Worksheet: spectral time series analysis

Joaquin Rapela

January 31, 2025

The first five problems will examine oscillatory activity in local field potentials recorded from the infralimbic cortex and from the basolateral amygdala of rats. These recordings are available in this repository, and two publications related to this data are cited in the repository's README.md.

In the following exercises you will use a 45-minutes local field potential recording saved at a temporal resolution of 30,000 samples per second, i.e., 30 kHz, in the file *08102017/time_data_pre_45sec.mat* of the previous repository. To speed up its processing, I downsampled this file to a resolution of 3,000 samples per second, i.e., 3 kHz. You can download the downsampled file from here.

1. Plot data from one channel in this recording. You can adapt the Python script provided here.

2. Estimate and plot the sample mean and covariance function of the data plotted above. Does the data appear to be wide-sense stationary?

   Hint: you could estimate the sample mean and covariance function from different section of the data. If these estimates change substantially among different segments, the data is probably not wide-sense stationary.

3. Estimate the spectral density using the periodogram method, as indicated in the section Step 4: Power spectral density, or spectrum of the lecture *The Power Spectrum (Part 1)*. You may want to complete the code provided here.

4. The periodogram is a noisy estimator of the spectral density. Use the function scipy.signal.welch to re-estimate the spectral density function using the Welch method, and check if this estimate is better than the periodogram one.

5. Both the periodogram and the Welch method assume that the time series is stationary. Use the spectrogram, as described in the section Step 6: The spectrogram of the lecture *The Power Spectrum (Part 1)*, to bypass this assumption, and to check if the spectral density changes with time.

6. (optional) We measure the LFP in human motor cortex with an Utah array. It is known that this LFP only has an oscillation at 11 Hz (i.e., $LFP(t) = \cos(\omega_0\, t)$ with $\omega_0 = 2\pi\, f_0$ rad/sec, $f_0 = 11$ Hz). However, when we sample this LFP at a frequency of 10 Hz (i.e., $\omega_s = 2\pi\, f_s$ rad/sec, $f_s = 10$ Hz) we only observe an oscillation at 1 Hz (Figure 6).

   (a) explain the appearance of the 1 Hz oscillation using the sampling theorem.

   Hints:

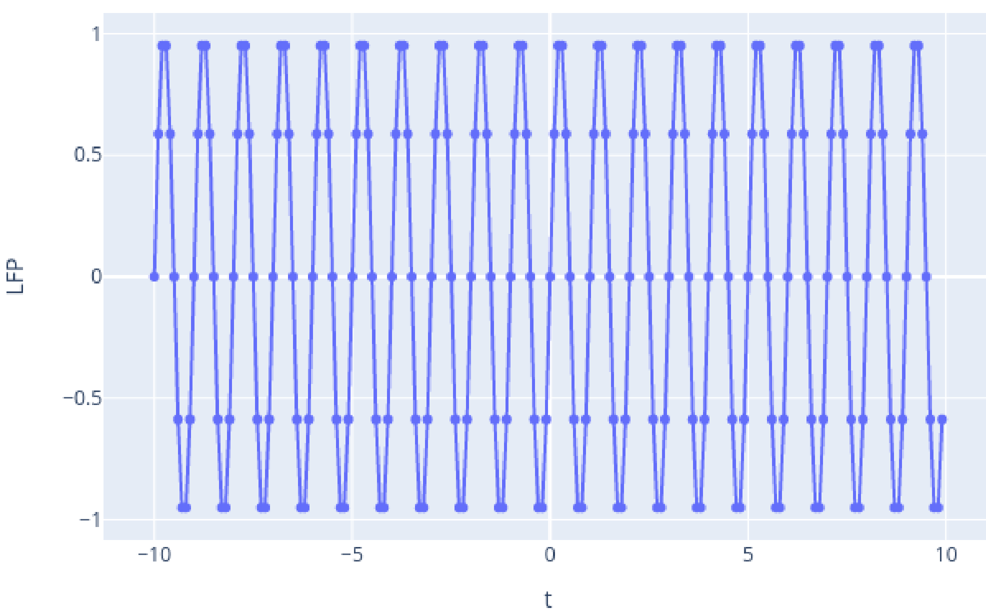
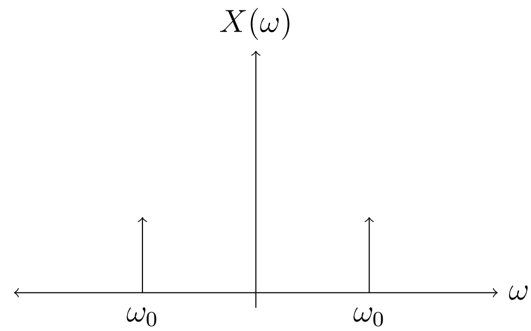


Figure 1: An LFP oscillating at 11 Hz (i.e., $LFP(t) = \cos(\omega_0 t)$ with $\omega_0 = 2\pi f_0$ rad/sec, $f_0 =$ 11 Hz) when sampled at a frequency of 10 Hz (i.e., $\omega_s = 2\pi f_s$ rad/sec, $f_s = 10$ Hz) only displays an oscillation at 1 Hz. Use the sampling theorem to explain this observation. Code to generate this figure appears here.

- the Fourier transform of a cosine is $\mathcal{FT}\{\cos(\omega_0 t)\} = \frac{1}{2}[\delta(\omega - \omega_0) + \delta(\omega + \omega_0)]$ and has the spectrum in the figure below.



- replicate the above spectrum, as indicated by the sampling theorem, with replicas at multiples of the sampling frequency $\omega_s = 2\pi\, f_s$ ($f_s = \frac{1}{T_s} = 10\ Hz$).
- check if any of the above replicates adds signal at 1 Hz (i.e., $\omega = 2\pi\, f$ rad/sec, $f = 1$ Hz).

Note: to avoid this type of problems where low-frequency oscillations appear due to frequencies in the signal above the Nyquist frequency (i.e., half of the sampling frequency), signals are low pass filtered with an analog filter at the Nyquist frequency before being sampled. This filter is called an **antialiasing filter**. Analog filters do not generate aliasing.

(b) use a sampling frequency above the Nyquist rate (i.e., $f_s > 2f_N$ where f_N is the largest frequency in the signal) and check that oscillations at 11 Hz appear in the sampled signal. You may want to use this code.

(c) build another example of an LFP having an oscillation at a high frequency that when sampled at a frequency below the Nyquist rate generates an oscillation at a lower frequency. You can use this code to verify that with your values of the LFP frequency and the sampling frequency an oscillation at a low frequency emerges.