Computer Assignment #3
Due Su. 2/25

LTI Systems and Fourier Series

Recall that the Fourier Series allows you to represent a periodic signal as a linear combination of (harmonic) complex exponentials

$$x[n] = \sum_{k=<N>} a_k e^{jk\omega_0 n} = \sum_{k=<N>} a_k e^{jk\frac{2\pi}{N}n},$$

where the period of $x[n]$ is N and the fundamental frequency is $\omega_0 = \frac{2\pi}{N}$. Also, LTI systems have the eigensignal property

$$z^n \longrightarrow H(z)z^n \qquad\qquad H(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n}.$$

Together, a periodic input signal $x[n]$ results in periodic output signal

$$y[n] = \sum_{k=<N>} b_k e^{jk\omega_0 n} = \sum_{k=<N>} H(e^{jk\omega_0}) a_k e^{jk\omega_0 n},$$

where $H(e^{j\omega})$ is known as the frequency response. The LTI system effectively scales the harmonic components of $x[n]$. The LTI system designer then would be looking to build $H(e^{j\omega})$ to affect harmonic frequencies in a desired manner. As an example, a low pass filter is a system $H(e^{j\omega})$ designed such that lower frequency harmonics (those with small k for frequency $\omega = k\omega_0$) do not change while higher frequency harmonics (k large) are zero.

The following exercises will demonstrate the effects of different LTI systems on periodic input signals and give some intuition.

Low Pass Filtering

A low pass filter can be thought of as a system that "smooths" a signal. By smoothing a signal, high frequency variations are reduced. A natural way to smooth a signal would be to average consecutive values with difference equation

$$y[n] = \frac{1}{M+1} \sum_{k=0}^{M} x[n-k].$$

This definition makes the output at time n the average of the last $M+1$ samples. This should remove high frequency variations in the signal. Notice that this difference equation is equivalent to an impulse response of

$$h[n] = \begin{cases} \frac{1}{M+1} & 0 \leq n \leq M \\ 0 & \text{else} \end{cases}.$$

This is known as and $M+1$ tap low pass averaging filter. Note that this is a causal averaging filter.

Exercises

1. Visualize the filter response by plotting the symmetric (non-causal) averaging filter

$$H(e^{j\omega}) = \frac{\sin(\omega(M+1/2))}{\sin(\omega/2)}$$

between $-\pi \leq \omega \leq \pi$ for $M = 4$. You only need to consider the magnitude response $|H(e^{j\omega})|$.

2. Use convolution (`conv.m`) to overlay the output $y[n]$ over the input $x[n]$ for input signals

   (a) $x_0[n] = \cos\left(\frac{\pi}{10}n\right)$
   (b) $x_1[n] = \cos(\pi n)$
   (c) $x_2[n] = x_0[n] + x_1[n]$

   Generate three plots in a row using the `subplot.m` command. Be sure to label your axis (`xlabel` and `ylabel`) and insert a legend (`legend.m`). You may want to play with the plot thickness and color for better visualization. Explain the effects of the LP filter on the signals.

3. Repeat 2, but this time add white Gaussian noise to the signal $x_i[n]$. You can add white noise using the function `randn.m`, e.g. `x0 = x0 + randn(1,100)` assuming `x0` has 100 samples. Explain the affects of the LP filter on the signals.

4. Repeat 2, but use the `filter` command rather than `conv`.

5. Repeat 3, using the `filter` command.

High Pass Filtering

In contrast to a low pass filter, the high pass filter will accentuate high frequency components and suppress low frequency. A simple high pass filter can be obtained by approximating a derivative function with $h[n] = \{1, -1\}$ or as the difference between samples.

Exercises

6. Visualize the filter frequency response by plotting

$$H(e^{j\omega}) = 1 - e^{-j\omega}$$

between $-\pi \leq \omega \leq \pi$. Note this is a complex function so you will need to plot the magnitude.

7. Use convolution to overlay the output $y[n]$ over the input $x[n]$ for input signals

   (a) $x_0[n] = \cos\left(\frac{\pi}{10}n\right)$
   (b) $x_1[n] = \cos(\pi n)$
   (c) $x_2[n] = x_0[n] + x_1[n]$

   Explain the effects of the HP filter on the signals.

8. Repeat 7, but this time add white Gaussian noise to the signal $x_i[n]$. Explain the affects of the HP filter on the signals.

9. Repeat 7, using the `filter` command.

10. Repeat 8, using the `filter` command.