

유전자 알고리즘을 이용한 네트워크 기반 제어 시스템의 원격 제어기 설계

김현희*, 이경창(부산대 대학원 지능기계공학과), 이석(부산대 기계공학부)

Remote Controller Design of Networked Control System using Genetic Algorithm

H. H. Kim, K. C. Lee(Intelligent Mech. Eng. School, PNU), S. Lee(Mechanical Eng. School, PNU)

ABSTRACT

As many sensors and actuators are used in many automated systems, various industrial networks are adopted for digital control system. In order to take advantages of the networking, however, the network implementation should be carefully designed to satisfy real-time requirements considering network delays. This paper presents the implementation scheme of a networked control system via Profibus-DP network. More specifically, the effect of the network delay on the control performance was evaluated on a Profibus-DP testbed, and a GA based PID tuning algorithm is proposed to demonstrate the feasibility of the networked control system.

Key Words Networked control system(네트워크 기반 제어 시스템), Remote controller(원격 제어기), Profibus-DP(프로피버스-DP) Fieldbus(필드버스), Network delay(네트워크 지연), Genetic algorithm(유전자 알고리즘)

1. 서론

최근 고성능의 마이크로 프로세서가 발전함에 따라, 대다수의 제어 시스템은 디지털 제어 시스템으로 대체되고 있다. 특히, 산업용 컴퓨터를 이용한 디지털 제어 시스템은 뛰어난 유연성(flexibility)과 적응성(adaptability)으로 인하여, 조립 자동화(assembly automation)나 공정 자동화(process automation) 분야에서 활발하게 적용되고 있는 추세이다. 또한, 향상된 시스템 성능을 얻기 위하여 더 많은 수의 필드기기를(field devices)이 사용됨에 따라, 제어 시스템에서 처리되어야 할 데이터량이 급속하게 증가되고 있다.

그래서 산업용 네트워크(industrial network)에 많은 관심을 기울이게 되었고, 활발한 연구가 진행되었다. 연구 결과 산업용 네트워크는 제어 시스템의 성능향상과 단순화, 신뢰성 증대와 같은 장점을 얻을 수 있게 되었다[1].

일반적으로, 산업용 네트워크에서 전송되는 데이터는 산발적 실시간과 주기적 실시간, 그리고 비실시간 데이터로 구분되며, 이러한 데이터들은 단일한 전송 매체를 통하여 동시에 전송될 수 있어야 한다.

본 논문에서는 필드기기의 실시간 요구 조건을 만족시킬 수 있는 네트워크 기반 제어 시스템(networked control system)의 구조와 유전자 알고리즘(Genetic Algorithm, GA)을 이용한 원격 제어기의 설계 방법을 제안하고, 이를 이용한 네트워크 기반 제어 시스템의 성능을 실험적으로 평가하였다.

2. Profibus-DP 프로토콜

2.1 Profibus-DP 프로토콜의 구조

Profibus-DP는 공정 시스템이나 제어 시스템에서의 적용을 위하여 개발되었으며, 이러한 응용물에서 필요한 실시간 요구 조건을 만족시키기 위하여 OSI 참조 모델 7개 계층 중 1, 2계층인 PHY(PHYsical layer)와 FDL(Fieldbus Data Link layer)만을 사용한다.

Profibus-DP의 전송 매체로는 RS485가 사용되며, 전송 속도는 9.6Kbps에서 12Mbps까지 지원된다. Profibus-DP에 접속되는 스테이션들은 마스터 스테이션(master station)과 슬레이브 스테이션(slave station)으로 구분된다. 일반적으로, 마스터는 슬레이브에게 데이터를 전송하거나, 전송을 요청할 수 있으며, 슬레이브는 마스터의 요청에 의해서만 통신에 참여할 수 있다.

2.2 Profibus-DP의 성능

본 절에서는 Profibus-DP가 분산된 필드기기에서 요구되는 실시간 요구 조건을 만족시키는 것을 평가하기 위하여 실험 모델을 구성하고, 성능 평가를 수행하였다. 실험 모델은 마스터 역할을 하는 PC의 컴퓨터와 슬레이브 역할을 하는 다수의 컴퓨터로 구성되어 있으며, 각 컴퓨터에는 Softing사의 PROF1-Board를 장착하였다[2].

그림 1은 Profibus-DP의 슬레이브 수에 따른 반응시간(overall response time)을 나타내고 있다. 여기에서, 반응시간이란 마스터가 첫 번째 슬레이브에게 통신 요청을 보내고 나서부터 마지막 슬레이브로부터 응답을 받을 때까지 걸린 시간으로 정의하였다. 이 실험에서 마스터는 슬레이브와의 폴링을 위하여 2바이트(byte)의 메시지를 보내며, 슬레이브는 2바이트의 메시지로 응답한다. 또한, 폴링 주기는 대다수의 실시간 시스템에서 사용되는 10msec로 설정하였다[3].

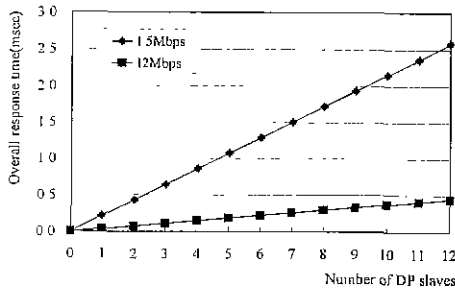


Fig 1 Overall response time of Profibus-DP

3. 네트워크 기반 제어 시스템

3.1 네트워크 기반 제어 시스템의 구조

Profibus-DP에서의 네트워크 기반 제어 시스템의 구조는 마스터/슬레이브 스테이션은 입/출력 레이어 프로세스 (input/output data process), 통신 프로세스 (communication process) 및 I/O 계측 프로세스(I/O instrumentation process)로 구성되어 있다 또한 네트워크 기반 제어 시스템에서 통신은 통신 사이클(communication cycle)과 계측 사이클 (instrumentation cycle)에 의하여 이루어진다 통신 사이클에서는 DPRAM과 통신 프로세스를 이용하여 마스터의 제어 정보와 슬레이브의 출력 정보가 서로 교환되며, 계측 사이클에서는 I/O 계측 프로세스를 통하여 플랜트 제어에 필요한 정보가 교환된다.

3.2 Profibus-DP에서의 네트워크 지연 특성

네트워크 기반 제어 시스템에서는 펄드버스의 사용으로 인하여, 제어 신호 및 피드백 신호의 전송에 지연이 발생한다. 이러한 네트워크 지연은 제어기가 경수 배의 샘플링 시간 이후에 플랜트의 피드백 신호를 사용하게 함으로서, 제어 성능을 악화시키게 된다[4][5]

네트워크 지연은 다음과 같은 원인에 의하여 발생한다. 첫째, 네트워크 기반 제어 시스템의 각 프로세스들이 완벽한 동기를 이루었다고 하더라도, 네트워크 지연은 Profibus-DP 프로토콜의 특성에 의하여 발생한다 둘째, 네트워크 지연은 각 프로세스들간의 비동기화에 의하여 발생한다

일반적으로, 이러한 네트워크 지연은 플랜트의 제어 성능을 악화시키기 때문에, 네트워크 지연이 최소화되도록 네트워크 기반 제어 시스템이 구성되어야 하며, 원격 제어를 설계할 때 네트워크 지연에 대한 세심한 고려가 필요하다 특히, 본 연구에서 설계된 Profibus-DP를 이용한 네트워크 기반 제어 시스템의 경우, 세 배 이상의 통신 주기에 해당하는 지연이 발생하는 것을 확인하였다.

4. GA에 의한 원격 제어기 설계

유전자 알고리즘은 1975년 Holland에 의하여 소개된 직자생존의 법칙에 기반을 둔 최적화 이론으로서, 최적화 문제나 PID 튜닝과 같은 분야에서 효과적인 것으로 알려져 있다 GA는 전체 탐색 공간에서 여러 개의 스트링(string), 즉, 인구(population)를 이용하여 동시에 탐색을 수행하기 때문에, 전역 최적값으로 수렴할 가능성이 클 뿐만 아니라, 미분 불가능이나 비선형, 멀티모달(multi-

modal)과 같은 문제에서 강인한 특성을 발휘한다고 알려져 있다[6]

본 절에서는 네트워크 기반 제어 시스템에서 발생하는 네트워크 지연을 보상하기 위한 방법으로서, GA에 의한 원격 제어기의 설계 방법을 설명한다.

4.1 GA에 의한 원격 제어기의 구조

그림 2에는 GA에 의한 원격 제어기 설계 방법에 대한 개념도를 나타내었다 그림의 아래 부분은 플랜트와 원격 제어기로 구성된 네트워크 기반 제어 시스템을 나타내고 있다 여기에서, 플랜트와 원격 제어기는 펄드버스로 연결되어 있으며, 플랜트는 Profibus-DP의 슬레이브 스테이션으로, 원격 제어기는 마스터 스테이션으로 설정되어 있다 그림의 위 부분은 유전 연산자(genetic operator)에 의하여 진화되는 PID 게인 조정기(PID tuner)로서, 다음과 같은 세 부분으로 구성되어 있다 (a) 플랜트 출력(y)이나 최대오버슈트(M_p), 정착시간(T_s)과 같은 시스템의 스텝 응답 특성으로부터 스트링의 적합도를 계산하는 적합도 계산기(fitness calculator), (b) 각 스트링의 적합도를 이용하여 차순(offspring)을 생성하는 GA 부분, (c) 스트링을 PID 게인으로 변환하는 PID 계산기(PID calculator)

GA에 의한 원격 제어기의 튜닝 절차는 다음과 같다

1 기준 입력(reference input, r), 기준 오버슈트(reference overshoot, M_p), 기준 정착시간(reference settling time, T_s)과 같은 플랜트의 설계사양을 선정하고, GA의 초기화 루틴을 통하여 인구(population) 수 만큼의 PID 게인 K_p, K_i, K_d 를 구한다.

2. 시스템에 기준 입력을 인가하고, K_p, K_i, K_d 를 변화시켜 가면서 플랜트의 출력(y), 최대오버슈트(M_p), 정착시간(T_s)을 구한다.

3 플랜트의 스텝 응답 특성과 설계사양(i, M_p, T_s)을 이용하여 적합도를 계산한다

4 여러 개의 서브스트링(sub-string)을 연결하여 하나의 스트링을 형성하는 다변수 크딩 방법(multi parameter coding)[7]을 이용하여, 3개의 PID 게인을 하나의 스트링으로 변환한다 그리고 나서, 적합도와 유전 연산자(genetic operator)를 이용하여 새로운 스트링을 생성한다

5. 생성된 스트링을 PID 게인으로 변환하고, GA의 종단조건(termination condition) 만족되거나 수행 횟수가 끝날 때까지 2~5 단계를 반복한다

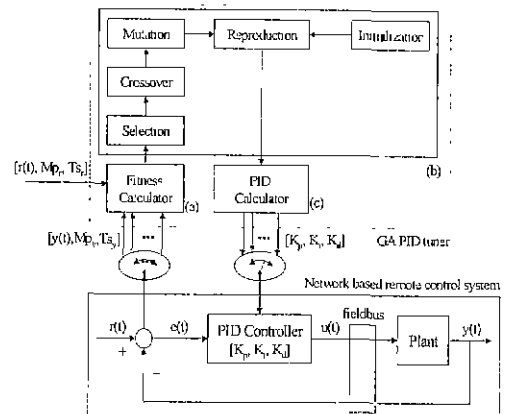


Fig. 2 Schematic diagram of remote controller tuned by GA

5. DC 모터를 이용한 성능 평가

본 절에서는 제안된 GA에 의한 인적 제어기의 성능을 평가하기 위하여, Profibus-DP를 이용한 네트워크 기반 제어 시스템의 실험 모델을 구현하고, 성능 평가를 수행하였다

5.1 플랜트 선정

본 절에서는 네트워크 기반 제어 시스템의 성능을 평가하기 위하여, Tamagawa사의 TS3728 DC 모터를 플랜트로 선정하였다. 플랜트의 제어 입력을 위하여 12비트 D/A 컨버터, 인코더 출력을 위하여 16비트 카운터를 사용하였으며, 모터 샘플링 시간은 10ms로 기준입력은 1000rpm으로 설정하여 속도제어를 수행하였다

스텝응답법을 이용하여 구한 DC 모터의 전달함수는 식 (3)과 같다[8]

$$G(s) = \frac{y(s)}{u(s)} = \frac{543}{0.12s + 1} \quad (3)$$

여기에서, PID 제어기의 설계사양을 최대오버슈트는 10% 이하, 5% 정착시간은 0.8sec 이하, 감쇠비는 0.5 이하로 설정하고, 근궤적법(root locus)을 이용하여 PID 제어기를 설계하면 식 (4)와 같다[8]

$$G_{PID}(s) = K_d s + K_p + \frac{K_i}{s} = \frac{0.0001s^2 + 0.011s + 0.2}{s} \quad (4)$$

5.2 GA에 의한 원격 제어기의 성능 평가

본 절에서는 제안된 원격 제어기의 성능의 우수성을 평가하기 위하여, 고전적인 방법에 의하여 설계된 PID 제어기의 성능과 비교하였다

고전적인 방법을 적용하기 위하여, 네트워크 지연을 공정 시스템의 시간 지연으로서 가정하였으며, 가장 널리 사용되고 있는 Ziegler-Nichols 계수조정법[9]을 사용하여 PID 제어기를 설계하였다. 여기에서, Ziegler-Nichols 계수조정법에서 시간 지연은 3 샘플링 타임, 즉 30msec가 발생하는 것으로 가정하였다

본 절에서 사용된 GA의 파라미터들은 표 1과 같다. 먼저, GA의 개체 수는 100으로 하였으며, 교배 확률은 0.6, 돌연변이 확률은 0.01로 설정하였다 또한, 스트링의 길이 n은 각각 15비트로 설정하였으며, 축적 계수 m은 4로 설정하였다 또한, 그림 9와 같은 PID 제이기가 직접 연결되었을 경우의 응답보다 더 좋은 특성을 얻기 위하여 적합도 함수의 파라미터들을 표 1과 같이 선정하였다 고전적 설계기준 $f_1(y)$ 의 Mp는 7%, Ts는 0.55sec로 설정하였으며, 최적제어적 설계 기준 $f_2(t)$ 의 적분 구간은 0에서 2초로 설정하였다 또한, 가중치 a는 0.4로, b는 0.2로 설정하였다

Table 1. Parameter of GA

	α	β	γ
Population size	30		
Crossover probability	0.9		
Mutation probability	0.01		
String length (bit)	15	15	15
Scale factor	4	4	4
Mp_r of $f_1(Mp_r, Mp_r)$	7%		
Ts_r of $f_1(Ts_r, Ts_r)$	0.55sec		
T of $f_2(t)$	0 - 2sec		
a	0.4		
b	0.2		

4.2 스트링의 구성

유전자 알고리즘에서는 PID 튜닝을 위하여 제어기의 Kp, Ki, Kd와 같은 PID 계인을 그대로 이용하지 않고, 스트링으로 변환하여 사용한다

본 논문에서는 고전적인 제어기 설계 방법에 의하여 구한 PID 계인에 튜닝 계수(tuning factor)를 곱한 값을 제어기의 PID 계인으로 정의하고, 이 튜닝 계수를 GA의 스트링으로 선택하는 튜닝 계수 코딩 방법(tuning factor coding method)을 제안한다

4.3 적합도 함수

GA에서 적합도 함수는 스트링의 우수성을 평가하기 위한 척도로서, 적자생존의 법칙에 따라 보다 높은 적합도를 가진 스트링이 더 많은 자손을 생성할 수 있는 확률이 높게 된다

본 논문에서는 식 (1)과 같이 가중치를 곱한 고전적 설계 기준과 최적 제어적 설계 기준의 조합을 적합도 함수로 설정하였다

$$f(t) = 1 - (af_1(Mp_r, Mp_r) + f_1(Ts_r, Ts_r)) + bf_2(t) \quad (1)$$

$$\text{where } f_1(t, y) = \begin{cases} 0 & \text{at } y \leq r \\ \frac{y-r}{r} & \text{at } r < y \leq 2r \\ 1 & \text{at } y > 2r \end{cases}$$

$$f_2(t) = \frac{\int_0^t (y(t) - r(t))^2 dt}{\int_0^t r(t)^2 dt}$$

식에서 첫 번째 항인 첫 번째 항인 $f_1(Mp_r, Mp_r)$ 와 $f_1(Ts_r, Ts_r)$ 는 플랜트의 Mp_r , Ts_r 와 설계사양 Mp_r , Ts_r 로의 차이를 설계 사양으로 나타내는 값으로서 0에서 1의 범위를 가진다. $f_2(t)$ 는 0에서 1의 범위를 가진다. 특히, 각 설계 기준에 대한 비중을 다르게 하기 위하여, 각 항에 가중치를 곱하였으며, 가중치들의 합은 1로 설정하였다. 이러한 방식에 의하여, 가중치를 조절함에 의하여, 두 개의 설계 기준 중 하나를 우선적으로 만족시킬 수가 있게 된다

4.4 유전 연산자

단순 GA(simple GA)에서는 선택(selection)과 교배(crossover) 돌연변이(mutation)와 같은 유전 연산자를 사용한다. 본 논문에서는 선택 방법으로서 stochastic remainder technique을 사용하였으며, 다수의 스트링의 적합도가 유사한 경우 부모와 동일한 비율의 자손들이 생성되는 것을 막기 위하여, 식 (2)와 같은 선형 스케일링(linear scaling)을 사용하였다[6]

$$f' = pf + q \quad (2)$$

$$\text{where } c = \frac{f_{\max} - f_{\min}}{f_{\max} - f_{\min}}$$

$$p = \frac{(c - 1) f_{\max}}{f_{\max} - f_{\min}}$$

$$q = \frac{f_{\text{avg}} (f_{\max} - c f_{\text{avg}})}{f_{\max} - f_{\text{avg}}}$$

식에서 f는 적합도 함수이며, f'은 스케일링(scaling)된 적합도 함수이다. 또한, f_{\max} 는 적합도 함수의 최대값이며 f_{\min} 는 최소값, f_{avg} 는 평균값이다.

교배 연산자는 스트링들의 더 많은 탐색(perturbation)을 위하여 균일 교배(uniform crossover)를 사용하였고, 돌연변이 연산자로는 한점 돌연변이(one point mutation)를 사용하였다 또한 더 빠른 진화를 위하여 원 세대의 최상의 개체가 다음 세대까지 살아 남을 수 있는 엘리트 스트 모델(elitist model)을 사용하였다[6]

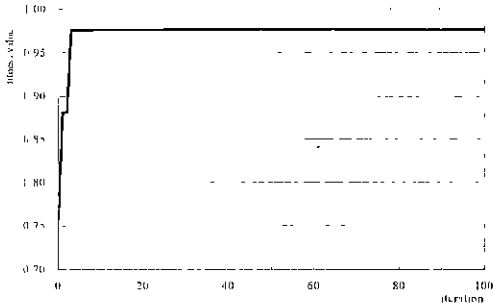


Fig 3(a) Fitness function in PID gain tuning

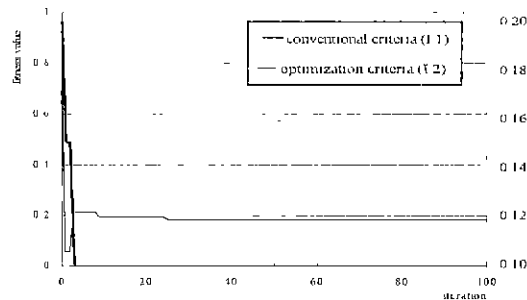


Fig 3(b) Conventional criteria and optimization criteria

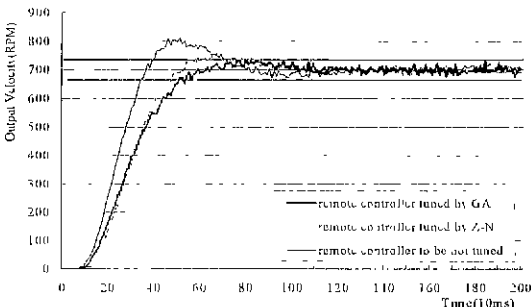


Fig 4 Comparison of responses of remote controllers tuned by Ziegler-Nichols and GA

그림 3(a)는 100세대가 진화되는 경우의 적합도를 나타내며, 그림 3(b)는 고전적인 설계기준과 최적제어적 설계기준의 적합도를 나타낸다. 그림에서 적합도는 초기에 대략 0.75822에서 점차적으로 향상되어 53세대가 지난 후에는 0.97643으로 수렴하였다. 특히 오버슈트와 정착시간에 대한 적합도는 3세대에서 이미 만족되고, 그 이후의 적합도 향상은 ISE의 최소화에 의하여 이루어짐을 확인할 수 있었다. 100세대가 지난 후의 α , β , γ 은 1.2098, 0.6886, 0.6160으로 나타났으며, 이를 이용하여 원격 제어기의 게인을 계산하면 $K_p=0.0133$, $K_i=0.1377$, $K_d=0.0000616$ 으로 나타났다.

그림 4에는 Ziegler-Nichols 계수조정법에 의하여 튜닝된 PID 제어기($K_p=0.008832$, $K_i=0.147$, $K_d=0.000132$)에 의한 응답과 100세대 진화를 통하여 조절된 원격 제어기에 의한 응답을 비교하였다. 그림에서 Ziegler-Nichols 계수조정법에 의하여 튜닝된 PID 제어기를 이용한 결과 최대오버슈트는 8%로, 정착시간은 0.85초로 나타났다. 반면, GA에 의하여 조절된 원격 제어기를 이용한 결과, 최대오버슈트는 4.57%로, 정착시간은

0.54초로 나타났다. 이러한 결과로 볼 때, GA에 의한 방법이 Ziegler-Nichols 방법에 비해 성능이 우수함을 알 수 있었다. 또한, Ziegler-Nichols 방법은 네트워크 지연을 직접 측정하여 그 크기를 알 필요가 있지만, GA에 의한 방법은 네트워크 지연에 대한 정보를 알 필요가 없어도 효과적인 튜닝이 이루어짐을 알 수 있다. 특히 네트워크 지연은 시스템의 구현 방식이나 네트워크의 통신 부하, 접속되는 스테이션의 수 등에 따라 변하므로, 시스템의 성능 평가를 충분히 수행하기 전까지는 정확한 값을 알기는 매우 어렵다. 따라서, 불확실한 네트워크 지연을 가지는 네트워크 기반 제어 시스템에서 제어기 설계 방법으로는 GA에 의한 제어기 설계 방법이 더 우수함을 알 수 있다.

6. 결론

본 논문에서는 Profibus-DP를 이용한 네트워크 기반 제어 시스템의 구축 방법을 제안하고, 네트워크 지연의 발생 원인을 규명하였다. 또한 GA에 의한 네트워크 기반 제어기의 설계 방법에 대하여 제안하였다. 이러한 결과로서 다음과 같은 결론을 얻을 수 있었다.

첫째, 네트워크 기반 제어 시스템에서 네트워크 지연이 발생함을 관찰하였으며, 이러한 지연이 제어 시스템의 전체적인 성능을 저하시킨다는 점을 실험적으로 확인하였다. 따라서, 필드버스를 이용한 네트워크 기반 제어 시스템의 설계에 있어서 네트워크 지연을 고려한 제어기 설계가 이루어져야 함을 확인하였다.

둘째, Profibus-DP에서의 네트워크 지연은 프로토콜의 특성과 각 프로세스들간의 비동기화로 인하여 발생함을 확인하였다. 특히, 응용 프로세스에서의 비동기화는 제어기와 플랜트의 응용 프로세스 설계 방식에 따라 증가될 수 있으므로, 네트워크 지연을 최소화시킬 수 있는 방향으로 설계되어야 한다.

셋째, 유전자 알고리즘을 이용하여 PID 제어기를 기초로 한 원격 제어기를 설계하는 방법은 네트워크 지연에 대한 직접적인 정보가 없는 상황에서도 효과적임을 확인하였다.

참고문헌

- [1] A. Ray, "Networking for Computer-Integrated Manufacturing," IEEE Network, vol. 2, no. 3, pp. 40-47, 1988.
- [2] Profibus Application Program Interface, user manual, version 5.2, rev. 01, 1998.
- [3] PNO, Profile for Variable Speed Drives, PROFIDRIVE, version 2.0, Profibus International, 1997.
- [4] G. C. Walsh, and Y. Hong, "Scheduling of Networked Control Systems," IEEE Control Systems Magazine, Vol. 21, No. 1, pp. 57-65, 2001.
- [5] Z. Wei, M. S. Branicky, and S. M. Phillips, "Stability of Networked Control Systems," IEEE Control Systems Magazine, Vol. 21, No. 1, pp. 84-99, 2001.
- [6] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, 1989.
- [7] 이상호, 김인준, 이경창, 이석, "공장 자동화용 네트워크를 위한 통합 성능관리기의 개발, 제어 자동화·시스템공학회 논문지, 제5권 제5호, pp. 600-613, 1999년.
- [8] K. Ogata, Modern Control Engineering, Prentice-Hall, pp. 257-270, 1990.
- [9] J. G. Ziegler, and N. B. Nichols, "Optimum Settings for Automatic Controllers," Transactions on ASME, vol. 64, 1942.