Audio Coding

Introduction

A central principle of data compression is that if a data stream has some describable structure then you can use that to reduce the number of bits need to represent the stream. For voice streams, the structure arises from the amplitude statistics of speech, and from the spectral characteristics due to the nature of the vocal tract and human language.

Speech PDF

Empirically, in speech, low sound levels occur much more frequently than high levels. If $x(t)$ is a speech signal, then a pretty good model for the probability density function of x is the *Laplacian distribution*

$$p_x(x) = \frac{1}{\sqrt{2}\sigma_x} e^{-\sqrt{2}|x|/\sigma_x} \tag{31.1}$$

Here $\langle x \rangle = 0, \langle x^2 \rangle = \sigma_x{}^2$. When viewed on a semilog plot, this has a triangular shape. Figure 31.1 shows an example.

Quantization

To go from the analog domain to the digital domain, we need to sample and quantize the signal $x(t)$. We know from the Sampling Theorem that the sampling frequency must be at least twice the highest frequency component of $x(t)$. For phone-quality audio, a sampling frequency f_s of 8 kHz is standard. For CD-quality sound a sampling frequency of 44.1 kHz is standard.

Quantization involves limiting ourselves to a finite number of discrete signal levels. If we have L levels then this will translate into $\log_2 L$ bits per sample. Our bit rate is then $f_s \log_2 L$. Clearly we want L to be as small as possible. We choose some quantization function $y = Q(x)$ that produces an output $y = y_k$ for an input in the interval $x_k \le x < x_{k+1}$. We then assign $\log_2 L$ bit binary codes to those levels.

As an example of a *uniform quantizer*, consider

$$Q(x) = \Delta \,\text{round}(x/\Delta) \tag{31.2}$$

where "round" gives the nearest integer. Here Δ is the *step size.* This is illustrated in Fig. 31.2. The output is $y_k = k\Delta$ for $(k-0.5)\Delta \le x < (k+0.5)\Delta$. The difference between y and x is the *quantization noise*.

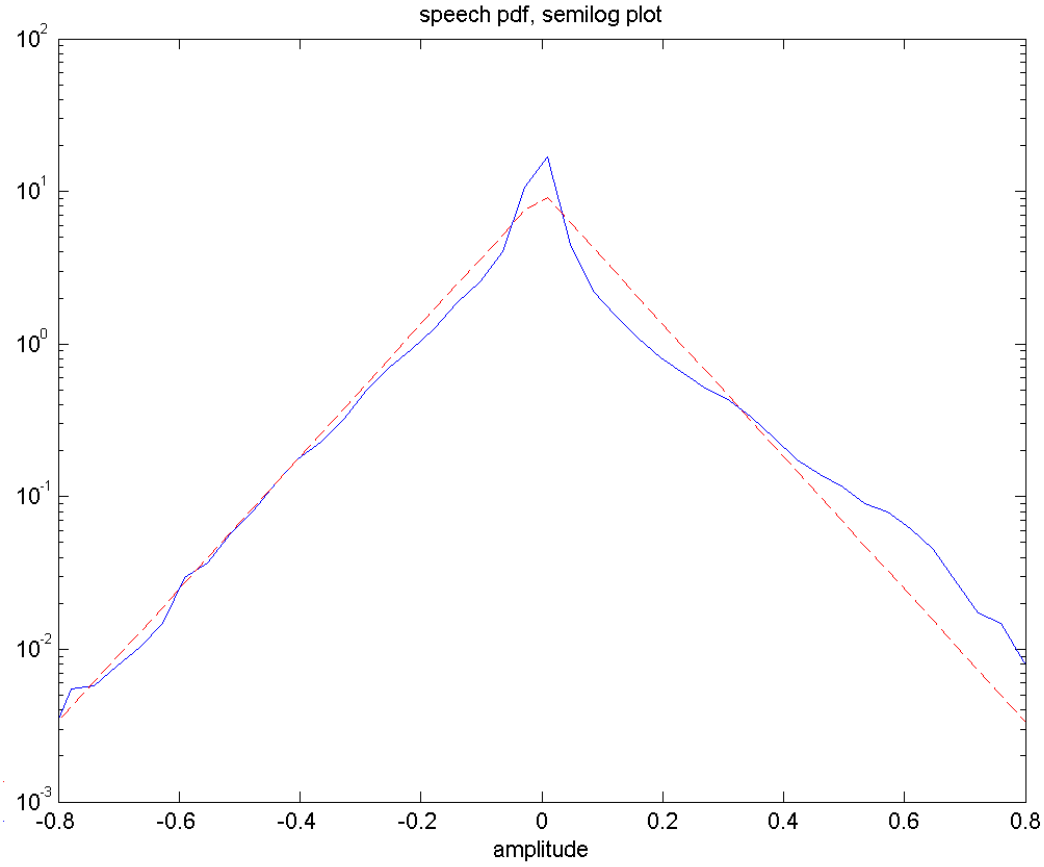


Figure 31.1: PDF of a few seconds of speech. Solid blue curve is measured pdf; dashed red curve is Laplacian distribution.

The variance of the quantization noise is

$$\begin{aligned}\sigma_q^{\,2} &= \int_{-\infty}^{\infty}\left[x - Q(x)\right]^2 p_x(x)dx \\ &= \sum_k \int_{x_k}^{x_{k+1}}\left[x - y_k\right]^2 p_x(x)dx\end{aligned} \tag{31.3}$$

We can then define a signal-to-quantization noise ratio

$$S/N_{dB} = 10\log\frac{\sigma_x^{\,2}}{\sigma_q^{\,2}} \tag{31.4}$$

For the uniform quantizer we can calculate

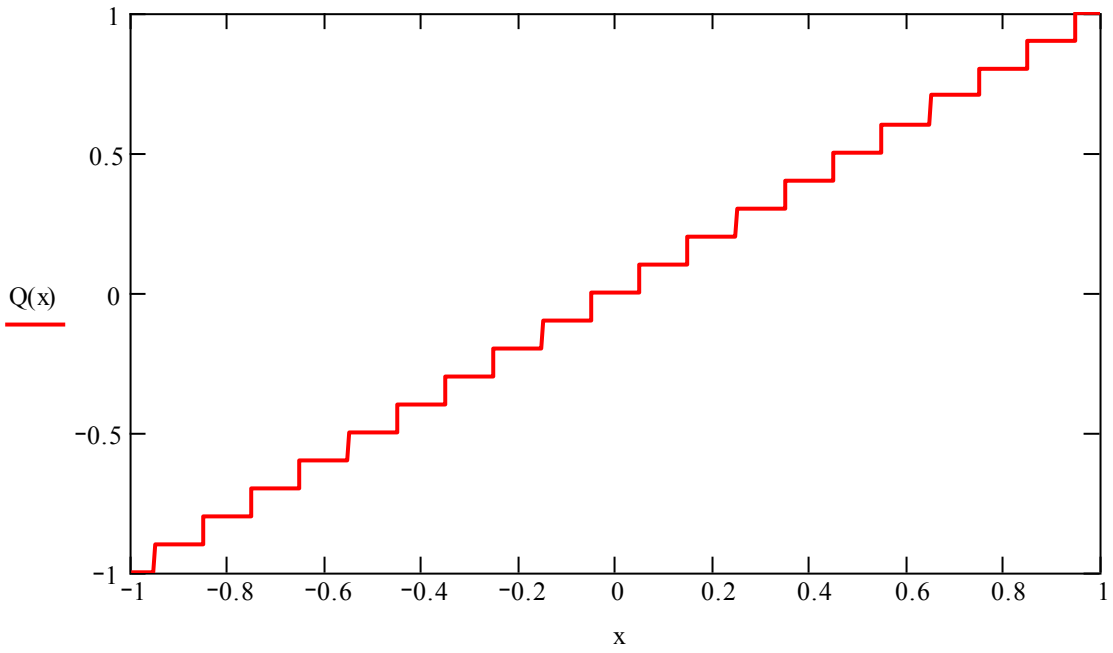


Figure 31.2: Uniform quantization.

$$\begin{aligned}\sigma_q^{\,2} &\approx \sum_k \frac{p_k}{\Delta}\int_{x_k}^{x_{k+1}}[x-y_k]^2\,dx \\ &= \sum_k \frac{p_k}{\Delta}\int_{-\Delta/2}^{\Delta/2}u^2\,du \\ &= \sum_k p_k\frac{\Delta^2}{12} \\ &= \frac{\Delta^2}{12}\end{aligned} \tag{31.5}$$

where p_k is the probability of being in the k^{th} interval, so the probability density there is approximately p_k/Δ. Clearly having a smaller step size leads to smaller quantization noise. But we also need to have enough steps to cover the full range of x. If $|x| \le x_{\max}$ and we have L steps, then $\Delta = \frac{2x_{\max}}{L}$. Plugging this into (31.5) we find $\sigma_q^{\,2} = \frac{1}{3}\left(\frac{x_{\max}}{L}\right)^2$. Therefore

$$\begin{aligned}S/N_{dB} &= 10\log\left[3L^2\left(\frac{\sigma_x}{x_{\max}}\right)^2\right] \\ &= 20\log L - 10\log\left[\frac{1}{3}\left(\frac{x_{\max}}{\sigma_x}\right)^2\right]\end{aligned} \tag{31.6}$$

If $L = 2^R$, R is the bit rate in terms of bits per symbol and

$$S/N_{dB} = 6R - 10\log\left[\frac{1}{3}\left(\frac{x_{\max}}{\sigma_x}\right)^2\right] \tag{31.7}$$

We gain 6 dB of *S/N* for each additional bit. For a uniform pdf

$$\begin{aligned}\sigma_x^{\,2} &= \int_{-x_{\max}}^{x_{\max}} x^2 \frac{1}{2x_{\max}}\,dx \\ &= \frac{x_{\max}^{\,2}}{3}\end{aligned} \tag{31.8}$$

and

$$S/N_{dB} = 6R \tag{31.9}$$

This is also referred to as the dynamic range. For example, with $R = 16$ bits, we get 96 dB, which represents CD quality, audio.

From (31.3) we see that more quantization error comes from those values of x where p_x is large. It would make sense, therefore, to have finer quantization in regions of large p_x and coarser quantization in regions of small p_x. This leads to *non-uniform quantization.* We first pass x through a *compressor* $c(x)$ and then perform uniform quantization. That is $y = Q(c(x))$. We choose $c(\pm x_{\max}) = \pm x_{\max}$ so that c covers the same range as x, but the slope can vary with x. If the slope dc/dx is large, then small changes in x produce large changes in c and hence cover relatively more quantization levels. At the receiver we *expand* using the invese of $c(x)$ to recoever x. The combination of compressor and expander is referred to as a *compander.* As an example, the so-called μ-law is

$$c(x) = \mathrm{sgn}(x)\,x_{\max}\frac{\ln\left(1+\mu|x|/x_{\max}\right)}{\ln(1+\mu)} \tag{31.10}$$

The resulting quantization is illustrated in Fig. 31.3 for $\mu = 255$. Near $x = 0$ where the pdf is largest, we can define the *compander gain* as

$$G_C = \left.\frac{dc(x)}{dx}\right|_{x\to 0} \tag{31.10}$$

because the effective step size is reduced to Δ/G_C. For the μ-law compander with $\mu = 255$, $G_C = 46$, and $20\log 46 = 33\,\mathrm{dB}$. This is better than the equivalent of 5 bits of improvement, so an 8-bit μ-law quantizer can give, on average, better performance than a 12-bit uniform quantizer. A phone-quality standard is an 8-bit μ-law quantizer at an 8 kHz sampling rate. This produces a 64 kbps data stream.

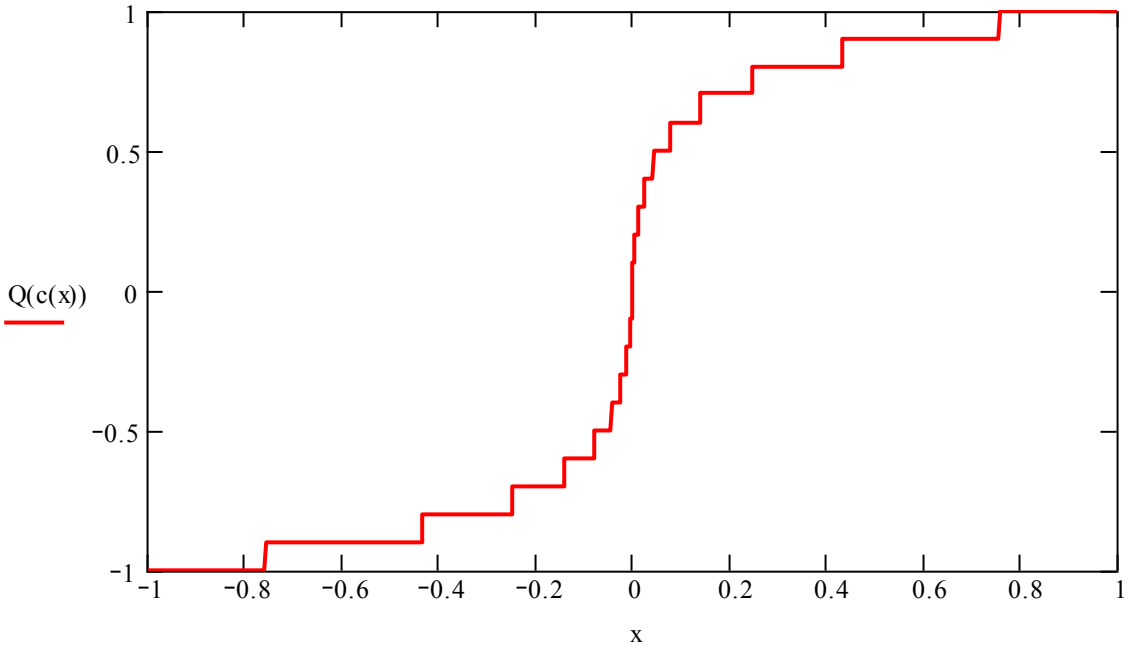


Figure 31.3: Non-uniform quantization using μ-law compander. Relatively more quantization levels are allocated to small values of x.

Predictive Coding

The idea of predictive coding is that if you can predict a signal, there is no need to have it transmitted to you. This is illustrated in Fig. 31.4

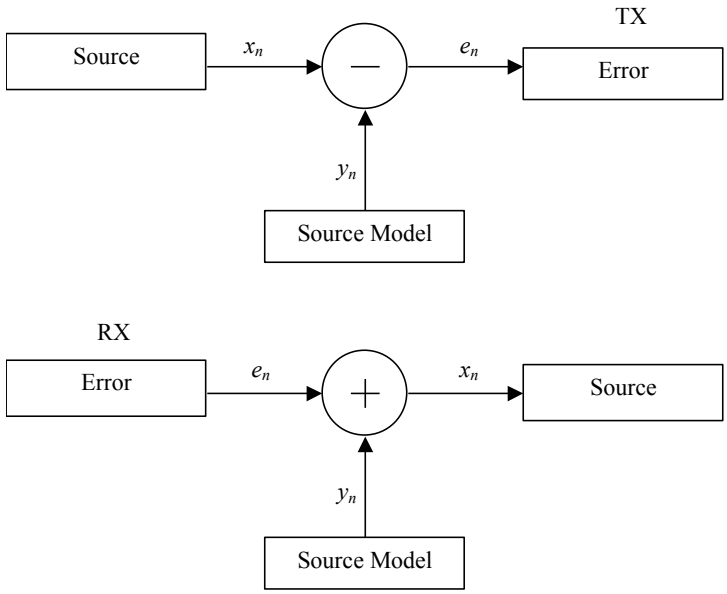


Figure 31.4: Predictive coding.

The source signal x_n is compared to a mathematical model of the source y_n. The difference is the error signal e_n. If transmitter and receiver share the same source model, then we need only transmit the error signal. If our model is a good one, then the error signal should require far fewer bits than the source itself. If the model were perfect then there would be no error at all. The difficultly here is coming up with good models for speech that are also mathematically tractable. Linear models are quite practical in this respect.

Linear Predictive Coding

A very simple source model is: "The source is a constant." Let x_n be the values produced by the source and y_n the modeled values. The "source is constant" model is then $y_n = x_{n-1}$, that is, the current value is equal to the last value. The error is $e_n = x_n - y_n$. More generally we might use $y_n = a_1 x_{n-1}$ where a_1 is some constant. This is a first-order linear predictor.

A second order linear predictor uses a straight line to model the source. If the last two source values were x_{n-1}, x_{n-2}, then the model is $y_n = x_{n-1} + (x_{n-1} - x_{n-2})$. More compactly we can write $y_n = 2x_{n-1} - x_{n-2}$. More generally we would have $y_n = a_1 x_{n-1} + a_2 x_{n-2}$.

An M^{th} order linear predicative coder predicts the current value of the waveform from a linear combination of the previous M values:

$$y_n = \sum_{k=1}^{M} a_k x_{n-k} \tag{31.12}$$

If the prediction is exact then $y_n = x_n$. Otherwise there is some error e_n and we write

$$x_n = y_n + e_n \tag{31.13}$$

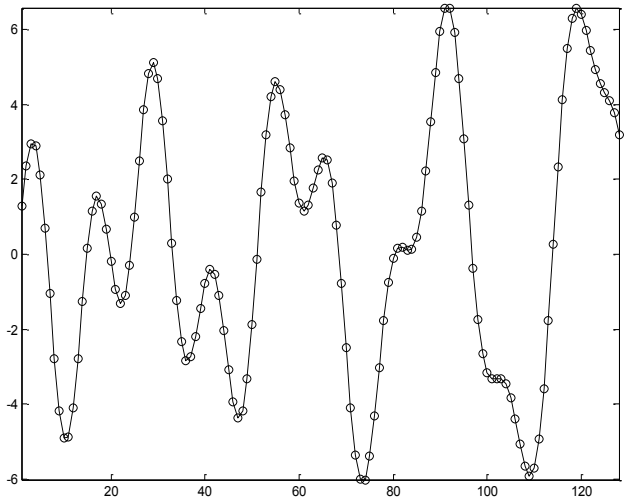
It turns out that a waveform consisting of K arbitrary, damped sinusoid components

$$x_n = \sum_{m=1}^{K} A_m e^{\alpha_m n} \cos(\omega_m n + \phi_m) \tag{31.14}$$

can be perfectly reconstructed, i.e., $e_n = 0$, by an LPC of order $M = 2K$, starting with the first M samples of the waveform.

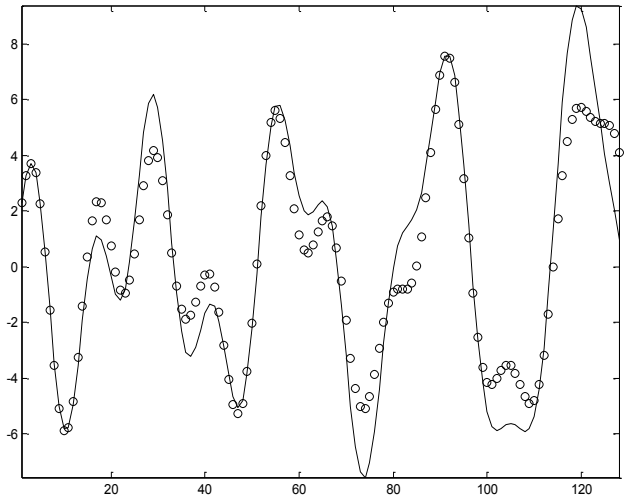
Example 31.1

Let $x_n = A_1 e^{\alpha_1 n} \cos(\omega_1 n + \phi_1) + A_2 e^{\alpha_2 n} \cos(\omega_2 n + \phi_2)$ with $A_1 = 3$, $\alpha_1 = -0.005$, $\omega_1 = 0.5$, $\phi_1 = -1.5$, $A_2 = 2$, $\alpha_2 = 0.01$, $\omega_2 = 0.2$, $\phi_1 = 1.0$. LPC coefficients for a 4th order predictor were calculated as $a_1 = 3.7262$, $a_2 = -5.4679$, $a_3 = 3.7418$, and $a_4 = -1.0101$. The following figure shows the original waveform and the LPC output.



The circles represent the original waveform and the line is the LPC output. We see that they agree perfectly, i.e., $e_n = 0$.

On the other hand, adding an additional sinusoidal term to x_n, $\cos(0.35n)$ in the example shown in the following figure, creates a waveform that can not be exactly described by a 4th order predictor..



In this case the error signal e_n is non-zero. Increasing the order to 6 would, however, allow perfect prediction.

What is the "secret" of an LPC? Let's recall the theory of linear ordinary differential equations with constant coefficients. A general M^{th} order equation has the form:

$$\frac{d^M}{dt^M}x(t)+b_1\frac{d^{M-1}}{dt^{M-1}}x(t)+\cdots+b_{M-1}\frac{d}{dt}x(t)+b_M=0 \tag{31.15}$$

The solutions are of the form e^{st}. The values of *s* are obtained from

$$s^M+b_1s^{M-1}+\cdots+b_{M-1}s+b_M=0 \tag{31.16}$$

There are *M* roots of this equation. The following is an M^{th} order "difference" equation:

$$x_n+b_1x_{n-1}+b_2x_{n-2}+\cdots+b_Mx_{n-M}=0 \tag{31.17}$$

We look for solutions of the form $x_n=z^n$. Plugging in we get

$$z^n+b_1z^{n-1}+b_2z^{n-2}+\cdots+b_Mz^{n-M}=0 \tag{31.18}$$

or

$$z^M+b_1z^{M-1}+b_2z^{M-2}+\cdots+b_{M-1}z+b_M=0 \tag{31.19}$$

This is an equation of the same form as (31.6), and there will be *M* roots. If we identify $t=n\Delta t$ and $z=e^{s\Delta t}$, then $z^n=e^{sn\Delta t}=e^{st}$. That is, the solutions of the M^{th} order difference equation are just sampled versions of the solutions of the M^{th} order differential equation.

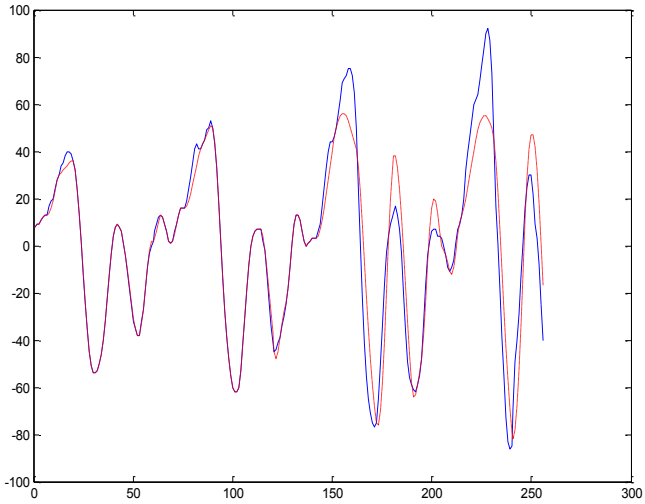
(31.17) can be rearranged to give

$$x_n=\sum_{k=1}^{M}a_kx_{n-k} \tag{31.20}$$

where $a_n=-b_n$.

Example 31.2

Here we have 256 samples of speech coded at 8 bits per sample. An 8th order linear predictor is used. The error signal is coded with 2 bits per sample.



In this way we achieve nearly 4x's compression with reasonable signal quality.

An M^{th} order LPC will perfectly predict the signal x_n if it contains precisely $M/2$ sinusoidal components.

$$e_n = x_n - \sum_{k=1}^{M} a_k x_{n-k} \tag{31.21}$$

$$\left\langle e_n^{\,2} \right\rangle = \left\langle x_n^{\,2} \right\rangle - 2\sum_{k=1}^{M} a_k \left\langle x_{n-k} x_n \right\rangle + \sum_{j=1}^{M}\sum_{k=1}^{M} a_j a_k \left\langle x_{n-j} x_{n-k} \right\rangle \tag{31.22}$$

Setting $\partial\left\langle e_n^{\,2} \right\rangle / \partial a_k = 0$ for $k = 1 \ldots M$ we get the equations that define the optimal predictor coefficients. With the definition $R_{xx}(k) = \left\langle x_n x_{n-k} \right\rangle$ these take the form

$$\mathbf{Ra} = \mathbf{r} \tag{31.23}$$

where the vector **a** has elements a_k and the elements of the matrix **R** and vector **r** are

$$\begin{aligned} R_{jk} &= R_{xx}(k-j) \\ r_k &= R_{xx}(k) \end{aligned} \tag{31.24}$$

The solution is

$$\mathbf{a} = \mathbf{R}^{-1}\mathbf{r} \tag{31.25}$$

CELP

Code Excited Linear Predictors dispense with sending the error signal e_n. Instead they have a library of "codes," (i.e., "typical" error signals). Each code is put through the linear predictor and it is noted how well that matches the actual voice signal. Then the address of the best-match code is sent along with the predictor coefficients.

References

1. Jayant, N. S. and P. Noll, *Digital Coding of Waveforms*, Prentice-Hall, 1984, ISBN 0-13-211913-7.