서 확률행렬 $A = \{a_{ij}, i, j = 1, \dots, n$. 이산시간 k 에서 오차신호의 확률은 $k-1$ 에서의 확률값과 파라미터 행렬 A 로부터 구할 수 있다. 다시 말해, 확률변수의 초기값 $p(e(0))$ 와 행렬 A 를 알면 변수 $e(k)$ 의 확률을 연속적으로 계산할 수 있다. 네트워크의 파라미터로 주어지는 조건부 확률은 시스템 관측값에 대하여 알맞은 학습 알고리즘을 이용하여 결정할 수 있으며, 이것을 DBN 파라미터 학습 또는 파라미터 추정이라고 한다[30]. 주로 maximum likelihood [31] 또는 expectation-maximization 알고리즘 [31]을 DBN 파라미터 학습에 적용하고 있으나 온라인 학습 시 다소 어려움이 대두되고 있다[29]. 본 논문은 [29]에서 제안한 DBN의 온라인 학습법을 이용하여 그림 3의 파라미터를 온라인 추정한다.

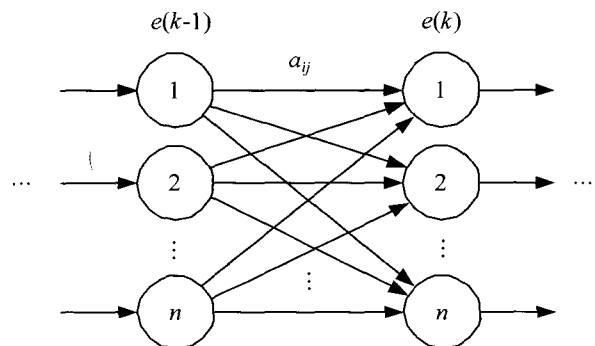


그림 3. 오차 신호의 DBN 모델.
Fig. 3. A DBN model for system errors.

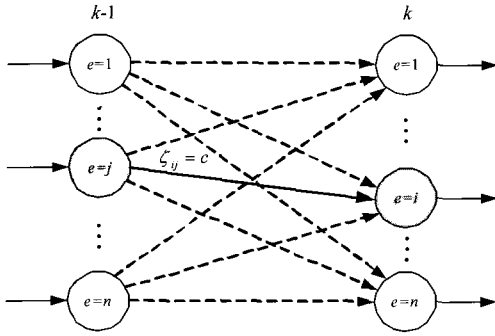


그림 4. DBN 파라미터 학습.
Fig. 4. DBN parameter learning.

4. DBN 온라인 파라미터 학습

먼저, (23)의 DBN 파라미터를 다음과 같이 정의하면,

$$a_{ij}(k) = \alpha m_{ij}(k) \tag{26}$$

여기서 α 는 (24)의 조건을 만족시키기 위하여 적용되는 정규화 상수(normalizing constant)이며, m_{ij} 는 평균 가능성으로서 다음과 같이 주어진다.

$$m_{ij}(k) = \frac{1}{k} \sum_{l=1}^k \zeta_{ij}(l) \tag{27}$$

여기서 ζ_{ij} 는 시스템 관측으로부터 결정되는 변수이며, m_{ij} 은 초기시간에서 현재시간 k 까지 ζ_{ij} 의 평균값으로 구해진다. (27)은 이전 m_{ij} 값과 현재시간 k 에서의 ζ_{ij} 값으로 나누어 다음과 같이 표현된다.

$$m_{ij}(k) = \left(\frac{k-1}{k}\right)m_{ij}(k-1) + \left(\frac{1}{k}\right)\zeta_{ij}(k) \tag{28}$$

여기서 현재시간 k 에서 ζ_{ij} 만을 결정하면 m_{ij} 가 회귀적으로(recursively) 계산된다. 앞에서도 언급한 바와 같이, ζ_{ij} 는 시스템 상태 관측으로부터 결정되며, 다음과 같은 규칙을 따른다.

$$\zeta_{ij}(k) = \begin{cases} c, & \text{if } e(k) = i \text{ and } e(k-1) = j \\ 0, & \text{otherwise} \end{cases} \tag{29}$$

여기서 $c > 0$ 이다. (29)에서 $k-1$ 에서 시스템 오차 $e(k-1)$ 이 j 번째이고, k 에서 $e(k)$ 가 i 번째 변수일 경우 ζ_{ij} 는 c 로 주어지며, 그렇지 않은 경우는 $\zeta_{ij} = 0$ 으로 선택된다. 그림 4는 이 내용에 대한 도해를 나타낸다. 요약하면, 시스템의 관측을 통해 (29)의 ζ_{ij} 을 결정하여 (28)의 m_{ij} 를 갱신한 후 정규화 계수 α 를 적용하여 최종적으로 (26)의 확률을 계산한다. (28)의 파라미터 수렴특성에 대한 해석은 [29]에 주어져 있다.

5. 슬라이딩 윈도우 기법

(28)의 파라미터 추정법은 초기시간에서 현재시간 k 까지의 모든 데이터를 이용하여 파라미터 값을 갱신한다. 이것은 실시간에서 많은 연산시간을 요구하며, 현재시간에서 먼

이전의 데이터는 현재 값을 추정하는데 있어 사실상 큰 영향을 주지 않는다[32].

이것은 현재시간에서의 파라미터 추정은 초기시간에서 현재시간 k 까지의 모든 데이터는 사실상 필요치 않으며, 어느 특정시간 $k_0 > 0$ 부터 현재시간의 데이터만 필요하다는 것을 의미한다. (28)에 k 대신 슬라이딩 윈도우 $N_w \in (0, k)$ 을 적용함으로써 이 내용을 내포할 수 있으며, (28)은 다음과 같이 수정될 수 있다.

$$m_{ij}(k) = \frac{1}{N_w} \sum_{l=k-N_w}^k \zeta_{ij}(l) \tag{30}$$

$$= \left(\frac{N_w-1}{N_w}\right)m_{ij}(k-1) + \left(\frac{1}{N_w}\right)\zeta_{ij}(k)$$

여기서 N_w 가 크면 비교적 많은 이전 데이터를 사용하게 되며, 반대로 작은 값의 N_w 는 비교적 최근의 데이터를 이용하여 파라미터 값을 갱신한다. 데이터의 특성에 맞는 적절한 크기의 윈도우를 결정하는 것이 중요하며, 최근엔 데이터의 분산(variance)정도에 따라 적응적으로 윈도우의 크기가 변화는 방법이 발표되었다[29].

IV. 사례 연구

본 논문에서 제안한 제어 알고리즘은 비선형, 시변 및 확률 특성이 매우 강한 TCP 네트워크의 트래픽 제어에 적용하였다. 일반적으로 TCP는 네트워크의 혼잡(congestion)을 방지하기 위하여 네트워크 전송부(source)에 윈도우 기법을 이용하여 패킷의 크기를 조절하거나 네트워크 링크에 위치하는 라우터의 큐(queue)를 제어한다. 이런 기법들은 TCP 네트워크의 자원(resource)을 최대한로 활용하여 신뢰성(reliability)을 확보하는데 주목적이 있다[33]. 후자의 방법은 능동 큐 관리(Active Queue Management; AQM)라고도 하며, 라우터내의 버퍼의 넘침(overflow)을 방지함으로써 네트워크의 혼잡을 미리 방지한다. AQM 시스템은 일반적으로 오픈루프 제어방식과 피드백 제어방식으로 크게 나눌 수 있으며, Random Early Detection(RED) [34]가 대표적인 오픈루프 AQM 방식이다. 그 이후 P, PI 또는 PID 제어방식을 적용한 AQM [35,36], H_∞ 등과 같은 고급제어이론을 적용한 AQM [37,38], 그리고 퍼지나 신경회로망과 같은 지능 알고리즘을 이용하여 설계한 AQM [39,40] 등이 발표되었다. 피드백제어 방식을 적용한 AQM 시스템은 라우터의 큐를 미리 설정한 기준값에 항상 유지하기 위한 제어전략으로, 앞서 설명한 바와 같이 TCP의 복잡한 동특성으로 인해 기존의 피드백 제어기보다 다소 어려운 설계과정이 요구되고 있다.

1. 지능형 PID AQM 시스템

시뮬레이션을 위한 시나리오는 [39]의 TCP 네트워크의 토폴로지(topology)를 참조하였다. 유한개의 TCP 전송부와 도착지(destination)가 병목(bottleneck) 특성을 갖는 한 개의 링크에 각각 연결되어 있으며, 그 링크에 라우터가 위치해 있다. 이 TCP 토폴로지에 대한 전송부의 윈도우 크기 ω 와 큐의 크기 q 에 대한 동특성 방정식은 다음과 같다.

$$\dot{\omega}(t) = \frac{1}{R(t)} - \frac{\omega(t)\omega(t-R(t))}{2R(t-R(t))}u(t-R(t)) \quad (31)$$

$$\dot{q}(t) = \begin{cases} N(t)x(t) - C & \text{if } N(t)x(t) \geq C \\ 0 & \text{otherwise} \end{cases}$$

여기서 확률값으로 주어지는 $u \in [0,1]$ 는 제어입력을 나타내고, C 는 링크의 용량, N 은 TCP 연결개수, x 는 패킷(packet)의 전송률(transmission rate)로서 $x = \omega/R$ 로 정의되며, 전체 지연시간 $R = q/C + \tau$ 로 구해진다. 여기서 τ 는 확률변수로서 전송부에서 목적지까지의 지연시간을 나타낸다. 시뮬레이션을 위하여 각 파라미터 값 또한 [39]를 참조하여 필요한 경우 그 값을 변경하였다. 주어진 네트워크의 파라미터 값은 다음과 같다. TCP 패킷의 크기는 512 bytes, $C = 15$ Mb/sec, 라우터의 최대 버퍼용량 $Q = 800$ packets, 그리고 τ 는 [0.16, 0.24]초 사이의 uniform 확률 분포값으로 하였다. 이 시스템에 대한 제어목적은 라우터의 이동평균값 \bar{q} 에 대한 시스템 오차 $e(t) = r(t) - \bar{q}(t)$ 를 최소화하는 최적의 제어입력 u 를 2.2절에서 제안한 제어기 설계법을 통해 PID 제어기를 구성하고 실시간 제어 시 발생하는 오차를 온라인 학습을 통해 보상한다.

2. PID 제어 파라미터 학습

(31)의 시스템 방정식을 이용하여 오프라인 학습을 통해 (4)의 이산 PID 제어기를 설계한다. (31)에서 $N = 250$ 으로 고정하였으며, 설정치 $r = 400$ packets으로 하였다. 샘플링 시간 $h = 0.01$ 초로 하였으며 (4)에서 파라미터의 초기값은 [39]에서 선정한 값과 정의한 샘플링 시간을 적용하여 $w_1(0) = 5 \times 10^{-4}$, $w_2(0) = 2 \times 10^{-5}$, $w_3(0) = 3 \times 10^{-6}$ 으로 구하였다. 또한 (13)-(15)에서 $\eta' = 2.5 \times 10^{-5}$ 으로 선정하였으며, 원하는 제어사양으로는 주어진 제어시간 100초 동안 오버슈트가 발생하지 않고 정상상태 도달시간이 보다 빠른 응답을 가지는 것이며, 여러 번의 반복학습을 실시하여 이 사양을 만족하는 최적의 결과를 얻었다. 최종적으로 PID 파라미터 값은 $w_1 = 5.23 \times 10^{-4}$, $w_2 = 5.06 \times 10^{-4}$, $w_3 = 3.14 \times 10^{-6}$ 으로 선정되었다. 그림 5와 6은 이 파라미터 값에 대한 시스템 응답, 즉 라우터의 큐와 제어입력에 대한 그래프를 각각 나타낸다. 그림 5로부터 시스템 응답은 약 15초 부근에서 정상상태에 도달하였고, 과도시간에서 오버슈트 응답이 전혀 발생치 않음을 알 수 있다. 이것은 그림 6에서도 볼 수 있듯이 초기시간에서 보다 큰 입력값을 적용함으로써 가능할 수 있었다. 정상상태에서 약간의 리플은 시스템 응답이 확률특성을 가지기 때문이다.

3. 오차신호의 DBN 모델링

(31)에서 $q \in [0, Q]$ 이고 $r > 0$ 이므로 오차의 절대값 $|e| < Q$ 를 만족한다. 3.3절에서 제안한 오차 신호의 DBN 모델링을 실시하기 위하여, 우선 연속 오차신호를 다음과 같이 이산화 변수로 정의한다.

$$e = \{e_1, e_2, e_3, e_4, e_5\} \quad (32)$$

여기서 각 변수에 대한 정의는

$$\begin{aligned} e_1 &:= \{|e| < c_1\} \\ e_2 &:= \{|e| \in [c_1, c_2)\} \\ e_3 &:= \{|e| \in [c_2, c_3)\} \\ e_4 &:= \{|e| \in [c_3, c_4)\} \\ e_5 &:= \{|e| \in [c_4, Q)\} \end{aligned} \quad (33)$$

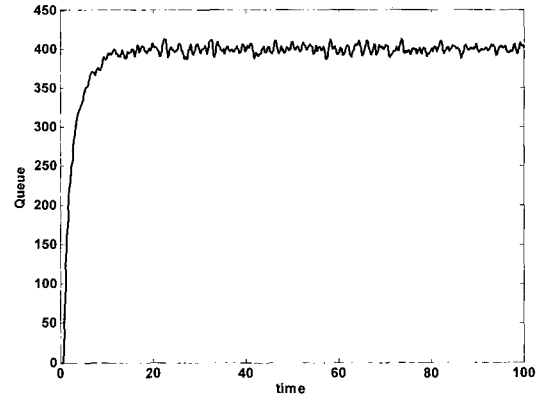


그림 5. 큐의 크기 (오프라인 학습).

Fig. 5. Queue length (offline learning)

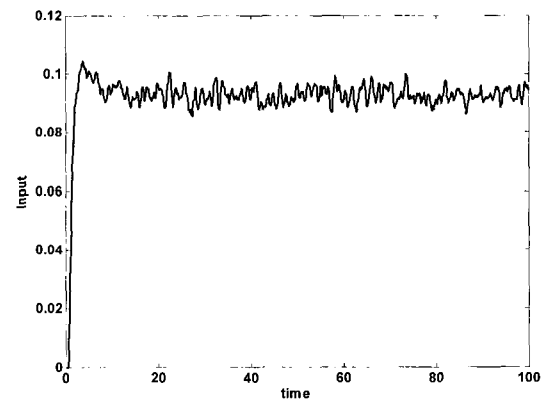


그림 6. 제어입력 (오프라인 학습).

Fig. 6. Control input (offline learning).

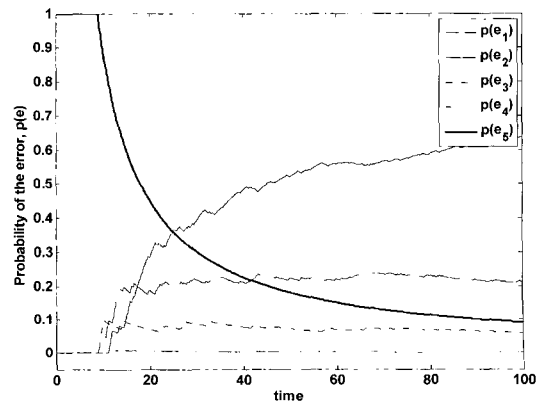


그림 7. 이산 오차신호의 확률값.

Fig. 7. Probability of system errors.

1) AQM 시스템의 제어입력은 패킷의 버림(drop) 확률을 나타내며, 보다 큰 제어입력은 라우터에 인가되는 패킷을 보다 많이 버림으로써 오버슈트를 방지한다[35,36].

여기서 $c_1 < c_2 < c_3 < c_4 < Q$. 그림 5의 시스템 응답에 대하여 DBN 모델링을 실시하였으며, 그림 7에서는 (33)에서 $c_1 = 5, c_2 = 10, c_3 = 15, c_4 = 20$ 으로 각각 정의하여 각 오차변수에 대한 확률값을 나타내었다. 이 결과로부터 과도 응답 구간에서 $p(e_5)$ 가 가장 큰 값을 가지며, 약 23초를 지나 정상상태에서 $p(e_1)$ 의 값이 현저히 증가함을 알 수 있었으며, 이는 시스템 오차가 시간이 갈수록 점점 줄어든다는 것을 확률로 보여준다.

4. 온라인 PID 제어

앞서 구현한 DBN 모델을 이용하여 3.1절에서 제안한 온라인 PID 제어를 구성하였다. 3.1절에서 언급한 바와 같이 실시간 제어 오차에 대한 온라인 학습은 w_1 값을 수정하는 것으로 실시되며, w_2 와 w_3 는 오프라인 학습에서 선정된 값을 그대로 적용한다. (30)의 $N_w = 7$ 로, (22)의 $\delta = 2$ 로 선정하였으며, 이것은 오차의 절대값이 10을 벗어나면 온라인 학습이 이루어짐을 의미한다(33 참조). 두 TCP 접속수의 경우에 대하여 시뮬레이션을 각각 실시하였으며, 앞서 구현 오프라인 PID 제어와 비교 및 검토하였다.

Case I: 오프라인 학습 시 고정되었던 TCP 접속 수를 그림 8과 같이 동적인 값으로 선정하여 시스템 응답을 나타내었다. 그림 9는 오프라인 및 온라인 PID 제어에 대한 큐의 크기를 보여주며, N 이 변하는 5군데 구간에서 큐의 크기 또한 급격히 바뀌는 것을 볼 수 있다. 특히 오버슈트 특성이 두 제어방식에서 공통으로 나타나며, 온라인제어 응답은 오프라인의 경우보다 정상상태로 다시 복귀하는 시간이 월등히 빠름을 알 수 있다. 다시 말해, 오프라인의 경우 약 28초의 과도응답 상태를 가지지만, 온라인 제어의 경우 약 4초의 과도시간을 가진다. 이것은 온라인 제어의 경우 시스템의 환경적 변화에 대한 대처 능력이 오프라인의 경우보다 월등히 우수함을 알 수 있다. 그림 10은 두 제어입력에 대한 파형을 보여주며, 정상상태에서의 입력값의 크기는 거의 비슷하나 과도응답에서 그 확연한 차이점을 볼 수 있다. 그림 11은 파라미터 w_1 의 변화에 대한 파형을 보여준다. 시스템 오차에 따라 새로운 값으로 갱신됨을 알 수 있으며, 동일한 방법으로 N 의 변화가 있는 구간에서는 큰 값의 폭으로 수정됨을 알 수 있다. 이것은 시스템 오차에 대하여 w_1 이 적응적으로 변하는 것을 보여준다.

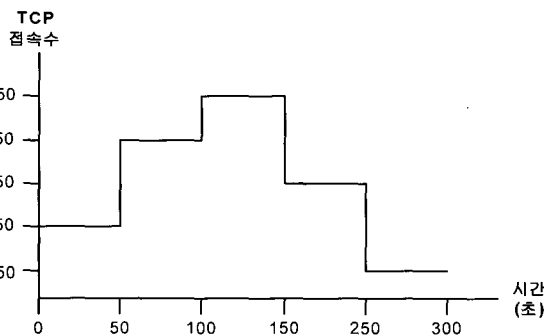


그림 8. TCP 접속수 (Case I).
Fig. 8. Number of TCP connections (Case I).

Case II: 이번 시뮬레이션은 $N=550$ 으로 증가시키고 큐의 설정치 $r=200$ 으로 줄였을 때 두 제어기에 대한 응답

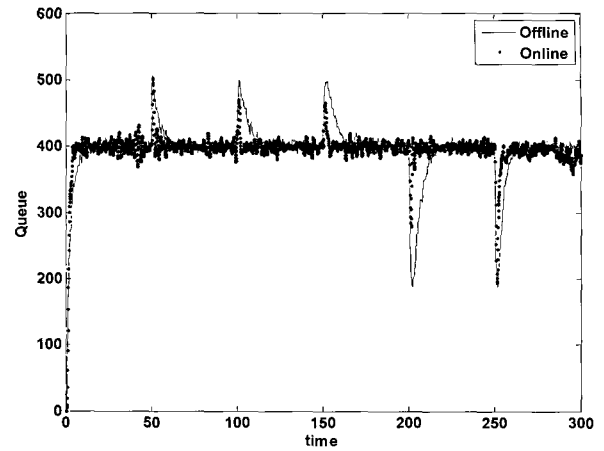


그림 9. 큐의 크기 (Case I).
Fig. 9. Queue length (Case I).

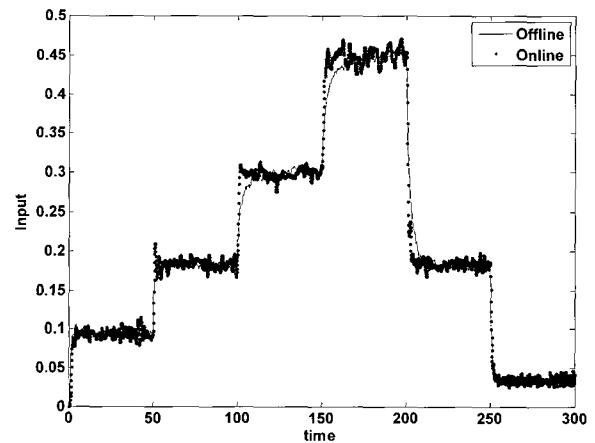


그림 10. 제어입력 (Case I).
Fig. 10. Control input (Case I).

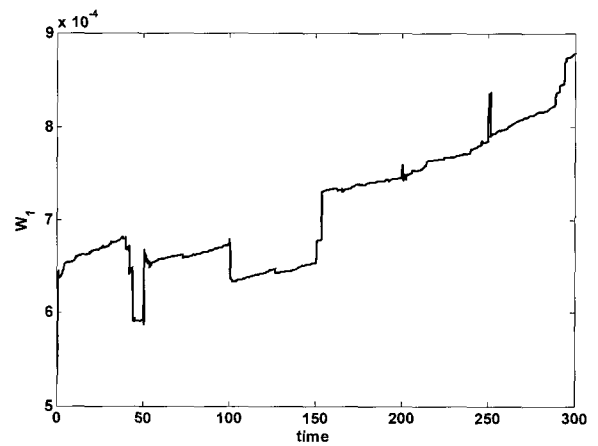


그림 11. w_1 변화 곡선 (Case I).
Fig. 11. Trajectory of w_1 (Case I).

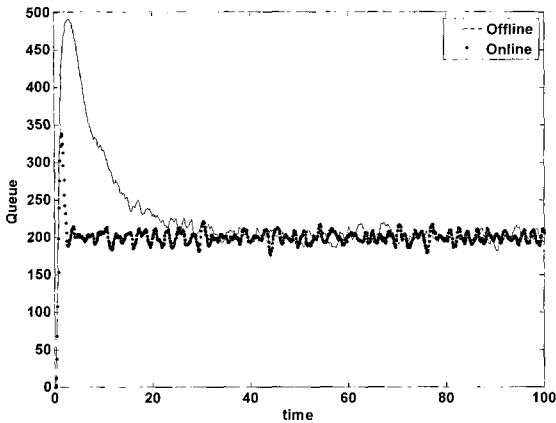


그림 12. 큐의 크기 (Case II).
Fig. 12. Queue length (Case II).

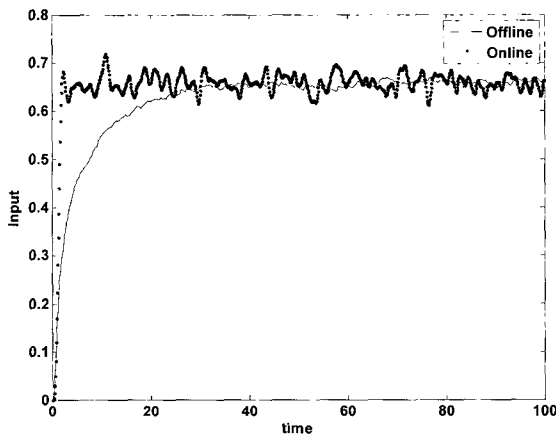


그림 13. 제어입력 (Case II).
Fig. 13. Control input (Case II).

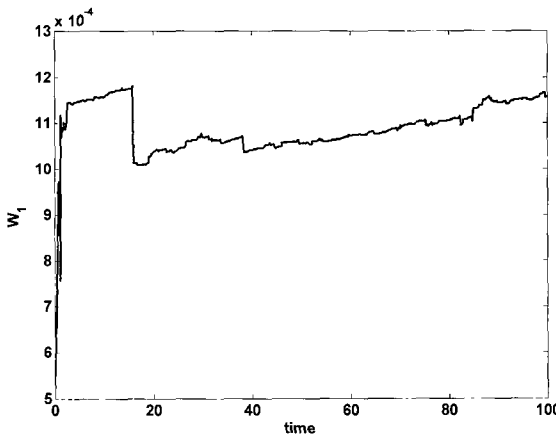


그림 14. w_1 변화 곡선 (Case II).
Fig. 14. Trajectory of w_1 (Case II).

을 비교하였다. 그림 12는 시스템 응답 파형을 각각 보여주며, 마찬가지로 과도응답에서 제어성능이 현저히 차이가 남을 알 수 있다. 공통적으로 초기상태에서 큰 오버슈트를 가

지며, 오프라인의 경우 약 480packets, 온라인 제어의 경우 약 340packets의 최대 오버슈트 값을 가진다. 정상상태 도달 시간도 상당히 차이가 나며 온라인의 경우 약 30초 정도가 개선되었다. 그림 13은 두 제어입력 파형을 나타내며, 보다 작은 설정치에 비해 N 의 급격한 증가로 인해 많은 트래픽이 인가됨으로 제어 입력값도 이에 대응하여 큰 값을 산출함을 확인할 수 있다. 이는 그림 6의 정상상태 응답보다 약 7배 증가함을 알 수 있다. 그림 14는 w_1 의 변화를 보여주며 과도상태에서 큰 변화를 보인 후 비교적 평형점 부근에 도달하는 궤적을 볼 수 있다.

V. 결론

본 논문은 학습을 통해 PID 제어기의 파라미터를 선정하였으며, 실시간 시스템 오차를 보상하기 위한 온라인 PID 제어 알고리즘을 제안하였다. 또한 시스템 오차에 대한 DBN 모델링을 실시하였으며, 실시간에서 온라인 제어의 실행여부를 결정하는 의사결정법에 적용하였다. 제안한 제어기법은 시변 및 확률특성이 매우 강한 TCP 네트워크의 AQM 시스템을 구성하는데 사용되었다. 기존의 고정 PID제어와의 컴퓨터 시뮬레이션 비교분석을 통하여 성능의 우수성을 검증하였다. 시뮬레이션 결과로부터 시스템 특성의 변화에도 제안한 PID는 만족스러운 제어성능을 발휘하는 것을 볼 수 있었다. 향후 연구에서 본 논문의 제어기법을 산업현장의 실제 시스템에 적용함으로써 실시간 실험을 실시하여 실제 구현에 있어 그 타당성을 검증하고자 한다.

참고문헌

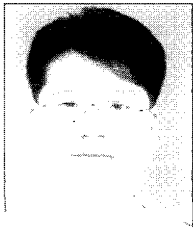
- [1] Y.-C. Chenh, L.-Q. Ye, F. Chuang, and W.-Y. Cai, "Anthropomorphic intelligent PID control and its application in the hydro turbine governor," *Proc. of Int. Conf. on Machine Learning & Cybernetics*, Beijing, China, pp. 391-395, 2002.
- [2] F. Karray, W. Gueaieb, and S. Al-Sharhan, "The hierarchical expert tuning of PID controllers using tools of soft computing," *IEEE Trans. on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 32, no. 1, pp. 77-90, 2002.
- [3] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall, Upper Saddle River, NJ, 1999.
- [4] B. Kosko, *Fuzzy Engineering*, Prentice Hall, Upper Saddle River, NJ, 1997.
- [5] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Professional, 1989.
- [6] D. Dasgupta, *Artificial Immune Systems and Their Applications*, Springer, 1998.
- [7] Y. Yu, H. Ying, and Z. Bi, "The dynamic fuzzy method to tune the weight factors of neural fuzzy PID controller," *Proc. of IEEE Int. Joint Conf. on Neural Networks*, Budapest, Hungary, pp. 2397-2402, 2004.

- [8] M. Guzelkaya, I. Eksin, and E. Yesil, "Self-tuning of PID fuzzy logic controller coefficients via relative rate observer," *Engineering Application of Artificial Intelligence*, vol. 16, pp. 227-236, 2003.
- [9] D. D. Kukolj, S. B. Kuzmanovic, and E. Levi, "Design of a PID-like compound fuzzy logic controller," *Engineering Application of Artificial Intelligence*, vol. 14, pp. 785-803, 2001.
- [10] L. Tian, "Intelligent self-tuning of PID control for the robotic testing system for human musculoskeletal joints test," *Annals of Biomedical Engineering*, vol. 32, no. 6, pp. 899-909, 2004.
- [11] G. M. Houry, M. Saad, H. Y. Kanaan, and C. Asmar, "Fuzzy PID control of a five DOF robot arm," *J. of Intelligent & Robotic Systems*, vol. 40, pp. 299-320, 2004.
- [12] G. Tan, H. Xiao, and Y. Wang, "Optimal fuzzy PID controller with adjustable factors and its application to intelligent artificial legs," *High Technology Letters*, vol. 10, no. 2, pp. 73-77, 2004.
- [13] L. Reznik, O. Ghanayem, and A. Bourmistrov, "PID plus fuzzy controller structures as design base for industrial applications," *Engineering Application of Artificial Intelligence*, vol. 13, pp. 419-430, 2000.
- [14] T. R. Rangaswamy, J. Shanmugam, and K. P. Mohammed, "Adaptive fuzzy tuned PID controller for combustion of utility boiler," *Control & Intelligent Systems*, vol. 33, no. 1, pp. 63-71, 2005.
- [15] A. S. Zayed, A. Hussain, and M. J. Grimble, "A nonlinear PID-based multiple controller incorporating a multilayered neural network learning submodel," *Control & Intelligent Systems*, vol. 34, no. 3, pp. 177-184, 2006.
- [16] H. Shu and Y. Pi, "PID neural networks for time-delay systems," *Computer & Chemical Engineering*, vol. 24, pp. 859-862, 2000.
- [17] D. Garg and N. Gulati, "Neural network based intelligent control and PID control of a magnetic levitation system," *Proc. of ASME Dynamic Systems and Control Division*, New Orleans, LA, pp. 1013-1020, 2002.
- [18] C. Riverol and V. Napolitano, "Use of neural networks as a tuning method for an adaptive PID application in a heat exchanger," *Institution of Chemical Engineers*, vol. 78, Part A, pp. 1115-1119, 2000.
- [19] M. Faradadi, A. S. Ghafari, and S. K. Hannani, "PID neural network control of SUT building energy management system," *Proc. of IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, Monterey, CA, pp. 682-686, 2005.
- [20] G. Zhenhai and Z. Bo, "Vehicle lane keeping of adaptive PID control with BP neural network self-tuning," *Proc. of IEEE Intelligent Vehicle Symposium*, Las Vegas, NV, pp. 84-87, 2005.
- [21] M. Trusca and G. Lazea, "An adaptive PID learning controller for periodic robot motion," *Proc. of IEEE Conf. on Control Applications*, Istanbul, Turkey, pp. 686-689, 2003.
- [22] R. A. Krohling and J. P. Rey, "Design of optimal disturbance rejection PID controllers using Genetic algorithm," *IEEE Trans. on Evolutionary Computation*, vol. 5, no. 1, pp. 78-82, 2001.
- [23] D. S. Pereira and J. O. Pinto, "Genetic algorithm based system identification and PID tuning for optimum adaptive control," *Proc. of IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, Monterey, CA, pp. 801-806, 2005.
- [24] Y. J. Lee, H. C. Cho, and K. S. Lee, "Immune algorithm based active PID control for structure systems," *J. of Mechanical Science & Technology*, vol. 20, no. 11, pp. 1823-1833, 2006.
- [25] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, Prentice Hall, Upper Saddle River, NJ, 2006.
- [26] M. Saerens and A. Soquet, "Neural controller based on back-propagation algorithm," *IEE Proceedings - F*, vol. 138, no. 1, pp. 55-62, 1991.
- [27] S. Ablameyko, M. Gori, L. Goras, and V. Piuri, editors, *Impact of Neural Networks on Signal Processing and Communications, of Limitations and Future Trends in Neural Computation*, NATO Science Series, 2003.
- [28] T. M. Mitchell, *Machine Learning*, McGraw-Hill International Editions, 1997.
- [29] H. C. Cho, "Dynamic Bayesian networks for online stochastic modeling," *Ph.D. Dissertation*, University of Nevada-Reno, 2006.
- [30] K. Murphy, "Dynamic Bayesian networks: Representation, Inference and Learning." *Ph. D. Dissertation*, University of California-Berkeley, 2002.
- [31] T. K. Moon and W. C. Stirling, *Mathematical Methods and Algorithms for Signal Processing*, Prentice Hall, Upper Saddle River, NJ, 2000.
- [32] P. Baldi and Y. Chauvin, "Smooth on-line learning algorithm for hidden Markov models," *Neural Computation*, vol. 6, no. 2, pp. 307-318, 1994.
- [33] V. Jacobson and M. Karels, "Congestion avoidance and control," *Proc. of ACM SIGCOMM*, pp. 314-329, 1988.
- [34] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. on Networking*, vol. 1, no. 4, pp. 397-413, 1993.
- [35] C. V. Hollot, V. Misra, D. Towsley, and W. Gong, "Analysis and design of controllers for AQM routers supporting TCP flows," *IEEE Trans. on Automatic Control*, vol. 47, no. 6, pp. 945-959, 2002.

- [36] K. B. Kim and S. H. Low, "Analysis and design of AQM based on state-space models for stabilizing TCP," *Proc. of American Control Conference*, pp. 260-265, 2003.
- [37] R. A. DeCarlo, S. H. Zak, and G. P. Matthews, "Variable structure control of nonlinear multivariable systems: A tutorial," *Proc. of the IEEE*, vol. 76, no. 3, pp. 212-232, 1998.
- [38] R. Fengyuan, L. Chuang, Y. Xunhe, S. Xiuming, and W. Fubao, "A robust active queue management

algorithm based on sliding mode variable structure control," *Proc. of IEEE INFOCOM*, pp. 13-20, 2002.

- [39] H. C. Cho, M. S. Fadali, and H. Lee, "Neural network control for TCP network congestion," *Proc. of American Control Conference*, pp. 3480-3485, 2005.
- [40] Y. H. Aoul, A. Nafaa, D. Negru, and A. Mehaoua, "FAFC: fast adaptive fuzzy AQM controller for TCP/IP networks," *IEEE Global Telecommunications Conf.*, pp. 1319-1323, 2004.



조 현철

1971년 9월 1일생. 1997년 2월 부경대학교 제어계측공학과 졸업. 1999년 2월 동아대학교 전기공학과 졸업(석사). 2006년 8월 미국 네바다대학교 전기공학과 졸업(Ph.D.). 2006년~현재 동아대 전기공학과 제어 및 시스템 연구실 포

닥연구원. 관심분야는 확률시스템, 지능제어, 신경회로망.

이 영진

제어·자동화·시스템공학회 논문지 제 10 권 제 12 호 p. 1147 참조.



이진우

1972년 7월 15일생. 1998년 경남대 전기공학과 졸업. 2000년 동아대 대학원 전기공학과 졸업(석사). 2006년 동대학원 졸업 공학박사. 현재 동아대 전기공학과 제어 및 시스템 연구실 포닥연구원. 관심분야는 영상처리, 적응제어,

지능제어, 항만 자동화 시스템 개발.

이권순

제어·자동화·시스템공학회 논문지 제 12 권 제 5 호 p. 431 참조.