

진화전략과 실질적인 소자 값 제한을 통한 물리적 구현이 가능한 아날로그 회로의 자동설계

(Automated Synthesis of Physically Implementable Analog Circuits using Evolutionary Strategy and Practical Constraints on Component Values)

박 현 수 ^{*} 김 경 중 ^{**}
(Hyun-Soo Park) (Kyung-Joong Kim)

요 약 아날로그 회로는 모든 디지털 시스템의 기본으로 현재도 높은 중요도를 가지고 있지만 설계에 많은 전문지식을 요구한다. 이러한 문제를 해결하기 위해 진화 연산을 이용한 설계 자동화 기법이 연구되어 왔으나, 많은 경우 재현이 어렵고 계산비용이 많이 요구되어왔다. 최근 들어, 진화 전략을 이용하여 적은 집단 크기로 아날로그 회로를 진화시킬 수 있는 방법이 제안되었다. 본 연구에서는 진화 전략을 이용하여 아날로그 회로를 설계하는 방법을 기초로 실제 구현이 가능한 회로를 제작하는 방법을 제안한다. 많은 경우 실제 소자를 이용하여 물리적으로 구현하는 것을 고려하지 않기 때문에, 진화 회로는 시뮬레이션에만 그치고 만다. 본 연구에서는 진화과정에서 회로가 가질 수 있는 소자 값을 산업체에서 선호하는 E-12시리즈로 제한한 진화전략을 통하여 적은 복잡도로 물리적으로 구현 가능한 회로를 만들 수 있음을 보인다. 물리적 구현을 위한 다른 기법들과 비교를 통해 제안하는 방법의 우수성을 보인다.

키워드 : 아날로그 회로, Low-Pass Filter, High-Pass Filter, 진화연산, 물리적 구현

Abstract Analog circuits are fundamental parts of modern digital electronic systems but they require much expert knowledge to design them. To solve this problem, there are a lot of works on the automated synthesis of analog circuits using evolutionary computation. However, it is difficult to reproduce their results because of their complexity and requires high computational cost. Recently, Kim et al. proposed a method to evolve analog circuits using evolutionary strategy maintaining a very small size population. In general, analog circuits are evolved using simulation software and they put less effort on hardware implementation. In this work, we propose a method to evolve physically implementable analog circuits using the evolutionary strategy-based algorithm and practical constraints on component values. This allows the evolution of realistic analog circuits with less computational cost and complexity. Experimental results show that the proposed method outperforms other alternatives.

Key words : Analog Circuit, Low-Pass Filter, High-Pass Filter, Evolutionary Computation, Physical Implementation

· 이 논문은 2009년도 세종대학교 교내연구비 지원에 의한 논문임

^{*} 학생회원 : 세종대학교 컴퓨터공학과
rex8312@gmail.com

^{**} 정 회 원 : 세종대학교 컴퓨터공학과 교수
kimkj@sejong.ac.kr

논문접수 : 2010년 9월 6일

심사완료 : 2011년 4월 7일

Copyright©2011 한국정보과학회 : 개인 목적이거나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지 : 소프트웨어 및 응용 제38권 제5호(2011.5)

1. 서론

아날로그 회로는 모든 디지털 시스템의 근간이 되며, 현재도 많은 용도로 사용하고 있다. 예를 들어 센서 샘플링, 디지털 신호와 아날로그 신호의 변환, 신호 증폭 및 filtering 등에 사용하고 있다. 일반적으로 아날로그 회로의 설계는 전문가의 지식에 의존하는 경향이 있다 [1]. 아날로그 회로 설계는 복잡성이 크고 변수도 많기 때문에, 전문가의 지식 없이 자동으로 설계하는 것은 어려운 일이다.

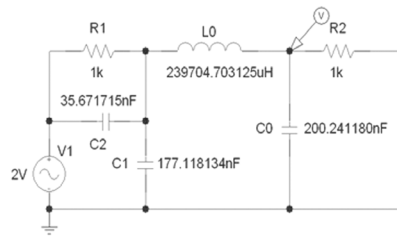
최근, 아날로그 회로를 자동으로 설계하려는 연구가 진행되고 있고, 특히 진화 연산을 활용하려는 연구가 관심을 받고 있다[2]. 대표적인 연구로 Koza가 진행한 유전자 프로그래밍을 이용한 아날로그 회로 자동 설계 연구가 있다[3]. 하지만, 이 연구의 경우 재현이 어렵고, 많은 계산 비용을 요구한다. 예를 들어, Low-pass filter를 진화하기 위해 집단의 크기를 32,000으로 설정하였다. 즉, 한 세대마다 32,000번의 아날로그 회로 시뮬레이션을 요구한다.

Koza의 연구 이후에 보다 단순한 형태로 아날로그 회로를 설계하려는 연구가 진행되었지만, 여전히 큰 집단의 크기를 유지해야만 하는 어려움이 있었고, 재현이 쉽지 않아서 연구가 널리 확장되는데 어려움이 있었다. 대표적인 관련 연구로 Lohn이 수행한 유전자 알고리즘을 이용한 아날로그 회로 설계 연구가 있다[4]. 하지만, 이 연구에서도 집단의 크기는 18000으로 설정하는 등 매우 높은 복잡도를 보인다.

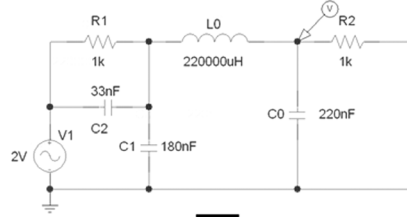
Kim 등은 진화 전략을 사용하여 매우 적은 집단의 크기로 아날로그 회로를 진화할 수 있는 방법을 제안하였다[5]. 진화 전략은 돌연변이만으로 진화를 수행하기 때문에 비교적 재현이 용이하며, 실험에서 집단의 크기는 20을 사용하였다. 기존 연구에 비해 매우 적은 계산 비용으로 회로를 진화할 수 있음을 보였다.

기존 아날로그 회로 진화 연구의 경우 실제 물리적인 구현까지는 고려하지 않고, 시뮬레이션을 통해 결과를 얻는 경우가 대부분이다. 그렇기 때문에 실제 진화를 통해 얻은 결과를 물리적으로 구현할 수 없다. 이러한 문제는 소자의 값이 실제로는 산업체에서 선호하는 E-12 시리즈로 제한되어 있기 때문에 발생한다. 물리적 구현을 고려하려면, 소자가 가지는 값을 산업체에서 선호하는 소자 값으로 제한할 필요가 있다.

현재까지 아날로그 회로 진화 연구에서 물리적 구현을 고려한 경우는 드물다. Kim 등은 진화의 최종 결과물인 회로의 소자들의 값을 산업체에서 선호하는 E-12 시리즈 소자 중에서 가장 가까운 것으로 치환하는 방법을 사용했다[5]. 이 방법은 가장 손쉽게 물리적으로 구



↓ 근사



↓ 실제 회로

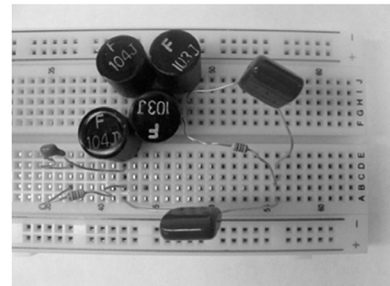


그림 1 진화 과정 중에 모든 소자 값은 위와 같은 근사 치환 과정을 거침

현 가능한 회로를 얻을 수 있지만, 치환과정에서 성능 하락이 크게 발생할 수 있다. Goh 등은 유전자 알고리즘 기반 아날로그 회로 진화기법에서 소자 값을 E-12 시리즈로 제한하는 방법을 사용했다[6].

본 연구에서는 적은 집단의 크기로 아날로그 회로를 진화시킬 수 있는 진화전략 기반 기법에 소자의 값을 E-12시리즈로 제한하는 방법을 적용하는 방법을 제안한다. 이러한 방법은 적은 비용으로 실제 물리적 구현이 가능한 회로를 자동 설계할 수 있도록 해준다. 제안한 방법의 우수성을 평가하기 위해 마지막 진화 결과를 근사 치환하는 방법, 평가 과정에서만 근사 치환하는 방법 등과 비교를 수행한다. 근사 치환하는 방법은 그림 1과 같다.

아날로그 회로를 진화하는 연구는 기존에 여러 연구팀을 통해 이루어져왔으며, 몇몇 회로는 매우 성공적으로 설계하기도 하였다[2-5]. 이러한 연구결과는 회로설계에 대한 전문적인 지식이 없더라도, 자신이 필요로 하

는 목적에 맞는 아날로그 회로를 자동으로 설계할 수 있는 가능성을 보여주었다. 본 연구는 기존 연구에서 깊이 있게 다루어지지 않은, 진화 회로의 실제 구현이라는 문제를 다루어보았다. 몇몇 연구에서 진화 알고리즘으로 설계한 아날로그 회로를 실제 구현하는 과정에서 발생하는 문제점들을 제기하긴 했지만[5,6], 어떻게 하면 우수한 실제 회로 결과물을 얻을 수 있을지에 대한 체계적인 실험과 다양한 방법제시는 없었다. 본 연구는 진화 알고리즘을 통해 설계하려는 아날로그 회로를 실제로 구현하는 것까지 고려할 때, 변경해 주어야 할 부분을 제안하고 그에 따른 성능차이를 실험적으로 평가해보았다. 또한, 실험결과를 설명할 수 있는 간단한 해공간 분석도 수행해 보았다. 이러한 시도는 아날로그 회로를 자동으로 설계하고 이를 실제로 구현하여 이용하도록 하는데 필요한 새로운 정보를 제공한다.

2. 관련 연구

2.1 아날로그 회로 진화

아날로그 회로는 기본 구성요소인 저항, Capacitor, Inductor, Transistor 소자 등이 모여서 이루어 진다. 자동으로 아날로그 회로를 설계하려면 회로에 사용될 소자의 종류와 값, 그리고 그들 사이의 연결관계를 결정해 주어야 한다. 일반적으로 아날로그 회로를 자동으로 설계하는 문제는 매우 큰 복잡도를 지닌다.

진화 연산은 자연계의 진화 현상을 모방한 탐색 알고리즘으로 높은 복잡도를 지닌 공학 문제를 해결하는데 좋은 성능을 보여왔다[7]. 대표적인 기법으로 유전자 알고리즘(Genetic Algorithm), 유전자 프로그래밍(Genetic Programming), 진화 전략(Evolutionary Strategy) 등이 있다.

아날로그 회로를 진화연산을 이용하여 자동 설계하려는 연구가 활발히 진행되어 왔다. 복잡한 해 공간에서 좋은 회로를 탐색하는데 진화 연산이 좋은 성능을 보였기 때문이다. 아날로그 회로를 진화시키는 과정은 다음과 같다. 1) 아날로그 회로가 수행해야 할 일을 정의한다. 2) 아날로그 회로를 컴퓨터 상에서 표현하기 위한 방법을 정의한다. 일반적으로 0과 1의 조합, 트리, 실수 벡터 등을 활용한다. 3) 임의로 N개의 아날로그 회로를 생성한다. 4) 아날로그 회로 시뮬레이터를 이용하여 각각의 회로가 1단계에서 정의한 일을 얼마나 잘 수행하는지 평가한다. 5) N개의 회로 중 우수한 회로를 일부 선택한다. 선택한 회로들을 임의로 선택하여 일부분을 교차하여 새로운 회로를 만든다. 선택한 회로들의 일부분을 변형한다. 6) 종료조건을 만족하지 않으면 4단계로 돌아간다.

가장 대표적인 연구는 John Koza가 수행한 유전자

프로그래밍을 이용한 회로 진화이다[3,8]. 그는 가장 기본이 되는 회로구조를 정의하고, 이 회로를 어떤 식으로 변형하여 새로운 회로를 만들어낼 지를 정의하였다. 즉 그가 진화시킨 것은 기본 회로에 컴포넌트를 어떻게 추가할지에 관한 일련의 명령어 집합이었다. 그는 이러한 명령어 집합을 Parse Tree형태로 표현하였고, 유전자 프로그래밍을 사용하여 진화하였다. 그는 실험을 통해 다양한 문제에 대해 우수한 아날로그 회로를 설계할 수 있음을 보였다.

Koza의 연구는 많은 가능성을 보여주었지만, 아쉽게도 재현에 많은 어려움이 따랐다. Tree기반 진화의 구현 복잡도가 컸기 때문이다. 이러한 이유로 보다 단순한 형태의 아날로그 회로 진화 기법들이 등장하였고, 대표적으로 유전자 알고리즘과 진화 전략을 이용한 기법이 있다[5,8,9]. 하지만, 대부분의 기법이 높은 계산 비용과 재현의 어려움 때문에 결과가 널리 확산되지 못하였다.

Kim은 진화 전략을 이용한 아날로그 회로 기법을 제안하였다[5]. 진화 전략은 돌연변이만을 사용하여 해를 탐색하는 방법으로 신경망 진화과정에서 적은 집단 크기로 좋은 성능을 얻을 수 있음을 보였다[10]. 이러한 결과를 토대로 아날로그 회로를 진화 전략을 이용하여 자동 설계하는 방법을 제시하였으며, 결과적으로 매우 적은 집단 크기로 회로를 설계할 수 있음을 보였다.

2.2 현실적 구현을 고려한 아날로그 회로 진화

산업체에서는 판매하는 소자가 가지는 값을 표준화하기 위해 {10, 12, 15, 18, 22, 27, 33, 39, 47, 56, 68, 82} 등으로만 값을 제한하고, 여기에 10n을 곱한 값을 허용하기로 하였다. 이러한 표준을 E-12 series라고 부른다. 이러한 표준을 따르지 않고 진화를 진행할 경우 시뮬레이션에서 좋은 성능을 얻은 회로를 실제 물리적인 소자들을 이용하여 구현하는데 어려움이 따른다.

이 문제에 대해 몇 가지 해결책이 제시되어 왔다. 가장 손쉬운 해결책은 다음과 같다. 진화를 통해 찾은 회로에서 필요한 소자 값이 E-12 시리즈에 없는 경우, E-12에서 구할 수 있는 소자들을 직렬 및 병렬 연결하여 특정 값을 만들어내는 방법이다. 이러한 방법은 하나의 소자를 만들기 위해 매우 많은 소자들이 동원되어야 하기 때문에 현실성이 낮다.

그 다음으로 사용할 수 있는 방법은 진화한 아날로그 회로의 소자를 가장 가까운 값을 가지는 E-12 시리즈 소자로 교체하는 방법이다[5]. 이 방법은 교체로 인한 값의 변화가 발생하기 때문에 성능이 하락할 수 있다. 또 다른 방법은 아날로그 회로를 진화할 때 소자의 값을 E-12 시리즈로만 한정하는 방법이다[6].

표 1은 물리적 구현을 고려한 아날로그 회로 진화관련 연구를 요약하고 있다. Kim의 연구는 진화 전략을

표 1 물리적 구현을 고려한 아날로그 회로 진화 요약 및 비교

참고문헌	진화 알고리즘	물리적 구현 고려 방식	집단 크기
Kim et al. [5]	진화 전략	진화 종료 후에 근사치환	20
Goh et al. [6]	유전자 알고리즘	진화 과정 중에 GA로 최적화	200×10
제안한 방법	진화 전략	진화 과정 중에 근사치환	20

기반으로 하였지만, 진화 종료 후에 근사 치환하는 방식으로 이루어졌다[5]. Goh의 연구는 유전자 알고리즘을 사용했으며, 유전자 알고리즘을 이용하여 진화 과정 중에 E-12시리즈 값을 가지는 소자 값을 최적화하였다[6]. 회로 최적화 과정에서 집단의 크기가 200이며, 소자 값 최적화 과정에서 집단의 크기가 10이므로, 총 2000의 집단 크기를 유지해야 한다. 본 논문에서 제안하는 방법은 진화 전략을 이용하여 적은 집단 크기를 유지하면서, 진화 과정 중에 근사치환을 하도록 하였다. Goh의 연구와 달리 추가로 최적화를 하지 않기 때문에 적은 집단의 크기를 그대로 유지하였다.

3. 물리적 구현을 고려한 아날로그 회로 진화

본 장에서는 아날로그 회로를 진화하는 기본 방법과 실제 구현을 고려한 기법에 대해 소개한다. 기본 방법에서 소자의 값에 제한이 없기 때문에 소자는 범위 내의 어떠한 실수 값도 가질 수 있다. 실제 구현을 고려한 세 가지 기법들은 기본적인 진화 기법을 변형하여 실제 구현이 가능한 형태의 회로를 결과물로 생성한다.

3.1 아날로그 회로 진화

아날로그 회로는 진화전략을 사용하여 생성한다. 이 기법은 돌연변이만을 사용하여 진화를 진행하며, 부모와 자식을 합한 집단에서 가장 우수한 개체 P 개가 새로운 집단을 만드는 방식이다. 그림 2는 본 논문에서 사용한 아날로그 회로 진화과정을 보여준다. 초기화 단계를 통해 P 개의 아날로그 회로를 임의로 생성하고, 여기에 돌연변이가 연산을 적용하여 새로운 자손 P 개를 생성한다. 회로가 지나치게 복잡해지는 것을 방지하기 위해 회로 단순화 과정을 거치고, 그 이후 SPICE 시뮬레이터를 통해 생성한 회로의 출력 응답이 목표하는 응답과 유사한 정도를 측정한다. 목표 응답과 유사한 결과를 제시하는 회로는 높은 적합도를 받고, 그렇지 못한 회로는 낮은 적합도를 받는다. 부모와 자식을 합한 2×P개의 아날로그 회로들 가운데, 적합도가 높은 P 개를 선택하여 새로운 집단을 형성한다. 이 새로운 집단을 대상으로 돌연변이를 적용하여 새로운 자손을 만들고 위와 같은 과정

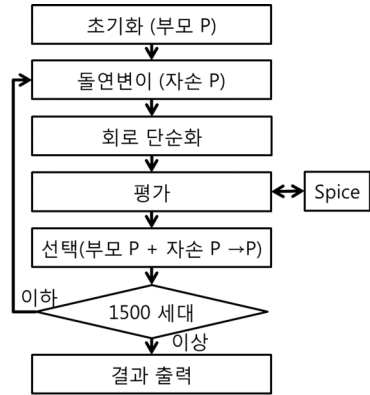


그림 2 아날로그 회로 진화 과정

을 최대 세대까지 반복하도록 한다. 각각의 단계에 대해 좀 더 상세히 서술하도록 한다.

3.1.1 Embryonic 회로를 이용한 초기화

회로를 최초로 임의로 만들 때 그림 3과 같은 Embryonic 회로를 토대로 한다. 이러한 방법은 아날로그 회로를 진화하는 연구에서 공통적으로 사용하는 접근법이다. 기본적으로 입력 전압이 어떤 것이고, 회로에 있는 source 저항과 load 저항은 정의해 주어야 한다. 또한, 접지 설정과 어떤 부분에서 출력 전압을 측정할지도 확정이 필요하다. 빈 박스로 표시한 부분에 새로운 소자들이 연결되며, 지속적으로 성장과 축소를 진행한다. 입력 전압 Vi는 2V이며, 출력 전압은 Vo이다. 1kΩ 저항 2개(R1, R2)가 그림과 같이 배치되어 있다.

P 개의 회로를 embryonic 회로를 토대로 임의로 생성한다. 빈 박스 부분에 Inductor 또는 Capacitor 소자가 놓일 수 있으며, 각 소자의 값은 사전에 정의한 범위에서 임의로 결정한다.

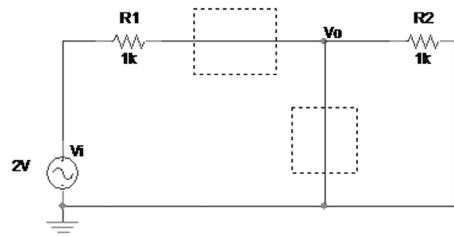


그림 3 회로 생성시 기본이 되는 Embryonic 회로

3.1.2 돌연변이 연산

돌연변이의 목적은 부모 회로를 일부 변형하여 새로운 자식 회로를 만드는 것이다. 각각의 부모 회로는 한 개의 자식회로를 8가지 돌연변이 중 하나를 사용하여 생성할 수 있다. 부모의 회로에는 일반적으로 한 개 이

상의 소자가 있을 수 있기 때문에, 돌연변이를 할 소자는 임의로 결정한다. 8가지 돌연변이 기법 중 어떤 것을 택할지도 임의로 결정한다. Embryonic 회로에 정의한 입력전압, source 저항, load 저항 등은 돌연변이의 대상에서 제외한다. 돌연변이의 예는 다음과 같다.

- 소자 값 변환: 소자의 값을 임의로 변경한다. 예를 들어, 330uH를 가지던 inductor를 돌연변이를 통해 580uH로 변경할 수 있다.
- 소자 변환: 소자의 종류를 임의로 변경한다. 예를 들어, inductor였던 소자를 capacitor로 변경한다. 이때 소자의 값은 그대로 유지한다.
- 병렬로 추가: 돌연변이 대상 소자와 다른 형태의 소자를 병렬로 연결한다. 즉 inductor소자를 돌연변이 하려고 한다면, capacitor 소자를 병렬로 연결한다. 이 capacitor 소자의 값은 임의로 정한다.
- 직렬로 추가: 돌연변이 대상 소자와 다른 형태의 소자를 직렬로 연결한다.
- 소자 삭제: 해당 소자를 삭제한다.
- 접지 연결: 해당 소자를 접지한다. 즉 GND에 연결한다.
- 소자 교체: 새로운 소자로 교체한다. 같은 종류의 소자로 교체가 가능하다. 교체할 소자의 값은 임의로 결정한다.
- 소자 추가: 임의로 선택한 두 지점에 새로운 소자를 하나 추가한다. 소자의 형태와 값은 임의로 정한다.

3.1.3 회로 단순화

회로가 불필요하게 복잡해지면, 상대적으로 구현비용이 높이 들어간다. 이러한 불편을 없애기 위해 진화단계에서 동일한 형태의 소자들이 직렬 또는 병렬로 연결될 경우, 하나의 소자로 바꾼다. 여러 개의 소자 값을 하나로 변형하는 것은 일반적인 회로 이론을 따라 처리한다(그림 4).

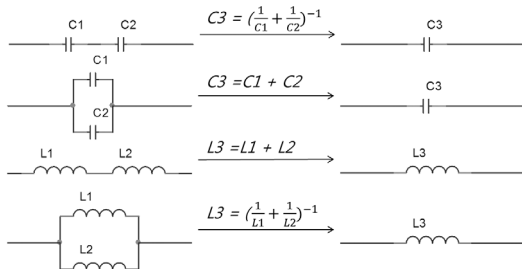


그림 4 회로 단순화의 예

3.1.4 회로의 평가

개체의 우수성을 평가하는 과정은 진화의 성공을 위해 가장 중요한 부분이다. 본 논문에서는 low-pass filter(LPF)와 high-pass filter(HPF)를 설계하는 것을

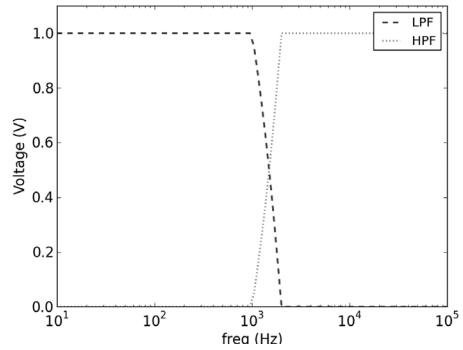


그림 5 목표 응답

목표로 한다. 그림 5는 본 연구에서 목표하는 응답을 보여준다. X축은 AC전압의 주파수를 보여준다. Y축은 출력 전압을 보여준다.

LPF의 경우 1000Hz이하에서는 1V가 출력되며, 2000Hz이상에서는 0V를 출력한다. HPF의 경우에는 반대로 1000Hz이하에서 0V, 2000Hz이상에서 1V를 출력한다. 두 Filter모두 1000~2000Hz사이의 구간에 대해서는 고려하지 않는다.

각각의 후보 회로는 SPICE 시뮬레이션을 통해 출력 응답을 구하며, 목표 응답과의 유사성을 측정하여 적합도를 계산한다. Vd_f 는 f 주파수 일 때 목표 출력이며, Va_f 는 f 주파수에서의 후보 회로의 실제 출력이다. 오차 e 는 다음과 같다.

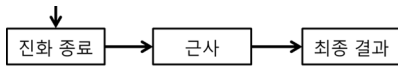
$$e = \sum_f \sqrt{Adj(Vd_f - Va_f)^2}$$

f 는 1Hz부터 100kHz까지 $10^{\frac{n}{20}}$ Hz마다 응답을 구한다. 이 때 $n \in \{x | x \in N, 1 \leq x < 60 \text{ or } 65 \leq x \leq 100\}$ 이다. $60 \leq x < 65$ 구간은 1000Hz와 2000Hz 사이 주파수기 때문에 고려하지 않는다. 여기서 $Adj(v)$ 는 v 가 0.01 이상일 경우 10을 곱하며, 상대적으로 오차가 큰 경우에 큰 가중치를 부여하는 방식이다.

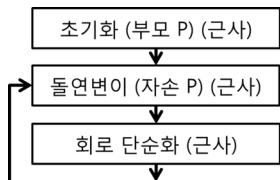
본 논문에서는 [8]에서 사용한 평가함수에 기반하여 간단히 특정 주파수마다 시뮬레이션 출력전압과 원하는 출력전압을 비교하여 그 차이를 모두 합한 전체 오차가 가장 작은 것을 좋은 회로로 평가한다. 하지만, 실제 LPF의 특성은 maximum passband flatness, attenuation (dB) 및 phase response로 나타내기 때문에, 이런 평가함수는 LPF의 일부 특성밖에 평가할 수 없다. 또한 passband ripple에 의한 오차와 attenuation에 의한 오차가 합쳐지므로 의도하지 않은 결과가 나올 수 있다. 때문에 설계하고자 하는 회로에 따라서는 좀 더 정교한 평가함수를 사용할 필요가 있을 수 있다.

3.2 물리적 구현을 고려한 진화

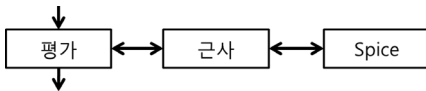
본 논문에서 사용한 기본적인 아날로그 회로 진화 방법은 물리적인 구현을 고려하지 않았기 때문에, 실제 진화 결과를 구현해서 테스트하기가 용이하지 않다. 본 논문에선 E-12 규약에 맞는 소자만으로 이루어진 회로를 생성해 내기 위해 사용할 수 있는 세가지 방법을 비교한다. E-12 series 소자 값은 다음과 같이 $\{10,12,15,18,22,27,33,39,47,56,68,82\} \times 10^i$ 에 해당하는 값을 가질 수 있다.



(a) Method 1



(b) Method 2



(c) Method 3

그림 6 물리적 구현을 고려한 방법 개요(기본 알고리즘에서 변경한 부분만 표시)

3.2.1 Method 0

이 방법은 실제 구현 가능 여부는 생각하지 않고 소자들이 어떤 값(실수)이라도 가질 수 있다고 가정하고 진화를 수행한다. 그림 2 알고리즘을 그대로 수행하고 결과를 출력한다. 이 방법은 구현을 고려하지 않으며, Method 1을 위한 중간 과정에 불과하지만, 실험 결과와 분석과정에서 비교를 위해 Method 0로 정의한다.

3.2.2 Method 1(진화 종료 후에만 근사 치환)

Method 0에서는 진화가 끝난 후에 가장 좋은 회로 하나만을 선택하여 최종결과로 삼지만, 결과적으로 실제 구현이 어렵다. 본 방법에서는 해당 회로에 있는 소자의 값과 가장 유사한 값을 가지는 E-12소자로 교체한다(그림 6(a)).

3.2.3 Method 2(제안하는 방법, 진화 과정 중에 근사 치환)

본 방법은 회로 진화 과정에서 각 소자가 가질 수 있는 값을 E-12에 적합한 경우만 허용하는 것이다. 초기화, 돌연변이, 회로 단순화 등에서 모두 E-12를 만족하는 소자만을 사용하도록 회로의 소자가 가지는 값은 근사 치환한다(그림 6(b)).

3.2.4 Method 3(비교를 위해 제시된 방법)

마지막 방법은 Method 2와 달리 평가과정에서만 근사 치환을 수행한다(그림 6(c)). 진화과정 중에 근사치환을 한다는 점에서 Method 2와 유사하지만, 단지 평가과정 중에만 치환이 일어나기 때문에 근본적으로 다른 방법이다.

4. 실험 결과

표 2는 실험에 사용한 파라미터 값을 보여준다. 본 실험에서 사용한 진화전략 기반 아날로그 회로 기법은 상대적으로 적은 집단 크기를 사용한다. 실험결과는 총 5회 반복한 평균결과이다.

표 2 실험 파라미터

집단 크기	20
최대 세대	1500
실험 횟수	5
SPICE 종류	Winspice[11]
Inductor 값의 범위	0.1~10000uH
Capacitor 값의 범위	1~10000nF
최대 Node 수	10

4.1 설계한 Filter의 오차

그림 7은 각 방법을 이용하여 설계된 Filter의 오차를 나타낸다. Method 1은 Method 0에서 찾아낸 결과를 구현 가능한 근사 회로로 변환하여 측정된 결과이다. Method 0와 비교해봤을 때 Method 1은 구현을 고려함으로써 성능하락이 발생하였다.

Method 2가 두 가지 회로 모두에서 다른 Methods에 비해 안정적으로 우수한 성능의 Filter를 설계했다. 다른 Method는 그에 비해 평균 오차가 높고 표준 편차도 큰 것으로 나타났다.

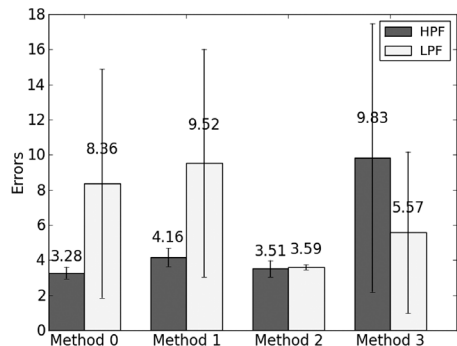


그림 7 Filter의 평균 오차

4.2 진화 속도

그림 8은 세대의 따라 오차가 어떻게 최소값에 수렴

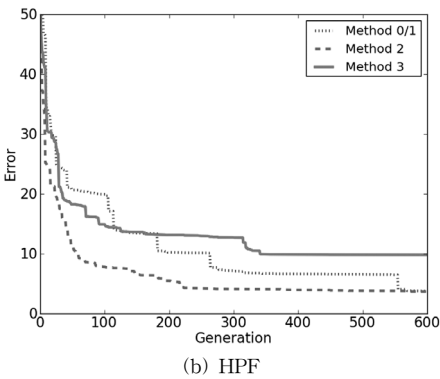
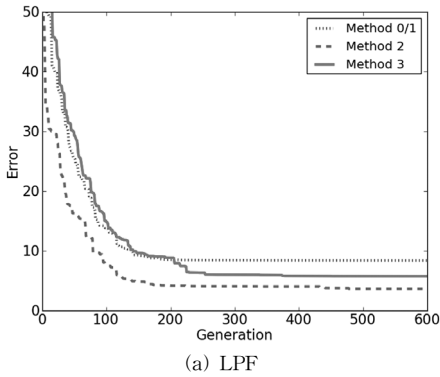


그림 8 수렴속도 비교(5번 평균)

해 가지를 나타낸다. Method 0, 1은 이 과정에서는 동일하다. 각 세대별 가장 우수한 개체의 적합도를 보여준 것이다. 600세대 이후에는 큰 변화가 없었다. 이 결과를 통해 Method 2가 가장 빠르게 높은 성능의 Filter를 설계함을 알 수 있다.

4.3 구현 가능한 진화 회로

그림 9는 각 방법을 사용하여 찾아낸 Low-pass filter들을 보여준다. 모두 구현이 가능한 상황이나 Method 2를 통해 생성한 회로가 셋 중에 가장 좋은 성능을 보였다. 또한 회로의 복잡도도 가장 낮았다. 그림 10은 각 방법에 대한 결과 회로의 출력 응답을 보여준다. 세 방법 모두 filtering을 성공적으로 수행하는 것을 볼 수 있다.

하지만, 설계된 회로 중 일부에서 문제점이 발견되었다. 회로를 제작할 때는 모든 Node들의 DC 전압이 명확해야 하지만, 그림 9의 (a) 및 (c) 회로를 비롯한 몇몇 회로는 그렇지 못한 부분이 있다. 두 capacitor 사이에 inductor가 있는 경우 inductor 양단의 DC 전압을 명확히 알 수 없기 때문이다. 이런 경우가 발생하는 이유는 진화전략에 회로 설계에 대한 전문적인 지식이 적용되지 않았기 때문이다. 이러한 부분에 대해서는 향후 연구에서 다룰 예정이다.

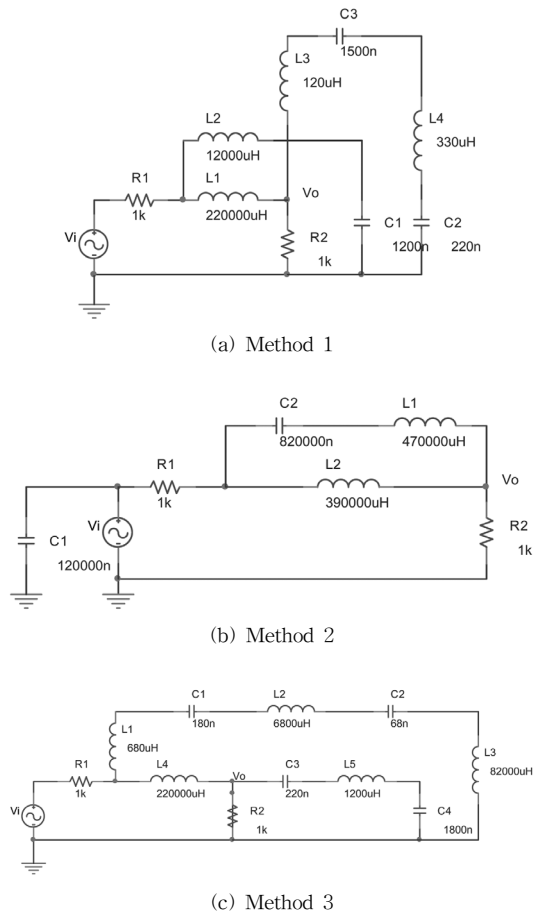


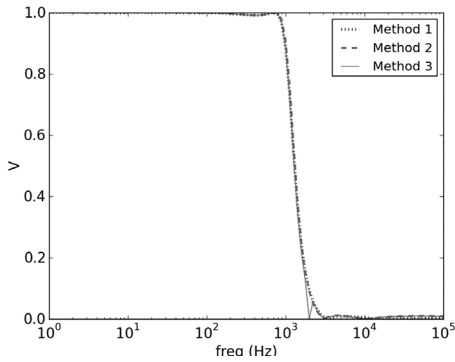
그림 9 각각의 방법으로 진화한 LPF 회로들

4.4 결과 분석

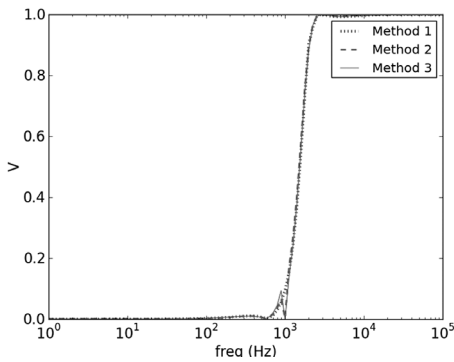
실험 결과, Method 2가 가장 안정적으로 좋은 성능의 Filter를 설계할 수 있었다. 실제로, 더 오랜 시간 진화 시키면 Method 0가 가장 성능이 좋은 회로를 설계할 수도 있지만, 이 회로는 실제로 구현이 불가능하기 때문에 Method 1을 사용하여 구현 가능한 회로로 변환하여야 한다. 변환 뒤에는 약간의 성능하락이 발생하여 대체로 Method 2보다 성능이 떨어진다.

각 방법의 근본적인 차이를 살펴보면, 다른 모든 단계에서는 동일하지만, 초기화, 돌연변이, 단순화 과정에서 값을 지정하는 부분만이 다른 것을 알 수 있다. 결과적으로 Method 2만이 그 영향을 받아 더 성능이 향상되었다.

이것을 좀 더 자세히 분석하기 위해, 기본 회로에 Capacitor가 하나뿐인 LPF를 가정하면, C 값에 따른 오차를 그림 11과 같이 나타낼 수 있다. Method 1은 부드러운 곡선을 그리고, Method 3은 계단 모양으로 나



(a) LPF



(b) HPF

그림 10 구현 가능한 회로들의 출력 응답

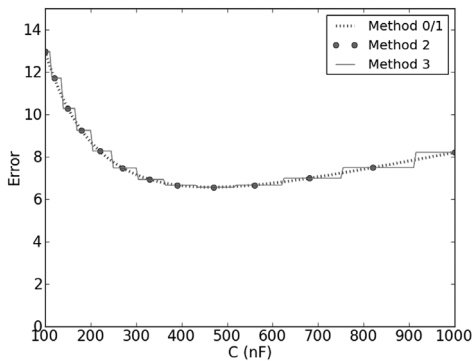


그림 11 해공간 분석

타난다. Method 2는 Method 1, 3의 선위에 점으로 나타난다. 각 방법은 값을 지정하는 부분만이 다르기 때문에 돌연변이 연산은 회로의 구조를 변경하지 않고, 값만 변경한다고 가정하면, 최적값을 찾을 확률은 무작위로 C 값을 집단크기(p)×전체 세대(n)번 선택하여 최적값인 430~515 사이의 값이 한번이라도 나올 확률과 같다. C 값은 1~1000사이의 값을 가지므로 Method 1, 3가 최

적해에 도달할 확률은 약 $\frac{85}{10000-1} \times pn \approx 0.0085pn$ 이 된다. 반면 Method 2는 점 위의 값 72개만을 가지므로 $\frac{1}{72} \times pn \approx 0.0138$ 이 되어 최적해를 찾을 확률이 더 높다.

Method 2의 해공간은 Method 1, 3보다 줄어들었다고 할 수 있다. 때문에 최적해에 도달할 확률이 높아지고 결과적으로 더 좋은 설계를 할 수 있다고 분석할 수 있다.

4. 결론 및 향후 연구

진화연산을 이용하여 아날로그 회로를 자동으로 설계하는 연구들은 많은 가능성을 보여주었지만, 재현의 어려움, 높은 계산 비용, 시뮬레이션 결과에만 머무름 등의 문제가 있었다. 본 연구에서는 적은 집단 크기로 아날로그 회로를 탐색할 수 있는 진화전략 기반 아날로그 회로 자동 탐색에 소자 값 제한을 적용하여 실제 물리적으로 구현 가능한 회로를 찾는 방법을 제안하였다.

동일한 진화전략기반 아날로그 회로 탐색에서 제안하는 방법과 다른 대안기법들을 비교해 보았을 때 제안하는 방법이 가장 우수한 성능을 보임을 알 수 있었다. 실험결과와 의미를 파악하기 위해 간단한 해 공간 분석을 수행해 보았고 제안한 방법이 다른 기법보다 우수한 이유를 설명하였다.

향후 연구로 실제 회로에서 발생할 수 있는 소자의 개수와 성능 사이의 trade-off관계에 대해 다루는 것이 필요하다. 소자의 개수가 많아지면, 비용이 늘지만, 좋은 성능의 회로를 설계할 가능성이 높아진다. 이러한 관계를 고려하여 진화하는 것이 필요하며, 이것은 서로 상충하는 두 목표를 조화해야 하기 때문에, 일반진화 보다는 multi-objective 진화연산을 적용해 보는 것이 필요하다.

또한, 그림 9의 (a) 및 (c)의 회로와 같이 진화연산에 전문적인 지식이 적용되지 않아 발생하는 문제에 대한 해결책도 필요하다. 전문적인 지식은 일반적으로 많이 이용되는 회로 패턴이라고 말할 수 있다. 이 패턴들은 전문가들이 자주 이용하는 형태, 또는 절대 이용하지 않는 형태 등으로 나타낼 수 있다. 향후 연구에서는 이러한 패턴들을 회로를 평가할 때 이용하여 전문지식을 설계에 적용하는 연구를 진행하려 한다.

참고 문헌

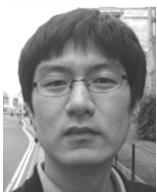
[1] R. Spencer, and M. Ghauri, *Introduction to Electronic Circuit Design*, Prentice Hall, 2002.
 [2] R. S. Zebulum, M. A. Pacheco, and M. M. B. Vellasco, *Evolutionary Electronics: Automatic Design of Electronic Circuits and Systems by*

- Genetic Algorithms*, CRC Press, 2001.
- [3] J. R. Koza, M. A. Keane, and M. J. Streeter, "What's AI done for me lately? Genetic programming's human-competitive results," *IEEE Intelligent Systems*, vol.18, no.3, pp.25-31, 2003.
- [4] J. D. Lohn, and S. P. Colombano, "A circuit representation technique for automated circuit design," *IEEE Transactions on Evolutionary Computation*, vol.3, no.3, pp.205-219, 1999.
- [5] 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.
- [6] C. Goh, and Y. Li, "GA automated design and synthesis of analog circuits with practical constraints," *Proceedings of the 2001 Congress on Evolutionary Computation*, vol.1, pp.170-177, 2001.
- [7] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989.
- [8] 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.
- [9] Y. Sapargaliyev, and T. Kalganova, "Constrained and unconstrained evolution of "LCR" low-pass filters with oscillating length representation," *Proceedings of the IEEE Congress on Evolutionary Computation*, pp.1529-1536, 2006.
- [10] K. Chellapilla, and D. B. Fogel, "Evolving neural networks to play checkers without relying on expert knowledge," *IEEE Transactions on Neural Networks*, vol.10, no.6, pp.1382-1391, 1999.
- [11] WINSPIICE, <http://www.winspice.com/>



박 현 수

2010년 세종대학교 전자공학과 졸업(학사). 2011년~현재 세종대학교 컴퓨터 공학과 재학(석사)



김 경 중

2000년 연세대학교 컴퓨터과학과 졸업(학사). 2002년 연세대학교 컴퓨터과학과 졸업(석사). 2007년 연세대학교 컴퓨터과학과 졸업(박사). 2007년~2009년 미국 Cornell 대학교 Postdoctoral 연구원
2009년~현재 세종대학교 컴퓨터공학과

조교수. 관심분야는 로봇지능, 게임지능, Bioinformatics