EE123 Spring 2019 — Miki Lustig

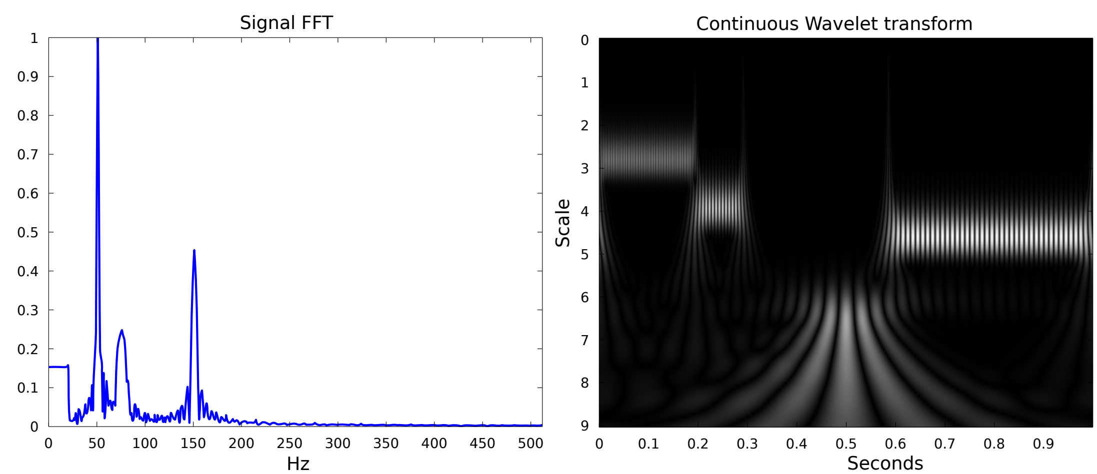
Digital Signal Processing

Assignment 6

Due March 11^{th} 2019

1. Self-grade Homework 5.
2. Read Chapter 4.1-4.4 Oppenheim and Schafer, 3rd ed.
3. *Adapted from Midterm I fall'12:* Time-Frequency Analysis
   We saw in class (and we will discuss more on Monday) that there are multiple ways to analyze the temporal-spectral components of signals. Here you will *qualitatively* determine a signal in the time domain based on these transforms.

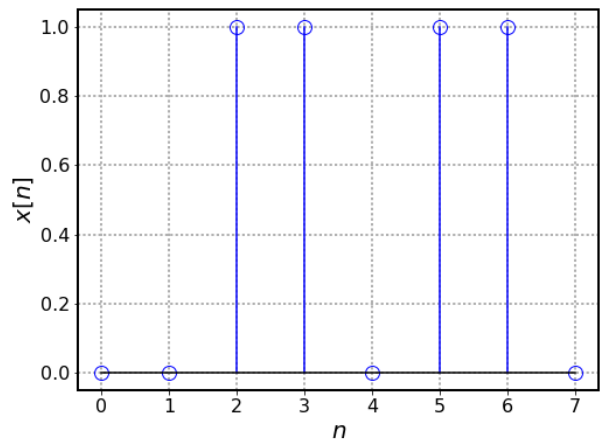
   You are given two graphs showing transforms of the signal s(t) and its samples. The left is the magnitude spectrum of a signal, the right is the continuous wavelet transform.



   Draw qualitatively the signal s(t). Provide explanation on how you got to the result based on the FFT and CWT. In particular explain the difference in width of each spectral component in the FFT plot. Label each portion of the plot!

4. *Adapted and modified from Midterm I fall'11:* Signal Interpolation

The following discrete 8-point signal, $x[n]$, is obtained by sampling a piece-wise constant continuous time signal. You would like to upsample the signal by a factor of 2 and get a 16-point sequence, $\tilde{x}[n]$, such that $\tilde{x}[2n] = x[n]$ and $\tilde{x}[n]$ is real. It turns out that there are many ways to interpolate the odd part of $\tilde{x}[n]$, which lead to different upsampled results. In each sub-problem below, you will implement an interpolation method and observe the interpolated sequence $\tilde{x}[n]$.



a) Use linear interpolation with the following procedure.

   i) Find the interpolated signal $\tilde{x}[n]$ by evaluating $\tilde{x}[2n+1] = \frac{1}{2}\left(x[n] + x[((n+1))_8]\right)$, for intergers $n \in [0, 7]$.

   ii) Draw $\tilde{x}[n]$. Does it represent the original continuous time signal well?

b) Use the following procedure to interpolate $x[n]$ with DFT.

   i) Find the 8-point DFT, $X[k]$, of $x[n]$.

   ii) Zero-pad and scale $X[k]$ to get a 16-point DFT $\tilde{X}[k]$, where $\tilde{X}[k] = 2X[k]$ for integers $k \in [0, 3]$ and $\tilde{X}[k] = 2X[k-8]$ for $k \in [12, 15]$. Find the interpolated signal $\tilde{x}[n]$.

   iii) Draw $\tilde{x}[n]$. Does it represent the original continuous time signal well?

c) Use the following procedure to interpolate $x[n]$ with Haar wavelet transform.

   i) Find the Haar-8 wavelet, $H[k]$, of $x[n]$.

   ii) Zero-pad and scale $H[k]$ to get a Haar-16 wavelet $\tilde{H}[k]$, where $\tilde{H}[k] = \sqrt{2}H[k]$ for integers $k \in [0, 7]$. Find the interpolated signal $\tilde{x}[n]$.

iii) Draw $\tilde{x}[n]$. Does it represent the original continuous time signal well?

5. *Wavelet Denoising*
This question assumes basic knowledge of probability. If you are having difficulties, contact me or the teaching staff.
Consider the following $N = 16$ discrete signal:

$$x[n] = [0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 0]$$

(a) What is $X[k]$, the Haar-16 transform of $x[n]$? How many non-zero coefficients are in $X[k]$?

(b) You observe noisy measurements,

$$y[n] = x[n] + v[n],$$

where $v[n] \sim \mathcal{N}(0, \sigma^2)$ are i.i.d. zero-mean Gaussian random variables. Note that $\sigma < 0.05$, so the signal to noise ratio is over 20.
Show that $V[k]$ which are the coefficients of the Haar-16 transform of $v[n]$ are also i.i.d. with $V[k] \sim \mathcal{N}(0, \sigma^2)$.

(c) You decide to denoise the signal by:

i. Computing $Y[k]$ the Haar-16 discrete wavelet transform (DWT) of $y[n]$

ii. Threshold $\tilde{Y}[k] = \begin{cases} 0, & \text{if } |Y[k]| < 0.3 \\ Y[k], & \text{otherwise} \end{cases}$

iii. Compute $\tilde{y}[n]$, the inverse DWT of $\tilde{Y}[k]$

This process pretty much guarantees that $\tilde{Y}[k]$ has zeros at the same indexes that $X[k]$ has zeros (what is the probability that $V[k] > 0.3 \geq 6\sigma$?)

What is the standard deviation of $\tilde{y}[n]$ for $0 < n \leq 15$? what is the amount of noise reduction compared to the standard deviation of $y[n]$?

(d) A moving average filter can also be used for reducing the noise in a signal. What is the length of filter needed to achieve approximately the same noise standard deviation as in the Haar denoising case?

(e) Let $v = [0.068,\ 0.009,\ -0.055,\ 0.044,\ 0.064,\ 0.033,\ -0.011,\ -0.068,\ -0.024,\ 0.034,\ 0.001,\ 0.017,\ -0.081,\ 0.013.\ -0.042.\ -0.02]$.
draw the result of filtering y with the moving average filter. Compared to $\tilde{y}[n]$ the wavelet denoising. (You can use Python)