Lab 4: Quantization, Oversampling, and Noise Shaping

Due Friday 04/21/17

Overview:

- This assignment should be completed with your assigned lab partner(s).
- Each group must turn in a report composed using a word processor (e.g., Word, Pages, LaTeX, etc...) including a cover page with full names of all group members. The remaining pages should contain (in order) the answers and MATLAB scripts for the exercises. MATLAB figures can be pasted into the document or saved as PDF files. When working on the project, please follow the instructions and respond to each item listed. Your project grade is based on: (1) your MATLAB scripts, (2) your report (plots, explanations, etc. as required), and (3) your final results. For all labs, you must clearly write the problem number next to your solution and label the axes on all plots to get full credit. Submission can be done electronically in PDF format or on paper.
- Plagiarism is a very serious offense in Academia. Any figures in the paper not generated by you should labeled "Reproduced from [...]". Any portions of any simulation code (e.g., MATLAB, C, etc...) not written by you be clearly marked in your source files. The original source of any mathematical derivation or proof should be explicitly cited.

1 Overview

In this lab, we will explore quantization, oversampling, and noise shaping in analog-to-digital (A/D), digital-to-analog (D/A), and digitial-to-digital (D/D) conversion. Amplitude quantization is a fundamental process in the conversion from continuous to discrete amplitude. This topic is quite interesting because there are a number of non-obvious "tricks" that can be used to improve performance. The first trick is the addition of random dither before quantization. Surprisingly, adding noise before quantization helps one reduce the negative effects of quantization noise. The second trick is sampling much faster than the Nyquist rate. The idea is that low-accuracy samples taken at a high rate can be used to get high-accuracy samples at a low rate. The last trick is noise shaping with sigma-delta converters. By cleverly combining A/D and D/A converters together, one can drastically improve the benefit of oversampling.

2 Background

To understand the effects of quantization, we first have to observe and understand the spectral behavior of white noise and colored noise. The power of a white-noise process is spread uniformly across the full spectrum $[-\pi, \pi]$, whereas the power of a colored noise process is spread non-uniformly across the spectrum. Let $X[n]$ be a stationary discrete-time white-noise process whose sample at time index n is denoted by the random variable $X[n]$. For example, one can generate a length-L sample from such a random process using the Matlab command `x=randn(1,L)`. The autocorrelation function of this random process is defined to be

$$R_{xx}[\tau] \triangleq \mathbb{E}\Big[X[n]X[n+\tau]\Big] \; ; \; \tau \in \{\ldots, -1, 0, 1, \ldots\}$$

and the power spectral density is related to it as $S_{xx}(e^{j\omega}) = \mathcal{F}\left(R_{xx}[\tau]\right)$, where $\mathcal{F}$ represents the discrete-time Fourier transform operator. Note that, since this process is *stationary*, $R_{xx}[\tau]$ does not depend on n but only on the time difference between the samples, τ.

Now let us consider the particular example of a white noise process generated from the standard normal distribution $\mathcal{N}(0,1)$. So each $X[n]$ is *independent* and distributed according to $X[n] \sim \mathcal{N}(0,1)$. From the definition of this distribution we know that if $Z \sim \mathcal{N}(0,1)$ then its mean is $\mathbb{E}[Z] = 0$ and its variance is $\mathbb{E}[Z^2] = 1$. Therefore, for our Gaussian white noise process, we have $\mathbb{E}\big[X[n]\big] = 0$ and

$\mathbb{E}\big[X[n]^2\big] = 1$. Let us now compute the autocorrelation function and power spectral density of this process. For $\tau = 0$, we get $R_{xx}[0] = \mathbb{E}\big[X[n]^2\big] = 1$. For $\tau > 0$, observe that $X[n]$ and $X[n+\tau]$ are two *independent* standard normal random variables. Since they are independent, the expectation of their product is the product of their expectations:

$$R_{xx}[\tau] = \mathbb{E}\Big[X[n]X[n+\tau]\Big] = \mathbb{E}\Big[X[n]\Big] \cdot \mathbb{E}\Big[X[n+\tau]\Big] = 0 \cdot 0 = 0.$$

Hence we have

$$R_{xx}[\tau] = \begin{cases} 1 & ,\tau = 0 \\ 0 & ,\tau \neq 0 \end{cases},$$

which is a discrete-time impulse function. This means the noise process is completely uncorrelated with itself for any non-zero lag. Computing the DTFT, we see that the power spectral density is given by $S_{xx}(e^{j\omega}) = 1$ and hence the process is "white".

Next, let us run this process $X[n]$ through a low-pass filter whose impulse response is given by $h[n] = 0.5\delta[n] + 0.5\delta[n-1]$. Hence the output process is given by $Y[n] = 0.5X[n] + 0.5X[n-1]$. Let us proceed to compute the autocorrelation function of the process $Y[n]$. We have

$$\begin{aligned}
R_{yy}[\tau] &\triangleq \mathbb{E}\Big[Y[n]Y[n+\tau]\Big] \\
&= \mathbb{E}\Big[\left(0.5X[n] + 0.5X[n-1]\right)\left(0.5X[n+\tau] + 0.5X[n+\tau-1]\right)\Big] \\
&= 0.25\Big[\mathbb{E}\big[X[n]X[n+\tau]\big] + \mathbb{E}\big[X[n]X[n+\tau-1]\big] + \mathbb{E}\big[X[n-1]X[n+\tau]\big] + \mathbb{E}\big[X[n-1]X[n+\tau-1]\big]\Big] \\
&= \begin{cases} 0.25\big[1+0+0+1\big] & \text{if } \tau = 0 \\ 0.25\big[0+1+0+0\big] & \text{if } \tau = 1 \\ 0.25\big[0+0+1+0\big] & \text{if } \tau = -1 \\ 0.25\big[0+0+0+0\big] & \text{if } |\tau| > 1 \end{cases} \\
&= \begin{cases} 0.5 & \text{if } \tau = 0 \\ 0.25 & \text{if } \tau = 1 \\ 0.25 & \text{if } \tau = -1 \\ 0 & \text{if } \tau > 1. \end{cases}
\end{aligned}$$

Therefore the power spectral density is

$$\begin{aligned}
S_{yy}(e^{j\omega}) &= \sum_{\tau=-\infty}^{\infty} R_{yy}[\tau]e^{-j\omega\tau} \\
&= 0.5 \cdot e^{-j\omega(0)} + (0.25) \cdot \left(e^{-j\omega(1)} + e^{+j\omega(1)}\right) \\
&= 0.5(1 + \cos(\omega)),
\end{aligned}$$

which equals the magnitude squared $\left|H(e^{j\omega})\right|^2$ of the low-pass filter $h[n]$ and is clearly not uniform on $[-\pi, \pi]$. Since this spectrum has more power at low frequencies than high frequencies, the noise process $Y[n]$ is called "colored" noise.

3 Exercises

3.1 Random Noise, Autocorrelation and Power Spectral Density

Let us complement our theoretical understanding of the spectral behavior of white noise and colored noise processes with MATLAB experiments.

(a) Let `x` be 2 seconds of white noise sampled at `Fs=22050` Hz. This can be generated using the `randn` in-built function in MATLAB. For example, if we need `L` samples of the process, we can obtain it using `x=randn(1,L)`. This generates `L` *independent* samples from the standard normal distribution $\mathcal{N}(0,1)$. Plot the deterministic autocorrelation $r_{xx}[\tau]$ and power spectral density

$R_{xx}(e^{j\omega})$ functions for the vector `x` using the `autocorr` and `pwelch` in-built functions, respectively, using their default options. Comment on the two plots in connection with the background theory. Play this signal using the `sound` function and briefly describe what you hear.

(b) Now, filter the signal `x` using the low-pass filter with coefficients `b=[0.5 0.5],a=[1]` to get a signal `ylp` and plot its autocorrelation and power spectral density using the same in-built functions as in the previous part. Comment on the two plots, in comparison to those for `x`. Play `ylp` using the `sound` function and briefly describe what you hear, compared to what you heard when you played `x`. Note that the `autocorr` function normalizes the plot so that $r_{xx}[0] = 1$.

(c) Repeat the previous part using the high-pass filter with coefficients `b=[0.5 -0.5],a=[1]` to get a signal `yhp`. Can this also be called colored noise? How does it sound?

The following are some additional comments/suggestions and you need not include these in your lab submissions. You can compare the `pwelch` plots with their corresponding `periodogram` plots with which you are more familiar now. The frequency response of the filters can be visualized using `freqz` and/or `fvtool` functions, or you could obtain the impulse response using the `filter` function and then observe its `pwelch` or `periodogram` plot. Realize that all of these are different ways to observe the spectrum, although `pwelch` is more sophisticated than `periodogram`. As mentioned in the MATLAB Help documentation for `pwelch`, for a wide-sense stationary process, like we have above, the `periodogram` is not a consistent estimator of the true power spectral density whereas `pwelch` is, since it averages the estimate over multiple windowed segments to reduce the variability in its answer.

3.2 Basic Quantization

For this exercise, we will quantize a sinusoid at different resolutions with and without dither.

(a) Let `x` be 2 seconds of a 600 Hz unit amplitude sinusoid sampled at `Fs=22050` Hz. Let `L` be the length of this sequence.

(b) For $\Delta = 2^{-m}$ with $m = 1, 3, 5, 7$, quantize `x` with

$$Q(x) = \left\lfloor \frac{x}{\Delta} + \frac{1}{2} \right\rfloor \Delta.$$

For $m = 1$, how many quantization levels do we have for the given signal? Hence, how many bits of quantization does this correspond to, ignoring boundary effects (at the maximum and minimum values for the signal)? Measure the SQNR in dB and compare your result (for the noise power) with the theoretical prediction from class. For each m, listen to both the quantized waveform and the quantization error signal. Describe the artifacts that you hear in each case. Plot the power spectral density of the quantization error using `pwelch` and explain the association between this plot and the audio artifacts.

Note: For this lab, always use the `sound` function. Avoid using the `soundsc` function since it scales the whole signal to make it as loud as possible (without clipping). So, for higher levels of quantization, this will play the (quantization) error signal loudly although it has very low energy in the full spectrum.

(c) Repeat the previous part using uniform dither over one quantization cell. In this case,

$$Q(x) = \left\lfloor \frac{x}{\Delta} + \frac{1}{2} + Z \right\rfloor \Delta,$$

where Z is uniform over $[-\frac{1}{2}, \frac{1}{2}]$ and drawn independently each time the quantizer is used. *Note*: You can generate such a sequence with `z=rand(1,L)-1/2`. Measure the SQNR in dB and compare with the theoretical prediction. For each m, listen to both the quantized waveform and the quantization error signal. Describe the artifacts that you hear in each case.

Note: The addition of Z before quantization adds $\Delta^2 \mathrm{Var}(Z) = \frac{\Delta^2}{12}$ to the theoretical quantization noise power but guarantees that the quantization error is uncorrelated with the signal.

(d) Use `[x,Fs] = audioread('singing8.wav')` to load a sound clip. Let `L=2*Fs` and `x=x(1:L)'` be the first 2 seconds of the clip. For $\Delta = 2^{-m}$ with $m = 1, 3, 5, 7$, quantize `x` without dither. Compute the actual SQNR and the theoretical prediction. For each m, listen to both the quantized waveform and the quantization error signal. Describe the artifacts that you hear in each case.

(e) For the signal in the previous part, use $\Delta = 2^{-m}$ with $m = 1, 3, 5, 7$ to quantize `x` with uniform dither over one quantization cell. Compute the actual SQNR and the theoretical prediction. For each m, listen to both the quantized waveform and the quantization error signal. Describe the artifacts that you hear in each case.

(f) For the signal in the previous part, use $\Delta = 2^{-m}$ with $m = 1, 3, 5, 7$ to quantize `x` with triangular dither:

$$Q(x) = \left\lfloor \frac{x}{\Delta} + \frac{1}{2} + Z_1 + Z_2 \right\rfloor \Delta,$$

where `z1=rand(1,L)-1/2` and `z2=rand(1,L)-1/2`. Compute the actual SQNR and the theoretical prediction. For each m, listen to both the quantized waveform and the quantization error signal. Describe the artifacts that you hear in each case. Can you explain why triangular dither is preferred in high-quality audio applications?

Note: The addition of $Z_1 + Z_2$ before quantization adds $\Delta^2 \mathrm{Var}(Z_1 + Z_2) = \frac{\Delta^2}{6}$ to the theoretical quantization noise power.

3.3 Oversampled Quantization

To evaluate an oversampled quantization system, we must have an oversampled waveform. The following MATLAB code computes this waveform.

```
% Load sound clip
[x,Fs] = audioread('singing8.wav');

% Band-limited oversampling of x by factor M
M = 6;                                % Oversampling factor
xo = zeros(M*length(x),1);            % Init oversampled waveform
xo(1:M:end) = x;                      % Upsample waveform
h = fir1(300,0.9/M,chebwin(301,80)); % Design interpolation filter
xo = M*filter(h,1,xo);                % Filter upsampled waveform keeping same amplitude
```

(a) For $\Delta = 2^{-m}$ with $m = 1, 3, 5, 7$, quantize the oversampled signal `xo` with

$$Q(x) = \left\lfloor \frac{x}{\Delta} + \frac{1}{2} + Z \right\rfloor \Delta,$$

where Z is uniform over $[-\frac{1}{2}, \frac{1}{2}]$ and drawn independently each time the quantizer is used.

(b) Measure the SQNR in dB and compare your result with the theoretical prediction from class. For each m, listen to both the quantized waveform and the quantization error signal. Use `sound(xo,M*Fs)` to play the clip at the oversampled rate.

(c) For $m = 1, 3, 5, 7$, apply the following to each quantized signal: (i) Filter the quantized oversampled signal `xoq` with `xoqf=filter(h,1,xoq)` to remove out of band quantization noise and (ii) measure the SQNR in dB by comparing `xoqf` with the filtered oversampled signal `xof=filter(h,1,xo)`. Compare your result with the SQNR gain predicted by theory.

Note: The reason that one should filter `xo` before comparing with `xoqf` is because the true noise power is represented by the power of the filtered error signal `ef=filter(h,1,xo-xoq)`. Additionally, this approach allows one to ignore any delay and/or scaling changes introduced by filtering.

3.4 Oversampled Quantization with Noise Shaping

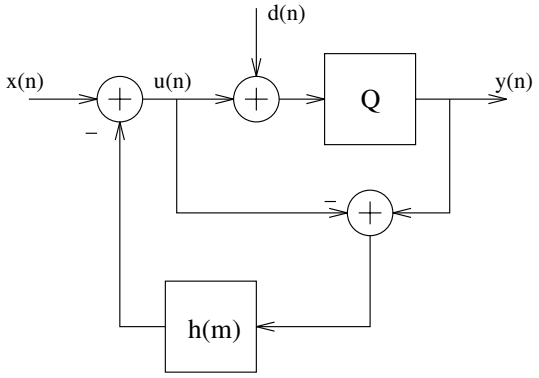
```
% Quantization of 'xo' with noise shaping
yo = zeros(1,length(xo));
y = 0;
v = 0;
for i=1:length(xo)
  w = xo(i)-y;                % Subtract previous quantized output
  v = v+w;                    % Accumulate all inputs so far minus all past outputs
  y = floor(v/d+rand(1))*d;   % Quantize accumulated error with uniform dither
  yo(i) = y;
end
```

(a) Use the above script with $\Delta = \texttt{d} = 1/8$ to quantize the oversampled signal `xo` with noise shaping. Measure the SQNR in dB by filtering `yo` and comparing with the filtered oversampled signal `xof=filter(h,1,xo)`. Listen to both the quantized waveform and the quantization error signal. Describe the artifacts that you hear in each case.

(b) Suppose the original signal `xo` is not bandlimited to $\Omega \in [-\frac{\pi}{M}, \frac{\pi}{M}]$, but instead has a spectrum that decays rapidly with frequency and becomes negligible for $|\Omega| > \frac{2\pi}{M}$. Can you think of a way to retain the high frequency components in the original signal and still take advantage of noise shaping?

Extra Credit: Digital Mastering and Noise Shaping

Noise shaping can also be used to improve the quality of D/A conversion and D/D quantization. The idea of digital-to-digital quantization is to reduce the number of bits per sample in an audio clip. With noise shaping, one can retain a high level of quality by shaping the noise into bands where hearing is less sensitive. The block diagram of a dithered noise-shaping quantizer is shown below.



In this diagram, $x(n)$ is the input, $d(n)$ is the dither sequence, $y(n)$ is the output, and the $h(m)$ is the impulse response of the noise-shaping filter. Using the same approach as we used in class, one can show that

$$Y(z) = X(z) + E(z)\,(1 - H(z))\,.$$

We note that $h(m) = 0$ for $m \leq 0$ is required for causality. Otherwise, choosing $h(0) = 1$ allows one to cancel all of the quantization noise. For audio applications with $F_s = 44100$, a good choice for $h(m)$ is

```
h = [0 1.47933 -1.59032 1.64436 -1.36613 0.926704 -0.557931 0.267859 -0.106726 ...
0.028516 0.001230 -0.006165 0.003067]
```

Try using `freqz([1 -h(2:end)])` to see the noise-shaping response of $1 - H(z)$.

Use this approach to convert `'singing8.wav'` from 16-bit resolution to 12-bit resolution. Plot the power spectral density of the quantization error to verify that your noise shaping is working correctly. Listen to both the quantized waveform and the quantization error signal. Describe the artifacts that you hear in each case.