

실수코딩 유전알고리즘을 이용한 제어기 동조

주 상 래* · 진 강 규**

Controller Tuning Using a Real Coding Genetic Algorithm

Sang-Rae Joo* · Gang-Gyoo Jin**

〈 목 차 〉

Abstract

- | | |
|----------------|--------------------|
| 1. 서론 | 2. 3 제어 파라미터 설정 |
| 2. 실수코딩 유전알고리즘 | 3. P1 제어기의 파라미터 동조 |
| 2. 1 표현 | 4. 결 론 |
| 2. 2 유전 연산자 | * 참고문헌 |

Abstract

In many controller design problems, the performance of systems is generally described in terms of overshoot, rising time, settling time, maximum input and so on. Keeping pace with the requirements of systems creates complex optimization problems. In this paper, a real coding genetic algorithm which can solve nonlinear constrained optimization problems is presented and the optimum setting of its control parameters is explored. The paper also includes the application of the proposed algorithm to PI controller tuning.

1. 서 론

최근 산업현장의 제어시스템들은 더욱 정교해지고, 복잡해짐에 따라 제어기 설계 시 요구되는 설계사양 또한 엄격해지는 경향이 있다. 등식 또는 부등식 형태의 제약조건들을 만족하고 성능이 우수한 제어기를 설계하는 문제는 복잡한 비선형 최적화 문제가 된다. 이러한

* 한국해양대학교 제어계측공학과 대학원생

** 한국해양대학교 자동화·정보공학부 교수

최적화 문제로부터 해를 얻기 위해서는 유전알고리즘(Genetic Algorithm)과 같은 강인한 탐색기법이 필요하다.

전통적으로 유전알고리즘의 표현법은 이진코딩(Binary Coding)이었다⁴⁾. 이진코딩을 채용하는 기존의 유전알고리즘들은 광범위한 분야에서 성공적으로 이용되고 있지만 문제의 탐색공간이 대단히 크거나, 고정밀도의 해를 요구하거나, 복잡한 제약조건이 존재할 때는 잘 동작하지 않는다. 이를 해결하기 위한 방법으로 변수의 표현을 문제공간에 접근시키는 코딩법의 개선, 다른 형태의 염색체를 다룰 수 있는 연산자의 개선, 부적합한 해의 발생과 연관된 복구전략(Repairing Strategy)과 벌점전략(Penalty Strategy) 등이 폭 넓게 연구되고 있다⁶⁾.

따라서 본 연구에서는 제약조건을 갖는 최적화 문제를 해결할 수 있는 실수코딩 유전알고리즘을 제안한다. 여기서 문제영역의 탐색 시 요구특성에 신속히 부합하기 위하여 실수 표현을 채용하고, 실수형 염색체를 효율적으로 다룰 수 있는 연산자들을 사용한다. 유전알고리즘의 성능은 연산자들이 가지는 파라미터 값에 크게 영향을 받기 때문에 이들을 변경시키면서 측정된 성능평가를 통해 최적의 영역을 도출한다. 제안된 유전알고리즘은 벌점 함수 전략과 더불어 PI 제어기 파라미터를 최적 동조하는 문제에 응용하고, 그 유효성을 확인한다.

2. 실수코딩 유전알고리즘

2.1 표현

전통적으로 유전알고리즘의 표현법은 이진코딩이었다. 이 표현법과 더불어 사용된 기존의 유전 연산자들은 이론적 분석을 용이하게 하여 이 결과 유전알고리즘의 이론적 기초가 되는 스키마이론(Schema Theorem)이 나오게 되었다^{1,2,4)}. 기존의 유전알고리즘은 탐색영역이 크거나, 고정밀, 제약조건을 가지는 최적화 문제에는 적용하기가 어렵다. 흔히 해에 대한 사전지식이 결여될 경우에는 큰 탐색영역이 선택되고, 비록 탐색영역이 작다 하더라도 정밀도를 높이기 되면 염색체 길이가 길어지게 된다. 매우 긴 염색체는 대단히 큰 탐색공간을 만들게 되어 이로 인해 계산부담이 증가할 뿐만 아니라 경우에 따라 잘 동작하지 않는다.

기존의 이진코딩이 가지는 이런 문제점을 해결하기 위하여 제안된 한 방법이 실수코딩이다. 실수 표현법을 채용함으로써 염색체 벡터는 해 벡터와 같은 요소(변수)들로 구성된 벡터로 코딩된다. 이로서 문제공간에서 가까운 두 점은 표현공간에서도 역시 가깝게 된다. 실수코딩의 장점은 상당히 큰 정의영역을 표현할 수 있고, 정밀도는 계산환경에 따라

다르지만 일반적으로 이진코딩의 경우보다 훨씬 우수하다. 또한 제약조건을 다루기 위한 도구를 설계하는 것도 더 쉽게 된다⁶⁾.

k 세대(또는 반복횟수)에서 실수코딩 염색체 벡터(이하 간단히 염색체라 함)는

$$s(k) = (x_1(k) \ x_2(k) \ \dots \ x_n(k)) \quad (1)$$

와 같이 행벡터로 표시된다. 따라서 벡터의 차원 n와 염색체의 길이 l은 서로 일치하게 된다. 이것으로 부터 N개의 염색체로 구성되는 개체집단은

$$P(k) = \{s_1(k) \ s_2(k) \ \dots \ s_N(k)\} \quad (2)$$

으로 표시할 수 있다. 여기서 N은 집단의 크기이다.

2.2 유전 연산자

기존의 유전알고리즘과 마찬가지로, 제안되는 유전알고리즘도 재생산(Reproduction), 교배(Crossover) 및 돌연변이(Mutation)를 기본 연산자로 채용한다. 그림 1은 유전알고리즘의 동작과정을 보여주고 있다.

```

GeneticAlgorithm()
{
    k= 0;
    Initialize P(k);
    Evaluate P(k);
    While(not termination condition)
    {
        k= k+1;
        Reproduce P(k) from P(k-1);
        Crossover and mutate P(k);
        Evaluate P(k);
    }
}

```

그림 1 유전알고리즘의 동작과정

2.2.1 재생산 연산자

재생산 연산은 자연도태(Natural Selection) 현상을 모방하려는 인위적인 메카니즘이다. 이를 알고리즘 형태로 구현하는 연산자는 현 세대 집단 $P(k-1)$ 내의 개체들을 그들의 적합도 값에 따라 선택해서, 다음 세대의 교배급원 $P(k)$ 를 형성한다. 이 연산자는 확률적으로 적합도가 큰 개체들을 더 많이 선택함으로써 전체 집단을 강하게 해준다. 일반적으로 유전 알고리즘 사용자들이 폭넓게 이용하고 있는 룰렛휠(Roulette Wheel)에 기초한 재생산 연산자는 확률적 동작특성 때문에 강한 개체들이 여러 번 복제되어 집단내 유전적 다양성이 결핍되고, 최적 개체가 선택되지 못할 가능성이 있는 등 몇 가지 단점을 가진다. 본 연구에서는 이러한 단점을 보완할 수 있는 Pham과 Jin에 의해 제안된 알고리즘을 재생산 연산자로 이용하는데, 그 동작을 요약하면 다음과 같다^{5,7)}.

단계 1: 개체들의 적합도 $f_i(k-1)$ ($1 \leq i \leq N$)을 계산하고, 최대 적합도 $f_b(k-1)$ 을 가지는 개체 $x_b(k-1)$ 을 구한다. 여기서

$$f_b(k-1) = \max_{1 \leq i \leq N} [f_i(k-1)] (> 0) \quad (3)$$

단계 2: 각 개체에게 새로운 값들을 할당한다.

$$x_{ij}(k) = x_{ij}(k-1) + \eta_i \frac{[f_b(k-1) - f_i(k-1)]}{f_b(k-1)} [x_{bj}(k-1) - x_{ij}(k-1)] \quad (4)$$

$(1 \leq i \leq N, 1 \leq j \leq n)$

여기서 $s_i(k) = (x_{i1}(k) \ x_{i2}(k) \ \dots \ x_{in}(k))$ 이고, 재생산 계수 η_i 는 0과 2 사이에서 적절히 선택되는 양의 값이다^{5,7)}.

단계 3: 새롭게 구성된 염색체로부터 교배급원 $P(k)$ 를 형성한다.

기존의 연산자와는 달리 이 연산자는 재생산동안 최적 개체의 생존을 보장하나 이후의 교배와 돌연변이 연산에서 파괴될 수도 있다. 집단내 최적의 개체가 소멸되지 않고 다음 세대로 전달되는 것을 보장하는 것이 엘리트 전략(Elitist Strategy)이다. 일반적으로 엘리트 전략은 유전알고리즘의 성능을 개선해 주는 것으로 알려졌다.¹⁾ 따라서 본 연구에서도

엘리트 전략을 채용하며, 만약 이전 세대의 최적 개체를 저장하고 있다가 이것이 현 세대에서 파괴되면 현 세대에서 가장 약한 개체와 교환해준다.

2.2.2 교배 연산자

교배 연산은 자연계 생물들의 성적 재생산을 알고리즘 형태로 흉내낸 것이다. 탐색공간상의 새로운 점을 찾기 위하여 교배급원으로부터 한 쌍의 아버지 염색체를 임의로 선택하고, 임의로 선택된 교배점 이후의 유전인자들을 서로 교환·결합함으로써 자손을 생성한다. 이러한 연산은 생성된 자손의 수가 집단의 크기와 같을 때까지 반복된다.

기존의 이진 표현법에서는 특정 변수에 대응되는 스트링을 직접 절단하는 것이 가능하였으나, 실수 표현에서는 이것이 불가능하다. 따라서 실수코딩 유전알고리즘의 교배는 교배점에서의 요소는 일차결합 형태로 연산하고, 이후의 요소들은 기존의 방법과 같이 서로 교환한다. 채용한 교배 연산자는 다음 단계를 거쳐 동작된다.

단계 1: 교배급원으로부터 아버지의 염색체 쌍을 임의로 선택한다.

$$s_v(k) = (x_{v1}(k) \cdots x_{vi}(k) x_{v,i+1}(k) \cdots x_{vn}(k)) \quad (5a)$$

$$s_w(k) = (x_{w1}(k) \cdots x_{wi}(k) x_{w,i+1}(k) \cdots x_{wn}(k)) \quad (5b)$$

$$\text{단, } v, w \in [1, N]$$

단계 2: 난수 $r \in [0, 1]$ 을 발생시켜 만약 $r \leq P_c$ (교배확률)이면, $[1, n]$ 사이의 교배점 i 가 발생되고 교배된다. 교배점에서의 두 요소는 일차결합 형태로 교배되고 이후 요소들은 서로 교환된다.

$$s'_v(k) = (x_{v1}(k) \cdots x_{wi}(k) x_{w,i+1}(k) \cdots x_{vn}(k)) \quad (6a)$$

$$s'_w(k) = (x_{w1}(k) \cdots x_{vi}(k) x_{v,i+1}(k) \cdots x_{wn}(k)) \quad (6b)$$

$$\text{단, } x_{wi}(k) = \lambda x_{vi}(k) + (1-\lambda) x_{wi}(k)$$

$$x_{vi}(k) = \lambda x_{wi}(k) + (1-\lambda) x_{vi}(k)$$

이 때 λ 값에 따라 서로 다른 형태의 교배가 기대되나, 여기서 λ 는 $[0, 1]$ 사이의 난수이다.

단계 3: 교배된 자손의 수가 N 이 될 때까지 단계 1, 2를 반복한다.

2.2.3 돌연변이 연산자

기존의 재생산과 교배는 세대가 진행되는 동안 집단을 강하게 해주지만, 탐색도중 지역 해나 사점(Dead Corner)에 빠지게 될 경우에는 이로부터 벗어나게 할 수는 없다. 이러한 원치 않는 해로부터 벗어나기 위한 한 방법으로 돌연변이가 이용된다. 기존의 돌연변이 연산자는 집단내 염색체의 유전인자들을 돌연변이확률 P_m 에 따라 변경시키는데, 각 유전인자의 돌연변이 발생유무는 P_m 에 따라 결정된다. 실수코딩의 경우에도 탐색영역을 더욱 확대하기 위하여 돌연변이가 요구되며, 이 경우에 기존의 연산자와는 다른 형태가 되어야 한다. 본 연구에서는 특정 요소에 돌연변이가 발생되면 그 요소의 정의영역 하한치와 상한치 사이에서 임의로 발생된 값으로 대체하는 방법을 사용하였다. 그 구체적인 연산은 다음 3 단계로 실행된다.

단계 1: 염색체에서 한 요소를 선택한다(순차적으로).

단계 2: 난수 $r \in [0, 1]$ 을 발생시켜 만약 $r \leq P_m$ 이면 선택한 요소에는 돌연변이가 일어나고 그 결과는 다음과 같게 된다:

$$s'_{vj}(k) = (x_{v1}(k) \cdots x'_{vj}(k) \cdots x_{vm}(k)) \quad (7)$$

여기서 $x'_{vj}(k)$ ($1 \leq j \leq n$)는 j 번째 요소 $x_{vj}(k)$ 의 정의영역 $[l_j, u_j]$ 사이에서 발생된 난수이다.

단계 3: 집단내 모든 염색체들의 요소에 대해 같은 연산이 행해질 때까지 단계 1, 2를 반복한다

2.3 제어 파라미터 설정

다른 최적화 알고리즘과 같이 실수코딩 유전알고리즘도 해의 정도와 탐색성능에 영향을 미치는 여러 제어 파라미터(Control Parameter)를 가지고 있는데, 집단크기(N), 재생산 계수(η_i), 교배확률(P_c), 돌연변이확률(P_m) 등이 여기에 속한다. 이들을 적절히 설정하는 것은 주어진 환경에서 최적의 성능을 얻고 또한 해가 원치 않는 쪽으로 수렴해 가는 것을 방지하기 위하여 대단히 중요하다. 이들을 최적으로 설정하기 위한 일의적 방법은 없으며, 대신 경험과 실험에 의존함으로써 어느 정도 합당한 결과를 유도해 낼 수 있다. 이러한 결과는 적절한 성능측정 수단을 기초로 하여 얻어진다^{1,3,5,7,8)}

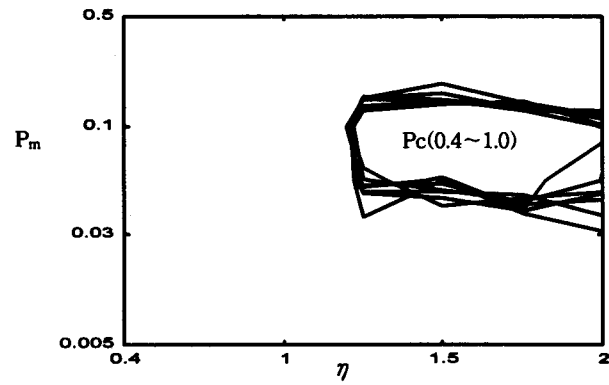
특히 집단의 크기는 유전알고리즘의 효율적인 동작에 크게 영향을 끼치는 중요한 변수

이다. 어느 범위내에서 이를 크게 하면 할수록, 유전적 다양성이 증가하여 수렴속도가 빨라지는 경향이 있지만, 집단크기에 비례해서 해를 얻기까지는 많은 연산시간이 소모되기 때문에 계산부담과 수렴속도 사이에 적절한 타협점이 요구된다. 재생산 계수 역시 유전알고리즘의 성능에 영향을 주는 변수로서, $[0, 2]$ 사이에서 안정적인 동작을 보장한다⁵⁾. 교배확률은 교배연산이 일어나는 빈도를 조절하게 되는데 이로써 전역탐색의 양이 조정된다. 일반적으로 교배확률이 낮게 설정되면 새로운 개체 발생이 적게 되어 탐색이 침체되며, 반면에 높게 설정되면 탐색공간을 더 빨리 탐색하는 경향을 갖지만 더 나은 개체를 생성하는 것보다 더 빨리 좋은 개체를 파괴하게 된다. 돌연변이를 통해 제공되는 전역탐색의 양은 돌연변이확률에 영향을 받게 된다. 만약 돌연변이확률을 낮게 설정하면 특별한 초평면(Hyperplane)을 지역탐색 하게 되고, 반대로 너무 높게 설정하면 탐색은 한 점에서 다른 점으로 불규칙적으로 점프하게 되어 불안정한 탐색 결과를 주게된다.

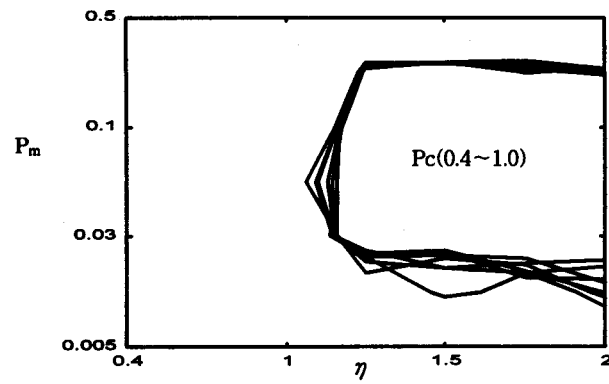
따라서 본 연구에서 최적의 제어 파라미터 집합을 얻기 위하여 앞서 수행한 연구 결과를 참조하여 제한된 범위에서 제어 파라미터들을 단계적으로 변경하면서 오프라인 전역강인성(Offline Global Robustness)을 계산하였다^{1,3,7,8)}. 이 때 집단크기는 $N \in \{10, 20, 30\}$ 3단계, 재생산 계수는 $\eta_i \in \{0.4, 0.7, 1.0, 1.3, 1.5, 1.7, 2.0\}$ 7단계, 교배확률은 $P_c \in \{0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$ 7단계, 돌연변이확률은 $P_m \in \{0.005, 0.01, 0.03, 0.05, 0.1, 0.3, 0.5\}$ 7단계로 선정하였다.

이러한 파라미터들의 변화에 대한 전역 강인성은 De Jong의 벤치마크 함수들¹⁾을 이용하여 계산되었고, 특히 함수 F_4 의 경우는 정규잡음을 고려하지 않았다. 유전알고리즘의 성능이 초기 집단의 선택에 좌우되는 것을 막기 위하여 각각 다른 seed값에 대해 프로그램을 10번 수행하였고, 결과치들을 각각 평균하였다.

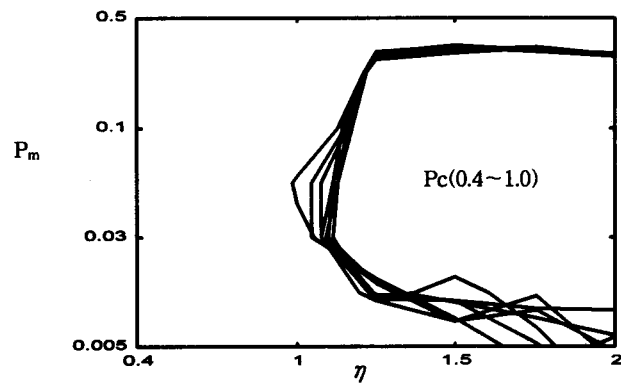
그림 2는 서로 다른 N 에 대해 유전알고리즘의 최적 파라미터 영역을 보여 주고 있다. 여기서 여러 등고선들은 서로 다른 P_c 에 대응된 그림이고, 한 등고선상의 모든 파라미터 쌍은 같은 강인성(97%)을 가지며, 더 나은 강인성은 안쪽 내부에서 찾을 수 있다. 그림에서 보면 η_i 는 1.3과 2사이에 최적 영역이 있고, P_c 는 클수록 최적 영역이 확대되는 경향을 갖지만 크게 영향을 주지 못한다. P_m 은 N 에 따라 최적 영역이 확대됨을 알 수 있는데, N 이 크면 낮은 돌연변이확률에도 최적값을 찾을 수 있었다. 비록 제한된 실험을 통해 얻어진 결과이지만 제안한 실수코딩 유전알고리즘은 N 에 관계없이 $1.4 \leq \eta_i \leq 1.9$ ($1 \leq i \leq N$), $0.5 \leq P_c \leq 1.0$, $0.05 \leq P_m \leq 0.1$ 범위내의 모든 파라미터 쌍에 대해 만족스러운 탐색성능을 얻을 수 있었다.



(a) $N=10$



(b) $N=20$



(c) $N=30$

그림 2 최적 제어 파라미터 영역

3. PI 제어기의 파라미터 동조

제안한 실수코딩 유전알고리즘의 성능을 확인하기 위하여 몇 가지 설계사양이 요구되는 제어시스템(그림 3)에서 PI제어기의 파라미터를 최적 동조하는 문제를 선정하였다. 이때 2차계의 플랜트는 $\zeta=1.7$, $\omega_n=1$, $k=1$ 로 선정하였다. 시스템 응답특성들의 제약조건들을 사전에 지정하고 시스템의 응답이 이 조건에 부합토록 유전알고리즘에 제약조건들을 부가하여 비례이득(K_P)과 적분시간(T_I)를 조정하였다.

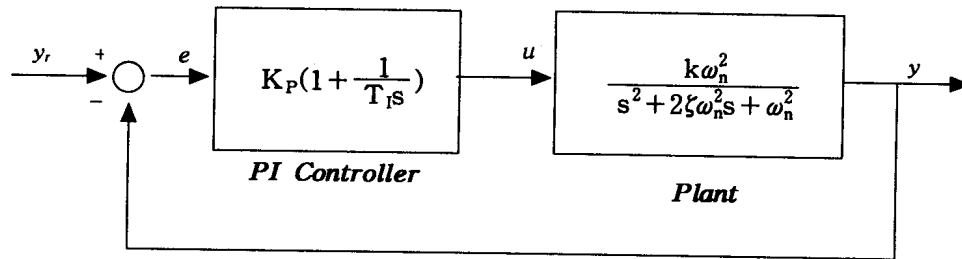


그림 3 PI 제어시스템

이 때 사용된 성능지수(목적함수)와 제약조건, 즉 최대오버슈트(M_p), 상승시간(t_r), 정정시간(t_s), 최대입력(u_{max})은 다음과 같다.

$$J(\mathbf{x}) = \int_0^{\infty} (ae^2 + u^2) dt \quad (8)$$

$$M_p(\mathbf{x}) \leq 10\%, t_r(\mathbf{x}) \leq 1.2 \text{ sec}, t_s(\mathbf{x}) \leq 3 \text{ sec}, u_{\max}(\mathbf{x}) \leq 5$$

$$\Omega = \{ \mathbf{x} \mid \mathbf{x}^{(L)} \leq \mathbf{x} \leq \mathbf{x}^{(U)} \}, \mathbf{x} = [K_P \ K_I]^T$$

여기서, Ω 는 탐색공간이고, \mathbf{x} 는 탐색되는 파라미터 벡터이다.

이상은 제약조건을 가지는 비선형 최적화 문제로서 해가 제약조건을 위반할 때 이를 처리하기 위한 방법이 필요하다. 따라서 해가 제약조건을 위반하면 그 위반 정도에 따라 목적함수에 벌점을 부가하도록 벌점함수 $p(\mathbf{x})$ 를 사용하였다. 이렇게 함으로써 주어진 문제는 제약조건이 없는 최적화 문제로 변환된다.

$$F(\mathbf{x}) = J(\mathbf{x}) + p(\mathbf{x}) \quad (9)$$

$$p(\mathbf{x}) = \begin{cases} 0 & , \text{if } \mathbf{x} \text{가 적합한 해일 때} \\ w_1(0.1 - M_p(\mathbf{x}))^2 + w_2(1.2 - t_r(\mathbf{x}))^2 + \\ w_3(3 - t_s(\mathbf{x}))^2 + w_4(5 - u_{\max}(\mathbf{x}))^2 & , \text{otherwise} \end{cases}$$

여기서 w_1, w_2, w_3, w_4 는 벌점상수이다.

이로부터 해를 얻기 위해 유전알고리즘의 제어 파라미터로 $N = 10, \eta_i = 1.7, P_c = 0.7, P_m = 0.1$ 을 사용하였고, 벌점상수로 $w_1 = 10, w_2 = w_3 = w_4 = 1$ 을 선정하였다.

그림 4는 파라미터의 동조 과정을 보여주고 있다. 그림에서 보면 약 400세대에서 최적 파라미터 $K_p = 4.518, T_i = 4.146$ 을 찾을 수 있었다.

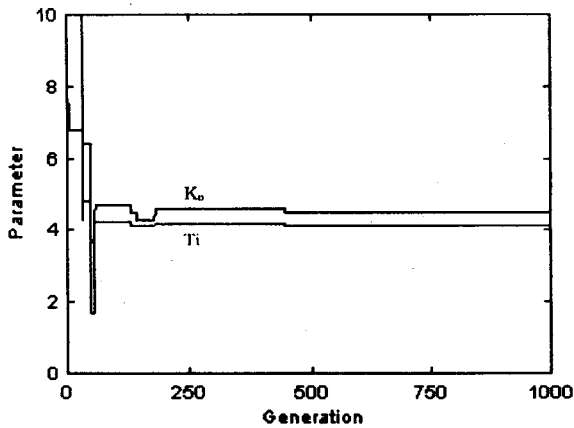
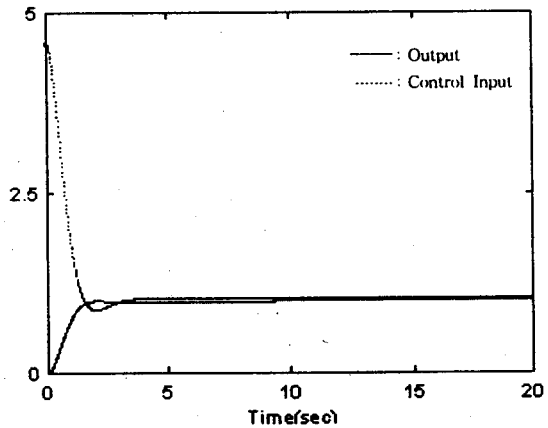


그림 4 파라미터의 동조 과정그림



5 PI 제어시스템의 계단 응답

그림 5는 최적으로 동조된 PI 제어시스템의 계단 응답을 그린 것이다. 그림에서 확인할 수 있듯이 제어시스템의 성능은 $M_p = 0.1\%, t_r = 1.12\text{sec}, t_s = 1.64\text{sec}, u_{\max} = 4.56$ 이 되어 주어진 설계사양을 만족하였다.

4. 결 론

본 연구에서는 여러 분야에서 발생하는 제약조건을 갖는 비선형 최적화 문제를 해결할 수 있도록 실수코딩 유전알고리즘을 제안하였다. 제안한 유전알고리즘은 실수형 염색체를 다루는 특수 연산자들을 기본 연산자로 채용한다. 유전알고리즘의 성능은 제어 파라미터에 좌우되기 때문에 오프라인 전역 강인성 평가를 통해 최적의 영역을 도출하였다. 제안한 유전알고리즘을 PI 제어기의 최적 동조 문제에 응용한 결과 만족스러운 결과를 얻을 수 있었다.

참 고 문 헌

- [1] K. A. De Jong, "An analysis of the behavior of a class of genetic adaptive systems," Doctoral Dissertation, University of Michigan, 1975.
- [2] D. E. Goldberg, Genetic algorithms in search, optimization, and machine learning, Addison-Wesley, 1989.
- [3] J. J. Grefenstette, "Optimization of control parameters for genetic algorithms," IEEE Trans. Syst. Man Cybern., Vol. SMC-16, pp. 122-128, 1986.
- [4] J. Holland, Adaptation in natural and artificial systems, University of Michigan Press, 1976.
- [5] G. Jin, "Intelligent fuzzy logic control of processes with time delays," PhD Thesis, University of Wales Cardiff, 1996.
- [6] M. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag, Berlin Heidelberg, 1992.
- [7] D. T. Pham and G. Jin, "A Hybrid Genetic Algorithm," Proc. of the 3rd World Cong. on Expert Systems, Seoul, Vol. 2, pp. 748-757, 1996.
- [8] J. D. Schaffer et al., "A study of control parameters affecting online performance of genetic algorithms for function optimization," Proc. of the 3rd Int. Conf. on Genetic Algorithms and Their Applications, Arlington, VA, pp. 51-60, 1989.

