Time-Frequency Analysis of Biophysical Time series

Tim Mullen
(with many contributions from Arnaud Delorme)

\[ S(f) = \frac{1}{N} \sum_{t=0}^{N-1} x(t)e^{-2\pi ift/N} \]
Common Oscillatory Modes in EEG

<table>
<thead>
<tr>
<th>Simulated</th>
<th>Real</th>
</tr>
</thead>
<tbody>
<tr>
<td>30-60 Hz Gamma</td>
<td></td>
</tr>
<tr>
<td>18-21 Hz Beta</td>
<td></td>
</tr>
<tr>
<td>9-11 Hz Alpha</td>
<td></td>
</tr>
<tr>
<td>4-7 Hz Theta</td>
<td></td>
</tr>
<tr>
<td>0.5-2 Hz Delta</td>
<td></td>
</tr>
</tbody>
</table>

1 second
Wide-sense stationary signals
Wide-sense stationary signals

By looking at the Power spectrum of the signal we can recognize three frequency Components (at 2, 10, 20Hz respectively).
Fourier’s Theorem

Any stationary, continuous process can be exactly described by an infinite sum of sinusoids of different amplitudes and phases.

Jean Baptiste Joseph Fourier (1768 –1830)
Fourier’s Theorem

Time domain

\[ x(t) \]

Frequency domain

\[ |S(f)| \]

Frequency

\[ f \]

Time

\[ t \]

Freq. decomp.

Sum of freq.

Figure, courtesy of Ravi Ramamoorthi & Wolberg
Aliasing and the Nyquist Frequency

Signal: $f = 100 \text{ Hz}$

Sampling: $f_s = 80 \text{ Hz}$

Alias: $f_{\text{alias}}(N) = 20 \text{ Hz}$

Nyquist Frequency:

The maximum frequency that can be uniquely recovered at a sampling rate of $f_s$

$$f_N = \frac{f_s}{2}$$

$$f_{\text{alias}}(N) = \left| f + N f_s \right|$$

$f_s = \text{sampling rate}$

Quiz: What is $N$ in the example above?
Euler’s Formula

\[ e^{i(\omega t + \theta)} = \cos(\omega t + \theta) + i \sin(\omega t + \theta) \]

\[ A \cdot \cos(\omega t + \theta) = \frac{A}{2} e^{i(\omega t + \theta)} + \frac{A}{2} e^{-i(\omega t + \theta)} \]

\[ = \text{Re}\{A e^{i(\omega t + \theta)}\} = \text{Re}\{S(\omega, t)\} \]

\[ \theta = \angle S(\omega, t) \]

\[ = \pi / 2 \]

Any sinusoid can be expressed as the sum of two complex numbers.

Phase shift: \( \theta = \pi / 2 \)
Angular frequency: \( \omega = 2\pi f = 2\pi \text{ rad/sec} \)

Real part: Cosine component
Imaginary part: Sine component

Phasor: \( |S(\omega, t)| = |A| \)
Phasors

Rotation velocity (Rad/S; Hz) = (angular) frequency (\(w; f\))

\[ A \cdot \cos(\omega t + \theta) = \text{Re}\{A e^{i(\omega t + \theta)}\} = \text{Re}\{S(\omega, t)\} \]

\[ |S(\omega, t)| = |A| \]

Shorthand phasor notation: \(A e^{i\phi}\)
Phasors: Example
Phasors: Example

\[ 3 \]

\[ \frac{\pi}{4} \]

\[ 0.7 \]
**Discrete Fourier Transform**

Forward transform:

\[ S(f) = \frac{1}{N} \sum_{t=0}^{N-1} x(t)e^{-2\pi ift/N} \]

Frequency \( \rightarrow \) Time:

\[ x(t) = \frac{1}{N} \sum_{f=0}^{N-1} S(f)e^{2\pi ift/N} \]

N = number of samples

Fast Fourier Transform (FFT)
Zero-padding

The DFT/FFT of a sequence of length $N$ produces power estimates at $N$ frequencies evenly distributed between 0 and the sampling rate ($F_s$) (or \(\lfloor N/2+1 \rfloor\) frequencies between 0 and the Nyquist rate, $F_n=F_s/2$).

Padding the signal with $Q$ zeros achieves the following:

1) Allows enforcement of signal length as a power of two enabling FFT
2) Increases the number of frequency bins between 0 and $F_s$ from $N$ to $N+Q$ (intermediate points are sinc interpolates)

Zero-padding does not increase frequency resolution (number of independent degrees of freedom)
Tapering

Fourier’s Theorem lets us exactly describe any continuous, stationary signal using an infinite weighted sum of sinusoids. Discontinuous functions can be infinitely approximated, but the approximation may be poor for finite sampling.

Gibbs Phenomenon

“Rippling” effect due to discontinuities in signal (e.g. edges of the truncated signal)

- Infinite number of frequencies required to approximate discontinuities
- This means infinite (or very large) number of samples required (not possible)

What can we do?

- Cesaro summation
Tapering

Smoothly decay signal to zero at endpoints to smooth discontinuity

Tapered EEG

Taper
Tapering reduces the effect of the Gibbs phenomenon making it easier to identify “true” peaks in the spectrum from spurious ripple peaks (minimized broadband bias or “spectral leakage”).

The cost is increased width of central peak (narrowband bias).
Spectral Estimation via Welch’s Method

Given $K$ windows:
- Variance (bias) is reduced by a factor of $K$
- Frequency resolution also reduced by a factor of $K$

Tapering also results in data loss $\rightarrow$ decreased frequency resolution (narrowband bias)
- Can we mitigate data loss?
Spectral Estimation via Welch’s Method

Overlap-add method results in reduced data loss due to tapering, while preserving spectral variance reduction 😊

\[ S_{Welch}(f) = \frac{1}{K} \sum_{k=1}^{K} |S_k(f)| \]
Averaging spectra over $n$ independent trials leads to further reduction of variance by a factor of $n$.

\[
\frac{1}{n} \sum_{k=1}^{n} |S_k(f)|^2
\]
Non-Stationary Signals

Non-stationary signals include bursts, chirps, evoked potentials, …
Spectrogram or ERSP

FFT
Spectrogram or ERSP

Average of squared values
Power spectrum and event-related spectral (perturbation)

\[ ERS(f, t) = \frac{1}{n} \sum_{k=1}^{n} |S_k(f, t)|^2 \]

Scaled to dB \(10\log_{10}\)

Here, there are \(n\) trials
Each trial is time-locked to the same event (hence “event-related” spectrum)
The ERS is the average power across event-locked trials
Absolute versus relative power

Absolute = ERS

Relative = ERSP (dB or %)

To compute the ERSP, we just subtract the pre-stimulus ERS from the whole trial.
A signal cannot be localized arbitrarily well both in time/position and in frequency/momentum.

There exists a lower bound to the Heisenberg product:

$$\Delta t \Delta f \geq \frac{1}{(4\pi)}$$

or

$$\Delta f \geq \frac{1}{(4\pi \Delta t)}$$

e.g. here are two possible ($\Delta f$, $\Delta t$) pairs:

$\Delta f = 1$Hz, $\Delta t = 80$ msec or
$\Delta f = 2$Hz, $\Delta t = 40$ msec

Note: $\Delta f$ means “difference between successive frequencies” or the inverse of the frequency resolution. Ditto for $\Delta t$.
Time-Frequency Tradeoff

Natural biophysical processes may exhibit sustained changes in narrowband low-frequency oscillations along with rapidly-changing (e.g. “burst”) high-frequency oscillations.

The Short-Time Fourier Transform has a constant temporal resolution for all frequencies.

Can we adapt the time-frequency resolution tradeoff for individual frequencies to improve spectral estimation?

Yes, we can!
We estimate the time-varying power at 10 Hz by convolving EEG signal with a tapered 10 Hz complex sinusoid (Morlet wavelet).
By convolving stretched and scaled versions of the “mother” wavelet with the EEG signal, we determine the time-frequency distribution of power.
Some Wavelet Families

Morlet

Daubechies_4

Daubechies_20

Coiflet_3

Haar_4

Symmlet_4

Meyer_2

Battle_3
Trading Frequency for Time
(and vice versa)

Wide window (temporally diffuse)

10 Hz

10 Hz
hi corr

8.5 Hz
low corr

11.5 Hz
low corr

High Freq. resolution
Low Time Resolution

Narrow window (temporally compact)

10 Hz

10 Hz
hi corr

8.5 Hz
hi corr

11.5 Hz
hi corr

Low Freq. resolution
High Time Resolution
FFT versus Wavelets

Fixed time res
No freq res

Fixed freq res
No time res

Fixed time res
Fixed freq res

Equal time and freq resolution

Variable time res
Variable freq res

Equal time and freq resolution

Adapted from http://www.cerm.unifi.it/EUcourse2001/Gunther_lecturenotes.pdf, p.10
FFT versus Wavelets

FFT (Wavelet(0))

Wavelet (1)
Wavelet scale

Wavelet (0)

1Hz
2Hz
4Hz
6Hz
8Hz
10Hz

Wavelet (1)

scale expansion factor (q)

constant window size (time resolution) for increasing frequency → increasing # cycles with frequency.

window size decreases by a factor of 2 for each octave (power of 2) → constant # of cycles at each frequency.
Wavelet scale expansion factor

Larger expansion factor produces larger scale decrements (increased time resolution, lower frequency resolution) for increasing frequency.

Number of cycles at highest frequency for an expansion factor of $q$:

$$C_{f_{\text{max}}} = \frac{f_{\text{max}}}{f_{\text{min}}} C_{f_{\text{min}}} (1 - q)$$
Inter-Trial Coherence (ITC)


Trial 1
Trial 2
Trial 3

POWER = mean(amplitudes^2)
0.44 or –8.3 dB

ITC = |mean(norm’d phase vectors)|
Magnitude: 0.33

amplitude 0.5  phase 0
amplitude 1  phase 90
amplitude 0.25  phase 180

>> EEG = pop_newtimef(EEG,..., 'plotitc','on');
Time-Frequency Analysis of Biophysical Time series:

Practicum
Plot periodogram (spectrum) using Welch’s method

‘winsize’, 256 (change FFT window length)
‘nfft’, 256 (change FFT padding)
‘overlap’, 128 (change window overlap)
Pure green denotes non-significant points.
Increase
# freq bins

padratio = 1

padratio = 2

Component number
Sub epoch time limits [min max] (msec)
Frequency limits [min max] (Hz) or sequence
Baseline limits [min max] (msec) (0->pre-stim.)
Wavelet cycles [min max/fact] or sequence
ERSP color limits [max] (min--max)
ITC color limits [max]
Bootstrap significance level (Ex: 0.01 -> 1%)
Optional newtime() arguments (see Help)
Shows the actual dominant phase of the signal.
To visualize both low and high frequencies

```matlab
freqs = exp(linspace(log(1.5), log(100), 65));
cycles = [linspace(1, 8, 47) ones(1,18)*8];
```
Component time-frequency
Exercise

• **ALL**
  Start EEGLAB, from the menu:
  load <eeglab_root>/sample_data/eeglab_data_epochs_ica.set
  or your own data

• **Novice**
  From the GUI, Plot spectral decomposition with 100% data and 50% overlap (`overlap`). Try reducing window length (`winsize`) and FFT length (`nfft`).

• **Intermediate**
  Same as novice but using a command line call to the `pop_spectopo()` function. Use GUI then history to see a standard call (“eegh”).

• **Advanced**
  Same as novice but using a command line call to the `spectopo()` function.
Exercise - newtimef

• **Novice**
  From the GUI, pick an interesting IC and plot component ERSP. Try changing parameters window size, number of wavelet cycles, padratio,

• **Intermediate**
  From the command line, use newtimef() to tailor your time/frequency output to your liking. Look up the help to try not to remove the baseline, change baseline length and plot in log scale. Enter custom frequencies and cycles (2 slides back).

• **Advanced**
  Compare FFT, the different wavelet methods (see help), and multi-taper methods (use timef function not newtimef). Enter custom frequencies and cycles. Look up newtimef help to compare conditions. Visualize single-trial time-frequency power using erpimage.