

퍼지 레귤레이터를 이용한 서보 모터 속도제어

Speed control of servo motor using Fuzzy regulator

윤 양 응*
오 훈**
이 회 승***

ABSTRACT

In this paper, PI controller with fuzzy regulator for D.C servo motor speed control in variable load are presented.

Fuzzy regulator is made up of logic for fuzzification, inference, and defuzzification.

The rule of fuzzy regulator and, scale mapping for the fuzzification of regulator input and the defuzzification of real regulator output are decide by experiment.

The function of PI controller with fuzzy regulator is compared PI(proportional integral) controller.

The parameters of PI controller are decided by the unit step-response method.

The robust and accurate regulation function of PI controller with fuzzy regulator is demonstrated by computer simulation.

I. 서 론

PI 제어기는 구조가 간단하고 과도응답특성이 좋으며 정상상태 오차를 제거할 수 있어 많이 사용되고 있다. 그러나 PI 제어기는 파라미터 변동이나 외란에 상당히 민감하게 반응하는 단점을 가지고 있다. 즉 PI 제어기의 비례이득을 너무 크게 설정할 경우 조그마한 외란에 대해서도 시스

템은 큰 오버슈트나 진동이 발생하여 불안정하게 된다. 이와같은 문제점에 대하여 퍼지이론을 도입 해결하고자 한다.

퍼지 이론은 현상의 불확실한 상태를 그대로 표현해 주는 방법으로서 1965년 미국 버클리 대학의 L.A. Zadeh 교수에 의해서 처음 소개 되었고, 애매하게 표현된 자료를 우리에게 유용한 자료로 만들기 위하여 퍼지 집합(fuzzy set), 퍼지 논리(fuzzy logic), 퍼지수(fuzzy number), 퍼지 관

* 원광대학교 공과대학 전기공학과 교수

** 원광대학교 공과대학 전기공학과 조교

*** 이리 농공전문대학 전기과 시간강사

계(fuzzy relations)등의 개념을 포함하고 있으며 수학적인 계산 방법도 잘 개발되어 있다.^{1,2,3)}

1972년 Zadeh에 의해 퍼지 제어의 가능성이 제시 되었으며, 1974년 Mamdani는 퍼지제어를 모형 증기기관의 제어에 직접 응용하였다. 1980년 덴마크의 Smidth사가 시멘트 킬른에 대한 퍼지 제어 시스템을 실용화한 이후 지하철의 자동제어 운전, 자동차의 속도제어, 퍼지 세탁기, 정수장 제어 등 여러 방면에서 실용화 되고 있다.

임의의 시스템 제어에 퍼지 이론을 적용시키고자 하는 가장 큰 이유는 시스템이 비선형이고 복잡한 시스템이라 할지라도 수학적 해석이 필요 없으며 제어기 설계시 인간의 사고를 모방한 알고리즘을 사용함으로써 외부 변화에 따라 종합적 판단으로 해결치 못하던 기존의 문제점을 개선할 수 있다는 장점 때문이다.^{4,5,6,7)}

본 논문에서는 직류 서보 모터의 속도제어를 위한 퍼지 레귤레이터(fuzzy logic regulator; FLR)를 갖는 PI제어 시스템을 제안하였다.

II장에서는 직류 서보 모터의 속도제어에 적합한 제어 시스템을 구성하고 퍼지 레귤레이터를 설계하였다. FLR의 설계는 레귤레이터 입력변수의 값을 측정하고 입력변수값의 영역(range)을 이에 상응하는 전체집합(universe of discourse)으로 크기변환(scale mapping) 시켜서 이 입력값을 퍼지화(fuzzification)시키는 단계와 퍼지화된 각각의 입력변수들을 퍼지 관계와 퍼지 논리의 추론 규칙을 도입하여 퍼지 레귤레이터 입력을 구해주는 추론단계, 그리고 추론 후 계산된 퍼지값은 실제 레귤레이터 출력으로 사용할 수 없으므로 이를 명확한 값으로 변환시키는 비퍼지화(defuzzification) 단계의 3단계로 설계하였다.

III장에서는 시뮬레이션 및 결과 고찰을 통하여 PI 제환 시스템과 퍼지 레귤레이터를 갖는 PI제환 제어 시스템을 비교하였다.

II. 퍼지 레귤레이터

2-1. 제어 시스템의 구성

제어 시스템은 그림 2.1과 같은 형태로 구성된다.

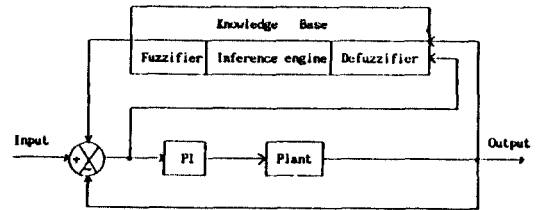


그림 2.1 퍼지 레귤레이터를 갖는 시스템 구성도
Fig. 2.1 System block diagram with fuzzy regulator

PI제어기는 구조가 간단하고 과도응답 특성이 좋으며 정상상태 오차를 제거할 수 있어 많이 사용되고 있다. 그러나 PI제어기는 파라미터 변동이나 외란에 상당히 민감하게 반응하는 단점을 가지고 있다. 이와같은 문제점을 해결하기 위하여 퍼지 레귤레이터를 갖는 PI 제환 제어 시스템을 제안하였다.

PI 매개변수는 Ziegler-Nichols가 제안한 단위 계단 응답법 (The unit step-responce method)에 의해 결정하였다.⁸⁾

퍼지 레귤레이터는 퍼지화, 추론, 비퍼지화로 구성된다.

퍼지화 도입부는 입력 데이터를 적절한 언어값으로 변환시킨다. 추론부는 퍼지 관계와 퍼지 논리의 추론 규칙을 도입하여 퍼지 레귤레이터 출력을 구해주는 기능을 한다. 비퍼지화는 실제 레귤레이터 출력의 전체집합에 맞추어 크기를 변환시킨다.

2-2. 퍼지 레귤레이터의 규칙

1) 퍼지화(fuzzification)

본 논문에서는 FLR의 두 입력 변수로 오차(E)와 오차 변화율(ΔE)을 각각 사용하였다.

소속함수의 구성시 오차(E)와 오차 변화율(ΔE)의 크기 정도에 따라 소속함수의 수를 작게 할 경우 FLR의 가장 큰 문제점중의 하나인 시간이 많이 걸리는 단점을 보완할 수 있으나 정밀도와 성능이 떨어지기 때문에 정밀하고 미세한 제어에는 적합하지 않다.

또한 소속함수의 수를 증가시켜 미세하게 구분할 경우 정밀도는 우수하나 알고리즘이 너무 복잡하고 계산 시간도 많이 걸린다.

이러한 소속 함수의 수 결정은 레귤레이터의 설계자가 시스템의 특성을 충분히 고려하여 선택하여야 한다. 시간과 정밀도를 고려하여 두 입력 변수의 소속 함수를 각각 5구간으로 하였으며 소속함수는 가장 일반적이고 계산이 간편한 삼각형 형태로 하였다. 두 입력변수인 오차와 오차 변화율의 소속함수를 선형적으로 배치하였을 경우 설정속도에 늦게 도달하는 문제점이 있어 소속함수를 비선형적 분포로 구성하여 그림 2.2, 그림 2.3과 같이 정하였다.

그림 2.2와 그림 2.3에서 보는 바와같이 오차와 오차 변화율 각각을 다음과 같은 5개의 퍼지 집합으로 구성하였다.

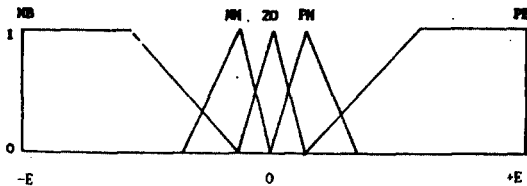


그림 2.2 오차 소속함수의 비선형적 분포
Fig. 2.2 Nonlinear distribution of error membership function

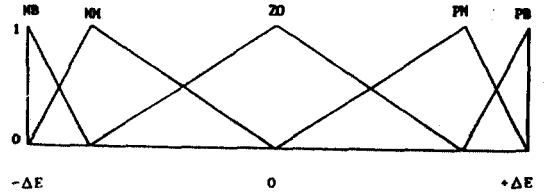


그림 2.3 오차 변화율 소속함수의 비선형적 분포
Fig. 2.3 Nonlinear distribution of error change ratio membership function

오차의 소속함수	오차 변화율의 소속함수
NB = {E, m _{NB} (E)}	NB = {ΔE, m _{NB} (ΔE)}
NM = {E, m _{NM} (E)}	NM = {ΔE, m _{NM} (ΔE)}
ZO = {E, m _{ZO} (E)}	ZO = {ΔE, m _{ZO} (ΔE)}
PM = {E, m _{PM} (E)}	PM = {ΔE, m _{PM} (ΔE)}
PB = {E, m _{PB} (E)}	PB = {ΔE, m _{PB} (ΔE)}

또한 출력 소속함수는 그림 2.4와 같이 선형적 분포로 배치하였다.

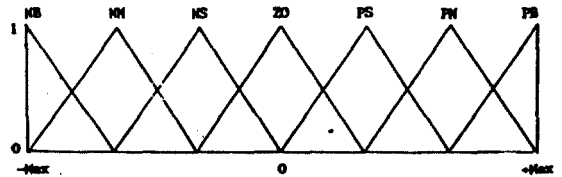


그림 2.4 출력 소속함수의 선형적 분포
Fig. 2.4 Linear distribution of output membership function

2) 추론(Inference)

퍼지화를 통해서 현 상태의 오차값과 오차 변화율 값이 위치한 소속함수와 그 집합내에서의 소속치를 검출하면 이 두 값을 추론하여 타당한 퍼지값을 얻어야 한다.

본 논문에서는 추론시 전술한 퍼지관계의 합성중 가장 대표적 방법인 Max-Min 방법을 사용하였다.

추론한 값이 출력 소속함수로 대응될때 다음과

대한 공정의 응답을 측정하고 이 단위 계단 응답 곡선의 상승하는 기울기와 지연시간을 토대로 PI제어기의 매개변수를 결정하는 방법이다.

개루프 시스템의 단위 계단 응답 곡선에서 상승속도를 R, 지연시간을 L이라 하면 PI제어기의 매개변수는 다음과 같다.

계수	Kp	Ti
제어기		
PI	0.9/R	3L

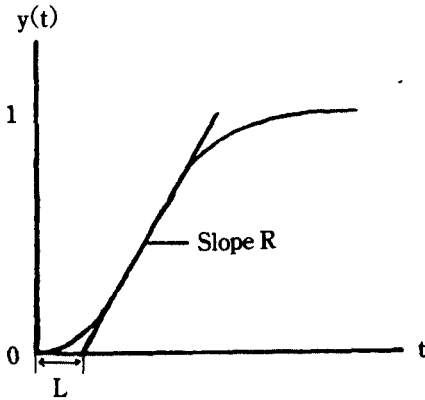


그림 3.1 개루프 시스템의 단위 계단 응답곡선
Fig. 3.1 The unit step-response curve of open loop system

퍼지 레귤레이터 설계시 모터의 설정속도를 2000[rpm]으로 정하였으므로 설정속도에서의 오차는 0이 된다. 그리고 모터의 무부하시 최대속도는 4000[rpm]이므로 이때의 설정속도와의 오차는 -2000[rpm]이 되고, 처음 기동시에는 설정속도와의 차이가 +2000[rpm]이 된다. 그러므로 모터의 최대 오차범위는 -2000[rpm]에서 +2000[rpm]으로 설정하고 이는 오차 소속함수의 범위를 의미한다.

샘플링 주기는 0.01초로 하였으며 이때 오차 변화율의 허용범위는 -200[rpm]에서 +200

[rpm]이 된다. 이 범위는 오차 변화율 소속함수를 의미한다. 출력 소속함수의 범위는 -1500[rpm]에서 +1500[rpm]이고, 플랜트의 허용오차 범위는 1(%)이다.

샘플링 주기때 검출된 오차와 오차 변화율이 다음 그림 3.2와 같다고 하자.

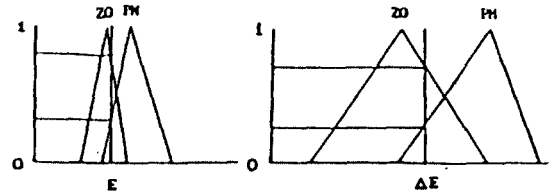


그림 3.2 검출된 오차와 오차 변화율
Fig. 3.2 Detected error and error change ratio

그림 3.2와 같을때 추론규칙은 다음과 같다.

IF E is ZO and ΔE is ZO then ΔU=ZO
or IF E is ZO and ΔE is PM then ΔU=PS
or IF E is PM and ΔE is ZO then ΔU=PS
or IF E is PM and ΔE is PM then ΔU=PM

그림 3.3은 본 논문에서 사용한 Max-Min 추론 방법이다.

그림 3.3에서 Max-Min 추론 방법에 의해 구해진 값은 퍼지값이므로 이 값을 비퍼지화해야 한다. 식(2.1)에 의한 최대값 방법으로 비퍼지화 시키면 이 값이 레귤레이터의 출력이 된다.

그림 3.4에서 (a), (b)는 기준부하시 PI 제어기와 퍼지 레귤레이터를 갖는 PI제환 제어기의 출력곡선이고 (c), (d)는 부하 2배 증가시 PI 제어기와 퍼지 레귤레이터를 갖는 PI 제환 제어기의 출력곡선이며 (e), (f)는 부하 3배 증가시 PI 제어기와 퍼지 레귤레이터를 갖는 PI 제환 제어기의 출력곡선이다.

그림 3.5에서 (a)는 기준부하로 운전중 부하를

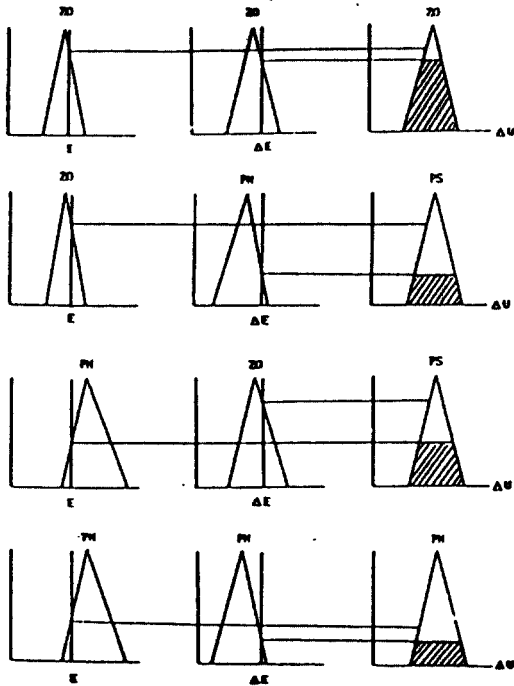
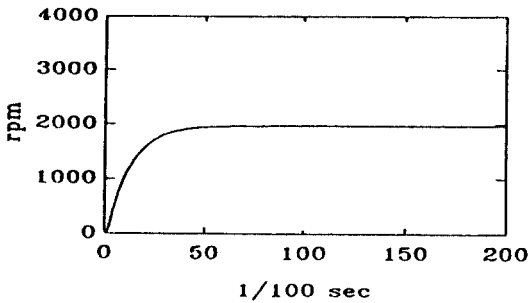
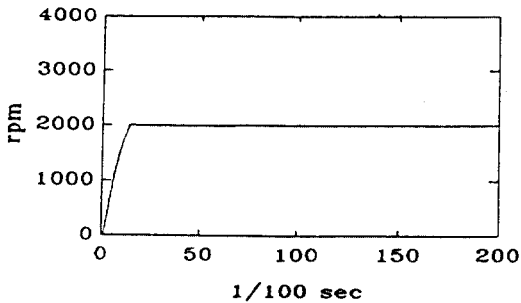


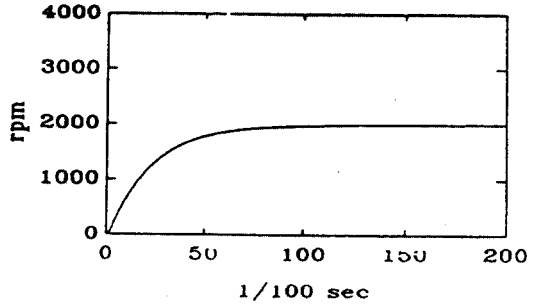
그림 3.3 Max-Min 추론의 예
Fig. 3.3 Example of Max-Min inference



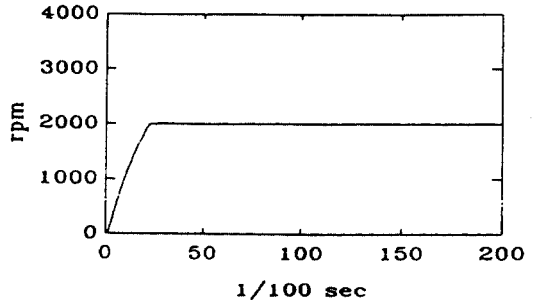
(a) 기준 부하시 PI 제어기



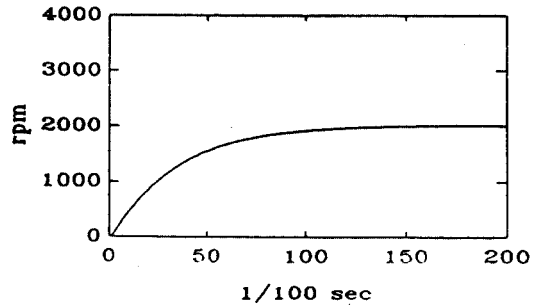
(b) 기준 부하시 퍼지 레귤레이터를 갖는 PI 변환 제어기



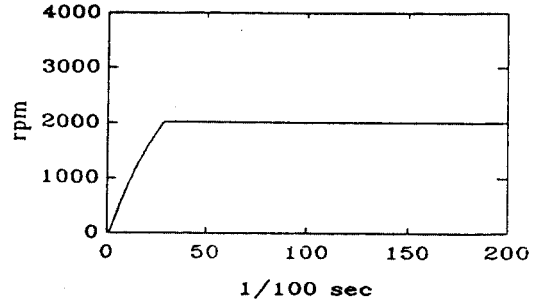
(c) 부하 2배 증가시 PI 제어기



(d) 부하 2배 증가시 퍼지 레귤레이터를 갖는 PI 변환 제어기



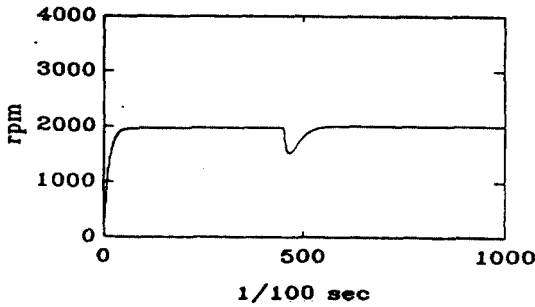
(e) 부하 3배 증가시 PI 제어기



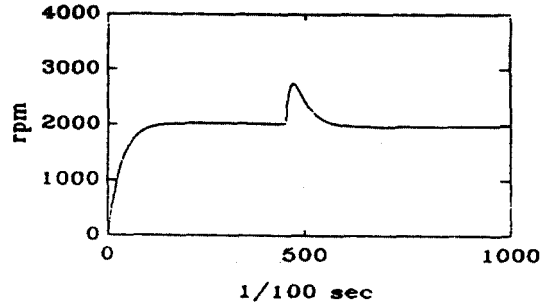
(f) 부하 3배 증가시 퍼지 레귤레이터를 갖는 PI 변환 제어기

그림 3.4 PI 제어기와 퍼지 레귤레이터에 의한 출력곡선

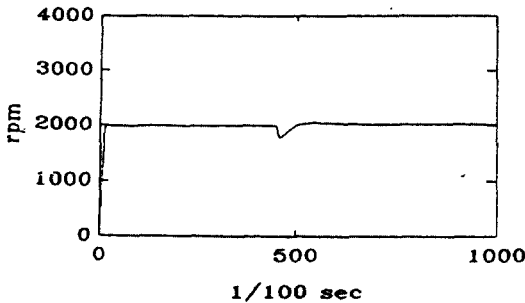
Fig. 3.4 Output curve by PI cotroller and fuzzy regulator



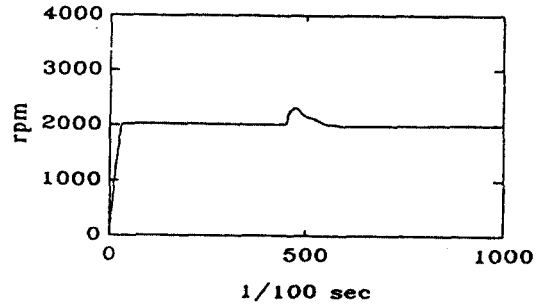
(a) 기준 부하로 운전중 부하를 2배로 증가시켰을 때 PI 제어기



(c) 부하를 3배로 증가하여 운전중 기준부하로 감소시켰을 때 PI 제어기



(b) 기준 부하로 운전중 부하를 2배로 증가시켰을 때 퍼지 레귤레이터를 갖는 PI 궤환 제어기



(d) 부하를 3배로 증가하여 운전중 기준부하로 감소시켰을 때 퍼지 레귤레이터를 갖는 PI 궤환 제어기

2배로 증가시켰을 때 PI 제어기에 의한 출력곡선이고, (b)는 기준부하로 운전중 부하를 2배로 증가시켰을 때 퍼지 레귤레이터를 갖는 PI 궤환 제어기의 출력곡선이다. (c)는 부하를 3배로 증가하여 운전중 기준부하로 감소시켰을 때 PI 제어기에 의한 출력곡선이고, (d)는 부하를 3배로 증가하여 운전중 기준부하로 감소시켰을 때 퍼지 레귤레이터를 갖는 PI 궤환 제어기의 출력곡선이다.

그림 3.4에서 알 수 있듯이 PI제어기는 설정속도에 도달하는 시간이 기준부하시 약1초, 부하 2배 증가시 약 1.3초, 부하 3배 증가시 약 1.5초 정도 걸리는 반면 퍼지 레귤레이터를 갖는 PI 궤환 제어기는 설정속도에 도달하는 시간이 기준부하

시 약0.2초, 부하 2배 증가시 약 0.25초, 부하 3배 증가시 약0.3초 정도면 도달하는 것을 볼 수 있다.

그림 3.5에서 알 수 있듯이 가변 부하시에도 이탈속도가 PI 제어기 보다 퍼지 레귤레이터를 갖는 PI 궤환 제어기가 현저히 적음을 알 수 있다.

IV. 결 론

본 논문에서는 직류 서보 모터의 속도제어를 위한 퍼지 레귤레이터를 갖는 PI 궤환 제어기를 제안하였다.

퍼지 레귤레이터 설계시, 두 입력변수인 오차와 오차 변화율의 소속함수를 선형적으로 배치하였

을 경우 설정속도에 너무 늦게 도달하는 문제점이 있어 소속함수를 비선형적 분포로 구성하여 제어성능을 개선하였다.

비퍼지화 단계에서는 최대값 방법을 도입하여 전체 알고리즘을 단순화하였다.

시뮬레이션을 통해 PI 제어기와 퍼지 레귤레이터를 갖는 PI 궤환 제어기를 비교해 본 결과 운전자가 원하는 설정속도에 PI제어기 보다 퍼지 제어기가 빠르게 도달하는 것을 볼 수 있다.

가변 부하시에도 이탈속도가 PI 제어기보다 퍼지 레귤레이터를 갖는 PI 궤환 제어기가 현저하게 적음을 알 수 있다.

이와같이 퍼지 레귤레이터를 갖는 PI 궤환 제어기는 빠른 응답과 가변 부하시에도 크게 영향을 받지 않고 정상 운전을 하는 제어 특성을 보인다.

본 연구는 플랜트의 특성이 수시로 변화되는 시스템의 제어기 설계에 기여할 것으로 사료된다.

참 고 문 헌

1. Dubois, D. and Prade, H., "Fuzzy Sets and Systems : Theory and Applications", Academic

press. 1980.

2. 박민용, 최항식, "퍼지 제어 시스템", 대영사, 1990.
3. 박민용, 최항식, "퍼지 시스템의 응용입문", 대영사, 1990.
4. Kickart, W.J.M. and MAMDANI, E. H., "Analysis of Fuzzy Logic Controller", Fuzzy Sets and System, Vol. 1, No. 1, pp.29-44, 1978.
5. King, P.J. and Mamdani, E. H., "The Application of Fuzzy Control Systems to Industrial Processes", Automatica, Vol. 13, pp.235-242. 1977.
6. Chuen Chien Lee, "Fuzzy Logic in Control Systems : Fuzzy Logic Controller - Part I, II", IEEE trans. System, Man and Cybernetics, Vol. 20, No.2, pp.404-435, 1990.
7. Kandel, A., "Fuzzy Mathematical Techniques with Application", Addison Wesley Publishing Company., 1986.
8. K. J. Åström and B. Wittenmakr, "Computer-Controlled System Theory and Design", Prentice-Hall International Edition, 1990.