論文 한국정밀공학회지 제14권 제9호 (1997년 9월)
Journal of the Korean Society of Precision Engineering Vol. 14. No. 9. September. 1997.

유전알고리즘을 이용한 유압모터의 속도제어파라메터 최적화

현 장 환*, 안 철 현*, 이 정 오*

Optimization of Control Parameters for Speed Control of a Hydraulic Motor Using Genetic Algorithms

Jang-Hwan Hyun*, Chul-Hyun Ahn*, Chung-Oh Lee*

ABSTRACT

This study is concerned with the optimizing method of control parameters for a hydraulic speed control system by using genetic algorithms which are general purpose search algorithms based on natural evolution and genetics. It is shown that the genetic algorithms satisfactorily optimized control gains of the PI speed control system of an electrohydraulic servomotor and that optimization of control parameters can be achived without much experience and knowledge for tuning. It is also shown that optimal gains may be determined from fitness distribution curves plotted in given gain spaces.

Key Words: Genetic algorithm(유전알고리즘), Electrohydraulic servomotor(전기유압식서보모터), Speed control(속도제어), PI controller(비례-적분 제어기), Control parameters(제어파라메터), Optimization(최적화)

1. 서 론

최근에 유압시스템의 성능을 향상시키기 위해서 제어파라메터를 최적화하는 연구가 관련분야 연구자들의 많은 관심을 받고 있다. 일반적으로 유압시스템의 제어에 흔히 이용되는 PID나 상태궤환제어기를 사용하는 경우에, 제어파라메터를 이론적으로 계산해내는 것은 매우 어렵다[1,2]. 비선형성이 강한 유압시스템을 정확하게 모델링하기가 쉽지 않기 때문이다[3,4]. 따라서 유압시스템을 제어할 때 일반적으로 제어파라메터는 조작자의 파라메터 조절에 대한 경험과 지식을 바탕으로 하여 많은 실험을 통해서 조정된다.

본 연구에서는 유압시스템의 제어파라메터를 최적화하기 위한 수단으로서 유전알고리즘(GA : Genetic Algorithm)을 이용하였다. 유전알고리즘은 자연계의 적자생존과 유전학에 근거를 둔 탐색알고리즘이다[5,6]. 제어성능을 반영하는 적절한 성능지수가 생존능력으로 주어지면 조작자에 의해 설정되는 탐색영역내에서 제어시스템에

* 한국과학기술원 기계공학과

대한 최적의 제어파라메터가 탐색될 수 있다[7,8,9]. 이러한 유전알고리즘을 이용하여 유압모터의 속도제어시스템에 사용된 PI제어기를 대상으로 제어파라메터를 최적화하는 연구를 수행하였다. 특히 대상제어기에 대한 조작자의 제어파라메터 조절경험을 이용하지 않고 시스템에 대한 최소한의 정보만을 이용하여 제어파라메터를 최적화할 수 있음을 보인다.

2. 유압시스템

Fig.1은 이 연구에서 사용한 유압서보모터시스템의 개략도이며 이러한 시스템은 공작기계, 산업용 Manipulator등의 정밀구동장치로 응용되고 있다.

실험장치는 유압원(Power unit), 서보밸브(Servovalve), 엔코더(Encoder), 그리고 제어용 인터페이스 카드(Interface card)를 내장한 컴퓨터로 구성되었으며 제원은 다음과 같다.

Parts/Instruments	Company/Model number	Specifications
Power unit	◦	Supply pressure : 1000 *psi*
Hydraulic motor	Moog/A084	Displacement : 0.506 in^3 / *rev*
Servovalve	Moog/73-102	Rated flow : 5 *gpm* (at 1000 *psi*)
Encoder	Sansei/OES-10-2M	Resolution : 1000 (*pulse* / *rev*)
Interface card	Nanotech/Labinmaster	12bit D/A(±10 volt) & 16bit Counter

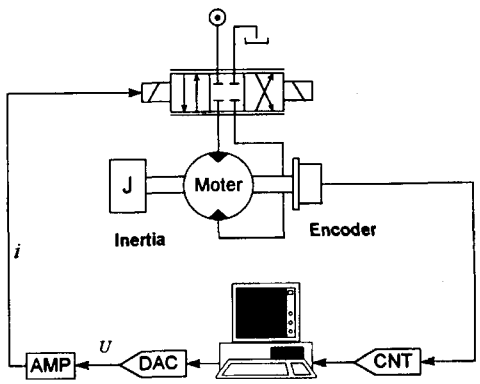

Fig. 1 Schematic diagram of the hydraulic motor control system

3. 유전알고리즘

3.1 유전알고리즘을 이용한 제어파라메터의 탐색

유전알고리즘을 이용하여 제어파라메터를 최적화하는 과정은 Fig.2와 같으며 각 블럭의 기능 및 원리는 아래와 같다.

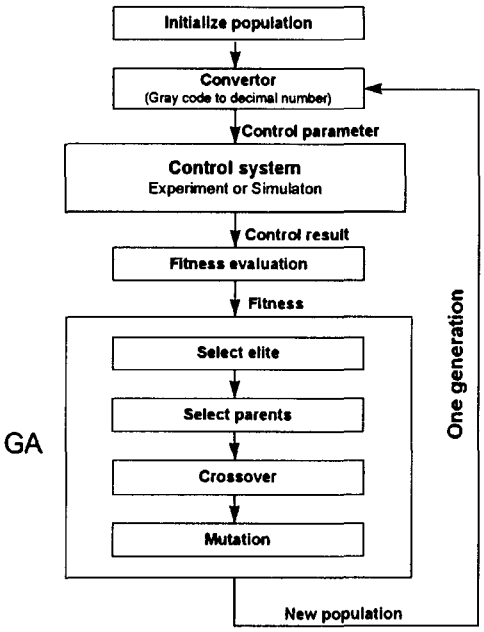

Fig . 2 Procedure of optimizing control parameters with GA

(1) Initialize population

이 연구에 사용된 각 개체(Chromosome)의 데이터구조는 다음과 같다. 유전알고리즘에 사용되는 2진수(Gray code and Binary code)와 실제 제어기에 사용되는 10진수(Decimal number)로 표현되는 각 제어파라메터값 그리고 생존적합성(Fitness) 및 각 세대에서의 생존적합성의 순위(Rank)를 포함하는 구조로 되어있다. 프로그래밍은 C언어를 사용하였으며 구조체(Structure)를 이용하여 다음과 같이 나타내었다.

```
struct Chromosome{
        int rank;
        int bin[bit1+bit2];
        int gray[bit1+bit2];
        double par[2];
        double fitness;}
```

여기서 bit1과 bit2는 2진수로 표현되는 개체에서 각 제어파라메터가 차지하는 부분의 길이를 나타낸다.

'Initialize population' 블럭에서는 개체의 데이터에서 Gray code로 표현되는 부분을 Fig.3과 같이 만들어 낸다. 여기에서 0과 1로 표현되는 각 bit(g_{ij})의 값은 무작위(Random)하게 선택된다.

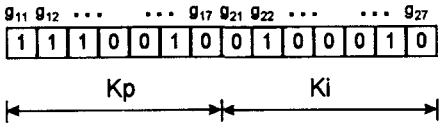

Fig. 3 Data structure of chromosome

(2) Convertor

이 과정에서는 Gray code부분의 데이터를 이용하여 제어시스템에 사용되는 십진수로 표현되는 제어파라메터를 만든다. Gray code는 먼저 Binary code로 변환된 후 십진수로 변환된다.

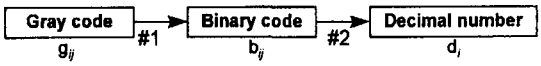

Fig. 4 Schematic of converting process

#1 과정

$$b_{ij} = g_{ij} \qquad at \quad j = j_{\max}$$

$$b_{ij} = XOR(b_{ij+1}, g_{ij}) \qquad for \quad j = [1, j_{\max} - 1]$$

#2 과정

$$d_i = \sum_{j=1}^{j_{max}} b_{ij} \cdot 2^{(j_{max}-j)}$$

(3) Control system

매 세대마다 PI제어기를 이용한 속도제어실험을 한 세대의 개체수만큼 반복하여 수행한다. 제어기의 구조는 아래와 같으며 궤환(Feedback)되는 속도신호는 엔코더신호를 수치미분하여 구하였으며, 샘플링간격은 0.01초이다.

$$U = Kp \cdot (V_d - V) + Ki \cdot \int (V_d - V) dt$$

여기서, U : *control input(Volt)*

V_d : *Desired motor velocity(rad/sec)*

V : *Motor velocity(rad/sec)*

Kp : *Proportional gain of PI controller*

Ki : *Intergral gain of PI controller*

Convertor를 통해 주어지는 개체수만큼의 제어파라메터조합을 이용하며 매 실험마다 제어결과를 저장한다. 제어결과는 유전알고리즘의 생존적합성을 계산하는데 필요한 정보를 저장한다. 여기서는 시간과 모터의 속도가 제어결과로서 저장된다.

(4) Fitness evaluation

유전알고리즘을 제어파라메터의 최적화에 사용하는 경우에 있어서는, 각 세대에서 개체에 해당하는 각 제어파라메터조합이 다음세대에 살아남을 생존적합성을 평가하기 위한 기준이 필요하다. 일반적으로 제어성능을 평가하는 기준으로 많이 쓰이는 ISE, ITSE, IAE, ITAE(I:integration, S:square, A:absolute, T:time-multiplied, E:error)등의 역수를 생존적합성을 나타내는 값으로 사용한다. 본 연구에서는 이들 중 ITAE를 이용하였으며 그 역수값을 생존적합성으로 정하였다. 제어시스템에서 저장된 제어결과를 이용하여 각 개체들의 Fitness를 구하고, 유전알고리즘에서 이용할 수 있도록 이를 저장한다.

$$Fitness = \frac{1}{ITAE} = \frac{1}{\int t \cdot |e(t)| \cdot dt}$$

(5) Select elite

이 과정에서는 Fitness의 크기를 기준으로 한 세대에서 주어진 개체들의 서열을 결정한다. 그리고 가장 큰 Fitness를 가진 개체를 Elite로 결정한다. Elite로 결정된 개체는 유전연산을 거치지 않고 다음 세대에 현 세대의 개체특성을 그대로 유지한다.

(6) Select parents

교배(Crossover)가 될 개체를 선택하는 과정으로서, Fitness value에 의하여 그 선택확률이 결정되는 Weighted roulette wheel방법을 이용하여 무작위로 선택한다. Elite를 제외한 개체수만큼의 선택이 이루어지며 매 선택에서 임의의 한 개체가 다음세대에 부모로 선택될 확률(%)은 다음과 같다.

$$\% = \frac{\text{임의 개체의 Fitness}}{\text{모든 개체의 fitness의 총합}}$$

(7) Crossover

선택된 한쌍의 부모들의 개체를 이용하여 개체의 임의 위치에서 유전자를 절단하여 부모가 서로 교환한다. 선택된 부모들 중에서도 (개체의 수)×(Crossover rate)만큼의 연산이 이루어지며, Elite와 Crossover rate확률에 의해 선택되지 않은 개체들은 Crossover연산을 거치지 않고 다음 세대에서 현 세대의 형질을 그대로 유지한다. Crossover 연산은 아래 그림과 같이 수행된다.

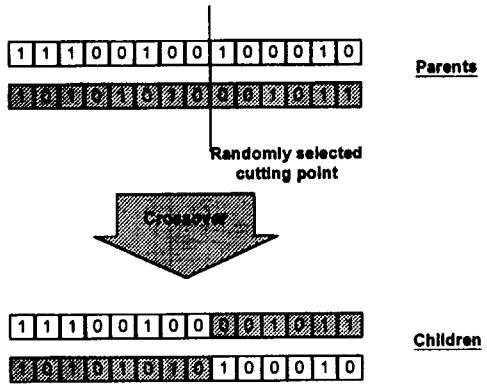

Fig. 5 Schematic of crossover process

(8) Mutation

Crossover에 의해서 만들어진 Elite를 제외한 개체들의 각 Bit에 대해서 Mutation rate만큼의 연산이 이루어진다. Mutation rate에 해당하면 해당 Bit의 값를 0인 경우는 1로, 1인 경우는 0으로 바꾸어준다. 즉 한 세대에 대해서 (개체의 수)×(개체의 길이)×(Mutation rate)만큼의 연산이 수행된다.

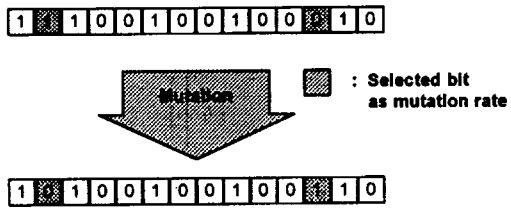

Fig. 6 Schematic of mutation process

3.2 유전알고리즘 파라메터

유전알고리즘의 탐색성능을 좋게 하기 위해서는 다음과 같은 유전알고리즘의 파라메터를 적절하게 선택해야 한다.

(1) Population size and string length

유전알고리즘의 탐색성능과 탐색효율에 영향을 미치는 인자들이다. 확률적으로 30이상의 Population size를 사용하는것이 일반적이지만 탐색대상이 제어게인일 경우에는 이 값이 실험회수와 관계가 있기 때문에 되도록 작은 값을 선택해야 한다. 그러나 너무 작은 값을 취하게 되면 충분치 못한 탐색으로 탐색성능이 매우 나빠진다. 따라서 실험회수와 탐색성능 두 조건을 모두 적절히 만족시킬 수 있는 값을 취한다.

일반적으로 유전알고리즘을 도입하는데 있어서 요구되는 조건으로 초기의 Population으로부터 Crossover만을 통하여 탐색영역의 모든 지점에 도달할 수가 있어야 한다. 이에 대한 일반적인 증명은 Reeve에 의해 이루어졌다[10]. Reeve는 String length가 20 인 경우에 있어서, Population size가 10일 때 초기의 Population에 하나의 유전인자(0 or 1)가 최소한 한번이상 나타날 확률이 95% 이상이 되고 이때 위의 조건이 만족됨을 보였다.

따라서 이 논문에서는 Population size는 10 그리고 String length는 14를 취하였다.

(2) Crossover rate

이 값은 클수록 탐색속도를 크게 하지만 너무 큰 값의 경우에는 좋은 성능을 낼 수 있는 구조의 개체가 다른 개체들을 발전시키기 이전에 사라질 수도 있다. 일반적으로 0.7 - 0.9의 값을 사용하며 이 논문에서는 0.7을 선택하였다.

(3) Mutation rate

이 값은 Population의 다양성을 높여 주는 제2의 탐색연산자이다. 그러나 너무 큰값이 되면 방향성 없는 무작위탐색이 된다. 일반적으로 0.01 - 0.05의 값을 사용하며 이 논문에서는 0.02를 선택하였다.

(4) 탐색공간

탐색공간의 선정은 유전알고리즘을 이용하여 제어파라메터를 선정하는 문제에 있어서 매우 중요한 부분이다. 탐색공간은 아래의 사항을 고려하여 시행착오를 거쳐서 결정된다.

첫째는 시스템의 안정성을 고려해서 선택을 해야한다. 즉 초기에 선택된 탐색공간이 시스템을 불안정하게 하는 영역인 경우 대부분의 개체가 제어불안정을 일으키고 시스템에 손상을 입히게 된다. 따라서 조작자가 대상시스템에 대한 제어파라메터 조정경험이 없는 경우에는 시스템의 정보를 이용하여 초기 탐색공간을 결정한다. 즉 탐색공간상의 최대 제어파라메터값이 적용되었을때 제어초기에 발생하는 큰 오차에 의해 제어기에 포화(Saturation) 현상이 발생하지 않도록 한다.

둘째는 충분한 탐색공간을 제공해야한다. 비록 시스템이 안정된 상태의 실험을 통해 제어파라메터를 탐색해나가는 경우라 할지라도 탐색의 수렴이 탐색공간의 경계영역에서 이루어지는 경우에는 경계영역을 확장하여 유전알고리즘에 충분한 탐색공간을 제공하도록 한다.

(5) 기타

그 밖에 유전알고리즘의 탐색성능을 향상시키기 위해서 Gray code와 Elitism을 사용하였다. Gray code는 일반 *Binary code*에 비해서 통계적인 Hill-climbing능력이 우수하다고 알려져 있으며 Elitism은 각 세대에서 가장 우수한 개체의 형질이 무작위선택과 교배에 의해서 퇴화할 가능성을 배제시켜준다[11].

4. 응용 및 결과

4.1 유전알고리즘에 의한 제어파라메터의 최적화

Fig.7의 (a)와 (b)는 유전알고리즘에 의해서 제어파라메터가 최적화되어가는 과정을 보여준다. Generation best는 각 세대에서 가장 높은 Fitness를 갖는 개체의 Fitness를 의미하며 Generation average는 각 세대에서의 평균 Fitness를 의미한다. Fig.7(a)로부터 약 7세대에 이미 최적화된 값으로 수렴함을 알 수 있다. 또한 Fig.7(b)로부터 약 10세대 이후가 되면 Generation best와 Generation average가 거의 같아짐을 알 수 있다. 이는 한 세대에서 대부분의 개체가 Generation best 값 근처에 있음을 의미하며, 따라서 탐색속도가 정체됨을 알 수 있다. 최적화를 위해서 수행된 총 실험횟수는 약 500회 정도로서 이는 탐색영역 설정을 위한 시행착오에 의한 실험을 포함한 횟수이다. Fig.8은 탐색된 제어파라메터에 의한 시스템의 속도제어성능을 나타낸다. Fig.8에서 보이는 오버슈트는 약간의 오버슈트를 최적으로 인식하는 ITAE를 성능지수로 사용했기 때문이다.

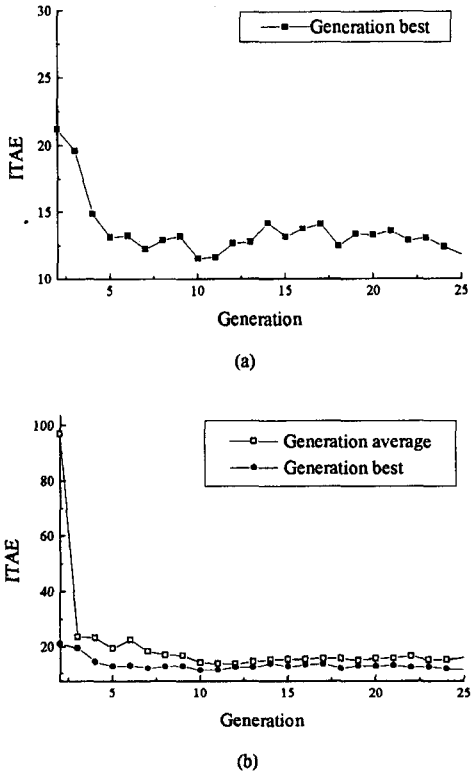

Fig. 7 Optimization of control parameters using GA

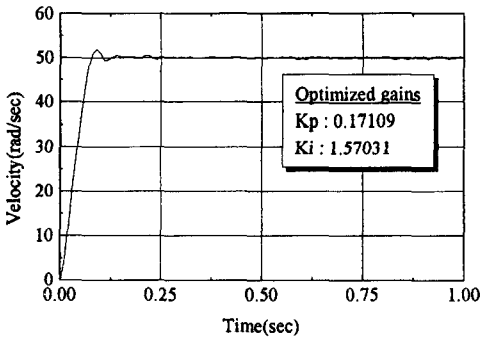

Fig. 8 Speed response with optimized gains obtained using GA, desired speed is 50 *rad/sec*..

4.2 탐색공간의 특성

유전알고리즘을 이용하여 탐색된 제어파라메터가 최적이라는 것을 보이기 위해서 설정된 탐색공간상에서 개체의 생존적합성을 나타내는 함수의 형태를 조사하였다.

Fig.9의 (a)와 (b)는 각각 Kp와 Ki로 구성되는 탐색영역에서의 생존적합성의 분포와 이를 투영하여 등고선

으로 나타낸 그림이다. 탐색된 제어파라메터가 탐색영역 내에서 최적임을 알 수 있다. Fig.9(b)에 표시한, Fitness가 낮게 분포되어 있는 세 영역에서는 다음과 같은 특성을 나타낸다.

Region 1 : 비례게인이 커서 제어가 불안정해지는 영역
Region 2 : 적분게인이 작아서 정상상태 오차가 큰 영역
Region 3 : 적분게인이 커서 과도한 오버슈트가 발생하는 영역

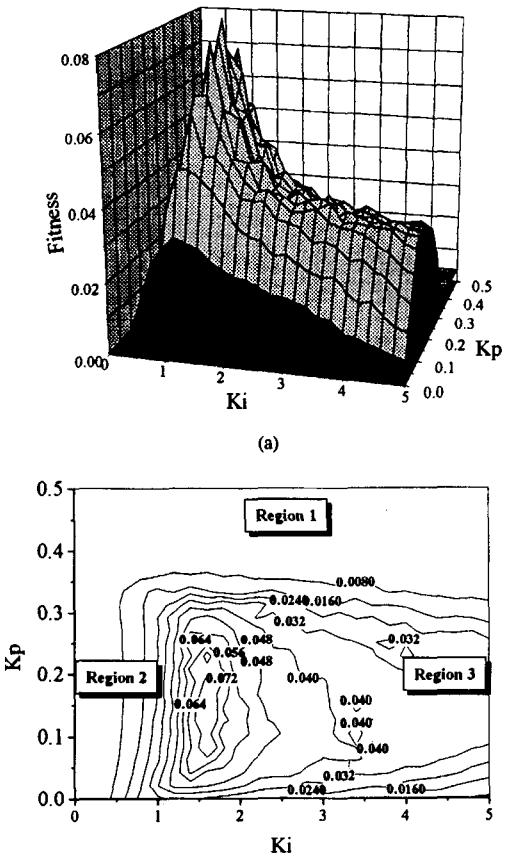

Fig. 9 Fitness distribution in Kp and Ki space

5. 결 론

본 연구에서는 유압시스템의 제어파라메터를 최적화하는 방법으로 유전알고리즘을 적용하였으며, PI제어기를 사용하는 유압모터의 속도제어시스템에서 유전알고리즘을 사용하여 적절한 실험횟수 이내에 만족할 만한 성능을 얻을 수 있는 제어파라메터를 탐색해 낼 수 있다. 최적화를 수행하는데 있어서 요구되는 정보는, 초기의 제어파라메터의 탐색영역의 설정에 필요한 제어기의 포화치에 대한 정보이고 초기설정된 탐색영역은 시행착오를 통하여 수정될 수 있다. 그리고 설정된 탐색공간 전역에 대한 생존적합성의 형태를 조사함으로써 유전알고리즘을 통해 탐색된 제어파라메터가 탐색공간에서 최적값임을 알 수 있다.

참 고 문 헌

1. H. E. Merrit, *Hydraulic Control Systems*, John Wiley & Sons Inc., New York, 1967.
2. J. Watton, *Fluid Power Systems*, Prentice Hall, New York, 1989.
3. J.H.Hyun and C.O. Lee, "A Study on the Position Control of Electrohydraulic Servosystem Using Adaptive Sliding Mode Control," J. of KSPE, Vol.11, No.6, pp. 143-157, 1994.
4. S.M.Chin, C.O.Lee and P.H.Chang, "An Experimental Study on the Position Control of an Electrohydraulic Servo System Using Time Delay Control," J. of Control Engineering Practice, Vol.2, No.1, 1994.
5. D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, New York, 1989.
6. P. Sutton and S. Boyden, "Genetic algorithms: A general search procedure," *American J. of Physics*, vol. 62, No. 6, pp. 549-552, 1994.
7. C. C. Ahn, "Optimization of Control Parameters for Fluid Power Sysems by Genetic Algorithms," KAIST, M.S. thesis, 1996.
8. I. K. Jeong, "A New Hybrid Genetic Algorithm and Its Application to Control," KAIST, M.S. thesis, 1994.
9. A. Varšek,T. Urbancic and B. Filipic, "Genetic Algorithms in Controller Design and Tuning," *Trans. on Sys., Man and Cybernetics*, vol. 23,

No. 5, pp. 1330-1339,1993.

10. C. R. Reeves, "Using Genetic Algorithms with Small Populations," in *Proc. of the Fifth Intern. Conf. on Genetic Algorithms*, pp. 92-99, 1993.
11. K. E. Mathias and D. Whitley, "Transforming the Search Space with Gray Coding," in *First IEEE Intern. Conf. on Evolutionary Computation*, pp. 513-518, 1994.