

Short-length running convolution을 사용한 저전력 필터 구조

*오세만, 이원상, 장영범
상명대학교 정보통신공학과

e-mail : ziza@smu.ac.kr, windstorm5@smu.ac.kr, ybjang@smu.ac.kr

Low power filter structure using Short-length running convolution

*Se-Man Oh, Won-Sang Lee, Young-Beom Jang

Department of Information and Telecommunication Engineering,
Sangmyung University

Abstract

In this paper, an efficient and fast algorithm to reduce calculation amount of FIR(Finite Impulse Responses) filtering is proposed. Proposed algorithm enables arbitrary size of parallel processing, and their structures are also easily derived. Furthermore, it is shown that the number of multiplication/sample is reduced, and number of instructions using MAC(Multiplication and Accumulation) processor are also reduced. For theoretical improvement, numbers of sub filters are compared with those of conventional algorithm. In addition to the theoretical improvement, it is shown that number of element for hardwired implementation are reduced comparison to those of the conventional algorithm.

I. 서 론

이 논문에서는 FIR 필터의 연산의 양을 줄이는 효율적인 직접방식의 고속 알고리즘을 제안하였다. 제안된 알고리즘은 임의의 다운샘플링 크기로 병렬화가 가능하며, 다운샘플링의 크기가 결정되면 쉽게 구조를 유도할 수 있다. 특히 제안된 알고리즘은 이론적인 샘플당 곱셈연산의 수를 감소시킴과 동시에 실제 구현에 있어서도 효과가 있음을 실험을 통하여 입증하였다. 이론적으로 연산의 양이 감소함을 보이기 위하여 부필터의 수와 샘플당 곱셈연산의 수를 기존의 고속 알고리즘과 비교하였으며, 실제적으로 구현의 효과를 입증하기 위하여 하드웨어 구현소자의 수와 MAC 프로세서를 사용한 소프트웨어 구현으로 역시 기존의 방식들과 비교하여 제안된 구조가 효과적임을 보였다.

II. 본 론

일반적으로 필터는 주파수 영역에서 다음과 같이 표현된다. 즉,

$$Y(z) = H(z)X(z) \quad (1)$$

이 식에서 $Y(z)$, $H(z)$, $X(z)$ 는 각각 출력신호 $y[n]$, 임펄스 응답 $h[n]$, 입력신호 $x[n]$ 의 z 변환이다. 다운 샘플링 3을 사용하는 알고리즘을 만들기 위하여 먼저 위의 입력, 출력, 전달 함수 등의 3개의 다항식들을 다음과 같이 3의 polyphase로 분해한다.

$$\begin{aligned} X(z) &= X_0(z^3) + z^{-1}X_1(z^3) + z^{-2}X_2(z^3) \\ Y(z) &= Y_0(z^3) + z^{-1}Y_1(z^3) + z^{-2}Y_2(z^3) \\ H(z) &= H_0(z^3) + z^{-1}H_1(z^3) + z^{-2}H_2(z^3) \end{aligned} \quad (2)$$

이와 같은 3의 polyphase 분해식 (2)를 식 (1)에 대입하면 다음과 같은 z 영역에서의 입출력 관계식을 얻을 수 있다.

$$\begin{aligned} &Y_0 + z^{-1}Y_1 + z^{-2}Y_2 \\ &= [H_0X_0 + z^{-3}H_2X_1 + z^{-3}H_1X_2] \\ &+ z^{-1}[H_1X_0 + H_0X_1 + z^{-3}H_2X_2] \\ &+ z^{-2}[H_2X_0 + H_1X_1 + H_0X_2] \end{aligned} \quad (3)$$

이 식의 모든 다항식들은 (z^3) 을 생략하고 표현하였다. 즉 H_0 는 $H_0(z^3)$ 을 나타낸다. 이 식을 행렬식으로 표현하면 다음과 같이 나타낼 수 있다.

$$\begin{bmatrix} Y_0 \\ Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} H_0 & z^{-3}H_2 & z^{-3}H_1 \\ H_1 & H_0 & z^{-3}H_2 \\ H_2 & H_1 & H_0 \end{bmatrix} \begin{bmatrix} X_0 \\ X_1 \\ X_2 \end{bmatrix} \quad (4)$$

이 식의 우변을 pseudocirculant 행렬이라고 부르며 9개의 부필터를 사용하여 구현할 수 있음을 알 수 있다. 사용되는 부필터의 수를 줄이기 위하여 식(4)의 Y_2 , Y_1 , Y_0 를 각각 다음과 같이 가공한다.

$$Y_2 = H_2 X_0 + H_1 X_1 + H_0 X_2$$

$$= H_1 X_1 + (H_0 + H_2)(X_0 + X_2) - H_0 X_0 - H_2 X_2 \quad (5a)$$

$$Y_1 = H_1 X_0 + H_0 X_1 + z^{-3} H_2 X_2$$

$$= z^{-3} H_2 X_2 + (H_0 + H_1)(X_0 + X_1) - H_0 X_0 - H_1 X_1 \quad (5b)$$

$$Y_0 = H_0 X_0 + z^{-3} H_2 X_1 + z^{-3} H_1 X_2$$

$$= H_0 X_0 + z^{-3} (H_1 + H_2)(X_1 + X_2) - z^{-3} H_1 X_1 - z^{-3} H_2 X_2 \quad (5c)$$

이와 같이 식 (5)를 사용하여 가공하면 H_0 , H_1 , H_2 , $H_0 + H_1$, $H_1 + H_2$, $H_0 + H_2$ 등과 같이 6개의 필터를 사용하여 계산할 수 있음을 알 수 있다. (5)의 식을 Toom-Cook 알고리즘을 사용하면 다음과 같이 나타낼 수 있다.

$$\begin{bmatrix} Y_0 \\ Y_1 \\ Y_2 \end{bmatrix} = \mathbf{C}_3 \left\{ \mathbf{A}_3 \begin{bmatrix} H_0 \\ H_1 \\ H_2 \end{bmatrix} * \mathbf{A}_3 \begin{bmatrix} X_0 \\ X_1 \\ X_2 \end{bmatrix} \right\} \quad (6)$$

이 식에서 *는 내적을 의미하며 \mathbf{A}_3 와 \mathbf{C}_3 는 각각 다음과 같다.

$$\mathbf{A}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \quad (7)$$

$$\mathbf{C}_3 = \begin{bmatrix} 1 & -z^{-3} & -z^{-3} & 0 & 0 & z^{-3} \\ -1 & -1 & z^{-3} & 1 & 0 & 0 \\ -1 & 1 & -1 & 0 & 1 & 0 \end{bmatrix}$$

\mathbf{A}_3 의 행의 수는 사용되는 부필터의 수를 나타내며, \mathbf{C}_3 는 출력을 표시하기 위한 부필터의 조합을 나타내고 있다. 식 (6)과 (7)을 사용하여 필터 구조를 만들면 그림 1과 같다.

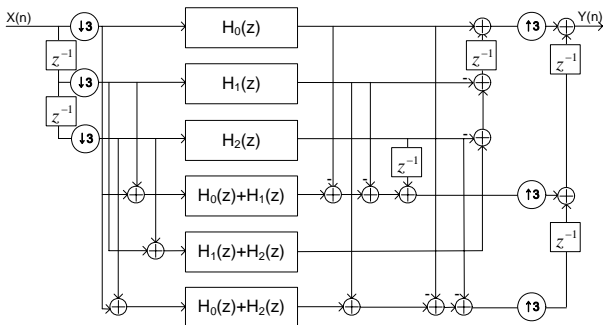


그림 1. 제안된 다운샘플링 3의 구조

III. 결론

제안된 알고리즘은 체계적인 방법으로 임의의 다운샘플링 크기로 병렬화 하는 것이 가능하다. 따라서 다운샘플링의 크기, 즉

병렬화의 크기가 결정되면 본 논문에서 제안된 알고리즘을 사용하여 쉽게 구조를 유도할 수 있게 되며, 부필터의 수도 간단히 구할 수 있는 관계식을 유도하였다. 제안된 고속 알고리즘의 효율성을 검증하기 위하여 기존구조^{[1]-[2]}와 샘플당 곱셈연산의 수, 그리고 MAC 프로세서를 사용한 구현 등의 측면에서 비교하였다.

표 1. 제안된 방식의 샘플당 곱셈연산의 수(840탭의 FIR 필터)

다운샘플링	부필터 수	부필터의 탭수	총곱셈 연산	샘플당 곱셈연산	%
1	1	840	840	840	100
2	3	420	1260	630	75.0
3	6	280	1680	560	66.67
4	9	210	1890	472.5	56.25
5	14	168	2352	470.4	56.0
6	19	140	2660	443.3	52.78
7	26	120	3120	445.7	53.06
8	33	105	3365	420.6	50.07

표 2. 제안된 고속알고리즘의 MAC프로세서 구현 명령어 수(36탭의 FIR 필터)

다운샘플링	부필터 수	부필터의 탭수	총곱셈 연산	샘플당 곱셈연산 (%)	명령어수 (%)
scalar필터	1	36	36	36 (100)	408,144 (100)
2 (방식 4)	3	18	54	27 (75)	335,326 (82.1)
2x2 (방식 4)	9	9	81	20.25 (56.25)	271,998 (66.6)
3 (방식 3)	6	12	72	24 (66.7)	312,021 (76.4)
3x3 (방식 3)	36	4	144	16 (44.4)	278,010 (68.1)
3 (제안방식)	6	12	72	24 (66.7)	301,288 (73.8)
3x3 (제안방식)	36	4	144	16 (44.4)	257,864 (63.1)
4 (제안방식)	9	9	81	20.25 (56.25)	266,886 (65.4)

샘플당 곱셈연산의 수에 대한 실험에서는 N=8의 병렬화 경우에 기본적인 N=1의 필터와 비교하여 49.93%의 샘플당 곱셈연산 감소효과를 보였다. MAC 프로세서를 사용한 구현 비교 실험은 N=3의 병렬화 경우에 대하여 기존의 구조와 비교하였으며, 기존구조와 비교하여 명령어의 수를 3.4% 감소시킬 수 있었다.

참고문헌

[1] M. Vetterli, "Running FIR and IIR filtering using multirate filter banks" IEEE Trans. Acoust., Speech, Signal Processing, vol. 36, pp. 730-738, May 1988.
 [2] Z. Mou, and P. Duhamel, "Short-length FIR filters and their use in fast nonrecursive filtering," IEEE Trans. Signal Processing, vol. 39, pp. 1322-1332, Jun. 1991.