First steps in digital signal processing

John Coleman

Phonetics Laboratory
University of Oxford
Analogue-to-digital conversion 1: Sampling
Analogue-to-digital conversion 2: Quantization
D-to-A, quantization error

Original signal level = 3.09 V

Reconstruction levels

Playback signal level = 3.05 V.

Error = 0.04 V.

Quantization decision levels
D-to-A, quantization error

- To reduce the quantization error, use more levels (dynamic range):
  - 8 bits: $2^8 = 256$ levels
  - 12 bits: $2^{12} = 4096$ levels
  - 16 bits: $2^{16} = 65536$ levels
Pulse Code Modulation
(Alec Reeves 1937)
Sampling theorem; Nyquist frequency

- Sampling rate must be at least twice the highest frequency you want to capture.
Operations on sequences of numbers

- Let's call the sample number $i$ and the $i$'th sample $x[i]$.
- **Sum** or **integral**, $\Sigma x[i]$.
- If $x[i]$ has positive and negative values, take $|x[i]|$ the absolute (i.e. unsigned) value of $x[i]$.
- Or, first calculate the square of $x[i]$, $x[i]^2$, and then take the square root, $\sqrt{x[i]^2}$. $\Sigma\sqrt{x[i]^2}$ is a measure of the overall energy of a signal.
- $x[i]^2$ gets bigger and bigger as $x[i]$ gets longer.
- The *average* amplitude of a signal, calculated over $n$ samples: $\sqrt{\Sigma x[i]^2/n}$. This is called the *root mean square* or RMS amplitude.
Local (moving) average

e.g., \( y[n] = \frac{1}{4} x[n] + \frac{1}{4} x[n-1] + \frac{1}{4} x[n-2] + \frac{1}{4} x[n-3] \)
Local (moving) average

Effect: low-pass filtering
Time-domain filtering

4 samples is very short, so its effects are very local – high frequency components. To smooth over a larger slice of the signal, we can do two things:

a) increase the number of samples in $x[n] \ldots x[n-m]$

b) make $y[n]$ depend in part on its own previous value, $y[n-1]$, or several previous values.
Time-domain filtering

- A filter of the second kind has the general form:

\[ y[n] = b_0 x[n] + b_1 x[n-1] + b_2 x[n-1] + \ldots + b_k x[n-k] \]
\[ - a_1 y[n-1] + \ldots + a_j y[n-j] \]

- By varying the a's, b's, and the number and spacing of previous x and y samples, a variety of filters with various kinds of frequency-selecting behaviours can be constructed.
Linear Prediction (preview)

\[ y[n] = b_0 x[n] + b_1 x[n-1] + b_2 x[n-1] + \ldots + b_k x[n-k] \]
\[ - a_1 y[n-1] + \ldots + a_j y[n-j] \]

- We can estimate the magnitude of the current sample as a linear combination of the previous \( p \) samples (typically 12 to 18 samples):

\[ x[t] = -a_1 x[t-1] - a_2 x[t-2] - a_3 x[t-3] \ldots - a_p x[t-p] + e[t] \]
Fourier (Spectral) Analysis

Jean Baptiste Joseph Fourier (1768-1830)
Windowing
Fast Fourier Transform

- Cooley and Tukey, mid 1960's
- e.g. for Power Spectrum
Fast Fourier Transform

- Cooley and Tukey, mid 1960's
- e.g. for Power Spectrum
Cepstrum (Noll 1967)
Cepstrum
Linear Prediction of speech

- We estimate the current sample as a linear combination of the previous $p$ samples:
- $x[t] = -a_1 x[t-1] - a_2 x[t-2] - \ldots - a_p x[t-p] + e[t]$
Linear Prediction of speech

- Original

- Error
Linear Prediction of speech

- Original
- Resynthesized excluding error
A magic trick

- 10 LPC coefficients from an $\alpha$

- Same 10 coeffs, scaled up and zero-padded to 512 samples long (251 each side)
A magic trick

- Same 10 coeffs, scaled up and zero-padded to 512 samples long
A magic trick

FFT

LP spectrum

F1  F2  F3  F4
Shameless plug