
로봇을 위한 진화 하드웨어 확장성 연구: Band-Pass Filter 사례 연구

Research on Scalability of Evolutionary Hardware for Robots: Band-Pass Filter Case Study

박현수,Hyunsoo Park*, 김경중,Kyungjoong Kim**

요약 아날로그 회로를 설계하기 위해서는 전문적인 기술이 필요하기 때문에, 진화연산을 이용한 자동설계가 많이 연구되었다. 일반적으로 설계해야 할 회로가 복잡해질수록 진화연산을 기반으로 하여 설계를 하는 것은 더욱 힘들어진다. 따라서 복잡한 회로를 설계하기 위한 방법에 대해 분석할 필요가 있다. 본 논문에서는 Low-pass filter 와 High-pass filter 에 비해서 상대적으로 복잡한 회로인 Band-pass filter 이용하여 실험을 하고 복잡한 회로를 설계하는 방법에 대해 분석한다.

Abstract Analog circuit design is a very difficult task. So there are many studies for automating analog circuit synthesis using genetic algorithm. But, designing more complex circuit using GA is much more difficult than doing simple circuits. In this paper, we study on band-pass filter cases which are more difficult than simple low-pass filter and high-pass filter to analyze the scalability issue in evolvable hardware.

핵심어: *Genetic Algorithm, Analog Circuit, Band-Path Filter*

이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업(2010-0012876) 및 뇌과학 원천기술개발사업(2010-0018948).

*주저자 : 세종대학교 컴퓨터공학과 e-mail: rex8312@gmail.com

**교신저자 : 세종대학교 컴퓨터공학과 교수; e-mail: kimkj@sejong.ac.kr

1. 서론

진화연산(Genetic Algorithm), 진화 프로그래밍(Genetic Programming) 등의 기법을 사용하여 아날로그 회로를 자동으로 설계하고자 하는 많은 연구가 있었다[1][2]. GA, GP 에서 사용한 방법은 방법은 다양한 회로를 임의로 생성한 뒤에 그 회로들을 평가하고, 그 중에 상대적으로 높은 성능의 회로를 기반으로 다시 다양한 회로를 생성하여 평가하는 것을 되풀이 한다. 이 과정을 반복하면, 마지막에는 좋은 성능의 회로를 설계할 가능성이 높아진다.

대부분 진화 연산에 기반하여 회로를 설계할 때는 회로가 작을수록 좋다. 고려해야 할 요소가 너무 많다면, 진화연산을 통해 최적의 설계를 찾기 힘들 수 있다. 따라서, 복잡한 회로를 자동으로 설계하기 위해서는 어떤 방법을 사용해야 효과적인지 분석할 필요성이 있다.

Filter 에는 Low-pass filter (LPF), High-pass filter (HPF), Band-pass filter (BPF), Band-stop filter (BSF)가 있다. BPF 와 BSF 는 LPF 나, HPF 에 비해서 상대적으로 복잡하다. 이러한 복잡한 회로를 자동으로 설계하는 방법은 크게 두 가지가 있다. 첫 번째 방법은 HPF 와 LPF 를 각각 자동으로 설계한 뒤에 두 필터를 조합하는 것이다. 이 방법의 단점은, LPF 와 HPF 를 각각 자동 설계하기 위해서 진화연산을 두 번 수행해야 하고, 설계된 회로도 2 배의 크기가 될 수 있다는 것이다. 하지만, 회로를 나눠서 진화시키기 때문에 탐색해야 할 범위가 줄어든다.

두 번째 방법은, 진화연산을 이용하여 처음부터 끝까지 회로를 설계하는 것이다. 이 방법은 평가 기준을 BPF 나 BSF 에 맞게 정의하기만 한다면 바로 설계할 수 있다. 하지만, 큰 회로를 한번에 설계하기 때문에 진화를 이용한 설계가 힘들 수 있다.

본 논문에서는 다양한 조건에서 첫 번째 방법과 두 번째 방법을 이용하여 BPF 를 설계하는 실험을 하였다. 그 결과 세대를 두 배, 회로크기를 두 배로 할 때, 가장 높은 성능을 가진 BPF 를 설계하였다.

2. BPF 를 자동으로 설계하는 방법

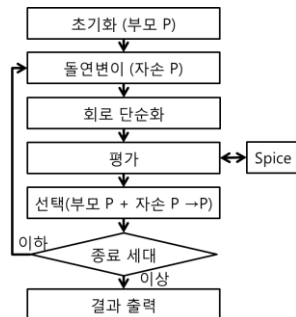


그림 1 아날로그 회로의 진화 과정

2.1 알고리즘

교배 연산 없이 돌연변이 연산만을 사용한 진화전략을 사용하여 다양한 회로를 생성한다.

2.1.1 초기화

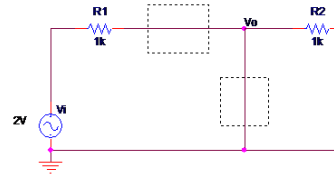


그림 2 Embryonic 회로

Embryonic 회로를 기반으로 하여 초기회로를 생성한다. 입력전압 V_i 는 2V 이며, Source 저항 R_1 , Load 저항 R_2 가 각각 $1k\Omega$ 이다. 출력 전압은 V_o 에서 측정한다. 점선으로 묶인 사각형 부분에 Inductor 또는 Capacitor 를 추가하고 정해진 범위 내에서 임의의 값을 정해준다.

2.1.2 돌연변이

돌연변이 연산은 부모회로를 변경하여 다양한 자식회로를 생성하는 방법을 뜻한다. V_i , R_1 , R_2 , V_o , Ground 를 제외한 부모회로의 어느 한 소자를 임의로 선택하고, 8 가지 연산 중 하나를 임의로 선택하여 연산을 수행한다.

1. 소자의 값을 임의로 변경
2. 소자의 종류를 임의로 변경
3. 대상 소자와 다른 종류의 소자를 병렬로 추가
4. 대상 소자와 다른 종류의 소자를 직렬로 추가
5. 해당 소자를 삭제
6. 해당 소자를 접지
7. 새로운 소자로 교체
8. 임의로 선택된 두 지점에 새로운 소자 하나를 추가

2.1.3 회로 단순화

회로가 불필요하게 복잡해지는 것을 막기 위해 동일한 소자가 직렬 또는 병렬로 연결된 경우 동일한 기능을 하는 하나의 소자로 바꾼다.

2.1.4 평가

생성된 회로를 Spice 를 이용하여 시뮬레이션 하여 출력을 얻어낸다. 이 출력과 원하는 출력을 비교하여 그 오차를 계산한다.

2.1.5 선택

부모 회로 p 개와 자식회로 p 개중에 오차가 가장 낮은 회로 p 개를 선택하여 다음 세대의 부모로 삼고, 나머지 p 개의 회로는 삭제한다.

2.3 BPF 를 설계하는 방법

크게 두 가지 방법을 이용해서 BPF 를 설계한다. 첫 번째 방법은 LPF 와 HPF 를 각각 설계한 뒤에 연결하여

BPF 를 설계하는 방법이고, 두 번째 방법은 BPF 를 한번에 설계하는 방법이다.

자동으로 회로를 설계하는 과정에서 N 세대 진화시키 최대 크기 M 인 회로를 설계했다고 할 때, 첫 번째 방법은 LPF 와 HPF 를 따로 설계하면서 N 세대씩 두 번(2N) 진화시키고, 두 회로를 연결한 회로는 최대 2M 의 크기를 가진다. 여기서 회로의 크기는 회로의 절점(Node)의 개수로 정의한다. 이 방법을 HPF+LPF 로 정의한다.

두 번째 방법에서는 최대 진화 세대가 N 또는 2N 인 경우와 회로의 최대 크기가 M 또는 2M 인 경우를 모두 고려하여 총 네 가지 경우에 대해서 실험한다. 각각 실험 NM, 2NM, N2M, 2N2M 으로 정의한다.

표 1 실험 방법

실험 방법	진화 방법	진화 세대	회로 크기
HPF+LPF	따로 진화	N+N	M+M
NM	한번에 진화	N	M
2NM	한번에 진화	2N	M
N2M	한번에 진화	N	2M
2N2M	한번에 진화	2N	2M

3. 실험 결과

표 2 실험 조건

실험 조건	
최대 진화 세대 N	300
복잡도, 절점의 수 M	10
집단의 크기 P	20
Capacitor 값의 범위	1~10000nF
Inductor 값의 범위	0.1~10000uF
Spice 종류	WinSpice

총 5 번 실험을 하여 그 평균을 결과로 한다. 실험 결과 가장 높은 성능을 가진 BPF 를 설계한 방법은 2N2M 방법이다. 평균적으로 가장 높은 성능을 보여줄 뿐만 아니라 그 표준 편차 또한 가장 작았다.

4. 결론

본 논문에서는 BPF 를 자동 설계함으로써 진화연산을 기반으로 하여 설계할 때 복잡한 회로는 어떻게 설계하는 것이 좋은지 알아보기 위해 다양한 조건에서 실험했다.

HPF 와 LPF 를 각각 설계한 뒤에 결합하는 방법(HPF+LPF)과, 진화연산을 통해 한번에 BPF 를 설계하는 방법 네 가지(NM, 2NM, N2M, 2N2M)를 비교했다. 이 중 2N2M 방법이 HPF 와 LPF 를 따로 설계하는 방법과 진화시간과 회로의 최대 크기가 같지만, 가장 좋은 성능의 BPF 를 설계했으며, 두 번째는 HPF+LPF 였다. 이런 결과가 나온 원인은 더 높은 크기의 회로를 설계할 수 있는 것이 넓은 해공간을 탐색해야 하는 부담으로 작용하기 보다, 더 다양한 회로를 시험해 볼 수 있는 기회가 되었기 때문이다.

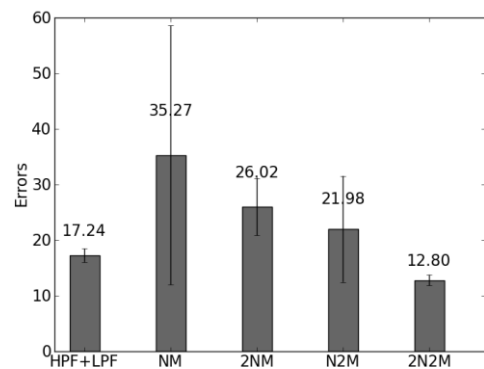


그림 3 평균 오차

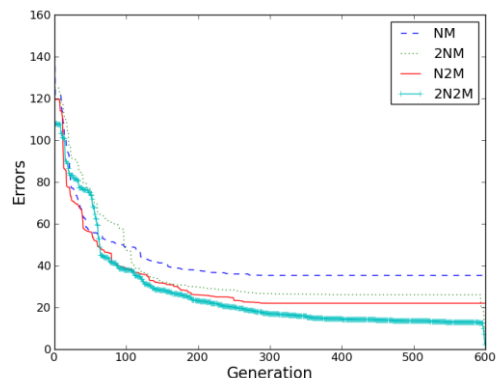


그림 4 평균 수렴 속도

참고문헌

- [1] J. R. Koza, F. H. Bennett III, D. Andre, M. A. Keane, and F. Dunlap, "Automated synthesis of analog electrical circuits by means of genetic programming," IEEE Transactions on Evolutionary Computation, vol. 1, pp. 109-128, 1997.
- [2] K.-J. Kim, A. Wong, and H. Lipson, "Automated synthesis of resilient and tamper-evident analog circuits without a single point of failure," Genetic Programming and Evolvable Machines, vol. 11 pp. 35-59, 2010
- [3] WINSPICE, <http://www.winspice.com>