Analyzing Neural Time Series Data

1.1 What is Cognitive Electrophysiology?

1.2 What is the purpose of this book?

- Gain a deeper understanding of data analysis, without requiring formal training in math and computer science.

1.3 Why You Shouldn't Use EEG Analysis Packages

- Lack of flexibility

1.4 Why Program Analysis, and Why in Matlab?

- Easy to use
- Matlab has many EEG analysis toolboxes
- Easily Sharable with others
- **Octave –** Free Matlab alternative
- Matlab Toolboxes for EEG:
    - Signal-processing Toolbox
    - Statistics Toolbox
    - Image-processing Toolbox

1.5 How to Best Learn from and Use This Book

- In front of a computer, running matlab
- In order from simple to more complicated analysis

1.6 Sample Data and Online Code

- www.mikexcohen.com/book

1.7 Terminology in this Book

- Always refer to as EEG, but also applies to other data collection techniques

1.8 Exercises

- Do them

1.9 Is Everything There is to Know about EEG Analysis in this Book?

- No, but it has the most useful, promising, and accepted approaches for linking EEG dynamics to cognitive processes.

2. Advantages and Limitations of Time- and Time-Frequency-Domain Analysis

2.1 Why EEG?

- Captures cognitive dynamics in the time frame the cognition occurs
    - Theta-band (4-8Hz)
        - Memory / Cognitive Control
        - Slower frequency
    - Gamma-band (30-80Hz)
        - Faster frequency
- Measures neural activity
    - Voltage fluctuations
- Multi-Dimentional
    - Voltage changes over time and space
        - Time
        - Space
        - Frequency
        - Power (strength of frequency-band-specific-activity)
        - Phase (timing of activity)

2.2 Why Not EEG?

- Not suited for studies where precise functional localization is important
- Not suited for testing hypotheses of deep brain structures
- Not suited for questions concerning slow processes with uncertain and variable time courses.

2.3 Interpreting Voltage Values from the EEG Signal

- Recorded in microvolts
    - Change in measured electrical potential between the electrode and a reference electrode placed elsewhere on the head
        - Readings change based on choice of reference and time period for baseline subtraction
        - Values differ across subjects because of

        - Skull shape / thickness
        - Scalp preparation
        - Orientation of dipole in brain
        - Cortical folding
        - If subject washed their had the morning of
    - Raw Values are difficult to compare and should not be overinterpreted
    - Interpret general pattern of effects and time-frequency-electrode characteristics of effects, rather than difference in microvolt values
- Analysis using scale transformations are advantageous
    - Individual differences in raw voltage values are eliminated

2.4 Advantages of Event-Related Potentials

1. Simple, fast to compute, require few analysis assumptions
2. High temporal precision and accuracy
    a. Applying low / high pass filters decreases precision
3. Extensive literature of ERP findings
4. Quick and useful data quality check of single subject data

Should be inspected for each subject to make sure data was properly collected.

2.5 Limitations of ERPs

1. Null results – many EEG data dynamics that aren't represented in ERP
2. Provide limited opportunities for linking results to physiological mechanisms
    a. Mechanisms that produce ERPs are less well understood than those producing oscillations

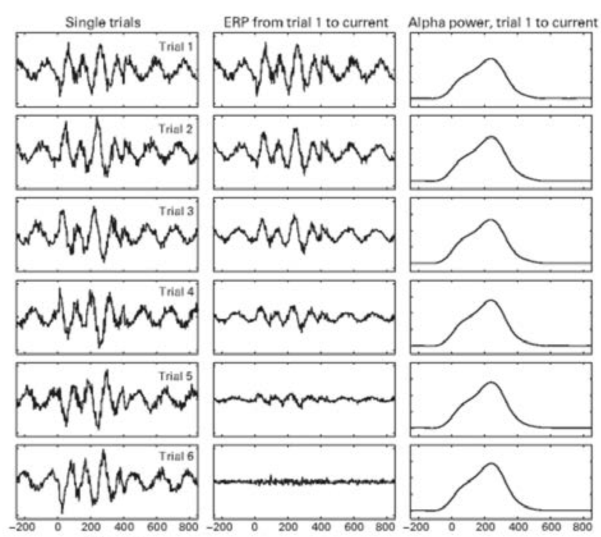


Figure 2.1
Simulated data showing how time-locked but not phase-locked activity (left column) is lost in ERP averaging (middle column) but is visible in band-specific power (right column). Each row in the left column shows a different trial, and each row in the middle and right columns shows averages from the first until the current trial.

2.6 Advantages of Time-Frequency-Based Approaches

1. Results can be interpreted in terms of neurophysiological mechanisms of neural oscillations
2. Oscillations are the most promising bridge that links findings from multiple disciplines within neuroscience and across multiple species
3. Many task-relavent dynamics that are retrievable using only time-frequency-based approaches

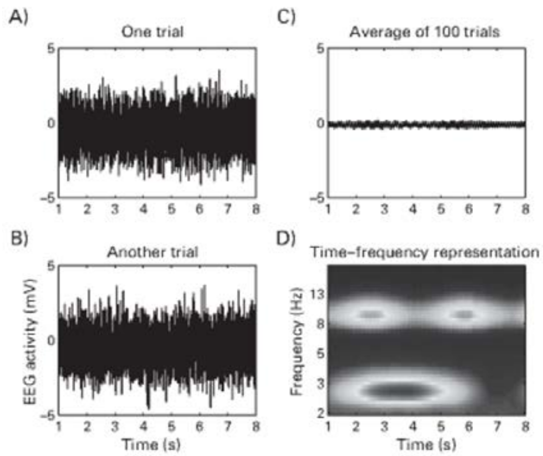


Figure 2.2
Simulated data showing that complex and multifrequency information contained in EEG data may have no representation in the ERP, if that information is non-phase-locked. One hundred trials were simulated; panels A and B show example trials. Panel C shows the ERP of those 100 trials, and panel D shows the time-frequency power. (This figure is adapted from Cohen 2011b).

2.7 Limitations of Time-Frequency-Based Approaches

1. Decrease of temporal precision → from time-frequency decomposition
    a. Lower frequency = more loss of temporal precision
    b. Low pass filtering of ERP data diminishes temporal precision
2. Large number of analyses that can be applied to EEG data
    a. Complexity of those analyses is intimidating

2.8 Temporal Resolution, Precision, and Accuracy of EEG

- Resolution → # of data samples per unit time
- Precision → certainty of measurement at each time point
- Accuracy → Relationship between timing of EEG signal and timing of biophysical events that lead to EEG signal
    - Distance of the dots to the center of the bull's eyes

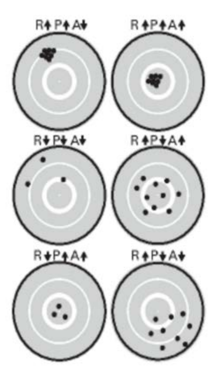


Figure 2.4
Bull's-eye illustration of the differences among resolution (R), precision (P), and accuracy (A). Up-and-down arrows indicate high and low levels. Resolution is illustrated by the number of dots, precision is illustrated by the spread of the dots, and accuracy is illustrated by the distance of the dots away from the center of the bull's-eye.

- Temporal resolution (Generally 250Hz – 1000Hz)
  - Determined by sampling rate of acquisition (100s – 1000s of samples per second)
  - Allows for extraction of frequency-band-specific information
  - Low temporal resolution
    - Extracting Delta-Band Power
  - High temporal resolution
    - Cross-frequency Coupling
- Temporal Precision
  - Depends on applied analysis / Selected Parameters / Frequency Band
    - Higher frequency (generally) = higher temporal precision
  - High Precision
    - Unfiltered ERPs
  - Low Precision
    - 1Hz bandpass-filtered activity
- When temporal precision is decreased by analysis, temporal resolution can decrease to match precision
- Temporal Accuracy – Is extremely high because
  - Brain electrical activity travels instantaneously from neurons to electrodes

2.9 Spatial Resolution, Precision, and Accuracy of EEG

- Spatial Resolution
  - Depends on # of electrodes
- Spatial Precision
  - Low, but can be improved by spatial filtering
    - Surface Laplacian / Adaptive source-space-imaging techniques
- Spatial Accuracy
  - Low
  - Electrodes collect data on cluster of neurons
- Organization of Brain networks
  - Microscoping Scale → less than a few cubic millimeters
  - Mesoscoping Scale → patches of cortex of several cubic (mm – cm)
  - Macroscopic Scale → large regions of cortex (many cubic cm)

2.10 Topographical Localization vs. Brain Localization

- Topographical → identify electrodes that show max effect
  - Description of observation
- Brain → identify regions in brain that generate activity (measured on scalp)
  - Interpretation of a result, supported by combination of theory, previous research, and data analysis in combination with spatial filtering

2.11 EEG or MEG?

- MEG → better at detecting high-frequency activity
    - Also better for source localization
- EEG → works better for radial sources

2.12 Costs of EEG Research

- Good EEG headsets are expensive
- If you have no equipment, the cost of EEG per subject gets close to, and sometimes supersedes the cost of MEG or MRI research

Interpreting and Asking Questions about Time-Frequency Results

3.1 EEG Time Frequency: Basics

- Rhythmic activity reflects neural oscillations
    - Fluctuations in excitability of populations of neurons
- Frequency
    - Speed of oscillation (Hz) – cycles per second
- Power
    - Energy in a frequency band that is squared amplitude of oscillations
- Phase
    - Position along sine wave at any given time point (Radians or Degrees)
- Power and phase are independent of eachother

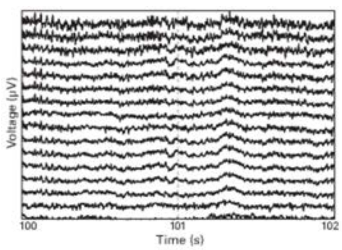


Figure 3.1
Raw EEG data (after 0.1-Hz high-pass filtering) showing oscillations at different speeds and for different lengths of time. Each line corresponds to an electrode.

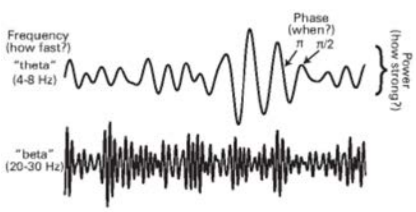


Figure 3.2
The three dimensions that define oscillations: frequency, power, and phase.

- EEG measures meso / macro – scopic cortical electrical activity
- Brain rhythms
    - Delta (2-4Hz)
    - Theta (4-8Hz)
    - Alpha (8-12Hz)
    - Beta (15-30Hz)
    - Lower Gamma (30-80Hz)
    - Upper Gamma (80-150Hz)
    - Subdelta (<600Hz)
- Generally: Better precision in time or frequency → Poorer precision on other domain

- Background activity exists
    - So we require baseline normalization
    - To remove artifacts that are consistently present, but not relevant to area of focus
    - Phase-locked
        - Phase is same or similar on each trial
    - Non-phase locked
        - Phase is different on each trial
- Spatial Autocorrection
    - Sometimes electrodes record activity from the same brain sources

3.2 Ways to View Time-Frequency Results

- Frequency Slice
    - Power (energy at each frequency band) vs. Frequency
    - No use of time
    - Useful when little time varying changes in frequency characteristics are expected
        - Ex. Resting State or Sleep Stage
- Time Slice
    - Select one frequency band
    - Plot its activity over time
    - Useful when comparing activity across multiple conditions or electrodes
        - And a prior reason to focus on a specific frequency
- Space Slice
    - One time-frequency point (or average over mult. Adjacent time freq. points)
    - Over electrodes on a topographical plot
    - Useful in visualizing topographical distribution of effect and facilities of topographical localization
- Time-Frequency Slice
    - Frequency vs. Time
    - Typically higher frequencies are plotted at the top
    - Color can be used to reflect:
        - Power
        - Phase clustering
        - Connectivity
        - Correlation Coefficient
- Other Slices
    - Activity of each electrode over time
    - Activity over different frequencies as a function of physical distance of subject to a target

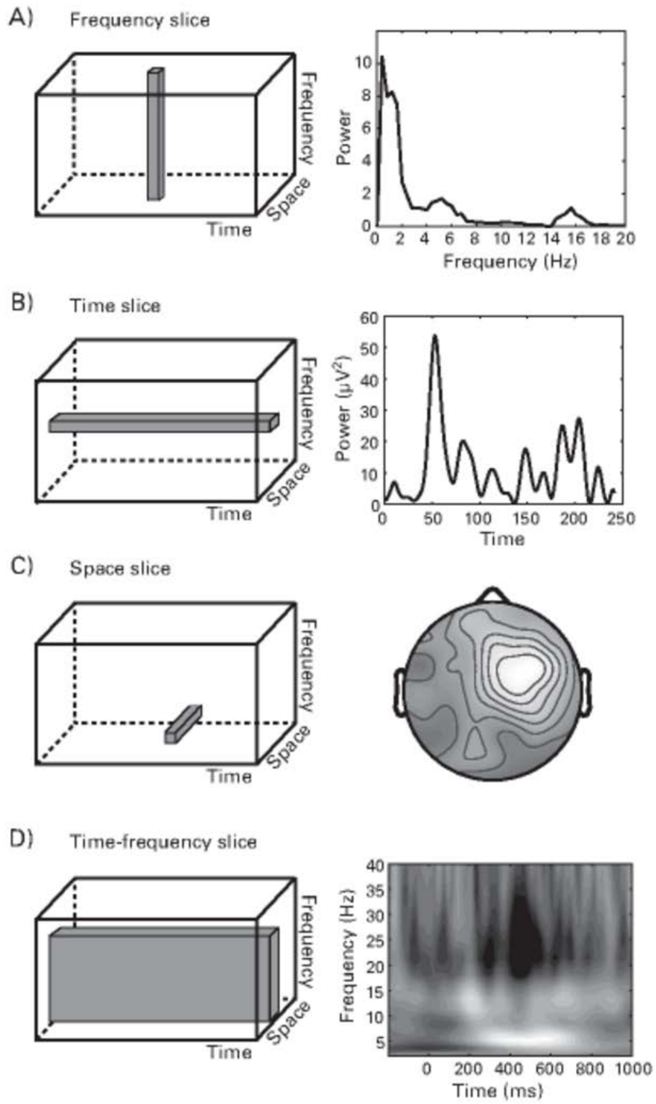


Figure 3.3
The data cube, containing information over time, frequency, and space, is difficult to view or conceptualize and therefore is sliced in different ways to illustrate 1-D or 2-D snapshots of the results.

3.3 Tfviewerx and erpviewerx

- Simultaneously shows time-frequency plot from one electrode and the topographical map at a selected time-frequency point

3.4 How to View and Interpret Time-Frequency Results

1. Determine what is shown in the plot
   - Power
   - Phase Clustering
   - Connectivity
   - Correlation with behavior
   - Understand conceptually what is being plotted
2. Inspect the ranges and limits of the plot
   - What are the time and frequency ranges?
   - Is there activity that is cutoff by boundaries?
   - Color limits? Symmetric or asymmetric or bounded by 0?
3. Inspect the Results
   - Activity at multiple freq. and time windows? Or all centered at one time-frequency?
   - Activity duration short or long? Freq. band-limited or spans multiple frequency bands?
   - Activity during prestimulus period?
   - Topographical specificity (are effects present selectively at some parts of scalp)?
   - Which electrode(s) are shown and why?
4. Link Results to Experiment
   - What does time = 0 refer to?
   - Mult. Events in experiment? How are they represented in time-freq. Results?
   - Results make sense? Are they consistent?
   - What do results suggest about cog. Process under investigation?
   - Results prove any new information about brain function?
5. Understand Statistical Procedures Used to Support the Interpretations
   - Statistical Threshold?
   - Hypothesis-driven or exploratory and data-driven?
   - Exploratory approaches generally require conservative statistical thresholds and corrections for multiple comparisons over time, frequency, and electrodes.
   - Hypothesis driven → increase sensitivity and theoretical relevance. Less stringent thresholds ($p < 0.05$ is acceptable)
   -If hypothesis driven, how were time-frequency-space windows selected for statistical analysis?

3.5 Things to Be Suspicious of When Viewing Time-Frequency Results

- Horizontal / Vertical stripes in Time-Frequency Plot
    - Ripple artifacts from poor filter construction
        - Filter widths are:
            - Too narrow
            - Applied to too little data
- Brief and large-power effects at high frequencies
    - EEG artifacts such as:
        - Amplifier saturation
        - Noise spike from bad electrode
- Broadband effects
    - Mechanical noise or excessive muscle activity from jaw or neck
- Fast color changes over time or frequency
    - Mistake in analysis – real part of analytical signal plotted instead of power
    - Fast change in lower frequencies are more suspect than in high frequencies
        - Increased temporal smoothing at lower frequencies
- Strange topographical distributions
    - Noisy or bad electrodes
    - Incorrect mapping between electrode label and physical location
    - High-pass spatial filters (Laplacian) increase topographical localization and highlight local spatial features
- High-frequency activity (over 100Hz)
    - Has low signal-to-noise ratio and may require many trials and special analysis techniques to enhance signal noise
- Low-Frequency Activity (<1z)
    - High pass filters that attenuate activity in lower frequencies
    - Apply high pass filter of 0.1 or 0.5 Hz to eliminate slow fluctuations

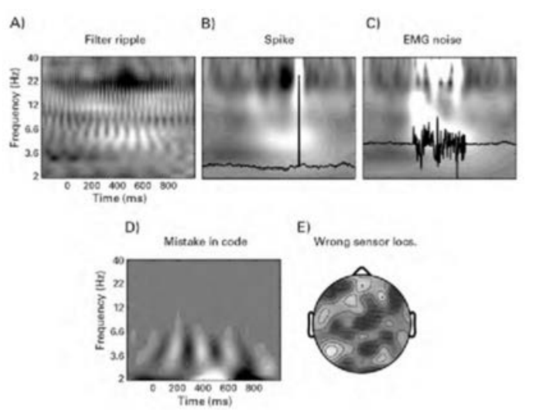


Figure 3.5
Some features of time-frequency results that should arouse suspicion, although they are not necessarily artifacts. In panels B and C, the offending single trial (out of 99 otherwise good-data trials) is superimposed on the time-frequency plot (EEG trace amplitude is arbitrarily scaled). The topographical map in panel E was produced by randomly swapping electrode label-location mappings.

3.6 Do Results in Time-Frequency Plots Men That There Were Neural Oscillations

- On one hand, EEG measures summed field potentials of populations of neurons
  - Strongly oscillatory
- On the other, Fourier’s theorem specifies that any signal can be represented using sine waves, and thus, even nonoscillatory signals have a representation in a time-frequency plot

4 Introduction to Matlab Programming

4.1 Write Clean and Efficient Code

- Clean code is easy to read and understand
- 1. Write Brief Comments before the code
- 2. Group lines of code by their common purpose
- 3. Use sensible, interpretable variable names
- Perform matrix manipulations instead of loops when possible
- Verbally plan out your code on paper before coding it

4.2 Using Meaningful File and Variable Names

- Put “I” at end of counting variables in loops

4.3 Make Regular Backups of Your Code and Keep Original Copies of Modified Code

4.4 Initialize Variables

- Reserve space in Matlab buffer by creating variable before populating it with data
  - Helps avoid memory crashes
  - Helps prefent data from previous iterations of loop contaminating current iterations
  - Helps you think about size, dimensions, and contents of large and important variables in advance

5 Introduction to the Physiological Basis of EEG

5.1 Biophysical Events That Are Measurable with EEG

- Magnetic fields are perpendicular to electric fields and pass through skull/scalp unimpeded

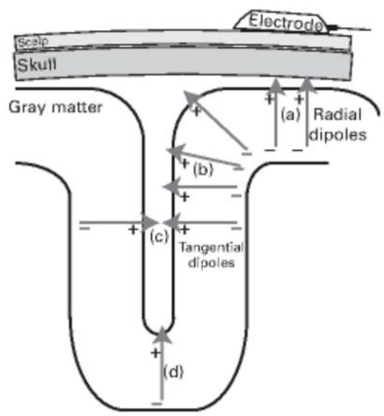


Figure 5.1
Illustration of dipoles in different orientations with respect to the skull. The dipoles illustrated in (a) will contribute the strongest signal to EEG, whereas the dipoles illustrated in (b) will contribute the strongest signal to MEG. The dipoles illustrated in (c) are unlikely to be measured because the dipoles on opposing sides of the sulcus produce electrical fields that are likely to cancel each other. The dipole illustrated in (d) will make a smaller contribution to EEG than dipole (a) because it is further away from the electrode. (This figure is inspired by figure 1 of Scherg 1990.)

- Populations of neurons in subcortical structures are not arranged in geometrically parallel orientation
  - o In synchronous population activity, electrical fields generated by individual neurons are likely to cancel eachother out at the macroscopic scale
- Slow fluctuations (<1Hz) are difficult to measure with EEG
  - o Most amplifiers have built-in high-pass filters that attenuate very slow fluctuations because they may cause amplifier saturations
    - ▪ There are DC-coupled amplifiers for fluctuations below 1Hz
- Fast fluctuations (>100Hz)
  - o High frequency activity generally has low power → difficult to distinguish from noise

5.2 Neurobiological Mechanisms of Oscillations

- Oscillation – Rhythmic alteration of states
  - o Rhythmic fluctuations in excitability of neuron populations
- Interaction between inhibitory interneurons and excitatory pyramidal cells
  - o Oscillations can be produced by excitatory or inhibitory neurons

5.3 Phase-Locked, Time-Locked, Task-Related

- Phase-locked
  - Aligned with time = 0
  - Observed in time-domain averaging and in time-frequency-domain averaging
- Non-Phase-Locked
  - Time locked, but not phase locked with time = 0
  - Time-frequency-domain averaging
- Time and/or frequency characteristics change as a function of engagement in task events
  - Background activity does not
  - Apply baseline normalization to remove background activity

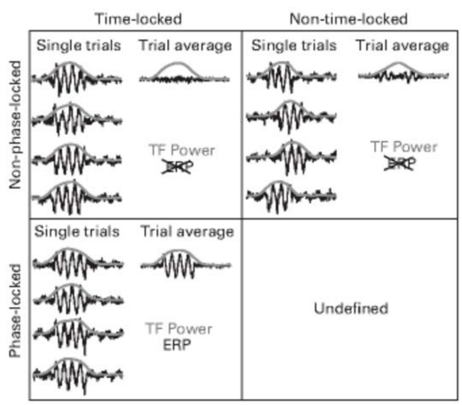


Figure 5.2
Illustration of whether time-frequency (TF) power and the ERP can measure phase-locked, non-phase-locked, time-locked, and non-time-locked activity. The left column of each cell shows four trials of simulated data, and the right column of each box shows the average of those four trials. Black lines show the raw time series, and gray lines show the time course of 10-Hz power. The ERP captures only phase-locked and time-locked activity. Time-frequency power can measure time-locked activity regardless of whether it is phase-locked or non-phase-locked. Activity that is not time-locked can be measured with time-frequency power, although the results will be smoothed and thus less temporally precise.

5.4 Neurophysiological Mechanisms of ERPs

- Additive
  - ERP reflects a signal elicited by an external stimulus (picture / sound)
  - Or internal event (manual response)
- Phase reset
  - ERP results from sudden alignment of phases of ongoing oscillations
  - Stimulus appears → ongoing oscillation at a particular frequency band is reset to a specific phase value
- Amplitude Asymmetry or Baseline Shift
  - Outward-going currents are less detectable from scalp
    - Producing asymmetry in oscillations measured by EEG
      - Unequally distributed peaks / troughs
  - Changes in overall power could produce asymmetries in ongoing oscillations

5.5 Are Electric Fields Causally Involved in Cognition?

- Theory 1: Long-term potentiation occurs at theta-band oscillations
- Theory 2: Timing of many neurons is constrained by local field potential
- Theory 3: Interregional oscillatory synchronization is a mechanism underlying the transmission of information across neural networks, and the synchronization-mediated connectivity is crucial for perceptual and cognitive processes

5.6 What if Electric Fields Are Not Causally Involved in Cognition?

- Doesn’t stop progress

6 Practicalities of EEG Measurement and Experiential Design

6.1 Designing Experiments: Discuss, Pilot, Discuss, Pilot

- Discuss with colleagues before data collection
- Test run without EEG headset
- Perform on 2 people and fully analyze those datasets

6.2 Event Markers

- Square-wave pulses sent from stimulus-delivering computer to EEG amplifier
- Recorded as separate channel in raw data file
    - Amplitude used to encode specific events (stimulus onset or response)
    - During data importing, markers are converted to labeled time stamps
        - Indicate when different events occurred
- Used to time lock EEG data
- Used to reconstruct different conditions and responsebetters
    - **Better more detail in these markers, More =**

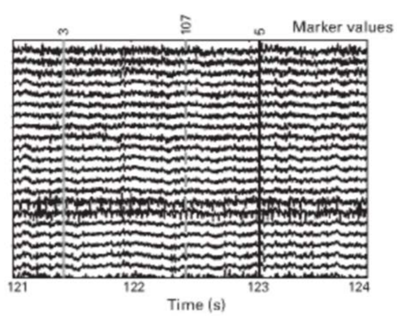


Figure 6.1
Example EEG data showing 3 s of data and three experiment markers. The experiment markers are represented as vertical lines, and the numbers on top of the vertical lines correspond to particular events. In this case the numbers 3 and 5 refer to two response buttons being pressed by the subject, and the number 107 corresponds to a particular stimulus. This picture was made using the eeglab lab function `eegplot`.

- Check for overlapping and dropped markers
- Temporal duration
    - Time when marker has non-zero value
    - (Should be at least a few samples ~5 ms)
- Data is useless without markers
    - Test by:
    - Sending codes 1-256 with 10ms spaces between markers

6.3 Intera- and Intertrial Timing

- Space out time between tasks by a few seconds (Intertrial Interval ~1000ms)
  - Allow brain response to subside after a task
  - Different tasks will take different times to subside
    - Ex. Pictures that evoke emotion
- Period of time for baseline normalization of task related data?
- What frequencies to analyze.
- Time Freq. Decomposition
  - End before trial onset (-500 to -200ms)
    - Bc. Temporal filtering may cause early poststimulus activity to leak into the prestimulus baseline period
- ERPs
  - End at time = 0
- Subjects nearly always generate temporal expectations about when the next trial will occur
- Constant Time Intervals
  - People can mentally prepare for an upcoming event
- Random Time Intervals
  - People can try to guess when the next event is upcoming

6.4 How Many Trials You Will Need

- Depends on:
  - Signal-to-noise ratio (how clean vs. how noisy) of the data
  - Size of the effect
  - Type of analysis to be performed
- Usually minimum of:
  - 50 trials per condition per subject
  - But there are unique cases where less is needed

6.5 How Many Electrodes You Will Need

- Depends on Experiment and what you're looking for:
  - Brain Source Reconstrucion Analysis
    - 100+ electrodes
  - Measure P3 Amplitude
    - 3 electrodes
      - Central parietal cortex for P3
      - Reference
      - Ground
  - At least 64 usually
    - Consider time to prepare the subject with an EEG cap
    - Storage and processing capacity required for more data

6.6 Which Sampling Rate to Use When Recording Data

- Times per second the data are acquired
- Defines data's temporal resolution
- Depends on:
  - Type of analysis
  - Frequencies to analyze
  - Available desk space
  - Processor Speed
- Nyquist Theorem
  - Only frequencies below half the sampling rate can be recovered
    - Looking at 50Hz, need to sample 100Hz
- Generally use (500Hz to 2000Hz) sampling rate
  - Better higher sampling rate and then downsizing
- Make it easy to convert between time in samples and time in ms
  - 1000Hz is optimal
    - 14ms is 14 samples

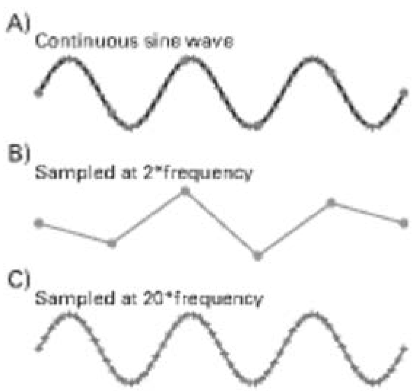


Figure 6.2
A continuous sine wave (panel A) and an illustration of the effect of subsampling that sine wave. Panel B shows that sampling the sine wave at twice its frequency (see gray dots along the sine wave in panel A) can reconstruct some features of the sine wave but fails to reconstruct the finer features, in particular the precise peak and trough times and the ongoing phases. Panel C shows that sampling at 20 times the frequency (see gray plus signs in panel A) can reconstruct the time-varying features of the sine wave with much higher accuracy.

6.7 Other Optional Equipment to Consider

- Response EMG or Force Grips
  - Provide data on muscular movement
- Eyetracker
  - Lets you remove trials where subjects looked away from the fixation spot
  - Use saccades and looking times as dependent measures
  - Remove oculomotor artifacts from EEG data
  - Changes in pupil dilation

- Electrode Localization Equipment
    - Have the precise location of the electrodes on the dead
- Comfortable Chair for Subject to Sit In
    - More comfort = less movement artifacts
- Good Response Device
    - Response device with good timing, comfortable, and intuitive
    - Easy to hold, intuitive layout
    - Not too easy to press (subjects must know that response was registered)
    - Not too hard to press (subject gets tired)
    - Clicks to show that a response was registered
        - Might also create artifacts

7 Preprocessing Steps Necessary and Useful for Advanced Data Analysis

7.1 What is Preprocessing?

- Any processing between collecting and analyzing the data
    - Organize data
        - Extract epochs from continuous data
    - Removing bad or artifact-ridden data w/ out changing clean data
        - Remove bad electrodes
        - Reject epochs with artifacts
    - Modifying Clean Data
        - Temporal filters
        - Spatial transformations
- Keep track of all details of preprocessing for each subject
    - Trials rejected
    - Electrodes interpolated
    - Independent components removed from the data

7.2 Balance Between Signal and Noise

- Signal and noise overlap, so it is hard to remove noise without removing signal too
- Noise example:
    - Amplifier Saturations (produce spikes)
- Threshold for noise depends on the experiment at hand

7.3 Creating Epochs

- Continuous data are cut into segments surrounding particular experimental events
- Must decide what to call "time = 0"
    - Options:
        - Time-lock to earliest event in each trial
        - Time-lock to the data of focus
- Must decide how much time before and after the "time = 0" event
    - Epoch must be at least as long as duration of trial
- Compute only ERPs
    - Epoch as long as time period to analyze + a baseline period
- Time-Frequency-Based Analysis
    - Longer epochs to avoid contaminating results with edge artifacts
        - Edge artifact:
            - Apply temporal filters to sharp edges → producing a high-amplitude broadband power artifact lasting hundreds of ms

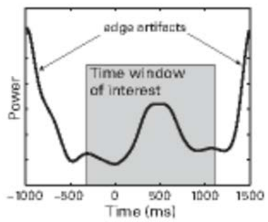


Figure 7.2
Edge artifacts resulting from discontinuous breaks in the time series between trials can contaminate the results if there are insufficient buffer zones to allow those edge artifacts to subside. In this case the edge artifacts are easily identifiable, and it is also clear that those artifacts subside before the time window of interest (gray area). In general, edge artifacts will contaminate up to three cycles of activity, but this could be less or more depending on the magnitude of the edges.

- The buffer zone you choose to include depends on frequencies you intend to extract
  - Longer epochs allow for edge artifacts to subside before and after experiment events
  - Lower frequency band to extract = more buffer zone to avoid edge artifacts
- Analyze one subject closely
  - Check if artifacts affect time period of interest
- General Rule:
  - 3 Cycles at the lowest frequency
  - Ex. 1500 ms for 2Hz activity
- Caveat: Large Epochs
  - Overlapping data in each epoch
  - A problem for Independent Component Analysis
    - Don’t expose ICA to the same data more than once
- Sufficient Buffer Required for:
  - Time-Frequency-Decomposition via:
    - Complex Morlet Wavelet Convolution
    - Filter-Hilbert Method
- Sufficient Buffer Not-Required:
  - Time-Frequency Decomposition via:
    - Short-Time FFT
    - Multitaper
- If analyzing data that’s already EEG epoched
  - Use Reflection – only when necessary
    - EEG data from each trial and electrode are reversed and put in beginning and end of trial
  - Makes Epoch 3 times longer
  - Discard reflected data after analysis
- NEVER – Taper the entire epoch time period

7.4 Matching Trial Count across Conditions

- Ideal for all conditions to have the same number of trials
- Analysis backed on phase:
  - More sensitive to trial count
  - Small # of trials introduces positive bias
- Analysis based on power or ERP
  - Less sensitive to trial count
  - Low trials ERP
    - Mean amplitude in time range is more robust to noise compared to peak times
- Large Differences in Trial Count Across Conditions
  - (less than 30 trials) – consider matching trial count across conditions
- Matching Trial Count
  - Identify condition with fewest trials
  - Selecting trials from other conditions
    - All conditions end up with equal # of trials
  - Method 1:
    - Select 1^{st} N trials from each condition
    - N is # of trials in smallest condition
    - Biases conditions to have more trials earlier in experiment (when subjects are less tired, more patience/motivated)
  - Method 2:
    - Select trials at random
    - No bias in terms of when trial occurred in experiment
    - But reanalyzes same data multiple times (unless store which trials were used already)
  - Method 3:
    - Select trials based on relevant behavioral or experimental variable (ex. Reaction time)
    - Select subset of trials from all conditions
      - Distributions of reaction times from the retained trials are similar across conditions
    - Disadvantage:
      - If there are reaction time differences between conditions, matching reaction times across conditions may bias trial selection from different regions of the reaction time trial distribution
    - Selective sampling based on any relevant behavioral measure:
      - Saccade speed
      - Pupil response
      - Subject difficulty rating
      - Luminance
      - Location of stimulus
- Trial matching when comparing EEG results across subjects with a behavioral variable that might be related to trial count

7.5 Filtering

- Notch filter at 50Hz or 60Hz
  - Attenuate electric line noise
- Not always necessary
  - Ex.
    - Don't need to low-pass filter time-domain data at 40Hz if perform time-frequency analysis to extract power from 2 – 20 Hz
- High Pass Filters
  - Only applied to continuous data (NOT EPOCH DATA)
    - Since edge artifacts of 0.5Hz may last up to 6 seconds (longer than epochs)
  - High-pass filter at 0.1 or 0.5Hz will minimize slow drifts

7.6 Trial Rejection

- Automatic rejection procedures
  - Fast
  - No user bias
  - Same trials rejected every time
  - Cons:
    - May use criteria only appropriate for some subjects but not others
- Time-frequency Decomposition
  - Sharp edges are more detrimental in TFD than ERPs
  - Small edges may not be picked up by automatic rejection procedures
    - Will have huge effect on TFD results

7.7 Spatial Filtering

- 1. Help to localize results
  - Ex. Confirm activity peak corresponds to left motor cortex
    - Surface Laplacian or fit a single dipole
- 2. Isolate topographical feature of data filtering low-spatial-frequency features
  - Ex. Visual stimuli that requires spatial attention
    - Difficult to separate visual processing (occipital cortex) from
    - Attention-related processing (parietal cortex)
    - Surface Laplacian (or distributed source imaging)
      - Minimize spatial overlap between brain regions
        - Increasing confidence in functional / anatomical distinctions
- 3. Preprocessing for connectivity analyses
  - Surface Laplacin (or distributed source imaging)
    - Minimize volume conduction (artifact contaminating connectivity analyses)
- When to use spatial filtering?
  - ERP on response preparation

        - Is ERP peak consistent with source in motor cortex?
            - Single dipole fit to grand-averaging ERP
        - Surface Laplacian to minimize volume conduction for connectivity analysis?
            - Laplacian applied to single-trial time-domain data before time-freq. decomposition
        - PCA on time-freq. power
            - PCA on single trials within subject
            - Or trial-averaged power across subjects

7.8 Referencing

- Only issue with EEG
    - Voltage values recorded from each electrode are relative to voltage values recorded elsewhere
    - Any activity in reference electrode is reflected as activity in all other electrodes
- Average mastoid (bone behind ear)
    - Reference electrode should not be close to brain region you expect to be activated
- Data can be re-referenced online because is a linear transformation
- Bipolar reference
    - One electrode is measured relative to another
    - Common measuring the eye

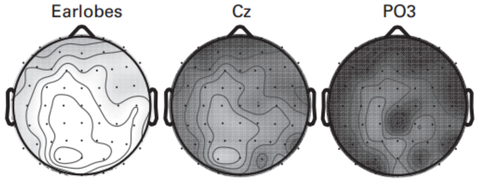


Figure 7.4
The effect of different reference electrodes on the same data. Earlobes refers to the average of electrodes placed on the two earlobes. In many situations, using one of the scalp electrodes as the reference is suboptimal.

7.9 Interpolating Bad Electrodes

- Data from missing electrodes are estimated based on the ativity and locations of other electrodes
    - Weighted Distance Metrics usually used include:
        - Nearest-neighbor
        - Linear
        - Spline

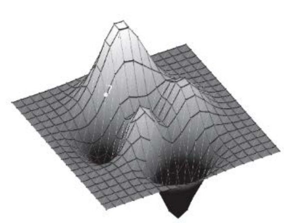


Figure 7.5
Topographical illustration of interpolation. Displayed is a smooth topographical landscape (analogous to a scalp-measured voltage) that is discretely sampled (black dots; analogous to electrodes). Interpolation involves estimating the activity at the white "electrode," given the activities and distances of all other electrodes. This topography was generated with the Matlab function.

- Interpolation is perfect weighted sum of all other electrodes
    - May cause problems with analysis requiring matrix inverses
- Another Option: Remove them altogether
- Interpolation important for spatial filters:
    - Surface Laplacin
    - Source reconstruction
    - Spherical Spline
    - Re-reference to average of all electrodes
        - "Activity of one bad electrode may contaminate signal of other electrodes"
- Do I interpolate?
    - 1. Inspect data
        - If a lot of noise, try filtering noise without interpolating
    - Can this data be salvaged?
        - Low pass filter of 30 Hz
        - If activity looks similar to surrounding electrodes, it can be salvaged. No interpolation.
        - If activity looks **different** (much smaller / larger magnitude) → unlikely real signals were recorded. Should be interpolated.

7.10 Start with Clean Data

- Noisy data is crap and won't give you anything at all

8 EEG Artifacts: Their Detection, Influence, and Removal

- Main EEG Artifacts
    - Blinks
    - Muscle movements
    - Amplifier Saturations
    - Line noise
    - Cognitive Artifacts

8.1 Removing Data Based on ICA

- Decomposes EEG time series into components
    - Components try to identify independent sources of variance
    - Set of weights for all electrodes so each component is a weighted sum of activity at all electrodes
        - Weights designed to isolate sources of brain activity
    - Analogy:
        - Cocktail party where everyone has a microphone. Tuning your system to dull the input of some microphones more than others to hear whom you want
- ICA
    - Clean EEG Data
        - Identify components that isolate artifacts
            - Artifacts found by:
                - Topographies
                - Time courses
                - Frequency Spectra
            - Take conservative stance when removing artifacts of noise
                - Don't want to be removing part of single in the process
                - Best case scenario, Only remove component regarding blink artifacts
    - Data Reduction
        - Analyze component time series (instead of electrode TS)
            - Max number of components = # of electrodes used
        - If Have 100+ Electrodes
            - Extract fewer components than electrodes
                - Faster analysis
                - Unlikely to be over 100 independent sources in the brain that are active AND can be statistically isolated with EEG

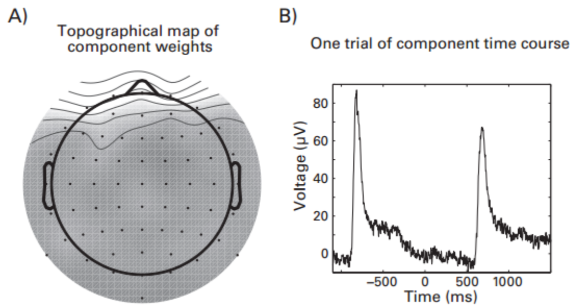


Figure 8.1
Example topographical map and example single-trial time course of an independent component that isolated a blink artifact. Panel A shows that the weights from this component are maximal at anterior electrodes, and panel B shows the time course of this component from one trial. You can see that on this trial, the subject blinked before and after the trial. The time course is a weighted sum of the activity of all electrodes, and the weights are defined by the results of the independent components analysis.

8.2 Removing Trials because of Blinks

- Blink artifacts linearly sum on top of the brain-generated EEG
- Removed with:
  - o ICA (Jung et al. 2000)
    - ▪ Appears to work better
  - o Regression-based Techniques (Graton, Coles, and DOnchin 1983)
- Other reasons to remove trials with blinks
  - o Multi-millisecond blinks can indicate subject is tired and cannot focus on task
  - o Poorly timed blink means subject didn't see visual stimulus
- Asking people to refrain from blinking
  - o Suppression of blinking is a demanding task (creates other signals)
    - ▪ Task-unrelated but stimulus-locked activity in frontoparietal oculomotor circuits
    - ▪ Distracted by inhibiting blinks and unfocused on stimuli
    - ▪ If subjects can blink during intertrial interval, the time period may not be useful
      - For baseline for normalization of time-frequency
- Best bet
  - o Ask subjects to avoid time-locking blinks with experimental events (pressing button)

8.3 Removing Trials because of Oculomotor Activity

- Saccades and microsaccades also contaminate data
    - Frontal and lateral frontal electrodes
    - Minimized by having centered fixation point on monitor
        - Tell subjects that eye movement will impact data
- Eye trackers or vert / horiz. EOG electodes

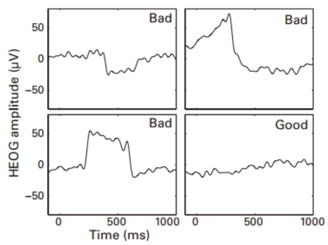


Figure 8.2
Horizontal EOG activity indicates eye movements after stimulus onset (time = 0). The first three panels are taken from trials that were removed prior to analyses.

- Microsaccades minimized by:
    - Small stimuli so subject does not need to saccade to see entire stimulus
    - Display on screen for short time
- Reference impact on data
    - Nose reference → Eye movement has larger influence over earlobe reference
- Concern of hypothesis
    - Anterior fontal or lateral frontal regions
        - EOG artifacts are a concern
    - Midcentral electrodes
        - EOG artifacts are less of a concern
- Isolate potential EOG artifacts
    - Spatial filtering techniques
- For tasks where subjects should have been fixating on point
    - Eye movement indicates not fully engaged in task

8.4 Removing Trials Based on EMG in EEG Channels

- EMG → bursts of 20- to 40-Hz activity
  - Maximal in electrodes around face, neck, ears
  - Deleterious for EEG data analyzed above 15Hz
- If EMG has constant amplitude before / after trial onset
  - EMG removed during baseline normalization
- IF EMG present in all conditions
  - Subtracted out during condition comparisons

8.5 Removing Trials Based on Task Performance

- Cognitive noise (artifacts)
- If trial responses can be accurate or not
  - Remove (or separate) error trials
- Also remove trials where:
  - Subjects don't make a response if they were instructed to do so
  - Trials with more responses than were required
  - Trials with fast reaction times
  - Trials with very slow reaction times
    - Slower than 3 standard deviations from each subject's median reaction time
  - If rest break lasts several tens of seconds, subjects not fully engaged on first trial after break
  - Subjects perform a few tens of trials with one set of instructions, then switch to another set of instructions

8.6 Removing Trials Based on Response Hand EMG

- EMG from muscles subjects use to indicate response
- Helps identify partial errors
  - Subject twitches muscle of incorrect response
  - Brain activity looks like that of an error even though response was correct
- Algorithm for partial error identification
  - Z-transform of derivative of EMG Signal from each hand
  - Rectified (absolute value)
  - **Partial Error When:**
    - Z-derivative signal of hand not used exceeds 2 standard deviations in the time between stimulus onset and actual button press
    - Mag. Of EMG peak must be >2x larger than largest EMG peak from -300ms to stimulus onset
- Record EMG from thumb muscles
  - Have response button that takes some effort to press
  - Requires more muscle engagement → bigger, cleaner EMG response

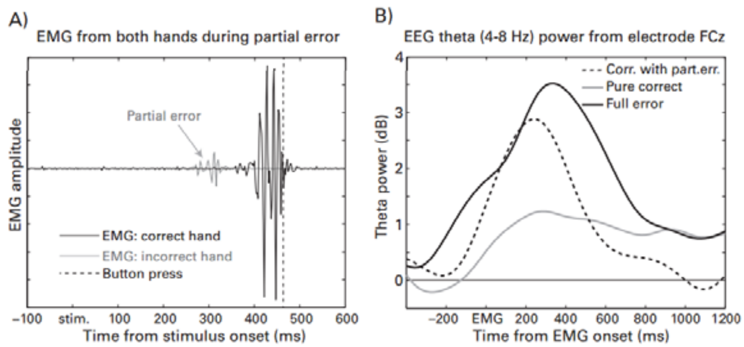


Figure 8.4
Partial errors can be detected with EMG recordings and are useful to identify correct trials with error-like brain responses. Panel A shows example EMGs from right and left thumbs showing a partial error—a muscle twitch of the incorrect hand although only the correct button was pressed in this trial. Correct trials that contain partial errors elicit error-like brain processes, as can be seen in panel B. Theta-band power from electrode FCz is time-locked to EMG onset; each line is the average of all trials from one subject. Partial errors can be identified and removed from the dataset or separately analyzed.

8.7 Train Subjects to Minimize Artifacts

- Show subject EEG data in real time on computer
- Explain: EEG Has brain activity and noise from muscles
- Have them: blink, clench jaw, tense neck/shoulder muscles, talk, smile, wiggle ears, etc.
    - When subject knows behaviors that produce artifacts, they can reduce them

8.8 Minimize Artifacts during Data Collection

- Keep an eye on real-time EEG data
    - Check every 30 seconds to see if they look ok
    - If data doesn’t look great, pause and do what you can to fix it

9 Overview of Time-Domain EEG Analysis

9.1 Event Related Potentials (ERPS)

- To create ERP
    - Align time-domain EEG to time = 0
    - Average across trials at each time point
    - Sum voltage at each time point across trials
        - Divide by number of trials
- ERP as quality inspection tool: ^^^ is all you need
- ERP to make inferences about cognitive processes
    - Learn about:
        - Component overlap

- Component quantification
- Appropriate Interpretation
- Statistical procedures

9.2 Filtering ERPs

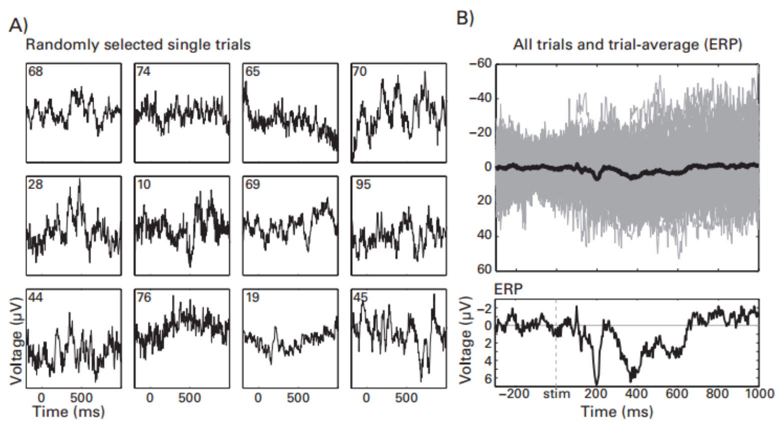


Figure 9.1
Panel A shows single-trial EEG traces from 12 randomly selected trials (number inside plot indicates trial number). Data are from electrode FCz. Panel B shows 99 single trials in gray and their average—the ERP—in black. Panel C shows the same ERP with focused *y*-axis scaling.

- Non-phase-locked activity doesn't survive time-domain averaging
    - Freq. above 15Hz usually not time locked
- Filtering ERPs:
    - Minimizes residual high frequency fluctuations
    - Makes ERPs look smoother
    - Facilitates peak-based component quantification
        - Reduces possibility that peak is a noise spike or non-representative outlier
- Poorly designed filters introduce ripples
    - Resulting from filters with narrow transition zones
- Applying low pass filter reduces temporal precision
    - Since voltage at each time point becomes a weighted average
    - Lower cutoff frequency of filter = more loss in temporal precision
- ERPs often filtered using frequency cutoff ~20 or 30Hz. Occasionally 5 or 10 Hz

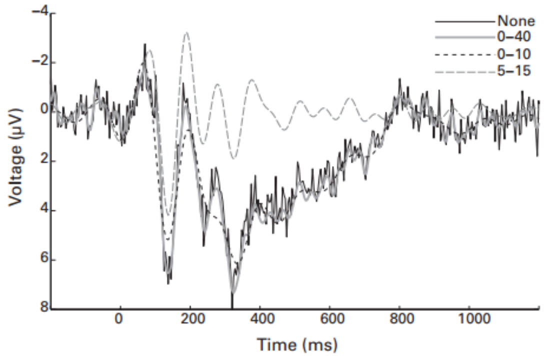


Figure 9.2
An ERP from electrode P7 with no filtering (black line) and with different filter settings (numbers in the legend indicate lower and upper frequency bounds in hertz). Note that some filter settings can have dramatic effects on the interpretation of specific ERP features. For example, the 5–15 Hz filter seems to have accentuated the first negative-going peak at around 100 ms and removed the later P3-type component, and the 0–10 Hz filter removed the negative-going peak at around 280 ms. The wide-band 0–40 Hz filter had the least effect on the larger ERP fluctuations while removing the high-frequency fluctuations. This plot is an illustration of why you should carefully consider the frequency range of the filter used for interpreting ERPs, particularly if you use a narrow frequency range.

9.3 Butterfly Plots and Global Field Power / Topographical Variance Plots

- ERP from all electrodes overlaid in same figure
- Global field power = standard deviation of activity over all electrodes at each point in time
- Useful as data quality indices + confirm timing of representations of task events in the data

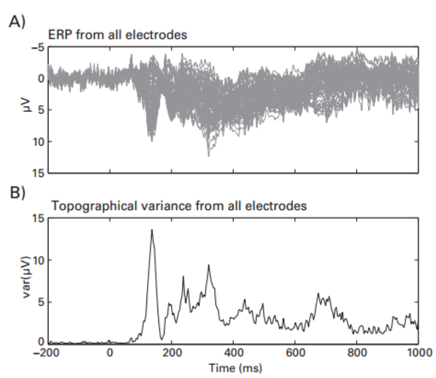


Figure 9.3
An example butterfly plot (panel A) and a topographical variance plot (panel B). Although they lack spatial information, these plots are useful for data inspection and provide an overview of the time periods with cortically diverse events, including approximately 180 ms, 220 ms, 320 ms, and 700 ms.

9.4 The Flicker Event

- Entrainment of brain activity to a rhythmic extrinsic driving factor
  - Ex. Looking at a strobe light flickering at 20 Hz → activity in visual cortex at 20Hz
- Also called:
  - Steady state evoked potential / Frequency Tagging / SSVEP / SSAEP
- Flicker effect allows you to “fake” high spatial resolution

- However, has poor temporal precision
- Stimulus flicker frequencies up to 100Hz can evoke a flicker effect
    - Lower frequencies generally elicit a stronger flicker effect
    - Size, luminance, and contract impact signal – to – noise ratio
- Generally: peak in frequency domain at flicker frequency should be readily observed in plot
    - Magnitude of frequency peak can be compared to same frequency before flickering stimulus began
    - Or compared to power of neighboring frequencies for which there was no flicker

9.5 Topographical Maps

- Show spatial distribution of EEG results

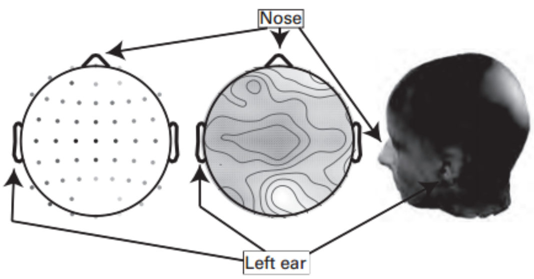


Figure 9.4
The same data shown by coloring dots on a topographical map (left) or interpolating those values over a surface (middle and right). Clearly, the interpolated maps are easier to interpret. Two-dimensional plots show data from all or most electrodes; 3-D plots (right) look more realistic but obscure data from part of the scalp. Locations of the nose and left ear are indicated to facilitate orientation and comparison.

- 2-D plots are less intuitive to interpret
    - Show activity simultaneously from all (or most) electrodes
- 3-D plots show activity from only a third of the head
    - Easier to interpret and look nicer in figures
- Allow you to:
    - Confirm timing of task events
    - Detect bad or noisy electrodes

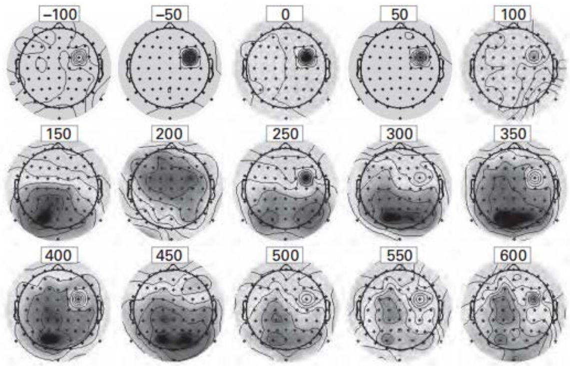


Figure 9.5 (plate 1)
Plotting topographical maps over time facilitates rapid data quality inspection. The numbers in white boxes indicate the latency at which the topographical data are plotted (in milliseconds) with respect to trial onset. These plots show, among other things, that there is one bad electrode. In this case the bad electrode was generated by replacing the true EEG activity at electrode FC4 with randomly generated numbers.

9.6 Microstates

- Temporal difference (global map dissimilarity) remains low for a period of time of tens to hundreds of milliseconds, then suddenly becomes relatively large
  - Sharp increase is a shift of microstates
- Stable maps used in hierarchical clustering analysis to identify small number of topographical maps during periods of stability (cluster maps)
  - Topography at each time point is labeled according to cluster map to which it is most similar
- Produces time course of map topographies to be used in task related and statistical analyses

9.7 ERP Images

- 2-D representation of EEG Data from single electrode
- Useful for:
  - Single-subject data inspection tool
  - Linking trial-varying task parameters or behaviors to the time-domain EEG signal
    - Sort EEG trials according to values of aligning event (reaction time or phase of a frequency-band-specific signal at certain time point)
- Images smoothed by convolving image with 2-D Gaussian
- ERP Also made for:
  - Frequency-band-specific activity (filtered signal, power, or phase)
  - Time-frequency plots because time is on x-axis and data are colored
    - Y-axis has no frequency information

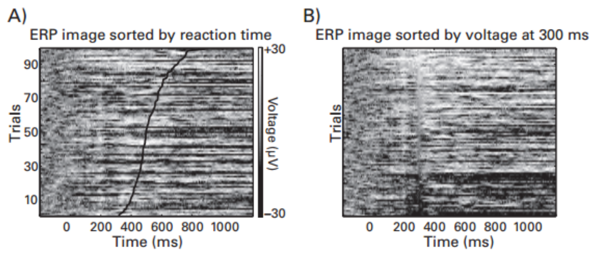


Figure 9.6
Example ERP images, both using data from electrode FCz. Time is on the *x*-axis, trials are on the *y*-axis, and the grayscale intensity (or color if you create this figure on your computer) corresponds to the EEG voltage values over time and trials. In panel A the trials are re-sorted according to the reaction time on each trial, and in panel B the trials are re-sorted according to the voltage value at 300 ms. The black line in panel A corresponds to the reaction time on each trial.

10 The Dot Product and Convolution

- Extracting time-varying, frequency-band-specific info from EEG Data

10.1 Dot Product

- Sum of elements in one vector weighted by elements of another vector
    - (signal-processing interpretation)
- Covariance or similarity between two vectors
    - (statistical interpretation)
- Mapping between vectors
    - Product of magnitudes of two vectors scaled by cosine of angle between them
    - (Geometric interpretation)
- To compute dot product:
    - Multiply each element in one vector by corresponding element in other vector
    - (First element in vector A x First element in vector B)
        - For all elements
    - All of the points are then summed
- Can conceptualize an EEG signal with 640 time points as a single point in a 640 dimensional space
    - Location of that point is defined by values of EEG signal at each time point

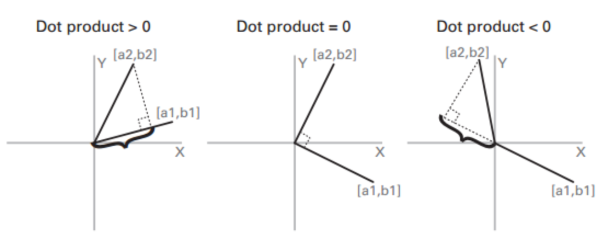


Figure 10.1
Graphical illustration of the geometric interpretation of the dot product between two two-element vectors. Curly brackets illustrate the magnitude of the projection of one vector onto the other (this is the dot product).

10.2 Convolution

- Dot product computed repeatedly over time or space
- Time series of one signal weighted by another signal that slides along the first signal
    - (Signal Processing Interpretation)
- Cross-covariance: similarity between two vectors over time
    - (Statistical Interpretation)
- Time series of mappings between two vectors
    - (Geometric Interpretation)
- Frequency Filter
- One vector is considered the signal (EEG Data) → other vector considered the kernel (Wavelet or sine wave)

10.3 How Does Convolution Work?

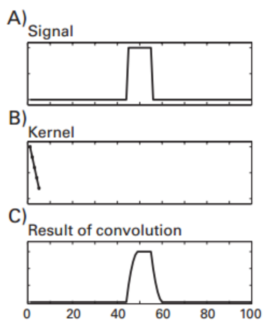
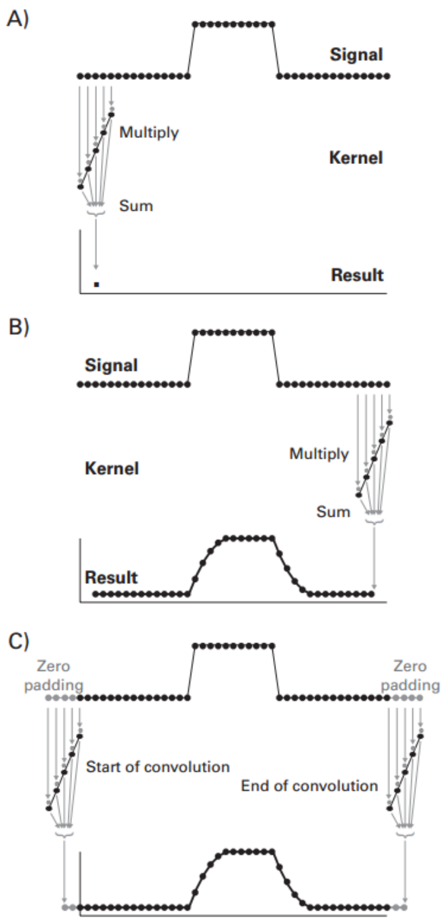


Figure 10.2
Example of convolution between two vectors, one labeled the signal (panel A) and one labeled the kernel (panel B). Panel C shows the result of the convolution between these two vectors.

- 1. Flip the kernel backward and line it up with the data
    - Compute dot product by multiplying each point in the kernel by each corresponding point in the data. Sum over all multiplications.
    - Place in a new vector in the position corresponding to the center of the kernel
        - Convenient to use kernel with odd number of data points so there's a center
- 2. Kernel is shifted to right by one time step. Signal does not move. Repeat dot product process and store result in position corresponding to center of kernel (now 1 time step to right)
    - Repeat until each end of data

- Begin convolution with time kernel to the far left. The right side of the (reversed) kernel overlaps with the leftmost points of the data
- (Length of kernel – 1) # of zeros added before kernel
    - After running convolution, trim results by removing ½ of the length of kernel from beginning and ½ from the end



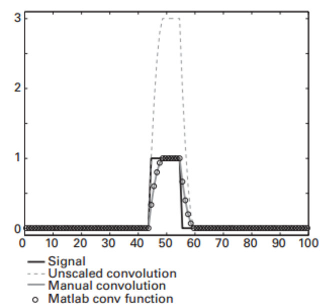


Figure 10.4
Overlap of results of manual time-domain convolution and the Matlab frequency-domain convolution function `conv`. Also shown is the result of convolution after normalization, which is not necessary when the sum of the kernel points is zero, as is the case with wavelets.

- To compare result of convolution with original signal → scale result of convolution by sum of kernel points
  - Note: post-convolution scaling is not same as mean-centering kernel before convolution

10.4 Convolution vs. Cross-Covariance

- Cross-correlation = cross-covariance scaled by variances
- Convolution → Kernel is reversed
  - Cross-covariance → Kernel kept in original orientation
- If convolution kernel is temporally symmetric
  - Convolution and cross-covariance yield identical results

10.5 Purpose of Convolution for EEG Data Analyses

- Isolate frequency-band-specific-activity
- Localize frequency-band-specific activity in time
  - As wavelet is dragged along EEG data, it reveals when and to what extent the EEG data contains features that look like the wavelet

- When convolution repeated on same data using wavelets of different frequencies, a time-frequency representation can be formed

11 Discrete Time Fourier Transform, the FFT, and the Convolution Theorem

- Computes dot product between the signal (EEG data) and sine waves of different frequencies (kernels)
- 3 Characteristics of Sine Waves:
    - Frequency – how fast (Cycles per second – Hz)
    - Power – amplitude squared
    - Phase – timing of the sine wave (radians or degrees)

11.1 Making Waves

- Sine waves are: A sin(2π*ft* + θ)
    - Amplitude
    - f = frequency of sine wave
    - t = time
    - θ = phase angle offset

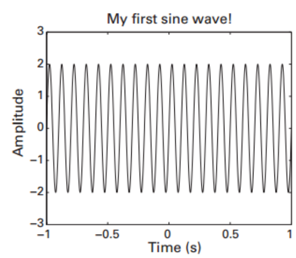


Figure 11.1
A sine wave that was created using the expression $2\sin(2\pi 10t + 0)$ (thus, amplitude of 2 and frequency of 10 Hz).

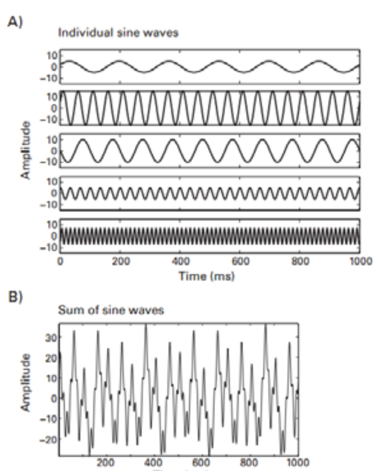


Figure 11.2
Several sine waves of differing amplitudes, frequencies, and phases, plotted separately (panel A) and after being added together (panel B).

11.2 Finding Waves in EEG Data with the Fourier Transform

- Have the time series

- Want to know which sine waves with which frequencies, amplitudes, and phases will reconstruct the time series

- Fourier transform = computes dot product between sine waves of different frequencies and the EEG data

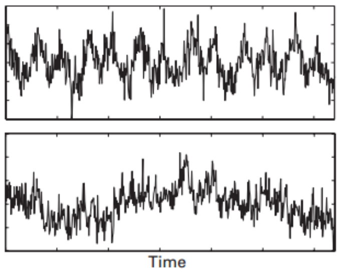


Figure 11.3
One of these time series was generated by summing different sine waves and adding noise; the other time series is real EEG data. Can you guess which is which? What evidence do you use to support your decision?

11.3 The Discrete Time Fourier Transform

```
N = 10; % length of sequence
data = rand(1,N); % random numbers
% initialize Fourier coefficients
fourier = zeros(size(data));
time = (0:N-1)/N; % time starts at 0; dividing by N normalizes to 1
% Fourier transform
for fi=1:N
    % create sine wave
    sine_wave = exp(-1i*2*pi*(fi-1).*time);
    % compute dot product between sine wave and data
    fourier(fi) = sum(sine_wave.*data);
end
```

- 1. Sine wave is created
- 2. Computes dot product between sine wave and the data
  - Number of unique frequencies that can be extracted from a time-series is one half the number of data points in that time series, plus the 0 frequency
- Discrete Fourier Transform
  - $X_f = \sum_{k=1}^{n} x(i)\, e^{-12\pi f(k-1)n^{-1}}$
    - n = # of data points in vector x
    - X_f = Fourier coefficient of time series variable x at frequency f

11.4 Visualizing the Results of a Fourier Transformation

- 2D plot with frequency on the x-axis and power (or amplitude) on the y-axis

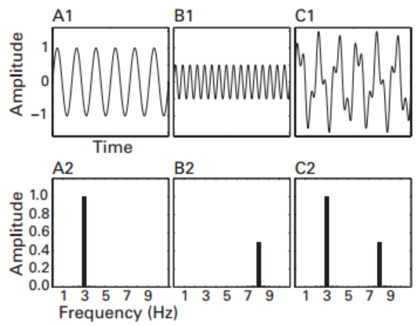


Figure 11.4
Two sine waves of 3 Hz and 8 Hz (panels A1 and B1), and their sum (panel C1) have a Fourier power spectrum that can be represented in plots of frequency (*x*-axis) by amplitude (*y*-axis) (see plots in the bottom row). The frequency of the sine wave corresponds to the position on the *x*-axis, and the amplitude of the sine wave corresponds to the position on the *y*-axis.

- 3D plot representing frequency-power-phase space
    - Phase = position of sine wave at each frequency when it crosses time = 0

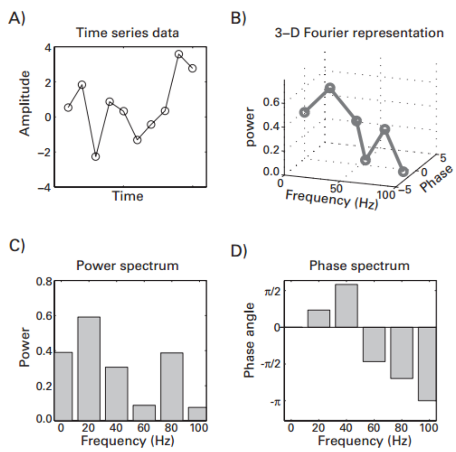


Figure 11.5
Panel A shows a time series of randomly generated data (sampled at 200 Hz). Panel B shows a frequency representation of the data as a line through 3-D Fourier space (frequency, power, and phase; see axis labels). Panels C and D show the projections of the 3-D Fourier result onto two dimensions at a time. Panel C shows only the frequency and power information from the Fourier transform, and panel D shows only the frequency and the phase information from the Fourier transform.

11.5 Complex Results and Negative Frequencies

- Half of the Fourier series has "positive" frequencies, and half has "negative"
    - Negative = capture sine waves that travel in reverse order around the complex plane compared to those that travel forward
- For a signal with only real numbers, the negative frequencies mirror the positive frequencies, so you can ignore them
    - Instead just double amplitude of positive frequencies
    - Don't remove negative frequencies (necessary for inverse Fourier transform)

11.6 Inverse Fourier transform

- Results of a Fourier transform in a series of coefficients that represent the dot product of each complex sine wave to the data
- Inverse Fourier transform:
    - Start with sine waves of different frequencies, amplitudes, and phases summed together to form a single time series
- To compute inverse Fourier transform:
    - Build sine waves of specific frequencies
    - Multiply them by respective Fourier coefficients at those frequencies
    - Sum all sine waves together
    - Divide by number of sine waves
- $X_f = \sum_{k=1}^{n} x(i)\, e^{i2\pi f(k-1)n^{-1}}$

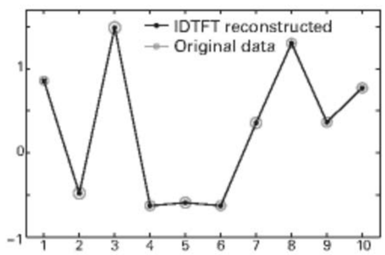


Figure 11.6
Time series data of randomly generated numbers and the reconstruction of the data using the inverse discrete time Fourier transform (IDTFT).

11.7 Fast Fourier Transform

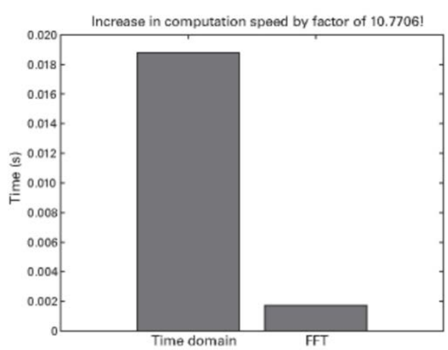


Figure 11.8
Computation times for the discrete time Fourier transform versus the FFT.

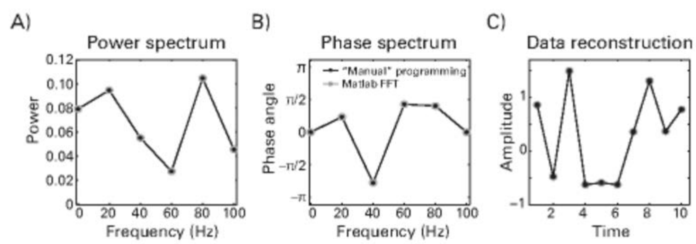


Figure 11.7
The fast Fourier transform (FFT) produces identical results as the discrete time Fourier transform and is much faster. (Note that due to very small computer rounding errors, you might see 0 or 2π, or $-\pi$ or $+\pi$, for the phase value at the Nyquist frequency.)

- With huge datasets, FFT is 1000x faster than FT

11.8 Stationarity and Fourier Transform

- Fourier transform assumption
    - Data is stationary – Data Elements Below Don't Change Over Time
        - Mean
        - Variance

        - Frequency Structure
    - ^^^ Doesn't actually happen in real life so…
- Perform temporally localized frequency-decomposition
    - Wavelet Convolution
    - Filter-Hilbert
    - Short-term FFT

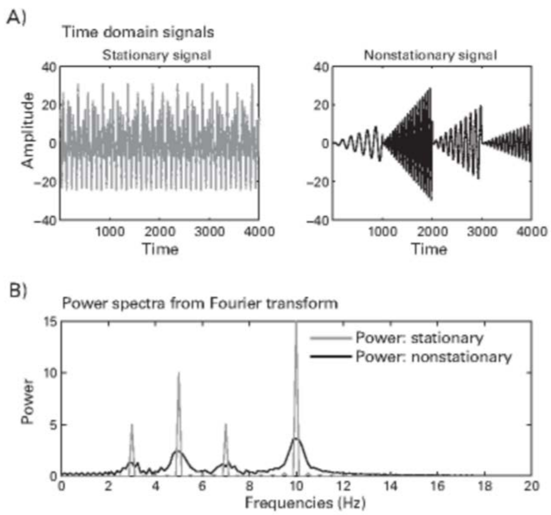


Figure 11.9
Violations of stationarity (here introduced by changes in frequency and amplitude over time) result in energy at frequencies that were not explicitly generated when creating the data. Nonetheless, spectral peaks can clearly be observed for both the stationary and the nonstationary time series. Note that for the stationary time series, the power peaks appear to be carrot shaped, but this is due to using lines instead of bars to represent discretely sampled frequencies. If you look closely, you can also see small but non-zero power at many frequencies for the stationary time series; these are artifacts resulting from the sharp edges at the beginning and ends of the time series.

- Temporally localized methods:
    - Assume data are stationary within relatively brief periods of time (few hundred ms)
- Second reason to perform temporally localized frequency decomposition
    - Fourier transform doesn't show how dynamics change over time
        - Time varying changes in frequency structure cannot be observed directly in power or phase plots

11.9 Extracting More or Fewer Frequencies than Data Plots

- Number of frequencies from Fourier Transform
    - N/2 + 1.
    - N is the number of time points in the data
    - +1 is for DC or zero-frequency component
- Can increase frequency resolution by:
    - Adding zeros at end of time series → (Zero padding)
    - Increases quantities of frequency
    - Improves frequency resolution
    - DOES NOT improve frequency precision
- Zero padding can make Frequency-Domain Convolution more convenient + faster to perform

11.10 Convolution Theorem

- States that convolution in time domain is same as multiplication in frequency domain
  - Therefore two means of performing convolution
- 1. Time-domain version of convolution
  - Slow
  - Flip kernel backward, slide it along signal, and compute dot product at each time step
- 2. Frequency-Domain Convolution
  - Fast
  - Fourier transforms of the signal and kernel
  - Multiplying Fourier transforms together point-by-point (freq-by-freq)
  - Taking inverse Fourier transform
  - Result of multiplication (ie convolution) is frequency structure common to both kernel and signal → conceptualized as frequency-domain filter

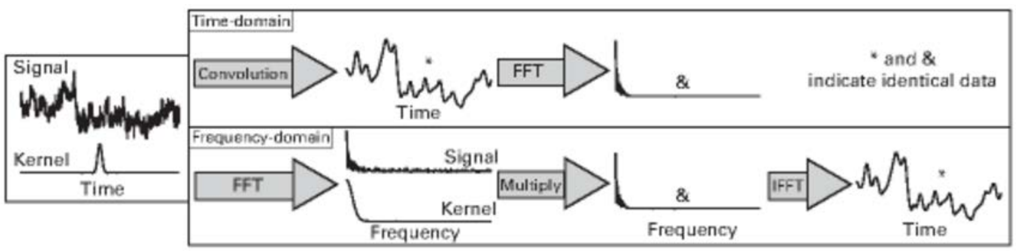


Figure 11.10
Illustration of the convolution theorem and the interchangeability of time-domain convolution and frequency-domain multiplication. The two time series with asterisks are identical, as are the two frequency spectra with ampersands.

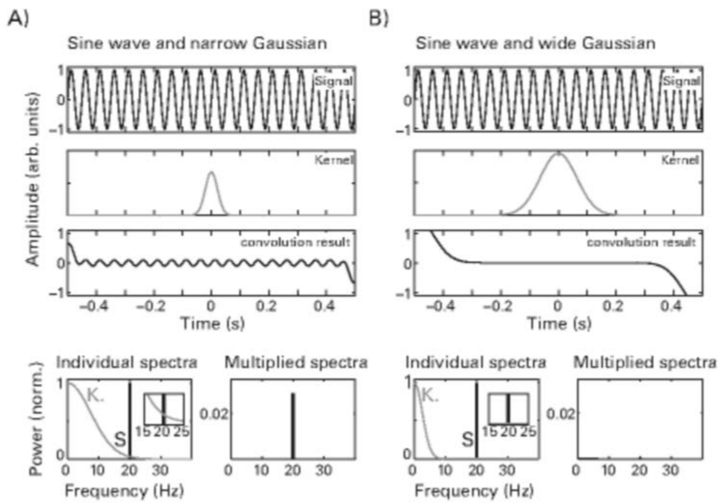


Figure 11.11

The convolution between a 20-Hz sine wave and a narrow Gaussian (panel A) dampens the sine wave, whereas the convolution between the same sine wave and a wide Gaussian (panel B) obliterates the sine wave. This is because the power spectrum of the sine wave (bottom row) overlaps slightly with the power spectrum of the narrow Gaussian at 20 Hz, but the power spectrum of the sine wave does not overlap with the power spectrum of the wide Gaussian. The gray line corresponds to the kernel (K), and the black bar corresponds to the signal (S). The insets in the power plots highlight the overlap (or lack thereof) between the frequency representations of the Gaussians and the frequency representation of the sine wave. The power spectra were normalized to 1 to facilitate visual comparison. The sharp rise and drop at the beginning and end of the result of convolution are edge artifacts.

- Power spectrum of narrow Gaussian has nonzero values that overlap with nonzero values of power spectrum of the 20Hz sine wave
- Power spectrum of wide Gaussian is zero at frequencies where power spectrum of sine wave is nonzero
    - Resulting freq.-spectrum multiplications produce:
        - Narrow Gaussian: amplitude-attenuated sine wave
        - Wide Gaussian: No sine wave
- BASIS OF WAVELET CONVOLUTION
    - Pass EEG data through set of filters (wavelets) tuned for specific frequencies
    - Result is frequency-band intersection between EEG data and wavelet

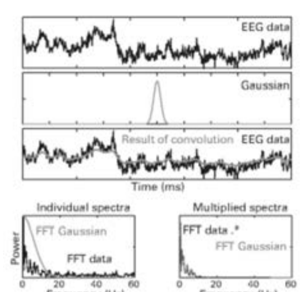


Figure 11.12

Convolving an EEG time series from one trial with a Gaussian low-pass-filters the data. This results from the frequency spectrum of the Gaussian tapering the higher frequencies in the EEG data. Note that this example here is meant for illustration of the convolution theorem and how convolution acts as a frequency filter; convolution with a Gaussian is not necessarily the best method for filtering EEG data. Chapter 14 contains more in-depth discussions of how to construct and apply bandpass filters to EEG data. The y-axis scaling of the power spectra is arbitrary to improve visibility.

- Gaussian kernel is a low-pass filter

11.11 Tips for Performing FFT-Based Convolution in Matlab

- Won't Produce Valid Convolution

```
result = ifft(fft(signal) .* fft(kernel));
```

- Result of convolution must be = length of signal + length of kernel – 1
- After computing inverse Fourier transformation →
    - Remove appropriate # of time points from beginning and end of time series
- Double-Check by Trying:

```
result = conv(signal,kernel,'same');
```

    - Full instead of same option → length of the result of the convolution
        - Ie. Length of signal + length of kernel + 1

12 Morlet Wavelets and Wavelet Convolution

- Fourier transform
  - Gives frequency-domain representation of EEG data
  - Limitations:
    - Changes in frequency structure over time are difficult to visualize
    - EEG data violate the stationarity assumption of Fourier analysis
  - Because of limitations, apply:
- Time resolved frequency decomposition representations
  - (ie. Time-frequency representations)
  - Retains advantages of time and frequency domain
    - Sacrifices little in temporal and frequency precision
- Morlet wavelet (or Gabor wavelet)
  - A sine wave in middle, but tapers off to zero at both ends
  - Useful for localizing changes in frequency characteristics over time

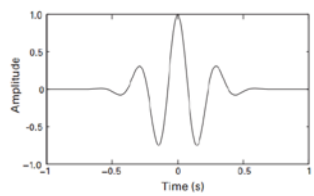


Figure 12.1
A Morlet wavelet, which is created by windowing a sine wave by a Gaussian.

12.1 Why Wavelets?

- Fourier transform doesn't show how frequency changes over time
  - Because kernel used has no temporal localization
  - (Amplitude of sine wave continues to fluctuate over its entire time series)
- To obtain temporally localized frequency information
  - Dot product should only be computed with part of sine wave in specific time window

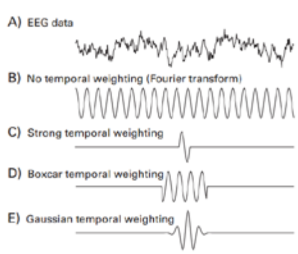


Figure 12.2
In order to extract time-varying frequency-specific information from EEG data (A), the data must be convolved with a sine wave. Without tapering of the sine wave (B), the result reflects frequency-specific information from the entire time series. Use of only one cycle (C) maximizes temporal precision but at the expense of frequency precision. A uniform boxcar tapering (D) is suboptimal because of decreased temporal specificity and potential artifacts from sharp edges. A Gaussian tapering (E) (also known as a Morlet wavelet) provides an adequate balance between temporal and frequency precision without introducing edge artifacts.

- Best option is to use Gaussian taper to window the sine wave (Morlet Wavelet)
    - Has no sharp edges that produce artifacts
    - Dampen influence of surrounding time points on estimate of frequency characteristics at time point
    - Allow you to control trade-off between temporal and frequency precision
- Many types of wavelets. Criteria:
    - Very close to 0 (or 0) at both ends
    - Mean value of 0
    - Morlet wavelet well suited for localizing frequency information in time
- Assumption of wavelet convolution:
    - Signal is stationary only during time period in which wavelet looks like a sine wave
    - Also, violations of local stationarity do not invalidate wavelet convolution results
        - Will decrease accuracy of frequency information
- **All time-frequency decomposition methods characteristic:**
    - Activity at each time point is an estimate of instantaneous activity and is influenced by activity from neighboring time points

12.2 How to Make Wavelets

- 1. Create a sine wave (Same number of time points/sampling rate as Gaussian)
- 2. Create a Gaussian wave (Same number of time points/sampling rate as ^^ sine wave)
- 3. Multiply them point by point
    - Frequency of wavelet is frequency of sine wave
- Frequency of a Morlet wave is its *peak (or center) frequency*
    - Wavelets contain energy in a range of frequency bands for which the frequency of the sine wave is the peak
- Create Gaussian Window

$$GaussWin = ae^{-(t-m)^2/(2s^2)}$$

- a = amplitude (height of Gaussian)
- t = is time
- m = an x-axis offset
- s = standard deviation or width of Gaussian

$$s = \frac{n}{2\pi f}$$

- f is frequency in Hz
- n is number of wavelet cycles
    - Determines trade-off between temporal and frequency precision
    - **Super important**
- Fourier transform
    - Many sine waves of different frequencies
- Time-frequency decomposition via wavelet convolution

    - Many wavelets of different frequencies that can be specified by you (rather than by number of data points in time series)
    - Number of wavelets to be used is not constrained at all
- Group of wavelets that share properties but differ in frequency = family of wavelets
- **Practical and theoretical limits of constructing families of wavelets:**
    - 1. Can't use frequencies slower than epochs
        - Generally: 1s of data, use wavelets 4Hz faster
    - 2. Frequencies of wavelets cannot be above ½ of the sampling rate (Nyquist Frequency)
    - 3. Due to frequency smoothing from time freq. presicion tradeoffs, frequencies close to eachother will likely provide similar or nearly identical results
        - Generally: 15 – 30 frequencies spanning 3Hz – 60Hz is sufficient

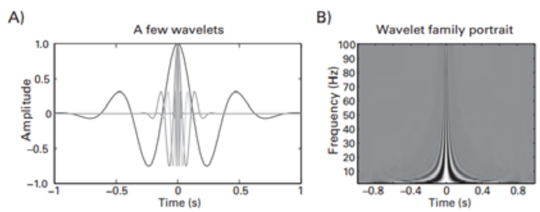
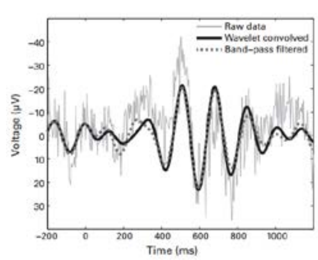


Figure 12.4
Different members of this wavelet family were created by changing the frequency of the sine wave while leaving other parameters unchanged.

- Figure 12.B
    - Y-axis corresponds to peak frequency of members of wavelet family
    - X – axis represents time
    - Color of plot represents amplitude
        - Red = positive deflection of wavelet
        - Blue = negative

12.3 Wavelet Convolution as a Bandpass Filter

- Convolution
    - A vector that expresses the time-varying mapping between a kernel and a signal



    - Wavelet convolution is bandpass filtering
        - When freq. spectrum of wavelet is multiplied by frequency spectrum of EEG, inverse Fourier transform is computed
            - Resulting time series contains frequency characteristics of EEG data that are tapered by frequency characteristics of the wavelet (Gaussian around peak frequency)

12.4 Limitations of Wavelet Convolution as Discussed Thus Far

- 1. Some elements not apparent in bandpass filtered signal
  - Time-frequency analyses
  - Power Information
  - Phase information
    - Could use Hilbert transformation of band pass-filtered signal (but simper option in chap. 13)
- 2. Result of convolution with a Morlet wavelet depends on phase offsets between wavelet and data

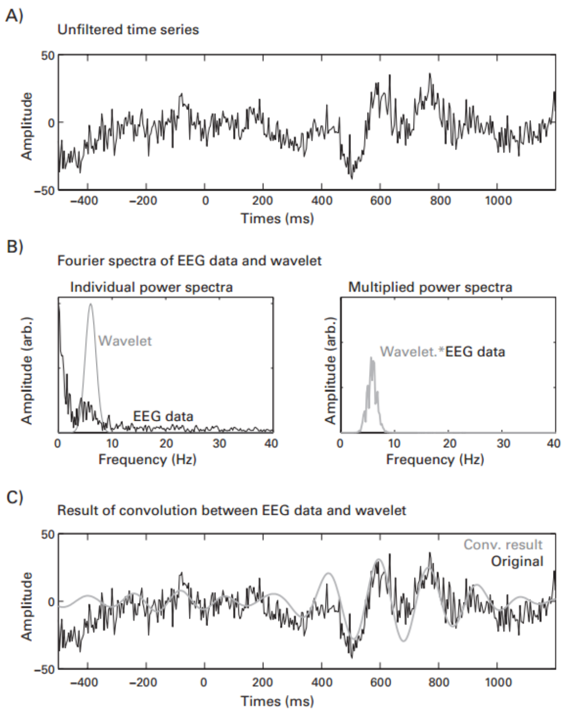


Figure 12.6
Illustration of why wavelet convolution acts as a bandpass filter. Panel A shows raw EEG data from one trial from electrode FCz. Panel B shows the power spectra of the Morlet wavelet and the EEG data. Note that the wavelet has a Gaussian shape in the frequency domain. The left-hand plot in panel B shows the point-by-point multiplication of the frequency spectra of the wavelet and the EEG data. Panel C shows the result of convolution overlaid on top of the original EEG data. This figure also illustrates how the result of convolution reflects activity that is maximal at the peak frequency of the wavelet (here, 6 Hz), but also activity from a weighted combination of surrounding frequencies (here ranging from around 3 Hz to 9 Hz).

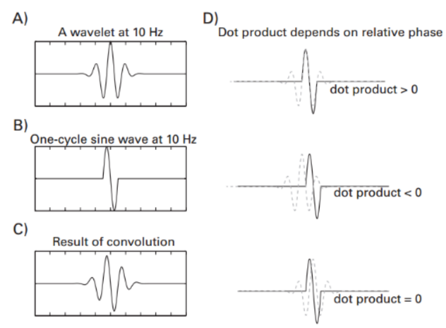


Figure 12.7
The result of each step of convolution (the dot product between the wavelet and the data) depends on the phase relationship between the kernel and the signal at that time point. This issue is resolved by using complex wavelets, as discussed in chapter 13.

- Same sine wave → different taperings
  - Convolution results show:
    - There are points where 2 vectors are orthogonal
      - When 90 degree phase lag between wavelet and one-cycle sine wave
    - Other points in time where two vectors have negative dot product
      - When 180 degree phase lag between two signals
- If looking for energy in EEG signal at specific frequency and at a specific time point
  - Align wavelet so it has 0 degree phase lag with EEG data at time point of interest
  - Then compute dot product
  - ^^^^ is not what you want

- Want relationship between wavelet and EEG data at all time points and all phase lags (not just some)

- Resolving Both Limitations of real-valued Morlet Wavelets
  - EEG data re convolved with complex Morlet wavelets
    - Wavelets with both real and imaginary components
    - Mapping does not depend on phase lags
    - INSTEAD, represented in a 2D space
      - Allows you to extract:
      - Bandpass-filtered signal
      - Time-frequency power
      - Phase information

13 Complex Morlet Wavelets and Extracting Power and Phase

13.1 The Wavelet Complex

- Complex Morlet wavelets can be used to extract estimates of time-varying frequency band-specific power and phase from EEG data
  - Occupies 3D space:
    - Time / Real / Imaginary

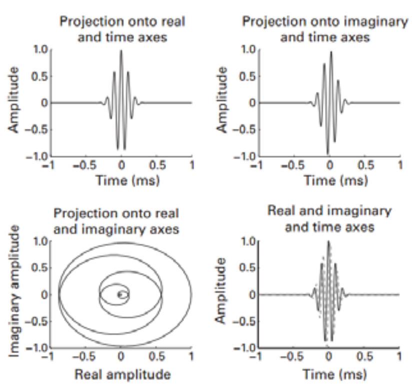


Figure 13.1
A complex wavelet is a 3-D (time, real, imaginary) function. Plotted here are projections onto various pairs of those dimensions.

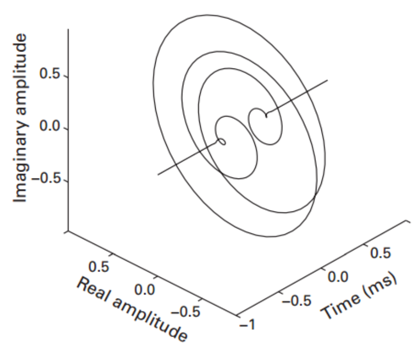


Figure 13.2
Three-dimensional view of a complex wavelet.

13.2 Imagining the Imaginary

- Imaginary numbers indicated with *i* or *j*

13.2 Rectangular and Polar Notation and the Complex Plane

- Polar coordinates useful for:
  - Describing properties of frequency-band-specific activity
    - Bandpass-filtered signal
    - Power
    - Phase
- Convert from polar to Cartesian (for imaginary component) using basic trig.

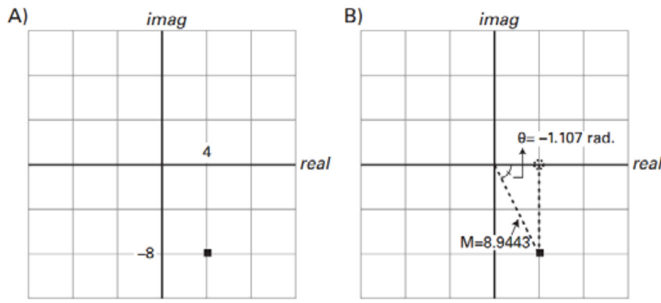


Figure 13.3
The same point in a complex space (with a real axis and an imaginary axis) can be represented using Cartesian (A) or polar (B) notations.

$$M = \sqrt{(real^2 + imag^2)} \qquad real = M\cos(\theta)$$

$$\theta = \arctan(imag\,/\,real) \qquad imag = M\sin(\theta)$$

$$real + imag = M\cos(\theta) + M\sin(\theta)$$

$$real + imag = M[\cos(\theta) + \sin(\theta)]$$

$$a + ib = M[\cos(\theta) + i\sin(\theta)]$$

13.3 Euler's Formula

- Allows you to represent complex numbers as points on a circle

$$Me^{i\theta} = M[\cos(\theta) + i\sin(\theta)]$$

  - e = base of natural log (2.718)
  - θ = any real number (angle in radians)
  - M = magnitude of vector
- Change cosine = change in real-axis position
- Change sine = change in imaginary-axis position
- Difference between real and imaginary parts of a complex wavelet is difference between sine/cosine
  - Sine / cosine related
    - ¼ counterclockwise rotation in complex space

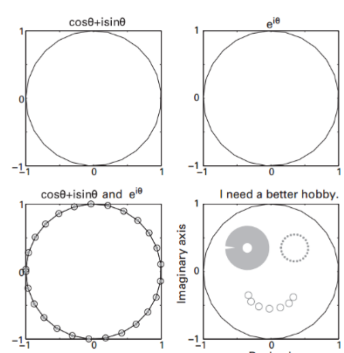


Figure 13.4
Illustration of equality of Euler and trigonometric representations and another illustration (lower right panel) of how the real component corresponds to cosine while the imaginary component corresponds to sine.

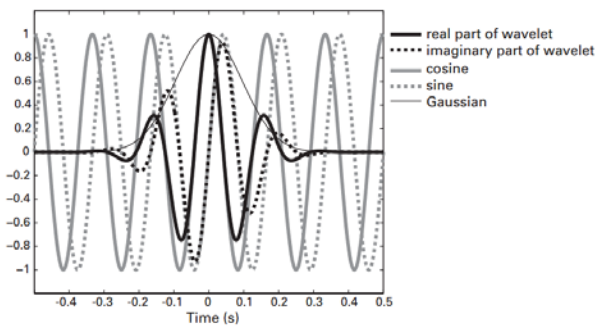


Figure 13.5
Overlaying the real and imaginary parts of a complex wavelet along with a cosine wave and a sine wave. This shows that the real part of the wavelet corresponds to a cosine wave, and the imaginary part of the wavelet corresponds to a sine wave.

- Complex Morlet Wavelet Equation

$$cmw = Ae^{-t^2/2s^2}e^{i2\pi ft}$$

$$A = \frac{1}{\left(s\sqrt{\pi}\right)^{1/2}}$$

- 1st part of equation: Gaussian
- 2nd part of equation: complex sine wave
    - Combination of Euler's method and 2*pi*f*t part of sine wave
- S = standard deviation of gaussian
- f = peak frequency
- A = frequency band-specific scaling factor
    - Not necessary if plan to apply:
        - Baseline normatization
            - Percentage change
            - Decibel
        - Complex wavelet convolution only to obtain phase angle time series
- Complex sine wave composed of:
    - Cosine (real) and sine (imaginary) component

13.4 Euler's Formula and the Result of Complex Wavelet Convolution

- Dot product between wavelet and one-cycle sine wave could be pos, neg, or zero
  - Depending on slight shifts in relative phase between signal and kernel
- Dot product between a complex wavelet and signal, result is complex number
  - Euler's formula applied to represent complex number as point in polar space
  - (Endpoint of a vector from origin of polar space to that point)

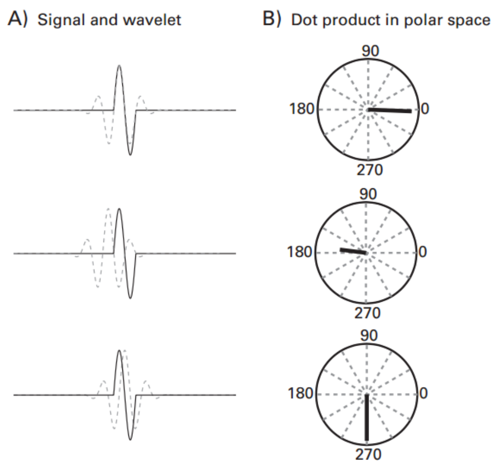


Figure 13.6
The dot product (one step of convolution) between a complex Morlet wavelet (real part shown here using a dotted line) and a one-cycle sine wave produces a complex number, here represented as a vector in a polar plot using the magnitude and angle. Note that the more the wavelet and the one-cycle sine wave overlap, the longer the vector is in complex space, regardless of the phase angle of that vector. This figure can be compared with figure 12.7.

- Notice:
  - When dot product with real-valued wavelet was less than zero, dot product with complex wavelet has vector pointing to the left
  - … was zero, dot product with complex wavelet has vector pointing down
  - … was positive, dot product with complex wavelet has vector pointing right
- The result of dot products with real-valued wavelet maps onto the real axis in the result of dot products with complex wavelets
  - Imaginary axis is ignored in dot product with real-valued wavelet

13.5 Euler's Formula and the Result of Complex Wavelet Convolution

- Dot product between wavelet and one-cycle sine wave can be: negative, positive, or zero
  - Depends in shifts in relative phase between signal and kernel
  - Can be resolved using complex wavelets
- Euler's Formula
  - Represent complex number as point in polar space

    - Endpoint of a vector from origin of polar space to that point
- The result of dot products with the real valued wavelet, maps onto the real axis in the result of dot products with complex wavelets
    - Imaginary axis ignored in dot product with real valued wavelet
- More overlap between wavelet and one-cycle sine wave = longer line in complex space
    - Length of line (from origin to point described by complex number) provides info on similarity of overlap
        - Between kernel and signal
        - Length NOT DEPENDENT on phase relationship between kernel and signal
    - Phase relationship between kernel and signal
        - Characterized by angle of vector
- Content to Extract from complex dot product:
    - 1. Projection onto the real axis (or imaginary axis_
        - Bandpass-filtered signal
        - Pos or neg. depending on phase relationship between signal and kernel
    - 2. Magnitude of vector from origin to point in complex space defined by result of dot product
        - Vector length is related to similarity in overlap between signal and kernel
        - Length of vector = amplitude
        - Length squared = power
            - Measure of instantaneous power at point in time corresponding to center of wavelet with respect to EEG data and peak freq. of wavelet
    - 3. Angle of the vector with respect to positive real axis
        - Estimate of phase angle at point in time corresponding to center of wavelet at peak frequency of wavelet
- Note: Power and phase values are estimates because influenced by activity at neighboring time points
- NOTE – COMPUTATIONAL EFFICIENCY:
    - Power extracted by:
        - Squaring length of complex vector: abs(x) .^2
        - Multiply complex vector by its conjugate: x.*conj(x)
            - Conjugate of complex # found by mult. Imaginary part by (-1)

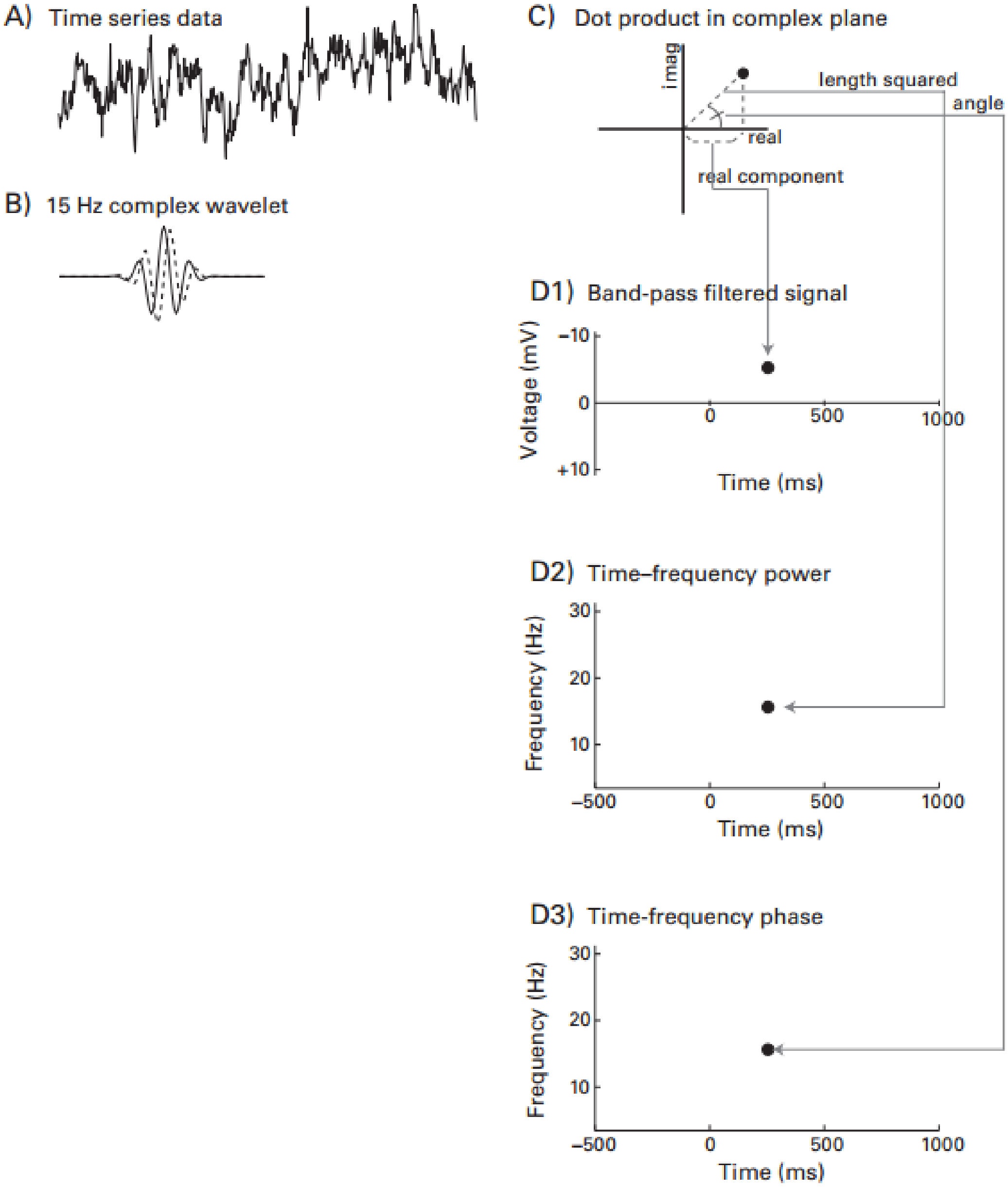


Figure 13.7

Graphical overview of how to extract the bandpass-filtered signal, power, and phase from one step of the result of complex convolution. Each step of convolution between the time series data (panel A) and a complex wavelet (panel B) is a dot product, which can be conceptualized as a point in a complex plane (panel C). The projection of that point onto the real axis is the bandpass-filtered signal (panel D1), the squared length of the vector from the origin to the dot product is power (panel D2), and the angle of the vector with respect to the positive real axis is the phase angle (panel D3). These values are then placed into time-frequency matrices at the frequency corresponding to the peak frequency of the wavelet and at the time point corresponding to the center time point of the wavelet and its corresponding temporal position in the EEG data (in the case illustrated here, the wavelet would be centered at 250 ms). Note that panels C and D illustrate general procedures for extracting power and phase from a complex signal resulting from any time-frequency decomposition method that produces an analytic signal, including the filter-Hilbert and short-time FFT methods.

13.6 From Time Point to Time Series

- Time-domain convolution involves computing time-sliding dot products between kernel and signal
  - o Extracting mag. And phase angle of result of a convolution
    - ▪ Construct time series of power or phase values from one frequency band
    - ▪ Repeated over multiple frequency bands to obtain a time-frequency map

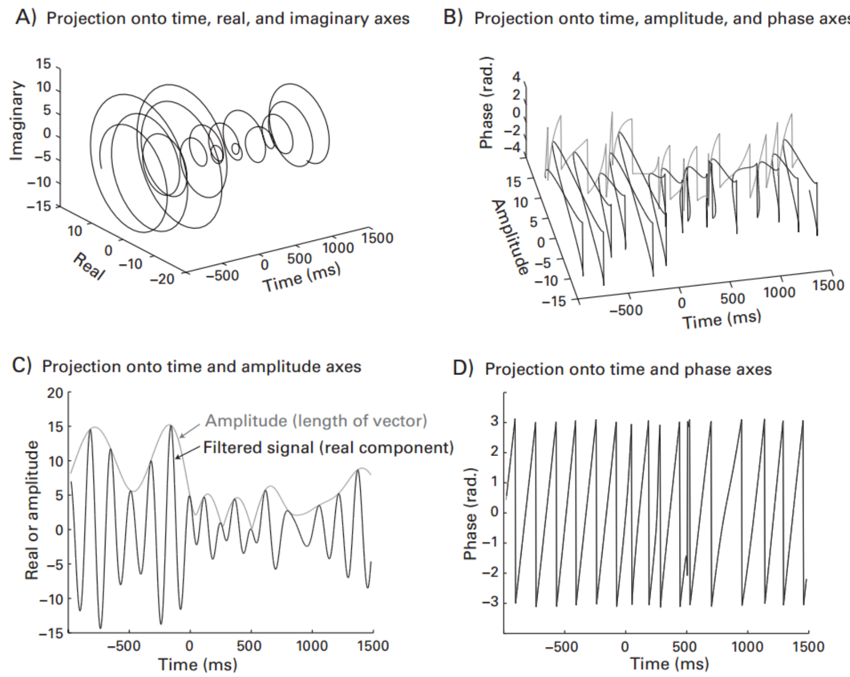


Figure 13.9
Try to conceptualize the result of complex Morlet wavelet convolution as a line in a 3-D space. After convolution, the three dimensions are time, real, and imaginary (panel A). For most EEG data analyses, the real and imaginary components are transformed to amplitude (or power) and phase (panel B). Panels C and D show the 3-D representation from panel B projected onto the time and amplitude axes (C) and the time and phase angle axes (D). Panel C also shows the real component from panel A, as a comparison with the amplitude. If you use the online Matlab code to generate this figure, you can freely rotate these images, which might help you conceptualize the 3-D nature.

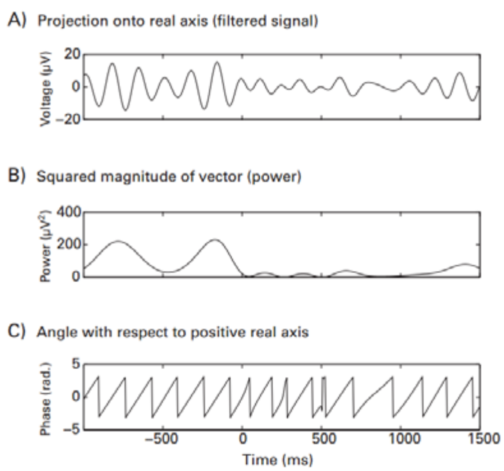


Figure 13.8
Different pieces of frequency-band-specific information extracted from EEG via complex wavelet convolution.

13.7 Parameters of Wavelets and Recommended Settings

- Some parameters discussed are also applicable in other time-frequency decomposition methods like: filter-Hilbert method.

13.7.1 How Low Should the Lowest Frequency Be

- Depends on:
    - Expectations about results
    - Task design
    - Timing of trial events
    - Length of data epochs
- If hypothesis driven, and only concerned with alpha-band activity
    - Stop at 5 – 6Hz
        - Analyze a bit below and above band to determine whether effect is frequency band specific
    - 1-s epochs
        - Activity below 4Hz will have low signal-to-noise ratio

13.7.2 How High Should the Highest Frequency Be

- Depends on
    - Expectations about results
    - Sampling rate
- Can't examine frequencies higher than Nyquist frequency
- Use many data points per cycle to increase signal-to-noise ratio
    - Ex. Sampling rate of 500Hz
        - Max frequency of 125Hz (4 sample points per cycle)
- No expectation about frequency band results
    - Range of 4 – 60 Hz
        - Captures main time-frequency dynamics
        - If task related activity close to lower/higher frequencies
            - Reanalyze using broader frequency range

13.7.3 How many Frequencies Should Be Used

- More frequencies
  - Freedom to select appropriate frequency ranges based on data
  - Allow to perform exploratory and post-hoc analysis + hypothesis based analysis
  - Smoother time-frequency plots (also easier to inspect visiually)
  - Negative:
    - Larger results matrices
    - Increased computation time
    - Increased requirements for multiple comparisons corrections
    - Frequency smoothing → auto-correction in time freq. plot
      - Overlap in time and frequency domains → similar EEG data extracted\
- Generally: 20-30 frequencies

13.7.4 Should Freq. be Linearly or Logarithmically Spaced

- Frequencies are often conceptualized on log. Scale
  - Use log. Spaced wavelet peak frequencies
  - Width between freq. will be approx. equal over range of freq. extracted
- Main results concern low-frequency activity
  - Use log. Scaling (highlights lower-freq. end of spectrum)
- Concern high-frequency activity
  - Linear scaling (highlights higher-freq. end of spectrum)
- Pay attention to scaling of X and Y axes!

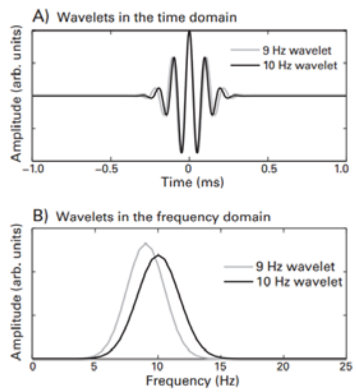


Figure 13.10
Time-domain (panel A) and frequency-domain (panel B) illustration of how wavelets with close frequencies have similar and largely overlapping time and frequency profiles.

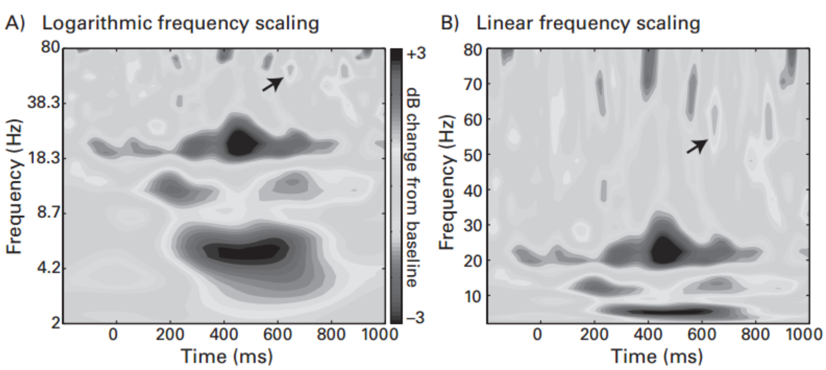


Figure 13.11 (plate 2)
Time-frequency results of the same analyses applied to the same data can look different depending on whether the frequencies in the *y*-axis are scaled logarithmically (panel A) or linearly (panel B). The black arrows, for example, show the same gamma power burst in both plots.

13.7.5 How Long Should Wavelets Be?

- Long enough that the lowest-frequency wavelet tapers to zero at neg. and pos. end of time
    - If wavelet is cut off → edge artifacts
- Unless wavelets with very low frequency,
    - Time range: -2 to +2 should work
- Wavelets should be centered in time window
    - Create wavelet using time vector from neg. to pos. number
    - Center of wavelet at time = 0
        - Also gives wavelet odd number of data points (good for centering)
- Wavelet has same sampling rate as EEG data

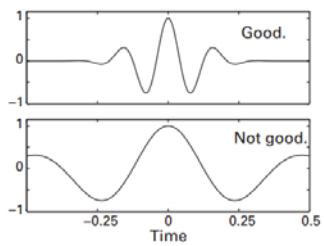


Figure 13.12
The lowest-frequency wavelet should taper cleanly to zeros. If your wavelet looks like the wavelet in the lower plot, increase the amount of time (in this example, –2 to +2 s would be enough time). There is no theoretical limit on how long wavelets can or should be, so it is best to err on the side of making them too long compared to making them too short.

13.7.6 How Macy Cycles Used for the Gaussian Taper?

- # of cycles defines: width → defines width of wavelet
- Controls trade-off between temporal and frequency precisions
    - (Temp / Freq. resolution determined by data-sampling rate)
- Large number of cycles
    - Better freq. precision
    - Worse temporal precision
    - (Heisenberg Uncertainty – more you know about when, less you know about where)

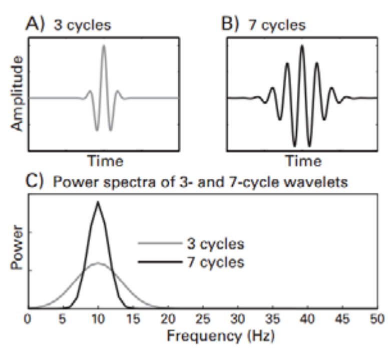


Figure 13.13
Two wavelets at the same frequency that differ only in number of cycles have different temporal and frequency characteristics and therefore different sensitivities to temporal and frequency features of the data. This has practical implications for the results, as shown in figure 13.14 (color plate 3).

- In Example:
    - 3 cycle wavelet better suited for detecting transient activations
    - 7 cycle wavelet more sensitive to longer activations at specific frequencies
    - 3 cycle wavelet more precise at localizing dynamics in time
    - 7 cycle wavelet more precise at determining frequency of dynamic

- Smaller # of cycles (3-4)
  - Looking for transient changes in activity
  - Hypothesis about speed of neural response in dissociating condition A from B
- Larger # of cycles (7-10)
  - Relatively long trial period where you expect freq-band-specific-activity. Looking for temporally sustained activity
  - Activity at frequencies within a narrow range (ie. Separating lower from upper alpha)
- If temporal and frequency precision are both important: perform the analysis twice.
- Example:
  - Panel A highlights temporal dynamics
  - Panel B highlights frequency dynamics of data

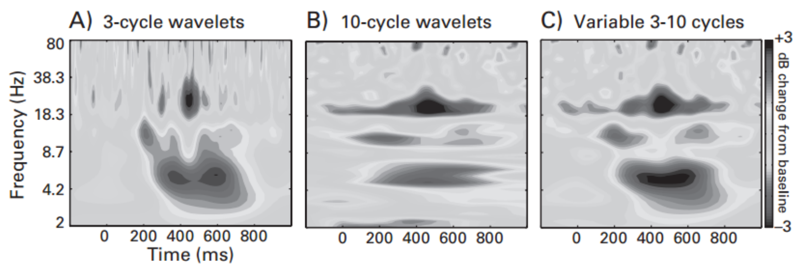


Figure 13.14 (plate 3)
The width of the Gaussian that is used to create the wavelets affects the features of the results that will be obtained from complex Morlet wavelet convolution. Different numbers of cycles can be used to highlight temporal precision (panel A) or frequency precision (panel B). Panel C shows that the balance between temporal and frequency precisions can change as a function of frequency; this increases temporal precision at lower frequencies and increases frequency precision at higher frequencies. The results were transformed to decibel (dB) change relative to a pretrial baseline. Baseline transformations are discussed in depth in chapter 18.

- A and B shows cases with constant # of wavelet cycles over frequency
- We can change # of wavelet cycles as a function of frequency by:
  - Increasing # of cycles with frequency of wavelet
- Generally: 3 – 14 cycles
  - Use more than 7: Check that they taper to 0
- Final Point:
  - Data should ideally be stationary during period in which wavelet is nonzero
  - More cycles = longer wavelet's nonzero components = lower time for which data is stationary
    - BUT: decreases accuracy of estimate of frequency characteristics

13.8 Determining Frequency Smoothing of Wavelets

- Extent to which neighboring frequencies contribute to results of wavelet convolution:
  - Reported in terms of full width at half-maximum (FWHM)
  - FWHM = frequency width for which power is at 50% on left and right sides of peak

$$FWHM = 2\sqrt{2\ln 2\sigma}$$

$$\sigma = Standard\ Dev.of\ Frequency\ response$$

- Equation ONLY valid for Gaussian Distributions
- Estimate FWHM by:
  - Normalizing power spectrum of wavelet so: min value = 0 | max value = 1
  - Next, identify frequency points prior to and following the spectral peaks closest to 0.5
  - Subtract those two frequencies

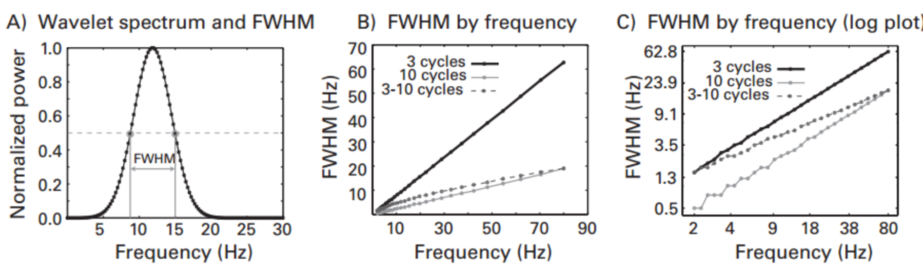


Figure 13.15
FWHM of wavelets, shown for one wavelet (panel A), and for a family of wavelets (panel B). Panel A illustrates how the FWHM can be computed as the range of frequencies corresponding to 50% of the power of the peak of the spectral representation. Panel B shows the FWHM as a function of frequency for the time-frequency plots shown in figure 13.14 (plate 3). Panel C shows the same information using logarithmic *x*- and *y*-axis scaling.

13.9 Tips for Writing Efficient Convolution Code in Matlab

- Use FFT of an order that is a power of two
  - FFT of 1024 point vector is father than FFT of 1023 point vector
  - Usually time series is zero-padded to get length of power of two
  - ***Remove these extra points after convolution"
- DON'T perform convolution on each trial separately
  - Faster to concatenate all trials into one long time series
    - Perform one convolution with all trials
    - Reshape results back to trial-by-trials matrix

13.10 Describing Analysis in Methods Section

- If use Morlet wavelet convolution, must include following details in methods section:
    - Minimum
    - Maximum
    - # of frequencies of wavelets
    - Whether frequencies increased linearly or logarithmically
    - # of wavelet cycles
    - Whether # of wavelet cycles changed as a function of frequency
    - Frequency domain FWHM of wavelets

14 Bandpass Filtering and the Hilbert Transform

- Time frequency decomposition method
    - Bandpass filtering
    - Applying Hilbert Transform
        - Analytic signal becomes a complex time series from which power and phase values are extracted
            - Same method as for complex wavelet convolution
- Benefit over complex wavelet convolution
    - Gives more control over frequency characteristics of filter
    - (Morlet wavelet is always Gaussian)
- Disadvantages
    - Matlab filter kernel construction functions are in Matlab signal-processing toolbox
    - Bandpass filtering is slower than wavelet convolution

14.1 Hilbert Transform

- Extract a complex signal (imaginary) from a signal that contains only a real part
    - Create/add phase quadrature component to Mcos(2*pi*f*t)
        - Phase quadrature created by rotating parts of complex Fourier spectrum of real value signal
- Hilbert Transform Procedure
    - 1. Compute Fourier transform of signal and create copy of generated Fourier coefficients multiplied by complex operator i
        - Turns: Mcos(2*pi*f*t) → i Mcos(2*pi*f*t)
    - 2. Identify positive and negative frequencies
        - Pos: 0 to Nyquist frequency
        - Neg: Nyquist frequency and above
    - 3. Convert iMcos(2*pi*f*t) → iMsin(2*pi*f*t)
        - By rotating positive-freq. coefficients ¼ cycle counter clockwise in complex space
            - (Multiplication with -i)
            - When rotated, pos freq. Fourier coefficients are added to original pos-freq. coefficients → Double original pos-freq. Fourier coefficients
        - By rotating negative-freq. coefficients ¼ cycle clockwise
            - (Multiplication with i)
            - When rotated, neg freq. Fourier coefficients are added to original neg-freq. coefficients → Double original pos-freq. Fourier coefficients
    - 4. Inverse Fourier transform modulated Fourier coefficients
- Results in:

- Analytical signal used in the same way one was used for complex Morlet wavelet conv.
- Follow process on chp. 13 to get:
  - Real component
  - Power
  - Phase-angle time series

- Hilbert transform can be computed by:
  - Doubling pos. frequency coefficients
  - Zeroing neg. frequency coefficients
- Using "Hilbert" function correctly in matlab
  - Plot phase angles over time
    - (Either will look as expected, or will look strange)
    - If strange:
      - Likely an error with Hilbert transform:
        - Time was not 1st dimension of matrix input to Hilbert function

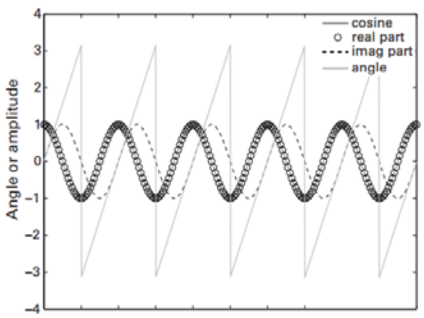


Figure 14.1
The Hilbert transform is used to extract complex information from a real-valued signal. Here, a cosine wave was created (solid gray line), and its Hilbert transform was taken. The real part of the result of the Hilbert transform is the original signal, while the imaginary part is a sine wave (which is what happens when a cosine is phase-shifted 90°).

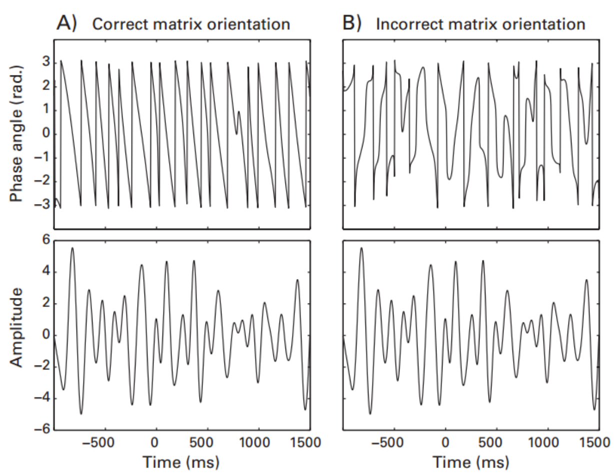


Figure 14.2
Applying the Matlab `hilbert` function to matrix data can produce incorrect results if the input data matrix is incorrectly oriented. This is easily identified by plotting the angle of the result of the Hilbert transform (top row). As mentioned in the text, the Hilbert transform does not change the real component (bottom row).

14.2 Filtering Data before Applying the Hilbert Transform

- Bandpass filtering not required:
  - Apply transform to broadband data
  - Result may be difficult to interpret because:
    - Frequencies with more power will contribute more to signal compared to frequencies with less power
- THEREFORE:
  - Filter data into separate frequency bands before applying the Hilbert transform
    - Results can be interpreted in: frequency-band-specific manner
- Main advantage:
  - More control over characteristics of filter
    - Gaussian shaped
    - Or plateau shaped

14.3 Finite versus Infinite Impulse Response Filters

- Finite Impulse (Single Input) Response – FIT
  - Response ends at some point
  - Preferred over IIR
    - More stable
    - Less likely to introduce nonlinear phase distortions
- Infinite Impulse Response – IIR
  - Butterworth IIR filter

14.4 Bandpass, Band-Stop, High-Pass, Low-Pass

- Bandpass
  - Keep activity between specified frequency (Most useful for time-frequency decomposition)
- Band-Stop
  - Remove activity between specified frequency
- High-Pass
  - Filters that retain high frequencies while attenuating low frequencies
  - Allows higher frequencies to pass through
- Low-Pass
  - Implied

14.5 Constructing a Filter

- When you convolve a wavelet with EEG data
  - Real component of conv. result is EEG data filtered around the peak frequency of the wavelet
  - Convolution result is:
    - Weighted combination of frequency structure of EEG data and frequency structure of wavelet
- Bandpass filtering
  - A kernel is constructed based on ideal frequency characteristics (you define)
    When kernel is convolved with EEG data, requested frequencies are preserved while undesired frequencies are attenuated
- Morlet Wavelet
  - Gaussian Shaped Power Spectrum
    - Produce overall smother time-frequency plots over frequency axis compared to those produced by Hilbert Method
- FIR Filter
  - Plateau shape recommended (but in theory can take any shape in freq. domain)
    - Provides bandpass filtering extra frequency specificity over wavelets

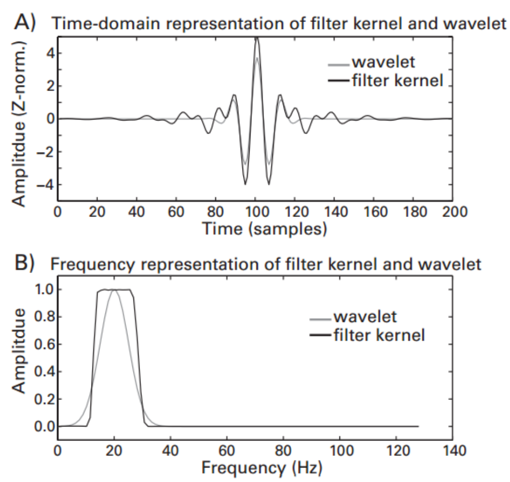


Figure 14.3
Comparison of the time-domain and frequency-domain representations of a Morlet wavelet and an FIR filter kernel. Both the wavelet and the filter kernel were designed to isolate 20-Hz activity. Data were amplitude-normalized to facilitate visual comparison. Panel B also illustrates why it is informative to refer to the "peak" frequency of a wavelet and the "center" frequency of a bandpass filter.

- Creating a Kernel in Matlab
  - Specify ideal filter shape and frequencies that define that shape
    - Frequencies are a specified as a fraction of Nyquist frequency
- In Signal Processing Toolbox
  - Firl
    - Same Input 1 as Firls
    - Same Input 2 except: only enter:
      - Lower Bounds
      - Upper Bounds
      - (Transition zones automatically = 0)
  - Firls
    - Finite impulse response filters via least squares
    - Input 1: Order Parameter – length of filter kernel (order + 1)
      - Order determines precision of frequency response
      - Larger Order: produce kernels with relatively better frequency precision
        - Increase computation time
      - Lower bound:
        - Resolve activity at a particular frequency, kernel must be long enough to contain at least one cycle at that frequency
        - Ex. Low freq. bound: 10Hz
          - Kernel must be 100+ ms long
          - Generally 2-5x lower frequency bound
            - 200 – 500 ms for 10Hz
      - Should be an even number: exclude filter representation for Nyquist freq.
        - Matlab always adds 1 to odd # orders
      - Input sample points for order parameters
        - (NOT in ms or s)

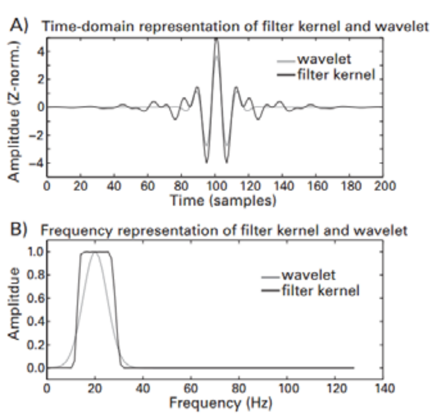


Figure 14.3
Comparison of the time-domain and frequency-domain representations of a Morlet wavelet and an FIR filter kernel. Both the wavelet and the filter kernel were designed to isolate 20-Hz activity. Data were amplitude-normalized to facilitate visual comparison. Panel B also illustrates why it is informative to refer to the "peak" frequency of a wavelet and the "center" frequency of a bandpass filter.

- Input 2. A vector of frequencies that defines the shape of the response
- For bandpass filter, use:
    - Zero frequency
    - Frequency of start of lower transition zone
    - Lower bound of the bandpass
    - Upper bound of the bandpass
    - Frequency of the end of the upper transition zone
    - Nyquist frequency
- Input 3. “Ideal” Filter Response Amplitude
    - Vector comprising as many #s as the second input
    - Contains 0s for frequencies want to attenuate / keep
    - Bandpass filter: [0 0 1 1 0 0]
        - Position:
        - 3 / 4
            - Lower/upper frequency bounds of bandpass plateau
        - 1 / 6
            - DC and Nyquist frequencies
        - 2 / 5
            - Frequency bounds of transition zones
        - Can use as many numbers as want to maximize control)
- Frequency width of bandpass filter
    - (length of plateau in freq. representation of kernel)
    - Narrower plateau → increased frequency precision (decreased temporal precision)
        - Narrow filters require longer kernels to resolve
    - As filter becomes infinitesimally smaller → perfect sine wave with an amplitude that never dampens
    - As filter becomes infinitesimally wider → time required to construct filter kernel becomes smaller
        - Temporal precision increases while freq. precision decreases
- Sharp edges in frequency domain can be avoided with Transition Zones
    - With firls (zones are 10% - 25% of lower and upper frequency bounds)
    - Sharper transition zone = better frequency response
        - Increase risk of time-domain ranging artifacts
    - Gentler transition zones = less frequency specificity
        - Decrease risk of ^^
    - Using transition zone < 10%
        - Smooth the filter kernel with Hann or Hamming window to minimize edge artifacts when filter is applied
        - Or use firl function

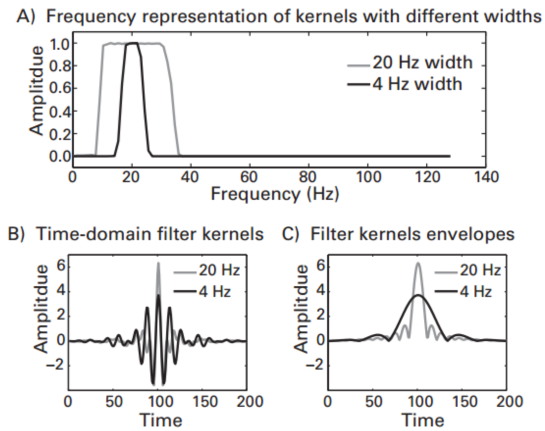


Figure 14.5

Illustration of how the width of the filter controls the time-frequency precision trade-off in bandpass filtering. Panel A shows the frequency representations of two filter kernels with the same center frequency (20 Hz) but with either 4-Hz or 20-Hz width (respectively, 2 Hz or 10 Hz on either side of the center frequency). Panel B shows the time-domain version of the same two filter kernels, and panel C shows their amplitude envelopes (the magnitude of the Hilbert transform of the time-domain kernel). Filters with wider pass-bands have kernels that taper to zero after fewer time points (gray lines in panels B and C, corresponding to a filter with a 20-Hz width), thus improving temporal precision at the expense of worse frequency precision. In contrast, filters with narrow pass-bands have kernels that taper to zero after more time points (black lines in panels B and C, corresponding to a filter with a 4-Hz width), thus improving frequency precision at the expense of worse temporal precision.

  - Fir2
    - Frequency-sampling-based filter construction
  - Firrocos
    - Raised cosine-shaped filter
  - Gaussfir
    - Gaussian-shaped filter
  - Firpm
    - Parks-McClellan
- Difference between kernels created by firls vs. firl
  - Transition zone:
    - Firl automatically sets zones to 0 then smoothes resulting filter kernel to minimize ringing artifacts
      - Smoothing creates nonzero transition zone
    - Firls
      - Use with transition zones of 0
      - Smoothing the resulting filter kernel with a Hamming window → nearly perfectly reconstruct filter kernel from firl
- Narrow-band filter
  - Firl
- Wide-band filter
  - Firls

- If unsure which kernel to use:
    - Plot kernel and frequency representation, and inspect sample data after applying different filters to see what's best

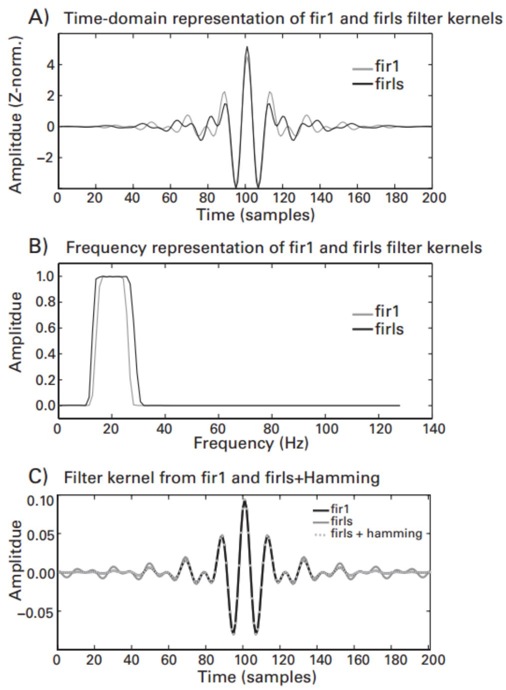


Figure 14.6
Comparisons of the time (panel A) and frequency (panel B) representations of FIR filter kernels created through Matlab functions `fir1` and `firls`. Panel C shows that the filter kernel computed by `fir1` is nearly the same as that produced by `firls` if you set the transition zones to zero and then smooth the filter kernel with a Hamming window.

14.6 Check Your Filters

- Poorly constructed filter
    - Poor match between frequency representation of filter kernel and the ideal filter that you specified
- Goodness of a filter quantified as:
    - Similarity between actual frequency characteristics of filter and frequency characteristics of the idea filter specified.

$$sse = \sum_{i=1}^{n} (ideal_i - actual_i)^2$$

SSE = sum of squared errors

N = number of frequencies specified in ideal filter

Ideal / actual = power spectrum

- SSE should be very close to zero. Don't use SSE above 1

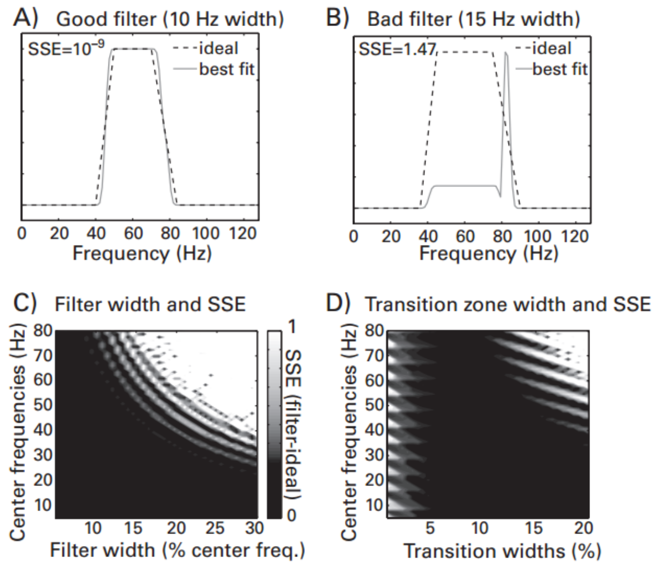


Figure 14.7
Examples of filter parameters that created poorly constructed filters. SSE is the sum of squared errors between the ideal response and the actual frequency response of the filter. In all cases the filter order was set to 200, and the Nyquist frequency was 128 Hz.

- First order and Nyquest frequencies were fixed for all simulations
- SSE changes with different filter orders
- Entire landscape changes with higher Nyquist frequency

14.7 Applying the Filter to Data

- Firls returns vector of length N + 1 (N is order)
  - This is the filter kernel
- Next, apply filter kernel to the data
  - Use filtfilt.
  - Inputs are filter kernel, scalar or vector of weighting coefficients, and data time series
    - Filter kernel is output of firls
      - Weighting coefficient = 1.0 (Unless want to apply a weighting)
- How does filtering work?
  - Filtfilt calls a function called filter
  - Filter introduces a phase delay in filtered signal
  - Phase delays can be reversed by refiltering the already-filtered data after reversing the filtered data in time.
    - (Phase delay in forward-going filter is reversed. Final result has no phase delays)
  - After filter is applied, result is:
    - Real-valued signal that contains only frequencies specified by frequency response of the filter kernel.
    - Hilbert transform then applied to bandpass-filtered signal
    - Result of transformation can be used to extract power and phase information

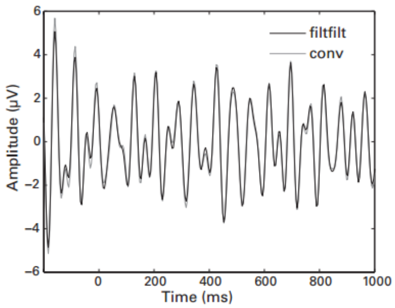


Figure 14.8
Convolution and zero-phase-shift filtering using the same filter kernel produce very similar results. A filter kernel was created using the Matlab function `firls`, and then this filter kernel was used to band-pass-filter EEG data using the Matlab function `filtfilt` (a zero-phase-shift filter; black line) and using convolution (gray line).

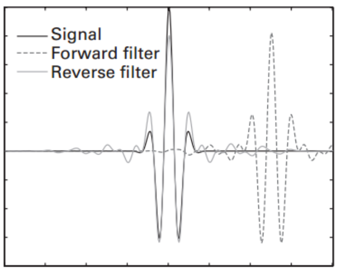


Figure 14.9
The Matlab function `filter` introduces a phase delay, which can be reversed by filtering the filtered data again after being flipped backward in time. The reverse filtered result should then be reversed again to arrive at the original orientation of time (in most cases, time originally goes forward).

14.8 Butterworth (IIR) Filter

- Similar to FIR filter construction and produces similar results. Reference matlab code for details.

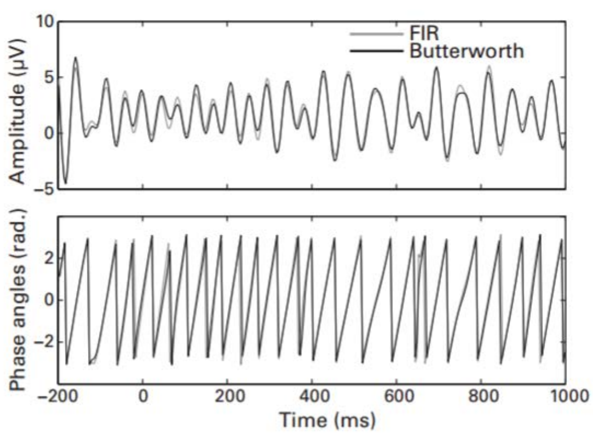
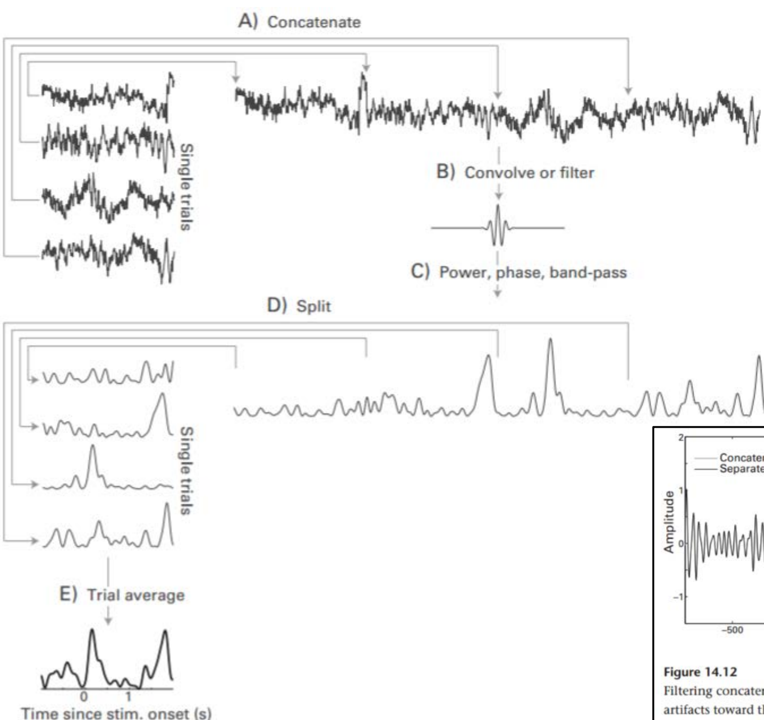


Figure 14.10
Comparison of results after FIR and IIR (fifth-order Butterworth) filter.

14.9 Filtering Each Trial vs. Filtering Concatenated Trials

- Faster to concatenate all trials into a long time series
    - Filter
    - Then cut out separate trials
    - INSTEAD OF FILTERING EACH SEPARATELY
- Edge artifacts pose an issue however
    - Sufficient buffer time at start/end of trial = subsided edge artifacts
- Don't filter using concatenated trials if:
    - Have short epochs around the time period you want to analyze
    - Ex. Epochs -200 to +800 around event of interest
    - Option: reflect data → then concatenate reflected trials



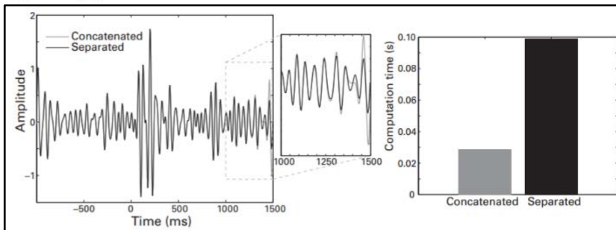


Figure 14.12
Filtering concatenated trials is more elegant and efficient than filtering single trials but produces minor artifacts toward the edges of the data (these are avoided in single-trial filtering due to signal reflection in the Matlab `filtfilt` function). The more trials you have, the bigger the increase in speed.

Figure 14.11
Illustration of concatenation procedure to improve performance of time-frequency decomposition. This method is appropriate for wavelet convolution and the filter-Hilbert method. Single trials are concatenated to produce one long time series (panel A), which is then convolved with a complex Morlet wavelet or is filter-Hilbertized (panel B). If convolution is used, you should zero-pad the concatenated time series so it has a length of a power of 2 to decrease FFT computation time. Power, phase, or the bandpass-filtered signal is then extracted from the entire time series (panel C), and thereafter the concatenated time series is reshaped back to individual trials (panel D) for trial averaging (panel E) or for any other analyses. Sharp transitions at the boundaries between trials will produce edge artifacts, but if there is sufficient buffer zone at the beginning and end of each trial, the edge artifacts will not contaminate the results (edge artifacts may also be present in nonconcatenated trials).

14.10 Multiple Frequencies

- With wavelet convolutions
  - Consider: examining activity over many frequency bands by looping over frequencies and re-filtering raw data into successive frequency bands
  - Frequency overlaps are acceptable (between 25 – 75 %)
    - Ex. 50% overlap: Upper band: 4-8Hz | Lower band: 6-10Hz
  - Increase in frequency → wider filters
    - Ex. Center frequency 6Hz (4.5 and 7.5 Hz bounds)
    - Center frequency 60Hz (45 and 75 Hz bounds)

14.11 A World of Filters

- Chapter 14 covers:
  - Constructing / applying bandpass filters
  - For use in combination with Hilbert transform for time-frequency analysis

14.12 Describing This Analysis in Your Methods Section

- Include enough info for others to replicate your analysis
  - Minimum
  - Maximum
  - Number of frequency bands
  - Pass Bandwidths of frequencies
  - Whether bandwidths changed as a function of frequency
  - Order of filter
  - Whether order changed as a function of frequency
  - Transition zones of filter shape
  - How filter kernel was created
  - If filters had a good fit to ideal shape
  - Parameters needed to be modified to due to poor filter construction
  - If tested range of filter parameters: report which range you tried and settings used for final analysis

15 Short-Time FFT

- Method for extracting time-frequency, power, and phase information
- Addresses issues with Fourier transform such as:
  - FT obscures time-varying changes in frequency structure of data
  - FT assumes data are stationary for time series duration

15.1 How Short-Time FFT Works

- Use FFT to extract frequency structure of brief segments of data time windows (rather than entire time series)
- Produces similar time-frequency map to that of complex Morlet wavelet convolution or filter-Hilbert method
- Can extract:
  - Time course of power at one frequency
  - Power spectrum at one-time point

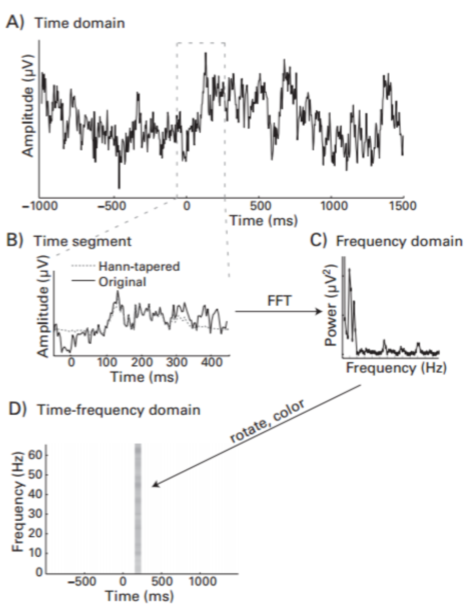


Figure 15.1
Overview of the short-time FFT method. From the time series data (panel A), a small segment of the data, comprising a few hundred milliseconds, is taken (panel B). A windowing taper is applied to that segment to minimize the possibility of edge artifacts, and then the Fourier transform of the tapered time series is taken (panel C). The power spectrum of that segment is then placed into a time-frequency space with the frequencies corresponding to that of the FFT and the time point corresponding to the center time point of the time segment from panel B. It is not necessary to perform these procedures separately for each trial because the Matlab function `fft` can accept a 2-D input (e.g., time by trials). Make sure you take the `fft` over time, not over trials. The FFT returns as many frequencies between DC and the Nyquist frequency as there are time points in the segment, although you can keep only the frequencies of interest, such as 4 Hz to 60 Hz. For resting-state data, the same procedure could be applied, but the FFT results would be averaged over time in panel D (e.g., see figure 2 in Allen and Cohen 2010).

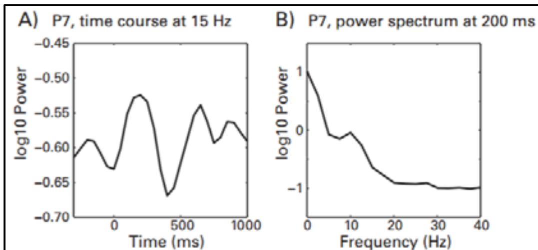


Figure 15.2
Power over time (panel A) and power structure at one time segment (panel B) obtained from the short-time FFT method. No baseline transformation was applied in these plots.

15.2 Taper the Time Series

- Taper data in segment before computing Fourier transform
    - Attenuates amplitude at beginning and end of time segment
        - Prevents edge artifacts from contaminating time-frequency results
    - Negative:
        - Attenuates valid EEG signal
        - Can be mitigated using temporally overlapping segments
- FOR short-time FFT:
    - Large buffer zones at necessary
- Types of Tapers:
    - Hann
        - Tapers data fully to 0 at beginning and end of time segment
            - Eliminates possibility of edge artifacts
    - Hamming
    - Gaussian
    - Kaiser
    - Cosine
    - Blackman

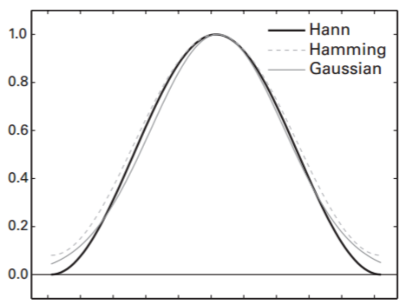


Figure 15.3
Different tapering functions often used for the short-time FFT in EEG data. The two features to look for in a taper are their width—narrow tapers attenuate more signal—and how close they are to zero at the beginning and at the end of the window.

- Extracting Requested Frequencies
    - Extract single closest frequency bin to each requested frequency
    - OR Take average of several frequency bins surrounding each requested frequency
        - Increases signal to noise ratio
        - Minimizes possibility that frequency estimate is driven by outliers or non-
        - representative data
    - Average power extraction
        - Results similar to Filter-Hilbert method
    - Average Gaussian-weighted neighboring frequencies
        - Results similar to complex wavelet convolution
    - Creating longer epochs than for ERP-only analysis
        - Because of time segment lengths
        - Ex. Estimate power at -200ms
            - Epochs start at -600ms
            - Assuming 800ms time segment for lower frequencies

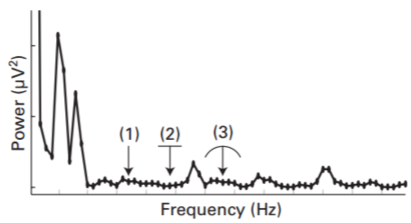


Figure 15.4
Illustration of different ways of extracting frequencies from the FFT of each time segment. Because the FFT returns more frequencies than you probably would want to include in the analyses, you can select "requested frequencies" from the full FFT result. To obtain the requested frequencies you can either select the single frequency bin closest to each requested frequency (arrow 1), average neighboring frequencies (arrow 2), or apply a distance weighting such as a Gaussian to neighboring frequencies to compute a weighted average (arrow 3). The latter two options increase the signal-to-noise ratio of the results of the short-time FFT. In practice any one of these methods can be used, but it should be consistently applied to all frequencies and all data within a study.

15.3 Time Segment Lengths and Overlap

- Shorter time segments
    - Better temporal precision
    - Poorer frequency precision and resolution
- Longer time segments
    - Better frequency precision and resolution
    - Poorer temporal precision
    - More frequencies can be extracted because freq. resolution of FT defined by # of points in time series
- Temp. vs. Freq. precision as function of frequency
    - Change time seg. Length as function of frequency
        - Lower freq. = longer time seg.
- Temporal resolution defined by data sampling rate – Not impacted by time segment length
- Time segment must be long enough to capture min 1 cycle of <u>lowest frequency</u>
    - Ex. Analyze 3Hz activity

        - Window = 333 ms long (for one window)
        - Preferably 667 or 999 ms to capture 2-3 cycles
            - 999 ms may be so long it fails to capture transient high-freq. activity
- Adapt short-time FFT for temporal vs. frequency precision as a function of frequency
    - Low freq. → longer time segment
    - Ex. Center point of 300ms
        - Compute FFT on data from -100 to 700 ms
        - Extract frequencies between 4-20Hz
    - Next, compute FFT on data from0 to 600ms
        - Extract frequencies between 20-40Hz (and so on)
- Short-time FFT and overlap between successive time segments
    - Temporal overlap useful because:
        - Improves temporal precision
        - Mitigates loss of signal from tapering
        - Smoothes time-frequency plots
    - Generally, 50-90% of the length of the segment is acceptable overall
        - Ex. Steps of 75ms for 300ms time segment = 75% overlap

15.4 Power and Phase

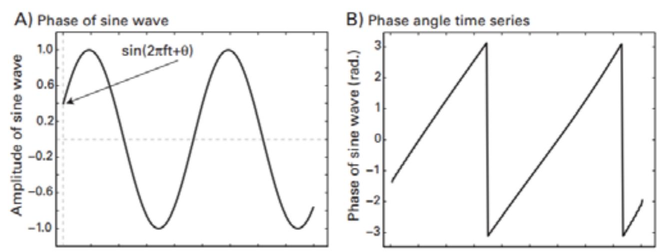


Figure 15.5
The difference between the phase values returned from the short-time FFT method (panel A) and wavelet convolution or filter-Hilbert (panel B). The short-time FFT returns one phase value per time window, which corresponds to the phase parameter of a sine wave. The phase angle time series, on the other hand, is a vector of phase angle values (one per time point) corresponding to the position along the sine wave at that time. These two values have different interpretations but can often be used in the same way in analyses, as shown in figure 15.6 and discussed chapter 19.

- Power values for short-time FFT
    - Same interpretation as wavelet convolution and filter-Hilbert
- Phase values for short-time FFT
    - Phase parameter of sine wave at each frequency
    - (y-axis intercept for the sine wave at zero on x-axis)
    - Can be used for:
        - Interatrial Phase Clustering
            - Measure of consistency of band-specific activity over trials
- Phase values for wavelet convolution / filter-Hilbert
    - Phase-angle-time series is an estimate of instantaneous phase value at each time point

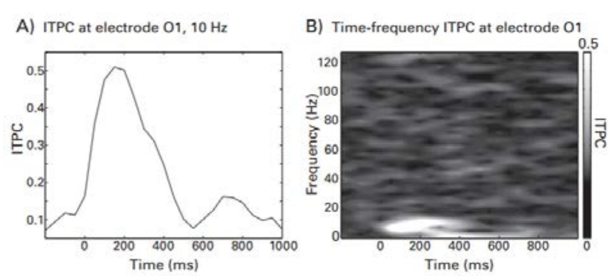


Figure 15.6
Intertrial phase clustering (ITPC), a measure of the consistency of phase values at each time-frequency point over trials, computed using the phase information from the short-time FFT method. Panel A shows a time course of ITPC at 10 Hz from electrode O1, and panel B shows a time-frequency plot of ITPC from O1. Details of the methods and interpretation of ITPC are presented in chapter 19.

15.5 Describing This Analysis in Your Methods Section

- If using short-time FFT, make sure to include:
    - Overlap between successive time segments
    - Number of frequencies extracted (requested frequencies)
    - Whether time segment length changed as a function of requested frequency
    - Whether neighboring frequencies were averaged together to increase signal-to-noise ratio
    - The function used to taper the data

16 Multi-tapers

- Extension of short-time FFT designed to increase signal-to-noise ratio of frequency representation
    - Applies several tapers that have slightly different temporal (and spectral) characteristics
- Especially useful when:
    - Low signal-to-noise ratio
        - (Higher-frequency activity)
        - OR (Single-trial estimates of power)
- Not useful for:
    - Frequencies lower than30Hz
        - High signal-to-noise ratio
        - Spectral smoothing (from multitaper) may impede frequency isolation
            - Ie. Activities from multiple frequency bands are averaged together

16.1 How the Multitaper Method Works

- Used to extract lower and phase information
    - Phase values from multitaper reflect an average phase from tapered data time series
        - Since phase value per frequency may change for each taper
    - Phase values are sine wave in time segment at each frequency
        - (NOT ongoing phase-angle time series)
- Begins similar to short-time FFT
    - First, one time segment is extracted from entire trial period
    - Data is multiplied by series of tapers (resulting in several tapered time series)
    - FFT of each tapered time series is taken
    - Resulting spectra are averaged together
- Difference between multitaper and short-time FFT
    - Multitaper uses more than 1 taper → and more than 1 FFT

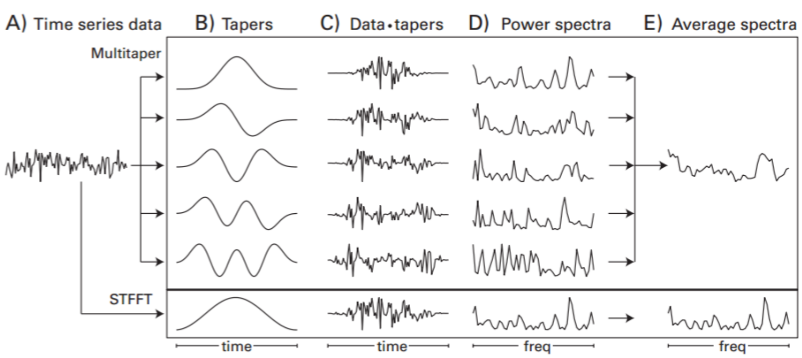


Figure 16.1
Overview of the multitaper method and comparison with the short-time FFT approach. The *y*-axes were arbitrary scaled to facilitate visual comparison. Note that the *x*-axes show time in columns A, B, and C and frequency in columns D and E. The dot between data and tapers in the label of panel C indicates the pointwise multiplication.

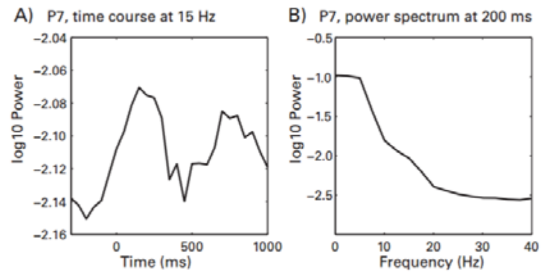


Figure 16.2
Same data as shown in figure 15.2, but the results were obtained via the multitaper method instead of the short-time FFT method.

16.2 The Tapers

- Discrete prolate spheroidal sequences (Slepian tapers)
  - (Matlab function dpss)
  - Orthogonal to each other
    - Dot product of any taper with another taper is 0
  - Have slightly different frequency characteristics
- More tapers = smoother results
- Useful for:
  - Higher frequency activity
    - Spectrum is broader than at lower frequencies
  - Account for individual differences between subjects

Ex. Two subjects have gamma-band peaks at 50Hz and 65Hz

Complex Morlet wavelet with 10 cycles has such high frequency precision. It Identifies unique bands of 2 subjects so precisely, that cross subject averaging fails to identify task-related gamma band activity at the group level.

- Multitapers produce temporal smoothing useful for:
  - Identification of high-frequency activity (cross-subject group level averaging)
  - Ex. Helps identify non-phase-locked gamma-power-increases
- How much smoothing is needed?
  - Smoothness vs. Accuracy tradeoff
    - Smoother result = more tapers = decreased accuracy of data's frequency characteristics
    - Smooth enough to increase signal-to-noise ratio
      - Not enough to lose sensitivity to detect time-frequency features
- Frequency bins
  - Multitaper returns as many frequency bins between 0 and Nyquist frequency as there are time points in each segment
    - Have option to request frequencies based on:
      - Single frequency bin
      - Average of neighboring frequency bins
  - Since Multitaper has spectral smoothing and averaging

        - Not usually necessary to further average over frequency bins when extracting power from Multitaper-averaged spectra

16.3 When You Should and Should Not Use Multitapers

- 3 situations when beneficial
    - 1. Have noisy data or small number of trials
        - Concerned about influence of noise on results
    - 2. Performing single-trial analysis
        - Analyze frequencies above ~30Hz
    - 3. Focus on high-frequency power
        - Above ~60 Hz
- Multitaper:
    - Increases signal-to-noise ratio of high-frequency activity
    - Temporal and frequency smoothing may facilitate cross-subject averaging
    - (Matlab toolbox fieldtrip – good one for Multitaper)
- 2 situations when not beneficial:
    - 1. Focus analysis on frequencies below 30Hz
        - Smoothing at low frequencies makes it difficult to isolate discrete time-frequency events
        - (Especially if events are close to each other in time-frequency space)
    - 2. Perform analysis where precise timing of high-frequency activity is required
        - Temporal precision of Multitaper is relatively low (compared to wavelet/Hilbert), timing of high-frequency activity is more difficult to determine
    - Could be useful for finding non-phase-locked power
    - Detrimental for cross-frequency coupling
        - Because temporal precision of high-frequency power should be high as possible

16.4 Multitaper Framework and Advanced Topics

- For more sophisticated applications:
    - Varying lengths of time segments
    - Number of tapers
    - Center frequency of spectral representation of tapers (as function of central frequencies)
- Advanced methods for estimating appropriate frequency smoothing and # of tapers to use
    - Adaptively weighting contribution of different frequency bands to optimize averaging
    - Performing statistical evaluation of whether spectral peaks are statistically significant
- More info:
    - *Spectral Analysis for Physical Applications: Multitaper and Conventional Univariate Techniques* (Percival and Walden 1993)

16.5 Describing the Analysis in Your Methods Section

- Mention the following:
    - Length of time segment
    - Number of frequencies extracted
    - Number of Slepian tapers
    - Resulting smoothing
    - List default and non-default parameters used

17 Less Commonly Used Time-Freq. Decomposition Methods

17.1 Autoregressive Modeling

- Values of signal are predicted from previous values of that signal
- Order = # of previous values
    - Must be carefully selected
    - (too large or too small causes poor fits of model to data)
- Rhythmic patterns in temporal lags can be converted from sample points to frequencies in Hz
- Estimate frequency-band-specific power from autoregressive model
    - Examine auto covariance as a function of lag
        - Strength is related to power at that frequency band
- Advantage:
    - Frequency not limited by number of data points in time segment
    - (share by wavelet convolution / filter-Hilbert method too)
- Infrequently used. Replaced with autoregressive model for time-frequency analysis

17.2 Hilbert-Huang (Empirical Model Decomposition)

- Phase, power, and frequency can be extracted
- Detecting time-frequency events in nonstationary data
    - Doesn't rely on covariance with template procedures
        - Unlike: wavelet convolution, filtering, and FT
- Empirical Mode Decomposition
    - Decomposes raw EEG into series of fundamental components (intrinsic mode functions)
    - 1. Identifies local minima and maxima
    - 2. Creates new time series by interpolating across local minima/maxima
        - After some adjustments (sifting) – *Sweeney-Reed and Nasuto 2007*
    - 3. Subtract mean from original signal (intrinsic mode function)
    - 4. Repeat steps 1-3 until only a few min/max left
- Advantages:
    - Don't need to specify which frequencies to search for
        - Frequencies naturally present in data will emerge
    - Identifies changes in instantaneous frequency in data that is not stationary (by scaling temporal derivative of phase by sampling rate)
- Disadvantages:
    - Frequency structure is likely to differ across subjects
    - Result may be influenced by low-pass filtering data

17.3 Matching Pursuit

- An adaptive algorithm that (like Hilbert-Huang transform) is based on decomposing signal into more basic components
  - Unlike Hilbert-Huang: Does not involve a template-matching procedure
  - Findings which template from the “dictionary” matches the data best (at that time window)
    - Computes residuals between data and all templates
- Templates (atoms)
  - Often Morlet wavelets
  - Or similar tapered frequency-band-specific signals
- Advantageous:
  - Rapid changes in frequency expected of experiment

17.4 P-Episode

- Occurrence and duration of oscillatory events
  - 1. Filters data into frequency bands (Wavelet conv. Or filter-Hilbert)
  - 2. Detects whether power fluctuation exceeds an amplitude threshold
    - Duration threshold can also be applied
  - Threshold computed from
    - All time points in experiment
    - OR Baseline period
    - OR Control condition
- Figure below: Threshold as 95% of distribution of power at all time points from all trials

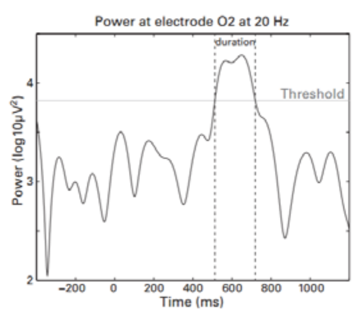


Figure 17.1
Illustration of the p-episode technique. The power time course is plotted from trial 50 (this trial was selected for illustration based on visual appearance of the results).

- Advantages:
  - Detects temporal duration of band-specific increase in power
  - (Most time-frequency decomposition focuses on amplitude)
  - Concerned with duration of power increase (at single-trial level)
  - Concerned with detecting high-frequency power
  - Ex. Gamma power bursts that aren’t: time-locked or phase-locked to time=0
    - Bursts may not survive cross-trial / cross-subject averaging
  - P-episode has increased sensitivity to brief but high-amplitude events

- Limitations:
    - 1. Based on detection of supra-threshold events
        - Threshold influences results
    - 2. Excessively noisy data isn't great
        - Noise produces transient increases in power
        - Computing condition differences should help with this instance
    - 3. Increase in power with modest amplitude, but consistent across trials might not be detected
        - Would be detected in trial averaging in ex. Wavelet convolution

17.5 S-Transform

- Address limitation of short-time FFT
    - Has limited sensitivity for detecting transient events that are shorter than the lenfth of the FFT time segment
- Same as a wavelet convolution: but different kernel
- Kernel
    - Tapered sine-wave (similar to Morlet wavelet)
- Taper
    - Gaussian-like shape (differs slightly from Gaussian)
- Selecting specific number of wavelet cycles
    - Produces identical results as Morlet wavelets
- Disadvantage:
    - Does not contain a parameter to change width of wavelet as function of frequency

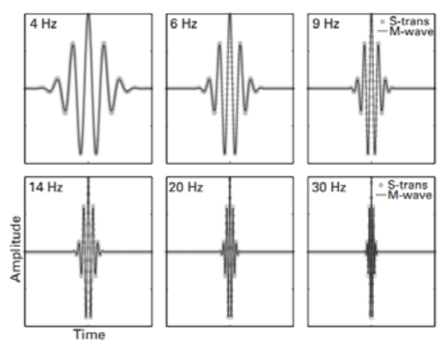


Figure 17.2
Comparison of the real component of the S-transform ("S-trans"; gray dots) and the Morlet wavelet ("M-wave"; black lines) for several frequencies. Morlet wavelets were created using six cycles for all frequencies.

18 Time-Frequency Power and Baseline Normalizations

- Extract estimates of time-varying frequency-band-specific power from EEG data
    - Wavelet convolution
    - Filter-Hilbert
    - Short-window FFT
    - Multitaper
- Learn to link fluctuations in power over time and frequency to task events
- Convert time-frequency power data to a scale amenable to qualitative visual inspection and quantitative statistical analysis

18.1 1/f Power Scaling

- Frequency spectrum shows decreasing power at increasing frequencies
- EEG time-frequency power obeys 1/f phenomenon
    - Power at higher frequencies has smaller mag. Than power at low frequencies
- 5 Limitations
    - 1. Difficult to visualize power across large range of frequency bands
    - 2. Difficult to make quantitative comparisons of power across frequency bands
    - 3. Aggregating effects across subjects can be difficult with raw power values
        - Individual differences in power are influenced by:
            - Skull thickness
            - Sulcal anatomy
            - Cortical surface area recruited
            - Recording environment

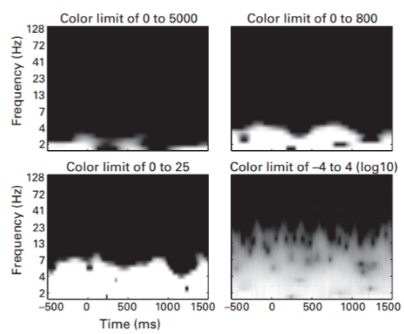


Figure 18.2
Time-frequency power plots showing that without any correction, no color scaling works optimally for all frequency bands. Taking the logarithm to base 10 of the data (log10; lower right panel) helps but does not fully eliminate the difference in scaling over frequencies. Furthermore, it is difficult to distinguish background (dynamics that do not change before and after time = 0) from task-related (dynamics that change as a function of stimulus onset) power.

    - 4. Task related changes in power can be difficult to disentangle from background activity
        - Particularly case for frequencies that tend to have higher power or frequencies that tend to have higher power during baseline periods (ie. Alpha over posterior parietal and occipital electrodes)
    - 5. Raw power values are not normally distributed
        - They cannot be negative
        - Strongly positively skewed
        - Limits ability to apply parametric statistical analysis to time-freq. power data

18.2 The Solution to 1/f Power in Task Design

- Use several kinds of baseline normalizations. All share following 4 advantages:
    - 1. Transform power data into same scale. Allows to compare (visually and statistically) results from different frequency bands, electrodes, conditions, and subjects
    - 2. Since normalization computed with respect to baseline, task-related time-frequency dynamics become disentangled from background or task-unrelated dynamics
    - 3. Baseline normalization put all power results into a common and easily numerically interpretable metric
    - 4. Since baseline-normalized power data are normally distributed, parametric statistical analysis may be appropriate. It facilitates quantitative group-level analyses. Facilitates integration with other kinds of data (behavioral performance, questionnaire results, and demographics)

18.3 Decibel Conversion

- Ratio between strength of one signal (frequency-band-specific power) and the strength of another (baseline level of power in same frequency band)
- Bel
    - Base unit
    - Logarithm of a ratio of numbers

$$dB_{tf} = 10\log 10\left(\frac{activity_{tf}}{\overline{baseline}_f}\right)$$

- Baseline(with bar above) = baseline time period
    - No t subscript (all time points within a frequency band use the baseline period)
- t = time
- f = frequency points
- After decibel conversion:
    - Shows change in power relative to baseline
- Baseline and activity equally affected by 1/f scaling
    - Any band-specific activity that’s constant over time (including BG activity) is removed
- Baseline
    - Time period before start of trial (no task related processing expected)
    - Generally, -500 to -200 ms before trial onset

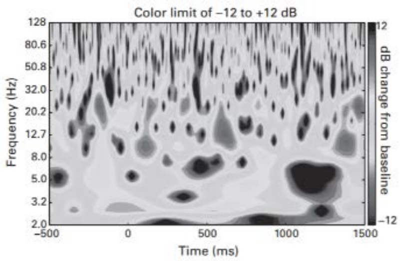


Figure 18.3 (plate 4)
Same time-frequency results that were shown in figure 18.2 but now decibel-transformed relative to a pretrial baseline period of –500 to –200 ms. Note that power across the entire range of frequencies can be compared.

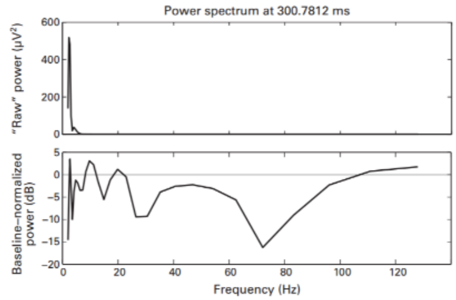


Figure 18.4
Comparison of raw and decibel-converted power at one time point. A horizontal line at 0 dB facilitates comparison of activity with respect to baseline power.

- Method for decibel conversion (SUPER IMPORTANT)
  - Average trials together
  - Then transform to decibels
- Figure 18.3 Plots
  - Symmetric color scaling (-1.5 to 1.5)
  - Color bar with numerical labels for min / max
  - Label color bar (or create legend to describe color bar – simplify interpretation to reader)

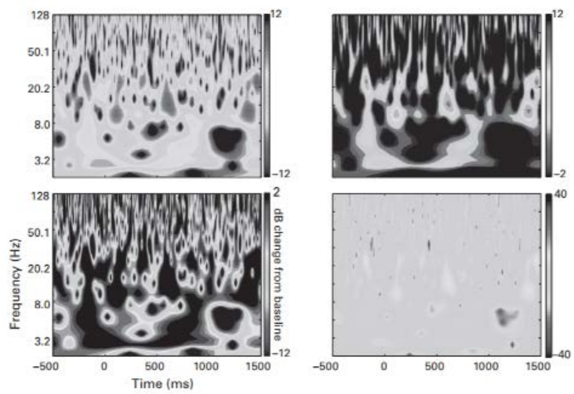


Figure 18.5 (plate 5)
Different color scaling applied to the same time-frequency power results. Symmetric color scaling should be preferred in most situations. Asymmetric color scaling can highlight or obscure different features of the data. Values in the color bars correspond to decibel change from a pretrial baseline of –500 to –200 ms. These data are taken from one trial and one electrode.

18.4 Percentage Change and Baseline Division

- Baseline normalization method
  - Results interpreted as changes in power relative to power during baseline

$$prctchange_{tf} = 100\frac{activity_{tf} - \overline{baseline_f}}{\overline{baseline_f}}$$

- Percentage change value at each time frequency point is computed relative to baseline power that is frequency band specific

18.5 Z-Transform

- Corrects for 1/f power law scaling
- Transforms power data to be comparable across frequencies, electrodes, conditions, and subjects
- Power scaled to standard deviation units relative to power data during baseline period
- Z values can be easily converted to p values
    - Ex. Z=1.96 → two tailed p = 0.05 (where n = # of time points in baseline period)
    - Denominator = standard deviation

$$Z_{tf} = \frac{activity_{tf} - \overline{baseline_f}}{\sqrt{n^{-1}\sum_{i=1}^{n}\left(baseline_{if} - \overline{baseline_f}\right)^2}}$$

- Differs from decibel and percentage change:
    - Based on both average baseline power and (+ bonus) standard deviation of baseline power over time
- Disadvantages
    - Estimates of stimulus-related power may be adversely affected by highly variable data in baseline
        - Bad with lots of noise or few trials

18.6 Not All Transforms Are Equal

- Decibel conversion= logarithmic scale
- Percentage change = linear scale
- Both ^^^ produce similar results when decibel and percentage change are close to 0
    - o Scales differ as move away from 0
- Decibels become steeper at increasing negative values compared to percentage change values
    - o Ex. (Decibel: -2 to +2 → Percentage change: -36.9% to +58.5%)

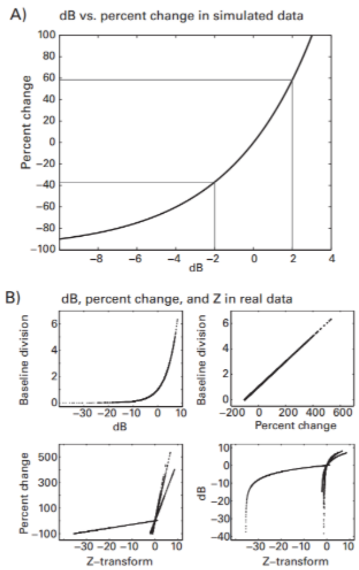


Figure 18.6
Comparisons of different time-frequency power baseline normalization methods. Panel A shows a comparison of percentage change and decibels, based on simulated data. Panel B shows the relationships among baseline division, percentage change, decibels, and Z-transform in real data. The Z-transformed data contain two frequencies with large negative outliers due to large variance during the baseline time period.

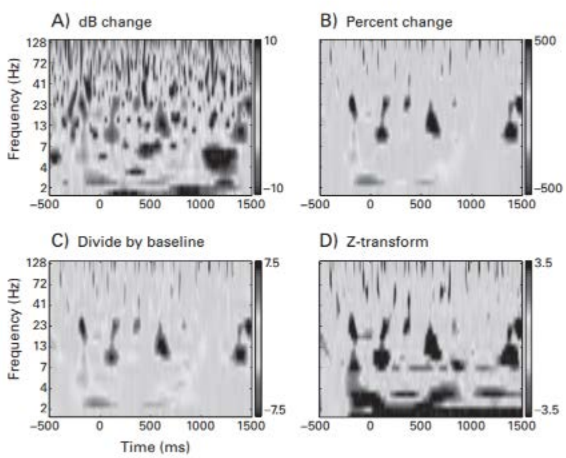


Figure 18.7 (plate 6)
Results from the same data and same analysis, but with different baseline normalization methods applied. As with previous figures in this chapter, the color scaling is larger here than what you would normally expect because data from one trial were used.

18.7 Other Transforms

- Other methods to normalize data based on:
  - Empirical mode decomposition
  - Frobenius norm
  - Other matrix norms
- Any procedure equaooy applied to all conditions, electrodes, and subjects and that addresses the limitations of raw power could be justifiable
  - Avoid linear baseline subtractions for time-frequency power
    - (doesn't address 1/f power-law scaling)

Mean vs. Median

- When mean across trials is inappropriate:
  - 1. Power values cannot be negative
    - Distribution of power across trials is likely to be positively skewed
  - 2. Outliers and other non-representative data are more likely to be larger than the mean (will bias toward larger values)
- Low trial count and noisy data → Outliers can skew results to incorrectly suggest a condition difference
- Median
  - Useful where outliers may influence the mean
  - # where ½ values are higher and ½ are lower

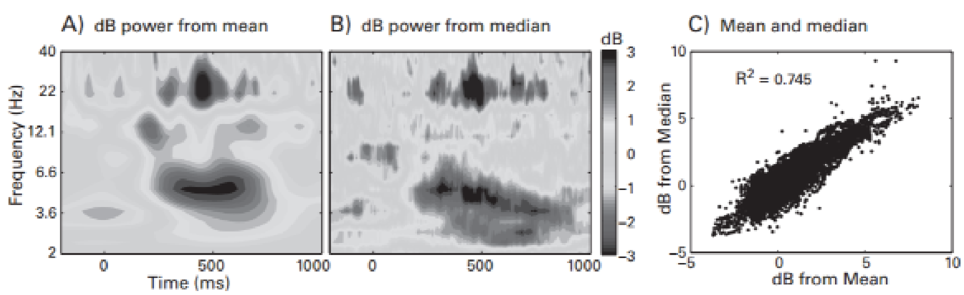


Figure 18.8 (plate 7)
Time-frequency power plots of the same data using mean (panel A) or median (panel B) to combine data across trials. Panel C shows the correlation between the results using mean and using median.

- Median instead of mean usually just when few trials and noisy data

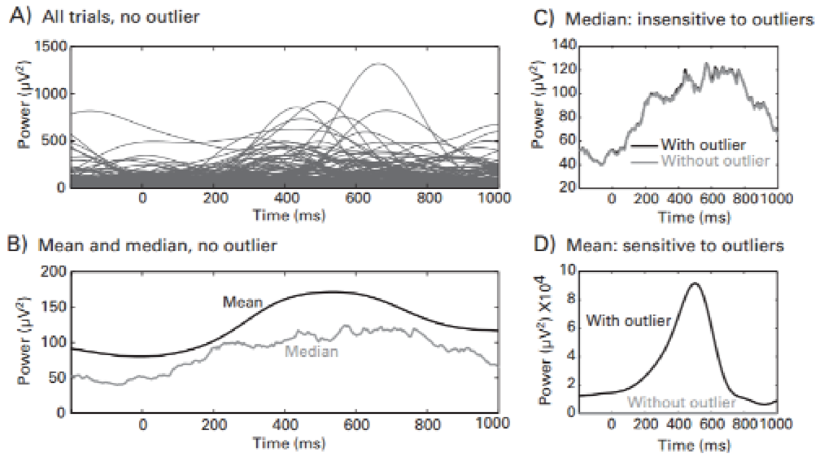


Figure 18.9

Panel A shows power from one frequency band and one electrode over all trials (each line is a trial). Panel B shows the mean and median power over all trials. Thereafter, one outlier trial was created by multiplying the time series data from one trial by the number 100. Panel C shows that the median power time course is unaffected by the presence of a single outlier trial. In contrast, panel D shows that the mean is very sensitive to the presence of an outlier. The line corresponding to the mean power without the outlier seems to be a flat gray line at the bottom of the plot; this is due to the large difference in scaling for the mean with the outlier included.