An Efficient Algorithm for Sample Rate Conversion from CD to DAT

Kannan Rajamani, Yhean-Sen Lai, and C. W. Farrow

Abstract—**This letter unveils an efficient algorithm for sampling rate conversion (SRC) technique from 44.1 kHz compact disc (CD) to 48 kHz digital audio tape (DAT). This method involves upsampling the input signal by two, and then passing the interpolated signal through a fractional delay filter that employs a simple decimation. This method can also be used for SRC from DAT to CD without changing the filter coefficients. The proposed algorithm is simulated in Matlab and can be implemented in a realtime digital signal processor (DSP). Compared with other existing methods, the proposed method has the advantage that it requires fewer million instructions per second (MIPS) and memory.**

I. Introduction

THERE exist different sampling rate conversion algorithms from compact disc (CD) sampling rate at 44.1 kHz to DAT sampling rate at 48 kHz [1]–[4]. It is essential to change between these two different sampling rates without degrading any of the useful information in the original digital signal. There are several techniques used traditionally. The simple method explained in [1] involves converting the digital input signal into an analog signal and then resampling the analog signal into the desired sampling rate. The harmonics and noise distortion occurring in the conversion of digital to analog (D/A) and analog to digital (A/D) will degrade the overall performance. A conventional multirate technique has been popularly used for digital sampling rate conversion for a fixed ratio M/N, where M and N are positive integers [5], [7]. In the application of SRC from CD to DAT, the M and N are 160 and 147, respectively. The input samples are first interpolated up by 160 and passed through a digital low pass filter and then decimated down by 147. The implementation of this algorithm in DSP needs a lot of MIPS and hence, this method is not realistic for SRC from CD rate to DAT rate. The technique suggested in [1], [2] uses the sinc function to generate a fractionally sampled signal for the sampling rate conversion from CD to DAT. The primary drawback of this approach is that it needs a very big look-up table to store the predetermined coefficients (80 × 63 coefficients), especially in real-time DSP implementation, where the memory usage is often a key factor to cost reduction. Also, another disadvantage is that it needs a separate look-up table for SRC from DAT rate to CD rate. In [3], a hybrid sampling rate conversion method was proposed based on B-spline polynomial interpolation. The method involved upsampling the CD signal by four and then filtering the output signal using a B-spline-based fractional delay filter. Paper [3] has the advantage of performing SRC from DAT to CD without changing the filter coefficients. The disadvantage of the method is that the input signal is interpolated by a factor of four, and the fractional delay filter coefficients are computed for each delay to generate an output sample at the DAT sampling rate, resulting in lots of MEPS for real-time implementation. In this letter, we propose a memory and MIPS efficient method for sample rate conversion from CD sampling rate to DAT sampling rate that realizes the fractional delay filter using Farrow structure [4], [6]. The algorithm description is given in Section II, and the simulation results and conclusion are provided in Sections III and Section IV, respectively.

II. Algorithm Description

Consider a transfer function of a filter with a flat delay τ, which is given by

$$F(\omega, \tau) = e^{j\omega\tau}. \tag{1}$$

If the transfer function of a FIR filter with coefficients C_k with $0 \leq k < N$ and sampling interval T, then the transfer function can be written as

$$F(\omega) = \sum_{k=0}^{N-1} C_k e^{jk\omega T}. \tag{2}$$

Let a delay parameter α be such that $\tau = \alpha T$ and the coefficients C_k with $0 \leq k < N$ be the polynomial function of α with the order M. With this condition, (2) can be written as

$$F(\omega, \alpha) = \sum_{k=0}^{N-1} \sum_{i=0}^{M-1} C_{k,\,i} \alpha^i e^{jk\omega T}. \tag{3}$$

In order to determine the coefficients $C_{k,\,i}$ with $0 \leq k < N$ and $0 \leq i < M$ of an interpolator, we minimize the mean-square value of E, defined as

$$E^2 = \int_{w_n 0}^{w_n 1} \int_{\alpha_n 0}^{\alpha_n 1} \left| \sum_{k=0}^{N-1} \sum_{i=0}^{M-1} C_{k,\,i} \alpha^i e^{jk\omega T} - e^{j\omega\alpha T} \right|^2 d\alpha\, d\omega \tag{4}$$

where $\omega_n 0$ and $\omega_n 1$ are the lower and upper band frequencies of the signal, respectively, $\alpha_n 0 = -(1/2)$, and $\alpha_n 1 = 1/2$. The optimal matrix $C_{k,\,i}$ with $0 \leq k < N$ and $0 \leq i < M$ can be solved from (4) as

$$\sum_{k=0}^{N-1} \sum_{i=0}^{M-1} C_{k,\,i} \alpha^i \cos(k\omega T) = \cos(\alpha\omega T) \tag{5a}$$

Manuscript received December 9, 1999. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Y. Shoham.

The authors are with Bell Laboratories, Lucent Technologies, Holmdel, NJ 07733 USA (email: ylai@lucent.com).

Publisher Item Identifier S 1070-9908(00)07542-8.

1070–9908/00$10.00 © 2000 IEEE

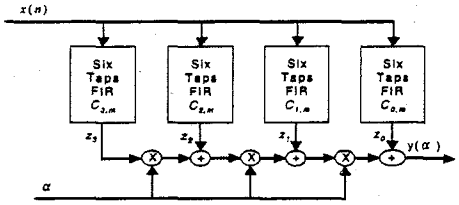

Fig. 1. Structure of third order fractional delay filter.

and

$$\sum_{k=0}^{N-1} \sum_{i=0}^{M-1} C_{k,i} \alpha^i \sin(k\omega T) = \sin(\alpha\omega T). \tag{5b}$$

By choosing the M particular values of α with $-(1/2) \leq \alpha \leq 1/2$, and $(N/2)$ (N = even number) or $(N-1)/2$ (N = odd number) values of ω, the $C_{k,i}$ with $0 \leq k < N$ and $0 \leq i < M$ can be computed through a matrix operation easily with MN variables and MN equations.

With the $C_{k,i}$'s, a fractional delay filter (FDF) can be formed as a FIR filter whose tap coefficients are a function of the desired delay. Then we have

$$y(\alpha) = \sum_{k=0}^{N-1} x(n-k) C_k(\alpha \tag{6}$$

where

y output sample at the desired delay α;
$x(n)$ input samples;
N order of FIR for a FDF.

From above discussion, $C_k(\alpha)$ can be expressed as a linear combination of polynomial with order M in α as in (7)

$$C_k(\alpha) = \sum_{i=0}^{M-1} C_{k,i} \alpha^i. \tag{7}$$

In the equation (7), we can rewrite equation (6) as follows:

$$z_i = \sum_{k=0}^{N-1} x(n-k) C_{k,i} \tag{8}$$

$$y(\alpha) = \sum_{i=0}^{M-1} \alpha^i z_i. \tag{9}$$

A structure of third order filter with six taps FIR is shown in Fig. 1. Fig. 2 illustrates the proposed structure using an interpolator and a fractional delay FIR filter. The interpolator consists of a symmetric FIR filter to increase the sampling rate by a factor of two, implemented as a polyphase structure [7]. In the application of sampling rate conversion from CD to DAT, it is very difficult to design a good fractional delay filter in third order to maintain a high quality of performance. Hence we employ an interpolator to interpolate the incoming samples up by a factor of two, and the output signal passes through a fractional delay filter to generate the desired sampling rate. Also, the fractional delay filter coefficients need not be computed every time, as a new sample gets into the delay line of the FDF. Only the α value changes as the new sample comes in. The unit of fractional delay α computation and input sample control in Fig. 2 generates the α value for each output sample at 48 kHz sampling rate, and it also determines whether to shift s input samples into the delay line of fractional delay filter. This unit is based on the fact that for every 147 input samples the filter generates 80 output samples. In other words, at every 80 output samples transfers, the input and output sampling instants will be synchronized again. Let α_n be the delay value at the nth output sample and s_n be the number of input samples shifted into the delay line of fractional delay filter from the data buffer. Then we have

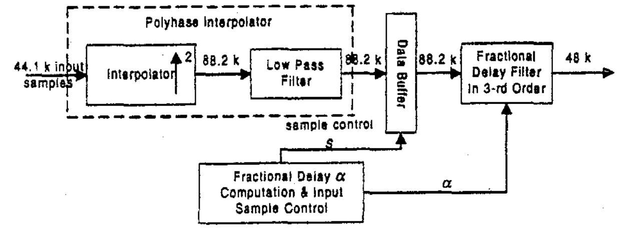

Fig. 2. Proposed structure for sampling rate conversion for CD to DAT.

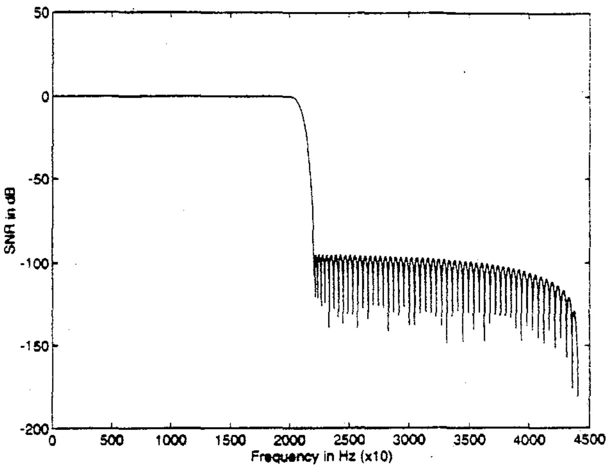

Fig. 3. Magnitude response of the proposed sample rate convertor.

$$\alpha_n = n\tfrac{147}{80} - \left\lfloor n\tfrac{147}{80} \right\rfloor, \tag{10}$$

$$s_n = \left\lfloor n\tfrac{147}{80} \right\rfloor - \left\lfloor (n-1)\tfrac{147}{80} \right\rfloor \tag{11}$$

where $n = 0, 1, 2, \cdots, 79$, and $\lfloor \times \rfloor$ denotes the largest integer but less than or equal to x. In order to save the MIPS for this application, (10) and (11) can be replaced by two table, where each table has 80 values. The proposed configuration can also be used for the sampling rate conversion from DAT to CD rate. In this application, the structure generates 147 output samples for every 320 input samples. Hence, (10) and (11) will be changed into

$$\alpha_n = n\tfrac{320}{147} - \left\lfloor n\tfrac{320}{147} \right\rfloor, \tag{12}$$

$$s_n = \left\lfloor n\tfrac{320}{147} \right\rfloor - \left\lfloor (n-1)\tfrac{320}{147} \right\rfloor \tag{13}$$

where $n = 0, 1, 2, \cdots, 146$.

III. Simulation Results

The simulation of our proposed sampling rate conversion algorithm has been implemented in Matlab with a 192-tap linear phase FIR interpolator. The SNR of the interpolation filter and the fractional delay filter with $\alpha = 0$ is shown in

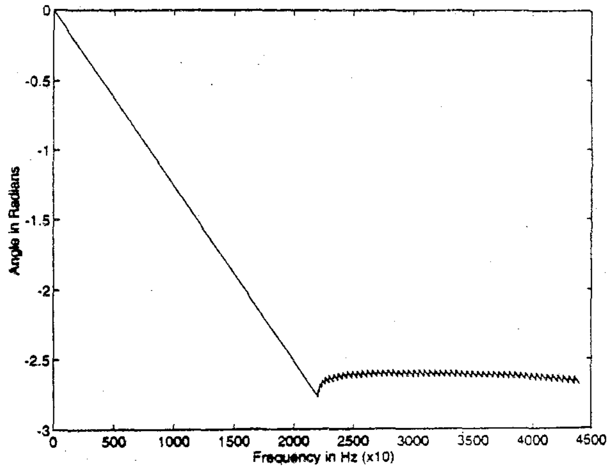

Fig. 4. Phase response of the proposed sample rate convertor.

Figs. 3 and 4. The result shows that the conversion maintains a very high quality of SNR close to 96 dB at the output signal. The proposed structure, called proposed method I, can be changed to have a look-up table for implementing the fractional delay filter, called proposal method II. The look-up table consists of fractional delay filter coefficients for different α (e.g., 80 in the case of CD to DAT). A comparison of memory and MIPS required among the proposed technique with other existing methods is shown in Table I. From Table I, it is obvious that the proposed method takes many fewer MIPSs with about same amount of memory and retaining the same degree of SRC flexibility as compared to method [3]. The proposed method has a superior advantage in memory usage and requires few more MEPSs in comparison with method [1]. Also, the proposed method can be used for sample rate conversion from DAT to CD without changing the filter coefficients, whereas the method [1] needs a separate look-up table increasing the memory requirements still further.

TABLE I
MEMORY AND MIPS COMPARISON AMONG THE PROPOSED METHOD AND OTHER EXISTING METHODS

	Proposed Method-I	Proposed Method-II	Method [1]	Method [3]
Memory	376 Words	832 Words	5040 Words	383 Words
MIPS	5.9	4.9	3.1	14.4

IV. CONCLUSION

A digital SRC technique has been presented that uses a digital interpolator followed by a fractional delay filter. The effectiveness of the proposed algorithm in terms of memory and MIPS usage for SRC with other existing methods has been described. This method can also be extended for sampling rate conversion from DAT to CD. The proposed algorithm can be easily implemented in a DSP for a real-time sampling rate conversion.

REFERENCES

[1] S. Park, "A real-time method for sample-rate conversion from CD to DAT," in *Proc. IEEE Int. Conf. Consumer Electronics*, Chicago, IL, June 18–20, 1990, pp. 360–361.

[2] S. Park, G. Hillman, and R. Robles, "A novel structure for real-time digital sample-rate converters with finite precision error analysis," *Int. Conf. Acoust. Speech Signal Processing*, pp. 3613–3616, 1991.

[3] S. Cucchi, F. Desinan, G. Parladori, and G. Sicuranza, "DSP implementation of arbitrary sampling frequency conversion for high quality sound application," *Int. Conf. Acoust. Speech Signal Processing*, pp. 3609–3612, 1991.

[4] C. W. Farrow, "A continuously variable digital delay element," *Int. Symp. Circuits and Systems*, pp. 2641–2645, 1988.

[5] D. F. Elliott, Ed., *Handbook of Digital Signal Processing Engineering Application.* New York: Academic, 1987.

[6] K. Rajamani, Y. S. Lai, and C. W. Farrow, "A novel structure for sample rate conversion from CD to DAT,", patent pending, 1999.

[7] R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1983.