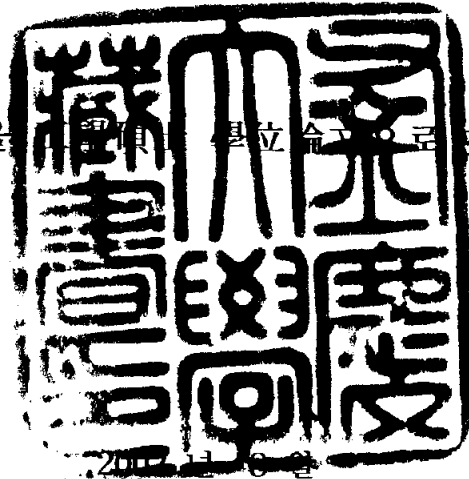


공 학 석 사 학 위 논 문

유전알고리즘을 이용한 직류 서보
모터 속도제어용 PI제어기 설계

지 도 교 수 박 한 석

이 論 文 을 韓 國 電 子 工 學 院 學 位 論 文 으 로 提 出 함



부경대학교 산업대학원

전기공학과

박 현 주

朴賢柱의 工學碩士 學位論文을 認准함

2002 年 6 月

主審 工學博士 洪 淳 一



委員 工學博士 金 仁 東



委員 工學博士 朴 瀚 錫



제 1 장 서 론

1. 1 연구의 배경

최근 산업의 발달과 더불어 여러 분야에서 인간의 작업을 기계가 대신하는 공장 자동화 설비에 관한 연구가 많이 이루어지고 있다. 이러한 자동화 설비의 바탕이 되는 것은 직류 서보 모터, 교류 서보 모터, BLDC 등의 전동기가 주류를 이루고 있다. 그 중에서 직류 서보 전동기는 산업용 로봇이나 수치제어용 공작기계, 사무 자동화 기기 등의 정밀제어에 널리 사용되고 있다. 이러한 직류 서보 전동기의 속도제어 방법은 1970년대 이후 하드웨어의 급속한 발달로 인하여 마이크로 프로세서의 가격이 싸지고 신뢰도가 높아짐에 따라 실시간 연산이 가능하게 되어 아날로그 제어기에서 시스템의 설계, 확장, 수정 등에 있어서 많은 융통성과 부가기능을 부여할 수 있는 디지털 제어기로 변화해가고 있다.

디지털 방식을 이용한 직류 서보 전동기의 속도제어기 중에서 산업체에서 가장 널리 사용되고 있는 제어기는 비례-적분-미분제어기(Proportional-Integral-Differential : PID)로써, 그 구조가 간단하고 견실하며 제어성능이 우수하다. 기존에 PID제어기의 설계 방법은 수학적으로 모델링된 제어 시스템을 이용하여 PID 계수를 시행착오방법으로 튜닝하였다. 그러나 수학적으로 모델링된 시스템을 이용하여 설계된 PID제어기는 외란, 부하변화 및 내·외부의 환경변화가 발생한 경우에는 최적의 제어 성능을 기대하기가 어렵다. 또한 기존의 제어기는 일반적으로 제어 시스템의 수학적인 모델링에 기반을

두고 이루어져 왔기 때문에 제어 시스템의 동특성에 대한 정확한 지식이 필요하였다 [1-3]. 그러나 일반적으로 제어 시스템은 비선형이고 시변인 경우에는 그 시스템을 정확하게 모델링을 하기가 어렵다. 최근에는 이러한 문제점을 해결하기 위해서 확률적인 최적화 알고리즘인 유전알고리즘 및 지능기법을 이용하여 제어기를 설계하고 있다 [4-11].

1960년대부터 다양화 최적화 문제를 해결하기 위하여 자연 현상들을 모의한 알고리즘들이 개발되어 왔다. 대표적인 것들은 유전알고리즘 (genetic algorithm : GA), 진화전략 (evolution strategies : ES), 진화 프로그래밍 (evolutionary programming : EP) 등이 여기에 해당된다 [12-27]. 특히 자연 진화과정 그 자체가 하나의 탐색 메커니즘이고 여기에 살고 있는 생물들은 계속적으로 변하는 환경에 효율적으로 적응해 간다는 사실에 착안하여 만들어진 알고리즘이 진화연산(evolutionary computation : EC)이다. 자연생태계의 진화를 모의한 확률적인 최적화 탐색 알고리즘인 유전알고리즘은 주어진 최적화 문제에 대해 하나의 해가 아닌 해집단(population)으로 탐색하며, 해집단 내에서 더 좋은 적합도(fitness)를 가지는 해들을 이용하여 새로운 해집단을 생성한다는 점이다. 유전알고리즘의 특징으로는 첫째, 연속미분가능과 같은 목적함수에 대한 제한이 없다. 둘째, 문자열에 대한 부호화를 통해 다수의 변수들을 동시에 최적화할 수 있다. 셋째, 결정론적인 방법에 아닌 확률적인 방법을 사용하므로 지역극소점에 빠질 가능성이 적다. 넷째, 탐색공간내의 다수의 점들을 사용하여 동시에 병렬연산을 행하므로 전역해를 찾을 가능성이 높다. 다섯째, 개별 문자열에 대한 적합도 외에 기타의 정보가 필요치 않다. 이러한 장점으로 인해 유전알고리즘은 최근 주목받는 최적화 기법중의 하나가 되었으며 특히 공학적인 분야에서 많은 적용사례들이 발표되고 있다.

1. 2 연구의 내용

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 서보 모터의 실험장치를 제작하여 실험을 수행하여 제안한 방법의 유용성을 입증하였다. 본 논문의 연구내용을 요약하면 다음과 같다.

- (1) 본 논문에서는 확률적인 최적화 알고리즘인 유전알고리즘을 이용하여 직류 서보 모터용 PI제어기의 설계방법을 제안하였다.
- (2) 본 논문에서 제안한 방법의 유용성을 입증하기 위하여 직류 서보 모터의 속도 방정식을 이용하여 컴퓨터 시뮬레이션을 수행하였고, 기존에 시행착오법으로 설계된 PI제어기와 제어성능을 비교하였다. RGA-PI제어기의 설계에 사용된 조건과 RGA-PI제어기의 강인성 평가에 사용된 조건에 대해 성능사양 즉, 상승시간, 정정시간, 오버슈트 등을 통하여 본 논문에서 설계한 RGA-PI제어기의 유용성을 평가하였다.
- (3) 제안한 RGA-PI제어기의 유용성을 평가하기 위하여 직류 서보 모터의 속도 제어용 실험장치를 제작하여 실험을 하였다. 실험을 하기 위해 드라이브 회로 및 속도 센서 회로 등을 제작하였다. 본 논문에서 제안한 RGA-PI제어기의 제어성능을 비교하기 위해 기존의 PI제어기와 비교·검토하였다.

제 2 장 직류 서보 모터

2. 1 직류 서보모터의 특징

서보(servo)란 물체의 속도, 방향, 위치 등을 제어량으로 하여 목표치의 변화량을 추종하도록 구성된 제어계를 말하며, 서보모터는 목표치의 변화량을 추종하기 위한 구동원으로 사용하는 모터를 말한다. 서보모터는 1952년 미국에서 개발한 NC공작기를 시작으로 1962년에는 산업용 로봇에 사용되는 등 종래의 유압서보에서 모터를 사용한 전기서보로 발전되어 FA산업으로 분류되는 공장자동화 기기분야에서 생산 시스템의 성역화, 자동화, 무인화, 고효율화 등을 목적으로 한 산업용 로봇, NC공작기계, 자동화기계 등으로 확산되어 정밀기계산업에 크게 기여하게 되었다. 1970년대 들어서 전자공업의 발전과 다양한 생산체제를 요구하는 산업적 요구에 따라 전기서보는 보다 복잡한 구성을 갖게되어 기계분야와 전자분야가 결합된 메카트로닉스(mechatronics)라는 새로운 분야를 탄생시켰으며, 1980년대에 들어와서는 시각과 촉각의 기능을 갖추어 생산, 조립, 검사 등의 다양한 분야에 사용되는 FMS(flexible manufacturing system)의 핵심 구동원으로 사용되고 있다.

1990년대 이후 전기서보계의 핵심 구동원인 서보모터는 제어기술의 발전에 따라 아날로그 제어에서 디지털제어, 마이크로 프로세서를 이용한 컴퓨터 제어, DSP(digital signal processor)응용 등으로 발전되고 있다. 서보모터는 사용전원에 의한 분류, 구조에 의한 분류, 용도에 의한 분류, 제어목적에 의한 분류 등 목적에 따라 여러 가지 형태로 분류할 수 있고 종류도 다양하다.

그 중에서 직류 서보모터는 전기에너지를 기계적인 에너지로 변환하는

토크(torque)변환기로서 일반 모터에 비하여 입력신호에 따라 신속하게 기계 부하를 조작, 구동시킨다. 또한, 높은 기동성, 제어응답성 등의 장점으로 정밀 측정시스템, 반도체 제조장비, 첨단 검사장비, 정밀 가공기기 등의 산업용기기분야, 컴퓨터 주변기기, 사무 자동화기기, 영상음향기기 등의 멀티미디어 분야, 가전기기분야 등에서 핵심적 역할을 담당하므로서 급속도로 용도가 확대되어 21세기에는 반도체와 더불어 산업에서 중요한 역할을 수행하고 있다.

2. 2 직류 서보모터의 모델링

그림 1은 직류 서보모터의 모델을 나타낸 것이다. 직류 서보모터는 여자방식에 따라 직권, 분권, 타여자 및 영구자석 계자형으로 분류할 수 있다. 그 중에서 가장 대표적인 타여자 방식의 직류 모터에 대해서 일정한 계자 전류를 공급한 경우에는 계자 전류와 자속의 비선형성이 더 고려할 필요가 없으므로 다음과 같은 선형 방정식 (1), (2)를 얻을 수 있다.

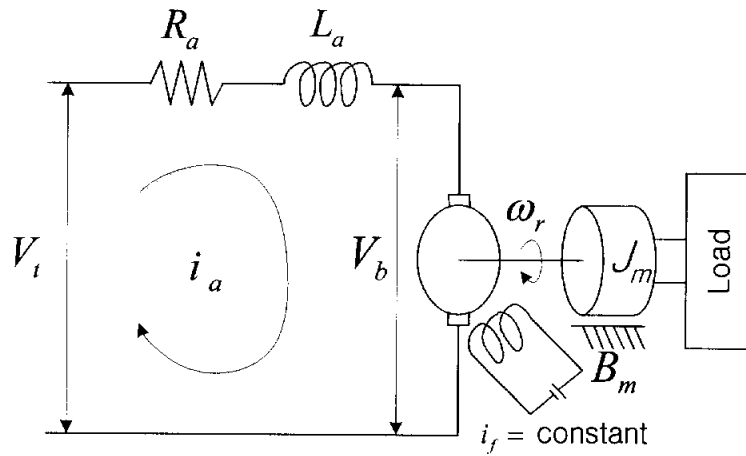


그림 1 직류 서보모터의 모델

Fig. 1 Model of DC servo motor

$$K_e \omega_r(t) = -R_a i_a(t) - L_a \frac{di_a(t)}{dt} + V_t(t) \quad (1)$$

$$K_T i_a(t) = J_m \frac{d\omega_r(t)}{dt} + B_m \omega_r(t) + T_L(t) \quad (2)$$

여기서, $\omega_r(t)$: 회전자 각속도 [rad/s]

$V_t(t)$: 단자 전압 [V]

$i_a(t)$: 전기자 전류 [A]

J_m : 회전자 관성 [Nm²]

$T_L(t)$: 부하 토크 [Nm]

R_a : 전기자 저항 [Ω]

L_a : 전기자 인덕턴스 [H]

B_m : 점성 마찰계수[Nm/rad/s]

K_T : 토크 상수[Nm/A]

K_e : 역기전력 상수[V/rad/s]

$T_e(t)$: 전동기에 의해 발생된 토크[Nm]

본 논문에서는 전류 제어 방식을 사용하며, 식 (2)에서 $T_e(t) = K_T i_a(t)$ 로 두면, 식 (3)과 같은 직류 모터의 속도 방정식을 얻을 수 있다.

$$T_e(t) = J_m \frac{d\omega_r(t)}{dt} + B_m \omega_r(t) + T_L(t) \quad (3)$$

여기서 $T_L(t) = \mu \omega_r(t)^2 \text{sign}(\omega_r(t))$

본 논문에서 사용한 전동기의 속도제어기에 대한 블록선도는 그림 2에 나타내었다. 그림 2에서 $G(s)$ 는 PI제어기에 대한 전달함수이고, 전류제어기 부분은 스위칭 소자에 대한 시지연으로 모델링하였고, T_a 는 전류제어기의 시정수이다. 그림 2의 전동기 속도제어기에 대한 블록선도는 식 (4), (5)와 같이 표현할 수 있다.

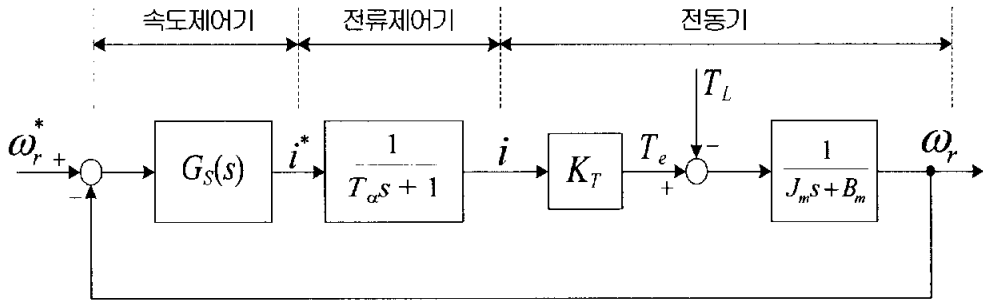


그림 2 전동기의 PI제어기에 대한 블록선도

Fig. 2 Block diagram for PI controller of motor

$$\frac{di_a(t)}{dt} = -\frac{1}{T_a} (i_a(t)^* - i_a(t)) \quad (4)$$

$$\frac{d\omega_r(t)}{dt} = \frac{1}{J_m} (K_T i_a(t) - B_m \omega_r(t) - T_L(t)) \quad (5)$$

제 3 장 실수형 유전알고리즘

3. 1 실수형 유전알고리즘의 특징

자연계 생물들은 유전자를 보존하고 생존을 지키기 위해 자손을 생산하고 끊임없이 환경과 투쟁하며 적응해 가는 일련의 과정은, 주어진 환경에서 최적이라는 사실에 입각하여, Holland는 유전학과 진화원리를 컴퓨터 알고리즘과 연결시키는 방법을 연구하였고, 그의 저서 “Adaptive in Natural and Artificial Systems”에 발표하게 되었는데, 이것이 최초의 유전알고리즘이다.

유전 알고리즘은 두 부모의 유전자로부터 그들 자손의 유전자를 형성하는 유성생식과 자연환경에서 일어나는 진화원리를 모의한 것이다. 이러한 과정으로 최적화 문제를 해결할 때 기본적으로 요구되는 전제는 인위적인 진화현상을 발생시킬 초기 해집단(population)을 구성하는 것이다. 해집단은 문제 공간에서 해를 뜻하는 다수의 개체로 형성되고 이들은 유전자의 역할을 반영하도록 흔히 비트형태로 표현된다. 이 해집단은 유성생식과 진화원리를 모방한 유전 연산자에 의해 개선된다. 각 개체는 집단의 구성원으로서 더 적합하고 덜 적합한 어떤 특징을 가지고 있기 때문에 적합도 함수에 의해 그 정도를 정량화 될 수 있다. 일반적으로 유전알고리즘은 스트링의 표현 방법에 따라 비트 유전알고리즘과 실수형 유전알고리즘(real-coding genetic algorithm : RGA)으로 분류된다. 실수형 유전 알고리즘은 실변수형 코딩방식으로 해집단을 이진수의 스트링이 아닌 실변수로 부호화하였다. 이렇게 실변수로 코딩함으로써 디코딩 과정이 필요 없고, 수학적인 기법을 적용할 수 있는 장점을

가지고 있다. 실수형 유전 알고리즘은 비트 유전 알고리즘과 동일한 과정으로 해를 탐색하지만, 유전 연산자인 교배 및 돌연변이 방법은 실변수에 적합하도록 변경해야 한다. 실수형 유전알고리즘의 기본 구성도는 그림 3에 나타내었으며, 실수형 유전알고리즘을 이용하여 최적해를 구하는 과정은 다음과 같다.

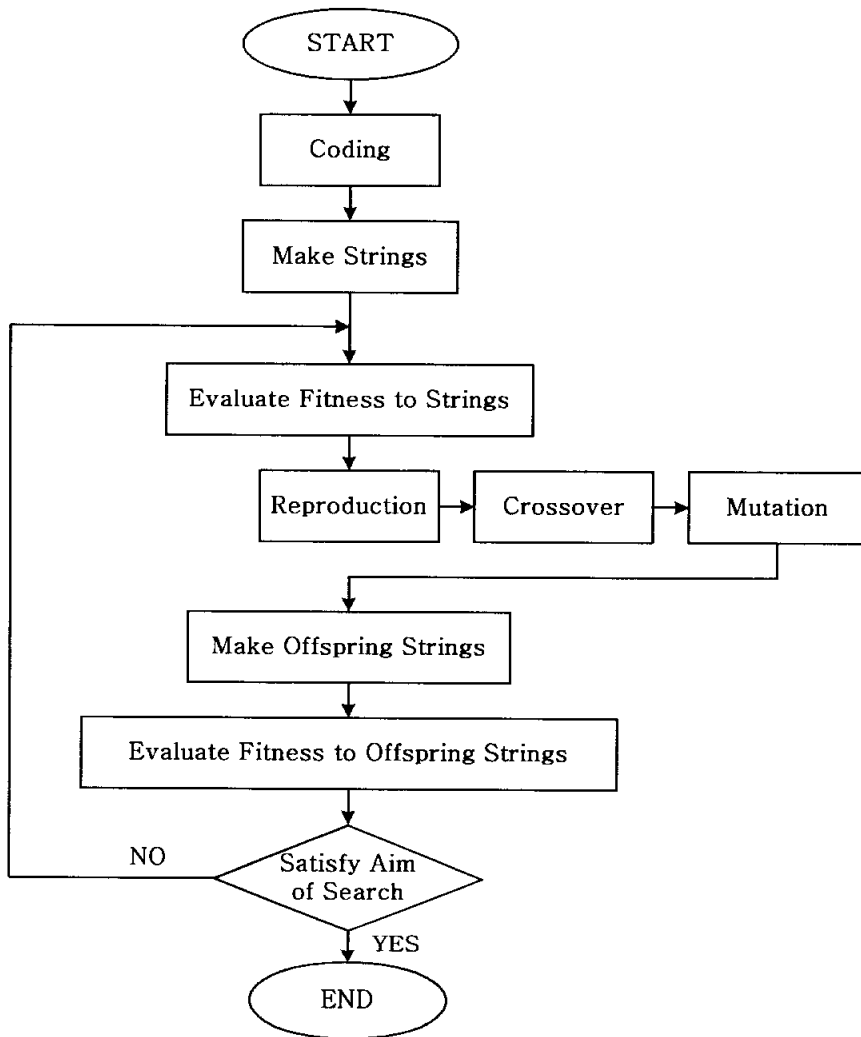


그림 3 실수형 유전알고리즘의 구성도

Fig 3 Configuration of real-coding genetic algorithm

단계 1) 초기화(initialize) : 주어진 문제의 제약조건을 고려하여 해가 될 가능성이 있는 개체들의 집단인 해집단을 형성한다. 이때 초기 해집단은 해 공간내에서 랜덤하게 분포되거나 아니면 경험적인 방법으로 선택한다.

단계 2) 평가(evaluation) : 각 스트링을 주어진 목적 함수에 따라 적합도를 계산한다. 각 개체의 적합도 값에 따라 현재 집단 내의 개체들의 재산을 위해 사용된다. 적합도가 높은 개체는 더 많이 복제되고 반면에 성능이 나쁜 개체들은 집단으로부터 소멸될 것이다. 이러한 적합도 함수는 목적함수의 값의 범위는 문제마다 다르기 때문에 보통 정해진 구간 사이의 양수값을 갖도록 표준화된 값을 사용한다. 즉, 표준화하기 이전의 적합도의 값을 raw fitness라고 하며 표준화되어서 실제로 개체 선택의 기준이 되는 함수를 적합도 함수 (fitness function)라고 한다. 적합도 함수는 해의 진화방향을 결정하는 함수로 유전 알고리즘에서는 큰 역할을 한다. 적합도 함수를 잘못 선정하면 최적해를 탐색하는데 많은 실행시간을 필요하게 된다.

단계 3) 복제(reproduction) : 복제단계에서는 적합도에 따라 부모 염색체들로부터 N 개의 염색체들을 랜덤하게 복제하는 과정이다. 자연 생태계의 적자생존의 원리를 이용한 것으로 높은 적합도를 가진 스트링일수록 다음 세대에 복제될 확률이 높다. 일반적으로 가장 널리 사용되는 복제방법으로는 Roulette wheel selection 방법, 토너먼트 선택법(tournament selection), 순위 선택법(ranking selection), 엘리트 보존 선택(elitist preserving selection) 등이 있다. 본 논문에서 사용

한 복제방법은 적합도에 비례하여 복제하는 룰렛휠(roulette wheel) 방법을 사용한다.

단계 4) 교배(crossover) 및 돌연변이(mutation) : 복제과정을 거친 염색체들을 일정한 확률로 부모염색체를 변화시켜 새로운 염색체를 생성하는 과정이다. 교배단계에서는 부모염색체의 유전자 정보를 서로 교환하여 새로운 정보를 가진 자손 염색체를 생성한다. 돌연변이 단계는 부모염색체 유전자형의 형태변화로 새로운 정보를 발생시키는 과정이다. 이러한 돌연변이는 매우 낮은 확률을 가지고 이행된다. 본 논문에서 사용한 교배 및 돌연변이 방법으로 식 (6)의 단순교배(simple crossover)방법과 식 (7)의 균일 돌연변이(uniform mutation)를 사용하였다.

① 단순교배 : t 세대에서 두 개의 스트링 $S'_v = [v_1, \dots, v_k, \dots, v_N]$ 와 $S'_w = [w_1, \dots, w_k, \dots, w_N]$ 를 임의로 선택하였을 때, k 번째 변수의 위치에서 교배가 발생했다면 t+1 세대의 자손은 다음 식 (6)과 같다.

$$\begin{array}{ll}
 \text{< 교배 전 >} & \text{< 교배 후 >} \\
 S'_v = [v_1, \dots, v_k, \dots, v_N] & S'^{t+1}_v = [v_1, \dots, v'_k, v'_{k+1}, \dots, v'_N] \\
 S'_w = [w_1, \dots, w_k, \dots, w_N] & S'^{t+1}_w = [w_1, \dots, w'_k, w'_{k+1}, \dots, w'_N]
 \end{array} \tag{6}$$

▲ 교배위치

여기서, $\dot{v}_j = \alpha_1 \times v_j + \alpha_2 \times w_j$

$$\dot{w}_j = \alpha_1 \times w_j + \alpha_2 \times v_j$$

α_1, α_2 : 0과 1사이의 임의의 수

v_j, w_j : 각 변수의 상한치와 하한치사이의 값

N : 각 스트링을 구성하는 변수의 개수

② 균일 돌연변이 : t 세대의 해집단 중에서 하나의 스트링

$S'_v = [v_1, \dots, v_k, \dots, v_N]$ 의 k 번째 변수가 돌연변이를 수행한
다면, $t+1$ 세대의 자손은 다음 식(7)과 같다.

<돌연변이 전>

< 돌연변이 후 >

$$S'_v = [v_1, \dots, v_k, \dots, v_N] \quad S'^{t+1}_v = [v_1, \dots, \dot{v}_k, v_{k+1}, \dots, v_N] \quad (7)$$

▲ 돌연변이 위치

여기서, $\dot{v}_k$: k 번째 변수의 상한치와 하한치사이의 임의의 수

단계 5) 수렴조건 : 일정한 세대수만큼 단계 2) ~ 단계 4)를 반복한다.

일반적으로 최적화 문제에서 실수형 유전알고리즘의 성능을 좋게 하기 위해서는 해집단의 크기, 염색체의 길이, 적합도, 교차 및 돌연변이 발생확률의 설정이 중요한 문제가 된다. 또한, 유전알고리즘은 전역적 탐색수행 등 많은 장점을 가지고 있지만 실행시간이 길고, 복잡한 다변수 비선형문제에서는 빨리 수렴하지 않는 경우도 발생한다.

3. 2 실수형 유전알고리즘을 이용한 PI제어기의 설계

일반적으로 PI제어기는 외란 및 제어시스템의 파라미터 변동에 대해서 뛰어난 제어 성능을 나타내지만, 다양한 외란 및 제어시스템의 파라미터 변동에 대해서 최적의 제어 성능을 얻을 수 없다. 따라서 본 논문에서는 다양한 외란 및 제어시스템의 파라미터 변동에 대해서 최적의 제어 성능을 얻기 위해서 PI제어기의 이득을 실시간으로 유전알고리즘을 이용하여 반복학습 방법으로 튜닝하였다. 그림 4는 실수형 유전알고리즘을 이용하여 PI제어기의 이득을 튜닝하기 위한 전체 구성도를 나타내었다. 그림 4에서 보는 것처럼 PI제어기의 입력으로 오차와 오차의 합을 사용하였고, 유전알고리즘의 입력으로 오차의 절대치 합을 이용하였다.

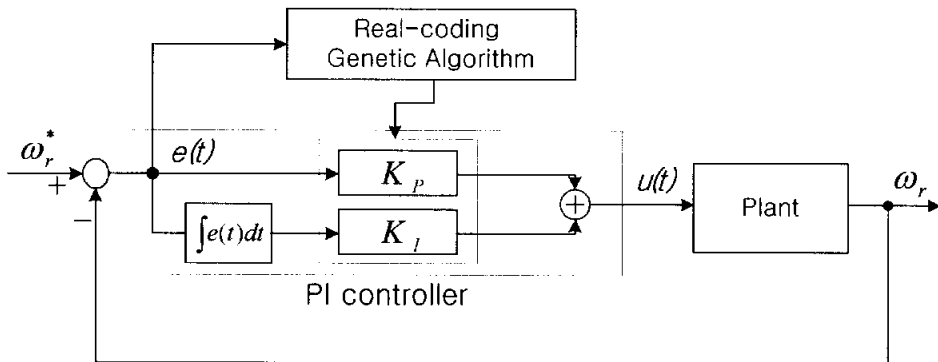


그림 4 실수형 유전알고리즘을 이용한 PI제어기 튜닝을 위한 구성도

Fig. 4 Configuration for the tuning of PI controller using RGA

실수형 유전알고리즘을 이용하여 PI제어기의 이득을 튜닝하기 위한 해집단의 구조는 그림 5에 나타내었다. 그림 5에서 보는 것처럼 실수형 유전알고리즘을 이용하여 PI제어기의 이득을 튜닝하는데 필요한 파라미터 수는 2개이다. 그리고 해집단의 각 스트링을 평가하기 위하여 사용된 적합도 함수는 식 (8)과 같이 오차의 절대치 합을 사용하였다.

$$\text{적합도} = \frac{1}{1 + \int_{t=0}^T |\omega_r^*(t) - \omega_r(t)|} \quad (8)$$

여기서, ω_r^* : 기준입력

ω_r : 제어대상의 출력

T : 특정한 시간 동안 획득한 데이터의 개수

S_1	$KP_{1,1}$	$KI_{1,2}$
S_2	$KP_{2,1}$	$KI_{2,2}$
		$\vdots$
S_n	$KP_{n,1}$	$KI_{n,2}$

여기서, KP_{ij} : 비례제어기의 이득

KI_{ij} : 적분제어기의 이득

n : 스트링 개수

그림 5 해집단의 구조

Fig. 5 Structure of population

실수형 유전알고리즘을 이용하여 PI제어기를 설계하기 위한 전체적인 흐름도는 그림 6에 나타내었고, 설계 과정은 다음과 같다.

단계 1) 해집단 구성: 그림 5과 같이 PI제어기의 이득을 이용하여 해집단을 구성한다.

단계 2) 평가: 식 (8)의 적합도 함수를 이용하여 단계 1)에서 생성된 각 스트링을 평가하기 위한 과정은 다음과 같다.

① PI제어기의 입력인 오차($e(t)$)와 오차의 합($\int e(t)dt$)을 계산한다.

② 아래의 식을 이용하여 PI제어기의 출력($u(t)$)을 계산한다.

$$u(t) = K_P \cdot e(t) + K_I \cdot \int e(t) dt \quad (9)$$

③ 미분방정식의 해법으로 제어대상 시스템을 푼다.

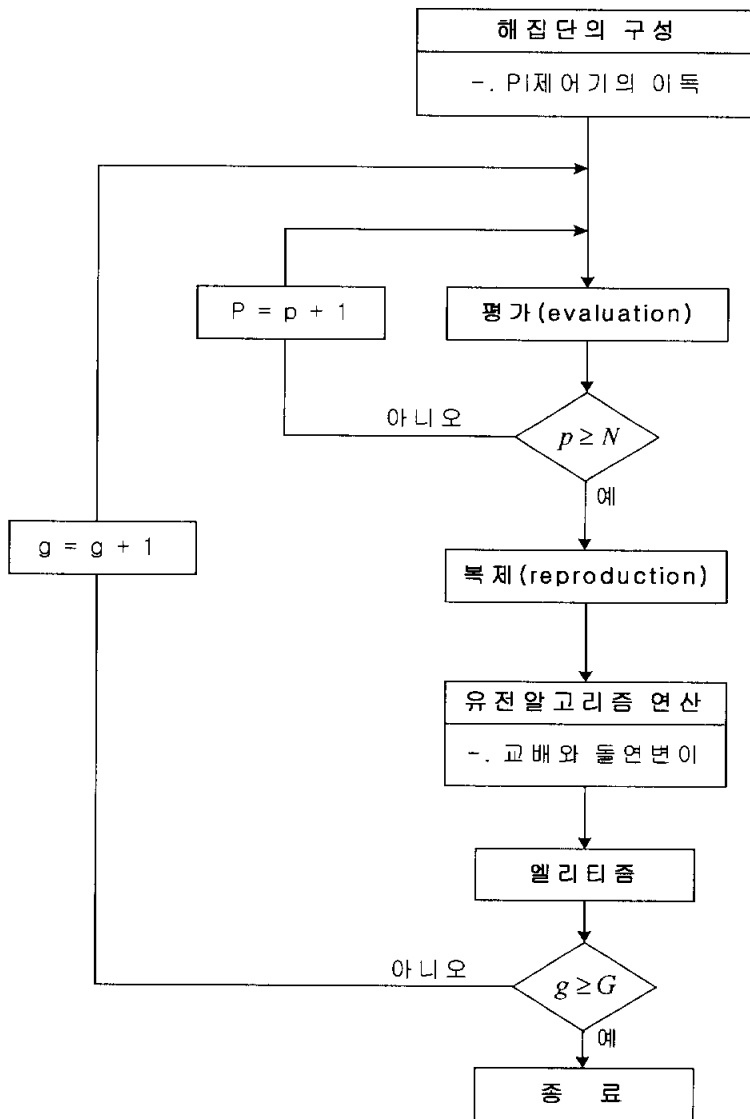
④ 시뮬레이션 시간 동안 ① ~ ③을 반복한다.

단계 3) 복제: 적합도에 비례하여 복제하는 룰렛휠 방법을 사용하였다.

단계 4) 교배 및 돌연변이: 실수형 유전알고리즘의 연산자인 교배와 돌연변이를 수행하여 자손을 생성한다.

단계 5) 엘리티즘: 적합도가 가장 높은 개체를 실수형 유전알고리즘의 해집단에 하나씩 복제하는 엘리티즘을 사용하였다.

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



여기서, N : 해집단의 크기

G : 특정한 세대수

그림 6 실수형 유전알고리즘을 이용한 PI제어기의 설계를 위한 흐름도

Fig. 6 Flow chart for the design of PI controller using RGA

제 4 장 시뮬레이션 및 실험

4. 1 시뮬레이션 방법

제안한 PI제어기의 성능을 검증하기 위해 제 2장에서 설명한 직류 서보 모터에 대해서 컴퓨터 시뮬레이션을 수행하였다. 식 (4)(5)에 대한 전동기의 미분방정식을 이용한 컴퓨터 시뮬레이션 과정은 다음과 같다. ①매 샘플링 시간($T_s = 0.01$)마다 기준속도에서 실제속도를 뺀 오차($e(t)$)와 오차의 합($\int e(t)dt$)을 계산한다. ②PI제어기의 출력 $u(t)(=i^*(t))$ 를 계산한다. ③매 샘플링시간마다 비선형 미분방정식의 해법인 Runge-kutta를 이용한 식(4)(5)를 푼다. ④위의 과정을 2초 동안 수행한다.

컴퓨터 시뮬레이션에 사용된 유전알고리즘의 시뮬레이션 계수 및 직류 서보모터의 파라메타는 표 1과 표 2에 나타내었다. 표 1과 같은 조건에서 실수형 유전알고리즘으로 PI제어기의 이득을 튜닝하는 데 소요되는 실행시간은 PC 586 1.5GHz로 약 2분 정도 소요되고, 시뮬레이션에 사용된 프로그램은 C언어로 구성하였다. 본 논문에서 제안한 실수형 유전알고리즘을 이용한 PI제어기(RGA-PI controller)와 성능을 비교하기 위해 시행착오법으로 튜닝한 PI제어기를 사용하였다.

표 1 실수형 유전알고리즘의 시뮬레이션 계수

Table 1 Coefficients for simulation using RGA

적용방법	해집단 수	교배 확률	돌연변이 확률	세대수	실행 시간
RGA	20	0.95	0.05	50	2분

표 2 직류 서보 모터의 파라미터

Table 2 Parameters of DC servomotor

대상	토크 정수 [Kgf.cm/A]	회전자 관성 [Kgf.cm/sec ²]	전기자 저항 [Ω]	전기자 인덕턴스 [mH]
직류 서보 모터	2.10	1.33×10^{-3}	2.12	1.02

그림 7은 매 세대별 적합도 수렴특성을 나타내었다. 그림에서 보는 바와 같이 세대가 증가함에 따라 적합도가 개선되고 있음을 알 수 있다. 이것은 PI제어기가 실수형 유전알고리즘에 의해서 최적화되고 있다는 것을 증명한다.

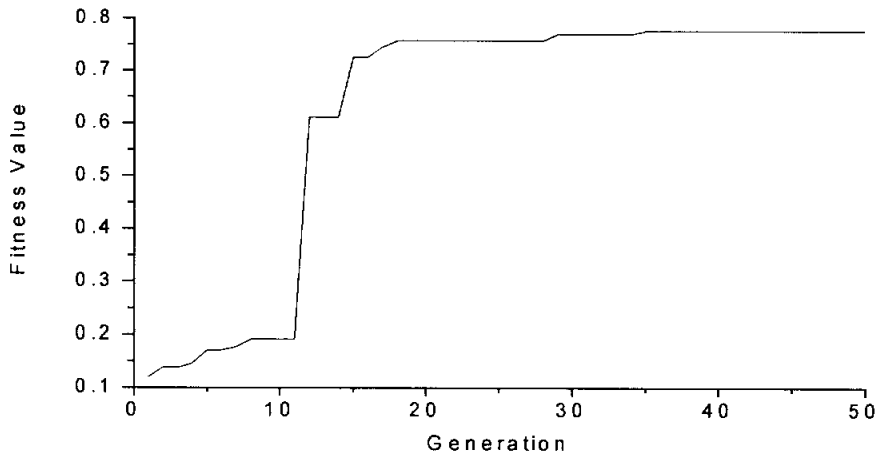


그림 7 매세대별 적합도 추이

Fig. 7 The change of fitness in each generation

4. 2 시뮬레이션 결과

본 논문에서 실수형 유전알고리즘에 의해 최적화된 PI제어기와 기존의 PI제어기와 제어성능을 비교하기 위해 사용된 조건은 표 3과 같다. 성능 비교에 사용한 조건은 PI제어기의 설계에 사용된 조건과 PI제어기의 강인성 평가에 사용된 조건으로 분류한다. PI제어기의 설계에 사용된 조건은 실수형 유전알고리즘으로 PI제어기의 튜닝에 사용된 조건은 표 3의 경우1에서 기준 속도가 1200[rpm]일 때를 이용하였다.

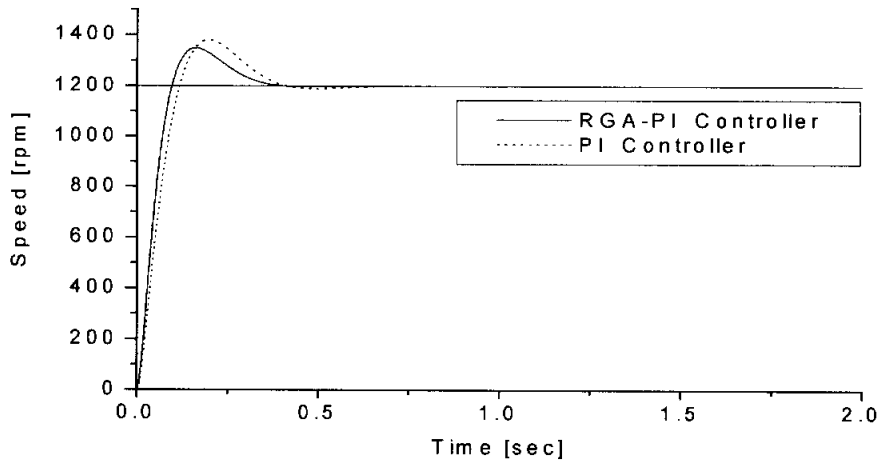
표 3. 제어기 성능평가에 사용된 시뮬레이션 조건

Table 3. Simulation cases used in evaluation of controller performance

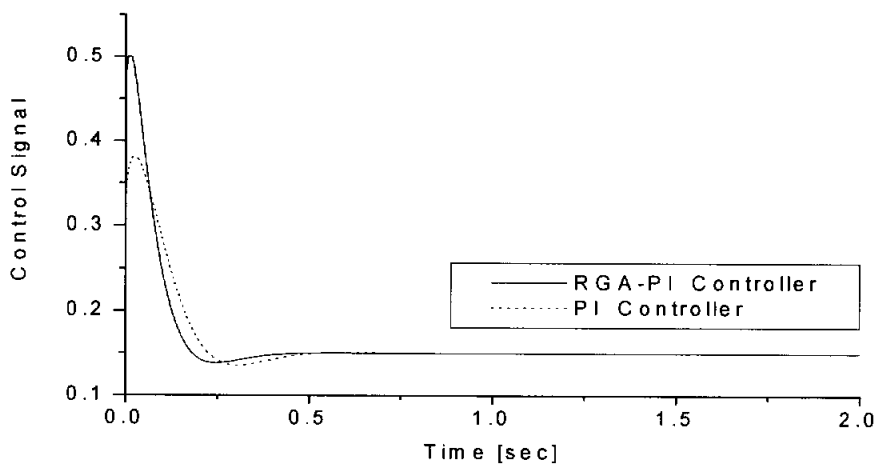
시뮬레이션 사례	동작조건	외란종류 (부하토크 변화)	시뮬레이션 시간
경우1 (일정부하토크)	400 [rpm]	$\mu=0.069$	2 [sec]
	800 [rpm]	$\mu=0.069$	2 [sec]
	1200 [rpm]	$\mu=0.069$	2 [sec]
경우2 (부하토크변화) 1초 후	400 [rpm]	$\mu=0.069 \rightarrow \mu=0.69$	2 [sec]
	800 [rpm]	$\mu=0.069 \rightarrow \mu=0.69$	2 [sec]
	1200 [rpm]	$\mu=0.069 \rightarrow \mu=0.69$	2 [sec]
경우3 (정·역 변화) 1초 후	400 [rpm]	$\mu=0.069$ (정 $\rightarrow$ 역)	2 [sec]
	800 [rpm]	$\mu=0.069$ (정 $\rightarrow$ 역)	2 [sec]

그림 8는 실수형 유전알고리즘을 이용하여 PI제어기의 설계에 사용된 기준속도에 대한 직류 서보모터의 시뮬레이션 결과를 나타내었다. 그림 8에서 보는 것처럼 RGA를 이용한 PI제어기가 기존의 PI제어기보다는 상승시간

과 최대오버슈트면에서 좋은 제어성능을 보임을 알 수 있다. RGA를 이용한 PI제어기의 강인성을 평가하기 위하여 PI제어기의 설계에 사용되지 않은 기준속도 및 부하토크 변화($\mu=0.069 \rightarrow \mu=0.69$)에 대해서 기존의 PI제어기와 제어 성능을 비교하였다. 그림 9는 기준속도가 400[rpm], 800[rpm]일 때 속도응답 특성을 나타내었다. 그림 9에서 보는 것처럼 RGA를 이용한 PI제어기가 상승시간 및 최대오버슈트면에서 기존의 PI제어기보다 더 우수한 제어 성능을 나타냄을 알 수 있다. 그림 10과 그림 11은 기준속도가 400[rpm], 800[rpm], 1200[rpm]이고, 1초 후에 토크 부하의 계수($\mu=0.069 \rightarrow \mu=0.69$)가 변화했을 때 속도응답 특성을 나타내었다. 그림 10과 그림 11에서 보는 것처럼 RGA를 이용한 PI제어기가 기존의 PI제어기보다 부하변화에 대해서 우수한 제어 성능을 보였다. 그림 12은 기준 속도가 400[rpm], 800[rpm]으로 회전하다가 1초 후에 역회전했을 때의 속도응답 특성을 보였다. 그림 12에서 보는 것처럼 RGA를 이용한 PI제어가 기존의 PI제어기보다 더 우수한 성능을 보임을 알 수 있다.



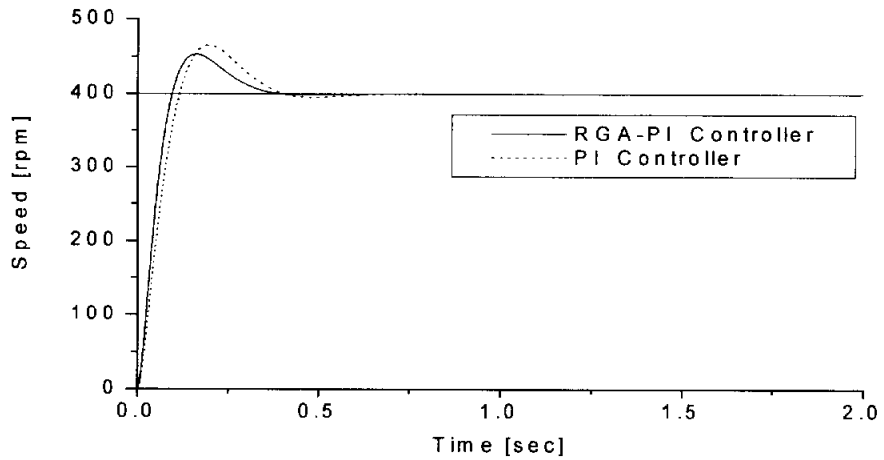
(a) 스텝응답 1200[rpm]



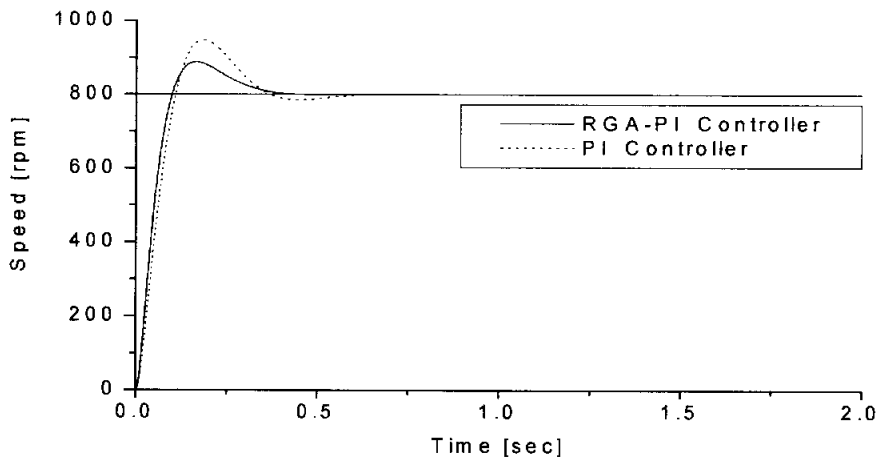
(b) 제어 입력

그림 8 기준속도가 1200[rpm]일 때 직류 서보모터의 속도응답

Fig. 8 Responses of DC servomotor in case of 1200 [rpm]



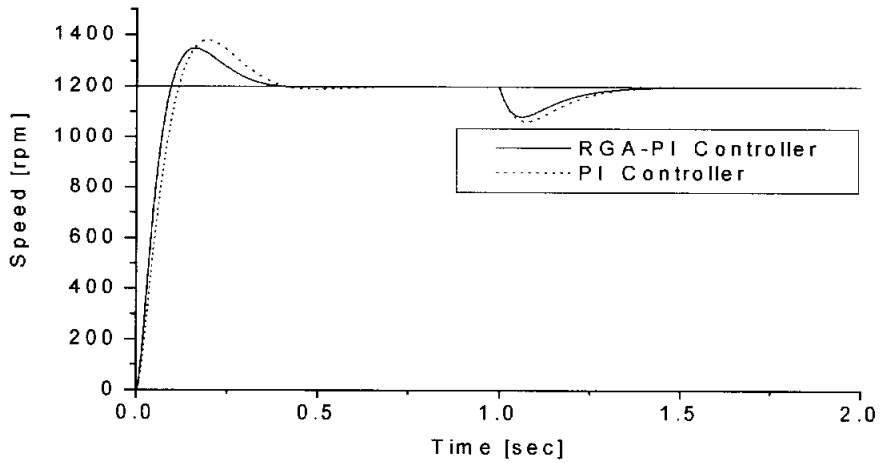
(a) 스텝응답 400[rpm]



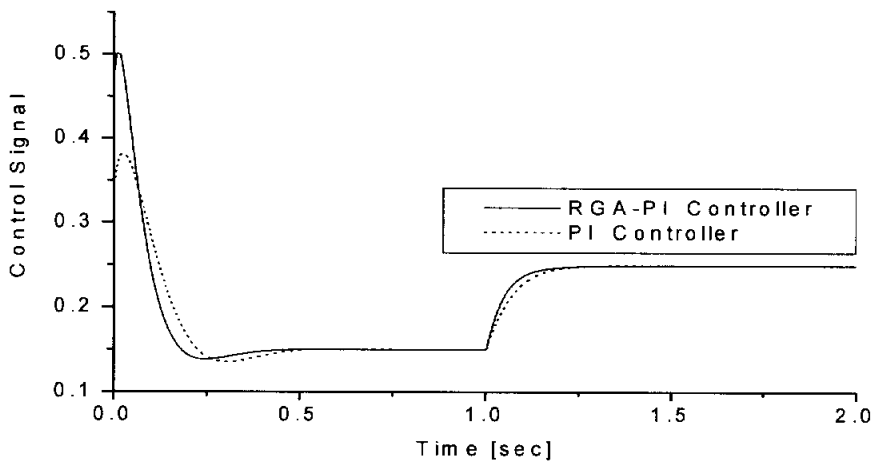
(b) 스텝응답 800[rpm]

그림 9 기준속도가 800[rpm], 400[rpm]일 때 직류 서보모터의 속도응답

Fig. 9 Responses of DC servomotor in case of 800[rpm] and 400[rpm]



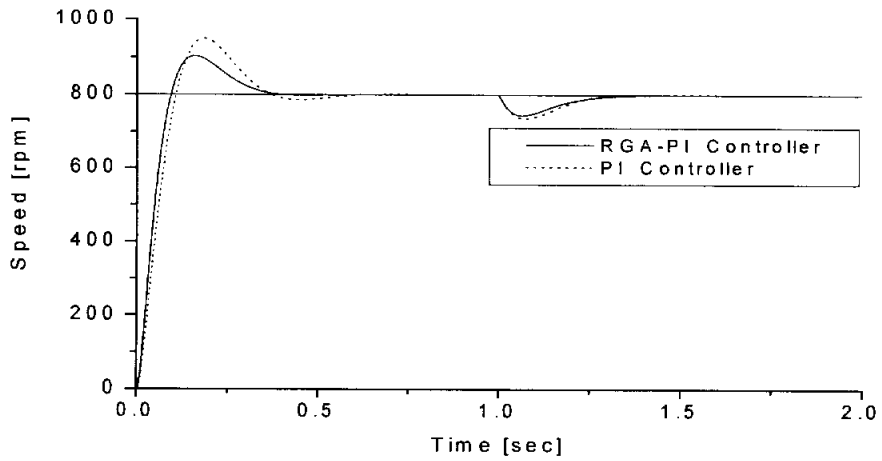
(a) 스텝응답 1200[rpm]



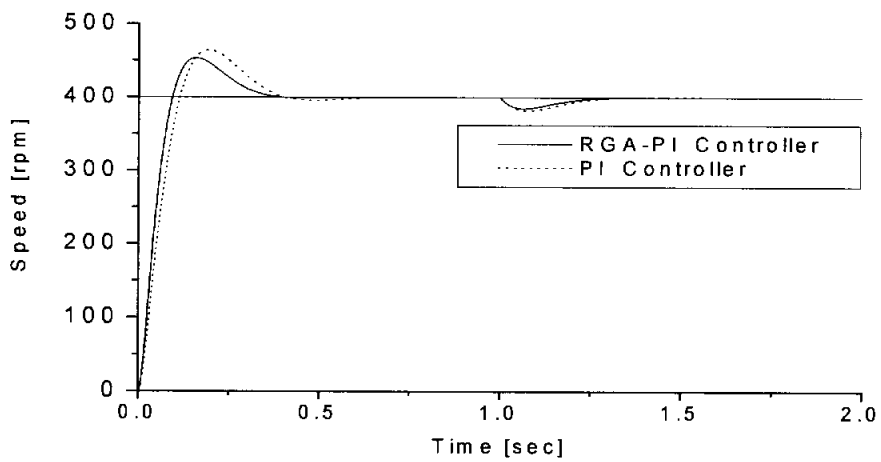
(b) 제어 입력

그림 10 기준속도가 1200[rpm]에 대해서 토크부하가 변화했을 때의 속도 응답

Fig. 10 Responses of DC servomotor for the change of torque load



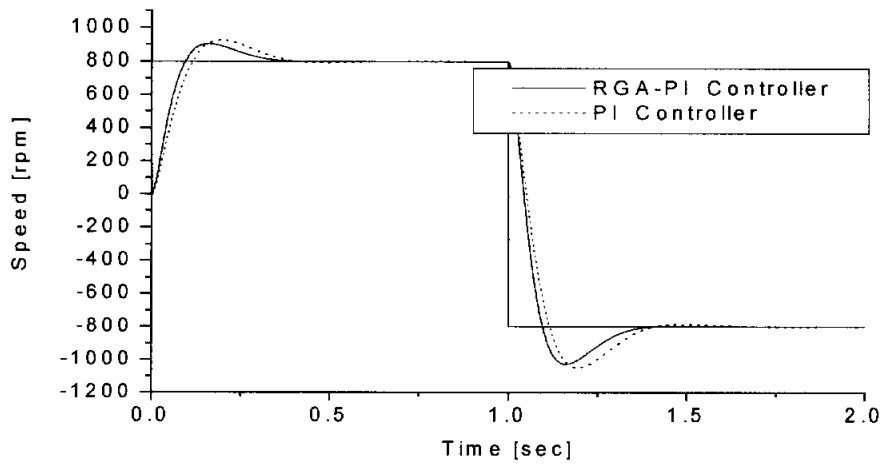
(b) 스텝응답 800[rpm]



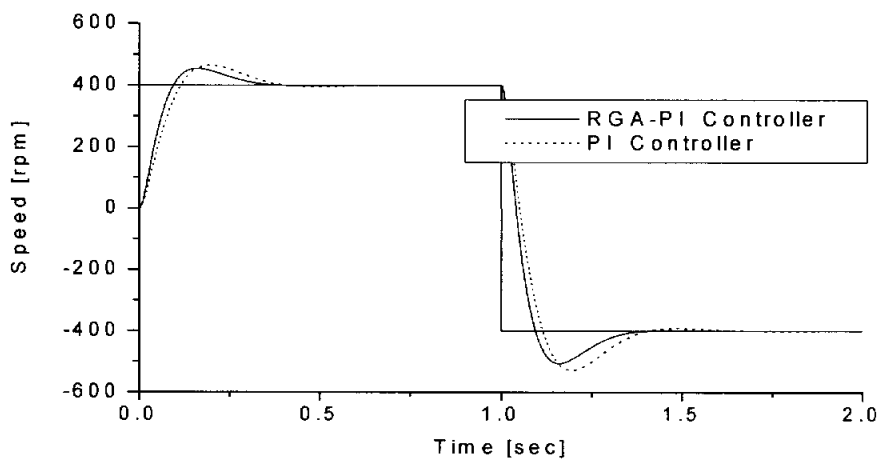
(c) 스텝응답 400[rpm]

그림 11 기준속도가 800[rpm], 400[rpm]에 대해서 토오크부하가 변화했을 때의 속도응답

Fig. 11 Responses of DC servomotor for the change of torque load



(a) 스텝응답 800[rpm]



(b) 스텝응답 400[rpm]

그림 12 직류 서보모터의 속도응답

Fig. 12 Responses of DC servomotor

4. 3 실험 방법

RGA를 이용한 PI제어기의 유용성을 검증하기 위해 직류 서보모터의 속도제어를 위한 실험장치를 구성하였다. 직류 서보모터의 속도제어를 위한 실제 구성은 그림 13에 나타내었다. 그림 13에서 보는 바와 같이 RGA를 이용하여 PI제어기의 이득을 반복튜닝 방법으로 해집단 수와 세대수를 만듦으로써 실행한다. 본 논문에서 RGA를 이용한 PI제어기 설계에 사용된 세대수는 50세대이고, 따라서 50세대 후, 최적의 PI제어기의 이득이 RGA에 의해 구해진다. 전체 속도 제어 시스템은 전류 루프와 속도루프로 구성되어 있으며, 전류루프의 샘플링 시간은 $250[\mu\text{sec}]$ 이고, 속도루프의 샘플링 시간은 $4[\text{msec}]$ 이다.

그림 13에서 전류센서에 의해 $250[\mu\text{sec}]$ 로 측정된 전류값은 하드웨어 인터페이스 카드인 818Lab Card를 이용하여 A/D변환을 거친 후 제어기의 출력과 비교된다. 이 전류값은 PWM제어기에 입력이 되고, PWM출력은 818Lab Card의 디지털 출력 port를 통해 구동회로로 들어간다. 구동회로는 이 신호를 이용하여 직류 서보모터를 제어한다. 속도루프에서는 속도센서인 엔코더를 이용하여 $4[\text{msec}]$ 로 서보모터의 속도를 측정하고 818Lab Card의 A/D변환을 거쳐 기준속도와 비교된 후, PI제어기의 입력신호로 사용된다. 직류 서보모터의 구동은 IGBT의 온/오프(on/off)의 반복으로 이루어지기 때문에, 동일한 상의 두 스위치가 동시에 온(on)되면 순간적으로 IGBT가 단락되어 소자가 파괴된다. 이러한 현상을 방지하기 위해서 PWM 전류 제어기의 후단에 지연 회로를 설치하여 동시에 온(on)이 되는 것을 방지하였다. 실험에 사용된 전동기와 부하로 사용한 발전기의 사양은 표 4와 같다.

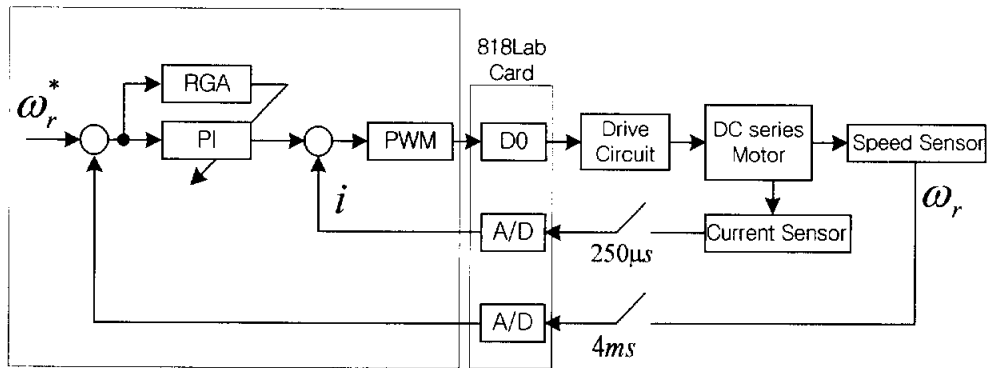


그림 13 직류 서보모터의 속도제어를 위한 실험장치

Fig. 13 Laboratory setup for DC servo motor speed control

표 4 직류서보모터 및 발전기의 사양

Table 4 Specification of motor and generator

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

그림 13의 직류 서보 모터 속도 제어를 위한 전체 시스템은 PC 586 350MHz에서 구성된 PI제어기 및 PWM 전류제어기와 구동회로로 구성된다. PI제어기와 PWM 전류제어기는 C언어를 이용하여 구성하였고, 속도 신호와 제어 신호를 PC와 인터페이스 하기 위해서 818 Lab Card를 사용하였다. 그리고 직류 서보모터의 실험장치는 그림 14에 나타내었고, 직류 서보모터의 스위칭 회로도와 구동회로는 그림 15에 나타내었다.

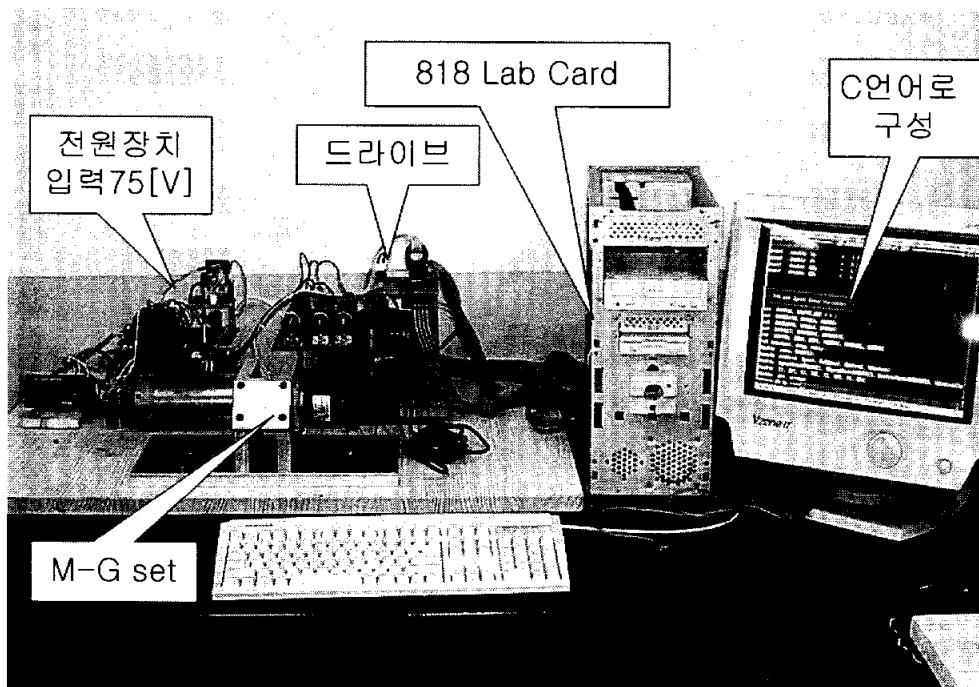
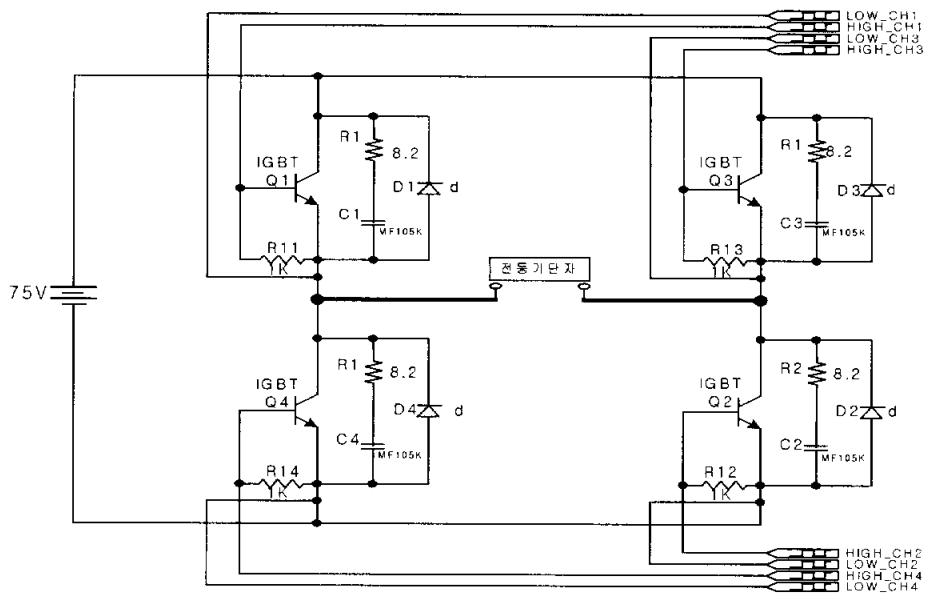
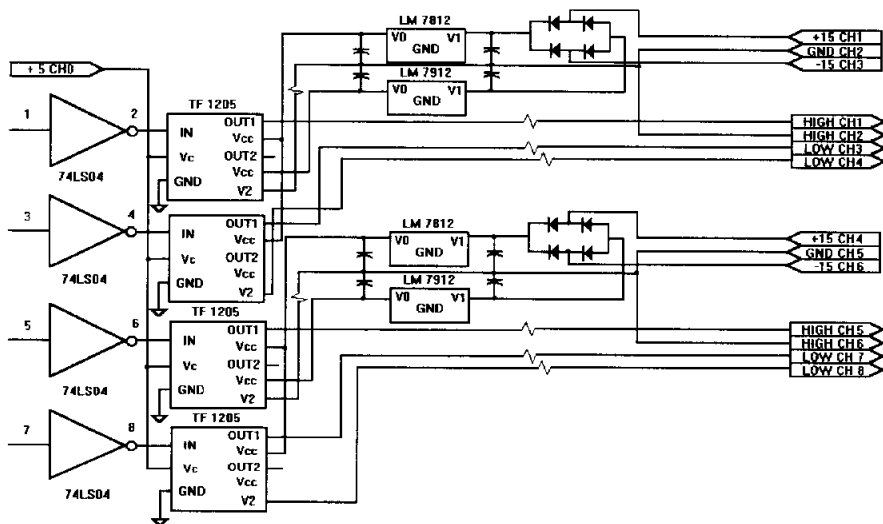


그림 14 직류 서보모터의 실험장치

Fig. 14 Experimental apparatus of a DC servomotor system



(a) 스위칭 회로도



(b) 구동회로

그림 15 직류 서보모터의 회로도

Fig. 15 Circuits of DC servo motor

4. 4 실험 결과

앞의 4. 3에서 실험장치를 이용한 RGA-PI제어기의 성능을 비교하기 위해 사용된 실험조건은 표 5와 같다. PI제어기의 설계에 사용된 조건은 실수형 유전알고리즘으로 PI제어기의 튜닝에 사용된 조건은 표 5의 경우 1에서 기준속도가 1200[rpm]일 때를 이용하였다.

표 5. 제어기 성능평가에 사용된 실험 조건

Table 5. Experimental cases used in evaluation of controller performance

실험 사례	동작조건	외란 종류 (부하 변화)	실험 시간
경우1 (일정부하)	400 [rpm]	100[W]	2 [sec]
	800 [rpm]	100[W]	2 [sec]
	1200 [rpm]	100[W]	2 [sec]
경우2 (부하변화) 1초 후	400 [rpm]	100[W]→200[W]	2 [sec]
	800 [rpm]	100[W]→200[W]	2 [sec]
	1200 [rpm]	100[W]→200[W]	2 [sec]
경우3 (정·역 변화) 1초 후	400 [rpm]	100[W] (정→역)	2 [sec]
	800 [rpm]	100[W] (정→역)	2 [sec]

그림 16은 실수형 유전알고리즘을 이용하여 PI제어기의 설계에 사용된 기준속도에 대한 직류 서보모터의 실험결과를 나타내었다. 그림 16에서 보는 것처럼 RGA를 이용한 PI제어기가 기존의 PI제어기보다는 상승시간과 정정시간면에서 더 좋은 제어성능을 보임을 알 수 있다. 제안한 PI제어기의 강인성

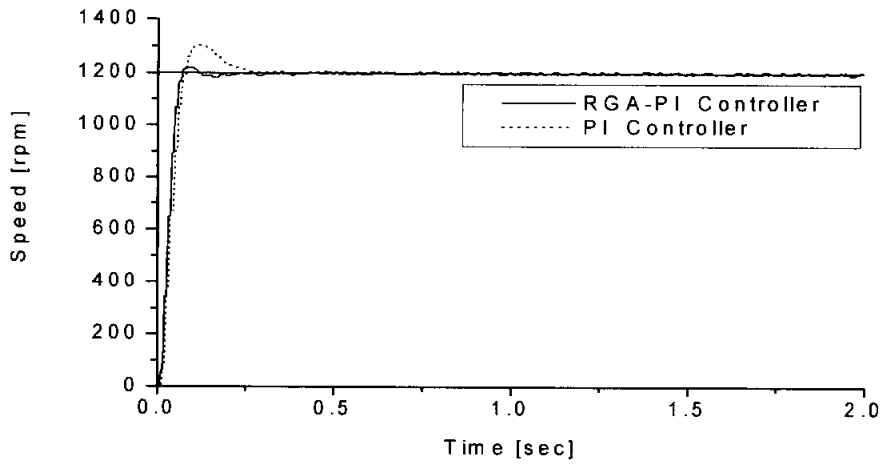
을 평가하기 위하여 PI제어기의 설계에 사용되지 않은 기준속도 및 부하변화에 대해서 기존의 PI제어기와 제어 성능을 비교하였다. 그림 17은 기준속도가 400[rpm], 800[rpm]일 때 속도응답 특성을 나타내었다. 그림 17에서 보는 것처럼 RGA를 이용한 PI제어기가 정정시간 및 상승시간면에서 기존의 PI제어기보다 더 우수한 제어 성능을 나타낼 수 있다. 그림 18는 속도가 800[rpm], 1200[rpm]이고, 1초 후에 부하가 변화했을 때 속도응답 특성을 나타내었다. 그림 18에서 RGA를 이용한 PI제어기가 기존의 PI제어기보다 부하변화에 대해서는 유사한 제어 성능을 보였다. 그림 19은 기준 속도가 400[rpm], 800[rpm]으로 회전하다가 1초 후에 역회전했을 때의 속도응답 특성을 보였다. 그림 19에서 보는 것처럼 RGA를 이용한 PI제어가 기존의 PI제어기보다 더 우수한 성능을 보임을 알 수 있다.

이상의 결과를 요약하면 표 6와 같고, 표 6에서 경우1은 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 5에서 보는 것처럼 경우1에 대해서는 RGA를 이용한 PI제어기가 기존의 PI제어기보다 정정시간에서는 0.05[sec] ~ 0.14[sec](25% ~ 54%) 정도 더 빨리 정상상태에 도달하였고, 최대오버슈트에서는 4 [%]~8[%] 정도 더 적었다. 경우 2에 대해서는 RGA를 이용한 PI제어기와 기존의 PI제어기가 정정시간과 오버슈트에 대해서 유사한 결과를 나타내었다.

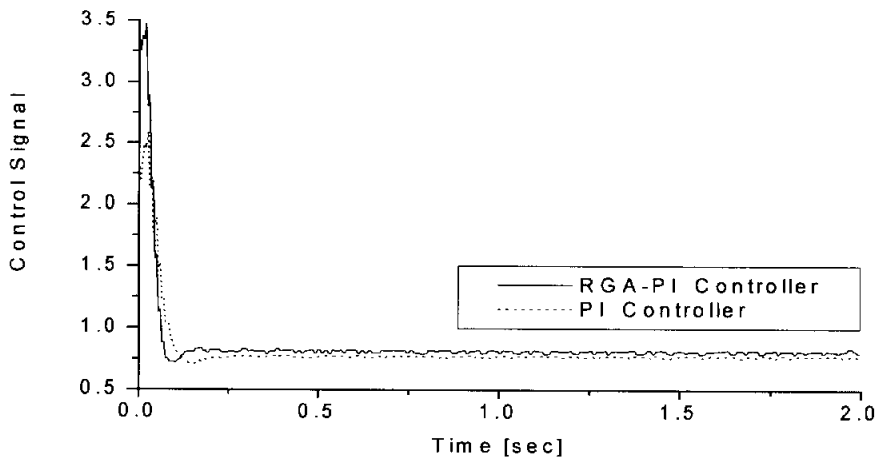
표 6 기존의 PI제어기와 제안한 PI제어기의 제어 성능 비교

Table 6 Comparisons of conventional and PI controller using RGA

방 법 기준속도		정정시간[sec]		오버슈트[%]	
		RGA-PI제어기	PI제어기	RGA-PI제어기	PI제어기
경우1 (일정 부하)	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 (부하 증가)	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



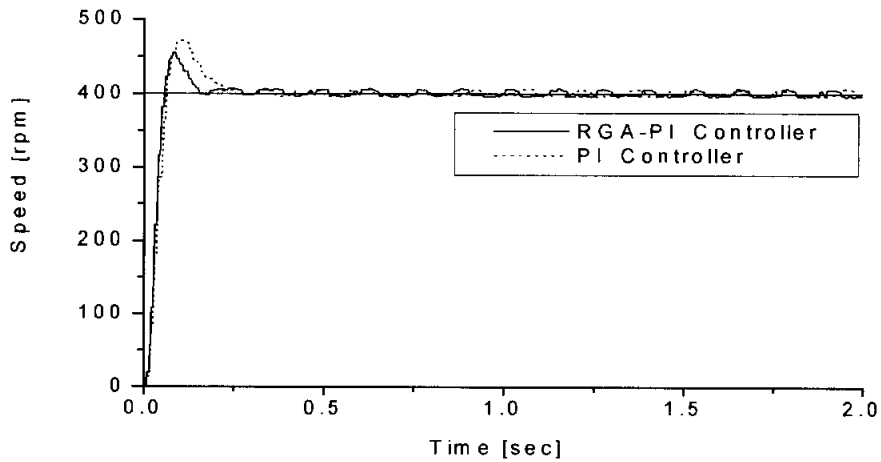
(a) 속도응답(1200[rpm])



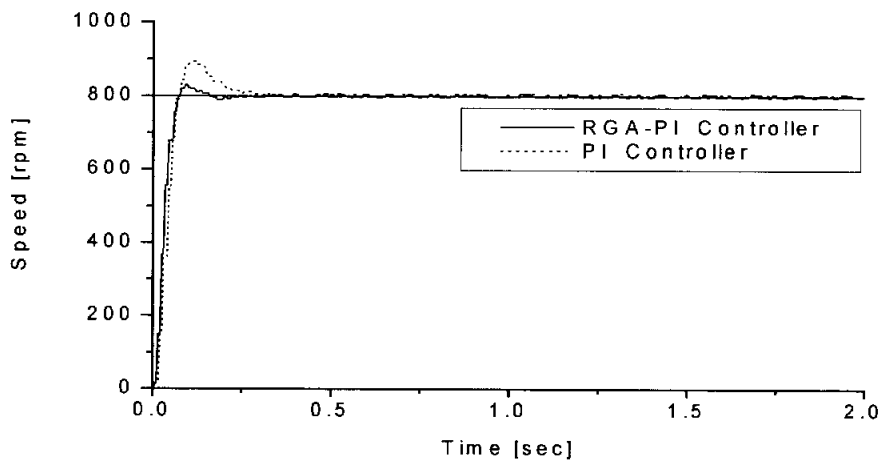
(b) 제어입력

그림 16 기준속도가 1200[rpm]일 때 직류 서보모터의 속도응답

Fig. 16 Responses of DC servomotor in case of 1200 [rpm]



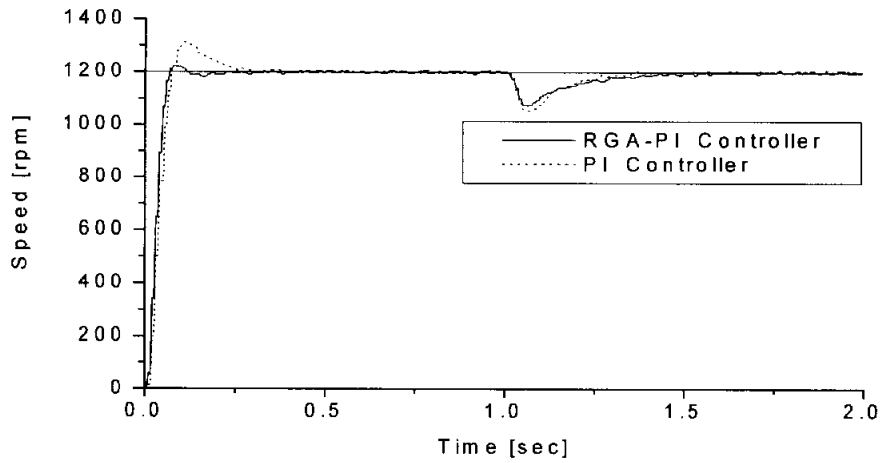
(a) 속도응답(400[rpm])



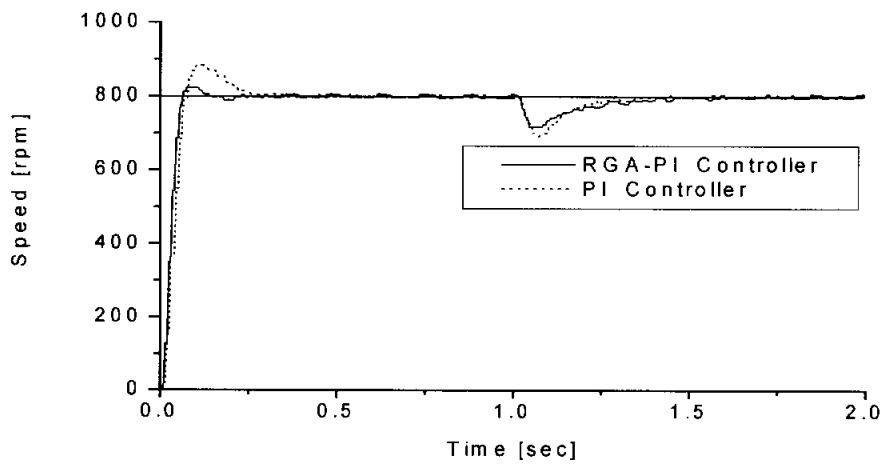
(b) 속도응답(800[rpm])

그림 17 기준속도가 400[rpm], 800[rpm]일 때 직류 서보모터의 속도응답

Fig. 17 Responses of DC servomotor in case of 400[rpm] and 800[rpm]



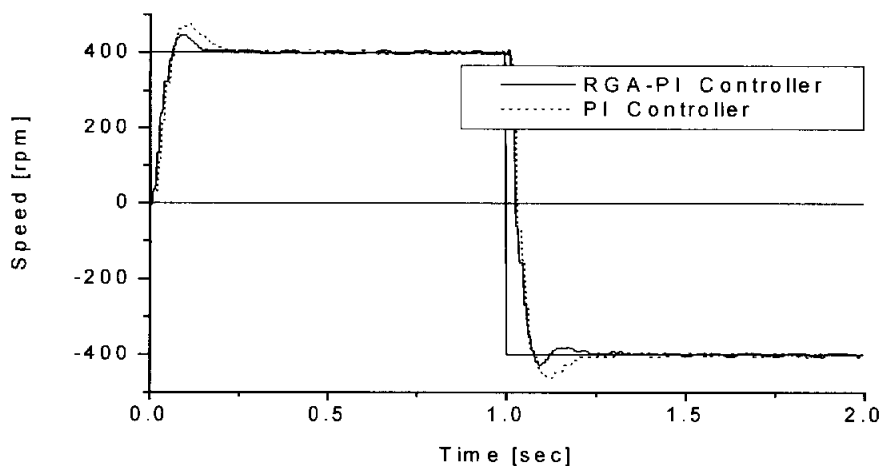
(a) 속도 응답(1200[rpm])



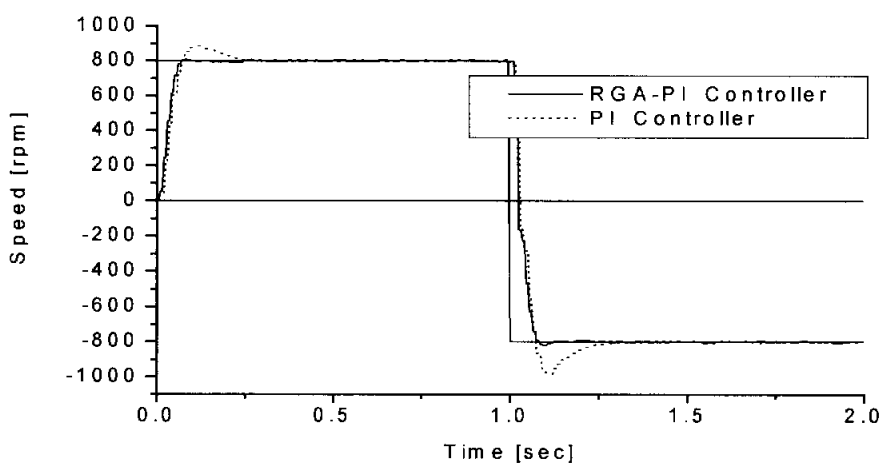
(b) 속도 응답(800[rpm])

그림 18 부하가 변화했을 때의 속도응답

Fig. 18 Responses of DC servomotor for the change of load



(a) 속도응답(400[rpm])



(b) 속도응답(800[rpm])

그림 19 직류 서보모터의 속도응답

Fig. 19 Responses of DC servomotor

제 5 장 결 론

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 시행착오 방법으로 설계된 PI제어기와 비교·검토하였다.

컴퓨터 시뮬레이션 결과, 다양한 기준속도 변화 및 토크부하 변화에 대해서 제안한 RGA를 이용한 PI제어기가 기존의 PI제어기보다 최대오버슈트 및 상승시간에서 더 우수한 제어성능을 보임을 알 수 있었다. 직류 서보모터의 실험결과, 일정부하에 대해서 기준속도가 400[rpm]에서 1200[rpm]까지 변화했을 때 제안한 RGA를 이용한 PI제어기가 기존의 PI제어기보다 정정시간에서는 0.05[sec] ~ 0.14[sec](25% ~ 54%) 정도 더 빨리 정상상태에 도달하였고, 최대오버슈트에서는 4[%] ~ 8[%] 정도 더 적었다. 1초 후에 부하의란이 증가하였을 때 RGA를 이용한 PI제어기와 기존의 PI제어기는 정정시간과 오버슈트에 대해서는 유사한 결과를 나타내었다.

참고 문헌

- [1] F. Betin, D. Pinchon and G. A. Capolino, "A time-varying sliding surface for robust position control of a DC motor drive", *IEEE Transactions 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] Seung-Min Baek and Tae-Yong Kuc, "An adaptive PID learning control of DC motors", *IEEE International Conference on Systems, Man, and Cybernetics*, Vol. 3, pp. 2877-2882, 1997
- [4] I. S. Baruch, R. Garrido, J. M. Flores and J. C. Martinez, "An adaptive neural control of a DC motor", *Proceedings of the 2001 IEEE International Symposium on Intelligent Control*, pp. 121-126, 2001
- [5] 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
- [6] Jun Oh Jang and Pyeong Gi Lee, "Neuro-fuzzy control for DC motor friction compensation", *Proceedings of the 39th IEEE Conference on Decision and Control*, Vol. 4, pp. 3550-3555, 2000

- [7] 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
- [8] A. Khoei, Kh. Hadidi and S. Yuvarajan, "Fuzzy-logic DC-motor controller with improved performance", *The 1998 IEEE Industry Applications Conference, Thirty-Third IAS Annual Meeting*, Vol. 3, pp. 1652-1656, 1998
- [9] J. T. Teeter, Mo-yuen Chow, J. J. Brickley and Jr, "A novel fuzzy friction compensation approach to improve the performance of a DC motor control system", *IEEE Transactions on Industrial Electronics*, Vol. 43, pp. 113-120, 1996
- [10] C. H. M. Kemenade, S. H. Kok and A. E. Eiben, "Raising GA Performance by Simultaneous Tuning of Selective Pressure and Recombination Disruptiveness", *IEEE International Conference on Evolutionary Computation*, Vol. 1, pp. 346-351, 1995
- [11] J. M. Renders and S. P. Flasse, "Hybrid Methods Using Genetic Algorithms for Global Optimization", *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, Vol. 26, No. 2, April, 1996
- [12] D. E. Goldberg, *Genetic Algorithms in Search, ptimization, and Machine Learning*, Addison-Wesley publishing Company, INC., 1989
- [13] D. B. Fogel, "An Introduction to Simulated Evolutionary Optimization", *IEEE Transactions on Neural Networks*, Vol. 5, No. 1, Jan. 1994

- [14] D. B. Fogel, Unearthing a Fossil from the History of Evolutionary Computation, *Fundamenta Informaticae*, PP. 1-16, 1998
- [15] Th. Back, Evolutionary Algorithms in Theory and Practice, *Oxford University Press*, New York, 1996
- [16] Th. Back and H. P. Schwefel, "Evolutionary Computation: An overview", *Proceeding of the Third IEEE Conference on Evolutionary Computation*, pp. 20-29, 1996
- [17] D. B. Fogel, Evolutionary Computation : Towards New Philosophy of Machine Intelligence, *IEEE Press*, Piscataway, NJ
- [18] J. Arabas, Z. Michalewicz, and J. Mulawka, "GAVaPS-a Genetic Algorithm with Varying Population Size", *IEEE International Conference on Evolutionary Computation*, pp. 73-78, 1994
- [19] D. Schlierkamp-Voosen and H. Muhlenbein, "Adaptation of Population Sizes by Competing Subpopulations", *IEEE International Conference on Evolutionary Computation*, pp. 330-335, 1996
- [20] M. Srinivas and L. M. Patnaik, "Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 24, No. 4, pp.656-667, April, 1994
- [21] W. M. Spears, Evolutionary Programming IV, *The MIT Press*, 1995
- [22] Robert Hinterding, "Gaussian Mutation and Self-adaption for Numeric Genetic Algorithms", *IEEE International Conference on Evolutionary Computation*, pp. 384-389, 1995

- [23] R. Hinterding, "Self-adaptation using Multi-Chromosomes", *IEEE International Conference on Evolutionary Computation*, pp. 87-91, 1997
- [24] H. P. Schwefel, "Evolution and Optimum Seeking", *Sixth Generation Computer Technology Series*, Wiley, 1995
- [25] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, *Springer-Verlag*, 1992
- [26] R. Hinterding, Z. Michalewicz and A. E. Eiben, "Adaptation in Evolutionary Computation : A Survey", *IEEE International Conference on Evolutionary Computation*, pp. 65-69, 1997
- [27] Mitsuo Gen and R. Cheng, Genetic Algorithms & Engineering Design, *A Wiley-Interscience Publication*, 1997

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

Hyun Ju Park

*Dept. of Electrical Eng. Graduate School
PuKyong National University*

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. We tuned the gains of PI controller using real-coding genetic algorithm. To evaluate the proposed method's effectiveness, we apply the proposed RGA-PI controller to the computer simulation and the speed control of an actual DC servomotor system.

The computer simulation and experimental results show that RGA-PI controller has the better control performance than PI controller in terms of settling time, rising time and overshoot toward the reference speed used in tuning RGA-PI controller. To evaluate the robustness of RGA-PI controller, we test over a new reference speed and changing load not used when tuning. RGA-PI controller shows the better control performance than PI controller.

감사의 글

뜻이 있는 곳에 길이 있고 노력하는 자만이 원하는 바를 얻을 수 있다는 평범한 진리와 함께 항상 희망과 용기를 주시고 오늘까지 성심과 열의로써 지도하여 주신 박한석 지도교수님께 먼저 깊은 감사를 드립니다.

본 논문의 심사에서 충고와 격려를 아끼지 않으시고 세심하게 지도하여 주신 홍순일 교수님, 김인동 교수님, 그리고 그동안 가르침을 주신 전기공학과 모든 교수님께도 감사를 드립니다.

본 논문이 완성되기까지 많은 도움을 주신 부산대학교 황기현 선생님께 진심으로 감사를 드리며, 언제나 제가 나태해 지려고 할 때 정신적인 도움을 많이 주신 김동완 교수님, 마음을 열고 다가와 충고를 아끼지 않으시던 남정락 교수님, 김준홍 교수님, 이종갑 선배님, 논문을 적으면서 시간에 얽매어 있을 때 시간적인 여유를 주신 동명대학 전기전자 교수님, 바쁜 학과 업무에 시달리면서도 많은 도움을 준 손금선, 강성우, 최정희 조교선생님께도 진심으로 감사를 드립니다.

오늘이 있기까지 사랑과 희생으로 키워주신 부모님과 사위를 위해서 항상 기도해 주시는 장인, 장모님, 집안의 장남으로 살아오면서 집안의 대사를 도맡으시던 큰형님과 형수님, 그리고 여러 형님, 누나들과 이 기쁨을 함께 나누고 싶습니다. 언제나 불평 불만 없이 뒷바라지 해주던 사랑하는 제 아내와 딸 효진이에게 조그마한 이 한 권의 책을 드리고 싶습니다.

지금껏 살아왔듯 맡겨진 일에 최선을 다하며 자아발전을 위해서도 노력하겠습니다. 뒤에 오는 후학을 두려워하라는 옛말을 교훈 삼아 뒤처지지 않고 그들을 이끌어 줄 수 있는 이가 되도록 노력하며 위로 선배님들과 선학의 말씀에 귀기울여 타산지석을 삼으며, 사람들을 사랑할 줄 알며 윗사람을 섬길 줄 아는 자가 되겠습니다. 지금까지 격려해주신 모든 분들께 감사의 뜻을 전합니다.

2002년 7월 박 현 주