1. In class lecture, a derivation of the Poisson Sum Formula was presented. In this problem you are to derive the dual to the Poisson Sum Formula,

\[ T \sum_{n=-\infty}^{\infty} x(t-nT) = \sum_{k=-\infty}^{\infty} X(k/T)e^{j2\pi kt/T}. \]  

Your derivation should parallel the one provided in lecture. Begin with an arbitrary time signal, \( x(t) \) with Fourier transform \( X(f) \). Then, for arbitrary choice of \( T > 0 \), form the time sum of translates, \( x_1(t) = \sum_{n=-\infty}^{\infty} x(t-nT) \). Show \( x_1(t) \) is periodic with period \( T \). Then, since \( x_1(t) \) is periodic, express its exponential Fourier series as

\[ x_1(t) = \sum_{m=-\infty}^{\infty} c_m e^{j2\pi mt/T} \]

where the Fourier series coefficients are given by \( c_m = (1/T) \int_{-T/2}^{T/2} x_1(t)e^{-j2\pi mt/T} dt \). Evaluate the integral, and finish the derivation.

2. A causal, discrete-time, linear time-invariant system with input \( x(n) \) and output \( y(n) \) has the unit-step response \( c(n) = (1/2)^n u(n) \).

   (a) Find the system transfer function, \( H(z) \).

   (b) Find all finite poles and zeros, sketch their location in the \( z \)-plane, and shade the region of convergence.

   (c) Find the system impulse response, \( h(n) \).

3. Find \( y(n) = x(n) * h(n) \) for the signals below. (The underline denotes the \( n = 0 \) sample.)

   (a) \( x(n) = (\cdots, 0, 1, 1, 1, 1, -1, -1, -1, 0, 0, \cdots) \), \( h(n) = (0.25, 0.5, 0.25) \).

   (b) \( x(n) = (\cdots, 0, 1, 1, 1, -1, -1, -1, -1, 0, 0, \cdots) \), \( h(n) = (-0.25, 0.5, -0.25) \).

   (c) \( x(n) = a^n u(n) \), \( h(n) = b^n u(n) \), where \( a \neq 0 \) and \( b \neq 0 \). Consider the case \( a = b \) as well as the case \( a \neq b \).

4. A discrete-time system is described by \( y(n) = ny(n-1) + x(n), n \geq 0, y(-1) = 0 \). Is the system i) linear, ii) time-invariant, iii) BIBO stable? Provide justification for your answer.

5. Find the \( z \)-transform of each of the following.

   (a) \( x(n) = (\cdots, 0, 3, 0, 0, 0, 0, 6, 1, -4, 0, \cdots) \)
(b) \( x(n) = \begin{cases} \left(\frac{1}{2}\right)^n, & \text{if } n \geq 5 \\ 0, & \text{otherwise} \end{cases} \)

(c) \( x(n) = \left(\frac{1}{2}\right)^n[u(n) - u(n - 10)] \). Also find all finite poles and zeros, and sketch their location in the \( z \)-plane.

(d) \( x(n) = \begin{cases} \left(\frac{1}{2}\right)^n, & \text{if } n \geq 0 \\ \left(\frac{1}{2}\right)^{-n}, & \text{if } n < 0 \end{cases} \) Also find all finite poles and zeros, sketch their location in the \( z \)-plane, and shade the region of convergence.

(e) \( x(n) = \left(\frac{1}{4}\right)^n - 2^n u(n) \). Also find all finite poles and zeros, sketch their location in the \( z \)-plane, and shade the region of convergence.

6. Let \( x(n) = (1, 1, 1, 1, 0, 0, 0, \ldots) \), for \( n = 0, 1, \ldots \). Let \( y(n) = x(n - n_0) \), where \( n_0 = 15 \).

(a) Use the Matlab \texttt{xcorr} function to compute the cross-correlation of \( x \) and \( y \), \( r_{xy}(k) \). Plot \( r_{xy}(k) \). Identify the delay, \( n_0 \), in the plot.

(b) Modify \( y(n) \) to \( y(n) = ax(n - n_0) + bw(n) \) where \( a \) and \( b \) are constants, and the signal \( w(n) \) is samples of a zero-mean, Gaussian random variable (noise) with variance 1. The noise samples can be generated in Matlab using the instruction \( w = \text{randn}(1, \text{length}(y)) \). Compute and plot \( r_{xy}(k) \) for the cases (i) \( a = 1, b = 0.2 \);
(ii) \( a = 1, b = 0.5 \); (iii) \( a = 1, b = 1.0 \).

(c) Repeat the case in b) for \( a = 1, b = 1.0 \) two more times (that is, different realizations of the Gaussian samples). Note: Use the Matlab subplot command to assemble multiple plots into one figure. Compare the plots in a)-c). When the noise has relatively small power \((b = 0 \text{ or } b = 0.2)\), can the cross-correlation be used to reliably identify the signal delay, \( n_0 \)? What about when the noise power is relatively large?

7. On my web page are links to four audio segments (Speech 1 (sent1.wav), Speech 2, Speech 3, and Speech 4). Pick one of these audio files and read it into Matlab. (Use the audioread command, e.g., \([xinput,Fs]=\text{audioread}('sent.wav');\) to read in the audio samples as the column vector \( xinput \), with the (read in from the file) sampling rate, \( Fs \).) Then select the 320 samples corresponding to samples 5701 to 6020 (e.g., \( x = xinput(5701:6020); \)). Call this signal the vector \( x \).

(a) Plot the signal \( x(n) \). Note the quasi-periodic characteristic.

(b) Compute, and plot, the autocorrelation function of \( x(n) \), denoted as \( r_{xx}(k) \). From the autocorrelation function, estimate the pitch period, \( T_p \), of the audio segment, \( x \), and from the pitch period determine the pitch frequency, \( f_p \).

(c) Compute the discrete Fourier transform (DFT) of the audio segment. Use the fast Fourier transform. (Use \( \text{X}=\text{fft}(x,1024); \) to compute a 1024-point DFT).
Then, plot the magnitude of the first 512 samples of the FFT result. Plot vs the frequency variable f, in Hz (in Matlab, this would be $f = [0:511]*8000/1024$; plot(f,abs(X(1:512))) ). Identify the fundamental frequency in the magnitude spectra, and compare this to the pitch frequency found in part b). For ease of comparison, it is convenient to place the three plots in a single figure (use the Matlab subplot command, e.g., subplot(3,1,1) followed by plot(x), subplot(3,1,2) and plot $r_{xx}(k)$, etc.)

8. Not required. A useful (and fun) application is called Audacity, and it can be readily downloaded and installed on a PC. Run the program, and use the File tab to select an audio file for processing. Try selecting Sent1.wav from problem 7. One can then play the entire sound segment, or just select a portion. Under the “Effect” tab is a menu of audio effects. For example, try the “change pitch” option, and change the pitch downward by a factor of two (e.g., by changing from A3 to A2, or make the change directly in the frequency block). What does this effect sound like? Can you think of any practical application of this effect?