

Estimating receptive fields in the presence of spike-time jitter

TIM GOLLISCH

Department of Molecular and Cellular Biology, Harvard University, 16 Divinity Avenue, Cambridge, MA 02138, USA

(Received 29 June 2005; revised 22 December 2005; accepted 4 January 2006)

Abstract

Neurons in sensory systems are commonly characterized by their receptive fields. These are experimentally often obtained by reverse-correlation analyses, for example, by calculating the spike-triggered average. The reverse-correlation approach, however, generally assumes a fixed temporal relation between spike-generating stimulus features and measured spikes. Temporal jitter of spikes will therefore distort the estimated receptive fields. Here, a novel extension of widely used reverse-correlation techniques (spike-triggered average as well as spike-triggered covariance) is presented that allows accurate measurements of receptive fields even in the presence of considerable spike-time jitter. It is shown that the method correctly recovers the receptive fields from simulated spike trains. When applied to recordings from auditory receptor cells of locusts, a considerable sharpening of receptive fields as compared to standard spike-triggered averages is observed. In addition, the multiple filters that are obtained from a conventional spike-triggered covariance analysis of these data can be collapsed into a single component if spike jitter is accounted for. Finally, it is shown how further effects on spike timing, such as systematic shifts in spike latency, can be included in the approach.

Keywords: *Receptive field, reverse correlation, spike-triggered analysis, spike timing, jitter*

Introduction

Our senses provide us with information about various facets of the world around us, and neurons throughout the nervous system have been shown to encode different aspects of sensory stimuli in their activity. The sensitivity of a neuron to activation of particular stimulus regions is captured by the neuron's receptive field. Many visual neurons, for example, are sensitive to light stimulation in a particular region of visual space and over a particular temporal window in the past (Kuffler 1953; Hubel & Wiesel 1962; Meister & Berry 1999; Reinagel & Reid 2000; Reich et al. 2000); these aspects are combined in the spatio-temporal receptive field of the cell. In the auditory system, the spectro-temporal receptive field (STRF) of a neuron describes the neuron's sensitivity to the time course and the frequency content of acoustic stimuli (Eggermont et al. 1983; Kim & Young 1994; Nelken et al. 1997; deCharms et al. 1998).

Accurately measuring receptive fields and determining how neural responses relate to their activation form a cornerstone in many electrophysiological investigations. Analyzing the temporal characteristics of receptive fields can provide considerable insight about the dynamics of

sensory processing and the representation of non-stationary stimuli. In the auditory system, where temporal properties of stimuli and responses are thought to be of particular importance, investigations of the structure of STRFs have recently helped elucidate, for example, the influence of natural stimulus statistics on the encoding process (Theunissen et al. 2000), the hierarchy of auditory processing (Sen et al. 2001), functional differences of cortical areas (Linden et al. 2003), task-related plasticity of receptive fields in auditory cortex (Fritz et al. 2003), and nonlinear interactions in the sensory representation (Machens et al. 2004).

Because of the importance of the concept of receptive fields, much research has been dedicated to the development of techniques that yield accurate descriptions of the receptive field's features. One of the most common and popular methods is based on presenting the neuron under investigation with a broad-band stimulus and analyzing which parts of this stimulus cause the neuron to fire, a technique that is called "reverse correlation". In the simplest case, the average stimulus segment preceding a spike, the "spike-triggered average" (STA) can be used to estimate the receptive field (de Boer & Kuyper 1968; de Boer & de Jongh 1978; Dayan & Abbott 2001). The generic model that underlies reverse-correlation techniques is composed of a linear filter in the appropriate stimulus space followed by a nonlinear transformation that may include, for example, thresholding and saturation. This results in the neuron's firing rate, and according to this rate the individual spikes are drawn independently by a random process. This model is often called linear–nonlinear (LN) cascade or linear–nonlinear–Poisson (LNP) cascade to emphasize the random nature of spike generation in the final step (Korenberg & Hunter 1986; Simoncelli et al. 2004). Its appeal stems from its mathematical simplicity combined with its straightforward interpretation: while the linear filter captures the receptive field, the nonlinearity accounts for how the activation of the receptive field is transformed into the neural response. If a model's response to Gaussian white noise is investigated, the STA is known to be an accurate estimate of the linear filter that constitutes the first step of the LNP cascade model (Chichilnisky 2001).

As an extension of the STA approach, analysis of the "spike-triggered covariance" (STC), the covariance matrix of stimulus segments preceding a spike, allows the determination of receptive fields when multiple linear filters are required to describe the neuron's sensitivity to particular regions in stimulus space (de Ruyter van Steveninck & Bialek 1988; Schwartz et al. 2002; Touryan et al. 2002; Lewis & van Dijk 2004; Simoncelli et al. 2004). The theoretical foundations and properties of STA and STC estimators of receptive fields are well understood (Paninski 2003). These techniques, however, rely on the assumption that the spike-generating stimulus segment precedes the spike with a fixed temporal relation. If this assumption is violated, for example if the spike timing is subject to noise that jitters the exact location of a spike, the receptive-field estimates are distorted and important temporal features may be washed out.

It is demonstrated here that it is possible to overcome this limitation of reverse-correlation analysis by taking spike jitter explicitly into account in the underlying model. Recent work has shown that STA estimates of receptive fields can be severely affected by spike jitter and that re-aligning the spike-generating stimulus segments before averaging can improve these estimates (Aldworth et al. 2005; Chang et al. 2005). Here, a generic framework is presented for extending STA and STC analyses in order to capture spike jitter as well as other effects on spike timing such as systematic shifts in spike latency. This results in an iterative algorithm for spike-train data analysis, which is closely related to the Expectation Maximization (EM) algorithm for obtaining maximum-likelihood estimators.

The algorithm is based on the idea that an estimate of the LNP cascade model can be used to assess how likely different jitter values are for each observed spike, even if only a single stimulus repeat is available for the analysis. This allows us to weight each stimulus

segment in the vicinity of a spike according to its likelihood of having generated the spike. By taking all these segments into account, the estimates of the LNP cascade model can be successively improved, resulting in a self-consistent model when this procedure converges. For both STA and STC analysis, it is first shown in the following that the algorithm recovers the linear filters and nonlinear transformations in simulated data in a reliable and robust manner. Subsequently, the applicability of the technique to physiological spike train data from auditory receptor neurons of locusts is demonstrated. Finally, it is shown how further effects on spike timing, such as systematic stimulus-dependent shifts of spikes, can be included in the approach.

Cascade models with spike jitter

The LNP cascade is a fairly general model class often used to investigate neuronal response features. We will here use the LNP model as a basis for the estimation of receptive fields as well as for generating simulated spike trains to test the method. For simplicity, only models whose input is a stimulus s that depends on time t only (such as full-field light intensity or sound intensity over time) are investigated, but the approach and the main conclusions also apply to cases where the stimulus depends on further variables, such as visual space or acoustic frequency. In all simulations, the stimulus $s(t)$ is a Gaussian white noise signal with zero mean and unit variance, discretized in time with a time step of $\Delta t = 1$ ms.

In the first step of the LNP model, the stimulus $s(t)$ is linearly filtered. We explicitly distinguish two types of models, depending on whether the first model stage is composed of a single or of multiple parallel filters. The single-filter model is typically analyzed with the STA; the multi-filter variant forms the basis of the STC analysis.

In the case of a single linear filter $f(n)$ of length N , the output $y(t)$ of the first model step is given by the convolution of $f(n)$ and the stimulus $s(t)$:

$$y(t) = \sum_{n=0}^{N-1} s(t-n) \cdot f(n). \quad (1)$$

In the second model step, $y(t)$ is transformed by a static nonlinearity $g(y)$ into the spike probability $r(t)$:

$$r(t) = g(y(t)). \quad (2)$$

Spikes are generated according to this spike probability $r(t)$ by comparing the output of a random number generator with $r(t)$ for each time bin. To extend the LNP model, each spike is finally jittered in time by a random amount τ drawn from a distribution $p_j(\tau)$, which is here either a Gaussian distribution or a uniform distribution over a certain interval of jitter values. This extended model will be called linear–nonlinear–Poisson–jitter (LNPJ) cascade. It is depicted schematically in Figure 1A. The particular shapes of the linear filter, the nonlinearity, and the jitter distribution used in the simulations will be shown together with the results of the analysis.

Note that, because of the jitter, the spike probability $r(t)$ corresponds to the probability of generating a spike, but not to the probability $r_{\text{obs}}(t)$ of observing a spike at time t . In fact, while the probability of spike generation $r(t)$ is generated by an LN cascade, $r_{\text{obs}}(t)$ can be viewed as resulting from an LNL cascade because of the additional linear filtering with the jitter distribution $p_j(\tau)$. In principle, if the full spike rate $r_{\text{obs}}(t)$ were measured, generic, though data-intensive parameter estimation procedures based on the theory of LNL cascades (Hunter & Korenberg 1986; Korenberg & Hunter 1986) would be applicable. We will here follow a different route and discuss parameter estimation that directly uses

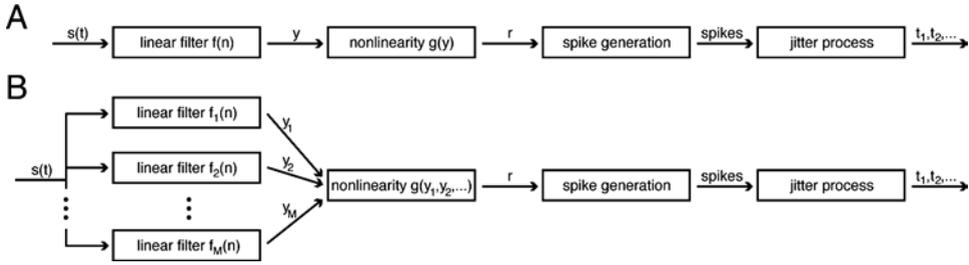


Figure 1. LNPJ cascade models. (A) Model with a single linear filter. The stimulus $s(t)$ is convolved with the filter $f(n)$. The resulting filter output y is transformed with the nonlinearity $g(y)$ to yield the spike probability r , according to which spikes are generated by a random process. Finally, each spike is jittered by a random time τ drawn independently from the jitter distribution $p_j(\tau)$. This results in the observed spike times t_k . (B) Model with multiple linear filters. In contrast to the single-filter model, the stimulus is convolved by M linear filters in parallel. All filter outputs y_m are combined by a multi-dimensional nonlinearity $g(y_1, \dots, y_M)$.

individual spike times and does not rely on the full rate $r_{\text{obs}}(t)$. The method will thus be applicable also to the case where only a single stimulus repeat is available. Furthermore, the explicit spike-jitter model of the last stage allows us to use a low-dimensional representation of the corresponding filter, i.e., jitter distribution, and can easily be extended to include other effects on spike timing that break the analogy to the LNL cascade (see, for example, the spike-time shifts discussed below).

The analogy with the LNL cascade shows us, however, that the nonlinearity, which is inevitable for most spiking neurons simply because of the existence of a threshold, will be essential for distinguishing between temporal stimulus integration and spike jitter. Without the nonlinearity, the two linear filters would collapse into one, leaving the contributions of stimulus integration and jitter ambiguous without additional measurements (such as the intracellular membrane potential) or additional assumptions about the shapes of the filters.

For the multi-filter model, the above scheme of the LNPJ cascade is modified, as shown in Figure 1B, by using M different linear filters $f_m(n)$, $m = 1, \dots, M$, in parallel,

$$y_m(t) = \sum_{n=0}^{N-1} s(t-n) \cdot f_m(n). \quad (3)$$

The subsequent nonlinearity then combines all outputs y_m of the linear filters to yield the spike probability $r(t)$:

$$r(t) = g(y_1(t), \dots, y_M(t)). \quad (4)$$

In the simulations, this was implemented by calculating the sum of squares of all $y_m(t)$ and then applying a one-dimensional nonlinearity to this sum similar to the case of the single linear filter. Spikes were then generated according to the spike probability $r(t)$ and subsequently jittered as explained above.

Iterative algorithm for analyzing spike trains with jitter

The challenge that we now pose for the data analysis is to extract the linear filter, the static nonlinearity, and the jitter distribution from simulated as well as from electrophysiologically recorded spike trains. We will first discuss the case where a single linear filter is assumed (extension of STA analysis) and subsequently explain how this is adjusted to incorporate multiple linear filters (extension of STC analysis). The approach implemented in the algorithm follows the intuitive idea that information about the spike jitter can be obtained from

how well the stimulus segments in the vicinity of a spike match the estimate of the receptive field. This allows us to calculate, based on the LNPJ model, the probability that a given spike observed at time t_k has some jitter τ_k . These probabilities can then be taken into account to obtain improved estimates of the model components. The basic idea for this is the following: if the actual jitter values τ_k of each spike were known, the parameters of the LNPJ model could be easily estimated by reverse correlation, triggered at the times of spike generation $t_k - \tau_k$. Since we only have the probabilities of the τ_k instead of the actual values, we will average the estimators of the model components over all jitter values in a Bayesian fashion. This effectively amounts to attaching a weight to each stimulus segment in the vicinity of a spike and using this enlarged set of stimuli for a weighted reverse-correlation analysis. This procedure can be iterated, following a self-consistency principle: the model parameters are derived from the weights for the different stimulus segments, and the weights, in turn, result from the model. When these two steps are in agreement, the algorithm has converged to a self-consistent solution.

The approach is related to the Expectation Maximization (EM) algorithm (Dempster et al. 1977), which allows an iterative maximum-likelihood estimation of model parameters in the case of incomplete or hidden data. The mathematical connection to the EM algorithm is explained in the Appendix. Example programs for the implementation of the algorithm can be found as C code at <http://www.fas.harvard.edu/~gollisch/jitteranalysis/>.

Outline of the algorithm

For each spike k , the weights of the stimulus segments in the vicinity of the spike are denoted by $w_k(\tau)$. These weights represent the probability that spike k was generated by the stimulus segment that terminates at $t_k - \tau$ and was subsequently shifted to t_k by the jitter τ . One iteration of the algorithms consists of the following operations:

1. Determine the weights $w_k(\tau)$ for each spike in the vicinity of its actual recorded time t_k based on the current model estimate.
2. Update the estimate for the linear filter $f(n)$ based on the spike times and the corresponding weights.
3. Update the estimate for the nonlinearity $g(y)$.
4. Update the estimate for the distribution $p_j(\tau)$ of jitter values based on the distribution of weights.

We will now discuss in more detail how the individual steps are implemented. The key ingredient to the algorithm is the calculation of the weights $w_k(\tau)$ for each spike k , based on the stimulus in the vicinity of the observed spike times t_k and given the model parameters $f(n)$, $g(y)$, and $p_j(\tau)$. For running the algorithm, of course, the true values of these parameters are not known; instead, the current estimates of each iteration are used to calculate the weights. Following the meaning of the weights stated above, it is clear that to calculate $w_k(\tau)$, we need the probability that the stimulus segment terminating at $t_k - \tau$ generates a spike as well as the probability that a jitter τ occurs. More formally, $w_k(\tau)$ is given by the posterior probability $p(\tau | \text{data})$ that the jitter for this particular spike is τ , given the observed data according to Bayes' rule:

$$w_k(\tau) = p(\tau | \text{data}) = \frac{p(\text{data} | \tau) \cdot p_{\text{prior}}(\tau)}{p(\text{data})}, \quad (5)$$

where $p(\text{data}) = \sum_{\tau'} p(\text{data} | \tau') \cdot p_{\text{prior}}(\tau')$ is the prior probability of the data and yields effectively a normalization constant. The data are in principle given by the whole spike train, but

the jitter probabilities of a given spike are mostly determined by the time of this spike itself. We therefore assume that the jitter probabilities for each spike are independent of the other spikes and use only the observed spike time t_k as the data. Effects of nearby spikes $k' \neq k$ on the weights $w_k(\tau)$ can be safely neglected as long as the typical interspike intervals are larger than the relevant jitter. The prior probability $p_{\text{prior}}(\tau)$ over the jitter τ is given in the model by $p_j(\tau)$. $p(\text{data} | \tau)$ is the probability that a spike is generated at time $t_k - \tau$ and therefore calculated from the model as $r(t_k - \tau)$ according to Equations 1 and 2. The weights $w_k(\tau)$ are thus obtained by multiplying the spike probabilities $r(t_k - \tau)$ with the jitter probabilities $p_j(\tau)$ and normalizing:

$$w_k(\tau) = \frac{1}{Z} \cdot r(t_k - \tau) \cdot p_j(\tau) \quad \text{with} \quad Z = \sum_{\tau'} r(t_k - \tau') \cdot p_j(\tau'). \quad (6)$$

The procedure for obtaining the weights is depicted schematically in Figure 2. As the jitter is not expected to reach arbitrarily large values, in practice we only regard such τ in the calculations whose absolute values are smaller than some maximal jitter τ_{max} , which is chosen large enough to encompass the relevant range of jitter values.

After obtaining the weights $w_k(\tau)$, the model parameters are updated. The linear filter is characterized by its values at discretized time points, $f(n)$, $n = 0, \dots, N - 1$. For the presented results, filters that extended over $N = 40$ time points were used. For a known set of spike-jitter values $\{\tau_k\}$, the $f(n)$ could be obtained from a reverse correlation triggered at the time points of spike generation $t_k - \tau_k$: $f(n) = \sum_k s(t_k - \tau_k - n)$. The average over the posterior distribution $w_k(\tau)$ of each jitter value is therefore

$$f(n) = \sum_k \sum_{\tau} w_k(\tau) \cdot s(t_k - \tau - n). \quad (7)$$

Note that using the posterior distribution $w_k(\tau)$ for each spike k separately is identical to averaging over the posterior distribution of all sets of jitter values $\{\tau_k\}$ because of the assumed independence of each jitter with respect to the other spikes: the posterior distribution of the set $\{\tau_k\}$ is simply the product of the individual (marginal) distributions $w_k(\tau_k)$. Note also that Equation 7 can be interpreted as a weighted reverse correlation that takes all stimulus segments in the vicinity of a spike into account.

Subsequently, the $f(n)$ are normalized so that their sum of squares equals one. Such a normalization is necessary to make the model unambiguous, as a scaling of the filter corresponds to a scaling of the nonlinearity $g(y)$ along the abscissa.

To regularize the filter and avoid overfitting, a simple smoothing procedure was used here. The smoothing was implemented by substituting $f(n)$ with $f(n)/2 + (f(n-1) + f(n+1))/4$, with $f(-1) = f(N) = 0$ at the edges. The danger of overfitting is a particular concern for methods based on self-consistency. When using the algorithm, one should be aware that, without regularizing the filters, small noise-related fluctuations in the receptive-field estimates could be amplified by assigning higher weights to such jitter values that lead to similar fluctuations. The applied simple smoothing procedure greatly reduces these effects, but depending on the particular data at hand, it may be desirable to combine the algorithm with more systematic approaches of regularization that incorporate smoothness or sparseness constraints (Sahani & Linden 2003; Machens et al. 2004). A suitable control to avoid overfitting is model cross-validation, which will be discussed below.

To calculate the nonlinearity $g(y)$, we first need to estimate the filter output $y(t)$ from the stimulus and the spike probability $r(t)$ from the observed spike times for each time t . $y(t)$ is obtained by again convolving the corresponding stimulus segment with the filter $f(n)$ as in Equation 1. To obtain the corresponding $r(t)$ we can use the interpretation that the

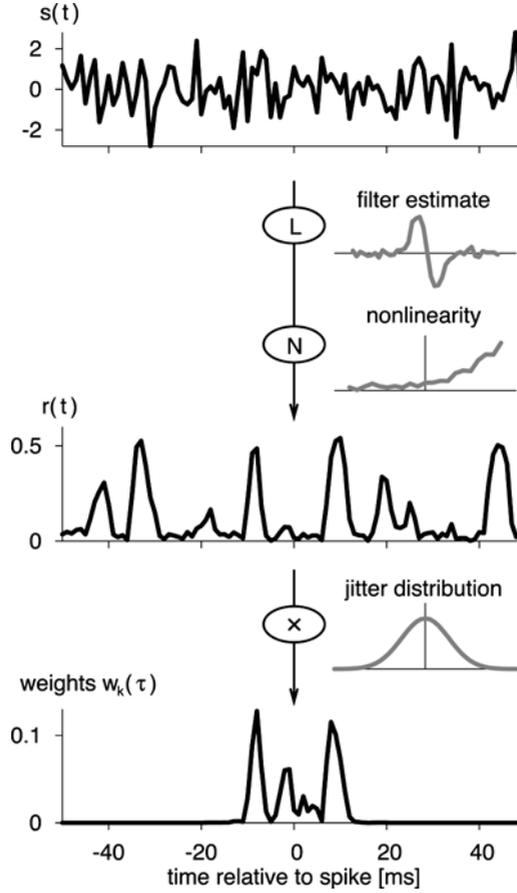


Figure 2. Calculating the weights. For each spike k , the weights $w_k(\tau)$ for the jitter values τ are obtained through a sequence of steps. The stimulus $s(t)$ around a spike (top row) is first convolved with the current estimate of the linear filter and subsequently transformed by the current estimate of the nonlinearity to obtain the spike probability $r(t)$ (middle row). The weights (bottom row) are obtained as a normalized product of the spike probability with the current estimate of the jitter distribution, which is parametrized as a Gaussian distribution. Note that this multiplication leads to a suppression of contributions from the spike probability for time points that are not near the spike. The estimates of the filter, the nonlinearity, and the jitter distribution used in the calculation are shown in gray. In this example, the spike probability at the actual time $t = 0$ of the observed spike is very small, indicating that it is more likely that the spike was generated by stimulus segments that terminate around 10 ms earlier or later and subsequently shifted to its observed location by the jitter.

$w_k(\tau)$ correspond to new weighted spikes at all times $t = t_k - \tau$ in the vicinity of an observed spike. We therefore calculate $r(t)$ as the sum of the weights $w_k(t_k - t)$ over all spikes k in the vicinity of t , divided by the number of stimulus repeats N_{rep} . Since we only consider jitter values whose absolute values are smaller than τ_{max} , this concerns for each time t only few spikes or no spike at all, in which case $r(t) = 0$. More formally, $r(t)$ also follows as the average over the posterior jitter distribution: If the τ_k were known, $r(t)$ could be estimated as $1/N_{\text{rep}} \cdot \sum_k \delta_{t, t_k - \tau_k}$, where δ_{t_1, t_2} is the Kronecker delta, which equals one for $t_1 = t_2$ and zero otherwise. The average over the posterior jitter distribution is therefore given by

$$r(t) = \frac{1}{N_{\text{rep}}} \sum_k \sum_{\tau} w_k(\tau) \cdot \delta_{t, t_k - \tau} = \frac{1}{N_{\text{rep}}} \sum_k w_k(t_k - t). \quad (8)$$

With the collection of pairs $(y(t), r(t))$ for all time points t , the nonlinearity $g(y)$ can then be determined according to a specific representation of $g(y)$. Here a look-up table is used with entries \bar{y} and \bar{r} that correspond to the average values of $y(t)$ and $r(t)$ in appropriately chosen bins. As an alternative, a parametrized functional form could be applied, but the look-up table has the advantage that it does not impose a particular functional relationship between y and r . The binning is achieved by dividing the pairs $(y(t), r(t))$ into B bins so that the B -th fraction with the smallest $y(t)$ are combined in the first bin, the next B -th fraction in the second bin and so on, so that all B bins are equally populated. Typically $B = 40$ bins were used here.

To evaluate this representation of $g(y)$ for a given value of y , the two bins nearest to y are linearly interpolated. If y is smaller than the smallest \bar{y} value in the bins or larger than the largest value, the resulting r is simply set to the corresponding \bar{r} value of the bin with the smallest or largest \bar{y} value, respectively.

The jitter distribution $p_j(\tau)$ is here modeled as a Gaussian distribution with zero mean and variance σ_j^2 as the only parameter. The Gaussian distribution is likely to capture spike jitter in many experimentally encountered situations, and its mathematical simplicity makes the estimation of the variance straightforward. Note, though, that the applicability of the algorithm is not limited to this particular parametrization of the jitter distribution. Also note that the zero-mean condition poses no restriction for the algorithm, as a non-zero mean of the jitter distribution is equivalent to a time translation of the linear filter. For a set of jitter values $\{\tau_k\}$, σ_j^2 can be calculated as the average of τ_k^2 , $\sigma_j^2 = 1/N_{\text{spikes}} \cdot \sum_k \tau_k^2$, where N_{spikes} denotes the number of spikes used for the calculation. This is the maximum-likelihood estimator of the variance. The average over the posterior jitter distribution is therefore obtained as the weighted average of τ^2 over all spikes and jitter values:

$$\sigma_j^2 = \frac{1}{N_{\text{spikes}}} \sum_k \sum_{\tau} w_k(\tau) \cdot \tau^2. \quad (9)$$

The following starting values are used to initialize the algorithm: the filter $f(n)$ is obtained from a conventional STA. The look-up table for $g(y)$ is constructed as explained above, but since no weights have been calculated yet, $r(t)$ is simply determined as the number of observed spikes that fell into the corresponding time bin, divided by N_{rep} . For the standard deviation σ_j of the jitter distribution, no particular procedure is employed to determine the starting value, but instead, a set of different initial values is used to confirm that the algorithm converges to the same final estimates independent of the starting value for σ_j .

The algorithm is typically carried out for several hundred iterations, and convergence is checked by examining the evolution of the model parameters. Alternatively, a predefined termination rule could be implemented for stopping the iteration.

Adjustments of the algorithm to the case of multiple linear filters

When a model with a single linear filter is insufficient to capture the details of the neuron's receptive field, the STA is often replaced by the analysis of the spike-triggered covariance (STC). In this analysis technique, multiple linear filters can be extracted as eigenvectors of the covariance matrix of those stimulus segments that preceded spikes (de Ruyter van Steveninck & Bialek 1988; Simoncelli et al. 2004). The eigenvectors whose eigenvalues differ significantly from the noise level denote the relevant filters. The method works best if a small number of filters suffices to describe the receptive field. These can then be identified as the eigenvectors corresponding to those eigenvalues that are distinct from most of the other eigenvalues. Spike jitter, however, can distort these estimates in similar ways as for the STA.

To account for spike jitter in the data analysis, we adjust the iterative algorithm to the case of an LNPJ cascade with multiple parallel linear filters (Figure 1B). The main difference in implementation is that, in analogy to the conventional STC analysis, the filters are now obtained as eigenvectors of a matrix constructed from the weighted stimulus segments. Again, each spike is associated with weights for the stimulus segments in its vicinity. These weights are taken into account to calculate the matrix $C(n_1, n_2)$:

$$C(n_1, n_2) = \frac{1}{N_{\text{spikes}}} \sum_k \sum_{\tau} w_k(\tau) \cdot s(t_k - \tau - n_1) \cdot s(t_k - \tau - n_2). \quad (10)$$

The prior stimulus covariance $C_{\text{prior}}(n_1, n_2) = \langle s(t - n_1) \cdot s(t - n_2) \rangle_t$, where $\langle \cdot \rangle_t$ denotes the expectation value over time t , is subtracted from $C(n_1, n_2)$. The M linear filters are then identified from this matrix as the M eigenvectors whose eigenvalues deviate most from the zero level of non-relevant eigenvalues. Note that the number M of linear filters that are included in the algorithm has to be specified *a priori*.

As the nonlinearity $g(y_1(t), \dots, y_M(t))$ is now multi-dimensional, the simple one-dimensional binning to create the look-up table for the representation of the nonlinearity does not work any longer. Two different approaches were used here, either a two-dimensional binning or a multi-dimensional Taylor expansion. In the two-dimensional binning, the outputs $y_1(t)$ and $y_2(t)$ of two filters are associated with the spike probability $r(t)$, which is obtained from the observed spikes and the weights in the same way as for the one-dimensional case. The bins in the $y_1 - y_2$ plane are determined in the following way: the pair $(y_1(t), y_2(t))$ is transformed into polar coordinates with length $l_y(t)$ and angle $\varphi_y(t)$. According to $\varphi_y(t)$, one of B_φ equally spaced angular segments $[(k - 1) \cdot 2\pi/B_\varphi, k \cdot 2\pi/B_\varphi]$, $k = 1, \dots, B_\varphi$, is assigned, and in each of these segments, B_l bins are filled according to $l_y(t)$ as before in the one-dimensional case: the first bin contains the B_l -th fraction of the data with the smallest $l_y(t)$ and so on. The nonlinearity is then parametrized by the mean values of $l_y(t)$, $\varphi_y(t)$, and $r(t)$ in each of the $B_\varphi \cdot B_l$ bins. In the presented examples, $B_\varphi = 8$ and $B_l = 8$ or 16 were used.

As an alternative parametrization of the nonlinearity, a simple second-order multi-dimensional Taylor expansion of the nonlinearity is applied:

$$g(y_1, \dots, y_M) = a + \sum_{i=1}^M b_i \cdot y_i + \sum_{i=1}^M \sum_{j=1}^i c_{ij} \cdot y_i \cdot y_j. \quad (11)$$

The parameters a , b_i , and c_{ij} are obtained from a least-squares fit of the corresponding input-output combinations $(y_1(t), \dots, y_M(t), r(t))$. For the case $M = 2$, the binning procedure and the Taylor expansion performed similarly; all results presented here for this case were obtained with the former unless otherwise stated. While the binning procedure is limited to the case $M = 2$, however, the Taylor expansion can be used independently of the number of linear filters taken into account. Note that the simple form of the nonlinearity in Equation 11 works well in the investigated examples, although it enforces neither a threshold nor a saturation. It can thus be viewed as a minimal-assumption low-dimensional parametrization of the nonlinearity and may be substituted by more specific parametrized functions tailored to the particular system at hand to improve the model estimation procedure.

To compare the linear filters that are extracted by the algorithm with the original ones used in the simulations, it must be kept in mind that the linear filters are not uniquely determined. In fact, any other set of linear filters that spans the same stimulus sub-space is equally good as a description of the first model stage (Paninski 2003; Simoncelli et al. 2004). To assess the performance of the algorithm, we will therefore evaluate how well the original filters are represented by the combination of extracted filters. This is done by calculating the

orthogonal projection of each original filter onto the space spanned by the extracted filters. If the analysis is successful, the extracted filters should span the same space as the original filters, and thus the projections should be identical to the original filters. If an erroneous sub-space results from the analysis, on the other hand, this will be reflected in deviations of the projections from the original filters.

Extension of the algorithm to include systematic spike-time shifts

One example will be investigated for extending the iterative algorithm to include effects on spike timing beyond jitter. For the case of a single linear filter, systematic shifts in spike timing that depend on how strongly the neuron is driven by the stimulus are included in the model. In the simulations, this is implemented by shifting each spike that is generated at time t by an amount $\alpha \cdot y(t)$, where $y(t)$ is the output of the linear filter stage at time t . The proportionality constant α is typically negative to lead to shorter spike latencies when the neuron is driven more strongly. The jitter is then added on top of this shift. Note that for this scenario the observed firing rate $r_{\text{obs}}(t)$ can no longer be viewed as the result of an LNL cascade as in the case of the stimulus-independent jitter.

To account for this systematic shift in the algorithm, we include α in the iterative estimation, starting with an initial value of zero. The calculation of the weights is modified in the following way: before multiplying $r(t_k - \tau)$ with the jitter distribution $p_j(\tau)$ (cf. Equation 6), we must first account for the fact that part of the difference between the observed spike time t_k and the assumed generating spike time $t_k - \tau$ is caused by the shift. Instead of multiplying with just $p_j(\tau)$, we therefore multiply with $p_j(\tau - \alpha \cdot y(t_k - \tau))$. To update α , we use the slope of a weighted straight-line fit to the relation between τ and $y(t_k - \tau)$. The new value for σ_j^2 is then obtained by extending Equation 9 to account for the systematic shift, resulting in the updating rules

$$\alpha = \frac{\langle \tau \cdot y \rangle_{k,\tau} - \langle \tau \rangle_{k,\tau} \cdot \langle y \rangle_{k,\tau}}{\langle y^2 \rangle_{k,\tau} - \langle y \rangle_{k,\tau}^2}, \quad \sigma_j^2 = \langle (\tau - \alpha \cdot y)^2 \rangle_{k,\tau}, \quad (12)$$

where $\langle \cdot \rangle_{k,\tau}$ denotes the weighted average with weights $w_k(\tau)$ over all spikes and all τ values.

Model validation

When the algorithm is tested on simulated data, its performance can be checked by how well it recovers the parameters that were used for simulating the data. In order to test more generally how the obtained LNPJ model compares to the standard STA- and STC-based LNP models, we use two different methods of cross-validation, which are both performed on novel data, i.e., data that were not used to estimate the model parameters.

1. If firing rates are available from the data, either because the data were obtained from a simulation or because measurements with a large number of stimulus repeats were taken, we can calculate the model-explained fraction of variance of the firing rate. This is computed as $1 - \text{MSE}(r_{\text{model}}, r_{\text{data}}) / \text{VAR}(r_{\text{data}})$, where $\text{MSE}(r_{\text{model}}, r_{\text{data}})$ is the mean square error of the model-predicted firing rate $r_{\text{model}}(t)$ compared to the firing rate $r_{\text{data}}(t)$ of the data and $\text{VAR}(r_{\text{data}})$ is the variance of $r_{\text{data}}(t)$. Both these quantities are calculated as averages over time. A perfect model would yield a model-explained variance of one, and any deviation of the model from the data lowers this value. For the LNPJ model, the calculation of the model firing rate $r_{\text{model}}(t)$ simply amounts to the application of the corresponding LNL cascade. For measured spike-train data, $r_{\text{data}}(t)$

is calculated here as the peri-stimulus time histogram (PSTH) by counting the spikes that fell in each time bin.

2. Using directly individual spike times without estimating firing rates from the data, we can calculate the model likelihood. This method can be used for both simulated and measured data even if only a single stimulus repeat is available. The model likelihood is calculated as the probability that the investigated model would produce the observed spike train. To do so, again the model prediction of the firing rate $r_{\text{model}}(t)$ is calculated, and the log-likelihood follows as

$$\log L(\text{model}) = \sum_{t=t_{\text{spike}}} \log[r_{\text{model}}(t)] + \sum_{t \neq t_{\text{spike}}} \log[1 - r_{\text{model}}(t)] \quad (13)$$

where the first sum runs over all spike times and the second over all other time points.

The cross-validation calculations can easily be included in the iterative algorithm so that model performance is evaluated after each iteration. In the first iteration, before including jitter in the estimated model, this procedure then calculates the validation measures for the STA- or STC-based LNP model.

Tests and applications of the algorithm

Analysis of simulated single-filter models

If spikes are subject to temporal jitter, the estimates of the temporal aspects of receptive fields as obtained by reverse correlation can be substantially distorted. Figure 3 illustrates this with a simple example. As shown schematically in Figure 3A, spike trains were simulated by passing a white-noise stimulus through a temporal linear filter and subsequently through a static nonlinearity resulting in a spike probability. The linear filter determines the neuron's receptive field. For each time bin, the occurrence of a spike was then determined with a random-number generator according to the calculated spike probability. Each spike time was subsequently jittered by some random amount τ , drawn from a Gaussian probability distribution $p_j(\tau)$ (Figure 3B). Example spike trains obtained from a simulation with this LNPJ cascade are shown in Figure 3C.

It is well known that, for Gaussian white-noise stimulation as used in this simulation, the spike-triggered average (STA) correctly recovers the linear filter of this type of model neuron in the absence of spike jitter (Chichilnisky 2001). Here, however, the spike jitter distorts this estimate considerably (Figure 3D); instead, the STA yields the convolution of the original filter and the distribution of jitter values. Furthermore, this also results in an erroneous estimate of the neuron's nonlinearity; the true nonlinearity is much steeper than found by this estimation (Figure 3E). If the jitter distribution were known, the original filter could be recovered by deconvolution of the STA,¹ but in the typical experimental situation, the jitter distribution cannot be directly measured. With the presented iterative algorithm, we can nevertheless recover the correct receptive field of the neuron, even without prior information about how strong the spike jitter in the data was.

Figure 4 shows the performance of the described algorithm for the example introduced in Figure 3. When we let the algorithm run for several hundred iterations, we find that it quickly converges to a solution without further changes in σ_j , the linear filter, or the nonlinearity. The final estimates of the filter, the nonlinearity, and the jitter distribution (Figure 4A–C, red

¹Note, however, that even if the jitter distribution for deconvolution were known, one would face the notorious problem of noise amplification for frequencies that are poorly represented in the jitter distribution.

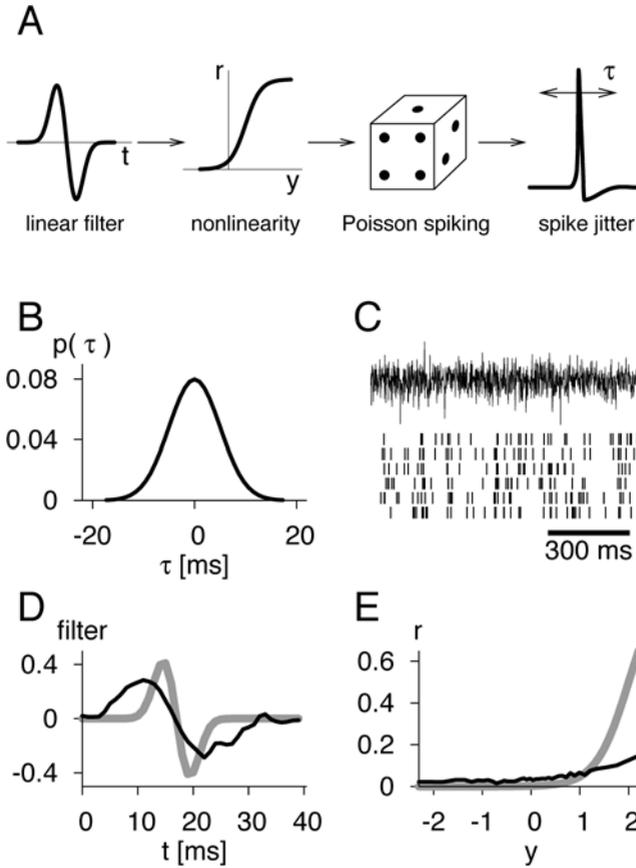


Figure 3. Effect of spike-time jitter on the spike-triggered average. (A) Schematic drawing of the model underlying the simulations as well as the data analysis. The stimulus is convolved with a linear filter, here shown as a biphasic kernel, and transformed by a nonlinearity. The output of the nonlinearity gives the spike probability, from which spikes are drawn randomly. Finally, the resultant spikes are jittered to produce the observed spike train. (B) Distribution of jitter values in the simulation. Each jitter value was drawn from a Gaussian distribution with a standard deviation of 5 ms. (C) Stimulus and example spike trains from the simulation. (D) STA (black) and actual filter used in the simulation (gray). The STA does not match the true filter because of the jitter. In fact, the STA results from the convolution of the true filter and the spike jitter distribution. (E) Nonlinearity used in the simulation (gray) and estimate obtained from the STA analysis (black). The estimated nonlinearity is considerably less steep.

lines) closely match the functions that were used for the simulation (blue lines). The iterative algorithm thus yields greatly improved estimates of the receptive field and the nonlinearity as compared to the original STA analysis (green lines). The evolution of the estimate for σ_j illustrates the process of convergence (Figure 4D). All final estimates are independent of the initial value for σ_j over a large range, as shown by the same final value of σ_j for initial values of 8 ms and 1 ms. The final estimate of σ_j deviates by about 5% from the actual standard deviation of 5 ms used in this simulation. Running the algorithm for many different instantiations of the simulation indicates that this slight underestimation of σ_j is a typical observation, but is reduced when more simulated data is included in the analysis.

The algorithm does not only recover the simulation parameters better than the STA analysis, but also leads to a quantitatively better model. This is shown by the increase of the model-explained variance of the firing rate (Figure 4E) and the model likelihood (Figure 4F).

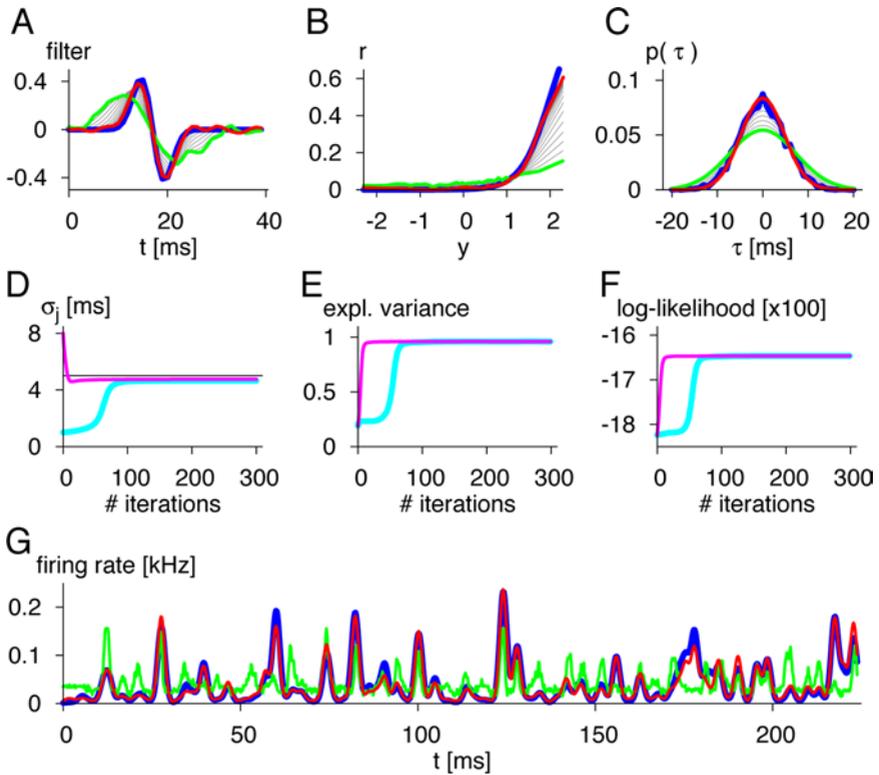


Figure 4. Recovery of receptive field and nonlinearity with the iterative algorithm. The simulation is based on about 2200 spikes in 50 s simulated time. (A) Estimate of receptive field. The filter obtained at the end of the algorithm (red) matches the true filter (blue) well, whereas the STA (green) substantially deviates. The thin gray lines show the evolution of the filter. (B) Estimate of the nonlinearity. Colors refer to the same cases as in (A). (C) Estimate of the jitter distribution. Colors refer to the same cases as in (A). The green curve here corresponds to one chosen initial value used for the algorithm. The blue curve shows the histogram of the actual jitter values, drawn from a Gaussian distribution with 5 ms standard deviation. (D) Evolution of the estimate for the standard deviation σ_j of the jitter distribution. For different starting values, σ_j converges to the same value close to the value of 5 ms used in the simulation (thin black line). (E) Model-explained fraction of variance of the firing rate evaluated after each iteration shown for both initial conditions of σ_j . (F) Model likelihood after each iteration. (G) Firing rates calculated for a novel stimulus segment for the original model (blue), the STA-based LNP model estimate (green), and the final LNPJ model estimate (red).

A direct comparison of the firing rates for the STA-based LNP model and the final estimate of the LNPJ model to the firing rate of the simulation is shown in Figure 4G. The obtained LNPJ model clearly yields a much better prediction of the firing rate. The reason for this is not so much a better representation of event timing and jitter, but rather that this model predicts the occurrence and the size of firing events with much higher fidelity than the LNP model because of its more accurate representation of the receptive field and the nonlinearity. Obtaining these more accurate representations, however, is a consequence of explicitly accounting for the jitter process in the fitted model.

In the above scenario, the jitter distributions in the simulation as well as in the parameter estimation procedure were both Gaussian. The algorithm is, however, robust enough to tolerate a difference in shape between the true underlying jitter distribution and the parametrization of the jitter distribution in the fitted model. This is illustrated in Figure 5 where the same analysis as above was performed, but now with simulated spikes that were jittered according

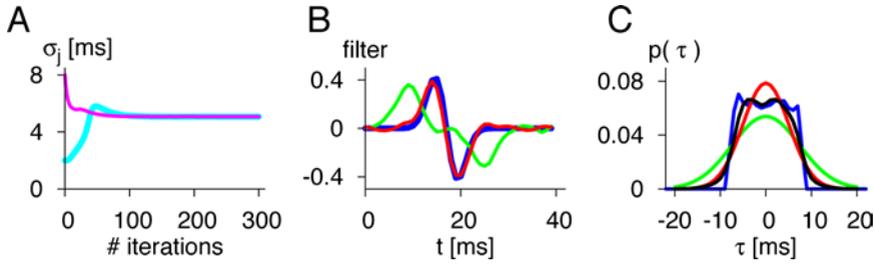


Figure 5. Performance of the algorithm for simulated data where the spike jitter was drawn from a distribution that is uniform in some range. (A) Evolution of the estimate for σ_j for two different initial values. (B) Estimate of receptive field. The final filter of the LNPJ model (red line) matches the original filter (blue line) much better than the STA (green line) does. (C) Estimate of jitter distribution. The jitter distribution is still parametrized by a Gaussian (red line), but the mean distribution of weights (black line) reveals a clear deviation from the Gaussian case and thus gives evidence of the original distribution of jitter values (blue line), which were drawn from a uniform distribution. The green line shows the distribution corresponding to the initial value of $\sigma_j = 8$ ms.

to a distribution that is uniform over the range from -8 to $+8$ ms. The algorithm still converges independent of the initial value for σ_j (Figure 5A), and the receptive field is accurately recovered (Figure 5B), although the assumed underlying Gaussian jitter distribution (Figure 5C, red line) differs from the true jitter distribution (blue line), which is nearly flat in the relevant range with fluctuations resulting from the finite number of simulated spikes.

To further examine the distribution of the jitter in this case, we can investigate how the weights $w_k(\tau)$ that are associated with the jitter values τ are distributed on average over all spikes k . The shape of this mean weight distribution (Figure 5C, black line) reveals differences of the underlying jitter distribution from the Gaussian model. In the earlier case where the jitter was drawn from a Gaussian distribution, the mean weights also displayed a Gaussian shape (data not shown), but in the case of the uniform distribution, the mean weights show systematic deviations. This could be used to develop a suited parametrization of the jitter distribution, which could subsequently be applied in the iterative algorithm. Here, however, we are not interested in the detailed shape of the jitter distribution, but rather in recovering the receptive field, for which the Gaussian parametrization appears to be sufficient.

In order to further test the consistency of the results obtained with the iterative algorithm, the unjittered simulated spike trains were also analyzed. As the spikes are drawn according to a Poisson process with a modulated rate, the spike trains are still variable, but since there is no jitter, this does not distort the estimation of the receptive field as obtained from the STA. In this case, even with an initial value of $\sigma_j = 12$ ms, the algorithm finds the correct $\sigma_j = 0$ solution within less than twenty iterations (data not shown). This confirms the consistency of the applied iterative procedure.

Receptive fields and jitter for locust auditory receptor neurons

We now apply the algorithm to electrophysiological data obtained from locust auditory receptor neurons. These were stimulated with an amplitude-modulated 5-kHz tone; new amplitude values were drawn from a Gaussian distribution in steps of $\Delta t = 0.2$ ms. The mean intensity was adjusted to elicit an intermediate firing rate for these neurons of around 100 Hz, and the standard deviation of the amplitude distribution in the experiments was either 5 or 10 dB. Example spike trains for repeated presentations of the same stimulus are shown in Figure 6A. For application of the algorithm, the amplitude modulation was used as the

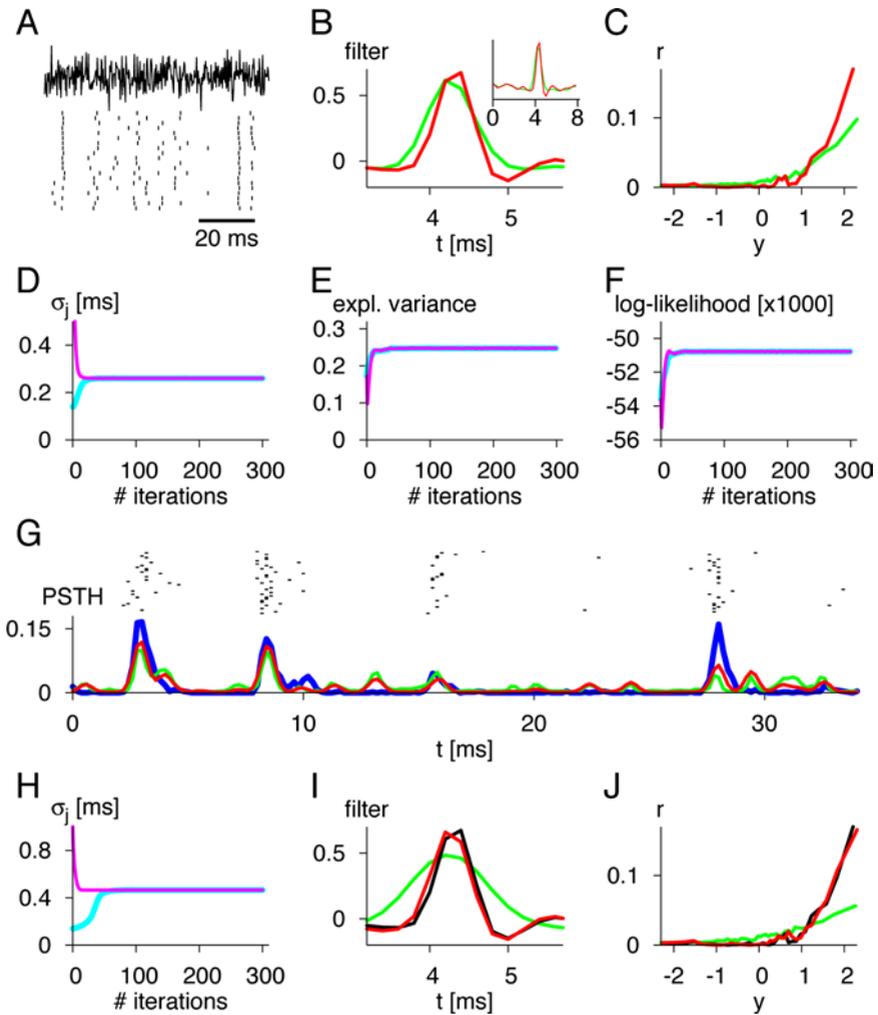


Figure 6. LNPJ model for locust auditory receptor cells. The analysis is based on 550 presentations of a 2-s stimulus resulting in slightly more than 56,000 spikes. (A) Amplitude modulation of sound wave and spike trains recorded from a single cell for several stimulus repeats in the locust auditory nerve. (B) Estimate of the receptive field. The receptive field is dominated by a single peak, which is narrower for the final estimate (red) as compared to the initial STA (green). Also, a slight negative component is revealed in the estimate of the filter. The full filter used in the algorithm, which extends over 8 ms, is shown in the inset. (C) Estimation of the nonlinearity. The initial nonlinearity (green) is considerably less steep than the final estimate (red) returned by the algorithm. (D) The standard deviation σ_j of the jitter distribution converges to the same value near 0.3 ms independent of the initial value. (E) Model-explained variance of the firing rate after each iteration for the two initial values of σ_j . (F) Model likelihood after each iteration. (G) Sample spikes (black dots) and corresponding PSTH (blue line) from the measured data and corresponding firing rates from the LNP model (green) and the final LNPJ model (red). (H) Evolution of the standard deviation σ_j of the jitter distribution in the case where each spike received an additional artificial jitter drawn from a Gaussian distribution with 0.4-ms standard deviation. (I) Estimation of receptive field in the case of additional artificial jitter. The STA estimate is distorted (green), whereas the recovered receptive field at the end of the algorithm (red) is nearly identical to the receptive field obtained without artificial jitter (black). (J) Estimate of the nonlinearity in the case of additional artificial jitter. Colors refer to the same cases as in (I).

stimulus, normalized to a mean of zero and a variance of one. An initial segment of 200 to 300 ms was discarded to avoid the sharp transient of spike frequency adaptation, and a final segment of 300 ms was ignored in the parameter estimation procedure and instead used for cross-validation.

We find again that the algorithm converges quickly and yields the same σ_j , receptive field, and nonlinearity independent of the initial value for σ_j . In the present example, the estimated jitter was around 0.3 ms (Figure 6D). The small jitter led to slight, but systematic changes between the original STA and the final estimate for the filter, which shows a more sharply peaked window of temporal integration (Figure 6B). In addition, a small negative component around 5 ms prior to the spike is now revealed in the filter. The half-peak width of the final estimate was 0.49 ms as compared to the original half-peak width of 0.66 ms for the STA. These findings were consistent for all four cells recorded under this experimental paradigm; in each case, the final linear filter was narrower and more sharply peaked than the original STA. On average, the half-peak width decreased from 0.76 ms (0.14 ms STD) for the STA to 0.59 ms (0.13 ms STD) for the final filter, and the corresponding final values for σ_j were on average 0.29 ms (0.05 ms STD).

As seen from the earlier simulations, the estimate of the neuron's nonlinearity can also be severely affected by spike jitter. For the present data we find that, by taking spike jitter into account, a much sharper nonlinearity is obtained with the iterative algorithm as compared to the STA-based estimate (Figure 6C); for large y , the slope of the final estimate for $g(y)$ is about twice as large as for the original estimate.

Cross-validation shows that the final LNPJ model indeed yields a better description of the data (Figure 6E and F). Nevertheless, only about a quarter of the firing-rate variance is explained by the model, and the comparison of the firing rates (Figure 6G) shows that the model does not fully capture the reliable and sparse firing of the neuron; while spike jitter appears to be an important factor in shaping the neuron's output, other known effects are here neglected, such as the LNLN-cascade-like stimulus integration (Gollisch & Herz 2005), refractoriness (Schaette et al. 2005), and different mechanisms of adaptation (Gollisch & Herz 2004).

Since we do not know the true underlying jitter distribution, we cannot assess the correctness of the estimation of the algorithm in the present case as easily as for the simulations. As a consistency check, however, we can add artificial jitter to all recorded spikes and test whether this leaves the final receptive fields invariant. As expected, the algorithm now returns a larger σ_j (Figure 6H), which reflects the combination of the original jitter in the recorded spike train and the artificially added jitter. In fact, the obtained value of $\sigma_j = 0.47$ ms closely matches what was expected from the earlier measured jitter value of 0.26 ms (Figure 6D) and the added artificial jitter with a standard deviation of 0.4 ms: $\sigma_j^{(expected)} = \sqrt{(0.26 \text{ ms})^2 + (0.4 \text{ ms})^2} \approx 0.48$ ms. Furthermore, the algorithm yields the same receptive field and nonlinearity after adding the artificial jitter as before (Figure 6I and J), underlining the consistency of the estimation procedure for the temporal receptive fields of the investigated receptor neurons.

Estimation of receptive fields with multiple linear filters

Calculating the STA always yields a single linear filter and is therefore not suited if multiple stimulus components interact to produce spikes, as, for example, in the energy model of complex cells in visual cortex (Adelson & Bergen 1985). As an alternative, STC analysis can extract the parameters of a multi-filter LNP cascade. To extend this analysis to include

the spike jitter of the LNPJ cascade, the iterative algorithm can be modified as explained above.

Figure 7 shows the performance of the multi-filter version of the algorithm for a simulation with two linear filters, one biphasic as in the previous analysis and the other with a Gaussian profile. Without spike jitter, the spectrum of eigenvalues would show two values that are distinct from the others. Due to the jitter, however, we see instead that the eigenvalues rather display a continuous distribution (Figure 7A, green points), and the corresponding estimates of the two filters are distorted compared to the true filters (Figure 7B and C). The algorithm again quickly converges and yields much more accurate filters. The final solution is independent of the initial value of σ_j , even though this parameter is again somewhat underestimated compared to the value of the simulation (Figure 7D). Cross-validation again

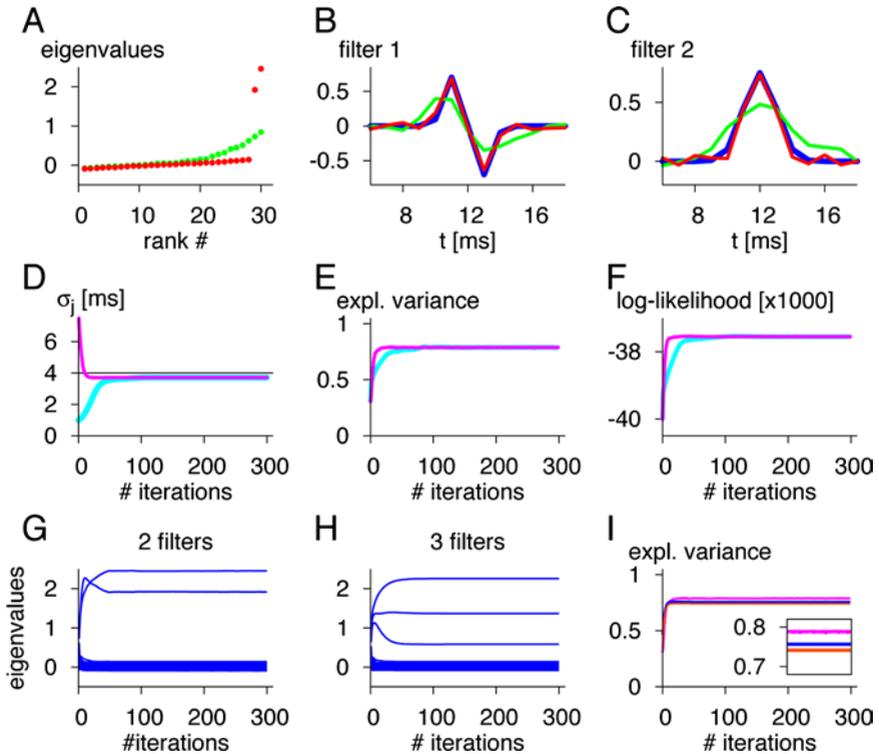


Figure 7. Estimation of multiple linear filters from simulated data. The analysis is based on 50 s simulated time, repeated ten times, and resulting in about 8000 spikes. (A) Spectrum of eigenvalues from the original STC analysis (green) and from the final iteration of the algorithm (red). (B, C) Estimation of the two filters of the receptive field. For both filters, the blue line shows the actual filter used in the simulation, which is compared to the projection of this filter onto the two most relevant eigenvectors as obtained from the standard STC analysis (green) and as obtained at the end of the algorithm (red). (D) Estimation of the standard deviation σ_j of the jitter distribution for two initial values. (E) Model-explained variance. (F) Model likelihood. (G) Evolution of eigenvalues when two filters are used in the iterative algorithm. (H) Evolution of eigenvalues when three filters are used in the iterative algorithm. The third eigenvalue decreases again after an initial increase, but still remains above the noise level given by the bulk of eigenvalues. (I) Comparison of model-explained variance of the firing rate for the two-filter model with the nonlinearity parametrized as a look-up table (magenta) or as a second-order Taylor expansion (orange) and for a three-filter model with a second-order Taylor expansion nonlinearity (blue). The inset presents an enlarged view of the values over the last 100 iterations.

shows the improvement in the obtained LNPJ model for predicting firing rate and spikes (Figure 7E and F).

Note that at the end of the simulation, two eigenvalues clearly stick out from the rest (Figure 7A, red points). These correspond to the two filters that describe the neuron's receptive field. When running the algorithm, however, the number of considered filters has to be specified beforehand, and in the present case, two filters were used. As a note of caution, it should be emphasized that the two distinct eigenvalues by themselves cannot be taken as evidence that two filters are needed and suffice to describe the data. For comparison, we can run the algorithm with three filters taken into account. The algorithm then leads to three raised eigenvalues (Figure 7H), although only two filters had been used in the simulation. However, the additional, third eigenvalue typically displays a substantial decrease (after an initial phase of increase), indicating that including this component may be superfluous. This is confirmed by investigating the fraction of explained variance in the firing rate (Figure 7I), which yields almost the same value for the three-filter analysis as for the two-filter analysis.

Test for multiple linear filters in the locust data

We now apply the analysis for multiple linear filters to the data from the locust auditory receptor neurons. A regular STC analysis yields two eigenvalues that are raised slightly above the noise level of the other eigenvalues, but the spectrum is nearly a continuum (Figure 8A, green points), from which it is difficult to determine whether the first two eigenvectors are indeed the relevant filters. These eigenvectors are dominated by peaks that are shifted with respect to each other (Figure 8B). The small dip near 4 ms prior to the spike in the second eigenvector is likely a reflection of the fact that the eigenvectors of the matrix, which is symmetric by construction, must be orthogonal to each other.

When we include the spike jitter in the analysis and iteratively estimate the relevant filters with a model that takes two filters into account, we find that only a single eigenvalue remains raised above the noise level (Figure 8A, red points). The corresponding eigenvector (Figure 8C, red line) is more sharply peaked than the components obtained earlier, and its peak lies in between those seen in Figure 8B. The second eigenvalue, on the other hand, has decreased to about the noise level after a short initial phase of increase (Figure 8D). The corresponding eigenvector shows no particular structure at the end of the analysis and is instead dominated by noise (Figure 8C, gray line). This indicates that a single-filter model is sufficient to describe the temporal receptive field of these neurons in response to amplitude-modulated sound with fixed carrier frequency. The two components obtained from the conventional STC analysis may be interpreted as reflecting an artificial division into components that correspond to "early" and "late" spikes, caused by spike jitter.

The nonlinearity of the model maps the two filter outputs y_1 and y_2 onto the spike probability r . For a simple visualization of this processing step, Figures 8E and F show one-dimensional representations of the nonlinearity separately for y_1 and y_2 . These are calculated as look-up tables analogous to the single-filter nonlinearity by using only y_1 or y_2 , respectively, as the input variable. While the nonlinearity becomes steep with respect to the first eigenvector (Figure 8F, red line), it is nearly flat for the second eigenvector, confirming that this eigenvector plays essentially no role for the neural output.

The standard deviation of the jitter distribution again converges to a value near 0.3 ms independent of the initial value (Figure 8G). The cross-validation measures show the

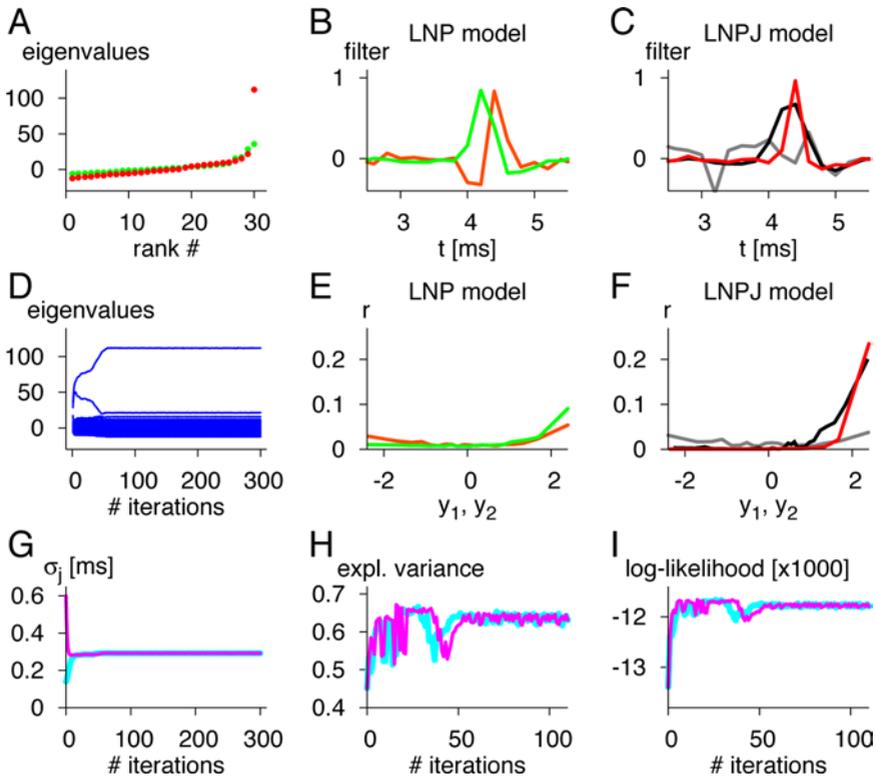


Figure 8. Estimation of multiple linear filters for the locust data. The analysis is based on the same data as in Figure 6. (A) Eigenvalue spectrum from the standard STC analysis (green) and from the final iteration of the algorithm (red). Only a single eigenvalue remains elevated, now clearly separated from the noise level. (B) Filters of the LNP model obtained as the two most relevant eigenvectors of the initial STC matrix. (C) Filters of the LNPJ model obtained as the two most relevant eigenvectors at the end of the algorithm. Only the eigenvector corresponding to the largest eigenvalue (red) shows a particular structure; the second eigenvector (gray) appears to consist of random fluctuations. The black line shows the filter that was obtained from the single-filter analysis (same as red line in Figure 6B). (D) Evolution of eigenvalues. The second eigenvalue quickly decreases after a short initial increase and is finally reduced to the noise level. (E, F) One-dimensional representations of the nonlinearity separately for the two filter outputs y_1 and y_2 , respectively, for the initial LNP model (E) and for the final LNPJ model (F). The black line in (F) shows the nonlinearity obtained from the single-filter analysis (same as red line in Figure 6C). (G) Evolution of the standard deviation σ_j of the jitter distribution. (H) Model-explained variance. (I) Model likelihood.

improvement in the obtained LNPJ model (Figure 8H and I). The model-explained variance of the firing rate is now substantially higher than for the single-filter LNPJ model obtained earlier by iteratively adjusting the STA. The difference results from the even narrower filter and steeper nonlinearity obtained here as compared to the earlier model, whose components are shown for comparison by the black lines in Figure 8C and F. We conclude that for these data the covariance-based analysis deals more successfully with the aforementioned deviations of this system from the simplified LNPJ structure. The narrow, sub-millisecond structure of the temporal receptive fields found in this analysis is in accordance with previous findings, based on stimulation with short clicks, that revealed temporal-integration processes over several hundred microseconds for these neurons (Gollisch & Herz 2005).

Extension of STA estimation to include systematic latency shifts

Finally, let us investigate how the presented analysis can be extended to include further effects on spike timing other than jitter. As an example, we use a systematic shift of the spike latency that depends on how strong the input to the neuron was. This is motivated by findings of systematic shifts in first-spike latency in cat auditory nerve fibers and A1 neurons in response to sound stimulation (Heil & Neubauer 2001, 2003); louder sounds yield shorter latencies. It may be expected that similar effects on spike timing occur also during continuous stimulation.

We here model such an effect by shifting each spike time by a certain amount, which depends on how strongly the neuron was activated. Within the applied model, stronger activation is set to come with shorter latencies, and the shift in spike timing is proportional to the output of the linear filter with proportionality constant α . As explained in an earlier section, the iterative algorithm is slightly extended to fit this model by taking such a shift into account when calculating the weights for the jitter values and by estimating α alongside with the standard deviation σ_j of the jitter distribution.

Figure 9 shows for simulated data that this extended algorithm converges nicely to the parameters that were used in the simulation. The cross-validation measures again confirm that the final model estimate yields a superior description of the data compared to the STA-based estimation. These results show that the algorithm is able to recover further effects on spike timing in addition to temporal jitter.

The same analysis was also performed on the locust data, but yielded no systematic effects on spike latency; α quickly converged to near zero, and the obtained receptive field and σ_j were the same as shown in Figure 6. Systematic latency shifts therefore do not appear to be

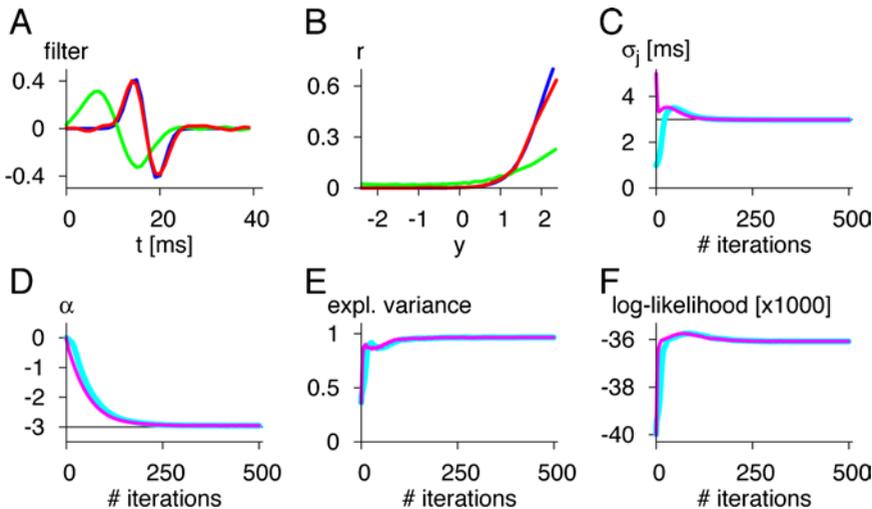


Figure 9. Estimation of the receptive field for a model with spike shift as well as spike jitter. The analysis is based on 50 s simulated time, repeated five times, and resulting in about 11,000 spikes. (A) The estimated filter at the end of the algorithm (red) recovers the shape of the true filter (blue) as well as its latency. The STA is shown in green. (B) True nonlinearity (blue) and estimated nonlinearity at the beginning of the algorithm (green) and at the end (red). (C) True nonlinearity (blue) and estimated nonlinearity at the beginning of the algorithm (green) and at the end (red). (C) The standard deviation σ_j of the jitter distribution converges to approximately the correct value of 3 ms (thin black line) as shown for two different initial values. (D) The proportionality constant α that relates the time shift of a spike to the output of the linear filter converges more slowly than σ_j , but also reaches approximately the correct value of -3 (thin black line). (E) Model-explained variance. (F) Model likelihood.

relevant for analyzing the receptive fields of the investigated neurons; in contrast, spike-time jitter does influence reverse-correlation estimates and should be accounted for in the analysis.

Discussion

Reverse correlation and spike-triggered analyses have been established as fundamental methods to measure neuronal receptive fields in a fast and reliable manner. Spike jitter, however, can strongly confound these analyses, in particular when the receptive fields contain features on similar (or smaller) time scales as the jitter. The presented iterative algorithm can improve estimates of the receptive field and output nonlinearity as well as yield a measure of the spike jitter in the data. Note that moderate numbers of spikes, large jitter values, and fairly variable responses (cf. spike trains in Figure 3C) have been used in the simulations to illustrate the applicability of the algorithm.

Modeling spike jitter

Commonly, spike jitter is only considered when spike-train recordings for several repeats of the same stimulus are available. The present approach allows us to define and work with spike jitter also in the case of only a single stimulus repeat. This increases the applicability of the algorithm and avoids the need to trade a thorough exploration of stimulus space for obtaining large numbers of repeats.

The spike-jitter distribution was modeled here as a Gaussian, which does not enforce causality of spiking; in the model, a large jitter value could lead to a spike that precedes the stimulus feature that elicited this spike. It may therefore be possible to exploit spike causality in the algorithm, for example by applying jitter distributions that are asymmetric or have a cut-off. In the present study, however, both the simulated and the experimental data had rather long latencies as apparent by the initial part of the filter with values near zero, leaving spike causality unaffected.

A central aspect of the applied model is an explicit distinction between precision (given by the temporal spike jitter) and reliability (determined by the Poisson spike-generation process). Variability in spike trains involves both lack of precision and lack of reliability, and it will generally not be apparent from the structure of the spike train alone to which degree these two aspects contribute. Variability resulting from the reliability of spike generation depends on the interaction between the stimulus and the neuron's stimulus-integration characteristics, whereas spike jitter contributes a stimulus-independent component to spike-train variability. Distinguishing between effects of precision and reliability is relevant for the analysis of receptive fields, as the variability resulting from the spike probability does not distort the estimates obtained from reverse-correlation analyses, but the spike jitter does. Moreover, a functional distinction between different types of variability in the spike train may ultimately help elucidate the mechanisms that shape the reliability of the neural code.

From a biophysical point of view, it is likely that spike-train variability results from a variety of processes that may contribute to both variability in spike generation and spike jitter. One such noise source is the stochastic opening and closing of ion channels in the neural membrane (Schneidman et al. 1998). This can lead to intricate effects on spike timing due to the interplay between cell-intrinsic conductances and the synaptic input (Schreiber et al. 2004). Stimulus-independent spike jitter can occur, for example, during the propagation of the action potential along the axon (Moradmand & Goldfinger 1995; Kurišćak et al. 2002). Also, the dynamics of the currents involved in spike generation have a strong influence on spike jitter during super-threshold activation by determining the susceptibility

of spike timing to input perturbations (Gutkin & Ermentrout 1998; Gutkin et al. 2005). The presented method is not aimed at distinguishing between these sources of jitter, but at estimating the undistorted receptive field independently of the biophysical origin of the jitter. Nevertheless, the resulting quantitative model may aid investigations of the origin and functional consequences of spike-timing variability in various experimental situations.

Extensions

As an example for extending the algorithm beyond spike-time jitter, we have seen that it is possible to extract also a systematic stimulus-dependent latency shift. The basic approach of iteratively adjusting the LNP cascade as a basic neuron model combined with different effects on spike timing may thus aid the investigation of various dynamics that shape neural spike trains. This may include a stimulus-dependency of the jitter distribution or effects of the spiking history on spike timing. Depending on the concrete model of history dependence, however, the latter might interfere with the assumed independence of spike jitter, thus complicating the calculation of the weights. Furthermore, history effects on spike timing should be combined with modeling the effect on the spike probability itself, as in the spike response model (Gerstner & Kistler 2002). The recent progress in fitting models with more realistic spike generation and refractory-period like dynamics (Miller & Mark 1992; Keat et al. 2001; Aguera y Arcas & Fairhall 2003; Pillow & Simoncelli 2003; Paninski et al. 2004; Pillow et al. 2005) may help to eventually combine these approaches.

The present analysis is based on Gaussian random stimuli, but several neural systems have been shown to be more effectively probed by stimuli with natural statistics. A future extension of the algorithm may thus explore the possibility to combine it with recent techniques that aim at applying reverse-correlation concepts to recordings made with natural stimuli (Theunissen et al. 2001; Paninski 2003; Sharpee et al. 2004; Touryan et al. 2005).

Limitations

In all investigated cases, we have seen that the algorithm converges reliably and fairly rapidly. There is, however, no mathematical proof of convergence and uniqueness of solution. In fact, if one were to start the algorithm with the unreasonable initial value $\sigma_j = 0$, no variation in spike timing would ever be explored by the algorithm, and the estimates would therefore stay fixed at their initial values. A useful check of convergence of the algorithm to a reasonable solution can be achieved by using different initial values for σ_j .

While the algorithm accurately recovers the correct receptive fields used in the simulations, the estimates of the underlying jitter distribution appear less reliable. In most cases, the final σ_j underestimates the value used in the simulations by a few per cent. This downward bias makes σ_j a conservative estimate of the amount of jitter in the data. The estimates of the other model parameters, on the other hand, appear to be very robust and do not depend on an accurate model of the true jitter distribution.

The algorithm relies on the fact that the investigated neuron's response characteristics can be at least approximated by an LNP cascade. For systems where additional processing stages are essential, application of the algorithm may be limited. For a detailed model of the auditory periphery, for example, the mechanical frequency filtering and the low-pass filter of the receptor-cell membrane suggest the use of two linear filter stages with a potential intermediate nonlinearity (Palmer & Russell 1986; Gollisch et al. 2002). The stimulus used

in this study aimed at minimizing this additional complication by using a single fixed carrier frequency and only varying the amplitude modulation.

Alternative approaches

The presented algorithm has a generic nature and makes few assumptions about the characteristics of the neural response. Depending on the data at hand, however, other approaches to handling spike jitter in the estimation of receptive fields also seem feasible and could be used for a comparative investigation. In some systems, discrete spike events can be distinguished, where isolated spikes occur reliably within a short temporal interval in nearly all trials for repeated stimulus presentations. If such events can be identified, the variance of spike timing in these events can be used to define a measure of spike jitter (Berry et al. 1997; Liu et al. 2001; Lestienne 2001) that could be directly compared to the estimate of σ_j returned by the iterative algorithm. It may then furthermore be feasible to measure the jitter distribution directly and thus deconvolve the STA or use the mean spike time of each event as the triggering time for a reverse-correlation analysis.

Recently, methods for dealing with spike jitter similar to the one presented here have been proposed by Chang et al. (2005) and Aldworth et al. (2005) (see also Dimitrov & Gedeon 2006). These algorithms aim at dejittering recorded spike trains by properly shifting the stimulus segments preceding each spike, based on a chosen measure of how well a stimulus segment matches the filter. In contrast to this shifting method, the approach presented here is based on weighting all stimulus segments in the vicinity of a spike and uses the predictions of the LNPJ model itself as a natural measure for the relevance of each of these segments. Together with the convenient mathematical relation to the EM algorithm as presented in the Appendix, this yields a robust and generic methodology that is also easily applied to the more complex cases of multiple linear filters or systematic shifts in spike latency as seen in this work.

For the considered single-filter neuron model, the measured STA is the convolution of the true filter and the spike-jitter distribution. As neither of the latter two are known in the estimation, the task is reminiscent of the “blind deconvolution” problem in the signal-processing literature (Haykin 1994). In the standard scenario for blind deconvolution, an observed signal results from the convolution of an unknown original signal with an unknown filter, and the challenge lies in the reconstruction of both the original signal and the filter under certain assumptions about the statistical properties of the signal source. In a similar way, we here attempt to deconvolve the STA and recover the underlying filter and jitter distribution, which are both initially unknown. Our approach differs from the standard concept of blind deconvolution, however, in that we do not make assumptions about the statistical characteristics of the filter, but rather rely on details of the observed stimulus–response relation.

Conclusion

Spike jitter is likely to contribute to the variability observed in recorded spike trains. Including it in the models that are used to analyze data has two advantages: first, it may lead to a more accurate characterization of how a neuron encodes and processes sensory information, as described, for example, by its receptive field; second, the spike jitter itself contains information about noise sources and other processes that influence the neuron’s operating characteristics. Spike jitter is likely to interfere with receptive-field estimation primarily in systems where temporal sensitivity and accuracy are important. In the sensory periphery, this includes mechanosensory and auditory systems where rapid processing of stimuli is a

primary issue. On slightly longer time scales, however, neurons in many cortical areas are likely to experience similar effects on spike timing resulting from intracortical network interactions (Shadlen & Newsome 1998). The generic nature of the algorithm introduced here for analyzing a neuron's input–output relation should provide for a wide range of possible applications of the presented methodology.

Acknowledgments

Most valuable comments on an earlier version of the manuscript by Andreas Herz, Markus Meister, Xaq Pitkow, and Inés Samengo are gratefully acknowledged. The experimental data had been obtained while the author was in the lab of Andreas Herz at the Institute for Theoretical Biology, Humboldt University Berlin. This work was supported through a long-term fellowship of the Human Frontier Science Program.

References

- Adelson EH, Bergen JR. 1985. Spatiotemporal energy models for the perception of motion. *J Opt Soc Am A* 2:284–299.
- Aldworth ZN, Miller JP, Gedeon T, Cummins GI, Dimitrov AG. 2005. Dejittered spike-conditioned stimulus waveforms yield improved estimates of neuronal feature selectivity and spike-timing precision of sensory interneurons. *J Neurosci* 25:5323–5332.
- Aguera y Arcas B, Fairhall AL. 2003. What causes a neuron to spike? *Neural Comput* 15:1789–1807.
- Berry MJ, Warland DK, Meister M. 1997. The structure and precision of retinal spike trains. *Proc Natl Acad Sci USA* 94:5411–5416.
- de Boer E, de Jongh HR. 1978. On cochlear encoding: Potentialities and limitations of the reverse-correlation technique. *J Acoust Soc Am* 63:115–135.
- de Boer E, Kuyper P. 1968. Triggered correlation. *IEEE Trans Biomed Eng* 15:169–179.
- Chang TR, Chung PC, Chiu TW, Poon PW. 2005. A new method for adjusting neural response jitter in the STRF obtained by spike-trigger averaging. *Biosystems* 79:213–222.
- Chichilnisky EJ. 2001. A simple white noise analysis of neuronal light responses. *Network* 12:199–213.
- Dayan P, Abbott LF. 2001. *Theoretical neuroscience: Computation and mathematical modeling of neural systems*. Cambridge, MA, MIT Press.
- deCharms RC, Blake DT, Merzenich MM. 1998. Optimizing sound features for cortical neurons. *Science* 280:1439–1443.
- Dempster A, Laird N, Rubin D. 1977. Maximum likelihood from incomplete data via the EM algorithm. *J Royal Stat Soc* 39:1–38.
- Dimitrov AG, Gedeon T. 2006. Effects of stimulus transformations on estimates of sensory neuron selectivity. *J Comput Neurosci*, in press.
- Eggermont JJ, Aertsen AM, Johannesma PI. 1983. Prediction of the responses of auditory neurons in the midbrain of the grass frog based on the spectro-temporal receptive field. *Hear Res* 10:191–202.
- Fritz J, Shamma S, Elhilali M, Klein D. 2003. Rapid task-related plasticity of spectrotemporal receptive fields in primary auditory cortex. *Nat Neurosci* 6:1216–1223.
- Gerstner W, Kistler WM. 2002. *Spiking neuron models*. Cambridge, UK, Cambridge University Press.
- Gollisch T, Herz AV. 2004. Input-driven components of spike-frequency adaptation can be unmasked in vivo. *J Neurosci* 24:7435–7444.
- Gollisch T, Herz AV. 2005. Disentangling sub-millisecond processes within an auditory transduction chain. *PLoS Biol* 3:e8.
- Gollisch T, Schütze H, Benda J, Herz AV. 2002. Energy integration describes sound-intensity coding in an insect auditory system. *J Neurosci* 22:10434–10448.
- Gutkin BS, Ermentrout GB. 1998. Dynamics of membrane excitability determine interspike interval variability: A link between spike generation mechanisms and cortical spike train statistics. *Neural Comput* 10:1047–1065.
- Gutkin BS, Ermentrout GB, Reyes A. 2005. Phase response curves determine the responses of neurons to transient inputs. *J Neurophysiol* 94:1623–1635.
- Haykin S (editor). 1994. *Blind deconvolution*. Englewood Cliffs, NJ, Prentice-Hall.
- Heil P, Neubauer H. 2001. Temporal integration of sound pressure determines thresholds of auditory-nerve fibers. *J Neurosci* 21:7404–7415.

- Heil P, Neubauer H. 2003. A unifying basis of auditory thresholds based on temporal summation. *Proc Natl Acad Sci USA* 100:6151–6156.
- Hubel DH, Wiesel TN. 1962. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J Physiol* 160:106–154.
- Hunter IW, Korenberg MJ. 1986. The identification of nonlinear biological systems: Wiener and Hammerstein cascade models. *Biol Cybern* 55:135–144.
- Keat J, Reinagel P, Reid RC, Meister M. 2001. Predicting every spike: A model for the responses of visual neurons. *Neuron* 30:803–817.
- Kim PJ, Young ED. 1994. Comparative analysis of spectro-temporal receptive fields, reverse correlation functions, and frequency tuning curves of auditory-nerve fibers. *J Acoust Soc Am* 95:410–422.
- Korenberg MJ, Hunter IW. 1986. The identification of nonlinear biological systems: LNL cascade models. *Biol Cybern* 55:125–134.
- Kuffler SW. 1953. Discharge patterns and functional organization of mammalian retina. *J Neurophysiol* 16:37–68.
- Kurišćák E, Trojan S, Wunsch Z. 2002. Model of spike propagation reliability along the myelinated axon corrupted by axonal intrinsic noise sources. *Physiol Res* 51:205–215.
- Lestienne R. 2001. Spike timing, synchronization and information processing on the sensory side of the central nervous system. *Prog Neurobiol* 65:545–591.
- Lewis ER, van Dijk P. 2004. New variations on the derivation of spectro-temporal receptive fields for primary auditory afferent axons. *Hear Res* 189:120–136.
- Linden JF, Liu RC, Sahani M, Schreiner CE, Merzenich MM. 2003. Spectrotemporal structure of receptive fields in areas A1 and AAF of mouse auditory cortex. *J Neurophysiol* 90:2660–2675.
- Liu RC, Tzovec S, Rebrik S, Miller KD. 2001. Variability and information in a neural code of the cat lateral geniculate nucleus. *J Neurophysiol* 86:2789–2806.
- Machens CK, Wehr MS, Zador AM. 2004. Linearity of cortical receptive fields measured with natural sounds. *J Neurosci* 24:1089–1100.
- Meister M, Berry MJ. 1999. The neural code of the retina. *Neuron* 22:435–450.
- Miller MJ, Mark KE. 1992. A statistical study of cochlear nerve discharge patterns in response to complex speech stimuli. *J Acoust Soc Am* 92:202–209.
- Moradmand K, Goldfinger MD. 1995. Computation of long-distance propagation of impulses elicited by Poisson-process stimulation. *J Neurophysiol* 74:2415–2426.
- Nelken I, Kim PJ, Young ED. 1997. Linear and nonlinear spectral integration in type IV neurons of the dorsal cochlear nucleus. II. Predicting responses with the use of nonlinear models. *J Neurophysiol* 78:800–811.
- Palmer AR, Russell IJ. 1986. Phase-locking in the cochlear nerve of the guinea-pig and its relation to the receptor potential of inner hair-cells. *Hear Res* 24:1–15.
- Paninski L. 2003. Convergence properties of three spike-triggered analysis techniques. *Network* 14:437–464.
- Paninski L. 2004. Maximum likelihood estimation of cascade point-process neural encoding models. *Network* 15:243–262.
- Paninski L, Pillow JW, Simoncelli EP. 2004. Maximum likelihood estimation of a stochastic integrate-and-fire neural encoding model. *Neural Comput* 16:2533–2561.
- Pillow JW, Paninski L, Uzzell VJ, Simoncelli EP, Chichilnisky EJ. 2005. Prediction and decoding of retinal ganglion cell responses with a probabilistic spiking model. *J Neurosci* 25:11003–11013.
- Pillow JW, Simoncelli EP. 2003. Biases in white noise analysis due to non-Poisson spike generation. *Neurocomputing* 52:109–115.
- Reich DS, Mechler F, Purpura KP, Victor JD. 2000. Interspike intervals, receptive fields, and information encoding in primary visual cortex. *J Neurosci* 20:1964–1974.
- Reinagel P, Reid RC. 2000. Temporal coding of visual information in the thalamus. *J Neurosci* 20:5392–5400.
- Sahani M, Linden JF. 2003. Evidence optimization techniques for estimating stimulus-response functions. *Adv Neural Information Proc Systems* 15:301–308.
- Schaette R, Gollisch T, Herz AV. 2005. Spike-train variability of auditory neurons in vivo: Dynamic responses follow predictions from constant stimuli. *J Neurophysiol* 93:3270–3281.
- Schneidman E, Freedman B, Segev I. 1998. Ion channel stochasticity may be critical in determining the reliability and precision of spike timing. *Neural Comput* 10:1679–1703.
- Schreiber S, Fellous J-M, Tiesinga P, Sejnowski TJ. 2004. Influence of ionic conductances on spike timing reliability of cortical neurons for suprathreshold rhythmic inputs. *J Neurophysiol* 91:194–205.
- Schwartz O, Chichilnisky EJ, Simoncelli EP. 2002. Characterizing neural gain control using spike triggered covariance. *Adv Neural Information Proc Systems* 14:269–276.

- Sen K, Theunissen FE, Doupe AJ. 2001. Feature analysis of natural sounds in the songbird auditory forebrain. *J Neurophysiol* 86:1445–1458.
- Shadlen MN, Newsome WT. 1998. The variable discharge of cortical neurons: Implications for connectivity, computation, and information coding. *J Neurosci* 18:3870–3896.
- Sharpee T, Rust NC, Bialek W. 2004. Analyzing neural responses to natural signals: Maximally informative dimensions. *Neural Comput* 16:223–250.
- Simoncelli EP, Pillow JW, Paninski L, Schwartz O. 2004. Characterization of neural responses with stochastic stimuli. In *The Cognitive Neurosciences* Gazzaniga MS, editor. MIT Press, Cambridge, MA, 3rd edition.
- de Ruyter van Steveninck R, Bialek W. 1988. Coding and information transfer in short spike sequences. *Proc Soc Lond B Biol Sci* 234:379–414.
- Theunissen FE, David SV, Singh NC, Hsu A, Vinje WE, Gallant JL. 2001. Estimating spatio-temporal receptive fields of auditory and visual neurons from their responses to natural stimuli. *Network* 12:289–316.
- Theunissen FE, Sen K, Doupe AJ. 2000. Spectral-temporal receptive fields of nonlinear auditory neurons obtained using natural sounds. *J Neurosci* 20:2315–2331.
- Touryan J, Felsen G, Dan Y. 2005. Spatial structure of complex cell receptive fields measured with natural images. *Neuron* 45:781–791.
- Touryan J, Lau B, Dan Y. 2002. Isolation of relevant visual features from random stimuli for cortical complex cells. *J Neurosci* 15:10811–10818.

Appendix

Relation to EM algorithm

The iterative algorithm for estimating the model parameters presented in the main text is related to the expectation–maximization (EM) algorithm, which is widely used to obtain maximum-likelihood solutions in an iterative fashion (Dempster et al. 1977). The starting point for applying the EM algorithm is typically the occurrence of hidden or unobserved data, which in the present case are the jitter values for each individual spike.

The general EM approach begins with the full log-likelihood of the model, i.e., the logarithm of the probability that the full data set, observed as well as hidden data, is produced by the model for a given set of model parameters. One then proceeds by alternating the two fundamental steps of the EM algorithm: in the E-step, the expectation value of the log-likelihood function given the current set of model parameters is calculated by integrating over the hidden data. In the M-step, the model parameters are updated by computing those model parameters that maximize the expectation value of the log-likelihood function.

In the present case, we largely stick to this prescribed procedure for the E-step, but substitute part of the complex maximization calculation with the easy-to-compute reverse-correlation technique for updating the model parameters. The integration over the hidden jitter values in the E-step results in the weights $w_k(\tau)$ for a jitter value τ of spike k .

To show this in more detail, we first write down the full likelihood function, which is defined as the probability of the data (observed spike times t_k as well as hidden jitter values τ_k) given the model parameters $f(n)$, $g(y)$, and $p_j(\tau)$:

$$\begin{aligned} L(f, g, p_j) &= p(\{t_k\}, \{\tau_k\} | f, g, p_j) \\ &= \prod_k g(s * f(t_k - \tau_k)) \cdot \prod_{t \neq t_k - \tau_k} [1 - g(s * f(t))] \cdot \prod_k p_j(\tau_k). \end{aligned} \quad (14)$$

Here, $s * f$ is defined as the convolution of the stimulus s and the filter f , $s * f(t) = \sum_n s(t - n) \cdot f(n)$. The first product in Equation 14 captures the probabilities that spikes were generated at the appropriate points in time, the second product gives the probabilities that other time points did not generate spikes, and the product of the $p_j(\tau_k)$ simply denotes the probability that the particular jitter values τ_k occurred. The log-likelihood of the parameters

is therefore given by

$$\begin{aligned}
 & \log L(f, g, p_j) \\
 &= \sum_k \log[g(s * f(t_k - \tau_k))] + \sum_{t \neq t_k - \tau_k} \log[1 - g(s * f(t))] + \sum_k \log[p_j(\tau_k)] \\
 &= \sum_k \log \frac{g(s * f(t_k - \tau_k))}{1 - g(s * f(t_k - \tau_k))} + \sum_t \log[1 - g(s * f(t))] + \sum_k \log[p_j(\tau_k)]. \quad (15)
 \end{aligned}$$

Ideally, we could now integrate out the unknown τ_k and then maximize the log-likelihood over all model parameters in a straightforward maximum-likelihood approach. The resulting integral, however, becomes highly complex, as the required marginal probabilities of the jitter values themselves depend on the model parameters. The advantage of the EM algorithm lies in the fact that these marginal probabilities are held fixed during the M-step, which makes the maximization at least somewhat simpler.

For the E-step, we have to calculate the expectation value $E[\log L(f, g, p_j)]$ by integrating over the hidden data τ_k . To do so, we must sum the log-likelihood over all sets of values of τ_k and weight each contribution with the probability $p(\{\tau_k\} | f, g, p_j)$ that this set $\{\tau_k\}$ occurs. As each individual jitter value τ_k depends almost exclusively on the stimulus in the vicinity of spike k itself and not on the times of other spikes, we can approximate this probability by the product of the marginal probabilities $p(\tau_k | f, g, p_j)$ for each individual spike k . The latter are nothing but the weights $w_k(\tau_k)$ from the main text. The use of the weights $w_k(\tau_k)$ can therefore be viewed as resulting from the prescription of the E-step in the EM algorithm. This analogy allows us to directly extend this procedure to other effects on spike timing such as the latency shifts discussed in the main text of this paper.

We thus compute the expectation value of the log-likelihood as

$$\begin{aligned}
 E[\log L(f, g, p_j)] &\approx \sum_{\tau_1, \tau_2, \dots} \left(\prod_k w_k(\tau_k) \right) \cdot \log L(f, g, p_j) \\
 &= \sum_k \sum_{\tau} w_k(\tau) \cdot \log \frac{g(s * f(t_k - \tau))}{1 - g(s * f(t_k - \tau))} + \sum_t \log[1 - g(s * f(t))] \\
 &\quad + \sum_k \sum_{\tau} w_k(\tau) \cdot \log p_j(\tau). \quad (16)
 \end{aligned}$$

We can interpret the above equation in the following way: instead of the original spike times t_k , we now include all times t between $t_k - \tau_{\max}$ and $t_k + \tau_{\max}$, but weight each of these “new spike times” by $w_k(\tau)$ with $\tau = t_k - t$.

In the M-step, the model parameters f , g , and p_j should be updated by maximizing this expectation value. The new parameters for p_j can be calculated independently by maximizing the last term in Equation 16. For the Gaussian parametrization of $p_j(\tau)$, this yields an updating rule that simply amounts to calculating the weighted average of τ^2 over all spikes and jitter values, Equation 9. The maximization with respect to f and g is still a computationally complex problem, but with the “weighted-spike-times” interpretation that we have gained from the likelihood picture, we now return to the simpler STA estimate and approximate this part of the M-step by calculating the spike-triggered average given the new, weighted spike times. In general, the STA and the maximum-likelihood estimates of the parameters differ, although they bear some resemblance, and interesting connections can be drawn between them (Paninski 2004). For this reason, the algorithm presented in the main text is not a true maximum-likelihood estimator, but comes close to it in spirit.