4
Time-frequency decomposition methods

Memory-related neural oscillations

Let us again consider the neural and cognitive processes that unfold between the perception of a test item and its classification as "old" or "new". Such a recognition memory test requires the interplay of perceptual, memory, decision, and response processes. Some of these processes are tightly linked to stimulus or response events (e.g., early perceptual processes unfolding directly after the presentation of a target stimulus or motor processes immediately preceding a response). Our focus in this book is what happens between these early perceptual and late response processes. It is likely that the processes allowing us to distinguish previously studied from new items are less tightly linked to easily measured time points and thus averaging neural activity to compute event-related potentials could obscure relevant signals.

To identify neural correlates of behavior that are only weakly related to stimulus and response events we need some way of measuring the signal in a given time series. Broadly speaking, this can be done in either the time or the frequency domain. In the time domain, we could measure basic properties of the voltage series, such as its mean and standard deviation. However, brain activity often exhibits rhythmic components, or oscillations, and these are not easily measured in the time domain. In Chapter 2 we mentioned several examples of rhythmic activity that may be related to cognitive processes. For example, the 4-8 Hz theta rhythm appears to be an important correlate of human spatial cognition, and in the timing of neural activity (?, ?). To determine whether memory-related variables influence the presence of theta activity we need some way of extracting rhythmic components from a complex time series of data. The basic methods of time-frequency analysis introduced below have wide-ranging applicability throughout the physical and life sciences and we will draw upon these methods extensively in subsequent chapters.

The Fourier Transform

One of the most important mathematical insights in history came to us from the French mathematician Jean Baptiste Joseph Fourier (1768 - 1830). In 1807 Fourier presented for review his work *Théorie analytique de la chaleur* (*The*

Analytical Theory of Heat) in which, among other things, he claimed that any continuous periodic signal could be perfectly represented as an infinite sum of sinusoids of different frequencies and phases. This notion built upon the work of other mathematicians (e.g., Leonhard Euler) but Fourier is credited for recognizing its broad impact and potential utility. Fourier's idea is so important, and radical, because it means that in principle it is possible to take any such signal and decompose it into its constituent parts, allowing us to understand complicated and seemingly irregular signals in terms of well-behaved sinusoidal waves. In Fourier's case, he used this idea to solve the heat equation, for which there existed at the time no general solution, although solutions did exist for special cases in which the heat source was a regular function such as a sinusoid. Fourier's insight was to represent complicated heat source functions as the sum of simple sinusoids, which allowed him to define a general solution to the heat equation as the sum of solutions to the simpler functions. As is often the case in academia, Fourier's new idea faced resistance from established figures, including Joseph Louis Lagrange (1736-1813), who argued that it was not possible to use sinusoids to perfectly represent functions with discontinuous slopes such as square and sawtooth waves (while this is true, it is possible to derive very close approximations). Luckily for us, Fourier was able to publish his work 15 years later and *Fourier analysis* is now a foundational tool across branches of the sciences, engineering, and mathematics.

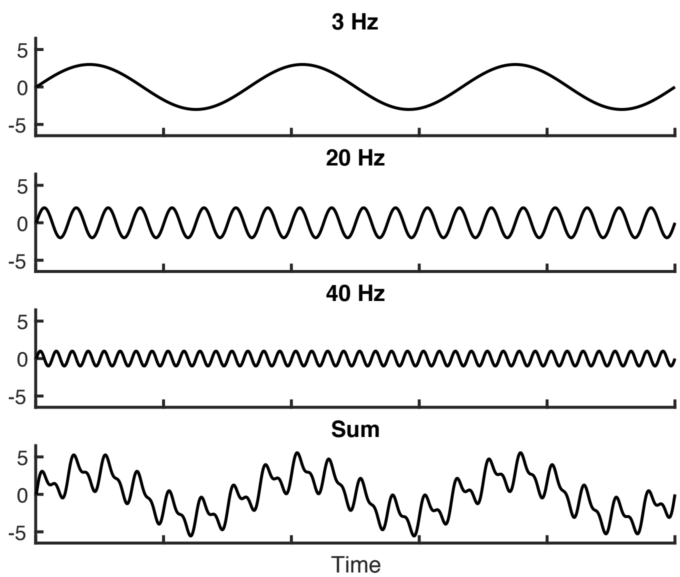

Figure 4.1: Constructing a signal by summing sine waves. The sum of only three sinusoids at 3, 20, and 40 Hz with amplitudes 3, 2, and 1 (arbitrary units) produces a signal of some complexity. Fourier showed that signals of any complexity can be arbitrarily well approximated as a sum of sine waves.

Figure 4.1 shows an example of building a complex periodic signal out of the sum of simple sinusoids. The large amplitude low frequency (3 Hz) component contributes the most to the underlying shape of the function, while the smaller amplitude high frequency (20 and 40 Hz) components produce smaller fluctuations in the signal amplitude. In this idealized example it is relatively straightforward to see how superposition of sinusoids can lead to functions that vary in complicated ways. In this example, we have also approached the task from the easier direction: constructing a new signal from known simple elements. The value of the Fourier transform is in its ability to go in the reverse direction: to reveal the underlying structure of signals that are much more difficult to intuitively understand. By uncovering the components of signals, we can develop tools to enhance the contributions of components that are interesting and attenuate the contributions of components that are not. In the case of audio signals, speakers' voices may be transmitted via communication systems that introduce undesirable noise. If the voices and noise occur at largely distinct frequencies, Fourier analysis can be used to filter the signal in order to remove the contribution of the noise. Another application is in the domain of image processing. Visual images can be thought of as two-dimensional signals, and decomposing image signals with methods based on the Fourier transform can provide information about what underlying frequencies contribute the most the most information to the appearance of the full image. This is the basis of image compression methods which seek to minimize the storage requirements for digital images while maintaining the fidelity of the picture being represented. Thus, the ability to decompose complicated signals into constituent components is of great general utility—in this and later chapters we will make similar use of Fourier-based methods to derive insights into the mechanisms of memory function in the brain.

Sinusoids

Before introducing the Fourier transform we first review the various mathematical notations for sinusoids, which are defined by three parameters: amplitude (A), one-half the peak-to-trough size of a sinusoidal wave; frequency (ω), the speed at which it oscillates; and phase (ϕ), the wave's offset relative to an arbitrary starting time $t = 0$:

$$A \sin(\omega t + \phi) \tag{4.1}$$

Alternatively, one can write the same wave as a function of *cosine*:

$$A \cos\left(\omega t + \phi + \frac{\pi}{2}\right) \tag{4.2}$$

where the cos wave is phase-shifted by $\frac{\pi}{2}$ relative to the sin wave. Although $\sin \omega t$ and $\cos \omega t$ are the standard notations for sinusoidal waves, any sinusoid can be written as the sum of a sin and cos of the same frequency:

$$A \sin(\omega t + \phi) = B \sin \omega t + C \cos \omega t \tag{4.3}$$

where $M = \sqrt{b^2 + a^2}$ and $\phi = \text{atan2}(b, a)$. The function atan2 is the version of the arctangent function which correctly handles negative values of a, providing the full range of $-180°$ to $180°$, or $-\pi$ to π, as possible phase values.

Sinusoids are often depicted as waves oscillating over time, but they can also be thought of as representing the motion of a point around the circumference of a circle. If we consider a circle of radius r centered on the origin, we can draw a vector $\mathbf{v}$ from the origin to the edge of the circle. If we imagine rotating $\mathbf{v}$ at frequency (speed) ω counterclockwise around the circle, the projection of $\mathbf{v}$ onto the x and y-axes over time traces out the $A \cos \omega t$ and $A \sin \omega t$ functions, respectively (where $A = r$).

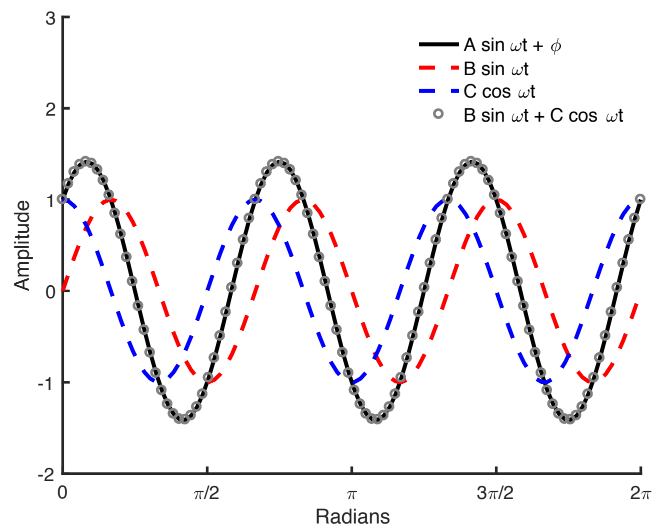

Figure 4.2: Sinusoid represented as the sum of sin **and** cos **waves.** A graphical representation of Equation 4.3 shows that the sum of $B \sin \omega t$ (dashed red line) and $C \cos \omega t$ (dashed blue line), shown in gray circles is equal to the sinusoid $A \sin \omega t + \phi$ (solid black line). Here, $\omega = 3\ Hz$.

Complex Numbers

By thinking about sinusoids as being composed of both a cos component and a sin component, and as representing rotation around a circle, we are primed to understand *complex sinusoids* and the important role they play in the Fourier transform and other methods of decomposing EEG signals into their spectral components. We can simplify much of the mathematics required for Fourier analysis by using the notation of complex numbers to represent sinusoids. Here, we briefly review complex notation.

A complex number is the sum of two parts, taking the form $a + bi$ where a is the *real* part, b is the *imaginary* part, and i is the imaginary number $\sqrt{-1}$.[1] The operators $\Re[]$ and $\Im[]$ are used when separating a complex number into its real and imaginary parts. Thus, the real part of the complex number $3 + 4i$ would be extracted by writing $Re[3 + 4i] = 3$, while the imaginary part would be extracted by writing $Im[3 + 4i] = 4$. In equations, complex numbers are usually represented by a single variable, i.e. $A = 3 + 4i$, which highlights an important property of complex numbers, namely that they are a convenient way to represent two quantities within a single variable.

[1] The traditional notation from mathematics is to use i to denote $\sqrt{-1}$, while in electrical engineering j is used to distinguish $\sqrt{-1}$ from *current*, which is represented by i. In this book we will use the convention from mathematics.

Complex numbers can be represented in terms of their real and imaginary parts, a and b, by using a two-dimensional plane in which the x-axis is the *real* axis and the y-axis is the *imaginary* axis. Figure 4.3 shows the number $A = 3 + 4i$ graphed in this way. The red dot shows the position of A as a point in the 2D plane and shows how it is composed of real part $a = 3$ and imaginary part $b = 4$. We can also see from Figure 4.3 that the same complex number (i.e. the same point in the complex plane) can be represented by two alternative parameters, M and ϕ. M is the *magnitude* of the complex number and ϕ is the angle subtended by M and the real axis. M and ϕ can be found for any complex number $A = a + bi$ using basics of trigonometry:

$$\begin{aligned} M &= \sqrt{Re[A]^2 + Im[A]^2} = \sqrt{a^2 + b^2} \\ \phi &= \operatorname{atan2}(b, a)\,. \end{aligned} \tag{4.4}$$

Expressing a complex number with M and ϕ is called using *polar notation*. We can also define the relation between polar notation and the notation of the complex plane, $a + bi$, also referred to as *rectangular notation*:

$$\begin{aligned} Re[A] &= M\cos(\phi) \\ Im[A] &= M\sin(\phi) \end{aligned} \tag{4.5}$$

By comparing Equations 4.4 and 4.5 to Equation 4.3, you may begin to see how a sinusoid could naturally be represented as a location in the complex plane, but you may also wonder why it would ever be useful to take a single sin or cos function and split it into the sum of both a sin and cos. As we shall see in a moment, there is a mathematical relationship between complex numbers represented as sinusoids and complex *exponential* functions that is foundational to analysis of sinusoidal functions in science and engineering.

By performing some substitution, we see that

$$a + bi = M\cos(\phi) + iM\sin(\phi) = M(\cos(\phi) + i\sin(\phi))\ , \tag{4.6}$$

which gives us the relation between the rectangular and polar notations of a complex number. We can now introduce an important definition in complex analysis, Euler's Relation:

$$e^{i\phi} = \cos(\phi) + i\sin(\phi) \tag{4.7}$$

One of the reasons that Euler's Relation is a such a critical tool in science and engineering is that, by allowing us to recast sinusoids as exponentials, Euler's Relation simplifies many of the mathematical operations needed to analyze sinusoidal functions.[2] For example, multiplying two sinusoids requires the application of a number of trigonometric identities, whereas doing so with exponential functions requires only adding the angles of the respective sinusoids and multiplying their magnitudes:

$$Me^{i\phi}Ne^{i\theta} = MNe^{i(\phi+\theta)} \tag{4.8}$$

As we will see in the next section, the complex exponential, $e^{i\phi}$ plays an important role in methods for spectral decomposition such as the complex Fourier transform.

The Fourier Transform

The Fourier Transform of the timeseries signal contained in an array x is given by

$$X[k] = \frac{1}{N}\sum_{n=0}^{N-1} x[n]e^{-i2\pi kn/N}\ , \tag{4.9}$$

where N is the number of data points in the array, k is the frequency being evaluated in units of the inverse time length of the array, and $X[k]$ is the resulting Fourier coefficient of the timeseries $x[n]$ at frequency k. The output $X[k]$ is called the frequency domain representation of the signal $x[n]$ at frequency k. Another way to think about $X[k]$, which will become clearer later on, is that it is the magnitude of a sinusoid of frequency k that is present in $x[n]$.[3]

Equation 4.9 expresses the basic idea of the Fourier transform: to derive the frequency domain representation $X[k]$, the time-domain signal $x[n]$ is

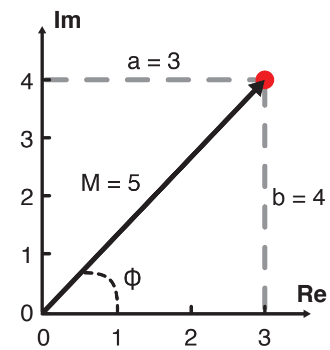

Figure 4.3: The complex plane. The complex number $A = 3 + 4i$ shown on the complex plane. The x- and y-axes are used to plot the *real* and *imaginary* parts of a complex number as a single point on 2D space (red dot). For $a = 3$ and $b = 4$, the magnitude of the vector from the origin to A, M, is equal to $\sqrt{3^2 + 4^2} = 5$. The angle subtended by M and the real axis is ϕ, which will become important later in our discussion of how to extract phase information from oscillatory EEG activity.

[2] For the interested reader, Equation 4.7 can be shown to be true using the Taylor series expansion of the exponential function.

[3] The definition of the Fourier transform given by Equation 4.9 is called the complex *discrete Fourier transform* (DFT). This is the Fourier transform that is used in spectral analysis of discrete signals, i.e. signals such as EEG that are recorded and sampled using digital electronics.

multiplied point-wise by a sinusoid of frequency k defined over the length of $x[n]$ (N samples). The sum of this point-wise multiplication is then taken over all N samples and normalized by $\frac{1}{N}$. Readers with some background in linear algebra will be familiar with another name for the point-wise multiplication/summation operation: the *dot product*. Just as the dot product provides a measure of the "similarity" of two vectors in space, the Fourier transform gives a measure of the similarity between a sinusoid of frequency k and your timeseries signal.

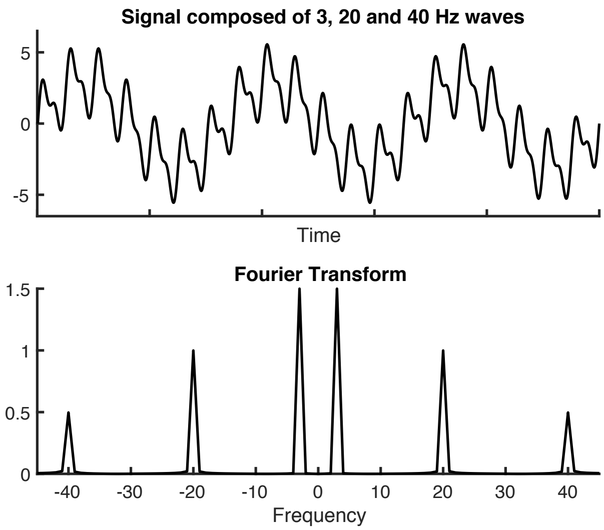

Figure 4.4: The output of the Fourier transform. The top panel shows a signal created by adding together sinusoids of frequencies 3, 20, and 40 Hz (see Figure 4.1). The bottom panel shows the Fourier transform of this signal and illustrates how the transform recovers the frequencies and amplitudes of the sinusoids from which the signal is composed (x-axis restricted to ± 40 for the purposes of visualization). However, this representation does not show phase information, which we discuss in more detail in chapter on connectivity.

The top panel of Figure 4.4 shows a signal created by adding together 3, 20, and 40 Hz sinusoids. The result of the Fourier transform of this signal is plotted in the bottom panel. The frequency of the recovered sinusoid is shown on the x-axis, while its amplitude is on the y-axis. Since the summed signal was constructed exactly from sinusoids at 3, 20 and 40 Hz (and nothing else), there are peaks at each of those frequencies, while all other frequencies are zero. The height of the peak for each frequency corresponds to its amplitude in the original signal. You will notice that the x-axis contains both positive and negative frequencies—this is a consequence of the use of the complex exponential to compute the Fourier transform and simply means that a sinusoid of frequency k that contributes to a signal will have its energy (amplitude) split between the k and $-k$ sides of the x-axis when its Fourier transform is graphed. In practice, the negative side of the spectrum is often omitted from visualization and the positive side is plotted with the amplitudes doubled to account for the contribution of the negative frequencies. In Figure 4.4 we graphed the Fourier transform of our signal for ω in the range $[-45, 45]$, but in fact the Fourier transform of a signal $x[n]$ of length N is defined over k equally-spaced frequencies between $[-\frac{N}{2}, \frac{N}{2}]$. What happens if you would like to extract frequency information about a particular frequency k that falls between the resolution provided by the transform? You can increase the length of the signal by adding zeros to the beginning and end of the array, which will increase the number of data points, N, in your signal without changing the information contained in the timeseries.

The Sampling Theorem

While zero-padding will increase the frequency resolution of the resulting Fourier transform, it will absolutely not change the range of frequencies that it is possible to recover from your digitally-sampled timeseries. That is, let's say that you collect EEG data from a participant in a memory experiment and you are interested in using the Fourier transform to investigate the amplitude of a 250 Hz oscillation. Will you be able to use the Fourier transform to decompose the EEG timeseries to extract the amplitude of this sinusoid? The answer to this question depends on the *sampling rate* you used when you recorded your EEG timeseries.

Like many signals that one encounters in nature, the voltage fluctuations that are measured by EEG electrodes can in principle vary continuously over time and take on an infinite number of values. Unlike the natural world, digital electronic devices like computers cannot represent an infinite set of continuously varying values and must instead store information in finite and discrete form. Therefore, any continuous signal that is recorded and stored in a digital device is a discrete and finite representation of the original continuous signal. The frequency at which values of the continuous signal are digitally recorded is called the sampling rate of the signal and is typically expressed in units of hertz (Hz; i.e., number of samples per second).

The sampling rate is the key parameter that determines how closely the digital signal resembles the original continuous signal that it represents. How high does the sampling rate need to be in order to produce a digital signal that is "close enough" to the original continuous signal? The *sampling theorem* provides an answer to this question.[4] The sampling theorem states that a continuous signal can be perfectly reconstructed from its digital representation if it contains no information at frequencies greater than one-half the sampling rate of the signal. The minimum sampling frequency that allows perfect reconstruction of a signal is called the *Nyquist* rate.

[4] The sampling theorem is sometimes referred to as the Shannon sampling theorem or the Nyquist sampling theorem after the authors who are credited with its discovery.

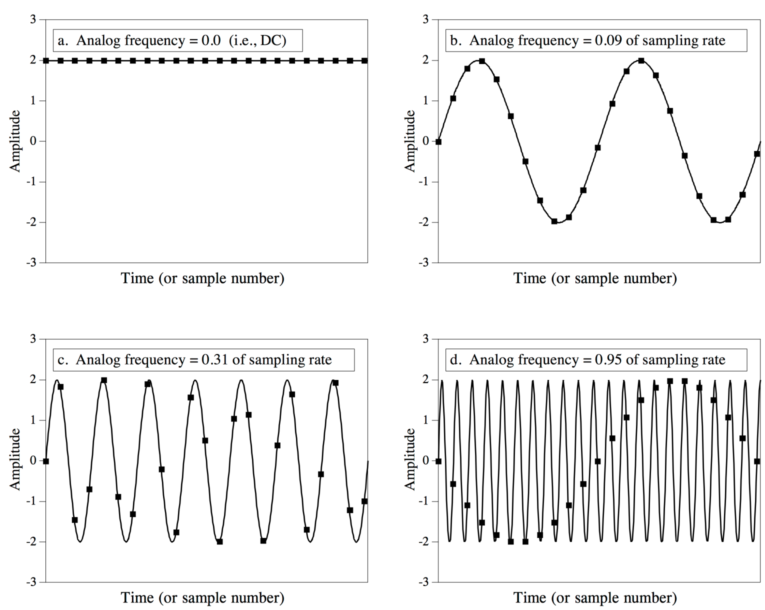

Figure 4.5: Illustration of the Sampling Theorem. In each panel the smooth curve represents a continuous signal and the squares represent the points at which the signal is digitally sampled. (a) A sinusoid of frequency 0 (straight line) can be perfectly reconstructed from the set of samples. (b) This sinusoid has a frequency that is 0.9 the sampling rate, as would be the case for a 90 Hz oscillation sampled at 1000 Hz. The spacing of the samples produces a good representation of the original sinusoid. (c) For a sinusoid with a frequency equal to 0.31 the sampling rate, one can reconstruct the original signal from the samples, although this is less obviously true than in (b). (d) When a sinusoid at frequency 0.95 times the sampling rate is sampled, it masquerades as a sinusoid of lower frequency, a phenomenon known as *aliasing*. Figure from Steven W. Smith *Digital Signal Processing*, 2003.

The sampling theorem means that if you are interested in using the Fourier transform to measure 250 Hz oscillations in your EEG data, then, in principle, you must sample the data at no lower than 500 Hz. However, measuring oscillatory activity at 250 Hz with a sampling rate of 500 Hz means that you will be relying on just two samples per cycle to provide information about 250 Hz activity, which is ill-advised in practice because EEG recordings (and all real signals) are noisy. It is therefore better to use a sampling rate somewhat higher than the Nyquist rate to increase the number of observations per cycle used in estimating oscillatory activity.

Figure 4.5 illustrates the consequences of using the same sampling frequency to sample four different continuous signals that vary in their frequency content. In (a–c), the sampling rate is greater than twice the frequency of the signal, which means the original signal can be reconstructed from the digitized data. Contrast this with (d), where the frequency of the signal is nearly as high as the sampling rate. In this case the higher frequency waveform takes the appearance of a lower frequency sinusoid in the sampled timeseries, a phenomenon called *aliasing*.

Up to this point in our discussion, we have reviewed how to extract the frequency content of a signal using the Fourier transform. One property of the Fourier transform that is worth pointing out is that it does not provide any information about how the frequency content of a signal varies over time—it returns a single estimate for each frequency for the input signal. In fact, an important assumption of the Fourier transform is that the frequency content of a signal does not vary over the duration of the time series. This is related to a second assumption of the Fourier transform, which is that the input signal must be stationary, meaning that the mean and variance of the signal do not change over time. While these assumptions hold for many types of signals for which Fourier methods are used, they are violated by EEG data. In addition, experiments designed to study cognitive processes such as memory encoding and retrieval are carried out with the explicit goal of comparing how the frequency content of the EEG signal varies in response to experimental events. Because of these limits on the situations in which it is valid to use the Fourier transform, other methods have been developed that provide time-varying estimates of the frequency content of a signal. These methods form the basis for the next sections.

Short-time Fourier Transform

The logic of the short-time Fourier transform (STFT) is that instead of applying the Fourier transform to the entire timeseries, the full length of the series is broken into smaller time windows and the Fourier transform is applied to each window separately. Doing so leads to an estimate of the frequency content of a signal for each time window. The resulting *time-frequency* representation will be sensitive to changes in frequency content that happen over time, for example in response to experimental events.

Two important parameters to consider when using STFT to analyze data are (1) the size of the window and (2) the degree of overlap between consecutive windows. The window size determines the tradeoff between resolving time and frequency information with the analysis. Selecting a window that is brief will lead to estimates that are well-localized in the time domain, meaning they will be derived from a relatively short interval, but this will lead to estimates that are poorly-localized in the frequency domain. This is because the frequency resolution of the Fourier transform is determined by the length of the analyzed signal (see previous section) so a window of shorter length will produce estimates at fewer frequencies. A more problematic issue is that the frequency estimates will be of poor quality because the window length does not provide enough cycles of lower frequency oscillations to obtain good levels of signal-to-noise. By taking longer windows, the signal-to-noise of the frequency estimates will be improved, but at the expense of the ability to localize frequency fluctuations in time. Thus, the window size for the STFT should be selected to allow sufficient estimation of the lowest frequency of interest—one rule of thumb is use a window long enough to include at least three cycles of the lowest frequency. So, if you are interested in frequencies at or above 3 Hz you would select a window size of at least 1 s. However, selecting a window length that is sufficient to obtain good estimates of the frequency content at the lowest frequency of interest leads to another problem which is that a window long enough to include a few cycles of the lowest frequency may be too long to resolve changes in activity at higher frequencies. In the 1 s that are needed to obtain reliable estimates of the 3 Hz oscillation, there could be prominent fluctuations in activity at, say, 100 Hz (see Chapter 3 for discussion of how such high-frequency activity relates to memory). These fluctuations will be reduced to a single estimate of 100 Hz activity for the entire 1 s window. This trade-off between resolution in the time domain (knowing when things happened) and resolution in the frequency domain (knowing what frequencies are present) applies to all of the spectral methods that we have discussed thus far, as well as those that we will encounter later.

Windowing Functions

The final decision that needs to be made when using the STFT to perform time-frequency decomposition is which windowing function to use. A windowing function is used to taper your data to attenuate the amplitude of the timeseries near the beginning and end. Why would one want to window the timeseries in this way, instead of simply using segments of data directly drawn from the original timeseries? The first issue to point out is that applying the Fourier transform to "unwindowed" segments of data assumes a model in which the original signal is windowed by a boxcar function. Viewed from this perspective, the question then becomes: what is the effect of the boxcar window on the Fourier transform of $x[n]$, and, if these effects are undesirable, what can be done to mitigate them? The choice of window function comes down to the issue of *spectral leakage*, which is a phenomenon in which the Fourier representation of a single sinusoid is spread over other frequencies. This phenomenon is exaggerated as the size of the window decreases, and is particularly problematic for the boxcar window function; for this reason, there are several other window functions that are often used that have more desirable spectral characteristics. One of the most commonly used is the *Hann* window, shown in Figure 4.6.

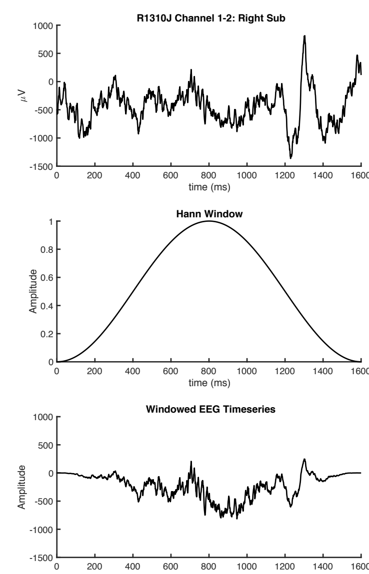

Figure 4.6: Windowing a timeseries. *Top*, an example EEG timeseries collected from the right subiculum of an intracranial EEG subject performing the FR1 task. *Middle*, the Hann windowing function. *Bottom*, windowing the EEG signal with the Hann window tapers the timeseries.

Convolution and Wavelet Transforms

One of the most often used methods for spectral decomposition is the Morlet wavelet transform. This technique addresses many of the limitations discussed in the previous section. Before covering the Morlet wavelet, we first introduce *convolution*: a mathematical operation defined for two functions that produces an output based on the dot product computed repeatedly over time.

Convolution and filtering basics

An intuitive way to think about convolution is to imagine starting with two timeseries, inverting one of them from left to right and sliding it along the other timeseries, incrementing one sample at a time. For each increment, convolution quantifies the dot product between the two signals—convolution is therefore a measure of how the similarity between two functions changes over time. Convolution is ubiquitous in the analysis of linear systems where the goal is to specify a system's predicted output given an arbitrary input function. Thus, another way to think about convolution is that it measures the effect of applying one function (the system) to another (the input signal).

Convolution of two timeseries x and h is defined as:

$$x[n] * h[n] = \sum_{m=0}^{N-1} x[m]h[n-m] \tag{4.10}$$

The array $x[n]$ is often called the signal while $h[n]$ is called the kernel. Because $x[n] * h[n]$ computes the dot product between two signals over time, the output of convolution is another timeseries that measures the similarity between $x[n]$ and $h[n]$ as the two timeseries are made to be more or less overlapping. The output of $x[n] * h[n]$ will therefore be largest at the timepoints where $x[n]$ and $h[n]$ are most similar.

Convolution is an operation defined on two signals in the time domain but it can also be expressed in terms of the frequency domain representations of the two signals:

$$x[n] * h[n] = \mathcal{F}^{-1}\{\mathcal{F}\{x[n]\} \cdot \mathcal{F}\{h[n]\}\} \tag{4.11}$$

where $\mathcal{F}\{x\}$ denotes the Fourier transform of x and $\mathcal{F}^{-1}$ denotes the inverse Fourier transform. Equation 4.11 is called the *Convolution Theorem* and states that convolving two signals in the time domain is equivalent to point-wise multiplication of the same two signals in the frequency domain and then applying the inverse Fourier transform. Practically, this property of convolution is useful because convolution in time is a very slow operation to carry out—the point-wise multiplication and summation operation has to be repeated for each timestep of overlap between x and h. Contrast this with the frequency domain version, in which point-wise multiplication is carried out only once. Because of the existence of algorithms like the fast Fourier transform (FFT), which can very efficiently compute the Fourier transform and its inverse, performing three Fourier transform operations in conjunction with a single multiplication operation can be orders of magnitude faster than computing the dot product repeatedly over time, especially for very long signals.

Morlet wavelet transform

The Morlet wavelet transform is a widely used spectral decomposition method that overcomes some of the limitations of strictly Fourier-based methods. Two key benefits of using wavelets are (1) the ability to extract time-resolved estimates of the frequency content of a signal; and (2) the use of kernels that scale as a function of frequency of interest. In this section, we describe the mathematics that underlie Morlet wavelets and describe how to use them to extract power and phase information from a signal.

A simple way to describe a wavelet is that it is a sinusoid windowed by a Gaussian:

$$\begin{aligned} h[n,k] &= Ae^{-\frac{(n-\mu)^2}{2\sigma^2}} e^{\frac{-i2\pi kn}{N}} \\ A &= \frac{1}{(\sigma\sqrt{\pi})^{1/2}}\,, \end{aligned} \tag{4.12}$$

where N is the length of the signal, k is the frequency of the wavelet, and μ, σ are the mean and standard deviation of the Gaussian. Here, σ is defined as follows:

$$\sigma = \frac{c_n}{2\pi k} \tag{4.13}$$

Figure 4.7 gives an example of windowing a sinusoid with a Gaussian to produce a wavelet. The parameter c_n is the wave number for the wavelet and affects the number of cycles that are present in the wavelet before it tapers to near zero. Changing this parameter changes the tradeoff between frequency and time resolution in the output of the wavelet transform. Increasing c_n will create a wavelet with more cycles, which means that the convolution between the wavelet and your timeseries at any particular timepoint will be influenced by data from a longer time interval. Integrating frequency over a longer period of time in this way will increase the resolution of the convolved signal in the frequency domain but will do so at the expense of resolution in the time domain.

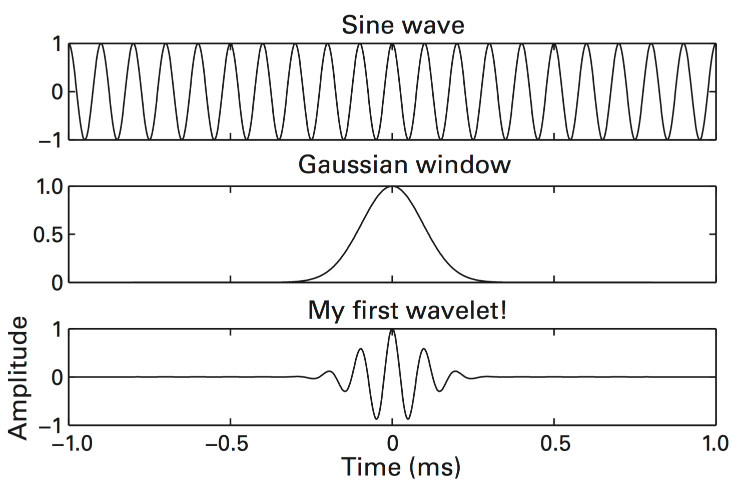

Figure 4.7: Constructing a wavelet. *Top*, a sinusoid. *Middle*, a Gaussian window. *Bottom*, applying a Gaussian window to a sinusoid to produce a wavelet. Figure reproduced from Michael X. Cohen's *Analyzing Neural Timeseries Data*, 2014.

Wavelet decomposition is performed by convolving the wavelet with the

EEG timeseries. Because the convolution operation pads the input signals with zeros, the output will contain values at the beginning and end that are significantly outside the range of values in the middle. This is due to the values of m in Equation 4.10 for which the signals are only partially overlapping. These undesirable signals are called *edge artifacts* because they occur at the edges of the timeseries and should not be analyzed as though they reflect true oscillatory brain signals. The solution to this problem is to add **buffer zones** to the beginning and end of your period of interest before convolving the timeseries with the wavelet. Then, you discard the buffers and are left with the uncontaminated convolution output for the period in the middle.

Amplitude, Power, and Phase Information

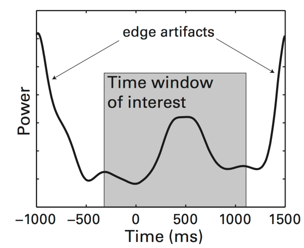

Figure 4.8: Edge Artifacts and Buffers. Convolution of an EEG timeseries and a wavelet leads to *edge artifacts* which are a consequence of zero-padding in the convolution operation and do not reflect true EEG signal. To deal with this issue when analyzing data, additional segments of time called *buffers* are added to the window of interest before the convolution and then discarded before proceeding with additional analysis.

The result of convolving a complex Morlet wavelet with your EEG timeseries will be another timeseries of complex numbers. How does one use this timeseries to extract information about sinusoidal activity in the data? Remember from earlier in the chapter that these complex numbers, $a + bi$ are sinusoids represented in the complex plane and are defined by a magnitude M and phase angle ϕ. Thus, extracting the amplitude of the sinusoid at each timepoint is simply a matter of computing $M = \sqrt{a^2 + b^2}$ and $\phi = \text{atan2}\,(b, a)$.

Although to this point in the chapter we have graphed the output of spectral decompositions, such as the Fourier transform, in terms of amplitude, it is in fact customary to graph the *power* of a signal:

$$\text{Power} = \text{Amplitude}^2 \qquad (4.14)$$

In the natural world there are often large differences in power at low and high frequencies (see next section). Thus we usually work with log-transformed power, and often normalize the resulting values (e.g., by transforming power into *decibel* units or by z-transforming log-power values).

Statistical methods in TF analysis

The power spectral density

A ubiquitous pattern in natural time series is that the associated power-spectrum declines with frequency such that power at any given frequency is approximately $1/f^{\alpha}$, where f denotes frequency and α is a parameter that describes how quickly power declines with frequency.[5] EEG data exhibit this pattern, as shown for an example signal in Figure 4.9. This has the practical implication that in analyses of EEG data, it is important to normalize the baseline power across low and high frequencies (see next section) in order to facilitate comparisons of relative changes in power values across frequencies. As we will see in Chapter 7, matching the range spanned by different features is important for ensuring that regression-based classification methods learn equally from low and high frequency features.

[5] Often times the exponent α is omitted and the spectrum is simply labeled $1/f$. Random noise that has a $1/f$ spectrum is called *pink noise* (distinct from *white noise* which has a flat frequency spectrum).

Baseline Normalization

As shown in the previous section, the spectral power of EEG signals follows a $1/f$ power law, indicating that lower frequencies tend to have higher

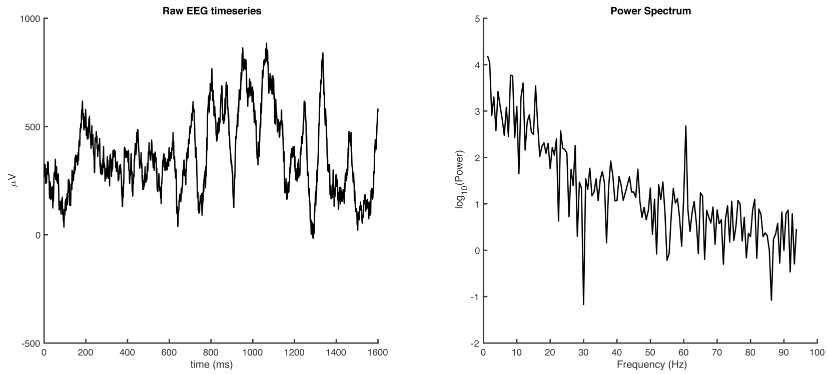

Figure 4.9: Example of $1/f$ power spectrum Data collected from a bipolar pair of intracranial electrodes implanted in the left middle temporal gyrus of a subject performing the FR1 task. The data were collected for the 0-1600 ms encoding interval relative to word onset for a single word event. *Top*, the raw EEG timeseries. *Bottom*, the power spectrum for this event, which exhibits the $1/f$ pattern.

power than higher frequencies. To facilitate the statistical analyses of EEG signals, baseline normalization techniques are often used to transform spectral powers at different frequencies onto the same scale. There are several ways to perform baseline correction in time-frequency analyses. The baseline value is typically taken to be the average power of the pre-event period (e.g., pre-encoding). One method is to calculate the percent change from the baseline of the signal by taking the difference between the signal and the baseline, and dividing this difference by the baseline. Another method is to log-transform the signal by taking the $\log_{10}$ of the quotient of the signal to the baseline, converting the signal into decibel units (dB). A third method involves calculating the z-score of the signal using the distribution of values in the baseline period by taking the difference between the signal and the baseline and dividing it by the standard deviation of the baseline. Table 4.1 summarizes these methods.

Method	Formula
$\%\Delta$	$(\mathbf{s}_{f,t} - \bar{\mathbf{b}}_f)/\bar{\mathbf{b}}_f$
dB	$10 \times \log_{10}(\mathbf{s}/\bar{\mathbf{b}}_f)$
z	$(\mathbf{s}_{f,t} - \bar{\mathbf{b}}_f)/\sigma_{\bar{\mathbf{b}}_f}$

Table 4.1: Baseline Normalization Schemes. $\%\Delta$: percent change, dB: decibel, z: z-transform. $\mathbf{s}_{f,t}$ indicates the signal vector (power values) at frequency f and time t. $\bar{\mathbf{b}}_f$ indicates the average power at frequency f during the baseline period and $\sigma_{\mathbf{b}_f}$ indicates the respective standard deviation.

Percent-change-normalization and z-normalization bring the variables of interest onto the same scale and eliminate their units. Log-transformation eliminates both the units and reduces the variability of the data by bringing the outliers in the original distributions closer to the rest of the observations. The percent-change-normalization of spectral power of EEG signals typically results in skewed distributions, a property that is not desirable for many parametric statical analyses. The z-normalization is sensitive to outliers in the baseline distribution and can produce unreliable transformed variables. One should carefully check for outliers in the baseline distribution before applying the z-transformation. The log-transform does not suffer from two aforementioned disadvantages of the other methods.

Researchers traditionally employ parametric methods, which typically assume normality of the distribution from which the data is drawn (also known as the data-generating distribution), to analyze the difference between variables of interest (for e.g., power values, ERP amplitudes) under different conditions. However, the normality assumption on the generative distributions of powers is often too restrictive and easily violated in practice. The normality of the *sampling distribution* of the test statistic (e.g., t-statistic) is sufficient even though the data-generating distribution might not be normal. Fortunately, the approximate normality of a typical test statistic such as the mean is guaranteed by the central limit theorem provided that the sample size is large enough and that the data-generating distribution does not deviate too much from normality. In other words, if the data-generating distribution is exactly normal, then the sampling distribution is normal regardless of the sample size. It does not take a large sample size ($N \geq 30$ is the typical rule of thumb) for the sampling distribution to approach normality if the data-generating distribution is close to normal and symmetric. On the other hand, if the data-generating distribution is extremely skewed (e.g., a Bernoulli distribution with 0.99 probability on 0 and 0.01 on 1), it would take a large amount of data (hundreds or even thousands of samples) before the sampling distribution is close to being normal. As a result, transformations (such as the logarithm) that reduce the variability and bring the distribution of the data closer to normality are desirable when one uses the parametric statistical framework. The log-transformation is particularly well-suited for powers because the distributions of powers are approximately distributed as chi-squared, which is positively skewed. As a result, taking the logarithm of power results in a distribution that is close to normality. In addition, log-transformed power exhibits a negative linear trend with increasing frequency. This relationship can be used to study broad-spectrum properties using the regression framework.

One should be mindful about what is being tested after the transformations of variables. For example, testing the mean of the log-transform of power is not equivalent to testing the mean of power.[6] As a result, we cannot make a statement about the mean of the original distribution using a statistical test on the log-transformed data.

[6] The difference comes from the fact that the logarithm of mean is not the same as the mean of logarithms. By Jensen's inequality and the log function being concave,

$$\frac{\log(x_1) + \cdots + \log(x_n)}{n} \leq \log(\frac{x_1 + \cdots x_n}{n})$$

The equality occurs if and only if $x_i = x_j, \forall i, j \in \{1, \cdots n\}$, which does not happen in practice.

Attenuating noise in the raw data

Standard approaches to analyzing both scalp and intracranial EEG data typically include multiple steps for preprocessing, including using statistical thresholds to identify channels or epochs that contain raw signals that are corrupted by noise. A common approach is to calculate statistical measures such as variance/standard deviation or kurtosis of the raw EEG signals. Epochs or channels that show outlying values relative to other epochs/channels are then removed before conducting further analysis. However, there is evidence that such preprocessing approaches are unlikely to improve the researcher's ability to identify brain-behavior correlations of interest (?, ?). Figure 4.10 shows a systematic analysis of the effect of such methods on the intracranial FR1 task. The analysis shows that that removing putatively noisy data does not increase t-statistics in the subsequent memory analysis.

Another class of noise removal methods involves review of raw data by researchers trained to identify abnormal physiological signals, for example epileptic activity. As shown in Figure 4.11 (left panel) such manual removal did not reliably increase univariate estimates of the subsequent memory effect. In the case of multivariate classification of subsequent memory, removing epochs significantly reduced classifier performance. This suggests that the cost of having fewer training observations was not outweighed by any benefit associated with removing epochs with potentially epileptic data.

Across all sets of analyses the preprocessing step that did significantly increase statistical power in the subsequent memory analysis was the use of bipolar referencing, compared to common average referencing. Bipolar referencing schemes use neighboring electrodes to reference one another, creating a new array of 'virtual' channels across all pairs in a participant's

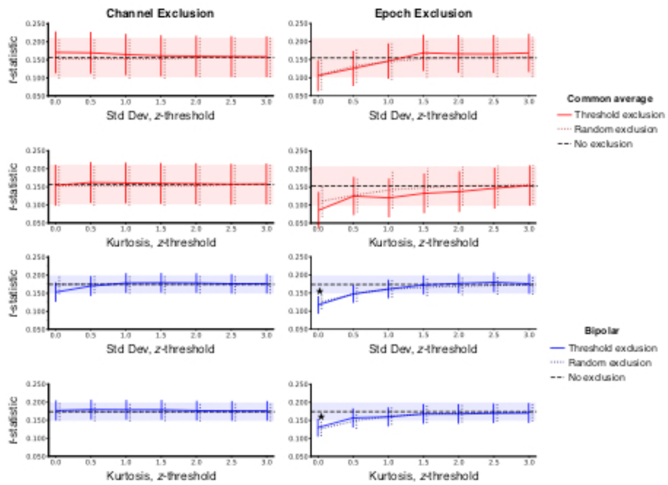

Figure 4.10: Effects of statistical exclusion of epochs and channels. A systematic analysis of statistical thresholding for identifying noisy data showed that removing data does not increase estimates of the subsequent memory effect.

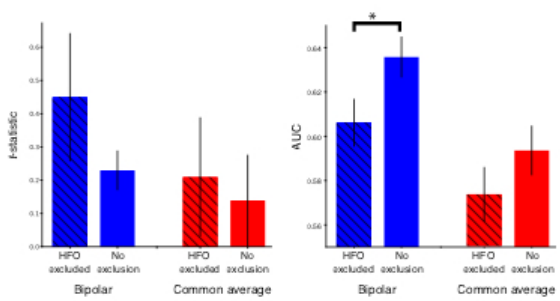

Figure 4.11: Effects of manual exclusion of epochs. Manual review and removal of epochs exhibiting high-frequency activity consistent with epilepsy did not increase estimates of the subsequent memory effect. In the case of multivariate classification (right panel) using bipolar referenced data, manual exclusion of such epochs reliably impaired classifier performance.

montage. While it is clear that such an approach increases the strength of the subsequent memory analysis of high-frequency activity in iEEG and in broadband classification, an open question concerns the effects of bipolar referencing in the analysis of lower frequency signals alone, as well as in data collected non-invasively at the scalp.

Multiple comparisons correction in time-frequency analyses

In Chapter 3, we addressed the multiple-comparisons issue associated with testing event-related potentials that involve many channels and time points. In addition to channels and time points, time-frequency analyses have frequency as an extra dimension. As a result, one faces a larger multiple-comparisons problem when testing spectral power. Fortunately, methods introduced in Chapter 3 to control for the family wise error rate or the false discovery rate can be used to mitigate this issue.