

# 오디오 신호를 위한 표본화율 변환 알고리즘 성능 비교

이 용 희, 김 인 철  
서울시립대학교 전자전기컴퓨터공학부

## A Performance Comparison of Sampling Rate Conversion Algorithms for Audio Signal

Yong-Hee Lee, Rin-Chul Kim  
Dept. of Electrical & Computer Eng., Univ. of Seoul  
E-mail : rin@uos.ac.kr

### Abstract

본 논문에서는 지금까지 소개된 44.1KHz compact disc (CD)에서 48KHz digital audio tape (DAT)로의 표본화율 변환기법들에 대해서 가청 주파수 대역에서 100dB 이상의 dynamic range와  $\pm 5 \times 10^{-4}$ dB 이하의 리플 크기를 유지할 수 있도록 각 기법들을 재설계하였으며, 메모리 요구량 및 계산량에 대해서 살펴보고자한다.1)

### I. 서론

Compact disc (CD), digital audio tape (DAT)등은 20~20,000Hz의 가청 주파수 대역을 충실히 표현하는 고품질 오디오 시스템으로, 각각 44.1KHz, 48KHz의 표본화 주파수로 오디오 신호를 표본화 한다. 따라서 CD와 DAT 등을 동시에 재현하는 오디오 증폭기를 구현하기 위해서는 다양한 표본화 주파수를 단일 주파수로 변환시킬 필요가 있다. 이를 위해 44.1KHz를 48KHz로 변환시키는 표본화율 변환 기법에 대한 연구가 많이 수행되었다. 본 논문에서는 지금까지 소개된 표본화율 변환 기법들을 검토하고, 그 성능을 비교하고자한다.

44.1KHz에서 48KHz로 표본화 주파수를 변환하는 기본적인 기법은 먼저, 160배 up-sampling 하고,  $\pi/160$ 의 차단 주파수를 갖는 저역통과 필터를 거쳐 147배 down-sampling 하는 것이다. 이때 저역통과 필터의 입력은 대부분 0이라는 것을 참조하여 다위상 (poly-phase) 구조로 구현하면 계산 복잡도를 감축시킬 수 있다. 그리고 4배 보간 후 40/147배의 표본화율 변환을 수행하는 다단계 (multi-stage) 구조를 사용하면 보다 효과적으로 구현할 수 있다[1]. 한편, 표본화율 변환은 샘플마다 비 정수 값의 가변적인 지연을 가지는 가변적인 시스템으로 볼 수 있으므로, fractional delay filter (FDF)를 이용하여 구현할 수 있다. 이러한 방법으로는 sinc 함수를 이용한 기법[2], B-spline을 이용하는 기법[3] 그리고 가변 지연 값의 함수로 보간을 수행하는 Farrow 구조를 이용하는 기법[4] 등이 있다.

본 논문에서는 계산량 및 메모리 요구량 등의 측면에서 앞에서 언급한 4가지 방법들을 비교하고, 고품질 오디오 시스템에 적합성 여부를 검토하고자 한다.

### II. 본론

#### 2.1 실험환경 및 성능 분석 기준

지금까지 알려진 표본화율 변환 기법들은 서로 상이한 기준에 의해 그 성능들이 제시되었다. 본 논문에서는 공정한 비교를 위해 동일한 성능을 얻는데 요구되는

본 논문은 2002-3년 산자부 산업기술개발사업중 (주) 디지털엔 아닐로그 위탁 연구의 일 부분입니다.

하드웨어 복잡도를 메모리 요구량과 계산량 측면에서 검토하여, 최적의 방법을 모색하고자 한다.

본 논문에서는 먼저, 44.1KHz의 표본화 주파수에서 24 비트로 표본화된 정현파 입력을 사용한다. 그리고 사용되는 필터들의 계수들을 모두 24비트로 표현한다. 이러한 환경 하에서, 가청 주파수 대역에서 100dB 이상의 dynamic range와  $\pm 5 \times 10^{-4}$  dB 이하의 리플 크기를 유지할 수 있도록 각 기법들을 재설계하였다.

본 논문에 사용된 모든 필터는 Matlab 상에서 Blackman-Harris 윈도우를 이용한 윈도우 기법으로 설계되었다. 그리고 각 기법의 성능 또한 Matlab 상에서 비교하고 평가하였다.

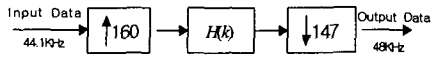
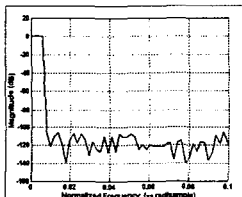


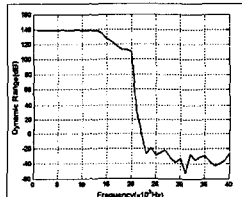
그림 1. 기본적인 기법을 이용한 구조.

2.2 기본적인 기법

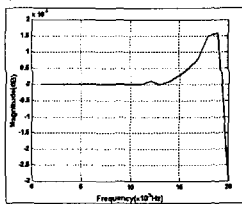
그림 1에 그 구조를 제시한 기본적인 기법에서는 입력을 160배 up-sampling하고 차단 주파수  $\pi/160$ 인 필터를 거친 다음, 147배 down-sampling하여 44.1KHz에서 48KHz로 표본화율을 변환한다. 그림 1의 구조에서  $H(k)$ 는 그림 2(a)와 같은 주파수 응답을 보이는 14241 taps 필터를 사용하였다. 그 결과, 그림 2(b) 및 (c)에 보인 바와 같이 가청 주파수 대역에서 100dB 이상의 dynamic range와  $\pm 5 \times 10^{-4}$  dB 이하 크기의 리플을 얻음을 확인하였다. 이때, 필터 계수가 대칭이라는 것을 참조하면, 7121 words의 메모리가 필요함을 알 수 있다. 또한 160배 up sampling 후에 필터를 거치므로 대



(a)  $H(k)$ 의 주파수 응답



(b) 출력 dynamic range



(c) 출력 주파수 응답

그림 2. 기본적인 기법의 성능.

부분의 값이 0이라는 점을 고려하여 다위상 (poly-phase)의 형태로 구현한다면, 한 샘플을 얻기 위해 곱셈 89회, 덧셈 88회의 연산이 필요함을 알 수 있다.

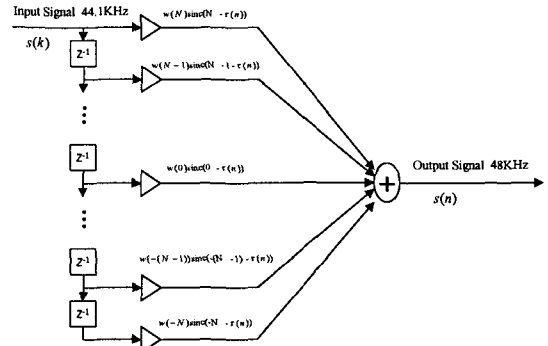


그림 3. sinc 함수를 이용한 구조.

2.3 sinc 함수를 이용한 기법

sinc 함수를 이용한 구조[2]는 그림 3과 같다. 이 구조에서는 입력 신호  $s(k)$ 에 대해 비 정수 값  $\tau(n)$  만큼의 지연이 있는 위치에서 출력 신호  $s(n)$ 을 얻는다고 보고 다음과 같은 식으로 출력을 계산한다.[2]

$$s(n) = \hat{s}[k - \tau(n)] = \sum_{m=-N}^N w(m) \text{sinc}[m - \tau(n)]s(k - m) \quad (1)$$

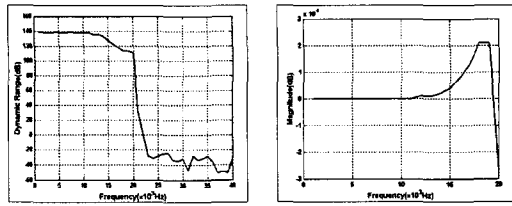
여기서  $w(m)$ 은 Blackman-Harris 윈도우로 side band의 리플을 작게 하기 위해서 사용하였다.[2] 식 (1)에서 출력을 얻기 위해 샘플 당 곱셈  $2N+1$ 회와 덧셈  $2N$ 회의 계산이 요구됨을 알 수 있다. 또한, sinc 함수 및 윈도우 함수의 곱은  $2N+1$  개의 계수로 표현된다. 한편, 44.1KHz에서 48KHz로의 표본화를 변환 비가 160/147이라는 것을 고려하면, 160개의 서로 다른  $\tau(n)$ 이 존재한다. 따라서 식 (1)을 이용하여 출력을 얻기 위해 길이가  $2N+1$ 인 필터 160개로 sinc 함수와 윈도우 함수의 곱이 표현되므로, 이 값들을 저장하는데 총  $160 \times (2N+1)$  words의 메모리가 요구된다.[2] 이러한 메모리 요구량은 식 (2)와 (3)에 보인 바와 같이

$$w(m) = w(-m) \quad (2)$$

$$\text{sinc}(m - \tau(n)) = -\text{sinc}(m + \tau(n)) \quad (3)$$

윈도우 함수가 우함수이고, sinc 함수가 기함수라는 사실을 참조한다면,  $160 \times (N+1)$  words로 감축된다.

Park[2]은 그림 3의 구조에서  $N=31$ 인 필터를 이용하여 약 83dB의 dynamic range를 얻었는데, 본 논문의 요구 조건을 만족시키기 위해서는  $N=44$ 인 필터를 사용하여야 함을 알 수 있었다. 이때, 그림 4에 보인 바와 같이 가청 주파수 대역에서 100dB 이상의 dynamic



(a) 출력 dynamic range (b) 출력 주파수 응답

그림 4. sinc 함수를 이용한 기법의 성능.

range와  $\pm 5 \times 10^{-4}$  dB 이하의 리플 크기를 보장함을 확인하였다. 즉, sinc 함수를 이용한 기법에서는 계수를 저장하기 위해 7200 (160×45) words의 메모리가 필요하고, 출력 한 샘플 당 곱셈 89회, 덧셈 88회가 요구된다.

한편, 그림 1의 기본적인 기법을 다위상 구조로 나타내면, 그림 3의 sinc 함수를 이용한 기법과 매우 유사함을 알 수 있다. 기본적인 기법의 다위상 구조는 하나의 필터를 0이 아닌 데이터의 분포에 따라 다위상 필터로 분해한 것임에 반해, sinc 함수를 이용한 기법은 이 다위상 분해를 비 정수 값으로 지연된 필터군으로 보았다는 점이 다르다. 두 기법에서의 차이는 구현된 필터의 특성에 따른 차이로 보인다.

### 2.4 다단계 구조를 이용한 기법

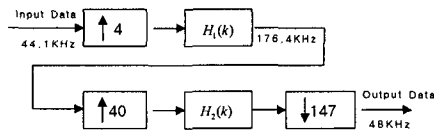
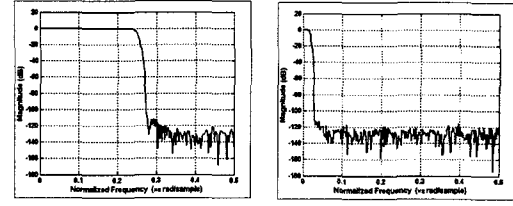


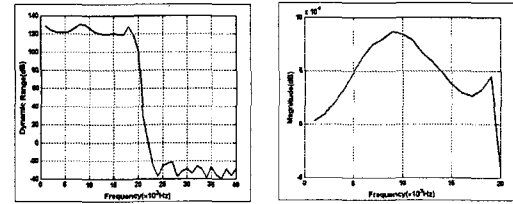
그림 5. 다단계 구조를 이용한 기법.

다단계(multi-stage) 구조를 [1] 이용하게 되면 그림 5와 같이 먼저 4배 보간하여 표본화 주파수가 176.4KHz인 신호로 변환한 다음, 40배 up-sampling과  $\pi/147$  필터 및 147배 down-sampling을 거쳐 표본화율을 변환한다. 이 기법이 본 논문에서 요구하는 성능을 만족시키려면, 차단 주파수가  $\pi/4$ 인  $H_1(k)$ 은 327 taps, 차단 주파수가  $\pi/147$ 인  $H_2(k)$ 는 781 taps이라는 필터 길이가 요구된다. 두 필터의 주파수 응답은 그림 6(a)와 (b)에 제시하였고, 이때 그림 6(c) 및 (d)에 제시한 바와 같이 100dB 이상의 dynamic range와  $\pm 5 \times 10^{-4}$  dB 이하의 리플 크기라는 요구 조건을 만족시킬 수 있다.

이 구조에서  $H_1(k)$  및  $H_2(k)$ 는 대칭 필터라는 점을 고려하면, 메모리는 555 words가 소요된다. 그리고 각각을 다위상(poly-phase)구조로 구현하면 출력 한 샘플 당 곱셈 102회, 덧셈 100회의 연산이 필요하다. 따라서



(a)  $H_1(k)$ 의 주파수 응답 (b)  $H_2(k)$ 의 주파수 응답



(c) 출력 dynamic range (d) 출력 주파수 응답

그림 6. 다단계 구조를 이용한 기법의 성능.

다단계 구조를 이용한 기법은 2.2절의 기본적인 기법 혹은 2.3절의 sinc 함수를 이용하는 기법에 비해 계산량은 비슷하지만 메모리 요구량을 감축시킨다.

### 2.5 B-spline을 이용한 기법

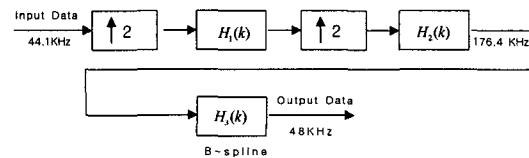


그림 7. B-spline 구조를 이용한 기법.

B-spline[3]을 이용한 보간은 이산 시간에서 정의되는  $n+1$ 개의  $x(t)$ ,  $t \in (t_0, \dots, t_n)$  을 이용해서 임의의 시간  $t_k < t < t_{k+1}$ 에서의 값  $x(t)$ 를 다음과 같이 얻는다.

$$x(t) = \sum_{k=0}^N x(t_k) B_k(t) \quad (4)$$

이때,  $B_k(t)$ 는 6차 B-spline 함수의 경우 식 (5)와 같이 구할 수 있다.[3]

$$\begin{aligned} B_0 &= \frac{1}{6!} (q)^6 \\ B_1 &= \frac{1}{6!} ((1+q)^6 - 7(q)^6) \\ B_2 &= \frac{1}{6!} ((2+q)^6 - 7(1+q)^6 + 21(q)^6) \\ B_3 &= \frac{1}{6!} ((3+q)^6 - 7(2+q)^6 + 21(1+q)^6 - 35(q)^6) \\ B_4 &= \frac{1}{6!} ((3-q)^6 - 7(2-q)^6 + 21(1-q)^6) \\ B_5 &= \frac{1}{6!} ((2-q)^6 - 7(1-q)^6) \\ B_6 &= \frac{1}{6!} (1-q)^6 \end{aligned} \quad (5)$$

여기서,  $q$ 는 서로 다른 표본화율 신호의 샘플 간 차이로,  $0 \leq q \leq 1$ 의 값을 갖게 된다. 그림 7와 같은 경우에 서로 다른 40개의  $q$ 값이 필요하다.

B-spline을 이용한 기법은 그림 7과 같이 먼저, 입력 신호를 차단 주파수가  $\pi/2$ 인  $H_1(k)$  및  $H_2(k)$ 를 사용하여 4배 up-sampling한 다음, 위에서 설명한 B-spline 보간을 통하여 표본화 주파수가 48KHz인 출력 신호를 얻는다. 여기서  $H_2(k)$ 는 B-spline의 감쇠가  $(\sin(x)/x)^7$ 과 같음을 고려하여 가칭 주파수 대역에서 그림 8(b)와 같이 이득을 보상해주는 필터를 설계하여야 한다.[3]

Cucchi등[3]에서는 필터의 길이가 각각 194, 24인  $H_1(k)$ ,  $H_2(k)$ 를 이용하였지만, 그림 8(c)와 (d) 같이 가칭 주파수 대역에서 100dB 이상의 dynamic range와 리플  $\pm 5 \times 10^{-4}$ dB 이하의 성능을 얻기 위해서는  $H_1(k)$ ,  $H_2(k)$  각각 165, 35개의 계수가 필요하고 B-spline  $H_3(k)$ 는 280(40×7)개의 계수가 필요하다. 메모리는 4배 up-sampling에서 101 words와 B-spline에서 280 words가 소요된다. 계산량은 4배 up-sampling에서 곱셈 101회, 덧셈 99회, 그리고 B-spline에서 7회의 곱셈과 6회의 덧셈이 필요하다. 즉, B-spline을 이용한 기법에서는 총 381 words의 메모리와 출력 샘플 당 곱셈 108회, 덧셈 105회의 연산이 요구된다. 만약 B-spline에서 계수를 연산을 통해 얻는다면 곱셈 37회, 덧셈 15회가 추가로 요구되고,[3] B-spline 계수를 저장하기 위한 메모리는 필요없다. 따라서 이러한 경우 메모리는 101 words, 그리고 출력 샘플 당 곱셈 138회, 덧셈 114회의 연산이 필요하다. 즉, 메모리 요구량과 계산량 등을 참조하여 B-spline 계수의 계산여부를 결정할 수 있을 것

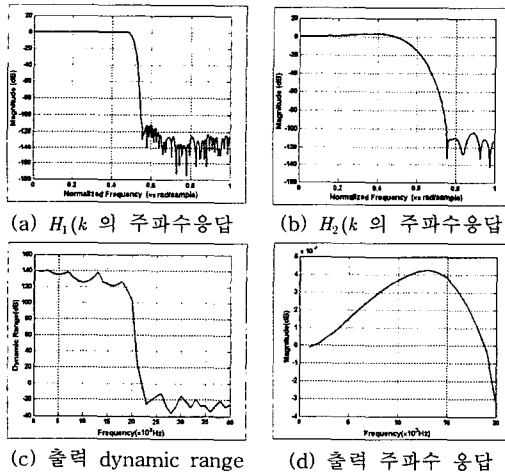


그림 8. B-spline 구조를 이용한 기법의 성능.

이다.

### III. 결론

이상에서 살펴본 4가지의 표본화율 변환 기법들에 대해서 메모리 요구량 및 계산량의 측면에서 비교를 해 보면 표 1과 같다. 각각의 값은 한 샘플을 얻는데 필요한 메모리와 계산량이며, B-spline의 경우는 계산을 통해서 결과를 메모리에 저장하고 있을 때를 의미한다.

표 1. 표본화율 변환 기법별 메모리 및 계산량.

	메모리(word) (24bit=1 word)	계산량(곱셈)
기본적인 기법	7121	89
sinc 함수	7200	89
다단계	555	102
B-spline	381	108

각각의 기법들에 대해서 24비트를 1 word로 보았을 때, 표 1과 같이 B-spline의 경우가 가장 적은 381 words가 필요한 것을 볼 수 있다. 그리고 계산량 측면에서는 기본적인 기법 및 sinc 함수를 이용한 기법이 약간 우수한 것으로 나타났다.

이러한 점을 살펴 볼 때, 표 1에서 제시한 4가지 기법들 중에서 B-spline을 이용한 기법이 계산량은 약간 많지만 메모리 요구량을 크게 감축시킬 수 있어 하드웨어 구현에 가장 적합한 기법으로 판단된다.

### 참고 문헌

- [1] A.V. Oppenheim and R.W. Schaffer, *Discrete-Time Signal Processing*. Prentice-Hall, 1998.
- [2] S. Park, "A Real-Time Method for Sample-Rate Conversion from CD to DAT," *IEEE 1990 International Conference on Consumer Electronics*, pp. 360-361, 1990.
- [3] S. Cucchi, F. Desinan, G. Parladori and G. Sicuranza, "DSP implementation of arbitrary sampling frequency conversion for high quality sound application," *International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, pp. 3609-3612, 1991.
- [4] K. Rajamani, Y. S. Lai, and C. W. Farrow, "An efficient algorithm for sample rate conversion from CD to DAT," *IEEE Signal Processing Letters*, vol. 7, no. 10, pp. 288-290, Oct. 2000.