Homework #10, EECS 451, W04. Due **Fri. Apr. 9**, in class

- Scores for Exams1 and 2 and HW1-8 are posted on the web. Please check them for typos. Please submit any score corrections for HW1-8 by Apr. 9.

Skill Problems

1. [B 0] Text 5.21. Concept(s): ***Analog sampling requirements***

2. [B 10] Concept(s): ***circular convolution***
Find $x[n] = x_1[n] \circledast_8 x_2[n]$ where $x_1[n] = n$ and $x_2[n] = \delta[n-1] + \delta[n+1] + \delta[n-3]$.

3. [B 0] Concept(s): ***DFT properties***
(a) [0] Determine the energy of the signal $x[n] = \cos\left(\frac{2\pi}{N} k_0 n\right)$, $n = 0, \ldots, N-1$.
Consider separately the cases where $k_0 \in \{1, \ldots, N-1\}$ and the case where $k_0 = 0$.
(b) [0] Suppose the real signal $x[n]$ satisfies the property $x[n + N/2 \bmod N] = -x[n]$ for N even.
Show that $X[k]$, the N-point DFT of $x[n]$, is zero for even values of k.
(c) [0] Determine the 9-point DFT of the signal $y[n]$ defined by $y[n] = \sum_{l=-\infty}^{\infty} x[n - 9l]$,
where $x[n] = n\left(\frac{1}{2}\right)^n u[n]$. Hint. First find the z-transform of $x[n]$.

Mastery Problems

4. [B 20] Concept(s): ***Linear phase filters.***
For each of the following statements, provide a proof if true, or a counter-example if false.
(a) [10] The sum of two linear-phase filters is a linear-phase filter.
(b) [10] The sum of two causal, FIR linear-phase filters of the same length is a linear-phase filter.
(c) [0] The cascade of two linear-phase filters is a linear-phase filter.

5. [B 40] Concept(s): ***up sampling and down sampling and DFT.***
For the following questions, assume $x[n] \overset{\text{DFT}}{\underset{N}{\longleftrightarrow}} X[k]$.
(a) [10] For upsampling by zero insertion: $y_1[n] = \begin{cases} x[n/2], & n \text{ even} \\ 0, & n \text{ odd}, \end{cases}$ relate the $2N$-point DFT of $y_1[n]$ to $X[k]$.
(b) [10] For downsampling (by discarding): $y_2[n] = x[2n]$, relate the $N/2$-point DFT of $y_2[n]$ to $X[k]$.
(c) [10] For upsampling by repeating: $y_3[n] = \begin{cases} x[n/2], & n \text{ even} \\ x[(n-1)/2], & n \text{ odd}, \end{cases}$ relate the $2N$-point DFT of $y_3[n]$ to $X[k]$.
(d) [10] For odd zeroing: $y_4[n] = \begin{cases} x[n], & n \text{ even} \\ 0, & n \text{ odd}, \end{cases}$ relate the N-point DFT of $y_4[n]$ to $X[k]$.
(e) [0] Think about duality. What would happen if we applied these operations in the frequency domain and instead looked for the corresponding time-domain relationships?

6. [B 0] Concept(s): ***DTFT vs DFT***
(a) [0] Find the DTFT $X(\omega)$ of $x[n] = \{\underline{0}, 0, 0, 2, 0, 0, 0, 2\}$.
(b) [0] Find the 8-point DFT (analytically) $\{Y[k]\}$ of $y[n] = \{\underline{1}, 0, 0, 0, 1\}$.
(c) [0] Relate $Y[k]$ to $X(\omega)$ mathematically by considering DFT/DTFT relationships and DFT properties.

7. [B 25] The DTFT of the signal $x[n] = (-1/2)^n u[n]$ is $X(\omega) = 1/(1 + \frac{1}{2} e^{-\jmath\omega})$. Ideally, taking N equally-spaced samples of this DTFT would give the values $X(\omega_k) = 1/(1 + \frac{1}{2} e^{-\jmath\frac{2\pi}{N}k})$ where $\omega_k = \frac{2\pi}{N}k$ for $k = 0, \ldots, N-1$. Even though $x[n]$ is not time-limited, it does decay to zero, so it is *approximately* time-limited. Thus we can use an N-point DFT of $x[n]$ to compute *approximate* samples of $X(\omega)$. This is very useful in practice.
   (a) [20] Use your `dft` routine (or MATLAB's `fft` routine) to compute the N-point DFT of $x[n]$ for $N = 4$, 8, and 16.
   For each N, make a stem plot of $|X[k]|$ vs ω_k (for $k = 0, \ldots, N-1$) and a line plot of $|X(\omega)|$ vs ω (for $\omega \in [0, 2\pi]$). Use MATLAB's `hold` function to put both graphs on the same axes so you can compare. Turn in m-file and 3 graphs.
   (b) [5] Describe what happens as N increases and why.

8. [G 0] Text 5.18. Concept(s): ***Reverse engineering a filter.***
   Hint: express $x[n]$ and $y[n]$ using DTFS.

9. [B 30] Concept(s): ***Convolution via FFTs***

   Consider the following DT signal: $x[n] = \begin{cases} 2, & 0 \le n < 16 \\ 3, & n = 16 \\ 2, & 16 < n < 32 \\ 1, & 32 \le n < 48 \\ 3, & 48 \le n \le 63 \\ 0, & \text{otherwise.} \end{cases}$

   You intend to simulate the effect of applying this signal as the input to a LTI system with system function $H(z) = (1 + z^{-1})/(1 - \frac{1}{2}z^{-1})$. Since determining the response $y[n]$ by hand could be painful, you wisely decide to use MATLAB.
   (a) [10] Use MATLAB's `filter` command to compute $y[n]$. Display as a stem plot.
   (b) [0] Does your stem plot describe $y[n]$ completely? Explain.
   (c) [10] Use 64-point DFT's to (approximately) compute $y[n]$, by multiplying the DFT of $x[n]$ with appropriate $H(\omega_k)$ values and then taking the inverse DFT. Plot the real and imaginary parts of the resulting signal as stem plots.
   (d) [0] Explain any differences between the plots in (c) and (a).
   (e) [0] Repeat (c) and (d) for $N = 128$.
   (f) [10] Now is the plot in (e) exactly $y[n]$ over the values of n shown? Explain.