Cross-Frequency Coupling

Naureen Ghani

April 28, 2018

Introduction

The interplay of excitation and inhibition is a fundamental feature of cortical information processing. The opposing actions of pyramidal cells and interneurons give rise to membrane and network oscillations, which provide temporal coordination of messages conveyed by cells. Neocortical interneurons are ideally positioned to control circuit dynamics and contribute to the vast and morphological variability of the cortex. Thus, inhibition is important in dictating the range and timing of activity in neural networks.

Recent studies have shown that network oscillations change in patients with major psychiatric diseases. In this way, studying brain rhythms is an effective method to understand the basis of neuropsychiatric disease and to identify target therapies.

In this tutorial, we will develop code to assess ***cross-frequency coupling (CFC). Cross-frequency coupling is the interaction between oscillations at different frequency bands.*** There are several synchronized neuronal assemblies in the brain, and each supports a frequency band of the network rhythm. We would like to quantify CFC to identify:

- interaction between local neural circuits
- changing of intrinsic properties in each circuit

Hilbert Transform

To build a CFC measure, the first step is to find the ***phase or amplitude envelope*** of the signal. We can do this by

```
% Generate sinusoid signal

dt = 0.001; % sampling interval [s]
t = dt:dt:1; % time axis [s]
f1 = 2.0; % freq of sinusoid [Hz]
phi0 = 0.0; % initial phase of sinusoid
d = sin(2.0*pi*t*f1 + phi0);

% Compute analytic signal
dA = hilbert(d);

% The built-in "hilbert" function in MATLAB returns the analytic signal. We can
   plot the original signal, and the real and imaginary parts of the signal.

figure; clf() % clf = clear current figure window
subplot(4,1,1) % 4x1 plot, 1st plot
plot(d); ylabel('Data');
subplot(4,1,2) % 4x1 plot, 2nd plot
plot(real(dA));
hold on;
plot(imag(dA), 'r');
```

```
hold off;
ylabel('Real (blue), Imag (red)');
axis tight

% Note that the imaginary part of the analytic signal is the real part of the
    analytic signal shifted by 90 degrees (pi/2)

% Compute phase of analytic signal
phi = angle(dA);

% Compute amplitude envelope
amp = abs(dA);

% Plot results
subplot(4,1,3); plot(phi); ylabel('Angle'); axis tight
subplot(4,1,4); plot(amp); ylabel('Amplitude'); axis tight
```

Amplitude and Phase

In mathematics and signal processing, an ***analytic signal*** is a complex-valued function that has no negative frequency components. When we apply the Hilbert transform to a real signal, the result is the ***analytic signal.*** In this section, we will clarify basic terminology.

The ***period*** is the peak-to-peak distance. The ***amplitude*** is the height from the center line to the peak (or trough). If the general equation of a signal is:

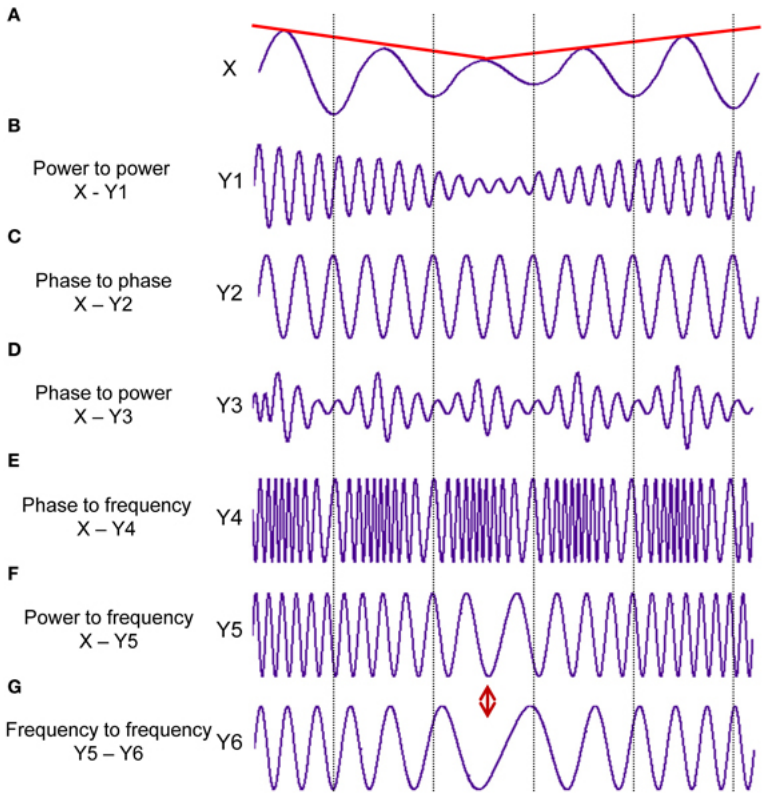
$$y = Asin\left(Bx + C\right) + D$$

- amplitude is A
- period is $2\pi/B$
- phase shift is $-C/B$
- vertical shift is D

Types of Cross-Frequency Coupling

There are 3 major types of cross-frequency coupling:

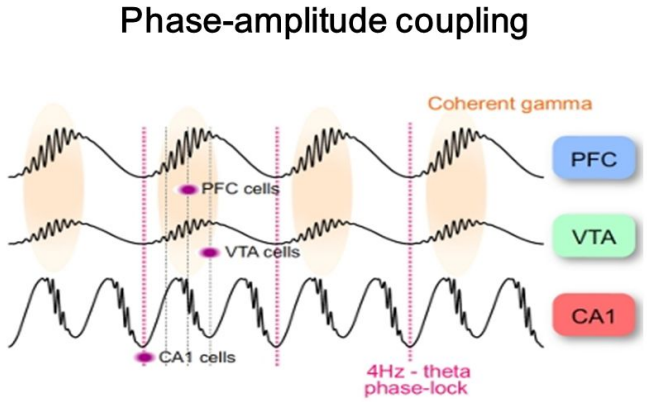
- phase-phase coupling
- amplitude-amplitude coupling
- phase-amplitude coupling



Phase-Amplitude Coupling

In neuroscience, one theory for the mechanism of phase-amplitude coupling is that:

- low frequency phase reflects local neuronal excitability
- high frequency power increases reflect:
  - a general increase in population synaptic activity (broad-band power increase)
  - selective activation of a connected neuronal subnetwork (narrow-band power increase)



- A well-studied mechanism is cross-frequency coupling. As described first in the hippocampus, the phase of theta oscillations biases the amplitude of the gamma waves [phase–amplitude (P–A) coupling or "nested" oscillations].

Measuring Power

The **periodogram** function in MATLAB computes the signal's Fast Fourier Transform (FFT) and normalizes the output (scales all values to be between 0 and 1) to obtain a ***power spectral density (PSD).*** The PSD describes how the power of a time signal is distributed with frequency. It represents power [V^2]/ frequency [Hz]. In non-neural data, the units are [watts]/[Hz]. The PSD is real-valued, and thus does not contain any phase information.

In this part of the demo, we will compute the PSD of a signal.

```
% Analytic signal has no power at negative frequencies. We will show this %
    result in a numerical simulation
% Generate a noisy signal
dt = 0.001;
t = dt:dt:1;
T = max(t);
N = length(t);
d = randn(N,1);

% compute analytic signal dA = hilbert(d);
pow_d = 2*dt^2/T * fft(d).*conj(fft(d));
pow_dA = 2*dt^2/T * fft(dA).*conj(fft(dA));

% Define freqency axis df = 1/T;
% Nyquist frequency: In order to recover all Fourier components of a
% periodic waveform, it is necessary to use a sampling rate "v" at least % twice
    the highest waveform frequency. so "fNQ" = 0.5v

fNQ = 1/dt/2;
faxis = -fNQ: df: fNQ-df;

figure;
subplot(2,1,1);
plot(t,d);
ylabel('Data');
xlabel('Time [s]');
subplot(2,1,2);
plot(faxis, pow_d);
ylabel('Power of Signal');
xlabel('Freq [Hz]');
hold on;
plot(faxis, pow_dA/4, 'r');
title('Power of Signal (red)');
hold off;
```

Measure for Cross-Frequency Coupling

```
%% Compute CFC of signal

load('data_1.mat');
% plot data and check for CFC by eye
t = (1:length(d))*dt; % time axis
N = length(t);
Fs = 1/dt; % sampling freq
plot(t, d);  xlabel('Time [s]');
periodogram(d, [], [], 1/dt);
```

```
% From this figure, there are two peaks:
% (i) sharp low freq peak near 6 Hz
% (ii) broad high freq peak from 60-140 Hz
% Note: high freq activity has much lower power than low freq activity
% Goal: develop measure to quantify CFC
% Step 1: Filter data into low and high freq bands
% we will use second order Butterworth filter
deg = 2; % filter order
Wn = [5*2/Fs, 7*2/Fs]; % low freq window of interest
[B,A] = butter(deg, Wn, 'bandpass'); % apply filter to isolate band
dlo = filtfilt(B,A,d);
Wn = [60*2/Fs, 140*2/Fs]; % high freq window of interest
[B,A] = butter(deg, Wn, 'bandpass'); % apply filter to isolate band
dhi = filtfilt(B,A,d);
% How well does filter work?
plot(t,d);
hold on;
plot(t, dlo, 'r');
plot(t, dhi, 'g');
hold off;
xlabel('Time [s]');
title('Data (blue), Low-freq (red), High-freq (green)');
% Note: issues at edges of data, side-effect of filtering
% center data:
dlo = dlo(N/4: end-N/4-1);
dhi = dhi(N/4: end-N/4-1);
taxis = t(N/4: end-N/4-1);

% Step 2: Compute phase of low freq signal and amplitude envelope of high % freq
   signal
phi = angle(hilbert(dlo)); % phase of low freq signal
amp = abs(hilbert(dhi)); % amplitude env of high freq signal
figure;
subplot(2,1,1); plot(taxis, dlo); hold on;
plot(taxis, dlo); hold on;
plot(taxis, phi, 'g'); hold off;
axis tight xlabel('Time [s]'); title('Low freq and phase');

subplot(2,1,2); plot(taxis, dhi); hold on; plot(taxis, amp, 'r');
hold off; axis tight xlabel('Time [s]');
title('High freq and amplitude envelope');

% Step 3: Determine if phase and envelope are related. We will do this by %
   dividing the phase into binds, find the times where low freq phase lies % in
   each bin, and then compute the average amplitude envelope of the
% high freq signal at those times

p_bins = -pi: 0.2: pi;
a_mean = zeros(length(p_bins)-1,1); % vector to hold avg amp env results p_mean =
    zeros(length(p_bins)-1,1); % vector to hold center of phase bins
for k = 1:length(p_bins)-1
        pL = p_bins(k); % phase lower limit for this bin
        pR = p_bins(k+1); % phase upper limit for this bin
        indices = find(phi >= pL & phi < pR); % find phase values in this       %
            range
        a_mean(k) = mean(amp(indices)); % compute mean amplitude at these       %
            phases
```

```
        p_mean(k) = mean([pL, pR]); % label the phase bin with the center       %
            phase
end

h = max(a_mean) - min(a_mean); % diff bw max and min modulation

% plot the mean envelope vs phase
figure;  plot(p_mean, a_mean, 'k', 'LineWidth',1);
axis tight
xlabel('Low freq phase'); ylabel('High freq envelope height diff'); title(['
    Metric h=' num2str(h)]);
```

References

- Code examples written by Dr. Mark Kramer, Boston University
- Buzsáki, G., & Watson, B. O. (2012). Brain rhythms and neural syntax: implications for efficient coding of cognitive content and neuropsychiatric disease. Dialogues in clinical neuroscience, 14(4), 345.
- https://www.mcgill.ca/bic/files/bic/cross_frequency_coupling_soheilasamiee.pdf