PURDUE UNIVERSITY

Weldon School of Biomedical Engineering

Biomedical Signal Processing
BME 511 - Fall 2021
Problem Set 4

This problem set accounts for 17% of your final grade (5 points for the final project proposal and 12 points for the other parts). Please submit your solutions through **Brightspace** as a single PDF file containing typed explanations, plots, and any code that has been requested explicitly. Keep the text as brief as possible. Each problem lists the plots and results that need to be included in your solutions report. Please label the axes of all plots. You need not include any code that is not explicitly requested, but no problem if you do.

Note: For the time-frequency energy estimation problems (i.e., fixed window-length spectrograms and wavelet spectrograms), you could use the provided `tfr.py` module, which was used in the Jupyter Notebooks that we created in class. If you are using MATLAB, an approximately equivalent function is provided on Brightspace with this problem set. However, if you prefer, you are also welcome to use other spectrogram and wavelet libraries (e.g., `pywt`, `scipy.signal`), or your own code.

1. Please turn in your final project proposal (roughly one page). The guidelines for the proposal can be found in the Final Project module in the course content area on Brightspace. Brief feedback will be provided within 2-3 days of submission, with the primary goal of the feedback being about whether the scope of the aims is too large and needs toning down or adjusting[1].

2. In not more than 2-3 sentences each,

   - describe the primary utility of tapering in the construction of spectrum estimates, and
   - describe the Welch method, and the main tradeoffs involved in constructing spectrum estimates using the Welch method.

3. Here, we will compare various spectrum estimation and time-frequency analysis trade-offs using a synthetic quadratic chirp signal. A chirp is a signal that is monotonically changing in frequency. For a quadratic chirp, the frequency is changing quadratically with time. Chirps can be generated by recognizing the relationship between the instantaneous phase and the instantaneous frequency of a (locally narrowband) signal. For a cosine with a constant frequency f_0, i.e., $cos(\phi(t)) = cos(2\pi f_0 t)$, the phase $\phi(t) = 2\pi f_0 t$ is changing linearly with time. That is, the frequency f_0 can be recognized as the normalized time-derivative of the phase, i.e., $f_0 = \frac{1}{2\pi}\frac{d\phi}{dt}$. Thus, if we want the frequency to increase linearly, then the phase will have to increase quadratically, and if we want the frequency to increase quadratically (as we do here), the phase will have to increase cubically. We will use this idea to generate our signals.

   (a) Let's say we want the frequency of the chirp to quadratically go from 100 to 500 Hz over the one-second interval from $t = 0$ to $t = 1$. That is, $f = 100 + (500 - 100)t^2 = 100 + 400t^2$. So we want the phase to be

   $$\phi(t) = 2\pi \left\{100t + (400/3)t^3\right\} + \phi_0 \quad (1)$$

   where ϕ_0 is the initial phase, which we can take to be zero. Use $cos(\phi)$ with the above definition of ϕ to generate a one-second-long quadratic chirp in a background of white noise

[1] Other parts of ths problem set will take longer to grade

(say standard deviation of 0.5). Plot the signal in time domain. Use a sampling rate of $10,000$ for this problem.

(b) Plot the periodogram spectrum estimate of this signal (treating it as a stationary signal) over the frequency range of zero to 1000 Hz. Overlay on top of it, a multitaper estimate of the PSD with four DPSS tapers (choose your frequency resolution for the multitaper estimate such that the 4 tapers will have high main-lobe concentration). Comment on the differences between the two estimates.

(c) If $x(t)$ is the original chirp signal, let's define $y(t) = x(-t)$, i.e., the time reversed version. Construct and plot a 4-DPSS-taper estimate for this new chirp. Comment on the differences between the estimated spectrum for this new chirp and the multitaper estimate you got for the original chirp.

(d) Plot spectrograms (over the frequency range of $50-800$ Hz) of both the original and time-reverse chirps with segments of constant length of $100ms$ using a single DPSS taper (and whatever frequency resolution that will allow). Use a logarithmic colormap (i.e., plot $10log_{10}(S_{xx}(f)$ instead of S_{xx}). Does the spectrogram reveal quadratic chirps with some background noise?

(e) Repeat the spectrogram plots but with 4 DPSS tapers and comment and the differences with 1 DPSS taper. Is the tradeoff evident from the plot?

4. Now let's apply multiresolution time-frequency analysis using a **wavelet** basis to understand the acoustic cues that differentiate a "ba" sound from a "da" sound of similar pitch in the same person's voice. What we mean by an acoustic "cue" in this context is the sound feature that allows our ears/brains to create distinct percepts of these the two sounds.

(a) Utterances of "ba" and "da" sounds can be found in the accompanying `speech.mat` file. Construct and plot wavelet-based spectrograms with nine-cycle long single-DPSS-tapered complex exponentials over the frequency range of $32 - 2048$ Hz with 20 frequency bins per octave (i.e., successive frequencies should have a ratio of $2^{1/20}$). It is common for wavelet analysis to have frequency bins that are logarithmically spaced, as suggested in this problem. The choice of 9 cycles is motivated by the fact that the human inner ear can be thought of as a filter bank with each filter having a quality factor (i.e., center-frequency divided by bandwidth) of $8 - 10$ at moderate sound levels. Plot the spectrogram with linear colormap (i.e., *not* in log scale). Is the wavelet analysis producing the expected time-frequency tiling (i.e., better frequency resolution at low frequencies and better time resolution at high frequencies)?

(b) For both sounds you should observe two hot horizontal bands in the spectrograms one around $800 - 850$ Hz and another around $1200 - 1250$ Hz. These are known as "formants" and arise from the resonant properties of different cavity shapes created by the articulators (e.g., lips, teeth, soft palate, uvula, glottis, etc.) in the speaker's mouth. The particular frequencies of the two formants here correspond to the vowel "aa". A nice demo of the articulators in action while producing some german vowels and some consonants can be seen in the dynamic MRI video from the Max Planck institute uploaded with this problem set!

While the formants convey the vowel, how the articulators move before landing on those formants convey the consonant in some cases. In particular, the difference between "ba" and "da" are in those **formant transitions**. Based on the spectrograms you generated, briefly describe the formant transition pattern that you observe for each of the two formants for each of the two consonants and comment on how they may be different.

5. Now let's explore the idea behind data compression techniques that use wavelet transforms. For this part, we will need implementations of wavelet transforms that are also readily invertible, i.e., we need to be able to go from wavelet transform back to the signal. Reconstruction (i.e., inverse transform) becomes simple if we can just add one wavelet per coefficient and get an exact reconstruction of the original signal. Certain families of wavelets are particularly well-suited for this purpose (e.g., Daubechies wavelets, Coiflets, Symlets, etc.).

   These are implemented in the PyWavelets module (i.e., `pywt`) available with Anaconda, and in MATLAB's signal processing toolbox. In both implementations, the functions have similar names (`wavedec` and `waverec`).

   (a) Obtain a wavelet decomposition of the "ba" and "da" sounds using the `coif3` wavelet. Now, drop the bottom 66% of all the coefficients (e.g., if you have $10,000$ coefficients, you would only retain the highest $3,333$ of them by absolute value – your number of coefficients will be diffferent). Drop coefficients by setting the smallest 66% to zero, reconstruct the sound back (this is 3x compression). Do you hear any difference between the intact sound and the reconstruction from only 33% of the coefficients? Please provide the code you use for this part within your PDF document (not as a separate file).

   (b) As you reduce the number of coefficients, at what point (approximately) can you tell that the sounds are processed (i.e., you hear it with a different quality than the original)? That is, if the objective of the compression was to reduce size while remaining roughly perceptually indistinguishable from the original, what degree of compression can you achieve?

   (c) As you reduce the number of coefficients, at what point (approximately) are you unable to tell the difference between "ba" and "da"? That is, if the objective of the compression was to reduce size while retaining discriminability between the two consonants (even though each might sound different from the original), what degree of compression can you achive?

6. As all electronic health records continue to be digitized, thoracic radiographs (chest X-rays) present a challenge in that these images occupy substantial memory in raw form (e.g., a 1024 x 1024 image with 8 bits per pixel would be about 1 MB), and are routinely done many clinical scenarios, thus accumulating a large amount of data. However, using wavelet techniques, considerable compression can be achieved.

   (a) To demonstrate this, use second order Daubechies wavelets (i.e., `db2`) and a 6-level decomposition to compresses the provided radiograph image (`chestradiograph.mat` file) about 20-fold (i.e., retain only 5% of the coefficients), and reconstruct it back to the original image space. Compare the original image and compressed reconstructions by displaying them side-by-side. **Note:** Here, given that data are images, you will want to use `wavedec2` and `waverec2` for decomposition and reconstruction.

   (b) To enhance the edge-details in the compressed image, use an appropriate 3×3 convolution kernel to sharpen the image. Compare the original and the (compressed + sharpened) version by dispalying them side-by-side. Briefly, what are your impressions of the quality of this encoding/display scheme compared to the original (no wrong answers here).