

선형 계획을 이용한 유도 정밀도 계수 FIR 필터의 설계

正會員 趙 南 翊*, 正會員 李 商 郁**

Design of FIR Filters with Finite Precision Coefficients
Using LP(Linear Programming)Nam Ik Cho*, Sang Uk Lee**, *Regular Members*

요 약

본 논문에서는 최적의 유한 정밀도 계수를 갖는 1차원 및 2차원 필터 설계 알고리즘을 제안하였다. 제안된 알고리즘은 선형 계획(LP-linear programming)을 반복적으로 사용하는 BaB(branch and bound) 알고리즘에 인접해 있는 SP(sub-problem)들의 최적해가 서로 매우 밀접한 관계가 있다는 사실에 기초를 두고 있다. 기존의 설계 알고리즘은 각 SP를 풀기 위하여 LP를 새로 수행한 반면에, 제안된 알고리즘은 인접한 SP들 사이의 관계를 이용하여 LP의 입력이 되는 주파수 제한조건의 수를 크게 줄인다. 따라서 기존의 방법에 비하여 전체적인 이산계수 FIR 필터 설계의 계산량이 줄어든다. 또한, 본 논문에서는 계수가 특히 2의 멱승의 합으로 주어지는 경우 기존의 방법보다 더욱 효율적인 가지치기 방법을 제안하였다. 이러한 가지치기 방법을 제안된 LP 해결 방법과 함께 이용함으로써 2의 멱승의 합으로 계수가 주어지는 필터의 설계 시간을 더욱 크게 줄일 수 있음을 보였다.

Abstract

In this paper, an optimal algorithm for the design of 1-D FIR filters with finite precision coefficients is proposed. The algorithm is based on the observation that the frequency constraints of a sub-problem(SP) in the branch and bound algorithm, which repeatedly use LP(linear programming), are closely related with those of neighboring SPs. By using the relationship between the SPs, the proposed algorithm reduces the number of constraints required for solving each SP with LP, whereas the conventional algorithm employs all the constraints, which are required for solving the initial problem. Thus, the overall computational load for the design of FIR filters with finite precision coefficients is significantly alleviated, compared to the conventional branch and bound algorithm. Also, a new branching scheme for the design of FIR filters with sum-of-power-of-two (SOPOT) coefficients is proposed. It is shown that the computational load for the design of SOPOT coefficient filters can be further reduced with the new branching scheme.

*서울市立大學校 制御計測工學科
Dept. of Control and Instrumentation Engineering, Seoul City University

**서울大學校 制御計測工學科
Dept. of Control and Instrumentation Engineering, Seoul National University

論文番號 : 94137
接受日字 : 1994年 5月 19日

I. 서 론

디지털 신호처리에서 FIR 필터의 설계와 구현은 가장 중요한 문제들 중의 하나이다. 필터 설계 문제에 있어서는 그 정밀도와 수행 속도가 가장 큰 부분을 차지한다. 즉, 디지털 필터를 하드웨어나 프로그램으로 구현할 때에는 계수들이 유한한 값으로 표현되는데 정밀성을 유지하기 위해서는 부동 소수점(floating point) 연산으로 필터링을 하지만 이 경우에는 하드웨어가 복잡해지거나 수행속도가 느려진다는 단점이 있다. 따라서 대부분의 경우, 특히 수행속도나 하드웨어 복잡성이 문제가 되는 경우에는 각 계수들과 입력의 곱은 고정 소수점(fixed point) 연산으로 수행된다. 또한 2차원 신호처리, 특히 HDTV 등에서와 같이 더욱 빠른 수행 속도가 요구되는 경우에는 수행속도와 하드웨어 복잡성이 더욱 큰 문제가 된다. 이와 같은 경우 계산 효율을 더 높이는 방법들 중의 하나가 필터의 계수를 2의 멱승의 합(SOPOT: sum-of-power-of-two)의 형태로 나타내는 것이다. 즉, 각 필터 계수는 둘 내지 세개의 비트만 1 또는 -1의 값을 가질 수 있으며 나머지는 모두 0으로 함으로써 곱셈을 몇개의 쉬프트와 덧셈으로 대체하는 것이다[1].

디지털 필터를 설계할 때, 일반적인 방법은 최적의 실계수 필터를 구하고 각 계수를 가장 가까운 정수값 또는 2의 멱승의 합으로 절삭 또는 반올림 하는 것이다. 그러나 이 경우 발생하는 작은 절삭 또는 반올림 오차가 필터의 특성을 심하게 저하시키는 경우가 있으며, 양자화 간격이 큰 2의 멱승 합인 경우는 이러한 현상이 더욱 심해진다. 따라서 계수가 정수 또는 2의 멱승의 합을 갖는 유한 정밀도 계수 FIR 필터의 최적 설계에 관한 연구가 많이 수행되어 왔다[1]-[7]. 이들 중 가장 기본적인 방법이 BaB(Branch and Bound) 알고리즘으로서, 그 계산 예가 그림 1에 나타나 있다. 여기서, 결정해야 할 필터의 계수가 N 개이며, 각 원이 선형 계획(LP-linear programming) 또는 기타 방법으로 풀어야 할 독립적인 문제들임을 나타낸다[1]. 즉, P_0 는 실계수 상에서의 계수를 구하는 최초의 LP 문제를 나타내고 있고, 나머지가 이로부터 파생되는 SP(sub-problem) 임을 나타내고 있다. 더 자세히 설명하면, SP_n^m 는 n 번째 레벨에서의 SP로서 이 문제가 파생한 SP의 실계수 가운데 하나를 양자화할 때 이를 i 번째 가까운 값으로 양자화 하여 이 상황에서 나머지 계수들의 최적 실수값을 구하는 것이다.

따라서, 레벨 $N-k$ 에서는 k 개의 계수가 이미 양자화되어 있고, 나머지 양자화 되지 않은 $N-k$ 개의 계수들 중 하나를 i 번째 가까운 값으로 고정하는 문제들을 만들고, 이 과정이 계속 반복되는 것이다. 가지치기 알고리즘은 가장 가까운 값으로부터 양자화를 하는 것이고, 이 상황에서 얻은 최적 해가 현재 가장 작은 minimax ripple 보다 더 작아질 가능성이 없으면 가지치기를 멈춘다.

이상과 같이 [1]에서의 알고리즘을 살펴본 결과, 설계할 필터의 차수가 커짐에 따라 풀어야할 SP의 갯수가 지수 함수적으로 증가함을 알 수 있다. 따라서 기존의 방법을 적용할 수 있는 필터의 차수는 매우 작은 값으로 제한되어 수십차의 필터만이 슈퍼 컴퓨터로 설계가 가능하다[5]. 이와같이 많은 계산량을 조금이라도 줄여 더 높은 차수의 최적 유한 정밀도 계수 필터를 설계하기 위하여 양자화할 계수의 순서를 최적으로 배열하는 방법 또는 중간과정에서 주파수 영역에서의 해상도를 낮추는 방법 등이 많이 연구된 바 있었다[1], [6]. 그러나 [1]의 경우 계산량이 거의 줄어들지 않으며 [6]의 경우 최적의 해를 보장할 수 없다는 단점이 있다. 최근에는 각 SP들의 상관관계를 이용하여 SP들의 계산량 자체를 줄일 수 있는 방법이 제안되었다[9]. 이 방법은 BaB 수형도에서 인접한 SP들끼리 서로 비슷한 결과를 낳는다는 사실에 착안을 하여 어느 SP에서의 결과를 이로부터 파생하는 SP들에 이용함으로써 각 SP들의 계산량을 크게 줄이는 것이다.

본 논문의 목적은 [9]에서 제안된 알고리즘을 개선하여 유한 정밀도 계수 필터 설계의 계산량을 더욱 줄이는 것이다. 제안된 알고리즘도 BaB 알고리즘에 기초를 하고 있으며, [9]에서는 위 레벨의 SP만이 이용되는데 반하여 제안된 알고리즘은 위의 SP 뿐만 아

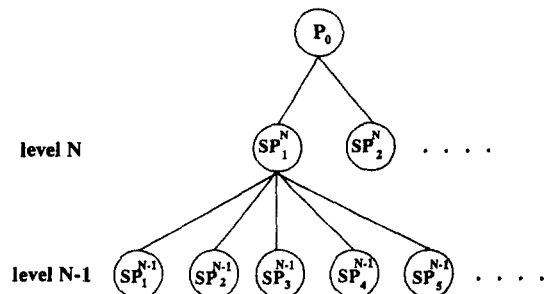


그림 1. BaB(branch and bound) 알고리즘(The branch and bound algorithm).

나라 같은 레벨에 있는 SP에서도 현재 풀고자 하는 SP에 대한 정보를 얻는 것이다.

일반적인 정수로 표현되는 유한 정밀도 계수 필터의 경우와는 달리 계수가 2의 멱승의 합으로 표시되는 경우에는 계수의 양자화 간격이 일정하지 않다는 특징이 있다. 이와 같은 성질을 이용하여 본 논문에서는 BaB 알고리즘에서 전체적으로 가지치기의 회수를 줄일 수 있는 새로운 가지치기 방법도 제시한다.

본 논문의 구성은 다음과 같다. 2절에서는 BaB 알고리즘에서 한 SP를 풀고자 할 때 주위의 SP로부터 많은 정보를 얻어 계산량을 줄일 수 있는 방법을 제시한다. 3절에서는 2의 멱승의 합으로 표시되는 값들의 양자화 간격이 일정하지 않다는 사실을 이용하여 가지치기 회수를 줄여 전체적인 계산량을 더욱 줄일 수 있는 방법을 제안한다. 4절에서는 제안된 알고리즘과 기존의 알고리즘을 몇가지 1차원 및 2차원 필터 설계의 경우에 적용하여 계산시간을 비교하고, 5장에서는 결론을 제시한다.

II. 유한 정밀도 계수 FIR 필터 설계 알고리즘

본 절에서는 최적 유한 정밀도 계수 필터 설계시 각 SP를 해결하기 위한 LP의 제한조건의 수를 줄여 계산량을 줄이는 방법을 제시하고, 2의 멱승의 합으로 계수가 주어지는 필터의 계산량을 더욱 줄일 수 있는 효율적인 가지치기 알고리즘을 제시한다.

A. 제한조건이 줄어든 알고리즘

본 논문에서 제시되는 알고리즘은 기존의 BaB 알고리즘과 마찬가지로 각 SP를 풀 때 LP를 이용한다. 기존의 알고리즘을 살펴보면 각 SP에서는 몇가지 계수가 이미 양자화된 상태에서 주어진 조건의 최적해를 얻을 수 있도록 나머지 계수를 조절하는 것이다. 이 때 각 SP는 위 레벨의 한 계수를 가까운 값으로 양자화 한 것이므로 그 해가 위 레벨이나 주위의 SP에 비하여 크게 변하지 않는다는 것을 관찰할 수 있다. 즉, 필터 응답으로 보면, minimax ripple을 갖는 주파수가 크게 다르지 않다는 것을 알 수 있다. 바꾸어 말하면, LP를 이용한 필터 설계시 주파수 영역이 적절한 값으로 샘플링 되고 각 샘플에서 제한조건을 구하여 이를 LP의 입력으로 사용하는데 [10], 각 SP의 최적해가 얻어지는 샘플의 집합이 주위 SP의 집합과 구성이 거의 같다는 것을 관찰할 수 있다. 기존의 알고리즘에서는 이러한 사실이 이용되지 않고 각

SP를 풀 때 하나의 계수가 양자화된데 따른 조건식의 변화만을 고려하여 모든 조건들로 새로 LP를 수행하였다. 따라서 본 논문에서는 한 SP를 풀 때 그 해의 최적값을 제공하는 제한조건의 집합이 주위 SP의 최적해를 제공하는 제한조건의 구성과 거의 같다는 사실을 이용하여 LP에 들어갈 제한조건의 수를 가능하면 작게 함으로써 전체적인 수행 속도를 높이는 방법을 제시한다.

더 구체적인 설명을 위하여 11차의 1차원 선형 위상, SOPOT FIR 필터 설계의 예를 살펴보자. 원하는 1차원 필터의 응답은 그림 2와 같고, 여기서 통과대역 주파수 $f_p = 0.2$, 차단대역 주파수 $f_s = 0.25$ 로 각각 주어졌다. 이 경우 [10]에서 제시된 바람직한 주파수 샘플 간격은 0.01이다. 즉, 그림 2에서 f 축이 0.01 간격으로 샘플링 되어 각 샘플에서의 주파수 응답에 대한 제한 조건을 LP의 제한 조건으로 사용하는 것이다. 주파수 응답에 대한 제한 조건은 절대값으로 주어지므로 각 샘플당 두개의 제한 조건이 필요하고, 따라서 이 문제를 푸는데 필요한 제한 조건은 총 94개가 되며(주파수 0.0부터 0.19까지 40개, 0.26부터 0.50까지 50개, 통과 대역 주파수 0.2 및 차단대역 주파수 0.25에 각 2개씩 4개), 결정해야 할 미지수의 수는 계수 6개와 minimax ripple δ 의 7개가 된다[10]. 본 논문의 예에서는 각 계수가 2의 멱승의 합으로 양자화 되며, 그 길이는 12비트로 주어지고 2개의 0이 아닌 값을 갖는 것으로 하였다. 이와 같은 조건에서 각 레벨에서 다음으로 넘어갈 때 하나의 계수가 가장 가까운 값으로 양자화 되는 경우, 각 레벨에서 동일 리플 주파수를 제공하는 제한조건의 번호(최적해를 제공하는 제한조건의 번호)를 표 1에 나타내었다. 이 표로부터 레벨 n 의 SP의 최적해를 제공하는 제한조건이 레벨 $n+1$ 에서의 제한조건과 거의 같다는 것을 알 수 있다. 즉 각 SP의 최적해를 제공하는 제한조건은 위 레벨의 조건과 거의 같음을 관찰할 수 있다.

LP의 특징은 위 예의 경우 P_0 의 최적해가 94개의 제한조건 중에 결국 표 1에서 보이는 7개 제한조건의 1차 연립방정식으로 얻어진다는 것이다. 따라서 LP의 해가 어떤 제한조건 주위에서 얻어질 것이라는 것을 미리 알고있으면 나머지 제한 조건은 사용하지 않아도 최적의 해를 얻을 수 있다. 마찬가지로 SP_1^6 의 해는 표에서 보인 6개만의 연립방정식으로 얻어진다. 그런데 이 예의 경우 SP_1^6 의 최적해를 제공하는 제한조건의 집합은 P_0 의 최적해를 공하는 제한조건의 집합의 부분집합임을 알 수 있다. 따라서 P_0 의 최적해

를 얻으면서 알게된 제한조건을 다음 레벨의 SP⁶를 풀기위한 제한 조건으로 사용하면 단 7개만의 제한조건으로 이의 최적해를 구할 수 있다. 반면에 기존의 알고리즘은 이와 같은 사실을 이용하지 않고, SP⁶을 풀 때 다시 처음과 같은 94개의 제한조건으로 문제를 푸는 것이다. 이와 같은 사실을 이용하여 [9]에서는, 레벨 n+1에서 얻어진 제한조건을 레벨 n에서의 SP를 풀기위한 처음 제한조건으로 사용하여 각 SP를 푸는데 필요한 제한조건의 수를 줄임으로써 LP의 계산시간을 줄이고 결국 전체적인 필터 설계 시간을 감소시켰다. 이 때 주의해야 할것은 SP², SP³, SP⁴, SP⁶의 경우에는 표 1에서 보는 바와 같이 제한조건의 집합이 바로 위 레벨의 제한조건의 부분 집합이므로 이러한 방법으로 최적해가 얻어지나, SP⁵의 경우에는 위 레벨의 SP⁶의 제한조건에 51번의 제한조건이 없으므로 이 방법으로 최적의 해를 얻을 수 없다는 것이다. 따라서 위의 제한조건을 처음의 제한조건으로 사용하고나서 항상 최적의 해인가를 확인하고 최적이지 아니하면 다시 다른 제한조건을 가지고 LP를 수행하는 과정이 필요하다. 따라서 위 레벨의 제한조건이 최적의 해를 제공하지 않는 경우가 많이 발생할 수록 풀어야할 LP의 회수가 증가하므로 계산량은 당연히 증가한다. 이러한 이유로 [9]에서의 알고리즘은 기존의 알고리즘에 비하여 약 5배 이내로 계산 속도가 증가하는데 그쳤다.

표 1. 최적의 해가 얻어진 제한조건의 번호(Indexes of the constraints with which the optimal minimax solutions are obtained).

level	notation of SP	indexes of constraints
	P ₀	1 28 49 68 89 91 92
6	SP ⁶ ₁	1 49 68 89 91 92
5	SP ⁵ ₁	1 89 91 92 51
4	SP ⁴ ₁	1 89 92 51
3	SP ³ ₁	1 92 51
2	SP ² ₁	1 51

따라서 본 논문에서는 초기 제한조건에 대한 정보를 위 레벨의 SP에서 뿐만 아니라 같은 레벨에서 이미 푼 SP로부터도 얻어서 처음 제한조건 추정이 더 정확하도록 함으로써 [9]에서보다 LP를 반복하는

회수를 줄이고 결국 전체 계산량이 줄어들도록 하였다. 우선, 같은 레벨의 SP들도 밀접한 관계가 있다는 것을 보이기 위하여 위와 같은 예에서 SP⁵부터 SP²⁵까지의 양자화 오차의 크기, 부호와 각 해의 최적값을 제공하는 제한조건의 번호를 표 2에 나타내었다. 그 결과를 보면 인접한 SP들끼리 밀접한 관계가 있으며, 특히 같은 양자화 오차 부호를 갖는 경우의 SP들이 그렇지 않은 경우에 비하여 제한조건의 유사성이 높다는 것을 알 수 있다. 즉, SP²부터 SP¹⁶까지는 제한조건이 완전히 같고 SP¹⁷은 이들과 비교하여 단 하나만의 원소가 다르다는 것을 알 수 있다. 이와 같이 SP²가 절대적으로 거리가 가까운 SP⁵에 비하여 거리가 먼 SP¹⁶과 더 상관 관계가 높은 이유는 이들이 같은 양자화 오차 부호를 갖고있기 때문이다.

표 2. 같은 레벨에 있는 SP들의 제한조건과 양자화 오차 부호의 예(Example for the sign of quantization error and the set of constraints for sub-problems at the same level).

	quantized coefficient	real coefficient	sign of quantization error	constraints
SP ¹	0.312500	0.30196	+	1 89 91 92 51
SP ²	0.281250	0.30196	-	1 49 68 91 92
SP ³	0.265625	0.30196	-	1 49 68 91 92
SP ⁴	0.257812	0.30196	-	1 49 68 91 92
SP ⁵	0.253906	0.30196	-	1 49 68 91 92
SP ⁶	0.251953	0.30196	-	1 49 68 91 92
SP ⁷	0.250977	0.30196	--	1 49 68 91 92
SP ⁸	0.250488	0.30196	--	1 49 68 91 92
SP ⁹	0.250244	0.30196	-	1 49 68 91 92
SP ¹⁰	0.250000	0.30196	-	1 49 68 91 92
SP ¹¹	0.249756	0.30196	--	1 49 68 91 92
SP ¹²	0.249512	0.30196	-	1 49 68 91 92
SP ¹³	0.249023	0.30196	-	1 49 68 91 92
SP ¹⁴	0.248047	0.30196	-	1 49 68 91 92
SP ¹⁵	0.246094	0.30196	-	1 49 68 91 92
SP ¹⁶	0.242188	0.30196	-	1 49 68 91 92
SP ¹⁷	0.234375	0.30196	--	3 49 68 91 92
SP ¹⁸	0.375000	0.30196	+	1 89 91 92 51
SP ¹⁹	0.218750	0.30196	-	3 49 68 91 92
SP ²⁰	0.187500	0.30196	--	3 49 68 91 92
SP ²¹	0.437500	0.30196	+	1 89 91 92 51
SP ²²	0.156250	0.30196	-	3 49 68 91 92
SP ²³	0.140625	0.30196	-	3 49 68 91 92
SP ²⁴	0.468750	0.30196	+	1 89 91 92 51
SP ²⁵	0.132812	0.30196	-	3 49 68 91 92

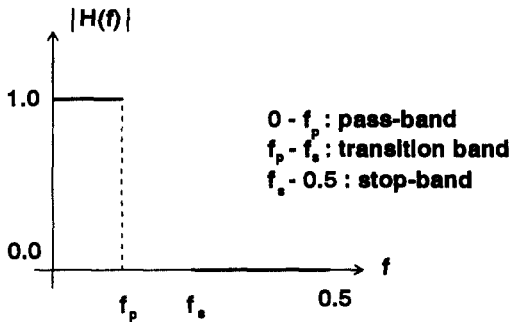


그림 2. 1차원 필터의 제한조건(Specification of 1-D FIR filter).

마찬가지로, SP_{18}^5 의 절대적인 거리는 앞의 것들보다 SP_1^5 에서 멀지만, 이들의 제한조건은 완전히 같다. 그 이유는 이들이 같은 유의 양자화 오차를 갖고 있기 때문이다. 이에 대한 수학적인 설명은 [8]의 오차 평면 또는 [11]의 ascent 알고리즘의 표현 등으로 할 수 있으나 지면상 생략한다.

또 다른 예로서 그림 1에서의 SP_1^{N-1} 가 표 3에서와 같은 부호를 갖는다고 가정하면 SP_1^{N-2} 의 제한조건은 SP_1^{N-1} 의 제한조건과 비슷하게 된다. 반면에, SP_4^{N-1} 의 조건은 SP_2^{N-1} 의 제한조건과 더 연관이 많다. 마찬가지로 SP_5^{N-1} 의 제한조건은 SP_4^{N-1} 보다는 SP_3^{N-1} 의 제한조건과 더 비슷하게 된다.

이와 같은 성질을 이용하여 그림 1의 SP_1^{N-1} 를 LP로 푸는 경우 처음의 제한조건 집합은 이 문제가 파생된 SP_N^N 의 최적해를 제공한 제한조건 집합과 SP_m^{N-1} 들 중에(m 은 i 보다 작은 정수) SP_1^{N-1} 와 같은 양자화 오차 부호를 가지면서 거리도 이에 가장 가까운 것의 최적해를 제공하는 제한조건 집합의 합집합으로 주어진다. 따라서, SP_1^{N-1} 를 풀 때 최대 $(2N-1)$ 개만의 제한조건이 사용된다. 예를 들어 표 1에서 레벨 3에 있는 SP를 풀 때 이 문제가 파생된 레벨 4의 제한조건 4개, 그리고 같은 레벨에 있는 SP들중 이미 푼 것이 있을 경우 가장 가까운 거리를 가지면서 같은 양자화 부호를 갖는 SP의 제한조건 3개, 총 7개중에서 중복되지 않는 제한조건들로 처음의 제한조건을 구성하여 푼다. 따라서 레벨 3의 SP를 풀기위한 최초의 제한조건은 최대 7개로서 기존의 알고리즘의 94개와 비교하면 매우 작은 값이다. 더 구체적으로 표 2에

서, SP_1^6 로부터 파생된 문제 중의 하나인 SP_{21}^5 를 푸는 경우를 살펴보자. 표 2에 따르면 이와 같은 양자화 오차를 가지면서 가장 가까운 SP는 SP_{18}^5 임을 알 수 있다. 따라서 이 문제를 풀기위한 초기 제한조건은 표 1에서 보는 바와 같이 SP_1^6 의 최적해를 제공하는 조건의 집합 {1, 49, 68, 89, 91, 92}와, SP_{18}^5 의 최적해 집합인 {1, 89, 91, 92, 51}의 합집합으로 한다. 따라서 제안된 알고리즘에서는 SP_{21}^5 를 풀기위한 최초의 제한조건 집합이 {1, 49, 51, 68, 89, 91, 92}로 주어진다.

그러나 앞서도 언급한 바와 같은 이와 같은 줄어든 제한조건 집합을 사용하는 경우 항상 최적 값이 얻어진다는 보장은 없다. 예를 들어 표 1의 레벨 5의 경우, 줄어든 제한조건으로 푼 해는 최적일 수가 없고, 경우에 따라 infeasible 또는 unbounded 해를 제공할 수 있다. 마찬가지로 표 2에서의 경우, 이의 최초 제한조건으로 사용되는 SP_1^6 와 SP_{18}^5 의 최적해 집합에 3번 제한조건이 없으므로 이와 같은 방법이 최적의 해를 제공할 수는 없다. 따라서 [9]에서와 마찬가지로 줄어든 제한조건으로 문제를 푼 경우 해가 최적인가를 항상 검사해야 한다.

본 논문에서 최적해의 검사는 [9]에서와 같은 과정으로 수행하였다. 즉, LP 자체에서 우선 해의 feasibility와 boundedness에 대한 검사는 이루어지는데 만약 처음의 제한조건이 해를 얻을 수 없는 조건이라고 판별이 되면, 별 수 없이 처음의 제한조건으로 문제를 해결하는 것으로 끝낸다. 그러나 이러한 경우는 많은 시뮬레이션을 수행한 지금까지 아직 발견된 바가 없다. 다음으로, 해의 최적 여부를 판별하는 방법은 [11]의 ascent 알고리즘에서와 같은 개념을 사용하였다. 즉, 얻어진 해를 각 제한조건에 대입하여 현재 얻어진 최소의 δ 보다 큰 오차가 발생하는 제한 조건을 현재의 제한조건에 포함시키는 것이다. Ascent 알고리즘의 경우 현재의 제한조건과 밖의 제한조건을 하나 또는 여러개 교환하는 방법을 사용하였는데, 실험적으로 LP의 경우에는 기존의 제한조건을 유지하는 것이 더 빠른 것으로 나타났다. 이러한 작업이 모든 제한조건이 현재의 δ 보다 작거나 같아질 때까지 반복되어 결국 최적의 해를 얻을 수 있다. 이 작업이 반복되면 간혹 제한조건이 수가 처음 문제의 제한조건 unbounded 해를 다룰 때와 마찬가지로 처음의 모든 제한조건으로 문제를 푸는 것으로 끝낸다. 그러나,

¹이론적으로는 새로 구성되는 제한조건이 수가 원래 문제의 제한조건 수보다 많아질 수는 없지만 계산 오차 전파가 간혹 이런 문제를 발생시킨다.

대부분의 경우 처음의 제한조건으로 최적의 해가 얻어지거나 한두번만의 반복으로 최적해가 얻이므로 모든 문제를 원래 문제의 제한조건으로 푸는 기존의 방법에 비하여 계산량이 크게 줄어든다.

제안된 알고리즘을 그림 1의 SP_5^{N-1} 를 푸는 예에 대하여 그림 3에 순서도로 나타내었다². 표 3에 따르면, 이 문제는 SP_3^{N-1} 과 같은 양자화 오차 부호를 가지므로 처음 제한조건 구성이 그림에서와 같이 SP_3^{N-1} 의 최적해 집합과 SP_1^N 의 최적해 집합의 합집합이 되는 것이다. 즉, 앞에서 설명된 바와 같이 지금 풀어야 할 SP_5^{N-1} 는 거리가 더 가까운 SP_4^{N-1} 보다 부호가 같은 SP_3^{N-1} 와 더 상관 관계가 높으므로 이의 조건을 사용하는 것이다. 마찬가지로, SP_4^{N-1} 를 푸는 경우에는 처음 제한조건을 SP_1^N 의 최적해 집합과 SP_2^{N-1} 의 최적해 집합의 합집합으로 한다. 그 이유는 표 3에서 보는 바와같이 SP_4^{N-1} 의 양자화 오차 부호는 SP_3^{N-1} 과는 다르고 SP_2^{N-1} 와 같기 때문이다. 그리고 SP_3^{N-1} 을 푸는 경우에는 이것이 SP_1^N 로부터 파생된 문제들 중 처음으로 양자화 오차를 갖는 SP이므로 이의 최초의 제한조건은 SP_1^N 의 최적해만으로 구성한다. 마찬가지로, SP_i^{N-1} 은 첫 SP이므로 SP_1^N 의 최적해만으로 시작한다.

B. SOPOT 계수 필터 설계를 위한 새로운 가지 치기 방법.

일반적인 정수 계수 필터를 설계하는 경우 각 계수의 양자화 간격은 일정하므로 $SP_1^N, SP_2^N, SP_3^N, \dots$ 의 양자화 오차 부호는 일정하게 반복된다. 그러나 2의 멱승의 합으로 계수를 표현하는 경우 양자화 간격이

C_0 : Original set of constraints

Input from SP_1^N : Set of constraints, defined as C_1 , with which the optimal solution for SP_1^N is obtained.
 Input from SP_3^N : Set of constraints, defined as C_3 , with which the optimal solution for SP_3^N is obtained.
 $C = C_1, UC_1$

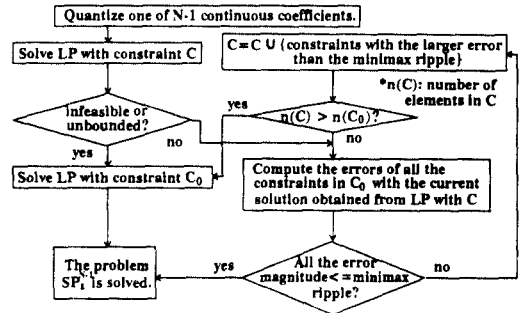


그림 3. 하나의 SP를 풀기위한 제안된 알고리즘의 순서도 (The signal flow graph of the proposed algorithm for solving an SP).

일정하지 않으므로 표 2 또는 표 3의 예에서 본 바와 같이 부호가 많이 반복되는 경우가 많다. 이러한 현상은 특히 계수를 표현하는 전체 비트 길이에 비하여 0이 아닌 값을 갖는 비트의 수가 작을수록 현저하게 나타난다.

그런데 여러 종류의 유한 정밀도 계수 필터 설계에 있어서 BaB 수형도에서 최적의 해가 얻어지는 경로를 따라가 보면 각 SP의 양자화 오차의 부호가 + 또는 -일 확률이 거의 0.5로 같다는 사실을 알 수 있다. 즉, 그림 4의 점선이 최적 해의 경로라고 할 때 $SP_{opt}^N, \dots, SP_{opt}^1$ 의 양자화 오차의 부호는 평균적으로 반이 +이고 반은 -임을 실험을 통하여 확인할 수 있었

표 3. SP의 양자화 오차부호의 예 (An example for the sign of quantization error of the sub-problems).

	SP_1^{N-1}	SP_2^{N-1}	SP_3^{N-1}	SP_4^{N-1}	SP_5^{N-1}	SP_6^{N-1}	SP_7^{N-1}	SP_8^{N-1}	SP_9^{N-1}
sign of the quantization error	+	+	-	+	-	-	-	+	+

²순서도를 보면 최적해가 얻어지지 않은 경우 현재의 δ 에 비하여 큰 오차를 갖는 제한조건을 현재의 제한조건에 포함시키게 되어있다. 그러나, 실험적으로 $\delta + \epsilon$ (ϵ 은 매우 작은 수)보다 큰 오차를 갖는 제한조건을 포함시키면 전체적인 제한조건의 수가 크게 늘어나지 않으면서도 해의 최적 여부에는 영향을 미치지 않는 것을 알 수 있었다. 3절의 실험에서는 ϵ 을 10^{-6} 으로 하였다.

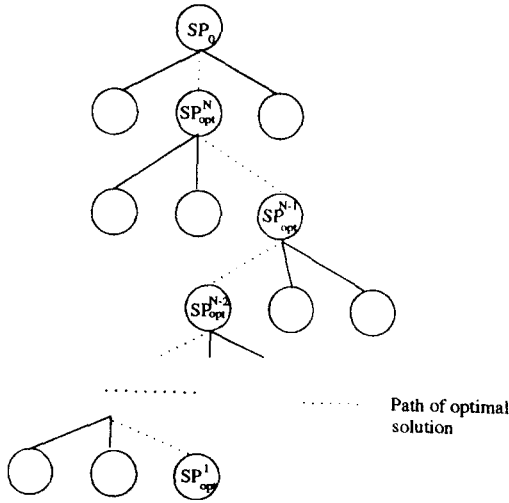


그림 4. BaB 수행도에서 최적의 해가 얻어진 경로(Path of optimal solution in the BaB tree).

다. 그 이유는 [8]에서 보인 바와 같이 SP의 오차 폭면이 오목하고 제한 조건을 만드는 주파수 샘플링 간격이 좁아서 그 수가 많을 때 최적해를 중심으로 그 모양이 거의 대칭이기 때문이다. 구체적으로 11차 1차원 필터의 예를 보면, 우선 그 전달함수 식은 다음과 같이 쓸 수 있고,

$$H(z) = h_0 + h_1 z^{-1} + h_2 z^{-2} + \dots + h_5 z^{-5} + h_4 z^{-6} + \dots + h_0 z^{-10} \quad (1)$$

이 때 앞에서와 같이 $f_p = 0.2$, $f_s = 0.25$ 이면 LP를 위한 제한조건은 다음과 같이 된다[10].

$$\begin{aligned} & |h_5 + 2h_4 \cos 2\pi f_i + 2h_3 \cos 4\pi f_i + \dots + 2h_0 \cos 10\pi f_p - 1.0| < \delta \\ & \text{for } 0.0 \leq f_i \leq 0.2 \\ & \dots \dots \\ & |h_5 + 2h_4 \cos 2\pi f_i + 2h_3 \cos 4\pi f_i + \dots + 2h_0 \cos 10\pi f_i - 0.01| < \delta \\ & \text{for } 0.25 f_i \leq 0.5 \end{aligned} \quad (2)$$

여기서 f_i 는 주파수 축의 0부터 0.5를 일정한 간격으

로 샘플링 한 값을 나타낸다. 가장 간단한 경우로서 BaB 알고리즘에서 h_5 부터 양자화를 시작하여 마지막 레벨에 이르렀을 때 위의 식은 다음과 같이 된다.

$$\begin{aligned} & |2h_0 \cos 10\pi f_i + a_i - 1.0| < \delta \\ & \text{for } 0.0 \leq f_i \leq 0.2 \\ & \dots \dots \\ & |2h_0 \cos 10\pi f_i + a_i| < \delta \\ & \text{for } 0.25 \leq f_i \leq 0.5 \end{aligned} \quad (3)$$

여기서 a_i 는 $h_5 + \dots + 2h_1 \cos 8\pi f_i$ 의 각 h_i 에 이들의 양자화된 값을 대입한 것이다. 따라서 f_i 의 간격을 0.01이라고 한 때 앞에서 본 바와 같이 제한조건은 총 94개이고($i = 1, \dots, 47$, 각 i 당 제한조건 2개) 위와 같은 제한조건을 갖는 LP의 해는 다음과 같은 최소최대(minimax)문제의 해와 같다[11].

Find the minimum of

$$\begin{aligned} & \text{Max } \{|2 \cos 10\pi \times 0.0\}h_0 + a_1 - 1|, \\ & \quad \{|2 \cos 10\pi \times 0.01\}h_0 + a_2 - 1|, \\ & \quad \dots \dots \\ & \quad \{|2 \cos 10\pi \times 0.49\}h_0 + a_{46}|, \\ & \quad \{|2 \cos 10\pi \times 0.5\}h_0 + a_{47}|, \end{aligned} \quad (4)$$

결국 위의 문제는 ascent 알고리즘으로 풀 수도 있는 문제이며 [11], i 번째와 j 번째 식을 동시에 만족하는 해가 최적의 해라고 하면 그림 5에서 보는 바와 같이 이는 i 번째 식이 나타내는 직선과 j 번째 식이 나타내

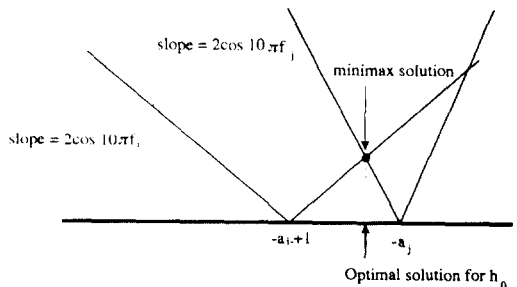


그림 5. 최소최대 문제의 해(Solution of minimax problem).

는 직선의 교점이라고 할 수 있다. 또한 최적해는 위의 식으로 나타나는 여러 직선의 교점들 중에서 그 위에는 다른 직선이 존재하지 않으면서 그 높이가 가장 작은 점이라 할 수 있다[11]. 따라서 제한 조건을 만들 때 f_i 의 간격이 매우 작으면 최적해 주변의 직선들의 기울기 $2 \cos 10\pi f_n$ 의 변화가 매우 작고 최적해를 중심으로 오차면이 거의 대칭이 된다. 제한 조건의 간격이 작지 않은 경우에도 식 (4)에서 보는 바와 같이 제한 조건들이 절대값으로 주어지므로 대부분 그림 6에서 보는 바와 같은 오차면을 갖게 된다. 따라서 h_0 를 근처의 이산값 h_0^- 또는 h_0^+ 로 양자화 할 때 최적의 해가 오른쪽에서 얻어질 확률과 왼쪽에서 얻어진 확률이 거의 같게 된다. 결국 어떤 SP에서 하위 레벨의 SP를 파생시킬 때 기존의 방법과 같이 양자화 오차의 크기가 커지는 순서대로 파생시키는 것보다는 부호가 엇갈리는 순서대로 파생시키는 것이 더 작은 δ 를 갖는 해를 일찍 찾을 수 있고, 따라서 풀어야할 SP의 수가 줄어들 가능성이 있다.

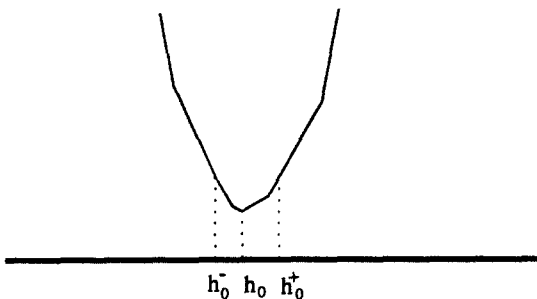


그림 6. 식 (4)의 최대최소 문제의 오차면(Error surface of the minimax problem)

구체적인 예로 표 2를 보면 SP_{17}^5 와 SP_{18}^5 로부터 최적의 해를 얻을 가능성이 거의 같은데도 불구하고 양자화 오차의 크기가 증가하는 순서로 문제를 풀면 SP_2^5, \dots, SP_{17}^5 의 모든 해 및 이들로부터 파생될 수 있는 문제들의 해를 모두 풀고 후에 SP_{18}^5 를 풀어야 한다. 제시된 방법에서는 SP_3^5 를 풀기에 앞서 SP_{18}^5 를 풀기 때문에 SP_3^5, \dots, SP_{17}^5 과는 다른 특성의 해를 구하는 것이며 이로부터 파생되는 문제들중 최적의 해로서 가능성이 있는 것을 일찍 발견할 수 있고, 그 결과로 SP_3^5, \dots, SP_{17}^5 를 풀때 하위 레벨로 내려가는 깊이를 작게할 가능성이 있다.

제안된 가지치기 방법의 다른 예로 표 3과 같은 순서로 양자화 오차 부호를 갖는 경우 제시된 방법은 이를 $SP_2^{N-1}, SP_3^{N-1}, \dots$ 와 같이 오차의 크기가 증가하는 순서대로 풀지 않고 그림 7과 같이 부호가 엇갈리는 순서대로 푸는 것이다. 또 한가지 예로, 제시된 방법으로 표 2와 같은 성질의 SP를 풀 때 그 순서는 $SP_1^5, SP_2^5, SP_{18}^5, SP_3^5, SP_{21}^5, SP_4^5, SP_{24}^5, SP_5^5, \dots$ 와 같이 된다.

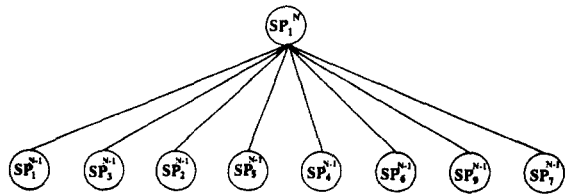


그림 7. 표 3에서와 같이 양자화 오차를 갖는 경우, SOPOT 계수 필터를 설계할 때의 SP의 순서(The order of the sub-problems for the design of power-of-two coefficient filters, when the signs of the SPs appear as shown in Table 3).

III. 알고리즘의 비교

제시된 알고리즘과 기존의 알고리즘을 비교하기 위하여 몇개의 1차원 및 2차원 FIR 필터를 각 알고리즘으로 설계하였다. 계산 시간 비교는 [1] 등에서와 같이 다른 제한조건을 갖는 몇개의 필터 설계에 대한 각각의 계산시간을 조사하고 이의 평균을 비교하는 방법을 사용했다. 공정한 비교를 위하여 기존의 알고리즘과 제안된 알고리즘의 전체적인 구조는 같도록 하였고, 사용된 LP subroutine도 같은 것으로 하였다. 즉, 같은 LP를 사용하면서 제안된 알고리즘에서는 LP의 수행 횟수가 더 많지만, 이의 입력의 갯수가 줄어들어 계산상의 잇점이 있다는 것을 보이는 것이다.

첫째로 표 4에서와 같은 조건을 갖는 11, 21, 31차의 필터를 각 알고리즘으로 설계하여 그 설계시간을 표 5에 나타내었다. 여기서 앞의 예와 같이 각 계수는 12 비트의 길이를 갖고 이중 2개만이 0이 아닌 값을 갖는 것으로 하였다. 그리고 사용된 컴퓨터는 Mips RS3000/33 워크스테이션으로서 40 MIPS 정도의 성능을 가지며, 프로그램의 변수는 모두 2배정도를 사용하였다. 전체적인 BaB의 구성에서는 기존의 알고

리들이나 제안된 알고리즘의 경우 모두, [8]의 방법에 따른 가지 증가 억제 방법을 사용하였다. 그러나 실험 결과 이 방법으로 위에서 언급한 일반적인 워크스테이션을 사용하였을 때 시간이 매우 오래 걸리며, 특히 기존의 방법으로는 설계가 거의 불가능할 정도로 계산량이 많다는 것을 발견하였다. 또한 비교적 차수가 작고 0이 아닌 비트 수가 두개인 경우에 많은 실험을 해본 결과 지금까지 설계해 본 모든 필터는 가지 수 10개 이내에서 최적해를 가지며 그 이상에서 발견된 해는 매우 멀리 떨어져 있고 10개 이내에서 발견된 해와 같은 최대 리플값을 갖는다는 것을 발견하였다. 예를 들어 첫번째 가지와 125번째 가지에서 최대 리플의 크기가 똑같은 해가 발견되는 경우도 있으므로 매우 먼 가지까지 계수를 찾는 것이 의미가 없을 수도 있다. 따라서 본 논문에서는 지나치게 많은 가지의 증가로 인한 계산량의 증가를 막기 위하여 11차 21차 필터의 경우 20개, 31차 필터의 경우 10개로 가지의 수를 제한하였다. 그러나 어느 범위에서 최적의 해가 얻어진다는 이론적인 근거는 없으므로 본 논문에서의 실험 결과는 엄밀하게 말해 각 방법의 최적해를 얻는데 걸린 시간의 비교라기 보다는 주어진 환경 내에서 각 sub-problem을 푸는데 걸리는 시간의 비교라고 해야 한다. 즉, 이와 같은 인위적으로 가지 수를 제한하였을 때에는 최적해를 보장할 수는 없다. 하지만 앞에서 언급한 바와 같이 본 논문에서 주어진 조건에서는 대부분 최적의 해가 얻어지므로 많은 경우에 최적해를 얻는데 걸린 시간의 비교라 할 수 있다. 또한 기존의 방법이나 제안된 방법에서 가지치기 방법은 모두 같으므로 같은 조건 내에서 BaB 알고리즘을 수행하는 시간의 비교로는 볼 수 있다.

결과를 보면 거의 모든 경우에 제시된 알고리즘이 기존의 방법에 비하여 수행 속도가 빠른 것으로 나타났으며 특히 그 차이는 차수가 증가할수록 두드러진다는 것을 알 수 있다. 또한 기존의 방법은 일반적인 워크스테이션으로 설계하기에는 매우 계산량이 많으며 제시된 알고리즘은 기존의 방법에 비하여 계산 시

표 4. 1차 FIR 필터의 제한조건 (Specifications of the 1-D FIR filters).

	1D-1	1D-2	1D-3	1D-4
f_p	0.12	0.16	0.20	0.24
f_s	0.38	0.34	0.30	0.26

간을 단축시키는 것으로 나타났다. 제시된 방법은 SP 자체를 푸는 효율적인 방법이므로, 이를 [6]의 grid-reducing 방법이나 다른 효율적인 가지 줄이기 방법, BaB에 기초를 둔 준 최적 알고리즘에 그대로 적용할 수 있으며, 이 경우 계산 시간은 더욱 줄어들 것으로 기대된다.

같은 조건에서 영상의 샘플링 구조 변환에 많이 사용되는 2차원 대각형 필터 설계에 대하여 각 알고리즘의 비교도 수행하였다[12]. 설계된 필터의 조건을 표 6에 나타내었는데, 여기서 f_p 와 f_s 는 그림 8에서와 같은 2차원 필터의 차단, 통과 대역 주파수이다. 이 경우에 대한 각 알고리즘의 계산시간 비교를 표 7에 나타내었다. 이 결과 역시 제시된 알고리즘이 기존의 방법에 비하여 효율적임을 보이고 있다.

표 5. 각 알고리즘의 1차 FIR 필터 설계 시간 비교. A: 기존의 알고리즘, B: 제안된 알고리즘, C: 새로운 가지치기 방법을 더한 알고리즘 (Comparison of the algorithms for the design of 1-D filters. A: conventional algorithm, B: proposed algorithm, C: proposed algorithm with a new branching scheme).

filter length : 11 (seconds)						
	1D-1	1D-2	1D-3	1D-4	최대시간	Average
A	4.6	3.2	29.3	6.6	29.3	10.9
B	3.2	2.5	15.8	2.5	15.8	6.0
C	4.2	3.2	25.1	3.2	25.1	8.9
filter length : 21 (hour: minute: seconds)						
	1D-1	1D-2	1D-3	1D-4	최대시간	Average
A	2:51:54	0:08:18	0:25:46	16:36:59	16:36:59	5:00:44
B	1:00:33	0:06:12	0:06:54	0:27:42	1:00:33	0:25:20
C	0:47:39	0:04:48	0:07:24	0:14:23	0:47:39	0:18:34
filter length : 31 (hour: minute: seconds)						
	1D-1	1D-2	1D-3	1D-4	최대시간	Average
A	79:26:19	16:51:22	0:25:34	370:09:46	370:09:46	116:43:15
B	22:08:09	6:18:31	0:12:59	53:48:41	53:48:41	20:37:05
C	3:30:03	1:47:09	0:10:45	50:26:21	50:26:21	13:58:35

표 6. 2차원 필터의 제한조건 (Specifications of the 2-D FIR filters).

	2D-1	2D-2	2D-3	2D-4
f_p	0.14	0.22	0.30	0.38
f_s	0.36	0.28	0.20	0.12

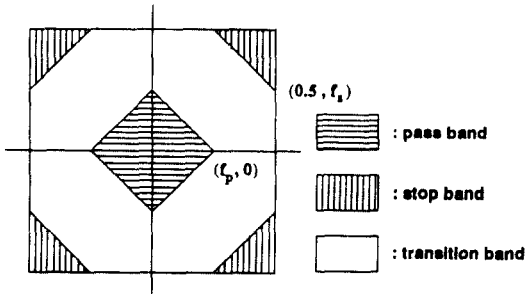


그림 8. 2차원 대각형 필터(Two-dimensional quincuncial filter)

표 7. 각 알고리즘의 2차원 필터 설계 시간 비교. A: 기존의 알고리즘, B: 제안된 알고리즘, C: 제안된 가지치기 방법을 더한 경우(Comparison of the algorithms for the design of 2-D filters. A: conventional algorithm, B: proposed algorithm, C: proposed algorithm with a new branching scheme).

filter length : 3 x 3 (seconds)						
	2D-1	2D-2	2D-3	2D-4	최대시간	Average
A	17.0	7.5	1.6	1.9	17.0	7.0
B	8.6	2.6	0.6	1.2	8.6	3.3
C	3.9	1.9	0.6	0.7	3.9	1.8

filter length : 5 x 5 (hour: minute: seconds)						
	2D-1	2D-2	2D-3	2D-4	최대시간	Average
A	100:44:54	347:15:50	16:01:04	45:04:09	347:15:50	127:16:29
B	21:46:07	47:07:27	1:39:15	1:00:14	47:07:27	17:53:16
C	2:09:34	19:26:25	0:22:25	0:12:53	19:26:25	5:57:49

IV. 결 론

본 논문에서는 유한 정밀도 계수 FIR 필터의 설계 알고리즘이 제안되었다. 제안된 알고리즘은 기존의 BaB(branch and bound) 알고리즘에서 고려되지 않은 각 SP(sub-problem)들간의 관계를 이용하여 각 SP의 수행시간을 단축시키는 것이다. 즉, 주위의 SP들로부터 현재 풀고자 하는 SP의 제한조건에 대한 정보를 미리 얻어서 LP의 제한조건 수를 줄임으로써 전체적인 계산량을 줄이는 것이다. 또한 필터 계수가 2의 멱승의 함(SOPOT)으로 주어지는 경우 BaB 알고리즘의 계산시간을 더욱 단축시킬 수 있는 가지치기 방법이 제시되었다. 이 방법은 SOPOT 표현시 양자화 간격이 일정하지 않다는 사실을 이용한 것으로

서 기존의 방법이 양자화 오차의 크기가 증가하는 순서대로 가지를 치는데 비하여 제시된 방법은 부호가 엇갈리는 순서대로 가지를 쳐서 최적의 해를 더 일찍 찾을 수 있는 확률을 높이는 것이다. 제시된 알고리즘을 기존의 방법과 비교해본 결과 계산시간이 많이 단축됨을 확인할 수 있었다. 특히, 제시된 알고리즘은 필터의 차수가 높은 경우에 더 효율적이었고, 새로운 가지치기 방법과 제한조건 줄이는 방법을 결합한 경우에는 SOPOT 계수 필터 설계에 더 많은 계산 감축의 효과가 있음을 알 수 있다. 또한 제안된 방법은 일반적인 정수 프로그래밍(integer programming)에도 적용될 수 있을 것으로 기대된다.

참 고 문 헌

1. Y. C. Lim and S. R. Parker, "FIR filter design over a discrete powers-of-two coefficient space," *IEEE Trans. Acoust., Speech, Signal Processing*, pp. 583-590, vol. 31, no. 3, June 1983.
2. Y. Chen, S. M. Kang, and T. G. Marshall, "The optimal design of CCD transversal filters using mixed-integer programming techniques," in *Proc. 1978 IEEE Int. Symp. Circuits Syst.*, pp. 948-751, New York, May 1978.
3. D. M. Kodek, "Design of optimal finite wordlength FIR digital filters using integer programming techniques," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 28, no. 3, June 1980.
4. D. M. Kodek and K. Steiglitz, "Comparison of optimal and local search methods for designing finite wordlength FIR digital filters," *IEEE Trans. Circuits and Systems*, vol. 28, no. 1, Jan. 1981.
5. Y. C. Lim and S. R. Parker, "Discrete coefficient FIR digital filter design based upon an LMS criteria," *IEEE Trans. Circuits and Systems*, vol. 30, pp. 723-739, Oct. 1983.
6. P. Siohan and A. Benslimance, "Finite precision design of optimal linear phase 2-D FIR digital filters," *IEEE Trans. Circuits and Systems*, vol. 36, no. 1, pp. 11-21, Jan. 1989.
7. H. Samuelli, "An improved search algorithm for the design of multiplierless FIR filters with

- powers-of-two coefficients," *IEEE Trans. Circuits and Systems*, vol. 36, no. 7, pp.1044-1047 1989.
8. B. Jaumard, M. Minoux, and P. Siohan, "Finite precision design of FIR filters using a convexity property," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 407-411, March 1988.
9. N. I. Cho, S. U. Lee, and K. H. Kim, "Design of FIR filter over a discrete coefficient space with applications to HDTV signal processing," *Proc. IEEE International Symposium Circuits and Systems*, pp. 76-79, Chicago, Illinois, May 1993.
10. L. R. Rabiner, "The design of finite impulse response digital filters using linear programming techniques," *The Bell System Technical Journal*, vol. 21, pp. 1177-1198, July August 1972.
11. E. W. Cheney, *Introduction to Approximation Theory*, McGraw-Hill, 1966.
12. P. Siohan, "2-D FIR filter design for sampling structure conversion," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 1, pp. 337-350, Dec. 1991.



趙南翊(Nam Ik Cho) 정회원
1986년: 서울대학교 공과대학 制御計測工學科, 학사
1988년: 서울대학교 대학원 制御計測工學科, 석사
1992년: 서울대학교 대학원 制御計測工學科, 박사
1994년~현재: 서울시立大學校 制御計測工學科 전임강사

1991년~1994년: 서울대학교 제어계측신기술연구센터 연구원

李商郁(Sang Uk Lee) 정회원
현재: 서울대학교 제어계측공학과 교수

정회원