Homework 3

Due: 2 Nov 2023
(late homeworks penalized 10% per day)

See the course web site for submission details. Reminder: rather than using the functions `pinv()` and `norm()`, use the linear algebra tools we learned in class. Do yourself a favor, and don't wait until the day before the due date... *start now*!

1. **LSI system characterization.** You are trying to experimentally characterize three auditory neurons, in terms of their responses to sounds. For purposes of this problem, the responses of these neurons are embodied in compiled matlab functions `unknownSystemX.p` with X=1, 2, 3. If you are using Python, import the `unknown_systems` module from the obfuscated Python file. Each takes an input column vector of length $N = 64$ whose elements represent sound pressure over time. In Python the response of each is a column vector (of the same length) representing the mean spike count over time. For each neuron,

   (a) "Kick the tires" by measuring the response to an impulse in the first position of an input vector. Check that the system is consistent with shift-invariance by comparing this to the response to an impulse at positions $n = 2, 4, 8$. Check that the system is consistent with linearity by testing whether the response to a sum of any two of these impulses is equal to the sum of their individual responses. Also examine respones to impulses at different n to determine how the system handles inputs near the boundary (i.e., whether the system does circular boundary-handling). Describe your findings.

   (b) If the previous tests succeeded, examine the response of the system to sinusoids with frequencies $\{2\pi/N, 4\pi/N, 8\pi/N, 16\pi/N\}$, and random phases, and check whether the outputs are sinusoids of the same frequency (i.e., verify that the output vector lies completely in the subspace containing all the sinusoids of that frequency). [Note: make all elements of the the input stimuli positive, by adding one to each sinusoid. The responses will then also be positive (mean spike counts).]

   (c) If the previous tests succeeded, verify that the change in amplitude and phase from input to output is predicted by the amplitude (`abs`) and phase (`angle`) of the corresponding terms of the Fourier transform of the impulse response. If not, explain which property (linearity, or shift-invariance, or both) seems to be violated by the system. If so, does the combination of all of your tests *guarantee* that the system is linear and shift-invariant? What set of tests would provide such a guarantee?

2. **Neuron in visual cortex.** The response properties of neurons in primary visual cortex (area V1) are often described using linear filters. We'll examine a one-dimensional cross-section of the most common choice, known as a "Gabor filter" (named after Electrical Engineer/Physicist Denis Gabor, who developed it in 1946 for use in signal processing).

   (a) Create a one-dimensional linear filter that is a product of a Gaussian and a sinusoid, $\exp\left(-\frac{n^2}{2\sigma^2}\right)\cos(\omega n)$, with parameters $\sigma = 3.5$ samples and $\omega = 2\pi * 10/64$ radians/sample. The filter should contain 31 samples, and the Gaussian should be centered

on the middle (16th) sample. Plot the filter to verify that it looks like what you'd expect. Plot the amplitude of the Fourier transform of this filter, sampled at 64 locations (MATLAB's `fft` function takes an optional additional argument). What kind of filter is this? Why does it have this shape, and how is the shape related to the choice of parameters (σ, ω)? Specifically, how does the Fourier amplitude change if you alter each of these parameters?

(b) If you were to convolve this filter with sinusoids of different frequencies, which of them would produce a response with the largest amplitude? Obtain this answer by reasoning about the equation defining the filter (above), and also by finding the maximum of the computed Fourier amplitudes (using the `max` function), and verify that the answers are the same. Compute the *period* of this sinusoid, measured in units of sample spacing , and verify by eye that this is matched to the oscillations in your plot of the filter. Which two sinusoids would produce responses with about 25% of this maximal amplitude?

(c) Create three unit-amplitude 64-sample sinusoidal signals at the three frequencies (low, mid, high) that you found in part (b). Convolve the filter with each, and verify that the amplitude of the response is approximately consistent with the answers you gave in part (b). (Hint: to estimate amplitude, you'll either need to project the response onto sine and cosine of the appropriate frequency, or compute the DFT of the response and measure the amplitude at the appropriate frequency.)

3. **Deconvolution of the Haemodynamic Response.** Neuronal activity causes local changes in deoxyhemoglobin concentration in the blood, which can be measured using functional magnetic resonance imaging (fMRI). One drawback of fMRI is that the haemodynamic response (blood flow in response to neural activity) is much slower than the underlying neural responses. We can model the delay and spread of the measurements relative to the neural signals using a linear shift-invariant system:

$$r(n) = \sum_k x(n-k)h(k), \tag{1}$$

where $x(n)$ is an input signal delivered over time (for example, a sequence of light intensities), $h(k)$ is the haemodynamic response to a single light flash at time $k = 0$ (i.e., the impulse response of the MRI measurement), and $r(n)$ is the MRI response to the full input signal.

In the file `hrfDeconv.mat`, you will find a response vector r and an input vector x containing a sequence of impulses (indicating flashes of light). Your goal is to estimate the HRF, h, from the data. Each of these signals are sampled at 1 Hz. Plot vectors r and x versus time to get a sense for the data. Use the `stem` command (or `plt.stem` in Python) for x, and label the x-axis.

(a) Convolution is linear, and thus we can re-write the equation above as a matrix multiplication, $r = Xh$, where h is a vector of length M, N is the length of the input x, and X is an $[N + M - 1] \times M$ matrix. Write a matlab function `createConvMat`, that takes as arguments an input vector x and M (the dimensionality of h) and generates a matrix X such that the response $r = Xh$ is as defined in Eq. (1) for any h. Verify that the matrix generated by your function produces the same response as Matlab's `conv` function when applied to a few random h vectors of length $M = 15$. Visualize the matrix X as an image (evaluate `imagesc(X)` in MATLAB or `plt.imshow` in Python), and describe its structure.

(b) Now, given the X generated by your function for $M = 15$, solve for h by formulating a least-squares regression problem:

$$h_{\text{opt}} = \arg\min_{h} ||r - Xh||^2$$

Plot h_{opt} as a function of time (label your x-axis, including units). How would you describe it? How long does it last?

(c) It's often easier to understand an LSI system by viewing it in the frequency domain. Plot the power-spectrum of the HRF (i.e. $|\mathcal{F}(h)|^2$, where $\mathcal{F}(h)$ is the Fourier transform of the HRF). Plot this with the zero frequency (DC) in the middle (in Matlab you can use a built-in function called `fftshift`), and label the x axis, in Hz. Based on this plot, what kind of filter is the HRF? Specifically, which frequencies does it allow to pass, and which does it block?

(d) Use the convolution theorem to now find h_{opt} by working in the fourier domain. You will need to use the matlab functions `fft` and `ifft`. Remember to be careful about how many samples you choose to have in your fft. Based on the operations you have done, what can you say about when this method will fail? On the same graph, plot the HRF impulse response you recovered from parts (c) and (d).

4. **Sampling and aliasing.** Load the file `myMeasurements.mat` into matlab. It contains a vector, `sig`, containing voltage values measured from an EEG electrode, sampled at 120Hz. Plot `sig` as a function of vector `time` (time, in seconds), using the flag 'ko-' in matlab's plot command so you can see the samples.

(a) The voltage signal is densely sampled (120Hz), and thus expensive to store. Create a *subsampled* version of the signal, which contains every *fourth* value. Plot this, against the corresponding entries of the `time` vector, on top of the original data (use matlab's `hold` function, and plot with flag 'r*-'). Is the subsampling operation linear? Shift-invariant? How does this reduced version of the data look, compared to the original? Does it provide a good summary of the original measurements? Explain.

(b) Examine your EEG result in the frequency domain. First plot the magnitude (amplitude) of the Fourier transform of the original signal, over the range $[-N/2, (N/2) - 1]$ (use `fftshift`). By convention, the "Delta" band corresponds to frequencies less than 4Hz, "Theta" band is 4-7Hz, "Alpha" band 8-15Hz, and "Beta" is 16-31Hz. For these data, which band shows the strongest signal? Is there any power in frequencies outside of these known bands, and if so can you explain the origin of this part of the signal?

(c) Write a function `signalPart = bandWiseReconstruct(bandName)` that reconstructs the signal (and plots the reconstruction) using only sinusoids from the band corresponding to the string `bandName` (i.e. for `bandName = 'Delta'` the reconstruction should be a sum of sinusoids with frequencies from 0-4Hz).

(d) Plot the Fourier magnitude for signals downsampled by factors of 2, 3, and 4, after upsampling them back to full size (i.e., make a full-size signal filled with zeros, and set every kth sample equal to one of the subsampled values, for subsampling by factor k). What is the relationship between these plots and the original frequency plot. What has happened to the frequency components of the original signal? Does the strongest signal band change?