

정규 구간선형 방정식을 갖는 저항성 회로의 효율적인 해석

(Efficient Analysis of Resistive Networks With Canonical Piecewise-Linear Equations)

趙俊暎*, 趙振國*, 權容世**, 金穎煥**

(Joon-Young Cho, Jin-Kuc Cho, Yong-Se Kwon and Young-Hwan Kim)

要約

본 논문에서는 선형 분할을 갖는 정규 구간선형 방정식을 해석하는 새로운 알고리즘을 제안하고, 저항성 회로의 해석을 통하여 효율성을 보인다. 제안 알고리즘은 Newton-Raphson (N-R) 계산 과정에서 다음 계산을 위한 추정해가 될 수 있는 점들의 함수 값과 실제해의 함수 값들을 비교함으로써 실제해에 가장 가까운 추정해를 선택하는 것을 기본 생각으로 하고 있으며, 기존의 Katzenelson 알고리즘에 비하여 해가 수렴하기 위하여 필요한 N-R 계산 회수의 대폭적인 감소가 가능하다. 회로의 해석에 이용하여 본 결과, 제안 알고리즘은 시험 회로의 크기에 따라서 Katzenelson 알고리즘에 비하여 N-R 계산 회수는 1/20~1/8, CPU 시간은 1/10~1/5로 줄임으로써 계산 효율이 크게 향상됨을 확인하였다. 또 회로의 동작 표현에 필요한 선형 구간의 수가 증가할수록 계산 효율도 같이 향상됨을 알 수 있었다.

Abstract

This paper proposes new algorithms to solve canonical piecewise-linear equations with linear partitions and illustrates their efficiency through the analysis of resistive networks. The basic idea of the proposed algorithm is to find the best next guess, closest to the actual solution, at each Newton-Raphson (N-R) iteration by comparing the images of next guess candidates and that of the actual solution. The proposed algorithm can reduce the number of the N-R iterations required for convergence greatly, compared to the Katzenelson algorithm. When applied to analyzing test circuits, the proposed algorithm required 8 to 20 times fewer N-R iterations and 5 to 10 times less CPU time than the Katzenelson algorithm, depending on the size of the circuits. The experimental results also exhibit that the efficiency of the proposed algorithm over the Katzenelson algorithm increases as the number of the piecewise-linear regions for the representation of the circuit

1. 서론

*學生會員, **正會員, 浦項工科大学校 電子·電氣工學科 (Dept. of Elec. Eng., Pohang Univ. of Science and Technology)
接受日字: 1994年 5月 21日

전자 회로의 동작을 조사하기 위하여 사용하는 시뮬레이터는 해석하고자 하는 회로를 기술하는 소자의 모델링 레벨에 따라서 기능레벨, 레지스터 전송레벨,

논리레벨 및 회로레벨 시뮬레이터 등으로 구분된다.¹¹¹ 이들 중 SPICE¹²⁾나 ASTAP¹³⁾과 같은 회로레벨 시뮬레이터는 해석 결과의 신뢰성이 높고, 과도해석(transient analysis)에 사용할 경우 정확한 아날로그 파형을 제공하므로 회로 설계자들이 가장 많이 사용하는 소프트웨어 중의 하나이다. 하지만 해석하고자 하는 회로의 크기에 대하여 해석시간이 비선형적으로 증가하므로 고집적회로의 설계에 사용하기 어려운 문제가 있다.^{15, 6)} 이와 같은 문제점을 해결하기 위하여 여러 가지 방법들이 시도되고 있으며, 타이밍 시뮬레이션¹¹⁾도 이러한 노력중의 하나이다. 타이밍 시뮬레이터는 relaxation 기법⁷⁾을 사용하여 방정식의 해를 구하며, 이 과정에서 Newton-Raphson 계산 회수를 줄임으로써 CPU 시간을 줄이는 것을 기본 생각으로 하고 있다. 하지만 수치 해석의 불안정성으로 인하여 해석결과의 신뢰성이 크게 떨어지므로 회로 설계자들은 사용을 기피하고 있는 현실이다. 최근에는 방정식의 해가 수렴할 때까지 Newton-Raphson 계산을 반복하여 수치 해석의 불안정성 문제를 해결하였으나^{18, 9)} 인덕터, 부동 캐패시터(floating capacitor) 또는 피드백 루프(feedback loop)를 포함한 회로의 경우 해석 속도가 현저히 떨어지며, 연산 증폭기와 같이 활동성(activity)이 높은 회로의 경우에는 SPICE와 같이 직접방식(direct method)을 사용하여 방정식의 해를 구하는 회로레벨 시뮬레이터보다 오히려 해석 시간이 오래 걸리기도 하는 단점이 있다.

디지털 시스템의 경우에는 게이트와 같은 macro 소자를 회로레벨 시뮬레이터를 사용하여 미리 해석한 후, 이를 라이브러리의 형태로 저장하여 설계 회로를 해석할 때 사용할 수도 있다.¹⁰⁾ 이 방법은 디지털 회로의 해석에 매우 효율적이지만, macro 소자의 종류와 입력 조건에 따른 수 많은 선해석(pre-analysis)을 필요로 하며 많은 양의 기억 소자를 필요로 하는 단점이 있다. 또 해석 회로의 기생 소자 값이 macro 소자의 기생 소자 값과 다를 경우 오차가 커지게 되며, 라이브러리에 없는 논리 블록을 사용한 경우 해석이 불가능하므로 설계 형태에 제약을 주게 된다. Macro 소자의 동작을 간단한 식으로 모델링하여 사용할 수도 있으나, macro 소자의 종류에 제한이 있으며 해석 오차가 클뿐 아니라, 인덕터나 부동 캐패시터를 포함하는 회로를 해석하는데 어려움이 있다.¹¹¹⁾

MOSFET 회로의 해석에만 적용되는 방법으로, MOSFET을 선형 저항으로 대치한 후 RC 딜레이 모델¹²⁾을 사용하여 신호의 전파 지연시간을 계산하는 방법도 있다. 이러한 방법의 경우, MOSFET의 등가

저항을 구하기 위하여 수 많은 회수의 회로레벨 시뮬레이션이 필요하며, 부동 캐패시터와 인덕터가 있는 회로를 해석하지 못하는 단점이 있다. 또 benchmark 회로들을 이용하여 MOSFET의 실효저항값을 충분히 조정된 후에도, 이를 다른 회로의 해석에 사용하면 해석의 정확도가 떨어진다. 이들과 다른 접근방법으로 비선형 회로 소자의 동작을 구간선형으로 모델링하여 해석하는 구간선형 시뮬레이션이 있다.^{113, 14)}

구간선형 시뮬레이션은 회로레벨 시뮬레이션과는 달리 비선형 방정식대신 구간선형 방정식의 해를 구함으로써 해석시간을 감소시키는 것을 목표로 하고 있다. 구간선형 시뮬레이터는 회로레벨 시뮬레이터와 같이 모든 종류의 소자를 지원하며 해석 회로에 제약이 없는 장점이 있다. 또 소자의 특성을 표현하기 위한 선형 구간의 수를 조정함으로써 해석 속도와 정확도의 조정이 가능하며, 현재 상용 제품으로도 제공되고 있다.¹¹⁵⁾

구간선형 시뮬레이션을 위해서는 회로의 동작 영역을 여러 개의 구간으로 나누고, 각 구간에서의 비선형 소자의 특성을 선형함수로 표현한다.¹¹⁶⁾ 회로의 동작은 구간선형 방정식의 형태로 주어지며, 추정해를 가정한 후 추정해가 속한 구간의 함수를 이용하여 해를 구한다. 만약 그렇게 구한 해가 추정해와 같은 구간에 존재하면 이는 실제해이고, 그렇지 않으면 다음 추정해를 선택하여 해를 구하는 과정을 되풀이한다. 따라서 이러한 반복 계산에서 얼마나 빨리 실제해에 수렴하는가 하는 것이 구간선형 시뮬레이션의 해석 효율을 결정하게 된다. 구간선형 방정식의 해를 구하는 대표적인 방법으로 구간선형 Newton-Raphson 알고리즘¹¹⁷⁾과 Katzenelson 알고리즘^{13, 18, 19)}이 있다. 구간선형 Newton-Raphson 알고리즘은 해를 구하는 과정에서 수렴이 보장되지 않으나, Katzenelson 알고리즘은 구간선형함수가 전단사이기만 하면 임의의 추정해에 대해서 수렴이 보장되므로 널리 사용되고 있다. 하지만 추정해를 포함하는 구간의 경계면에서 다음 추정해를 구하므로 Katzenelson 알고리즘은 실제해를 찾기 위해서 많은 반복 계산을 필요로 한다. 따라서 많은 구간으로 나뉘어지는 대형회로의 방정식을 해석하기에는 비효율적이다. 하지만 지금까지는 Katzenelson 알고리즘의 효율을 개선하기 보다는 복수의 해를 가지는 구간선형 방정식의 해들을 구하는 알고리즘을¹⁾개발하는데 많은 노력을 기울여 왔다.

1) Katzenelson 알고리즘은 구간선형 방정식의 해가 하나일 경우 이를 구하는 알고리즘임.

^{20, 21, 22)}. 본 논문에서는 Katzenelson 알고리즘의 효율성 단점을 보완하여, 대형회로도 효율적으로 해석할 수 있는 구간경계 뛰어넘기 알고리즘'을 제안한다. 본 논문의 구성은 다음과 같다. II 장에서는 본 논문에서 제안하는 알고리즘의 설명에 필요한 정의와 정리를 기술하고, III 장에서는 제안 알고리즘과 정규 구간선형 표현식에서의 응용 방법에 대하여 기술한다. IV 장에서는 시험 회로의 해석을 통하여 제안 알고리즘과 Katzenelson 알고리즘의 성능을 비교한후, V 장에서 결론을 내린다.

II. 정의와 정리

본 장에서는 제안 알고리즘과 그 수렴 조건을 유도하기 위해 필요한 정의와 정리들을 기술한다.

정의 1 구간선형함수^[14]:

함수 $f: R^n \rightarrow R^n$ 이 선형 분할을 갖는 구간선형함수일 필요충분조건은 다음과 같다.

1. 정의구역 R^n 은 유한개의 n 차원 다면체 구간 $R^{(1)}, R^{(2)}, \dots, R^{(s)}$ 으로 나누어진다. 두 다면체 구간사이의 경계는 다음 방정식으로 주어지는 $(n-1)$ 차원의 평면의 부분집합으로서 $(n-2)$ 차원의 평면에 포함되지 않아야 한다:

$$(\alpha_m, x) - \beta_m = 0$$

$\alpha_m, x \in R^n, \beta_m \in R^1, m=1, 2, \dots, \sigma$ 이며, " $<, >$ "는 내적을 나타낸다. σ 는 경계면의 수이고 α_m 과 β_m 는 경계면 m 을 특징짓는 상수이다.

2. 함수 f 는 $x \in R^{(k)}$ 에 대하여 다음과 같은 선형특성을 가져야 한다:

$$f(x) = J^{(k)}x + w^{(k)} \tag{1}$$

여기서 $J^{(k)}$ 는 구간 $R^{(k)}$ 내의 x 에 대한 f 의 Jacobian 행렬이며, $w^{(k)} \in R^n$ 는 상수이다.

3. 구간선형함수 f 는 이웃한 두 구간 $R^{(k)}$ 와 $R^{(j)}$ 사이의 경계면에서 연속이다. 즉, $x \in (R^{(k)} \cap R^{(j)})$, $k \neq j$,에 대하여 다음을 만족하여야 한다:

$$J^{(k)}x + w^{(k)} = J^{(j)}x + w^{(j)}$$

정의 2 선형 분할을 갖는 정규(canonical) 구간 선형 표현식^[14, 23].

선형 분할을 갖는 정규 구간선형 표현식 $f: R^n \rightarrow R^n$ 은 다음과 같이 표현될 수 있다:

$$f(x) = a + Bx + \sum_{m=1}^{\sigma} c_m |S_m(x)| \tag{2}$$

여기서 $B \in R^{n \times n}, m=1, 2, \dots, \sigma$ 에 대하여 $\alpha, c_m, \alpha_m, \in R^n, \beta_m \in R^1$ 이고, $S_m(x)$ 는 다음과 같이 주어진다:

$$S_m(x) = (\alpha_m, x) - \beta_m \tag{3}$$

$|S_m(x_i)|$ 는 x_i 에서 경계면 m 까지의 수직거리이며, 동작영역은 경계면 m 을 중심으로 $S_m(x) < 0$ 와 $S_m(x) > 0$ 인 영역으로 나누어진다.

x 를 미지수로 하는 구간선형 방정식

$$f(x) = y \tag{4}$$

에 대해 다음의 보조정리가 성립한다.^[19]

보조정리 1 Lipschitz condition^[19] :

$f: R^n \rightarrow R^n$ 는 연속이므로, 만약 식(4)가 모든 $y \in R^n$ 에 대하여 유일한 해를 가지면 역함수 f^{-1} 는 다음의 Lipschitz condition을 만족한다:

$$\|f^{-1}(p) - f^{-1}(q)\| \leq M \|p - q\|$$

여기서 M 은 임의의 유한 상수이다.

III. 제안 알고리즘과 정규 구간선형 표현식에서의 응용

본 장에서는 구간선형 방정식의 해를 구하는 기존의 알고리즘에 대하여 기술한다. 또 제안 알고리즘을 유도한 후, 이를 정규 구간선형 방정식에 적용하는 방법을 기술한다. 임의의 i 번째 추정해 x_i 를 이용하여 구간선형 방정식(4)의 해를 구하는 과정은 다음과 같다. 정의 1에 의해 추정해 x_i 가 구간 R^n 에 속할때 다음 등식이 성립한다:

$$f(x_i) = J^{(m)}x_i + w^{(m)}$$

식 (4)의 가상해 x' 도 구간 R^n 에 속한다고 가정하면 다음 등식이 성립한다:

$$y = J^{(m)}x' + w^{(m)}$$

위 두 식으로부터 x' 은 다음 식으로 주어진다:

$$x' = x_i + J^{(m)-1} \{y - f(x_i)\}$$

만약 위 식에서 구한 x' 이 가정한 바와 같이 $R^{(m)}$ 에 존재하면, 이는 구간선형 방정식의 실제해이다. 만약 x' 이 $R^{(m)}$ 밖에 존재하면, 다른 추정해를 선정 한 다음 x' 을 구하는 과정을 되풀이 한다.

구간선형 Newton-Raphson 알고리즘과 Katzenelson 알고리즘은 x' 이 $R^{(m)}$ 밖에 존재할 때

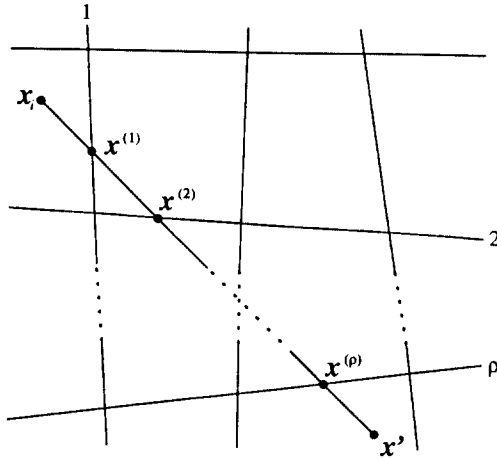


그림 1. 구간경계와 선분 $\overline{x_i x'}$ 의 교점들

Fig. 1. Intersections of region boundaries and line segment $\overline{x_i x'}$.

다음 추정해를 선택하는 방법에 차이가 있다.¹¹⁶⁾ 구간선형 Newton-Raphson 알고리즘은 식(5)에서 구간 x' 을 다음 계산을 위한 추정해로 선택을 한다. Katznelson 알고리즘은 추정해 x 와 x' 을 연결하는 선분이 구간 $R^{(m)}$ 의 경계면과 만나는 점을 다음 계산을 위한 추정해로 선정을 한다. 그림 1에 이를 나타내었다. 그림에서와 같이 선분 $\overline{x_i x'}$ 가 구간경계면들과 만나는 교점들을 $x^{(1)}, \dots, x^{(p)}$ 로 나타내면 Katznelson 알고리즘은 $x^{(1)}$ 을 다음 추정해로 선택을 한다. 구간선형 Newton-Raphson 알고리즘은 같은 구간들을 반복해서 찾아가 해에 수렴하지 않는 경우도 있다. 이에 비하여, Katznelson 알고리즘은 구간선형함수 f 가 전단사함수이면 식(4)의 실제해로의 수렴이 보장되는 장점이 있으나, 수렴 속도가 느려서 동작영역이 많은 구간으로 나뉘어 있는 대형회로의 경우에 계산 효율이 떨어지는 단점이 있다.

y 와 임의의 점 x 의 함수값 $f(x)$ 와의 차이를 $E_n(x)$ 로 아래와 같이 정의하자:

$$E_n(x) = \|y - f(x)\| \quad (6)$$

추정해를 사용하여 식(4)의 해를 구하는 과정을 거듭함에 따라 실제해에 수렴하기 위해서는 각 과정에서

$$E_n(x_{i+1}) < E_n(x_i) \quad (7)$$

을 만족하여야 한다. 여기서 x_{i+1} 는 $(i+1)$ 번째 추정해이다. Katznelson 알고리즘의 경우와 같이 그림 1

의 $x^{(1)}$ 을 다음 추정해로 선택할 경우, 모든 반복 계산 과정에서 식(7)을 만족하여 실제해에 수렴하게 된다.¹²⁴⁾ 전술한 바와 같이, Katznelson 알고리즘은 구간선형함수 f 가 전단사함수라는 것을 수렴의 충분조건으로 하고 있다. 구간선형함수 f 가 전단사함수이면, 보조정리1에 의하여 역함수 f^{-1} 는 다음의 Lipschitz condition을 만족한다:

$$\|f^{-1}(f(x^*)) - f^{-1}(f(x_{i+1}))\| \leq M \|f(x^*) - f(x_{i+1})\|$$

위에서 x^* 가 식(4)의 실제해라면,

$$\|x^* - x_{i+1}\| \leq M \|y - f(x_{i+1})\| \quad (8)$$

이므로 $f(x_{i+1})$ 이 y 에 가까울수록 추정해 x_{i+1} 도 실제해 x^* 에 더 접근함을 알 수 있다. 따라서 그림 1의 $x^{(2)}, \dots, x^{(p)}$ 중에서 $x^{(1)}$ 보다 작은 $Err(x)$ 를 가지는 교점을 다음 추정해 x_{i+1} 로 선택한다면 Katznelson 알고리즘보다 효율적으로 실제해에 수렴할 수 있다. 제안 알고리즘은 greedy 알고리즘¹²⁵⁾의 일종으로, 각 반복 계산 과정에서 실제해인 x^* 에 가장 가까운 교점을 다음 추정해로 선택하는 것을 기본 생각으로 하고 있다. 즉, 제안 알고리즘은 $x^{(1)}$ 에 비하여 작거나 같은 크기의 $Err(x)$ 를 가지는 교점을 다음 추정해로 선택하며, 따라서 모든 반복 계산 과정에서 식(7)을 만족하므로 Katznelson 알고리즘과 마찬가지로 구간선형함수 f 가 전단사함수이면 수렴을 하게 된다. 회로가 단일변수 구간선형함수로 표현되는 두단자 저항, 독립전원, 종속전원 및 선형소자들로 구성된다고 가정하자. 이 경우 Nodal 해석, Mesh 해석, Loop 해석, Cutset 해석, Hybrid 해석, Modified Nodal 해석 또는 Tableau 해석방법을 사용하여 회로 방정식을 세웠을때 식(2)와 같은 정규 구간선형식으로 표현할 수 있음이 알려져 있다.¹¹⁸⁾ 다음은 제안 알고리즘을 이용하여 정규 구간선형 방정식의 해를 효율적으로 구하는 방법을 기술하기로 한다.

식(5)에 의해 x' 을 구하면, 먼저 x' 이 실제해인지 조사할 필요가 있다. 이는, 정규 구간선형 표현식의 경우, x 가 경계면 m 을 지날때 $S_m(x)$ 의 부호가 바뀐다는 성질을 이용하여 간단히 조사를 할 수 있다. $S_m(x)$ 는 식(3)에서 정의하였다:

· 각 경계면 m 에 대하여 $S_m(x')$ 을 계산한다. 만약 모든 m 에 대하여 $\text{Sgn}(S_m(x')) = \text{sgn}(S_m(x_i))$ 이면²⁾, x' 와 x_i 는 같은 구간 내에 있으

2) $x > 0$ 이면 $\text{sgn}(x) = 1$, $x < 0$ 이면 $\text{sgn}(x) = -1$ 이다. $\text{sgn}(0)$ 는 정의되지 않는다.

므로 x' 는 실제해이다. $S_m(x_i)$ 는, 식(5)를 이용하여 x' 을 구하는 과정에서 $f(x_i)$ 를 계산해야 하며, 이를 위해서 식(2)를 이용할 때 $S_m(x_i)$ 를 계산하여야 하므로 이를 사용하면 된다.

만약 $Sgn(S_m(x')) \neq sgn(S_m(x_i))$ 인 경계면 m 이 존재하면 이는 x' 과 x_i 사이에 존재하는 경계면이다.

x' 이 실제해가 아니며, 그림 1에서와 같이 x' 과 x_i 사이에 ρ 개의 경계면이 존재한다고 하자. 제안 알고리즘은 이들 경계면과 선분 $x_i x'$ 의 교점 중에서 최소의 $ETr(x)$ 를 갖는 점을 다음 추정해로 선택한다. 이를 구하기 위해서는, 각 교점 x 를 구한 후, 식(2)에 의하여 $f(x)$ 를 계산하여야 하는데, 제안 알고리즘의 효율은 이와 같은 부가 계산량에 크게 의존한다. 본 논문에서 제안하는 알고리즘은 각 교점에서의 x 와 $f(x)$ 를 구하지 않고, x_i 와 x' 만으로 각 교점에서의 $ETr(x)$ 를 먼저 계산하여, 최소치 $ETr(x)$ 를 가지는 교점에 대해 x 를 계산하여 그것을 다음 추정해로 한다. 이를 위해 부가 계산을 극소화하는 $ETr(x)$ 의 수식을 유도 제시한다.

임의의 두 점 x_1 과 x_2 가 같은 구간내에 있는 경우 $S_m(x_1)$ 과 $S_m(x_2)$ 의 부호가 같다. 따라서 $sgn(\)$ 를 사용하면, x_i 가 속해 있는 구간의 선형 방정식 $F(x)$ 는 다음과 같이 주어진다:

$$F(x) = a + Bx + \sum_{m=1}^{\sigma} c_m |S_m(x)|$$

$$= a + Bx + \sum_{m=1}^{\sigma} c_m sgn(S_m(x_i)) S_m(x)$$

$x^{(j)}$ 과 $x^{(p)}$ 사이의 임의의 교점 $x^{(k)}$ 를 고려하면, 이는 경계면 S_k 상에 존재하므로 $S_k(x_k) = 0$ 이다. 따라서 $x^{(p)}$ 에 대해 $F(x^{(p)})$ 는 다음과 같이 주어진다:

$$F(x^{(k)}) = a + Bx^{(k)} + \sum_{m=1}^{k-1} c_m sgn(S_m(x_i)) S_m(x^{(k)})$$

$$+ \sum_{m=k+1}^{\sigma} c_m sgn(S_m(x_i)) S_m(x^{(k)})$$

한편 $F(x^{(p)})$ 는 아래와 같이 나타낼 수 있다:

$$f(x^{(k)}) = a + Bx^{(k)} + \sum_{m=1}^{k-1} c_m sgn(S_m(x^{(k)})) S_m(x^{(k)})$$

$$+ \sum_{m=k+1}^{\sigma} c_m sgn(S_m(x^{(k)})) S_m(x^{(k)})$$

위 두 식으로부터,

$$F(x^{(k)}) - f(x^{(k)}) = \sum_{m=1}^{k-1} c_m [sgn(S_m(x_i)) - sgn(S_m(x^{(k)}))] S_m(x^{(k)})$$

$$+ \sum_{m=k+1}^{\sigma} c_m [sgn(S_m(x_i)) - sgn(S_m(x^{(k)}))] S_m(x^{(k)}) \quad (9)$$

x_i 와 $x^{(k)}$ 사이의 경계면 $m \in \{1, 2, \dots, k-1\}$ 에 대해서는

$$sgn(S_m(x_i)) = -sgn(S_m(x^{(k)}))$$

인 관계가 성립하고 경계면 $m \in \{k+1, k+2, \dots, \sigma\}$ 에 대해서는 다음의 관계가 성립한다.

$$sgn(S_m(x_i)) = sgn(S_m(x^{(k)}))$$

그러므로 식(9)는 다음과 같이 간단한 수식으로 정리된다:

$$F(x^{(k)}) - f(x^{(k)}) = -2 \sum_{m=1}^{k-1} c_m [sgn(S_m(x^{(k)})) S_m(x^{(k)})]$$

$$= -2 \sum_{m=1}^{k-1} c_m |S_m(x^{(k)})|$$

그러므로 $y - f(x^{(k)})$ 는 다음과 같이 표현된다:

$$y - f(x^{(k)}) = y - F(x^{(k)}) - 2 \sum_{m=1}^{k-1} c_m |S_m(x^{(k)})| \quad (10)$$

위 식에서 우항의 $H(x^{(k)})$ 항을 소거하기 위하여, 길이 비를 나타내는 v_k 를 다음과 같이 정의하면,

$$v_k = \frac{\lambda \frac{|x^{(k)} - x'|}{|x_i x'|}}{|x_i x'|} = \frac{S_k(x')}{S_k(x') - S_k(x_i)}$$

$y - F(x^{(k)})$ 와 $S_m(x^{(k)})$ 는 다음과 같이 주어진다:

$$y - F(x^{(k)}) = v_k (y - F(x_i)) = v_k (y - f(x_i))$$

$$|S_m(x^{(k)})| = \frac{v_m - v_k}{v_m} |S_m(x')| \quad (11)$$

따라서 위 두 식을 식 (10)에 대입하면 다음 최종 식을 얻는다:

$$E_{\rho}(x^{(k)}) = \|y - f(x^{(k)})\|$$

$$= \left\| v_k (y - f(x_i)) - 2 \sum_{m=1}^{k-1} c_m \frac{v_m - v_k}{v_m} |S_m(x')| \right\| \quad (12)$$

위 식에서 $f(x_i)$, v_k 와 $S_m(x')$ 은 x' 과 교차 경계면들을 찾는 과정에서 계산되어 있다. 따라서 위 식을 사용하면 $x^{(k)}$ 를 구하지 않고 이미 계산된 값들을 이용하여 $ETr(x)$ 의 최소치를 갖는 교점을 찾아 낼 수 있다. $x^{(k)}$ 가 $ETr(x)$ 의 최소치를 갖는다면 다음 추정해 $x_{\rho+1}$ 는 다음 식으로부터 얻는다:

$$x_{i+1} = x^{(k+1)} = x' - v_k(x' - x_i) \quad (13)$$

IV. 실험 및 고찰

식(12)에 의하면 $x^{(k)}$ 가 x_i 로부터 멀수록 $Err(x)$ 를 구하기 위한 계산량이 증가한다. 따라서 본 연구에서는 다음과 같이 두가지 방법으로 제안 알고리즘을 구현하였다.

방법 1 (Global minimum method) : $x^{(1)}$ 과 $x^{(n)}$ 사이의 구간경계 교점들 중에서 가장 작은 $Err(x)$ 를 가지는 교점을 다음 추정해로 한다.

방법 2 (Local minimum method) : 추정해 x_i 에 가장 가까운 교점 $x^{(1)}$ 부터 $Err(x)$ 를 계산하다가 $Err(x)$ 가 증가하면 계산을 멈추고 그 전의 교점을 다음 추정해로 선택한다.

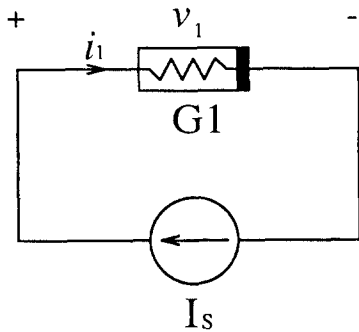


그림 2. 시험 회로 1
Fig. 2. Test circuit 1.

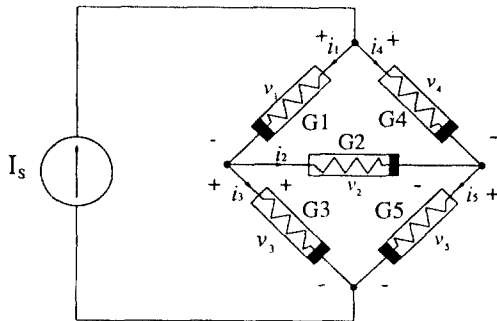


그림 3. 시험 회로 2
Fig. 3. Test circuit 2.

여기서 $Err(x)$ 는 식(12)에서 L_2 norm(유클리디안 거리)으로 계산했다. 방법 1은 방법 2에 비하여 실제 해를 찾는 데 소요되는 전체 반복 계산의 회수는 줄 것으로 기대되지만, $Err(x)$ 의 계산으로 인하여 해석 시간은 더 많이 소요될 수도 있다. 제안 알고리즘은 UNIX³⁾ 운영 체제에서 돌아가는 SUN⁴⁾ Sparc 10 컴퓨터에서 C 언어로 구현하였다.

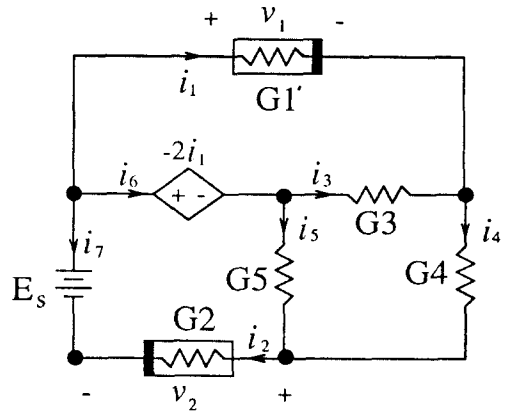


그림 4. 시험 회로 3
Fig. 4. Test circuit 3.

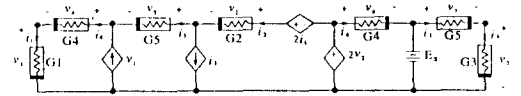


그림 5. 시험 회로 4
Fig. 5. Test circuit 4.

제안 알고리즘과 Katzenelson 알고리즘을 비교하기 위한 시험회로들을 그림 2, 3, 4, 5에 보였다. 이중 그림 3, 4, 5의 시험 회로는 참고 문헌 [18]에서 사용한 회로들이다. 회로의 구간선형 저항들의 전류, 전압 관계는 다음과 같이 정규 구간선형 표현식으로 표현된다.

$$i_k = a_k + b_k v_k + \sum_{m=1}^{\sigma_k} c_{km} |v_k - V_{km}|$$

여기서 i_k 는 저항을 통해 흐르는 전류이고, v_k 는 저항 양단의 전압이며, σ_k 는 구간경계의 수이다. 회로 방정식은 Modified Nodal Analysis 방법을 이용하여 구하였다. 행렬의 차원의 크기는 회로의 노드 수와 Nodal Analysis로 취급할 수 없는 소자의 수에 의해 결정된다. 각 회로의 특성을 표 1에 보였다. 표

3) UNIX는 AT & T사의 등록상표임.
4) SUN은 SUN Microsystems사의 등록상표임.

에서 n 은 회로 방정식의 행렬 차원의 크기를 나타내고, σ 는 회로 방정식의 구간 경계면의 수를 나타낸다.

표 1. 시험회로의 특성과 필요한 해석 회수
Table 1. Characteristics of test circuits and numbers of matrix solving computations required for consistent result.

	시험회로 1	시험회로 2	시험회로 3	시험회로 4
행렬 차원의 크기 (n)	1	3	8	15
구간 경계면의 수 (σ)	101	307	130	341
해석 회수	120	31,993	31,996	33,245

Katzenelson 알고리즘과 제안 알고리즘의 해석 속도의 비율은 최초 구간의 선택에 따라서 달라질 수 있다. 그러므로 정확한 비교를 위해서는 모든 구간에서 최초 추정해를 선택하여 해석한 다음, 평균 CPU 시간을 비교하여야 한다. 하지만 시험회로의 방정식의 구간 수는 각 경계면의 기하학적 관계에 따라서 달라지고 수식적으로 정확하게 구할 수 없는 문제가 있다. 따라서 해석 시간의 평균비가 안정될 때까지 임의의 추정해를 무작위로 선정하여서 해석하기로 하였다. 시험 회로 4를 임의의 추정해를 사용하여 해석한 후, 해석 회수에 따른 해석 시간의 평균 비율이 변화하는 모습을 그림 6에 나타내었다. 그림에서 X축은 해석 회수를 나타내고, Y축은 Katzenelson 알

고리즘에 대한 제안 알고리즘의 해석 시간과 반복 계산 회수의 향상 비율을 나타낸다. 또 실선은 방법 1의 경우를, 점선은 방법 2의 경우를 나타낸다. 시험 회로 4의 경우에는 해석 수가 100을 넘으면서 Katzenelson 알고리즘에 대한 제안 알고리즘의 향상 비율이 안정되기 시작하며, 해석 수가 10,000 회가 되면 거의 변화가 없는 것을 알 수 있다. 이와 같이 해석 속도를 비교하기 위한 해석 회수는 평균 향상 비율의 변화가 1% 이내가 되도록 하였다. 평균 향상 비율을 얻기 위해 각 시험 회로에 수행한 해석 회수를 표 1에 나타내었다.

각 시험회로들을 Katzenelson 알고리즘과 제안 알고리즘을 이용하여 해석한 후 반복 계산의 회수와 해석 시간을 비교한 결과를 표 2, 3, 4, 5에 보였다. 표 2에서는 각 시험 회로에 대하여 Katzenelson 알고리즘과 제안 알고리즘 방법 1이 해에 수렴하기 위해 필요한 반복 계산 회수를 비교하였다. 표에서 "평균 반복 계산 회수"는 평균 향상 비율이 안정되었을 때의 평균 반복 계산 회수를 나타내고, "최고 효율"과 "최저 효율"은 평균 향상 비율이 안정될 때까지 있었던 개선 정도가 가장 높을 때와 가장 낮을 때의 반복 계산 회수를 나타낸다. 표를 살펴 보면 제안 알고리즘 방법 1은 Katzenelson 알고리즘에 비하여 실제해에 수렴하기 위해 필요한 반복 계산의 회수가 평균적으로 1/15 이하로 감소하였으며, 시험 회로 4의 경우 최고 55 배 빨리 수렴함을 알 수 있다. 최저 효율의 경우를 살펴 보면 두 알고리즘이 같은 회수의 반복 계산을 필요로 한 경우도 있으나, 이는 2 회 또는 3 회만 반복 계산을 필요로 할 정도로 최초 추정해가 실제해에 가까운 경우이다.

표 3에서는 Katzenelson 알고리즘과 제안 알고리즘 방법 1의 해석 시간을 비교하였다. $Err(x^{(k)})$ 를 구하기 위한 부가 계산으로 인하여 반복 계산 회수의 경우보다 효율이 낮아 지나, 제안 알고리즘은 Katzenelson 알고리즘에 비하여 평균적으로 6 배 이상 빨리 수렴하는 것을 알 수 있다. 시험 회로 1의 경우에는 최고 및 최저 효율은 해석 시간이 너무 짧아서 비교가 불가능하였다.

표 4와 5에서는 각 시험 회로에 대하여 Katzenelson 알고리즘과 제안 알고리즘의 방법 2가 해에 수렴하기 위해 수행한 반복 계산 회수와 해석 시간을 비교하였다. 표 4를 보면, 반복 계산 회수의 경우 방법 2는 Katzenelson 알고리즘에 비하여 평균적으로 약 14 배 빨리 수렴하였음을 알 수 있다. 또 방법 1과 비교할 때 방법 2는 $Err(x)$ 의 국부 최소치를 이용하므로 방법 1보다 개선 정도가 낮아 지

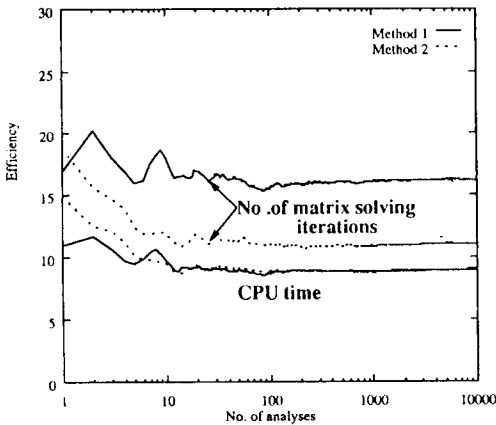


그림 6. Katzenelson 알고리즘에 대한 제안 알고리즘의 평균효율 변화(시험회로 4)

Fig. 6. Average improvement of the proposed algorithm over the Katzenelson algorithm using test circuit 4.

는 결과를 보였다. 해석 시간의 경우, 표 5를 보면 Katzenelson 알고리즘에 비하여 평균적으로 약 7.5 배 빨리 수렴을 하였으며, $Err(x)$ 를 계산하는 양이 감소하므로 방법 1보다 나은 결과를 나타내었다.

표 2. Katzenelson 알고리즘과 제안 알고리즘 (방법 1)의 반복 계산 회수 비교

Table 2. Comparison of numbers of matrix solving computations by the Katzenelson algorithm and the proposed algorithm(Method 1).

Katz.: Katzenelson algorithm

	평균 반복 계산 회수			최고 회수			최저 회수		
	Katz	방법 1	회율	Katz	방법 1	회율	Katz	방법 1	회율
시험회로 1	47.4	2.53	19.62	75	2	37.5	2	2	1
시험회로 2	97.95	5.52	18.18	140	2	70	11	3	4.67
시험회로 3	42.55	5.22	8.25	74	2	37	3	3	1
시험회로 4	167.18	10.71	16.19	341	6	56.87	35	8	1.48
평균			15.56						

표 3. Katzenelson 알고리즘과 제안 알고리즘 (방법 1)의 해석 시간 비교

Table 3. Comparison of CPU times spent by the Katzenelson algorithm and the proposed algorithm(Method 1).

Katz.: Katzenelson algorithm

	평균 해석 시간 (Sec.)			최고 회율			최저 회율		
	Katz	방법 1	회율	Katz	방법 1	회율	Katz	방법 1	회율
시험회로 1	0.021	0.0048	1.38						
시험회로 2	0.27	0.05	6.23	0.28	0.01	28	0.18	0.08	2.25
시험회로 3	0.21	0.05	1.82	0.32	0.01	32	0.01	0.01	1
시험회로 4	6.04	0.68	9.15	6.41	0.26	24.65	2.92	0.38	4.52
평균			6.145						

표 4. Katzenelson 알고리즘과 제안 알고리즘 (방법 2)의 반복 계산 회수 비교

Table 4. Comparison of numbers of matrix solving computations by the Katzenelson algorithm and the proposed algorithm(Method 2).

Katz.: Katzenelson algorithm

	평균 반복 계산 회수			최고 회율			최저 회율		
	Katz	방법 2	회율	Katz	방법 2	회율	Katz	방법 2	회율
시험회로 1	47.4	2.52	19.61	75	2	37.5	2	2	1
시험회로 2	97.95	5.74	17.55	140	2	70	11	3	4.67
시험회로 3	42.55	5.33	8.46	74	2	37	3	3	1
시험회로 4	167.18	16.21	10.98	341	5	52.8	149	55	2.71
평균			11.05						

표 5. Katzenelson 알고리즘과 제안 알고리즘 (방법 2)의 해석 시간 비교

Table 5. Comparison of CPU times spent by the Katzenelson algorithm and the proposed algorithm(Method 2).

Katz.: Katzenelson algorithm

	평균 해석 시간 (Sec.)			최고 회율			최저 회율		
	Katz	방법 2	회율	Katz	방법 2	회율	Katz	방법 2	회율
시험회로 1	0.021	0.0044	1.77						
시험회로 2	0.27	0.04	10.51	0.32	0.01	32	0.05	0.02	2.7
시험회로 3	0.21	0.04	5.76	0.32	0.01	32	0.01	0.02	0.5
시험회로 4	6.04	0.69	9.61	10.99	0.12	26.17	5.28	2.09	2.54
평균			7.548						

실험 결과를 요약하면 다음과 같다. 시험회로에 대하여 제안 알고리즘은 Katzenelson 알고리즘보다 수렴에 필요한 반복 계산 회수는 1/20 1/8로, 해석 시간은 1/10~1/5로 크게 감소시킴으로써 효율적으로 구간선형 시스템을 해석할 수 있음을 보였다. 제안 알고리즘이 Katzenelson 알고리즘보다 늦게 수렴하는 경우는 나타나지 않았다. 방법 1과 2를 비교한 결과, 반복 계산 회수의 면에서는 방법 1이, 해석 시간의 면에서는 방법 2가 더 좋은 결과를 보였다. 또 구간경계면의 수가 증가함에 따라 제안 알고리즘의 Katzenelson 알고리즘에 대한 효율도 향상되는 것으로 판단된다. 표에서 구간 경계면이 100 개 부근의 시험 회로 1과 3의 경우에는 속도 향상이 5 배정도인데, 구간 경계면의 수가 300 개 이상인 시험 회로 2와 4의 경우에는 6 배에서 10 배의 해석 속도의 향상을 얻을 수 있었다. 이는 구간경계의 수가 증가할수록 그만큼 뛰어 넘는 구간의 수도 많아져 반복 계산 회수가 감소하기 때문이다. 또 선형 방정식 $Ax = b$, $A \in R^{n \times n}$, $x, b \in R^n$ 의 해를 구하는 계산의 복잡도는 n^3 에 비례하여 증가하지만, $Err(x^{(k)})$ 를 계산하는 복잡도는 식(12)에서 나타난 바와 같이 선형적으로 증가하는 장점을 지니고 있다. 따라서 시스템의 차원이 커질수록 방정식을 푸는 계산량에 비하여 $Err(x^{(k)})$ 를 구하는 계산량은 무시될 수 있으며, Katzenelson 알고리즘에 대한 제안 알고리즘의 효율도 따라서 증가한다.

V. 결론

Katzenelson 알고리즘은 구간선형함수 f 가 전단사 함수일 경우 항상 실제해에 수렴을 하며, 가장 널리 사용되고 있는 방법중의 하나이다. 하지만 가상해를 구한 후, 현재 추정해를 포함하고 있는 구간의 경

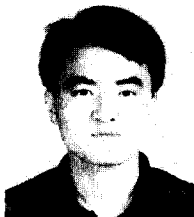
계면에서 다음 추정해를 선택하므로 수렴이 느리며, 그로 인해 대형회로의 해석에 사용하기에는 비효율적이다. 본 논문에서는 Katzenelson 알고리즘과 같은 충분한 수렴조건을 가지면서, 효율적으로 구간선형 방정식의 해에 수렴할 수 있는 구간경계 뛰어넘기 알고리즘을 제안하였다. 제안 알고리즘은 greedy 알고리즘의 일종으로, 가상해를 구하기 위한 추정해를 선택하는 과정에서 추정해와 가상해 사이의 구간경계 교점 중 실제해에 가장 가까운 교점을 다음 추정해로 정함으로써 효율적으로 해에 수렴할 수 있도록 하였다. 제안 알고리즘을 선형 분할을 갖는 정규 구간선형 방정식에 적용하여, 다음 추정해를 효율적으로 구하는 방법을 제시하였다. 이를 시험회로에 적용하여 본 결과 Katzenelson 알고리즘에 비하여 반복 계산의 회수는 1/20~1/8로 감소하였으며, 해석 시간은 1/10~1/5로 감소함을 확인하였다. 또 시험회로의 구간 경계면의 수가 많을 수록 효율이 향상됨을 확인하였다. 제안 알고리즘의 방법 1과 2를 비교하면, $I_{rr}(x)$ 의 국부 최소치를 사용한 방법 2가 해석 시간 면에서 방법 1보다 우수한 결과를 나타내었다. 본 논문에서 제안한 알고리즘은 회로의 해석뿐 아니라 선형 분할을 갖는 기타 구간 선형시스템의 해석에도 응용이 가능하다.

参 考 文 献

- [1] S.M. Rubin, *Computer Aids for VLSI Design* Addison-Wesley, 1987.
- [2] L. W. Nagel, *SPICE2: A computer program to simulate semiconductor circuits*. Memorandum No. UCB/ERL M520, University of California, Berkeley, May 1975.
- [3] W. T. Weeks, A. J. Jimenez, G. W. Mahoney, D. Mehta, H. Qassemzadeh, and T. R. Scott, "Algorithms for ASTAP-A network-analysis program." *IEEE Trans. Circuit Theory*, vol. CT-10, pp. 628-634, Nov. 1973.
- [4] A. R. Newton, "Techniques for the simulation of large-scale integrated circuits." *IEEE Trans. Circuits Syst.*, vol. 26, pp. 741-749, Sept. 1979.
- [5] A. R. Newton, *The Simulation of large-scale integrated circuits*. Memorandum No. UCB/ERL M78/52, University of California, Berkeley, July 1978.
- [6] B. R. Chawla, H. K. Gummel, and P. Kozak, "MOTIS-An MOS timing simulator." *IEEE Trans. Circuits Syst.*, vol. CAS-22, pp. 901-910, Dec. 1975.
- [7] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*. New York: Academic Press, 1970.
- [8] A. P. E. Acuna, J.P. Dervenis and R. Saleh, "iSPICE3: A new simulator for mixed analog/digital circuits." in *Custom Integrated Circuit Conference*, May 1989.
- [9] A. R. Newton and A. L. Sangiovanni-Vincentelli, "Relaxation-based electrical simulation." *IEEE Trans. Electron Devices*, vol. ED-30, pp. 1184-1207, Sept. 1983.
- [10] V. B. Rao, T. N. Trick, and I. N. Hajj, "A table-driven delay-operator approach to timing simulation of MOS VLSI circuits." in *Proc. Int. Conf. Computer-Aided Design*, (New York), Nov. 1983, pp. 445-448.
- [11] D. Overhauser and I. Hajj, "A tabular macromodeling approach to fast timing simulation including parasitics." in *Proc. Int. Conf. Computer-Aided Design*, (Santa Clara), Nov. 1988, pp. 70-73.
- [12] J. Rubinstein, P. P. Jr., and M. A. Horowitz, "Signal delay in RC tree networks." *IEEE Trans. Computer-Aided Design*, vol. CAD-2, pp. 202-211, July 1983.
- [13] J. Katzenelson, "An algorithm for solving nonlinear resistive networks." *Bell System Tech. J.*, vol. 44, pp. 1605-1620, Oct. 1965.
- [14] L. O. Chua and A. Deng, "Canonical piecewise-linear representation." *IEEE Trans. Circuits Syst.*, vol. CAS-35, pp. 101-111, Jan. 1988.

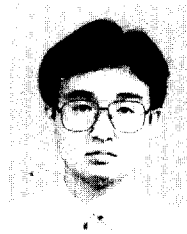
- [15] Saber/DesignStar Information. Analogy. 1989.
- [16] L. O. Chua and P. Lin, *Computer-Aided Analysis of Electronic Circuits*. Englewood Cliffs, NJ: Prentice-Hall, 1975.
- [17] L. O. Chua, "Efficient computer algorithms for piecewise-linear analysis of resistive nonlinear networks," *IEEE Trans. Circuit Theory*, vol. CT-18, pp. 73-85, Jan. 1971.
- [18] L. O. Chua and R. L. P. Ying, "Canonical piecewise-linear analysis," *IEEE Trans. Circuits Syst.*, vol. CAS-30, pp. 125-140, Mar. 1983.
- [19] T. Fujisawa and E. S. Kuh, "Piecewise-linear theory of nonlinear networks," *SIAM J. Appl. Math.*, vol. 22, pp. 307-328, Mar. 1972.
- [20] M. J. Chien and E. S. Kuh, "Solving nonlinear resistive networks using piecewise-linear analysis and simplicial subdivision," *IEEE Trans. Circuits Syst.*, vol. CAS-24, pp. 305-315, June 1977.
- [21] Q. Huang and R. Liu, "A simple algorithm for finding all solutions of piecewise-linear networks," *IEEE Trans. Circuits Syst.*, vol. 36, pp. 600-609, Apr. 1989.
- [22] K. Yamamura and M. Ochiai, "An efficient algorithm for finding all solutions of piecewise-linear resistive circuits," *IEEE Trans. Circuits Syst.*, vol. 39, pp. 213-221, Mar. 1992.
- [23] S. M. Kang and L. O. Chua, "A global representation of multi-dimensional piecewise-linear functions with linear partitions," *IEEE Trans. Circuits Syst.*, vol. ~CAS-25, pp. ~938-940, Nov. 1978.
- [24] M. Chien and E. S. Kuh, "Solving nonlinear resistive networks using piecewise-linear analysis and simplicial subdivision," *IEEE Trans. Circuits Syst.*, vol. CAS-24, pp. 305-317, June 1977.
- [25] G. Brassard and P. Bratley, *Algorithmics: theory and practice*. Prentice-Hall, 1988.

 著者紹介



趙俊暎(學生會員)

1971年 2月 20日生. 1993年 2月 포항공과대학 전자전기공학과 졸업 (공학사). 1993년 3월 ~ 현재, 동대학원 전자전기공학과 석사 과정 재학중. 주관심 분야는 회로 시뮬레이션, 상위 레벨 합성 등임.



趙振國(學生會員)

1970年 6月 5日生 1993年 2月 포항공과대학 전자전기공학과 졸업 (공학사). 1993년 3월 ~ 현재 동대학원 전자전기공학과 석사 과정 재학중. 주관심 분야는 회로 시뮬레이션, VLSI 설계 등임.

著 者 紹 介



權容世(正會員)

1959年 8月 14日生. 1983年 2月 한국항공대학 전자공학과 졸업(공학사). 1986年 2月 한국과학기술원 전기 및 전자공학과 졸업(공학석사). 1986年 3月~1989年 3月 국방부 사무관. 1989年 3月~현재 포항공과대학교 대학원 전자전기공학과 박사과정 재학중. 주관심 분야는 회로 시뮬레이션, 타이밍 검증, VLSI 설계 등임.



金穎煥(正會員)

1955年 4月 25日生. 1977年 2月 경북대학교 전자공학과 졸업(공학사). 1985年 12月~1988年 12月 미국 University of California, Berkeley 전기공학과 졸업(석사/박사). 1977年 3月~1982年 8月 국방과학연구소 연구원. 1983年 8月~1988年 12月 Electronic Research Lab. UCB, Post Graduate Researcher. 1988年 12月~현재 포항공과대학교 전자전기공학과 조교수. 주관심 분야는 시뮬레이션, 타이밍 검증, VLSI 설계 등임.