11 Analysis of Spike-Field Coherence during Navigation

Synopsis

Data 100 trials of 1 s of local field potential and spike train data sampled at 1000 Hz.

Goal Characterize the coupling between the spike and field activity.

Tools Fourier transform, spectrum, coherence, phase, generalized linear models.

11.1 Introduction

11.1.1 Background

In the previous chapters, we focused on two types of data: field data (e.g., EEG, ECoG, LFP) and spiking data (i.e., action potentials), and we developed techniques to analyze these data. In this chapter, we consider the simultaneous observation of both data types. We analyze these multiscale data using the techniques developed in previous chapters and focus specifically on computing the coherence between the spike and field recordings. Understanding the relations between activity recorded at different spatial scales (i.e., a macroscopic field and microscopic spikes) remains an active research area.

11.1.2 Case Study Data

Our experimental collaborator has implanted an electrode in rat hippocampus as the animal performs a task requiring navigation and decision making. From these data, he is able to extract the local field potential (LFP) as well as the spiking activity of a single neuron. He would like to characterize how these multiscale data—the population field activity and the single neuron spiking activity—relate. Based on existing evidence in the literature and experimental intuition, he expects that rhythmic activity in the LFP impacts the probability that a spike will occur. As his collaborator, we will help him to develop tools to examine this hypothesis. He provides us with 100 trials of simultaneous LFP and spike train data with a sampling frequency of 1000 Hz. The duration of each trial is 1 s, corresponding to a fixed temporal interval following a particular decision of the rat.

11.1.3 Goal

Our goal is to understand the coupling between the spiking activity and the LFP following the stimulus. To do so, we analyze the multiscale data recorded simultaneously. To assess this coupling, we will start with two visualizations of the data: the spike-triggered average and the field-triggered average. We then compute the *spike-field coherence*, a coupling measure that builds upon previous development of the Fourier transform and spectrum. We also examine how the firing rate impacts measures of coupling and how to mitigate this impact.

11.1.4 Tools

In this chapter, we focus primarily on computing the spike-field coherence. Development of this measure makes use of skills developed in previous chapters. In computing the spike-field coherence, we continue to utilize the Fourier transform. We also consider how generalized linear models (GLMs) can be used to construct a measure of spike-field association with an important advantage over the spike-field coherence.

11.2 Data Analysis

11.2.1 Visual Inspection: Spike-Triggered Average and Field-Triggered Average

To access the data for this chapter, visit

`http://github.com/Mark-Kramer/Case-Studies-Kramer-Eden`

and download the file `Ch11-spikes-LFP-1.mat`. We begin the analysis by visualizing examples of the simultaneously recorded spike train and LFP data. Let's load these multiscale data into MATLAB and plot the activity of the first trial:

```
load('Ch11-spikes-LFP-1.mat')  %Load the multiscale data,
plot(t, y(1,:))                %...and plot LFP in 1st trial,
hold on                        %...freeze graphics,
plot(t, n(1,:),'k')            %...and plot spikes in 1st trial,
hold off                       %...release graphics,
xlabel('Time [s]')             %...and label axis.
```

The data file consists of three variables, which correspond to the LFP data (`y`, in units of millivolts), the simultaneously recorded spiking activity (`n`), and a time axis (`t`, in units of seconds). Notice that the data are matrices, in which each row indicates a separate trial, and each column indicates a point in time. In this case, the variable `n` is binary; `n(k,i)=1` indicates a spike in trial `k` at time index `i`.

Q: What is the sampling frequency for these data?

A: We are given the time axis `t`. To compute the sampling frequency, we compute the sampling interval: `dt = t(2)-t(1)` and find `dt = 0.001`. The sampling interval is therefore 1 ms, so the sampling frequency (f) is $f = 1/dt = 1000$ Hz.

The data for this trial are plotted in figure 11.1. Visual inspection immediately suggests that the LFP data exhibit a dominant rhythm. By counting the number of peaks (or troughs) in 1 s of data, we estimate the dominant rhythm to be ≈ 10 Hz. However, careful inspection suggests that other features appear in the LFP from this first trial of data (i.e., additional lower-amplitude wiggles in the signal). Let's keep this in mind as we continue the analysis. To visualize the spikes from the neuron, we plot the activity in the first row of the data matrix (black curve in figure 11.1); this is a crude representation of the activity but sufficient for the initial inspection.

Q: Continue your visual inspection for other trials of the data. What do you observe?

Visual inspection suggests that the neuron is active (i.e., it spikes) during the trial. Of course, we may visualize and analyze features of the spike train and the LFP using the methods described in earlier chapters (e.g., see problem 1). However, our goal here is to characterize the relation (if any) between the LFP and spikes. Let's consider a relatively simple characterization of this relation, the spike-triggered average.

Spike-Triggered Average. The *spike-triggered average* (STA) is a relatively simple procedure to visualize the relation between the LFP and spiking data. To compute the STA, we implement the following procedure.

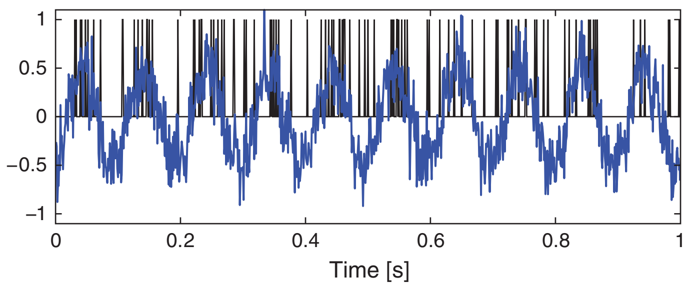

Figure 11.1
Traces of the LFP data (*blue*) and spike train (*black*) recorded in the first trial.

For each trial $k = \{1, \ldots, K\}$, do the following:

1. Identify the time of each spike occurrence $t_{k,i}$, where $i \in \{1, , N_k\}$, and N_k is the number of spikes in the k^{th} trial.
2. For each spike time $t_{k,i}$, determine the LFP within a small temporal interval near the spike time $LFP_{k,i}$.
3. Average $LFP_{k,i}$ across all spikes.

Despite this seemingly complicated procedure, the STA is a relatively intuitive measure. The intuition is to find each spike and determine how the LFP changes nearby. The procedure to compute the STA for each trial is relatively straightforward to perform in MATLAB:

```
win = 100;                %Choose a window size of +/- 100 ms.
K = size(n,1);            %Define the no. of trials.
N = size(n,2);            %Define no. of data points per trial.
STA=zeros(K,2*win+1);     %Create a variable to hold the STA.
for k=1:K                 %For each trial,
  spks=find(n(k,:)==1);   %...find the spikes,
  for i=1:length(spks)    %...and for each spike,
    if (spks(i) > win & spks(i)+win<N)   %...average the LFP.
     STA(k,:)=STA(k,:)+y(k,spks(i)-win:spks(i)+win)/length(spks);
    end
  end
end
```

In this MATLAB code, we must be careful to include only appropriate time intervals when computing the STA.

Q: What is the purpose of the *if-statement*:

```
if (spks(i) > win & spks(i)+win<N)
```

in the code?

Notice that the variable `STA` is a matrix, with each row corresponding to a separate trial. Figure 11.2 shows the results for the STA in four trials and in all trials. The individual trial results suggest an approximate rhythmicity in the STA; visual inspection reveals that the STA fluctuates with a period of approximately 100 ms. However, these fluctuations are not phase-locked across trials. For some trials, the LFP tends to be positive when the cell spikes (i.e., at $t = 0$ in figure 11.2), while in other trials the LFP tends to be negative when the cell spikes. The initial results do not suggest a consistent relation exists between the spikes and the LFP across trials.

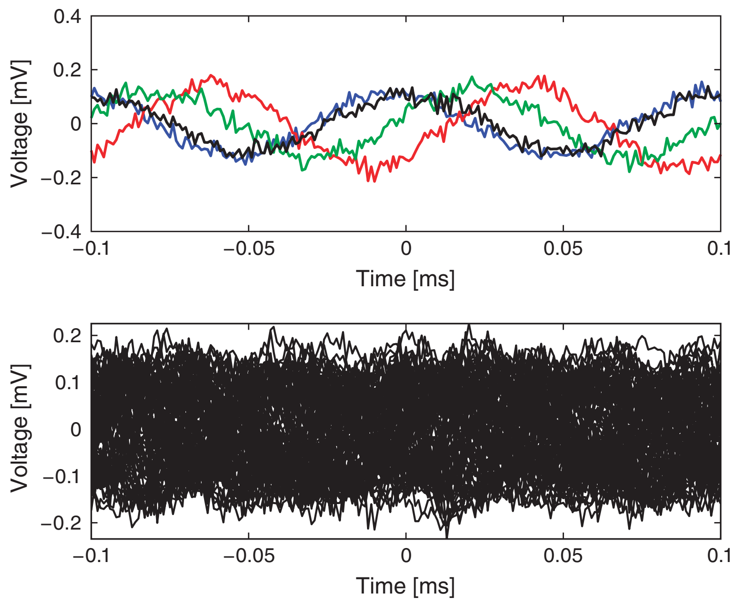

Figure 11.2

Spike-triggered average (*top*) for individual trials (trial 1 in *blue*, 6 in *red*, 10 in *green*, 16 in *black*) and (*bottom*) all 100 trials.

However, let's not abandon all hope yet. We might be concerned that the rhythmicity in the STA (figure 11.2) is consistent with the dominant rhythm of the LFP (figure 11.1). Because the STA is an average of the LFP, we might expect the largest-amplitude features of the LFP to make the biggest impact on the STA. Perhaps this large-amplitude rhythm in the LFP is hiding more subtle features embedded in lower-amplitude activity in the LFP. Let's continue the search.

> **Q:** How would you update the preceding MATLAB code to compute both the average LFP (i.e., the STA) and the standard deviation of the LFP across spikes for each trial?

Field-Triggered Average. Let's now implement another visualization, the *field-triggered average* (FTA). The FTA is similar in principle to the STA. However, for the FTA, we use the field to organize the activity of the spikes (i.e., we use the field to trigger the spikes). Here we choose a particular feature of the field: the phase. We have already investigated the important role of phase in organizing neural rhythms (see chapter 5). Now we examine the role of the LFP phase in organizing the spiking activity.

For each trial $k = \{1, \ldots, K\}$, do the following:

1. Filter the LFP data in trial k into a narrow band, and apply the Hilbert transform to estimate the instantaneous phase.
2. Sort the spiking data in trial k according to the phase of the LFP.

We discussed the Hilbert transform and instantaneous phase in the analysis of cross-frequency coupling in chapter 7. We apply the same procedures here, but to a different end. To implement computation of the FTA in MATLAB,

```
dt = t(2)-t(1);      %Define the sampling interval.
fNQ = 1/dt/2;        %Define Nyquist frequency.
Wn = [9,11]/fNQ;     %Set the passband,
ord  = 100;          %...and filter order,
b  = fir1(ord,Wn);   %...build bandpass filter.

FTA=zeros(K,N);      %Create a variable to hold the FTA.
for k=1:K            %For each trial,
 Vlo = filtfilt(b,1,y(k,:));%... apply the filter,
 phi = angle(hilbert(Vlo)); %... and compute the phase,
 [~,indices] = sort(phi);  %... then sort the phase,
 FTA(k,:) = n(k,indices);   %... and store the sorted spikes.
end
%Plot the average FTA versus phase.
plot(linspace(-pi,pi,N), mean(FTA,1))
```

Notice the steps to set up the filter. We choose a bandpass filter from 9–11 Hz. We choose this interval to focus on the LFP rhythm of largest amplitude ($\approx$ 10 Hz), which we identified through visual inspection (figure 11.1). For each trial, we apply the filter to the LFP and then use the Hilbert transform (`hilbert`) to estimate the phase. Finally, we sort this phase and use the sorted indices to arrange the spikes and store the results. We show the FTA averaged across all trials in figure 11.3. In this case, no modulation in the number of spikes is apparent across trials. Instead, the number of spikes at each phase appears equally likely.

We may apply the FTA analysis to different frequency intervals of the LFP. Choosing a frequency interval may be motivated by our knowledge of the neural system generating the activity or by inspection of the field and spiking data.

Q: Investigate different frequency bands in the FTA analysis. Do you observe any interesting features? *Hint*: Consider frequencies near 45 Hz.

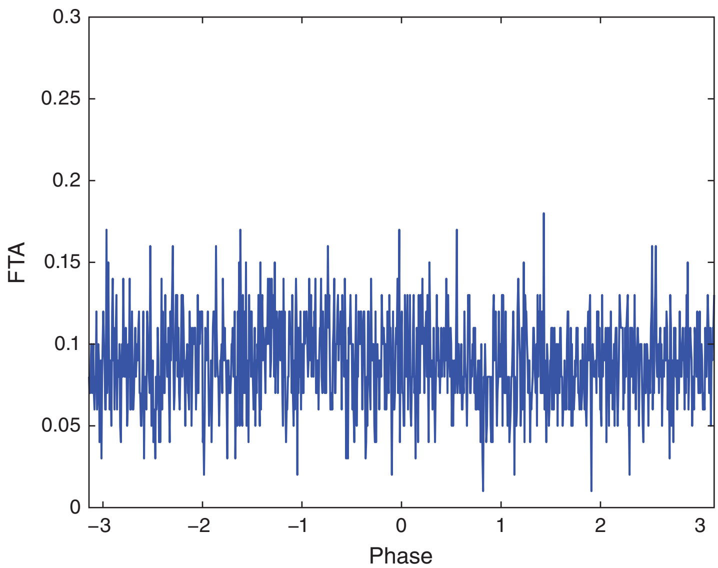

Figure 11.3
Field-triggered average for LFP data in the frequency range 9–11 Hz, averaged across all trials.

One final note about the FTA. The purpose of this measure is visualization, not statistical testing. Hopefully, this visual inspection will provide some insight into the data and guide continuing studies in promising directions. In what follows, we consider approaches to test for significant effects when we build a GLM to assess spike-field relations.

11.2.2 Spike-Field Coherence

To characterize the relation between the LFP and spikes, we have so far visualized the data and computed relatively simple and intuitive aids to visualization. Now we examine a more sophisticated and powerful method: the spike-field coherence. We investigated the coherence when applied to field activity (namely, ECoG data in chapter 5); we may refer to this type of coherence as field-field coherence to distinguish it from spike-field coherence of interest here. In practice, this distinction is usually unnecessary, as in most cases the context is clear. However, in this chapter, we are careful to distinguish field-field coherence from spike-field coherence.

The *field-field coherence* is a frequency domain measure of linear association between two continuous time series.[1] We showed in chapter 5 that two fields are coherent across

1. In practice, we observe a sampled version of a presumably continuous signal. We explored some implications of this sampling on spectral estimators in chapters 3 and 4.

trials at frequency f_0 if the fields possess a constant phase relation across trials at that frequency. The same relation holds for the *spike-field coherence*. However, differences arise because of the point process nature of the spike train data. These differences have profound implications with dangerous consequences. In this chapter, we explore some of these issues. For a deeper mathematical discussion and potential solutions, see [32, 33].

Mathematical Description of spike-field coherence. Let's begin with a mathematical description of the spike-field coherence. To do so, we need to introduce some notation, which is identical to that used in earlier chapters, but we include it here for completeness. A more detailed description may be found in [32].

We considered spectral estimators for a field in chapter 3 and for a point process in chapter 10. We restate the Fourier transform for a time series x,

$$X_j = \sum_{n=1}^{N} x_n \exp(-2\pi i f_j t_n), \tag{11.1}$$

where x_n is the signal at time index $t_n = n\Delta$, and the frequencies $f_j = j/T$, where $j = \{-N/2+1, -N/2+2, \ldots, N/2-1, N/2\}$. The spectral density of the time series is then

$$S_{xx,j} = \frac{2\Delta^2}{T} X_j X_j^*. \tag{11.2}$$

Here, the time series can be either a field (i.e., the LFP) or a point process (i.e., the spike train). Notice that we employ the same mathematical formula to compute the spectrum for each time series.

> For the spike train data, we first subtract the mean or expected number of spikes in each time interval and then apply the Fourier transform. In other words, the signal is the *centered increments* (see chapter 8).

Then, to estimate the coherence between two time series x and y, compute

$$\kappa_{xy,\,j} = \frac{|< S_{xy,\,j} >|}{\sqrt{< S_{xx,\,j} >}\sqrt{< S_{yy,\,j} >}}, \tag{11.3}$$

where $|< S_{xy,\,j} >|$ indicates the magnitude of the trial-averaged cross-spectrum, and $|< S_{xx,\,j} >|$ and $|< S_{yy,\,j} >|$ indicate the magnitude of the trial-averaged spectra of x and y, respectively. So far, there's nothing new here; we've just restated the standard expressions for the spectrum and coherence. To compute the spike-field coherence, we simply interpret one of the time series as a point process. To make this more apparent in the mathematical

10234_011 2016/7/20 11:08 Page 351 #9

expression, we replace x in (11.3) with the symbol n as a reminder that this time series represents the number of spikes,

$$\kappa_{ny,\,j} = \frac{|< S_{ny,\,j} >|}{\sqrt{< S_{nn,\,j} >}\sqrt{< S_{yy,\,j} >}}. \tag{11.4}$$

In (11.4) the numerator is now the magnitude of the trial-averaged cross-spectrum between the field y and spikes n, and the denominator contains the trial-averaged spectrum of the spike n and the trial-averaged spectrum of the field y.[2]

Computing the Spike-Field-Coherence in MATLAB. As discussed in chapters 3, 4, and 10, many issues are involved in spectral analysis, for example, the notions of tapering. These important issues apply for the computation of spike-field coherence as well. In practice, multitaper methods are used to compute the spike-field coherence. An important advantage of the multitaper method is that we make a favorable trade-off: a reduction in variance for a loss of frequency resolution. We briefly illustrate the application of this multitaper method to compute the coherence here. To do so, we make use of the software package Chronux [2] (see section 1.24 of chapter 1).

Let's now apply the multitaper method to compute the spike-field coherence for the data of interest here. It's relatively straightforward to do so in MATLAB:

```
dt = t(2)-t(1);              %Define the sampling interval.
Fs = 1/dt;                   %Define the sampling frequency.
TW = 3;                      %Choose time-bandwidth product of 3.
ntapers = 2*TW-1;            %Choose the no. of tapers.
%Set the parameters of the MTM,
params.Fs = Fs;              %... sampling frequency,
params.tapers=[TW,ntapers]; %... time-bandwidth product & tapers,
params.pad = -1;             %... no zero padding.
params.trialave = 1;         %... trial average.
%Compute the MTM coherence.
[C,~,~,Syy,Snn,f]=coherencycpb(transpose(y),transpose(n),params);
```

In the third and fourth lines we define the time-bandwidth product (`TW`) and the number of tapers to apply (`ntapers`). Then, in lines 6–9, we set the parameters of the multitaper method using the variable `params`; this includes the sampling frequency (`Fs`), zero padding (here set to none), and trial averaging of the results (`params.trialave=1`). Finally, we estimate the spectra and coherence of the spike and LFP data in line 11 using the Chronux

2. We could instead write the *sample* coherence because this equation uses the observed data to estimate the theoretical coherence that we would see if we kept repeating this experiment. However, this distinction is not essential to the discussion here.

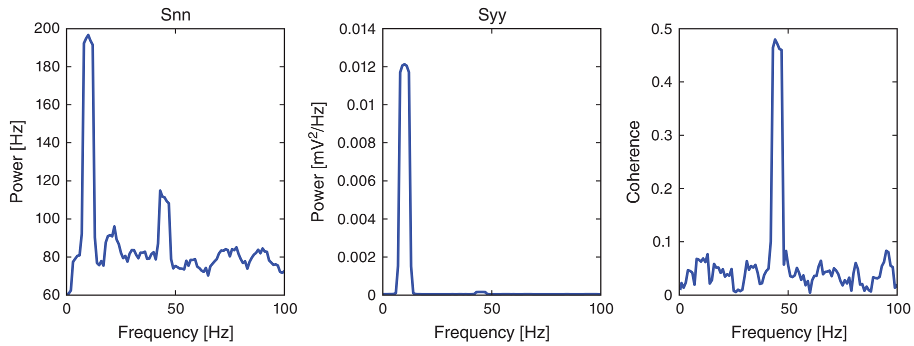

Figure 11.4
Spike spectrum (S_{nn}), field spectrum (S_{yy}), and spike-field coherence (*right*) computed using multitaper method.

function `coherencycpb`. The format of this function requires that we input the data in the form [samples × trials], and so we apply the `transpose` function to the variables `y` and `n` as inputs to `coherencycpb`. The results of these computations are plotted in figure 11.4.

Q: Consider the spike spectrum, S_{nn}, plotted in figure 11.4. What are the dominant rhythms? At frequencies beyond these dominant rhythms, the spectrum appears to fluctuate around a constant value. What is this constant value?

A: To answer the first question, we determine through visual inspection of figure 11.4 that the dominant rhythm (i.e., the frequency with the most power) occurs at 10 Hz. We also note the presence of a second peak near 45 Hz.

To answer the second question, we note that the spike spectrum asymptotes at the expected spike rate (see chapter 10). For these data, we can estimate the expected spike rate as

```
lambda = mean(sum(n,2))/(N*dt);
```

Computing this quantity in MATLAB, we find an expected spike rate of approximately 89 Hz, consistent with the high-frequency behavior of S_{nn} plotted in figure 11.4.

Q: Consider the field spectrum, S_{yy}, plotted in figure 11.4. What are the dominant rhythms? Do you observe any other interesting features in this spectrum?

A: Visual inspection of figure 11.4 reveals that the dominant rhythm occurs at 10 Hz. At first glance, no additional spectral features stand out.

These observations of the spike spectrum and field spectrum reveal that both signals exhibit rhythmic activity at 10 Hz. Therefore, a reasonable place to look for spike-field coherence is near 10 Hz, where both the spikes and the field are rhythmic. However, visual inspection of the spike-field coherence (figure 11.4) does not indicate coherence at this frequency. Instead, we find a large peak in the spike-field coherence at 45 Hz. Identifying this strong coherence at 45 Hz suggests that we reexamine the spectra. Indeed, careful inspection of the spike spectrum and field spectrum does suggest rhythmic activity at 45 Hz.

Q: Consider the field spectrum on a decibel scale (see chapter 4). What rhythms do you observe?

Q: Compare the results of your spike-field coherence analysis with the FTA plotted in figure 11.3. How does the peak in the spike-field coherence relate to interesting structure in the FTA?

The spike-field coherence results again reveal an important feature of coherence analysis. Two signals with high power at the same frequency are *not* necessarily coherent at this frequency; two signals may possess rhythmic activity at the same frequency, but these rhythms may not coordinate across trials. Conversely, two signals with low power at the same frequency may have strong coherence at that frequency; although the rhythm is weak, the two signals may still coordinate activity across trials at this frequency. These notions apply both to spike-field coherence and field-field coherence (the latter illustrated in chapter 4).

The multitaper method to compute the spike-field coherence is a powerful tool in our data analysis arsenal. There's much more to say about this approach, and interested readers are directed to [32, 34, 35].

The Impact of Firing Rate on the Spike-Field Coherence. Often, in the analysis of neural data, we compare the coherence between two pairs of signals. For example, in analysis of scalp EEG data, we might compare the coherence between voltage activity recorded at electrodes A and B with the coherence between voltage activity recorded at electrodes A and C. If we find that electrodes A and B have higher coherence at some frequency than electrodes A and C, we may conclude that the two brain regions A and B coordinate more strongly at this frequency. In this thought experiment, we are comparing the *field-field* coherence, which is not affected by the amplitude of the signals. For example, if we multiply the amplitude of signal C by a factor of 0.1, the field-field coherence does not change. To gain some intuition for this result, note that in the computation of the coherence (11.3), we divide by the spectrum of each signal. In this way, a multiplicative change in signal amplitude appears in the numerator and denominator of the coherence formula and therefore (in this case) factors out.

We might expect the same for spike-field coherence. To test this, let's manipulate the experimental data provided by our collaborator. We scale the field data by a factor of 0.1 and recompute the spike-field coherence. Scaling the field data is easy to do in MATLAB:

```
load('Ch11-spikes-LFP-1.mat')    %Reload the multiscale data,
y = 0.1*y;                       %...and scale the LFP.
```

With this change in the LFP data (`y`), we now recompute the spike-field coherence. We find that this multiplicative change in the amplitude of the field data does not impact the spike-field coherence (figure 11.5). This result is consistent with our intuition from field-field coherence; the height of the field does not matter. Instead, it's the consistency of the phase relation between two signals across trials that is critical for establishing the coherence.

Now, let's consider manipulating the spiking data. Right away, we notice a difference compared to the field data. In this case, a direct multiplicative change of the spiking data does not make sense. For example, consider multiplying the spike train data (`n`) by a factor of 0.1. Recall that the spike train data consist of two values: 0 or 1. Therefore, the new data after the scaling consist of two values: {0, 0.1} and the interpretation of the variable `n` no longer makes sense. What does it mean to have 0.1 spikes in a time interval? Instead, to

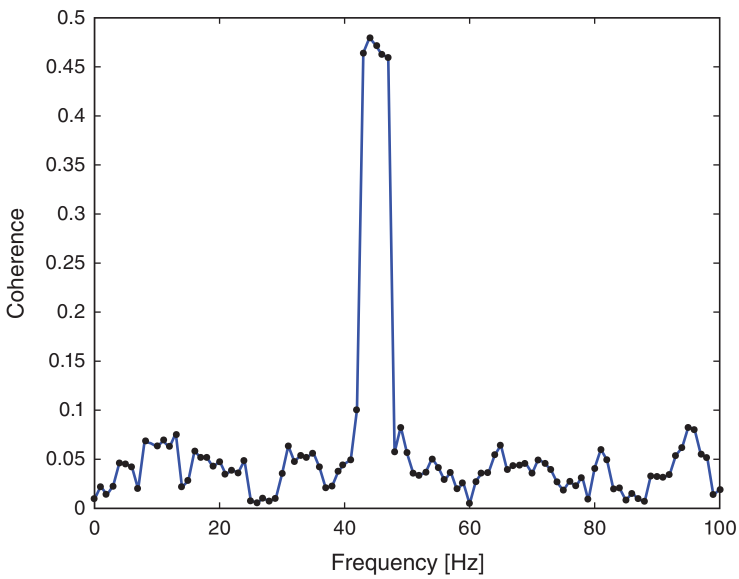

Figure 11.5
Spike-field coherence for scaled LFP data (*blue*) is identical to spike-field coherence for original, unscaled LFP data (*black dots*).

scale the spiking data, we change the average firing rate. We do so in a particular way: by removing spikes from the data, a process we refer to as *thinning*. The process of thinning is useful when comparing the spike-field coherence computed for two neurons with different firing rates. A reasonable, intuitive worry is that the firing rate of a neuron will impact the spike-field coherence. For example, we might consider that a neuron with a higher firing rate has the advantage of more opportunities to align with the field and therefore necessarily will possess a larger spike-field coherence. By thinning, we reduce the higher firing rate and establish the two neurons on an equal footing, both with the same opportunity to align with the field. The objective of the thinning procedure is to eliminate the contribution of firing rate differences to the spike-field coherence and allow direct comparison of spike-field coherence results computed for different neurons.

Let's now thin the spiking data. Here, we implement a simple procedure by randomly selecting and removing spikes from each trial of the spiking data. We assume that in selecting spikes at random to remove, we eliminate both spikes phase-locked to the field and spikes independent of the field. In this way, neither spikes coupled to the LFP nor spikes independent of the LFP receive preferential treatment in the thinning procedure. So, any relations that exist between the spikes and the field are presumably preserved, and we might expect this thinning procedure, on its own, to not affect the spike-field coherence. Let's implement this thinning procedure in MATLAB:

```
load('Ch11-spikes-LFP-1.mat')       %Reload the multiscale data.
thinning_factor = 0.5;              %Choose a thinning factor.
for k=1:size(n,1)                   %For each trial,
 spikes = find(n(k,:)==1);          %...find the spikes,
 n_spikes = length(spikes);         %...determine no. of spikes,
 spikes=spikes(randperm(n_spikes));%...permute spikes indices,
 n_remove=floor(thinning_factor*n_spikes);
                                    %...no. of spikes to remove,
 n(k,spikes(1:1+n_remove))=0;       %... remove the spikes.
end
```

Note that within the for-loop, we first find the indices corresponding to spikes in trial `k`. We then randomly permute these indices, and select the appropriate proportion of these indices for removal. Figure 11.6 illustrates the results of this thinning procedure. In figure 11.6a we plot the thinning factor (i.e., the proportion of spikes removed from each trial) versus the empirical firing rate estimated from the data. As expected, removing more spikes decreases the firing rate.

We also plot in figure 11.6b the spike-field coherence for four different levels of thinning, one of which corresponds to the choice of a thinning factor of 0.5. We find that as the number of spikes removed increases, the peak of spike-field coherence decreases. Why? Intuition suggests that removing spikes at random (i.e., removing spikes coupled to the phase of LFP and removing spikes independent of the LFP) should preserve the spike-field

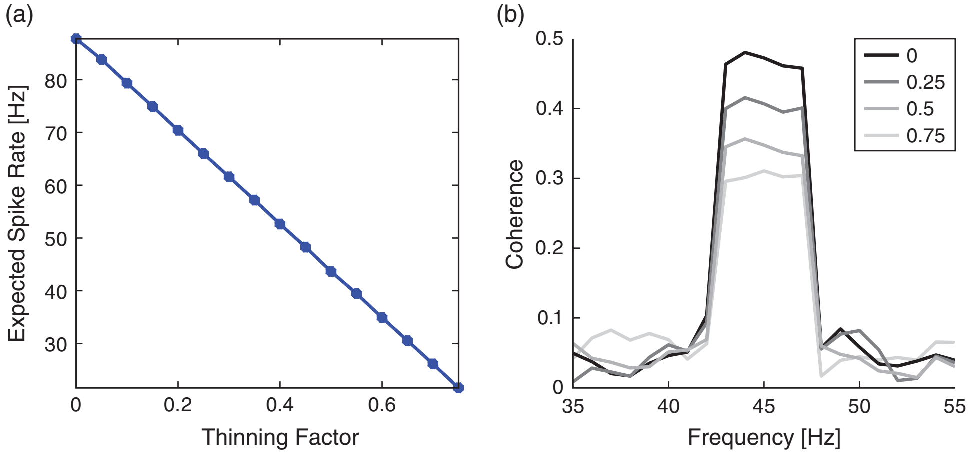

Figure 11.6

Spike-field coherence for multiscale data with original and thinned spike train. (*a*) Expected spike rate depends on the amount of thinning. (*b*) Spike-field coherence decreases as the thinning increases. Legend indicates value of `thinning_factor`.

coherence. Perhaps we were unlucky in the thinning procedure and selected to remove more phase-locked spikes than non-phase-locked spikes?

> **Q:** Repeat the analysis with `thinning_factor = 0.5` to select another random batch of spikes to remove. How does the spike-field coherence change compared to `thinning_factor = 0.0`? Try this a couple of times, and investigate the peak spike-field coherence at 45 Hz. Is the peak in the spike-field coherence always reduced upon thinning?

Repeating the thinning procedure and selecting new instances of random spikes to remove preserves the qualitative result. The peak spike-field coherence at 45 Hz decreases. Perhaps we made a conceptual error in the thinning procedure or an error in the MATLAB code? In fact, this result is not a numerical artifact or an error in the code or a problem with the estimate; it's a property of the spike-field coherence. In [32] it's proven that the spike-field coherence depends on the firing rate. An important result from [32] is:

> As the firing rate tends to zero, so does the spike-field coherence.

Therefore, we must be very careful when interpreting the spike-field coherence, especially when comparing the spike-field coherence of two neurons with different firing rates. A reduction in spike-field coherence may occur either through a reduction in association between the spikes and the field, or through a reduction in the firing rate with no change in association between the spikes and the field. This is an important and perhaps counter-intuitive result of spike-field coherence. The problems at the end of this chapter further illustrate this result through simulation. In addition, some procedures exist to mitigate the dependence of spike-field coherence on the firing rate, as discussed in the next section.

> Spike-field coherence responds to overall neural spiking activity, making comparisons between two pairs of spike-field time series difficult when the average spike-rate differs in the two spike-field pairs [32].

11.2.3 Point Process Models of the Spike-Field Coherence

A variety of techniques exist to address the impact of firing rate on the spike-field coherence. We have already outlined the thinning procedure, a transformation-based technique in which the firing rates of two neurons are made equal by randomly removing spikes. Here, we focus on an additional technique that utilizes the generalized linear modeling framework. We choose this technique (described in detail in [33]) because it allows us to utilize the GLM framework applied in chapters 9 and 10. The fundamental idea of this procedure is to model the conditional intensity of the point process λ as a function of the LFP phase. More specifically, we consider the model

$$\lambda_t = e^{\beta_0+\beta_1 \cos(\phi(t))+\beta_2 \sin(\phi(t))}, \tag{11.5}$$

where $\phi(t)$ is the instantaneous phase of a narrowband signal in the LFP. To compute the phase, we bandpass-filter the LFP and apply the Hilbert transform, as described in our computation of the field-triggered average. Then, using the canonical log link, we fit the GLM to the spike train data (see chapters 9 and 10) to estimate the model parameters. We note that the first parameter, β_0, accounts for the overall activity of the neuron, while the other two parameters, β_1 and β_2, capture the association between the LFP phase and spiking activity. In this way, the overall firing rate and the impact of the field on the spiking activity are separately modeled, which mitigates the impact of firing rate on the measure of spike-field association.

For the analysis of spike-field association, we select a small frequency band of interest, bandpass-filter the field data, and then estimate the phase; the procedures to do so are identical to those used to compute the FTA. Building from those steps, we now focus on the procedures to estimate the phase and GLM in MATLAB:

```
phi=zeros(K,N);              %Create variable to hold phase.
for k=1:K                    %For each trial,
  Vlo=filtfilt(b,1,y(k,:)); %...apply the filter (see FTA code),
```

```
  phi(k,:)=angle(hilbert(Vlo)); %...and compute the phase.
end

n = n(:);                    %Convert spike matrix to vector.
phi = phi(:);                %Convert phase matrix to vector.
X = [cos(phi) sin(phi)];     %Create a matrix of predictors.
Y = [n];                     %Create a vector of responses.
[b1,dev1,stats1]=glmfit(X,Y,'poisson'); %Fit the GLM.

phi0=transpose((-pi:0.01:pi));      %Define new phase interval,
X0 = [cos(phi0) sin(phi0)];         %...and predictors,
[y0 dylo dyhi]=glmval(b1,X0,'log',stats1);%...evaluate the model.
```

Note that the filter settings in the variable `b` of line 3 are defined in the MATLAB code used to compute the FTA. In fact, the first five lines are similar to the MATLAB code used to compute the FTA. The difference is that we now store the phases in a matrix (variable `phi`) that we use in the GLM procedure. To fit the GLM, we first collect the spikes and phases across trials into a vector (lines 6 and 7). Then we define the predictors (`X`, which are functions of the phase) and the response (`Y`, the spikes) and fit the model using the function `glmfit`. Finally, we evaluate the model for a chosen phase interval using the function `glmval`.

The modeling estimates are shown in figure 11.7: the FTA (identical to figure 11.3) and the estimates of the GLM (computed using the MATLAB function `glmval`). The agreement is excellent. Notice for the 9–11 Hz frequency band the lack of modulation in the estimated conditional intensity, which suggests that the probability of spiking is not affected by the phase of the LFP in the 9–11 Hz frequency range. Let's now repeat this analysis but instead bandpass-filter the LFP data for 44–46 Hz; we choose this frequency interval motivated by the spike-field coherence results (see figure 11.4). Now, for this frequency interval, we find a modulation of the estimated conditional intensity, with an increase in the probability of a spike near 0 radians. These results illustrate the close correspondence between the FTA and GLM procedures. An important advantage of the GLM approach is the ability to estimate confidence intervals. The confidence intervals in figure 11.7 are estimated in the `glmval` function and returned as outputs `dylo` and `dyhi`.

For the LFP data filtered at 44–46 Hz, let's check the significance of the parameters related to the LFP phase, β_1 and β_2, via a Wald test (see chapter 9):

```
pval1=stats1.p(2);       %Significance of parameter beta_1.
pval2=stats1.p(3);       %Significance of parameter beta_2.
```

We find that β_1 is highly significant (`pval1=1.2903e-52`) and β_2 is not significant (`pval2=0.7087`), and we conclude from (11.5) that the firing rate is highly dependent on the cosine of the LFP phase.

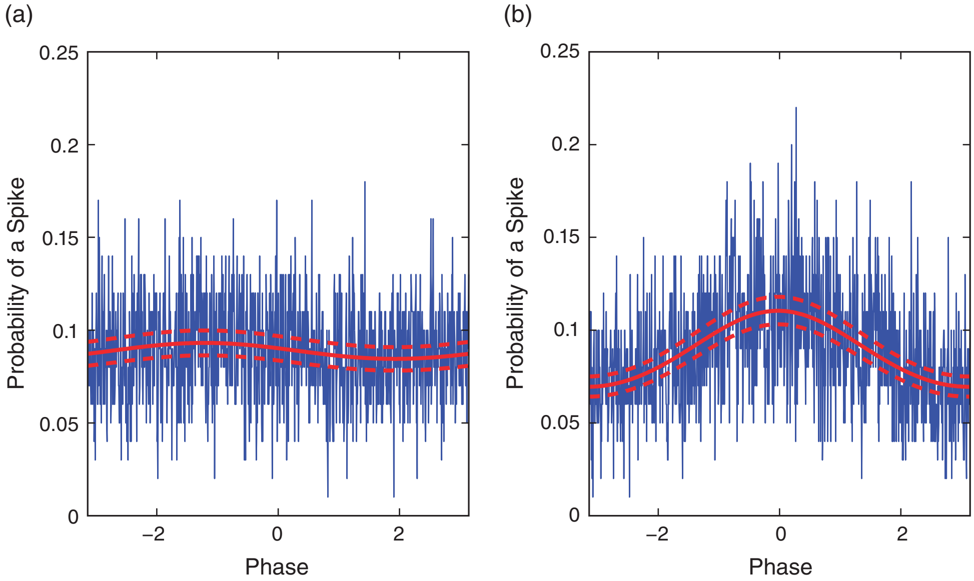

Figure 11.7

FTA (*blue*) and GLM (*red*) estimates of spike-field association for data filtered at (*a*) 9–11 Hz and (*b*) 44–46 Hz. Red dashed lines indicate 95% confidence intervals.

In chapter 9, we showed that for nested models (where one model can be made equivalent to the other by setting some parameters to specific values), under the null hypothesis that the data arise from the smaller model, the difference in the deviance between the two models should have a chi-square distribution where the number of degrees of freedom is equal to the number of extra parameters in the larger model. In this case, let's compare the model (11.5) to a model that lacks dependence on the LFP phase (i.e., a reduced model in which β_1 and β_2 are set to zero). First, we must construct and estimate this reduced model. In MATLAB,

```
X0 = ones(size(phi));                  %Define reduced predictor.
                                       %Fit reduced model.
[b0,dev0,stats0]=glmfit(X0,Y,'poisson','constant','off');
```

Here `X0` is a constant predictor, and we omit the default constant term in `glmfit`. Then, to compute the p-value for this test in MATLAB,

```
pval = 1-chi2cdf(dev0-dev1,2);         %Compare two nested GLMs.
```

Q: Why do we set the second input to the function `chi2cdf` equal to 2?

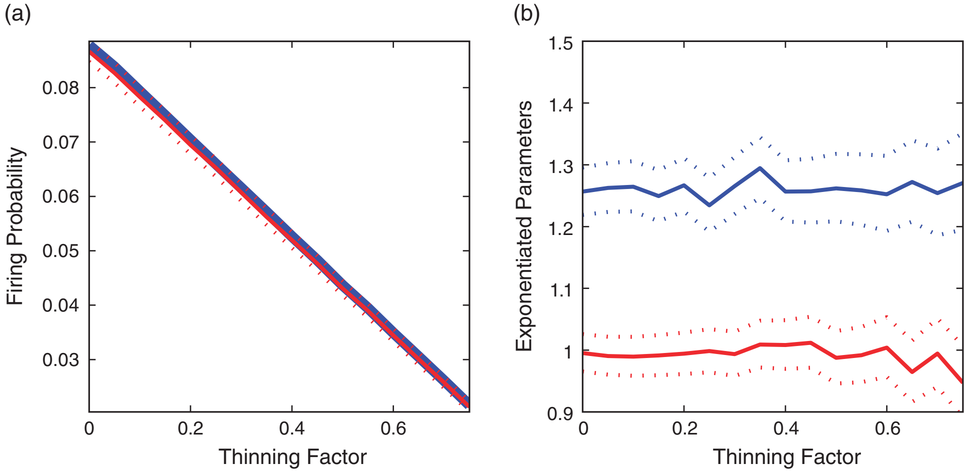

Figure 11.8

GLM framework identifies consistent spike-field association despite changing firing rates. (*a*) Direct estimate of the probability of a spike (*blue*) and GLM estimate of e^{β_0} (*red*) are consistent. (*b*) Estimates of e^{β_1} (*blue*) and e^{β_2} (*red*) do not change significantly with thinning factor. Dotted lines indicate 95% confidence intervals.

We find `pval=0` and would therefore be very unlikely to see this result if the reduced model (i.e., the model lacking dependence on the LFP phase) were correct.

Finally, let's examine how thinning the spiking data impacts the results of the GLM procedure. We choose a range of thinning factors (see figure 11.6) and repeat the analysis for LFP filtered at 44–46 Hz to recompute the FTA and estimate the GLM model for each thinning factor. Figure 11.8 shows how the estimates of the exponentiated model parameters (e^{β_0}, e^{β_1}, and e^{β_2}) vary as a function of thinning factor. As the thinning factor increases, the probability of a spike decreases. This probability can be estimated from the spike train data as

```
p = mean(sum(n,2)/(N));
```

where `n` corresponds to the thinned spike train.

Figure 11.8a plots the estimate of the probability versus e^{β_0} estimated from the GLM model. The two are in excellent agreement; as expected, as the thinning factor increases, the probability of a spike decreases. Figure 11.8b shows the exponentiated estimates of the other model parameters, e^{β_1} and e^{β_2}. Recall that these parameters represent the impact of the LFP phase on the firing rate. There are two important features to notice about these estimates. First, the thinning factor does not affect the model estimates. The changing firing rate is captured in the parameter β_0 and prevented from impacting the estimates of the spike-field association expressed in the other model parameters. Second, the exponentiated parameter e^{β_1} is significantly above 1, which indicates a significant association between the cosine phase of the LFP in the 44–46 Hz frequency band and the spiking.

Summary

In this chapter, we considered associations between data recorded from different spatial scales: the macroscale LFP and the microscale spiking. We developed methods to visualize the associations between scales, and applied tools developed in other chapters, such as the spectrum and coherence. We computed the spike-field coherence and found a strong association between the spatial scales near 45 Hz despite only the weak appearance of this rhythm in the spectra. We also considered the impact of firing rate on the spike-field coherence and illustrated that as the firing rate decreases, so does the spike-field coherence [32].

To account for the impact of firing rate on coherence we implemented a GLM, in which the firing rate depends on the LFP phase. In general, GLMs provide a powerful tool to estimate spike-field associations. The example considered here illustrates the ability of the GLM framework to estimate the influence of the LFP phase on the spiking and avoid the confounding effect of a changing firing rate. For details describing this approach and its extensions, see [33].

Problems

11.1. Analyze the properties of the spike trains in the dataset `Ch11-spikes-LFP-1.mat` using the tools developed in chapter 8. Analyze the evoked response present in the LFP using the tools developed in chapter 2. Consider these results in the context of your spike-field analysis.

11.2. Load the file `Ch11-spikes-LFP-2.mat`, available at

`http://github.com/Mark-Kramer/Case-Studies-Kramer-Eden`

into MATLAB. You will find three variables. The variable `y` corresponds to the LFP data, in units of millivolts. The variable `n` corresponds to simultaneously recorded binary spiking events. The variable `t` corresponds to the time axis, in units of seconds. Both `y` and `n` are matrices, in which each row indicates a separate trial, and each column indicates a point in time. Use these data to answer the following questions.

a. Visualize the data. What rhythms do you observe? Do you detect associations between the LFP and spikes?
b. Plot the spectrum versus frequency for these data. Are the dominant rhythms in the spectrum consistent with your visual inspection of the data?
c. Compute and display the STA and FTA. Do you find evidence for associations between the two types of data?

d. Compute and display the spike-field coherence using the methods developed in this chapter. Do you find evidence for spike-field coherence?

e. Apply the GLM model (11.5) to these data. Are the results consistent with your analysis of spike-field coherence?

f. Describe your results, as you would to a colleague or collaborator.

11.3. Load the file `Ch11-spikes-LFP-3.mat`, available at

`http://github.com/Mark-Kramer/Case-Studies-Kramer-Eden`

into MATLAB. You will find three variables. The variable `y` corresponds to the LFP data, in units of millivolts. The variable `n` corresponds to simultaneously recorded binary spiking events. The variable `t` corresponds to the time axis, in units of seconds. Both `y` and `n` are matrices, in which each row indicates a separate trial, and each column indicates a point in time. Use these data to answer the following questions.

a. Visualize the data. What rhythms do you observe? Do you detect associations between the LFP and spikes?

b. Plot the spectrum versus frequency for these data. Are the dominant rhythms in the spectrum consistent with your visual inspection of the data?

c. Compute and display the STA and FTA. Do you find evidence for associations between the two types of data?

d. Compute and display the spike-field coherence using the methods developed in this chapter. Do you find evidence for spike-field coherence?

e. Apply the GLM model (11.5) to these data. Are the results consistent with your analysis of spike-field coherence?

f. Describe your results, as you would to a colleague or collaborator.

11.4. In this question, we consider a simple example that illustrates the fundamental features of spike-field coherence. Let's consider the case in which the field is a sinusoid plus Gaussian noise, and the spike train is drawn from a Bernoulli distribution. For a Bernoulli distribution, the probability of a spike at any time is not related to previous spiking behavior. In this case, we also assume no relation between the field and point process. Therefore, we expect to find no spike-field coherence. Let's simulate some synthetic data in MATLAB, compute the spike-field coherence, and see what we find.

As a first step, let's create 100 trials of multiscale data, each trial of 1 s duration with a sampling rate of 1000 Hz. We define these parameters in MATLAB:

```
K = 100;        %Define no. of trials.
```

```
N = 1000;        %Define no. of samples per trial.
dt = 0.001;      %Define sampling interval.
```

Now, let's define the synthetic data. We create in each trial a field, which here will be a sinusoid; and a spike train, which here will be drawn from a Bernoulli distribution with a probability p of a spike in each sampling interval. In MATLAB,

```
y = zeros(K,N);          %Matrix to hold field data.
n = zeros(K,N);          %Matrix to hold spike data.
for k=1:K                %For each trial ...
    %...define the LFP as a 10 Hz sinusoid + noise.
    y(k,:) = sin(2.0*pi*(1:N)*dt * 10)+0.1*randn(1,N);
    %...draw spikes from a Bernoulli distribution lambda=0.01
    n(k,:) = binornd(1,0.01,1,N);
end
```

In this code, the frequency of the sinusoid is set to 10 Hz, and the probability of a spike in each sampling interval is set to $p = 0.01$. These choices are reasonable yet arbitrary; as part of this simulation experiment, you are encouraged to consider the impact of different simulation choices.

With these synthetic multiscale data defined, repeat the analysis performed in this chapter. In particular, estimate

a. the spectra of the field and spiking data,

b. the STA and FTA,

c. the spike field coherence,

d. the GLM model (11.5).

Do the results of each method match your expectations? In particular, do you expect to observe spike-field coherence between these simulated data?

11.5. Let's now consider a simulation in which we expect a nonzero spike-field coherence. To produce a multiscale interaction, we must introduce a relation between the spikes and the field. We do so by making the probability of a spike in each time interval a function of the field. Let's fix the number of trials (`K=100`), the number of samples per trial (`N=1000`), and the sampling interval (`dt=0.001`) at the same values used in the first simulation example. Then let's define the spike and field data for each trial. In MATLAB,

```
f = 0.01;            % Parameter for scaling of rate.
b = 1;               % Parameter for background spiking.
y = zeros(K,N);      % Matrix to hold field data.
```

```
n = zeros(K,N);      % Matrix to hold spike data.
for k=1:K            % For each trial ...
  % ...define the LFP as a 10 Hz sinusoid + noise.
  y(k,:) = sin(2.0*pi*(1:N)*dt * 10)+0.1*randn(1,N);;
  % ...draw spikes from a Bernoulli distribution,
  p = f*(b+exp(y(k,:)));%...with probability dependent
  n(k,:) = binornd(1,p,1,N);%...LFP.
end
```

Note that the probability p of a spike in each time interval depends on three factors: (1) an overall scaling term `f`, (2) a baseline level of probability `b`, and (3) the exponentiated field `exp(y)`. We exponentiate the field y so that the probability is always positive. In this way, the probability of a spike depends on the field.

With these synthetic multiscale data defined, repeat the analysis performed in this chapter. In particular, estimate

a. the spectra of the field and spiking data,
b. the STA and FTA,
c. the spike field coherence,
d. the GLM model (11.5).

Do the results of each method match your expectations? In particular, do you expect to observe spike-field coherence between these simulated data?