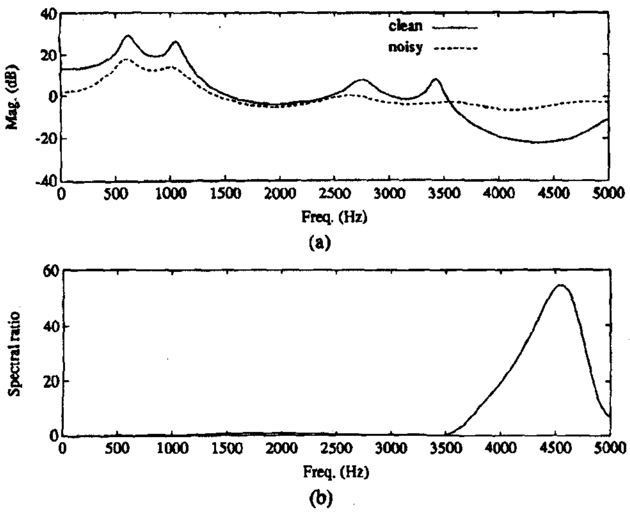

Fig. 6. (a) LPC spectrum for the nonpreemphasized clean speech signal (solid line) and the nonpreemphasized noisy signal (dashed line). (b) The spectral ratio of the noisy spectrum to the clean spectrum in (a).

figure shows that the proposed weighting procedure has significantly reduced the spectral distortion between the fourth formant and the Nyquist frequency compared to the Soong and Sondhi weighting function.

IV. Discussions

The preemphasis of the speech signal before LPC analysis is routinely used in many speech processing applications. Paliwal [4], in an experiment using clean speech data, has shown that preemphasis of the speech signal degrades vowel recognition performance. But he also conjectures that for consonant and isolated word recognition, preemphasis may be beneficial. He does not consider the case where the recognizer is trained and used under different noise conditions.

Consider the case where all AR models are from the nonpreemphasized data. If the noise conditions differ between training and testing of the recognizer, the FWIDM can be used in this case also to reduce the spectral distortion between the reference and test AR models. The weighting function can be the "bandwidth-broadened" test spectrum. Is it better to use nonpreemphasized data than preemphasized data with the FWIDM? An additional experiment was done to compare these two cases. Fig. 6(a) compares the LPC spectrum of the clean, nonpreemphasized speech signal to the LPC spectrum of the noisy, nonpreemphasized signal. The data are the same as in the previous examples. Fig. 6(b) is the spectral ratio of the noisy to the clean spectrum. The FWIDM between the clean and noisy spectrum was calculated for different values of α, and the results are shown in Fig. 7. The bandwidth-broadened nonpreemphasized test spectrum was used as the weighting function.

Comparing Figs. 7 and 4, it is clear that the reduction in the distance obtained is of the same order for both the preemphasized and the nonpreemphasized signals. This result implies that, in terms of the reduction in distance possible with the frequency weighting procedure, both approaches yield similar results. Using nonpreemphasized signals has the advantage of saving a pth-order LPC computation. However, this does not answer the more general question of how preemphasis affects the *recognition accuracy* of an isolated word recognition system.

V. Summary

It is shown that the frequency-weighted Itakura spectral distortion measure described by Soong and Sondhi [1] does not yield any improvements over the standard Itakura distance if the noise is added to the speech signal before preemphasis. We considered a modified frequency-weighted Itakura distance that uses the bandwidth-broadened test LPC spectrum of the *nonpreemphasized* speech signal as the weighting function. With this weighting function, the distance between the clean reference spectrum and the noisy test spectrum is reduced compared to the Itakura distance.

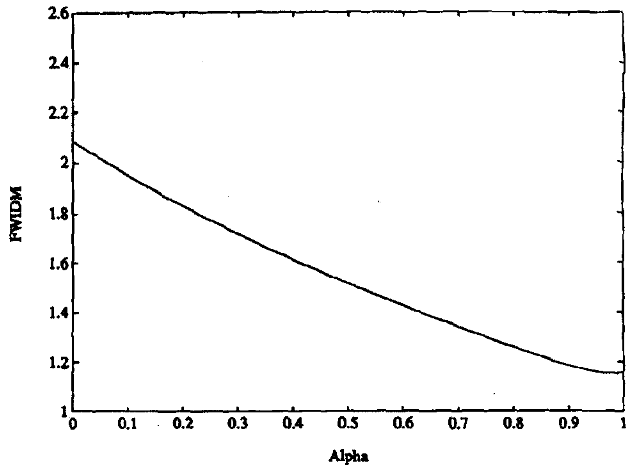

Fig. 7. The variation of d_{wI} with α when using nonpreemphasized data. The weighting function is the nonpreemphasized test spectrum.

References

[1] F. K. Soong and M. M. Sondhi, "A frequency-weighted Itakura spectral distortion measure and its application to speech recognition in noise," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 41-48, Jan. 1988.

[2] F. Itakura, "Minimum prediction residual principle applied to speech recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-23, pp. 67-72, 1975.

[3] J. Markel and A. Gray, *Linear Prediction of Speech*. New York: Springer-Verlag, 1976.

[4] K. K. Paliwal, "Effects of preemphasis on vowel recognition performance," *Speech Commun.*, vol. 3, pp. 101-106, 1984.

An Iterative Method to Compensate for the Interpolation Distortion

FAROKH A. MARVASTI

***Abstract*—In this correspondence, we show that iterative techniques can be used to compensate for sample-and-hold or, in general, interpolation distortion. Simulation results show that the iterative method is a powerful tool for reducing distortion compared to other well-known techniques.**

I. Introduction

There are many applications in digital signal processing and communication systems where interpolating functions are used to reconstruct the original signal from its uniform samples. D/A con-

Manuscript received August 29, 1988; revised December 30, 1988.

The author is with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL 60616.

IEEE Log Number 8930020.

0096-3518/89/1000-1617$01.00 © 1989 IEEE

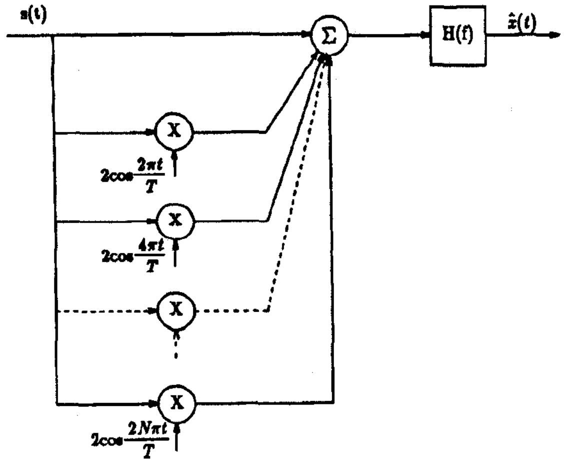

Fig. 1. The harmonic method.

verters use zero- or first-order hold as a specific interpolating function. The process of interpolating (e.g., sample and hold) and low-pass filtering, unlike the ideal sampling, distorts the original signal. The classical way to reduce the distortion is to sample at a higher rate than the Nyquist rate. This high rate is not always acceptable in signal transmission and processing where bandwidth and memory become a limitation. An alternative method is to equalize distortion by a compensating filter, e.g., $\mathrm{sinc}^{-1}(\cdot)$ for sample-and-hold interpolation. However, this filter is not realizable and can only be approximated. A modular method was proposed by the author [1] for the compensation of interpolation distortion at the Nyquist rate; this method consists of mixing the interpolating function with a cosine wave and its harmonics and then low-pass filtering the resultant signal—Fig. 1. Reference [1] also showed that this modular technique outperforms the method of inverse filtering as far as noise is concerned.

In this paper, we propose an iterative technique to compensate for the interpolation distortion.[1] We shall prove that this iterative method converges to the true signal in the limit. Finally, the simulation results of the iterative technique are compared to the modular method.

II. The Iterative Method to Compensate for the Interpolation Distortion

The interpolation function of a discrete signal is either modeled as the output of a linear time-invariant system when the input is the discrete impulsive samples or the interpolation function is characterized by a polynomial fit. An example of the first type is a zero-order hold (or sample-and-hold) or, in general, an nth-order hold interpolation. An example of the second case is spline functions. In this section, we shall prove that iterative methods can be used to recover the original analog signal from the interpolated discrete signal. We shall first consider a zero-order-hold (sample-and-hold) interpolation, which is widely used in digital-to-analog conversion.

A. Iterative Method for Sample-and-Hold Distortion

The iteration is given by

$$x_{k+1}(t) = \lambda PSx(t) + (P - \lambda PS)x_k(t) \tag{1}$$

where λ and $x_k(t)$ are the convergence factor and the kth iteration, respectively; $x_0(t)$ can be any function of time, but $x_0 = PSx(t)$ would be the most appropriate choice for the fastest convergence. P and S are, respectively, low-pass and sample-and-hold operators. PSx is the low-pass version of the sample-and-hold and is known. Given the above definitions for operators P and S, (1) can be written in the following form:

$$x_{k+1}(t) = \lambda h(t) * x_s(t) + h(t) * x_k(t) - \lambda h(t) * x_{sk}(t) \tag{1'}$$

where $*$ is the convolution sign, and $x_s(t)$ and $x_{sk}(t)$ are, respectively, the sample-and-hold versions of $x(t)$ and $x_k(t)$, i.e.,

$$x_s(t) = \sum_n x(nT)\,\Pi\left(\frac{t-n}{T} - \frac{1}{2}\right)$$

and

$$x_{sk}(t) = \sum_n x_k(nT)\,\Pi\left(\frac{t-n}{T} - \frac{1}{2}\right).$$

Equation (1) converges to $x(t) = \lim_{k\to\infty} x_k(t)$ if we have contraction [3], i.e.,

$$\|x_{k+1} - x_k\| < \|x_k - x_{k-1}\| \qquad \text{for all } k$$

where $\|x\|$ is defined as the norm of x in the L_2 Hilbert space or, alternatively,

$$\|P(x_k(t) - x_{k-1}(t)) - \lambda PS(x_k(t) - x_{k-1}(t))\| \le r\|x_k - x_{k-1}\| \tag{2}$$

where $0 \le r < 1$.

Therefore, if we can prove that the sample-and-hold distortion satisfies (2), then the iterative method described by (1) will recover $x(t)$ in the limit. The proof is as follows. By invoking the Parseval's theorem, the left-hand side of (2) is equivalent to

$$\Big\| X_k(f) - X_{k-1}(f) - \lambda\,\Pi\,(fT)\,\mathrm{sinc}\,(fT) \cdot \sum_i \left[X_k\left(f - \frac{i}{T}\right) - X_{k-1}\left(f - \frac{i}{T}\right)\right]\Big\| \tag{3}$$

where $\Pi\,(fT)$ is an ideal low-pass filter with a cutoff frequency at $f = 1/2T$ and $\mathrm{sinc}\,(fT)\,\Sigma_i\,[\cdot]$ is the Fourier transform of the sample-and-hold operation. If we assume that the sampling rate is greater than or equal to the Nyquist rate, we can write

$$\Pi\,(fT)\,\mathrm{sinc}\,(fT)\sum_i\left[X_k\left(f - \frac{i}{T}\right) - X_{k-1}\left(f - \frac{i}{T}\right)\right] = \mathrm{sinc}\,(fT)[X_k(f) - X_{k-1}(f)]. \tag{4}$$

Combining (3) and (4), we get

$$\|[X_k(f) - X_{k-1}(f)][1 - \lambda\,\mathrm{sinc}\,fT]\| \le \|X_k - X_{k-1}\|[1 - \lambda\,\mathrm{sinc}\,fT]_{\max}. \tag{5}$$

Now, if $0 < \lambda < 2$, the right-hand side of (5) compared to (2) is a contraction, i.e.,

$$0 < r = \max|1 - \lambda\,\mathrm{sinc}\,fT| < 1.$$

For example, for $\lambda = 1 \Rightarrow r = (1 - 2/\pi) < 1$.

Therefore, an iterative technique of the form (1) will converge to the original signal in the limit. The above proof could be generalized for an nth-order hold. For instance, for a linear interpolation, similar analysis shows that the contraction is of the form

$$r = \max|1 - \lambda\,\mathrm{sinc}^2 fT|, \qquad 0 \le \lambda < 2, \tag{6}$$

e.g., for $\lambda = 1 \Rightarrow r = (1 - 4/\pi^2)$.

B. Iterative Methods for Polynomial Interpolation

Polynomial interpolation such as spline fitting is a nonlinear/time-varying process. The iterative methods are still valid provided that

$$\|(P - \lambda PS)(X_k - X_{k-1})\| \le r\|X_k - X_{k-1}\| \tag{7}$$

where $0 \le r < 1$.

[1]For ideal nonuniform samples, see [2] and [4].

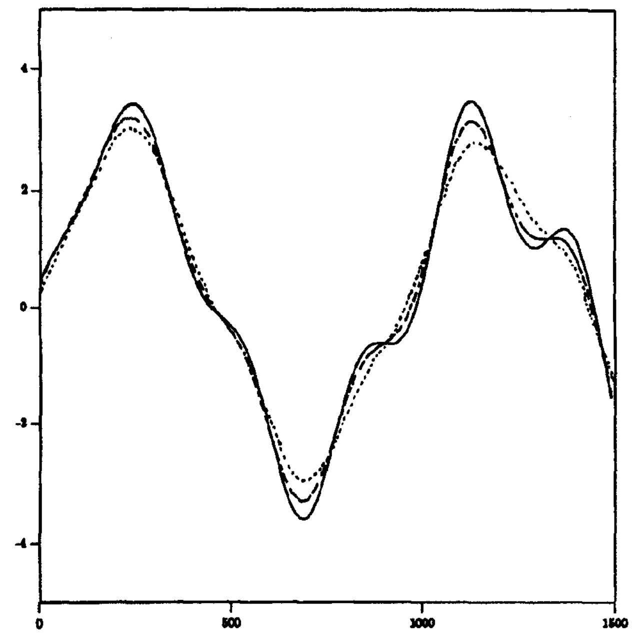

Fig. 2. —— Original signal. - - - Recovered by filtering zero-order interpolation function. . . . . Recovered by adding one harmonic to interpolation function.

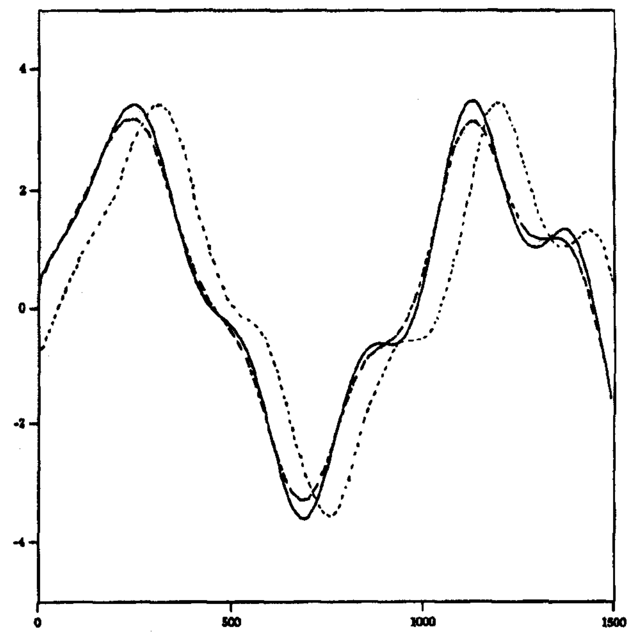

Fig. 4. —— Original signal. - - - Recovered by filtering zero-order interpolation function. . . . . Recovered by adding 60 harmonics to interpolation function (the delay is due to the sample and hold).

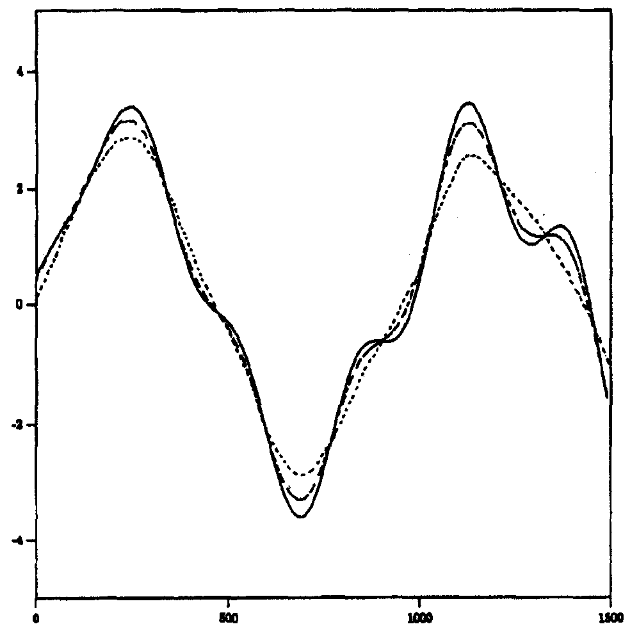

Fig. 3. —— Original signal. - - - Recovered by filtering zero-order interpolation function. . . . . Recovered by adding four harmonics to interpolation function.

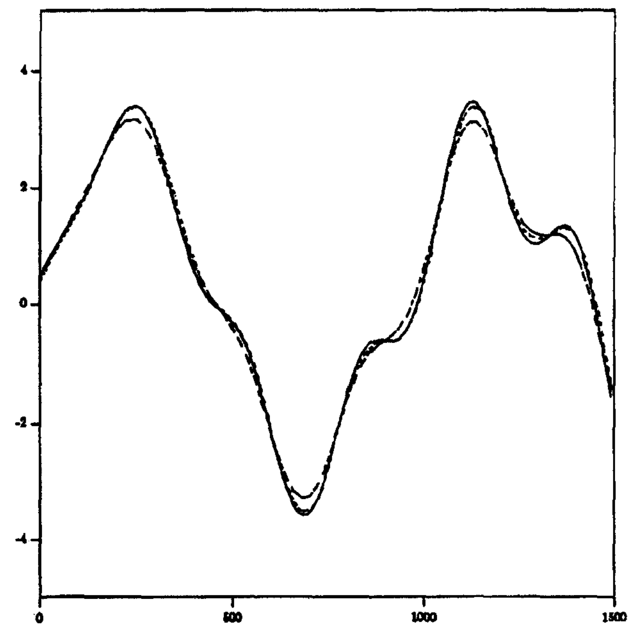

Fig. 5. —— Original signal. - - - Recovered by filtering zero-order interpolation function. . . . . Recovered by one iteration.

λ and P are defined as before and S is the spline fit. In general, the inequality represented in (7) is true because we expect to get

$$PS(x_i - x_j) \approx (x_i - x_j) \quad \text{for } \lambda = 1. \tag{8}$$

Otherwise, the interpolation loses its meaning. If (8) is true, then (7) is true, and therefore, the iterative method (1) will recover the original signal from the polynomial interpolation. In order to prove (7), we need the order of the polynomial and the way the polynomial is interpolated. We also expect to get the same results with fewer iterations when the interpolation is better.

III. Simulation Results

An almost periodic signal was simulated on the computer; this signal is given below:

$$x(t) = \frac{1}{2}\sin\frac{2\pi t}{278} + \frac{1}{2}\cos\left(\frac{2\pi t}{300}\right) + \frac{3}{2}\cos\frac{2\pi t}{900} + \frac{3}{2}\sin\frac{2\pi t}{1000}.$$

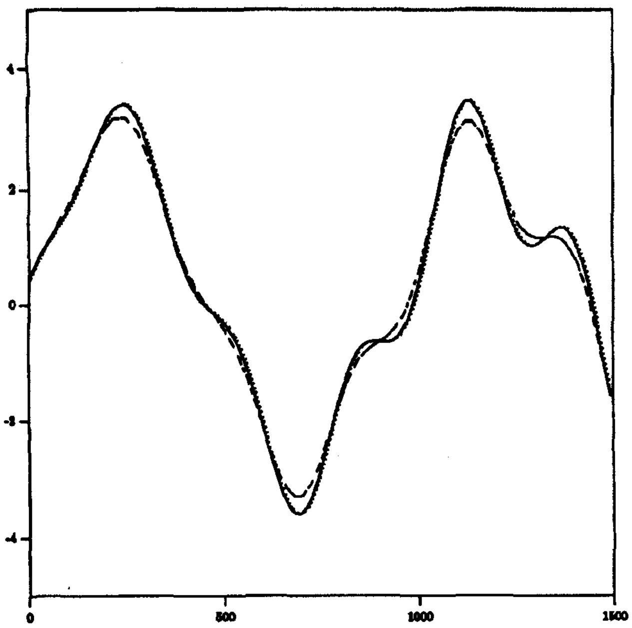

Fig. 6. —— Original signal. --- Recovered by filtering zero-order interpolation function. .... Recovered by four iterations.

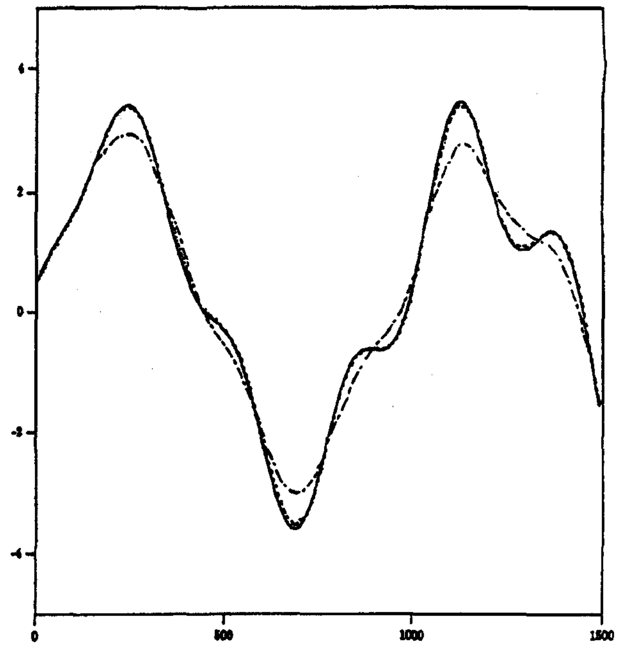

Fig. 8. —— Original signal. --- Recovered by filtering first-order interpolation function. .... Recovered by adding four harmonics to interpolation function.

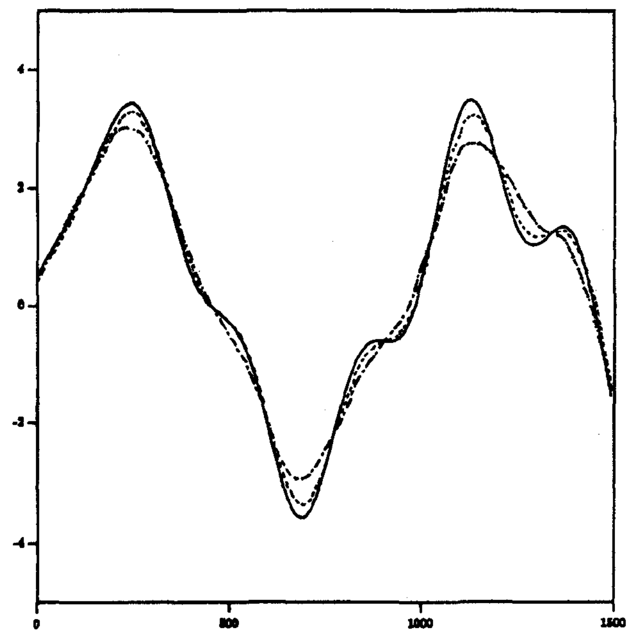

Fig. 7. —— Original signal. --- Recovered by filtering first-order interpolation function. .... Recovered by adding one harmonic to interpolation function.

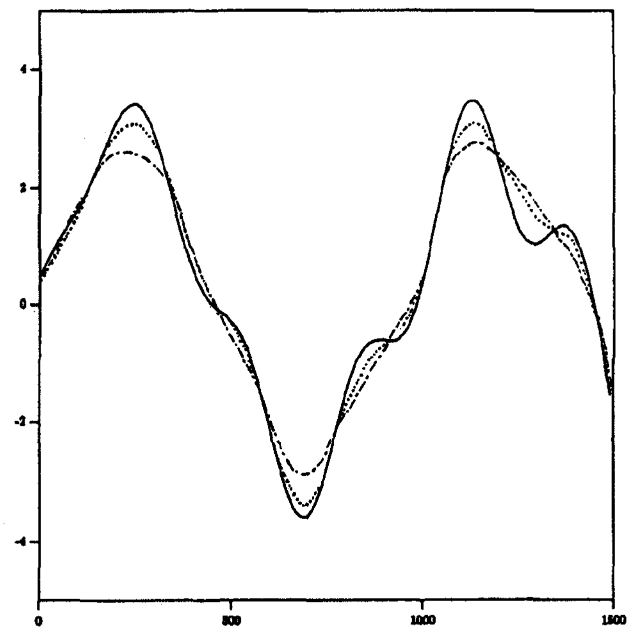

Fig. 9. —— Original signal. --- Recovered by filtering first-order interpolation function. .... Recovered by one iteration.

The maximum frequency component of $x(t)$ in the above equation is $1/278$ where 278 is the period of the first sine wave in units of time. The sampling rate is chosen to be $1/125$, which is slightly higher than the Nyquist rate $(2/278 = 1/139)$. Fourteen-thousand points of $x(t)$ are stored in a computer (each point represents one unit of time). To avoid transient effects at end points, only the points between 7000 and 8499 are plotted in Figs. 2–10 (denoted as 0–1500 on the horizontal axis).

In the iterative method, the value of λ in (1) is taken as 1, which turns out to be the best choice for fastest convergence [5]. The first iteration $x_0(t)$ is taken to be the low-pass filtered version of the interpolated signal (namely, the zero- and first-order hold).

An ideal low-pass filter with 4000 samples and a Hamming window is used throughout the simulation. The zero-order hold of this signal is low-pass filtered and then compared to the modular and the iterative methods (Figs. 2–6). The modular technique (denoted by adding harmonics to interpolation function) performs poorly for the zero-order hold for this type of signal (it takes 60 modules!). However, the iterative method does astonishingly well after one

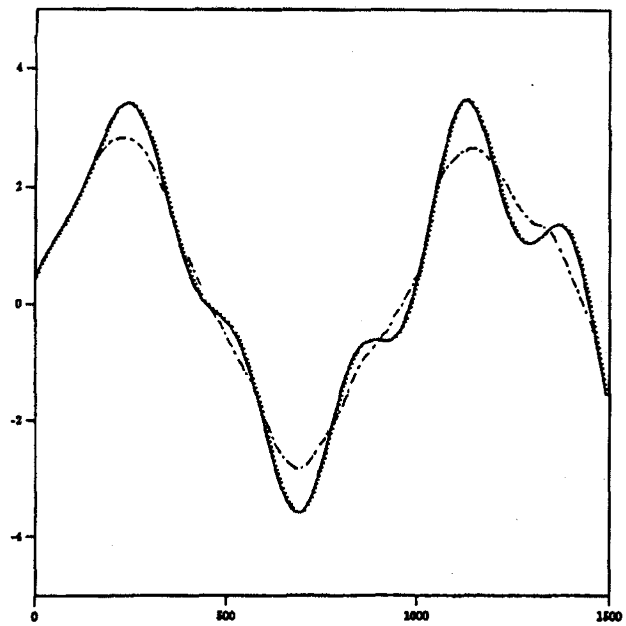

Fig. 10. —— Original signal. - - - Recovered by filtering first-order interpolation function. .... Recovered by four iterations.

iteration and almost converges after four iterations. The results for the linear interpolation are shown in Figs. 7-10. The modular method (Figs. 7 and 8) performs quite well and is almost the same as the iterative technique (Figs. 9 and 10). The computer time for the iterative method is, however, much more than the modular method. The circuit implementation of both methods is relatively simple. The modular method requires mixers, but the iterative method requires adders and extra filters with compensating delay lines.

Computationally, the modular method requires $N + M - 1$ additions and $N + 1$ multiplications (where N, M are the total number of filter samples and the number of added harmonics, respectively). On the other hand, the iterative method requires $(M + 1)(N - 1)$ additions and $(M + 1)N$ multiplications (where M is the number of iterations).

IV. Conclusion

We have proven that the iterative technique can be used to compensate for any kind of interpolation distortion; the most common type is sample-and-hold distortion. The simulation results show that the iterative method performs equal to or better than the modular technique. The computer time of the iterative technique is, however, a disadvantage.

Acknowledgment

I would like to thank S. Aghagolzadeh for simulation results and C. Liu for typing and proofreading this correspondence.

References

[1] F. Marvasti, "A new method to compensate for the sample-and-hold distortion," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, June 1985.

[2] ——, "A unified approach to zero-crossings and nonuniform sampling of single and multidimensional signals and systems," Oak Park, IL: NONUNIFORM Publishing, 1987.

[3] I. W. Sandberg, "On the properties of some systems that distort signals I," *Bell Syst. Tech. J.*, vol. 42, Sept. 1963.

[4] F. Marvasti and M. Analoui, "Recovery of signals from nonuniform samples using iterative methods," in *Proc. Int. Symp. Circuits Syst.*, Portland, OR, May 1989.

[5] R. Schafer, R. Mersereau, and M. Richards, "Constrained iterative restoration algorithms," *Proc. IEEE*, vol. 69, Apr. 1981.

Modified Finite Limiter for a Band-Limited Gaussian Random Process with Applications to A/D Conversion

DENNIS R. MORGAN

Abstract—**An analysis of finite limiting effects for a band-limited Gaussian random process was presented in [1], where it was applied to calculate the signal-to-distortion-plus-quantization-noise ratio (SDQR) of a uniform A/D converter as a function of input power. In this correspondence, a slightly modified limiter is considered in order to obtain a better representation of the input signal. For light clipping (≤ 1 percent), the modified limiter reduces the distortion by approximately 3.2 dB. However, when quantization noise is added in, only a modest (<1 dB) improvement is obtained in the peak SDQR, owing to the steepness of the distortion curve.**

I. Introduction

An analysis of finite limiting effects for a band-limited Gaussian random process was presented in [1], where it was applied to calculate the signal-to-distortion-plus-quantization-noise ratio (SDQR) of a uniform A/D converter. A suggestion has been received [2] to modify the A/D converter by interpreting the highest quantization level to be of slightly higher value, thereby reducing the distortion associated with finite limiting. This modification is explored here in order to establish the performance advantage that can be so obtained. The principal results of [1] that are pertinent to this discussion are first summarized below.

The input $x(t)$ to a finite limiter is assumed to be a band-limited random process with unit power, and the output $y(t)$ is formed by limiting the input at level l, which is written

$$y(t) = \begin{cases} l, & x \geq l \\ x(t), & |x| \leq l \\ -l, & x \leq -l. \end{cases} \tag{1}$$

The output can also be expressed as

$$y(t) = \alpha x(t) + e(t), \tag{2}$$

where α is the fraction of input signal that is passed undistorted to the output, and $e(t)$ is the error or distortion introduced by the limiting process. For a Gaussian input process, the error and the input are uncorrelated for the value

$$\alpha = E\{xy\} = \sqrt{2/\pi} \int_0^l e^{-x^2/2}\, dx = \operatorname{erf}(l/\sqrt{2}), \tag{3}$$

where $E\{\cdot\}$ is the statistical expectation operator and

$$\operatorname{erf}(x) \equiv \frac{2}{\sqrt{\pi}} \int_0^x e^{-u^2}\, du$$

is the error function.

Manuscript received July 13, 1988; revised January 11, 1989.
The author is with AT&T Bell Laboratories, Whippany, NJ 07981.
IEEE Log Number 8930013.

0096-3518/89/1000-1621$01.00 © 1989 IEEE