

유전알고리즘을 이용한 직류 서보 모터 속도제어용 PI제어기의 설계

박한석\* 박현주\* 김동완\*\* 황기현<sup>o</sup> 우정인<sup>oo</sup>  
 \*부경대 전기공학과 \*\*동명대학 전기전자계열 <sup>o</sup>부산대 컴퓨터 및 정보통신연구소 <sup>oo</sup>동아대 전기공학과

Design of PI Controller for DC ServoMotor Speed Control Using Genetic Algorithm

Han-Suk Park\* · Hyun-Ju Park\* · Dong-Wan Kim\*\* · Gi-Hyun Hwang<sup>o</sup> · Jung-In Woo<sup>oo</sup>  
 \*Pukyong National Univ. \*\*TongMyong College <sup>o</sup>Pusan National Univ. <sup>oo</sup>Dong-A Univ.

**Abstract** - This paper proposes the design of PI controller using real-coding genetic algorithm showing a good performance on convergence velocity and diversity of population among evolutionary computations. To evaluate the proposed method's effectiveness, we apply the proposed GA-PI controller to the speed control of an actual DC servomotor system.

The experimental results show that GA-PI controller has the better control performance than PI controller in terms of settling time, rising time and overshoot.

1. 서 론

과학기술의 발달과 더불어 산업 자동화분야에서는 산업용 플랜트의 복잡성에 따른 더욱 강한 제어알고리즘을 요구하고 있다. 기존의 산업체에서 널리 사용되고 있는 PID(Proportional-Integral-Derivative) 제어기는 수학적으로 모델링된 제어 시스템을 이용하여 PID 계수를 시행착오방법으로 튜닝하였다. 그러나 수학적으로 모델링된 시스템으로 설계된 PID제어기는 외란, 부하변화 및 내·외부의 환경변화가 발생한 경우에는 최적의 제어 성능을 기대하기가 어렵다[1,2]. 기존의 제어기는 제어 시스템의 수학적인 모델링에 기반을 두고 이루어져 왔기 때문에 제어 시스템의 동특성에 대한 정확한 지식이 필요하였다. 그러나 일반적으로 제어 시스템은 비선형이고 시변인 경우에는 그 시스템을 정확하게 모델링을 하기가 어렵다. 최근에는 이러한 문제점을 해결하기 위해서 확률적인 최적화 방법인 유전알고리즘 및 지능기법을 이용하여 제어기를 설계하고 있다[3-6].

본 논문에서는 진화연산 중에서 해의 다양성과 수렴속도면에서 좋은 성능을 나타내는 실수형 유전알고리즘을 이용하여 PI제어기를 설계하였다. 제안한 실수형 유전알고리즘에 의한 PI제어기의 유용성을 평가하기 위하여 컴퓨터 시뮬레이션 및 직류 서보모터의 실험장치를 제작하여 실험을 수행하고, 시행착오방법으로 설계된 PI제어기와 성능을 비교·검토한 결과 제안한 실수형 유전알고리즘을 이용한 PI제어기가 기존의 PI제어기보다 과도특성 및 강인성면에서 더 우수한 특성을 가짐을 알 수 있었다.

2. 직류 서보모터

본 논문에서 사용한 전동기의 속도제어기에 대한 블록선도는 그림 1에 나타내었다. 그림 1에서  $G(s)$ 는 PI제어기에 대한 전달함수이고, 전류제어기 부분은 스윙링 소자에 대한 시지연으로 모델링하였고,  $T_a$ 는 전류제어기의 시정수이다. 그림 1의 전동기 속도제어기에 대한 블록선도는 식 (1), (2)와 같이 표현할 수 있다.

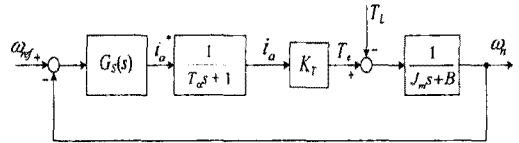


그림 1 전동기의 PI제어기에 대한 블록선도

$$\frac{di_a(t)}{dt} = \frac{1}{T_a} (i_a(t)^* - i_a(t)) \quad (1)$$

$$\frac{d\omega_n(t)}{dt} = \frac{1}{J_m} (K_T i_a(t) - B\omega_n(t) - T_L(t)) \quad (2)$$

여기서,  $\omega_n(t)$  : 회전자 각속도 [rad/s]

$i_a(t)$  : 전기자 전류 [A]

$J_m$  : 회전자 관성 [Nm<sup>2</sup>]

$T_L(t)$  : 부하 토크 [Nm]

$B$  : 댐핑 계수 [Nms]

3. 실수형 유전알고리즘을 이용한 PI제어기의 설계

일반적으로 PI제어기는 외란 및 제어시스템의 파라메타 변동에 대해서 뛰어난 제어 성능을 나타내지만, 다양한 외란 및 제어시스템의 파라메타 변동에 대해서 최적의 제어 성능을 얻을 수 없다. 따라서 본 논문에서는 다양한 외란 및 제어시스템의 파라메타 변동에 대해서 최적의 제어 성능을 얻기 위해서 PI제어기의 이득을 실시간으로 유전알고리즘을 이용하여 반복학습 방법으로 튜닝하였다. 그림 2는 실수형 유전알고리즘을 이용하여 PI제어기의 이득을 튜닝하기 위한 전체 구성도를 나타내었다. 그림 2에서 보는 것처럼 PI제어기의 입력으로 오차와 오차의 변화분을 사용하였다.

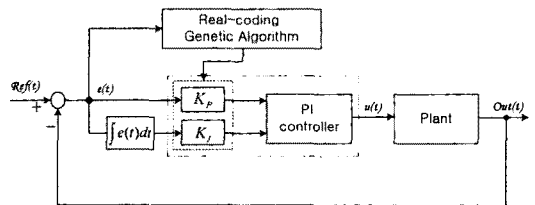
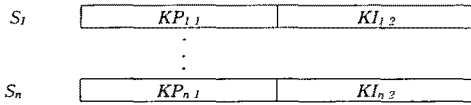


그림 2 RGA를 이용한 PI제어기 튜닝을 위한 구성도

실수형 유전알고리즘을 이용하여 PI제어기의 이득을 튜닝하기 위한 해집단의 구조는 그림 3에 나타내었다. 그림 3에서 보는 것처럼 실수형 유전알고리즘을 이용하여 PI제어기의 이득을 튜닝하는데 필요한 파라메타 수는 2개이다. 그리고 해집단의 각 스트링을 평가하기 위하여 사용된 적합도 함수는 식 (3)과 같이 오차의 절대치 합을 사용하였다.

$$\text{적합도} = \frac{1}{1 + \int_{t=0}^T |Ref(t) - Out(t)|} \quad (3)$$

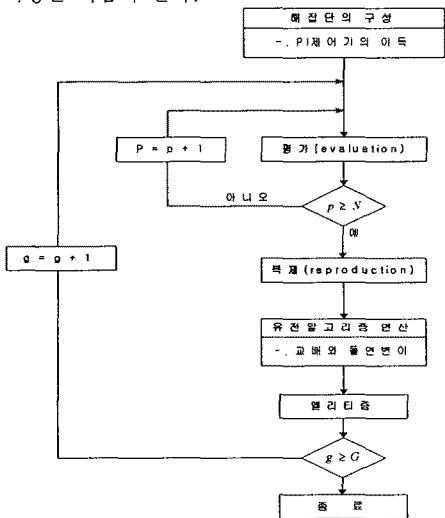
여기서,  $Ref(t)$  : 기준 입력  
 $Out(t)$  : 제어대상의 출력  
 $T$  : 특정한 시간 동안 획득한 데이터의 개수



여기서,  $KP_{ij}$  : 비례제어기의 이득  
 $KI_{ij}$  : 적분제어기의 이득  
 $n$  : 스트링 개수

그림 3 해집단의 구조

실수형 유전알고리즘을 이용하여 PI제어기를 설계하기 위한 전체적인 흐름도는 그림 4에 나타내었고, 설계 과정은 다음과 같다.



여기서,  $N$  : 해집단의 크기,  $G$  : 특정한 세대수

그림 4 RGA 이용한 PI제어기의 설계를 위한 흐름도

단계1)해집단 구성: 그림 3과 같이 PI제어기의 이득을 이용하여 해집단을 구성한다.

단계2)평가: 식 (3)의 적합도 함수를 이용하여 단계 1)에서 생성된 각 스트링을 평가하기 위한 과정은 다음과 같다.

- ①PI제어기의 입력인 오차 ( $e(t)$ )와 오차의 합 ( $\int e(t)dt$ )을 계산한다.
- ②아래의 식을 이용하여 PI제어기의 출력 ( $u(t)$ )을 계산한다.

$$u(t) = K_P \cdot e(t) + K_I \cdot \int e(t) dt$$

- ③미분방정식의 해법으로 제어대상 시스템을 푼다.
- ④시뮬레이션 시간 동안 ① ~ ③을 반복한다.

단계3)복제: 적합도에 비례하여 복제하는 무울렛법 방법을 사용하였다.

단계4)교배 및 돌연변이: 실수형 유전알고리즘의 연산자인 교배와 돌연변이를 수행하여 자손을 생성한다.

단계5)엘리트즘: 적합도가 가장 높은 개체를 실수형 유전알고리즘의 해집단에 하나씩 복제하는 엘리트즘을 사용하였다.

단계6)수렴조건: 특정한 세대수를 만족할 때까지 단계 2)~단계 6)을 반복한다.

#### 4. 실험결과

RGA를 이용한 PI제어기의 유용성을 검증하기 위해 직류 서보모터의 속도제어를 위한 실험장치를 구성하였다. 직류 서보모터의 속도제어를 위한 실제 구성은 그림 5에 나타내었다. 전체 속도 제어 시스템은 전류루프와 속도루프로 구성되어 있으며, 전류루프의 샘플링 시간은 250[μsec]이고, 속도루프의 샘플링 시간은 4[msec]이다. 실험에 사용된 전동기와 부하로 사용한 발전기의 사양은 표 1과 같다. 직류 서보모터의 실험장치는 그림 6에 나타내었다.

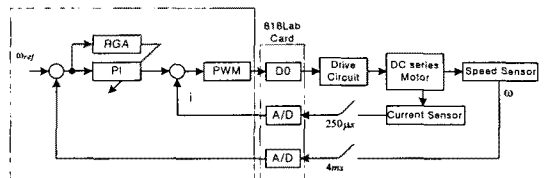


그림 5 직류 서보모터의 속도제어를 위한 실험장치

표 1 전동기 및 발전기의 사양

사양 \ 분류	전동기	발전기
정격 전압	75V	90V
정격 회전수	3000 rpm	3000rpm
정격 출력	200W	150W
정격 전류	3.3A	2A

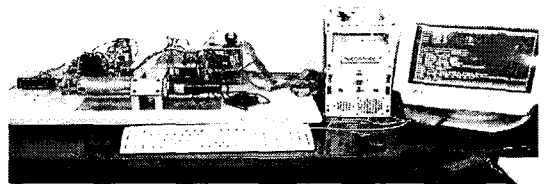


그림 6 직류 서보모터의 실험장치

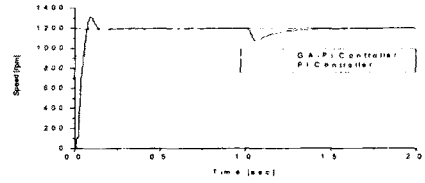
그림 7은 실수형 유전알고리즘을 이용하여 PI제어기의 설계에 사용된 기준속도에 대한 직류 서보모터의 실험결과를 나타내었다. 그림 7에서 보는 것처럼 RGA를 이용한 PI제어기가 기존의 PI제어기보다는 상승시간과 정정시간면에서 더 좋은 제어성능을 보임을 알 수 있다. 제안한 PI제어기의 강인성을 평가하기 위하여 PI제어기의 설계에 사용되지 않은 기준속도 및 부하변화에 대해서 기존의 PI제어기와 제어 성능을 비교하였다. 그림 8은 기준속도가 400[rpm], 800[rpm]일 때 속도응답 특성을 나타내었다. 그림 8에서 보는 것처럼 RGA를 이용한 PI제어기가 정정시간 및 상승시간면에서 기존의 PI제어기보다 더 우수한 제어 성능을 나타냄을 알 수 있다. 그림 9는 속도가 800[rpm], 1200[rpm]이고, 1초 후에 부하가 변화했을 때 속도응답 특성을 나타내었다. 그림 9에서 RGA를 이용한 PI제어기가 기존의 PI제어기보다 부하변화에 대해서는 유사한 제어 성능을 보였다.

이상의 결과를 요약하면 표 2와 같고, 표 2에서 경우 1은 일정부하에 대해서 기준속도가 400[rpm]에서 1200[rpm]까지 변화했을 때 정정시간과 오버슈트를

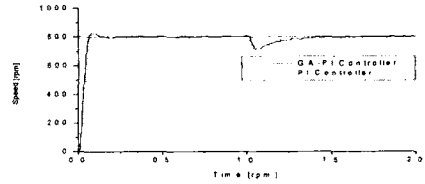
비교하였고, 경우2는 1초 후에 부하가 증가했을 때 기준속도에 대한 정정시간과 오버슈트를 비교하였다. 표 2에서 보는 것처럼 경우1에 대해서는 RGA를 이용한 PI제어기가 기존의 PI제어기보다 정정시간면에서는 0.05(sec) ~ 0.14(sec) 정도 더 빨리 정상상태에 도달하였고, 최대오버슈트면에서는 4(%) ~ 8(%) 정도 더 적었다. 경우 2에 대해서는 RGA를 이용한 PI제어기와 기존의 PI제어기가 정정시간과 오버슈트에 대해서 유사한 결과를 나타내었다.

표 2 PI제어기와 제안한 PI제어기의 제어 성능 비교

기준속도	방법	정정시간(sec)		오버슈트(%)	
		GA-PI제어기	PI제어기	GA-PI제어기	PI제어기
경우1	400(rpm)	0.15	0.20	13.75	17.75
	800(rpm)	0.14	0.28	3.88	11.88
	1200(rpm)	0.12	0.26	1.67	8.58
경우2	400(rpm)	0.31	0.24	11.00	13.00
	800(rpm)	0.37	0.34	10.38	13.38
	1200(rpm)	0.42	0.34	10.33	12.17

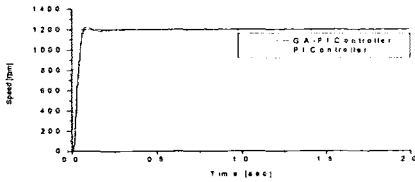


(a) 속도응답(1200(rpm))

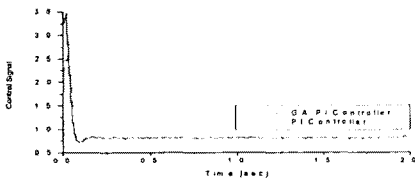


(b) 속도응답(800(rpm))

그림 9 부하가 변화했을 때의 속도응답

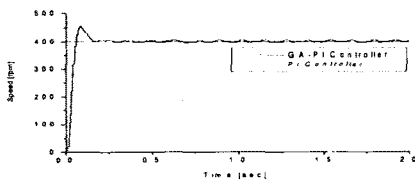


(a) 속도응답(1200(rpm))

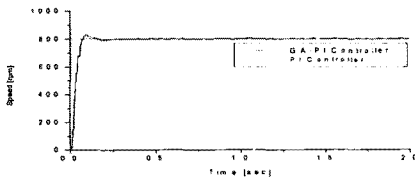


(b) 제어입력

그림 7 직류 서보모터의 속도응답



(a) 속도응답(400(rpm))



(b) 속도응답(800(rpm))

그림 8 직류 서보모터의 속도응답

## 5. 결론

본 논문에서는 진화연산 중에서 해의 다양성과 수렴속도면에서 좋은 성능을 나타내는 실수형 유전알고리즘을 이용하여 PI제어기를 설계하였다. RGA를 이용한 PI제어기의 유용성을 평가하기 위하여 컴퓨터 시뮬레이션 및 직류 서보모터의 실험장치를 제작하여 실험을 수행하였으며, 본 논문에서 제안한 RGA를 이용한 PI제어기의 성능을 비교하기 위해 시행착오방법으로 설계된 PI제어기와 비교·검토하였다.

직류 서보모터의 실험결과, 일정부하에 대해서 기준속도가 400[rpm]에서 1200[rpm]까지 변화했을 때 제안한 RGA를 이용한 PI제어기가 기존의 PI제어기보다 정정시간면에서는 0.05(sec) ~ 0.14(sec) 정도 더 빨리 정상상태에 도달하였고, 최대오버슈트면에서는 4(%) ~ 8(%) 정도 더 적었다. 1초 후에 부하가 증가하였을 때 RGA를 이용한 PI제어기와 기존의 PI제어기는 정정시간과 오버슈트에 대해서 유사한 결과를 나타내었다.

## [참 고 문 헌]

- [1] F. Betin, D. Pinchon and G. A. Capolino, "A time-varying sliding surface for robust position control of a DC motor drive", *IEEE Trans. on Industrial Electronics*, Vol. 49, pp. 462-473, April, 2002
- [2] Wang Jang, Wang Jiajun, Zhao Zhongtang and Fei Xiangyang, "The application of optimal sliding mode control in DC motor", *Proceedings of the 3rd World Congress on Intelligent Control and Automation*, Vol. 4, pp. 3001-3004, 2000
- [3] A. Rubaai and R. Kotaru, "Online identification and control of a DC motor using learning adaptation of neural networks", *IEEE Transactions on Industry Applications*, Vol. 36, pp 935-942, May-June, 2000
- [4] Jun Oh Jang and Pyeong Gi Lee, "Neuro-fuzzy control for DC motor friction compensation", *Proceedings of the 39th IEEE Conference on Decision and Con*, Vol. 4, pp. 3550-3555, 2000
- [5] B. Cakir, A. B. Yildiz, N. Abut and N. Inanc, "DC motor control by using computer based fuzzy technique", *Applied Power Electronics Conference and Exposition, APEC '99, Fourteenth Annual*, Vol. 1, pp. 391-395, 1999
- [6] Mitsuo Gen and R. Cheng, *Genetic Algorithms & Engineering Design*, A Wiley-Interscience Publication, 1997