ECE551, Fall 2016: Homework Problem Set #5
Due Oct. 18 2016 in class or eMail **by class time**.
Rev 1: minor updates and clarifications

1. **Sampling and Interpolation for Band-Limited Vectors**
A vector $x \in \mathbb{C}^M$ is called *band limited* when there exists an odd $0 \le k_0 \le M-1$ such that its DFT coefficient sequence X satisfies

$$X[k] = 0 \quad \text{for all} \quad \frac{k_0+1}{2} \le k \le M - \frac{k_0+1}{2} \tag{1}$$

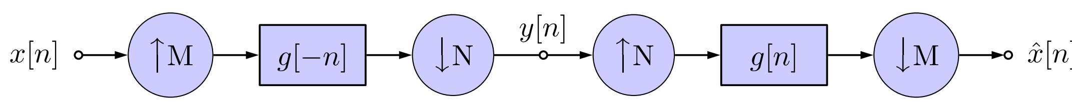
For a given bandlimited $x \in \mathbb{C}^M$, we call the smallest such k_0 the *bandwidth* of x. A vector in $\mathbb{C}^M$ that is not band limited is called a full-band vector.

(a) Determine the bandwidth of the following $x_1, x_2 \in \mathbb{C}^M$ for *every* $M \ge 1$:

$$x_1[n] = 1 + \cos(2\pi n/M) + \cos(8\pi n/M), \quad x_2[n] = \cos(3\pi n/M)$$

(b) The set of vectors in $\mathbb{C}^M$ with the bandwidth of at most k_0 is a subspace. For x in such a band limited subspace, find Φ so that the sampling followed by interpolation described by $\Phi\Phi^*$ in Section 5.2.1 achieves perfect recovery, $\hat{x} = x$.

2. **Band Limited Space With Rational Sampling Rate Changes**
Consider sampling followed by interpolation in the figure below



where the input sequence $x \in BL[-\frac{2\pi}{K}, \frac{2\pi}{K}]$ is band-limited, and the rectangular blocks are convolution systems. For the cases below, what condition on the filter g ensures that $\hat{x} = x$?

(i) $M = 2, N = 3$ and $K = 3$.

(ii) $M = 2, N = 3$ and $K = 4$.

(iii) General M, N, and K (with $M < N$).

3. **Multirate system**
Given is the discrete-time system $y = U_3 G D_2 x$, with G an LSI filter, U_3 the upsampling-by-3 operator, and D_2 the downsampling-by-2 operator.

(a) Express the z-transform of the output sequence y in terms of the z-transform of the input sequence x and the z-transform of the filter g.

(b) Suppose that the input sequence x is obtained by sampling a continuous-time function $q(t)$ at sampling frequency $\frac{1}{T}$ Hz, namely $x[n] = q(nT)$, where the function $q \in BL[\frac{-\pi}{T}, \frac{\pi}{T}]$ is band limited. Write the DTFT $Y(\omega)$ as a function of the Fourier transform $Q(\omega)$ and the DTFT $G(\omega)$. What are the conditions on $Q(\omega)$ to avoid aliasing?

4. **Pseudo-Inverse of Interpolation Filter: Single Channel case**
Read about the polyphase decomposition and the single channel filter bank in supplementary notes #2 (until and including Section 2.1).

Consider the discrete-domain sampling followed by interpolation depicted in Figure 1.

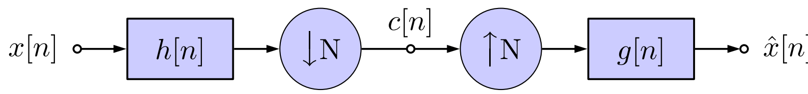


Figure 1: Sampling and Interpolation

Given an interpolation sequence $g[\cdot]$, **our goal is to find** $h[\cdot]$ **such that the output** $\hat{x}$ **best approximates the input** x in standard norm error: $\|x - \hat{x}\|^2$.

Recall from Eq. (5) in supplementary notes #2 the input/output polyphase relation: $\widehat{X}_p(w) = G_p(w)\tilde{H}_p(w)X_p(w)$. Show that for a given $g[\cdot]$, the optimal sampling sequence $h[\cdot]$ has the (type-II) polyphase vector

$$\tilde{H}_p(w) = (G_p^T(w^{-1})G_p(w))^{-1}G_p^T(w^{-1}).$$

Hint: the output $\hat{x}$ is a sum of shifted instances of g, living in a shift-generated subspace:

$$\hat{x} \in V := \overline{\text{span}}\{\sigma^{kN}g\}_{k\in\mathbb{Z}}$$

here $(\sigma g)[n] = g[n-1]$. By the orthogonality principle, the best $\hat{x}$ we can come up with has an orthogonal residue $x - \hat{x}$ to the subspace V, or equivalently

$$x - \hat{x} \perp V \iff \left\langle x - \hat{x}, \sigma^{kN}g \right\rangle = 0, \quad \text{for all } k \in \mathbb{Z}. \tag{2}$$

5. **Ideally-Matched Sampling and Interpolation with Nonorthogonal Filters**
Assume the setup of Problem 4 with $N = 2$ and $g[n] = \delta[n] + \frac{1}{2}(\delta[n-1] + \delta[n+1])$.

   (a) Find the optimal sampling filter h (in the mean square error) for that given interpolation filter g.

   (b) Find the shortest possible h that is *consistent* with the g (i.e $\tilde{H}_p(w)G_p(w) = 1$).

6. **Python Exercise: DTFT Approximation using DFT**
Let $x \in \ell_1(\mathbb{Z})$ be a signal whose DTFT is $X(\omega)$. Let $I = \{n_0, \dots, n_{N-1}\}$ be a set of N indices, and let $\hat{x} := \begin{bmatrix} x[n_0] & x[n_1] & \dots & x[n_{N-1}] \end{bmatrix}$ be a vector of the corresponding values of x on I. Lastly, let $\Omega = \{\omega_0, \dots, \omega_{M-1}\}$ be a set of M frequencies in $[0, 2\pi]$.

   (a) Write an approximation of $X(\omega)$ at the frequencies Ω based on the samples in $\hat{x}$, and implement in a Python function `dtft_approx(I,hat_x,omegas)`.

   (b) Assume that the indices in I are consecutive: $n_k = n_0 + k$ where $k = 0, \dots, N-1$, and the frequencies in Ω are equi-spaced: $\omega_m = 2\pi\frac{m}{M}$ for $m = 0, \dots, M-1$.
   Write an approximation of $X(\omega)$ using the DFT of the vector $\hat{x}$ (**hint:** consider two separate cases: $M \geq N$ and $M < N$). Implement your formula as a Python function `eq_dtft_approx(hat_x,n0,M)`. Use `numpy.fft.fft()`.
   Verify that the functions of part (a) and part (b) give similar results for a random $\hat{x}$ with $N = 4096$ and $M = 200,\ 5000$. Which of the functions runs faster ?

   (c) Use your function from (b) to plot the magnitude and phase of the DTFT of the filters/signals in problems 2(b) and 3(b) of Homework $\{3, 4\}$. Explain how the choice of n_0, N and M affects the approximation. Comment on the results.

7. **Python Exercise: Image Scaling with Separable Filters**

In some applications we store down-scaled versions of images (reducing pixel count), and rescale to their original size when required. This *lossy* process is depicted in Figure. 1.

The actual downscaling is done by simple downsampling:

$$c[m,n] = (D_N r)[m,n] := r[mN, nN],$$

and stores approximately $\frac{1}{N^2}$ of the original pixel count.

To prevent aliasing, the sampling is often preceded by a 2D convolution filter:

$$r[m,n] := \sum_{m',n'} h_2[m'-m, n'-n]x[m,n].$$

We will assume a *separable* convolution filter, namely, the same 1D sequence $h[\cdot]$ is convolved with the rows and with the columns of the image, or $h_2[m,n] = h[m]h[n]$. For any sequence $h[\cdot]$ we define the seprable filter system L_h as

$$(L_h x)[m,n] = \sum_{m',n'} h[m'-m]h[n'-n]x[m,n]. \tag{3}$$

and the down-scaled image is $c = D_N L_h x$. Upscaling $c[\cdot,\cdot]$ back to the original size is done by upsampling U_N:

$$(U_N c)[m,n] = \begin{cases} c[i,j] & m = iN, n = jN \\ 0 & \text{otherwise} \end{cases},$$

followed by a separable interpolation filter L_g, as defined in (3). The recovered image,

$$\hat{x} = L_g U_N D_N L_h x,$$

depends on the sampling and interpolation sequences $h[\cdot]$ and $g[\cdot]$. We are interested in experimenting with different choices of pairs (h, g).

Assume a separable interpolation filter $g[n] = \delta[n] + \frac{1}{2}(\delta[n-1] + \delta[n+1])$ and a down-sampling factor $N = 2$. Experiment with the following four different pre-filters:

(a) $h = \delta$ - no pre-filtering.

(b) $h[n] = g[-n]$ - the adjoint of the interpolation operator as the sampling operator.

(c) h is the shortest *consistent sampling* filter you designed in Problem 5(b).

(d) h is the *optimal (pseudo-inverse)* sampling filter you designed in Problem 5(a).

Evaluate these 4 methods with this test image, using the mean square error $\|x - \hat{x}\|^2$ as performance index. Comment on aliasing (focus on legs/scarf), and total filtering gain.
Useful functions: `scipy.signal.convolve2d`, `outer` (to make a separable filter), `scipy.signal.filtfilt` (apply a stable rational IIR filter).