EE123
Digital Signal Processing

Lecture 19
Practical ADC/DAC
Ideal Anti-Aliasing

\[ x_c(t) \rightarrow \text{Analog Anti-Aliasing Filter } H_{LP}(j\Omega) \rightarrow \text{sampler } x[n] = x_c(nT) \rightarrow \text{Quantizer} \]

**Analog Anti-Aliasing Filter**

\[ X_c(j\Omega) \]

and \( \Omega_s < 2\Omega_N \)

\[ X_s(j\Omega) \]

\[ X_c(j\Omega)H_{LP}(j\Omega) \]

and \( \Omega_s < 2\Omega_N \)

\[ X_s(j\Omega) \]
Non Ideal Anti-Aliasing

\[ X_c(j\Omega)H_{LP}(j\Omega) \]

• Problem: Hard to implement sharp analog filter
• Tradeoff:
  – Crop part of the signal
  – Suffer from noise and interference (See lab II !)
Oversampled ADC

\[ x_c(t) \overset{\text{Sharp Analog Anti-Aliasing Filter } H_{LP}(j\Omega)}{\longrightarrow} \overset{T}{\longrightarrow} x[n] = x_c(nT) \rightarrow \text{Quantizer} \]

\[ x_c(t) \overset{\text{Simple Analog Anti-Aliasing Filter}}{\longrightarrow} \overset{\text{C/D}}{\longrightarrow} T = \frac{1}{M} \left( \frac{\pi}{\Omega_N} \right) \overset{\text{Sharp Digital Anti-aliasing filter } \frac{\pi}{M}}{\longrightarrow} \downarrow M \rightarrow \text{Quantizer} \]
Oversampled ADC

\[ X_c(j\Omega)H_{LP}(j\Omega) \]
Oversampled ADC

\[ X_c(j\Omega)H_{LP}(j\Omega) \]

after oversampling x2

\[ \hat{X}(e^{j\omega}) \]

aliased noise
Oversampled ADC

after oversampling x2

\[ \hat{X}(e^{j\omega}) \]

\[ \hat{X}_d(e^{j\omega}) \quad T_d = M T \]

after digital LP and decimation

aliased noise

\(-\pi\) \quad \pi\]
Sampling and Quantization

\[ x_c(t) \]

\[ \frac{C/D}{T} \]

\[ x[n] = x_c(nT) \]

Quantizer

\[ \hat{x}[n] \]

ADC A/D

\[ \Delta \]

\[ 2X_m \]
Sampling and Quantization

• for 2’s complement with B+1 bits

\[-1 \leq \hat{x}_B[n] < 1\]

\[\Delta = \frac{2X_m}{2^{B+1}} = \frac{X_m}{2^B}\]

\[\hat{x}[n] = X_m \hat{x}_B[n]\]
Quantization Error

Model quantization error as noise

\[ \hat{x}[n] = x[n] + e[n] \]

In that case:

\[ -\Delta/2 \leq e[n] < \Delta/2 \]

\[ (-X_m - \Delta/2) < x[n] \leq (X_m - \Delta/2) \]
Noise Model for Quantization Error

• Assumptions:
  – Model $e[n]$ as a sample sequence of a stationary random process
  – $e[n]$ is not correlated with $x[n]$, e.g., $\mathbb{E} e[n] x[n] = 0$
  – $e[n]$ not correlated with $e[m]$, e.g., $\mathbb{E} e[n] x[m] = 0 \mid m \neq n$ (white noise)
  – $e[n] \sim U[-\Delta/2, \Delta/2]$

• Result:
  – Variance is: $\sigma_e^2 = \frac{\Delta^2}{12}$, or $\sigma_e^2 = \frac{2^{-2B} X_m^2}{12}$ since $\Delta = 2^{-B} X_m$
  – Assumptions work well for signals that change rapidly, are not clipped and for small $\Delta$
Quantization Noise

Figure 4.57 (continued) (b) Quantized samples of the cosine waveform in part (a) with a 3-bit quantizer. (c) Quantization error sequence for 3-bit quantization of the signal in (a). (d) Quantization error sequence for 8-bit quantization of the signal in (a).
For uniform B+1 bits quantizer:

\[ \text{SNR} \quad Q \quad = \quad 6.02B + 10.8 - 20 \log_{10} \left( \frac{X_m}{\sigma_x} \right) \]

\[ \sigma_e^2 = \frac{2^{-2B} X_m^2}{12} \]

\[ SNR_Q = 10 \log_{10} \left( \frac{\sigma_x^2}{\sigma_e^2} \right) = 10 \log_{10} \left( \frac{12 \cdot 2^{2B} \sigma_x^2}{X_m^2} \right) \]
SNR of Quantization Noise

\[ \text{SNR}_Q = 6.02B + 10.8 - 20 \log_{10} \left( \frac{X_m}{\sigma_x} \right) \]

- Improvement of 6dB with every bit
- The range of the quantization must be adapted to the rms amplitude of the signal
  - Tradeoff between clipping and noise!
  - Often use pre-amp
  - Sometimes use analog auto gain controller (AGC)
- If \( \sigma_x = \frac{X_m}{4} \) then \( \text{SNR}_Q \approx 6B - 1.25\text{dB} \)
  so SNR of 90-96 dB requires 16-bits (audio)
Quantization noise in Oversampled ADC

\[
\begin{align*}
    x_c(t) & \xrightarrow{\text{C/D}} x[n] & \hat{x}[n] = x[n] + e[n] & \xrightarrow{\text{LPF}} \hat{x}_d[n] = x_d[n] + e_d[n] \\
    T &= \frac{\pi}{\Omega NM} \\
    X_c(j\Omega) & \rightarrow \hat{X}(e^{j\omega}) & \hat{X}_d(e^{j\omega})
\end{align*}
\]
Quantization noise in Oversampled ADC

- Energy of $x_d[n]$ equals energy of $x[n]$
  - No filtering of signal!
- Noise std is reduced by factor of $M$

$$SNR_Q = 6.02B + 10.8 - 20\log_{10}\left(\frac{X_m}{\sigma_x}\right) + 10\log_{10} M$$

- For doubling of $M$ we get 3dB improvement, which is the same as 1/2 a bit of accuracy
  - With oversampling of 16 with 8bit ADC we get the same quantization noise as 10bit ADC!

M. Lustig, EECS UC Berkeley
Practical ADC (Ch. 4.8.4)

- Scaled train of sinc pulses
- Difficult to generate sinc $\Rightarrow$ Too long!
Practical ADC

D.T

\[ x[n] = x(t) \big|_{t=nT} \]

Interp. Filter

\[ h_0(t) \Leftrightarrow H_0(j\Omega) \]

Recon. Filter

\[ h_r(t) \Leftrightarrow H_r(j\Omega) \]

\[ x_r(t) \]

\[ h_0(t) \text{ is finite length pulse} \Rightarrow \text{easy to implement} \]

\[ = \sum x[n]h_0(t-nT) \]

C.T analog processing

- For example: zero-order hold

\[ H_0(j\Omega) = Te^{-j\Omega \frac{T}{2}} \text{sinc} \left( \frac{\Omega}{\Omega_s} \right) \]
Practical ADC

Zero-Order-Hold interpolation

\[ x_0(t) = \sum_{n=-\infty}^{\infty} x[n] h_0(t - nT) = x_0(t) * x_s(t) \]

Taking a FT:

\[ X(j\Omega) = H_0(j\Omega) X_s(j\Omega) \]

\[ = H_0(j\Omega) \frac{1}{T} \sum_{k=-\infty}^{\infty} X(j(\Omega - k\Omega_s)) \]
Practical ADC

Output of the reconstruction filter:

\[ X_r(j\Omega) = H_r(j\Omega) \cdot H_0(j\Omega) \cdot X_s(j\Omega) \]

\[ = H_r(j\Omega) \cdot T e^{-j\Omega \frac{T}{2}} \text{sinc}\left(\frac{\Omega}{\Omega_s}\right) \cdot \frac{1}{T} \sum_{k=-\infty}^{\infty} X\left(j(\Omega - k\Omega_s)\right) \]

\[
\begin{align*}
H_r(j\Omega) & \quad \text{recon filter} \\
H_0(j\Omega) & \quad \text{from zero-order hold} \\
X_s(j\Omega) & \quad \text{Shifted copies from sampling} \\
\end{align*}
\]
Practical ADC

Ideally:

\[ X_s(j\Omega) H_{LP}(j\Omega) \]
Practical ADC

\[ X_s(j\Omega) \]

... ... ...

Practically:

\[ X_s(j\Omega)H_0(j\Omega) \]

... ... ...
Practical ADC

$$X_s(j\Omega)$$

... 

Practically:

$$X_s(j\Omega)H_0(j\Omega)$$

... 

= * 

M. Lustig, EECS UC Berkeley
Practical ADC

$$X_s(j\Omega)$$

... ... ...

Practically:

$$X_s(j\Omega) H_0(j\Omega) H_r(j\Omega)$$

... ... ...
Easier Implementation with Digital upsampling

\[ x[n] \rightarrow \uparrow L \rightarrow x_e[n] \rightarrow \text{LPF} \]

Practically:

\[ X_s(j\Omega) H_0(j\Omega) H_r(j\Omega) \]

\[ x[n] \rightarrow \uparrow L \rightarrow x_e[n] \rightarrow \text{LPF} \text{ gain}=L \]

\[ \pi / L \]

\[ \cdots \]

\[ \cdots \]
Easier Implementation with Digital upsampling

- Easier implementing with analog components
- Need analog components made of Nonobtainium