EE123
Digital Signal Processing

Lecture 19
Practical ADC/DAC
Ideal Anti-Aliasing

$x_c(t)$

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

ADC A/D

sampler \[ t = nT \]

Quantizer

$X_c(j\Omega)$

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

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

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

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

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

after oversampling x2

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

M. Lustig, EECS UC Berkeley
Oversampled ADC

after oversampling x2

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

\[ \frac{1}{T} \]

aliased noise

after digital LP and decimation

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

\[ T_d = MT \]

M. Lustig, EECS UC Berkeley
Sampling and Quantization

\[ x_c(t) \]

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

ADC A/D

Quantizer

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

\[ 2X_m \]

\[ \Delta \]

M. Lustig, EECS UC Berkeley
Sampling and Quantization

- for 2’s complement with B+1 bits \(-1 \leq \hat{x}_B[n] < 1\)

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

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

- Model quantization error as noise

\[ x[n] \rightarrow \text{Quantizer} \rightarrow \hat{x}[n] \]

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

- 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: $\sigma^2_e = \frac{\Delta^2}{12}$, or $\sigma^2_e = \frac{2^{-2B} X_m^2}{12}$ 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).
SNR of Quantization Noise

- For uniform B+1 bits quantizer: $$\sigma_e^2 = \frac{2^{-2B} X_m^2}{12}$$

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

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

Quantizer range

rms of amp

M. Lustig, EECS UC Berkeley
SNR of Quantization Noise

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

Quantizer range

rms of amp

• 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 \( \sigma_x = X_m/4 \) then \( \text{SNR}_Q \approx 6B - 1.25dB \)
    so SNR of 90-96 dB requires 16-bits (audio)
Quantization noise in Oversampled ADC

\[ x_c(t) \xrightarrow{\text{C/D}} x[n] \xrightarrow{\pi} e[n] \xrightarrow{\text{LPF}} \hat{x}[n] = x[n] + e[n] \xrightarrow{\downarrow M} \hat{x}_d[n] = x_d[n] + e_d[n] \]

\[ T = \frac{\pi}{\Omega_NM} \]

\[ X_c(j\Omega) \xrightarrow{\text{C/D}} \hat{X}(e^{j\omega}) \]

\[ \hat{X}_d(e^{j\omega}) \]
Quantization noise in Oversampled ADC

• Energy of $x_d[n]$ equals energy of $x[n]$
  – No filtering of signal!

• Noise var is reduced by factor of $M$

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

• 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!
x[n] = x(t)|_{t=nT} \rightarrow \text{sinc pulse generator} \rightarrow x_r(t) = \sum_{n=-\infty}^{\infty} x[n] \text{sinc} \left( \frac{t - nT}{T} \right)

- Scaled train of sinc pulses
- Difficult to generate sinc ⇒ 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
\[ h_r(t) \Leftrightarrow H_r(j\Omega) \]

C.T analog processing

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

- \( h_0(t) \) is finite length pulse \( \Rightarrow \) easy to implement

- 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

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

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:

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

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

\( H_r(j\Omega) \) recon filter

\( H_0(j\Omega) \) from zero-order hold

\( X_s(j\Omega) \) Shifted copies from sampling

\( X(kT) \)

M. Lustig, EECS UC Berkeley
Practical ADC

Ideally:

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

... ...

M. Lustig, EECS UC Berkeley
Practical ADC

\[ X_s(j\Omega) \]

... 

\hline

Practically:

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

...
Practical ADC

\[ X_s(j\Omega) \]

... ... ...

Practically:

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

... ... ...

= *
Practical ADC

\[ X_s(j\Omega) \]

... ... ... ...

Practically:

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

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

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

Practically:

\[ X_s(j\Omega) H_0(j\Omega) H_r(j\Omega) \]
Easier Implementation with Digital upsampling

easier implementing with analog components

Need analog components made of low-loss unobtainium transistors