Quantifying Statistical Interdependence by Message Passing on Graphs-Part I: One-Dimensional Point Processes

Quantifying Statistical Interdependence by Message Passing on Graphs—Part I: One-Dimensional Point Processes

J. Dauwels
jdauwels@mit.edu
Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139, U.S.A., and Amari Research Unit, RIKEN Brain Science Institute, Saitama 351-0198, Japan

F. Vialatte
fvialatte@brain.riken.jp
Laboratory for Advanced Brain Signal Processing, RIKEN Brain Science Institute, Saitama 351-0198, Japan

T. Weber
theo_w@mit.edu
Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA 02139, U.S.A.

A. Cichocki
cia@brain.riken.jp
Laboratory for Advanced Brain Signal Processing, RIKEN Brain Science Institute, Saitama 351-0198, Japan

We present a novel approach to quantify the statistical interdependence of two time series, referred to as stochastic event synchrony (SES). The first step is to extract "events" from the two given time series. The next step is to try to align events from one time series with events from the other. The better the alignment, the more similar the two time series are considered to be. More precisely, the similarity is quantified by the following parameters: time delay, variance of the timing jitter, fraction of noncoincident events, and average similarity of the aligned events.

The pairwise alignment and SES parameters are determined by statistical inference. In particular, the SES parameters are computed by maximum a posteriori (MAP) estimation, and the pairwise alignment is obtained by applying the max-product algorithm. This letter deals with one-dimensional point processes; the extension to multidimensional point processes is considered in a companion letter in this issue.

By analyzing surrogate data, we demonstrate that SES is able to quantify both timing precision and event reliability more robustly than

Neural Computation **21, 2152–2202** (2009)

classical measures can. As an illustration, neuronal spike data generated by the Morris-Lecar neuron model are considered.

1 Introduction

Quantifying the interdependence between time series is an important and challenging problem. Although it is straightforward to quantify linear dependencies, the extension to nonlinear correlations is far from trivial. A variety of approaches have been proposed, stemming from research fields as diverse as physics, statistics, signal processing, and information theory (see, e.g., Stam, 2005; Quian Quiroga, Kraskov, Kreuz, & Grassberger, 2002; Pereda, Quiroga, & Bhattacharya, 2005; Kreuz, Haas, Morelli, Abarbanel, & Politi, 2007; Tiesinga, Fellous, & Sejnowski, 2008).

In this letter we propose a novel measure to quantify the interdependence between two point processes, referred to as stochastic event synchrony (SES). It consists of the following parameters: time delay, variance of the timing jitter, fraction of "noncoincident" events, and average similarity of the events. SES captures two different aspects of synchrony: timing precision and reliability. Those concepts can be understood from the following analogy: when you wait for a train in the station, the train may come at the station, or it may not come at all (e.g., it may be out of service due to some mechanical problem). If the train comes, it may or may not be on time. The former uncertainty is related to reliability, whereas the latter is related to precision. A similar distinction between timing precision and reliability has been made in Mainen and Sejnowski (1995) and Tiesinga et al. (2008). SES quantifies precision and reliability by the variance of the timing jitter and the fraction of the noncoincident events, respectively.

The pairwise alignment of point processes is cast as a statistical inference problem, which is solved by applying the max-product algorithm on a graphical model (Jordan, 1999; Loeliger, 2004; Loeliger et al., 2007). In the case of one-dimensional point processes, the graphical model is cycle free. The max-product algorithm is then equivalent to dynamic programming and is guaranteed to find the optimal alignment. For multidimensional point processes, the max-product algorithm is applied on a cyclic graphical model; this algorithm yields the optimal alignment as long as the optimal alignment is unique. This letter deals with one-dimensional point processes; the companion letter in this issue considers the extension to multidimensional point processes.

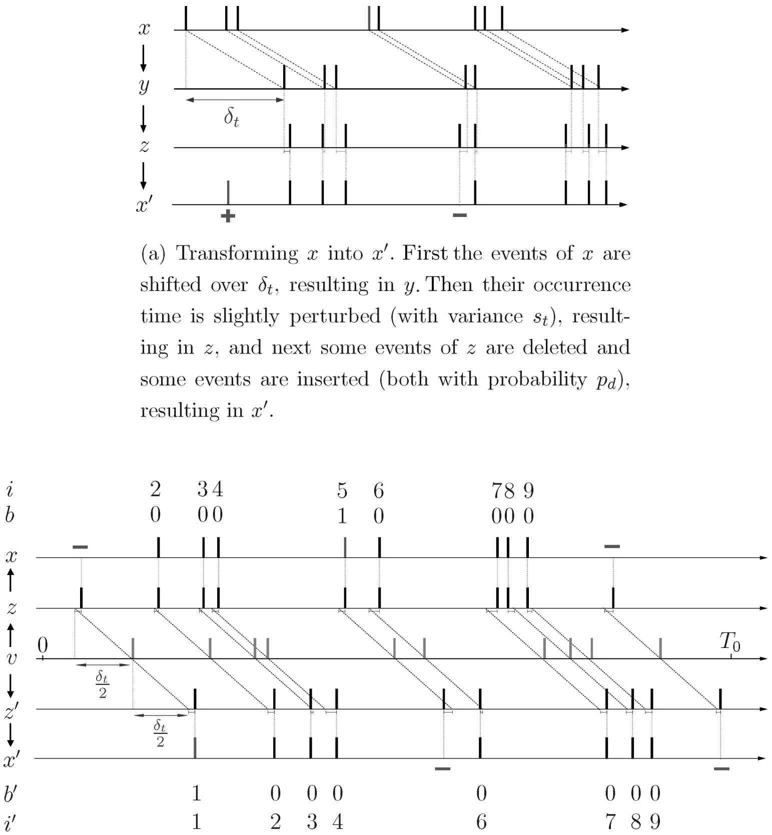
Although the method may be applied to any kind of time series (e.g., finance, oceanography, and seismology), in this and the companion letter, we solely consider time series that occur in the context of neuroscience. Synchrony is indeed an important topic in neuroscience. For instance, it is hotly debated whether the synchronous firing of neurons plays a role in cognition (Varela, Lachaux, Rodriguez, & Martinerie, 2001) and even in consciousness (Singer, 2001; Crick & Koch, 2003). The synchronous firing paradigm has also attracted substantial attention in both the experimental (e.g., Abeles, Bergman, Margalit, & Vaadia, 1993) and the theoretical neuroscience literature (e.g., von der Malsburg, 1981; Amari, Nakahara, Wu, & Sakai, 2003). Moreover, medical studies have reported that many neurophysiological diseases such as Alzheimer's disease are often associated with abnormalities in neural synchrony (Matsuda, 2001; Jeong, 2004). Therefore, the proposed method may be helpful to diagnose such mental disorders. The companion letter presents promising results on the early prediction of Alzheimer's disease based on electroencephalograms (EEG).

This letter considers the interdependence between two point processes. The proposed methods, however, can be extended to a collection of point processes. This extension is nontrivial: aligning a collection of point processes involves a significantly more complex combinatorial optimization problem; the optimal alignment becomes in general intractable. Therefore, one needs to resort to approximate inference methods. Those issues go beyond the scope of this letter and the companion letter and will be addressed in a future report.

In the next section, we introduce SES for the case of one-dimensional point processes. In section 3, we describe the underlying statistical model and explain in section 4 how one can perform inference in that model. In section 5, we review several classical (dis)similarity measures for one-dimensional point processes, since they will serve as benchmark for SES; more precisely, we will consider the Victor-Purpura distance metric (Victor & Purpura, 1997; Aronov, 2003; Victor, Goldberg, & Gardner, 2007), the van Rossum distance metric (van Rossum, 2001), the Schreiber et al. similarity measure (Schreiber, Fellous, Whitmer, Tiesinga, & Sejnowski, 2003), the Hunter-Milton similarity measure (Hunter & Milton, 2003), and the event synchronization measure proposed in Quian Quiroga, Kreuz, and Grassberger (2002). In section 6 we investigate the robustness and reliability of those classical (dis)similarity measures and SES by means of surrogate data. In section 7 we consider an application related to neuroscience: we quantify the firing reliability of Morris-Lecar type I and type II neurons using classical methods and SES. We offer some concluding remarks in section 8.

2 Principle

Let us consider the one-dimensional point processes ("event strings") x and x' in Figure 1a. Ignore y and z for now; they could be point processes in time, for example, ($x_1 = 1.3$ s, $x_2 = 5.8$ s, ...) or space, for example, ($x_1 = 1.3$ m, $x_2 = 5.8$ m, ...), or any other dimension. We wish to quantify to what extent x and x' are synchronized. Intuitively, two event strings can be considered as synchronous (or "locked") if they are identical apart from a time shift δ_t, small deviations in the event occurrence times ("event timing jitter"), or a



(a) Transforming x into x'. First the events of x are shifted over δ_t, resulting in y. Then their occurrence time is slightly perturbed (with variance s_t), resulting in z, and next some events of z are deleted and some events are inserted (both with probability p_d), resulting in x'.

(b) Symmetric procedure to generate x and x'. One first generates a process v and then makes two identical copies of v and shifts those over $-\delta_t/2$ and $\delta_t/2$, respectively. The events of the resulting point process are slightly shifted (with variance $s_t/2$), and some of those events are deleted (with probability p_d), resulting in x and x'.

Figure 1: One-dimensional stochastic event synchrony. An asymmetric (top) and symmetric (bottom) procedure relating x to x'.

few event insertions or deletions. More precisely, for two event strings to be synchronous, the event timing jitter should be significantly smaller than the average interevent time, and the number of deletions and insertions should comprise only a small fraction of the total number of events. This intuitive concept of synchrony is illustrated in Figure 1a. The event string x' is obtained from event string x by successively shifting x over δ_t (resulting in y), slightly perturbing the event occurrence times (resulting in z), and, eventually, by adding (plus sign) and deleting (minus sign) events, resulting in x'. Adding and deleting events in z leads to "noncoincident" events in x and x' (see Figure 1a): a noncoincident event in x is an event that cannot be paired with an event in x', and vice versa.

This intuitive reasoning leads to a novel measure for synchrony between two event strings, that is, stochastic event synchrony (SES). For the one-dimensional case, it is defined as the triplet (δ_t, s_t, ρ), where s_t is the variance of the (event) timing jitter and ρ is the percentage of noncoincident events,

$$\rho \stackrel{\triangle}{=} \frac{n_{\text{non-co}} + n'_{\text{non-co}}}{n + n'}, \tag{2.1}$$

with n and n' the total number of events in x and x', respectively, and $n_{\text{non-co}}$ and $n'_{\text{non-co}}$ the total number of noncoincident events in x and x', respectively. We will denote the standard deviation of the (event) timing jitter by σ_t, and hence $s_t = \sigma_t^2$. SES is related to the metrics ("distances" or "kernels") proposed in Victor and Purpura (1997), Aronov (2003), Victor et al. (2007), Shpigelman, Singer, Paz, and Vaadia (2005), Eichhorn et al. (2004), and Schrauwen and Van Campenhout (2007), which are single numbers that quantify the similarity of event strings. In contrast, we characterize synchrony by means of three parameters, which allows us to distinguish two fundamentally different types of synchrony, as we will demonstrate in section 7 (see Figure 13). Moreover, our approach is rooted in statistical inference, in contrast to the metrics of Victor and Purpura (1997), Aronov (2003), Victor et al. (2007), Shpigelman et al. (2005), Eichhorn et al. (2004), and Schrauwen et al. (2007), which are derived from either optimization theory (Victor & Purpura, 1997; Aronov, 2003; Victor et al., 2007) or in the context of kernel machines (Shpigelman et al., 2005; Eichhorn et al., 2004; Schrauwen et al., 2007).

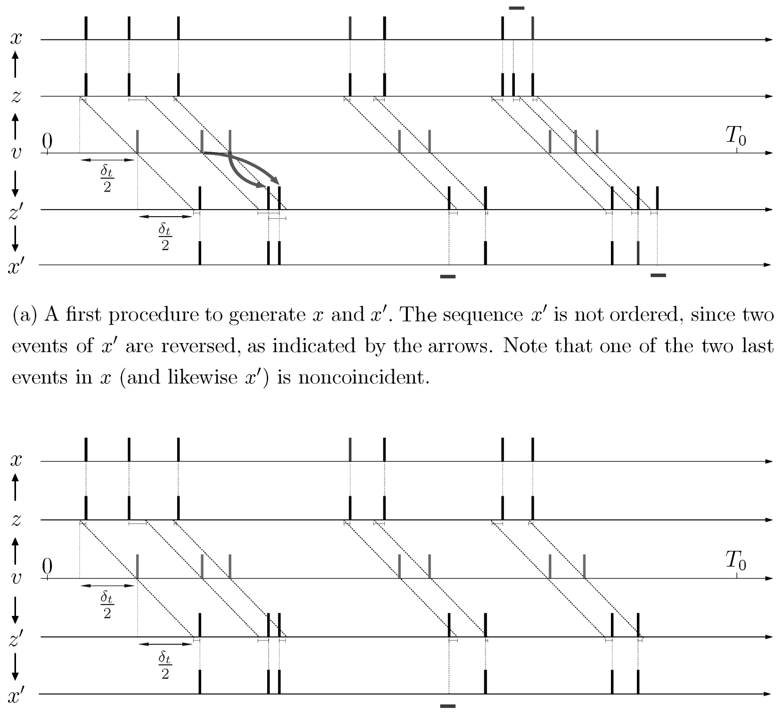
3 Statistical Model

We compute the SES parameters by performing inference in a generative probabilistic model for the sequences x and x'. In order to describe that model, we consider a symmetric procedure to generate x and x', depicted in Figure 1b. Note that the procedure of Figure 1a corresponds to a conditional distribution $p(x'|x; \delta_t, s_t)$, which is asymmetric in x and x'. First, one generates an event string v of length ℓ, where the events v_k are mutually independent and uniformly distributed in $[0, T_0]$. The strings z and z' are generated by delaying v over $-\delta_t/2$ and $\delta_t/2$, respectively, and by (slightly) perturbing the resulting event occurrence times. We will model those perturbations as zero-mean gaussian random variables with variance $s_t/2$. Next, some of the events in z and z' are removed, resulting in the sequences x and x'; each event of z and z' is removed with probability

p_d ("deletion"), independent of the other events. We denote by z_{r_k} and z'_{r_k} the events in z and z', respectively, that correspond to v_k. In the example of Figure 1b, $r = (1, 2, \ldots, 10) = r'$. Occasionally, a pair of events (z_{r_k}, z'_{r_k}) is removed (with probability p_d^2), referred to as "event-pair deletion," but more often, either z_{r_k} or z'_{r_k} is removed (a single-event deletion). If none of the events (z_{r_k}, z'_{r_k}) is removed, there is an event in x and in x' that corresponds to v_k; we denote this event pair by (x_{j_k}, x'_{j_k}). In the example of Figure 1b, $j = (1, 2, 3, 5, 6, 7, 8)$ and $j' = (2, 3, 4, 5, 6, 7, 8)$. Note that if z_{r_k} is deleted but not z'_{r_k}, the corresponding event in x' becomes a noncoincident event, and vice versa. In the example of Figure 1b, events z_1 and z'_5 are deleted (single-event deletions), and as a result, x'_1 and x_5 are noncoincident events; also the pair (z_{10}, z'_{10}) is removed (an event-pair deletion). It is easily verified that the expected length of the sequences x and x' is $(1 - p_d)\ell$.

It is noteworthy that this procedure of generating the pair of point processes x and x' may easily be extended to a collection of point processes. However, inference in the resulting probabilistic model is tractable only for pairs of point processes. If one considers more than two point processes, one needs to resort to approximate inference techniques. Such methods will be presented in a future report.

From now on, we will assume that the event pairs $(x_{j_k}, x'_{j'_k})$ are ordered: $(x_{j_k}, x'_{j'_k})$ occurs after the pair $(x_{j_{k-1}}, x'_{j'_{k-1}})$, or more precisely, $x_{j_k} \geq x_{j_{k-1}}$ and $x'_{j'_k} \geq x'_{j'_{k-1}}$ (for all k). This assumption is reasonable, since without it, there would be an unwieldy number of possible ways to generate the same point processes x and x', and therefore, the problem of inferring the SES parameters would be ill posed. In fact, virtually all measures of event synchrony make use of this assumption explicitly or implicitly (see section 5 for a brief review). However, this assumption has some important consequences, as illustrated in Figure 2. If s_t is large, there is a high probability that events in x and x' will not be ordered in time (see Figure 2a). Ignoring this fact will result in estimates of s_t that are smaller than the true value s_t. Obviously this issue concerns not only SES but event synchrony in general. In addition, some event deletions may be ignored: in Figure 2a one of the last two events of x (and likewise x') is noncoincident; however, in the procedure of Figure 2b, they are both coincident. The latter generative procedure is simpler in the sense that it involves fewer deletions and the perturbations are slightly smaller. As a result, the parameter ρ (and hence also p_d) is generally underestimated. Again, this problem concerns not only SES but any measure that quantifies how reliably events occur (see section 6). Both issues may be resolved to some extent by incorporating additional information. For example, in the case of spike trains, one may incorporate information about the spike shape; each spike is then described by its occurrence time and some additional parameters (e.g., shape parameters such as height and width). SES can be extended to incorporate such additional information, as



(a) A first procedure to generate x and x'. The sequence x' is not ordered, since two events of x' are reversed, as indicated by the arrows. Note that one of the two last events in x (and likewise x') is noncoincident.

(b) A second procedure to generate the same point processes x and x' without order reversal. The two last events of x and x' are now considered to be coincident.

Figure 2: Inherent ambiguity in event synchrony: two equivalent procedures to generate the point processes x and x'.

we describe in the companion letter in this issue. When matching events of x and x', we then no longer assume that those events are ordered in time; we allow reversals as in Figure 2a.

In the following, we discuss the statistical model that corresponds to the above symmetric procedure of generating the pair of point processes x and x'. For clarity, we list in Table 1 the most relevant variables and parameters associated with that model. We now clarify each of those variables and parameters. The statistical model takes the form

$$\begin{aligned}p(x, x', b, b', v, \delta_t, s_t, \ell) &= p(x \mid b, v, \delta_t, s_t)p(x' \mid b', v, \delta_t, s_t)p(b, b' \mid \ell) \\ &\quad \cdot p(v|\ell)p(\ell)p(\delta_t)p(s_t), \end{aligned} \tag{3.1}$$

where b and b' are binary strings that indicate whether the events in x and x' are coincident. More specifically, $b_k = 1$ if x_k is noncoincident, $b_k = 0$

Table 1: Variables and Parameters Associated with Model $p(x, x', b, b', v, \delta_t, s_t, \ell)$, Equation 3.1.

Symbol	Explanation
x and x'	The two given point processes
v	Hidden sequence from which the observed sequences x and x' are generated
z and z'	Point processes obtained by shifting v over $\delta_t/2$ and $-\delta_t/2$, resp., and perturbing the timing of the resulting sequences (variance $s_t/2$)
b and b'	Binary sequences that indicate whether events in x and x', resp., are coincident
i and i'	Indices of the events in v that generated x and x', resp.
j and j'	Indices of the coincident events in x and x', resp.
n and n'	Length of x and x', resp.
n_{del} and n'_{del}	Number of deletions in z and z', resp.
$n^{\text{tot}}_{\text{del}}$	Total number of deletions in z and z'
$n_{\text{del,single}}$ and $n'_{\text{del,single}}$	Number of single-event deletions in z and z', resp.
$n_{\text{del,pair}}$ and $n'_{\text{del,pair}}$	Number of event-pair deletions in z and z', resp.
$n_{\text{non-co}}$ and $n'_{\text{non-co}}$	Number of noncoincident events in x and x', resp.
$n^{\text{tot}}_{\text{non-co}}$	Total number of noncoincident events in x and x'
ℓ	Length of v
δ_t	Timing offset between x and x'
s_t	Timing jitter between x and x'

otherwise, and likewise for b'_k. For mathematical convenience, we choose a geometric prior for the length ℓ:

$$p(\ell) = (1 - \lambda T_0)(\lambda T_0)^{\ell}, \tag{3.2}$$

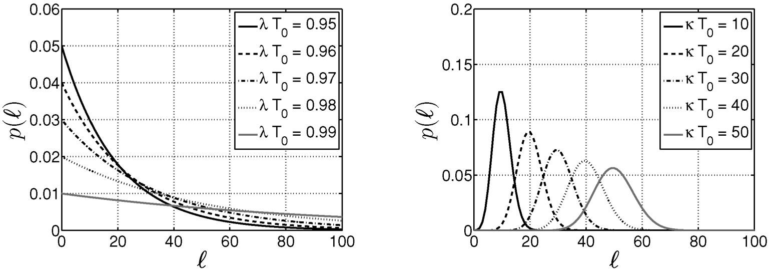
with $\lambda T_0 \in (0, 1)$, as illustrated in Figure 3a. Since the events v_k are assumed to be mutually independent and uniformly distributed in $[0, T_0]$, we have

$$p(v \mid \ell) = \prod_{k=1}^{\ell} p(v_k \mid \ell) = \prod_{k=1}^{\ell} T_0^{-1} = T_0^{-\ell}. \tag{3.3}$$

Therefore,

$$p(v, \ell) = p(v \mid \ell)p(\ell) = (1 - \lambda T_0)\lambda^{\ell}. \tag{3.4}$$

At first sight, it may seem more natural to model v as a Poisson process (Gallager, 1996), which corresponds to a model $p(v, \ell) = p(v \mid \ell)p(\ell)$, where $p(v \mid \ell)$ is also given by equation 3.3 but with different prior $p(\ell)$ for the number of events ℓ (see Figure 3b), that is, a Poisson distribution with



(a) Geometric prior (see equation 3.2) with $\lambda T_0 = 0.95, 0.96, \ldots, 0.99$.

(b) Poisson prior (see equation 3.5) with $\kappa T_0 = 10, 20, \ldots, 50$.

Figure 3: Prior distributions $p(\ell)$ for ℓ: geometric distribution (left) and Poisson distribution (right).

parameter κT_0:

$$p(\ell) = e^{-\kappa T_0} \frac{(\kappa T_0)^\ell}{\ell!}. \tag{3.5}$$

A comparison of Figures 3a and 3b shows that a Poisson prior is more informative than a geometric prior, especially if the parameter λT_0 of the geometric prior takes values close to 1. In fact, among all discrete probability distributions $p(\ell)$ supported on $\{1, 2, 3, \ldots\}$ with given expected value $\mathrm{E}[\ell] = L$, the geometric distribution with parameter $\lambda T_0 = 1 - 1/L$ is the one with the largest entropy. Therefore, if little prior knowledge about the length ℓ is available, it makes sense to use a geometric prior. On the other hand, if substantial prior knowledge about ℓ is available, it can in principle readily be encoded by a Poisson prior. However, for our purposes, the prior, equation 3.5, is mathematically less convenient. We come back to this issue at the end of section 3 and in section 4. Therefore, we adopt a geometric prior even if there is prior knowledge about ℓ. With that choice of prior, the estimates of the parameters δ_t, s_t, and ρ will in principle be less reliable than with a Poisson prior, since the model does not then incorporate all available information. However, the resulting loss in reliability is expected to be negligible. For reasonable lengths ℓ (e.g., $\ell > 30$), most information is typically contained in the observed sequences x and x' and not in the prior $p(\ell)$. The posterior $p(\ell \mid x, x')$ is then tightly concentrated around the true value of ℓ, and the prior $p(\ell)$ varies only slightly over the support of $p(\ell \mid x, x')$. Besides this qualitative argument, we will show experimentally in section 6 that with a geometric prior $p(\ell)$, the obtained estimates of δ_t, s_t, and ρ are reliable (apart from biases due to the ambiguity inherent in event synchrony; see Figure 2). Interestingly, for both types of priors, the events in v occur to a large extent independent of each other, since for given length

ℓ, they are assumed to be mutually independent and uniformly distributed in $[0, T_0]$ (cf. equation 3.3).

We now continue our discussion of model 3.1. The prior on the binary strings b and b' is given by

$$p(b, b' \mid \ell) = p(b \mid \ell)p(b' \mid \ell) = (1 - p_d)^{n+n'} p_d^{2\ell-n-n'} = (1 - p_d)^{n+n'} p_d^{n_{\text{del}}^{\text{tot}}}, \tag{3.6}$$

where $n_{\text{del}}^{\text{tot}}$ is the total number of deleted events in x and x':

$$n_{\text{del}}^{\text{tot}} = n_{\text{del}} + n'_{\text{del}} = 2\ell - n - n', \tag{3.7}$$

with n_{del} the number of deleted events in x:

$$n_{\text{del}} = \ell - n, \tag{3.8}$$

and similarly, n'_{del} the number of deleted events in x':

$$n'_{\text{del}} = \ell - n'. \tag{3.9}$$

For later convenience, we now write ℓ as a function of b and b'. We first expand n_{del}, the number of deleted events in z, as $n_{\text{del}} = n_{\text{del,single}} + n_{\text{del,pair}}$, where $n_{\text{del,single}}$ is the number of single-event deletions in z and $n_{\text{del,pair}}$ is the number of event-pair deletions. Likewise, we can write $n'_{\text{del}} = n'_{\text{del,single}} + n_{\text{del,pair}}$, where $n'_{\text{del,single}}$ is the number of single-event deletions in z'. Since a single deletion in z results in a noncoincident event in x', it follows that

$$n_{\text{del,single}} = n'_{\text{non-co}} = \sum_{k=1}^{n'} b'_k, \tag{3.10}$$

and likewise,

$$n'_{\text{del,single}} = n_{\text{non-co}} = \sum_{k=1}^{n} b_k. \tag{3.11}$$

As a consequence, we have $n_{\text{del}} = n'_{\text{non-co}} + n_{\text{del,pair}}$ and $n'_{\text{del}} = n_{\text{non-co}} + n_{\text{del,pair}}$, and therefore

$$n_{\text{del}}^{\text{tot}} = n_{\text{non-co}} + n'_{\text{non-co}} + 2\,n_{\text{del,pair}} = n_{\text{non-co}}^{\text{tot}} + 2\,n_{\text{del,pair}}, \tag{3.12}$$

with $n^{\text{tot}}_{\text{non-co}} = n_{\text{non-co}} + n'_{\text{non-co}}$. Combining equation 3.12 with 3.7 results eventually in the following expression for ℓ:

$$\ell = \frac{n^{\text{tot}}_{\text{del}} + n + n'}{2} = \frac{n^{\text{tot}}_{\text{non-co}}}{2} + n_{\text{del,pair}} + \frac{n + n'}{2}. \tag{3.13}$$

Note that only the first term on the right-hand side depends on b and b'. In the example of Figure 1b, $\ell = 10$, $n = 8 = n'$, $n_{\text{del}} = 2 = n'_{\text{del}}$, $n^{\text{tot}}_{\text{del}} = 4$, $n_{\text{del,single}} = 1 = n'_{\text{del,single}}$, $n_{\text{del,pair}} = 1$, and $n^{\text{tot}}_{\text{non-co}} = 2$.

Let us now return to model 3.1. The conditional distributions in x and x' are equal to

$$p(x \mid b, v, \delta_t, s_t) = \prod_{k=1}^{n} \left(\mathcal{N}\left(x_k - v_{i_k}; -\frac{\delta_t}{2}, \frac{s_t}{2} \right) \right) \tag{3.14}$$

$$p(x' \mid b', v, \delta_t, s_t) = \prod_{k=1}^{n'} \left(\mathcal{N}\left(x'_k - v_{i'_k}; \frac{\delta_t}{2}, \frac{s_t}{2} \right) \right), \tag{3.15}$$

where v_{i_k} is the event in v that corresponds to x_k and likewise $v_{i'_k}$, and $\mathcal{N}(x; m, s)$ is a univariate gaussian distribution with mean m and variance s. In the example of Figure 1b, $i = (2, 3, 4, 5, 6, 7, 8, 9)$ and $i' = (1, 2, 3, 4, 6, 7, 8, 9)$. For simplicity, we adopt improper priors $p(\delta_t) = 1 = p(s_t)$.

Substituting equations 3.4, 3.6, 3.14, and 3.15 in 3.1 amounts to

$$\begin{aligned} p(x, x', b, b', v, \delta_t, s_t, \ell) = &\prod_{k=1}^{n} \left(\mathcal{N}\left(x_k - v_{i_k}; -\frac{\delta_t}{2}, \frac{s_t}{2} \right) \right) \\ &\cdot \prod_{k=1}^{n'} \left(\mathcal{N}\left(x'_k - v_{i'_k}; \frac{\delta_t}{2}, \frac{s_t}{2} \right) \right) \\ &\cdot (1 - p_d)^{n+n'} p_d^{n^{\text{tot}}_{\text{del}}} (1 - \lambda T_0) \lambda^{\ell}. \end{aligned} \tag{3.16}$$

We now marginalize equation 3.16 with regard to v; later we will marginalize that model with regard to the length ℓ. For a given v_k, three cases are possible:

- The event was not deleted from either x or x', and the corresponding events in x and x' are denoted by x_{j_k} and $x'_{j'_k}$. In equation 3.16, the following term appears:

$$\left(\mathcal{N}\left(x'_{j'_k} - v_k; \frac{\delta_t}{2}, \frac{s_t}{2} \right) \right) \cdot \left(\mathcal{N}\left(x_{j_k} - v_k; -\frac{\delta_t}{2}, \frac{s_t}{2} \right) \right). \tag{3.17}$$

Integrating this term over v_k yields $\mathcal{N}(x_{j_k} - x_{j'_k}; \delta_t, s_t)$. Note that there are $n_{\text{co}}^{\text{tot}}$ such terms, where $n_{\text{co}}^{\text{tot}}$ is the total number of coincident event pairs:

$$n_{\text{co}}^{\text{tot}} = n - n_{\text{non-co}} = n' - n'_{\text{non-co}}. \tag{3.18}$$

- The event was deleted once (from either x or x'). In equation 3.16, there is only one gaussian term that corresponds to v_k. Integrating that term over v_k results in the term 1.
- The event was deleted twice (from x and x'). There are no gaussian terms associated with v_k in equation 3.16. Therefore, expression 3.16 may be considered as constant with regard to v_k. Integrating this equation over v_k then leads to a term T_0. There are $n_{\text{del,pair}}$ such terms.

Eventually we obtain:

$$\begin{aligned} p(x, x', b, b', \delta_t, s_t, \ell) &= \int p(x, x', b, b', v, \delta_t, s_t, \ell)\, dv \\ &= \prod_{k=1}^{n_{\text{co}}^{\text{tot}}} \mathcal{N}\big(x'_{j'_k} - x_{j_k}; \delta_t, s_t\big)(1 - p_d)^{n+n'} p_d^{n_{\text{del}}^{\text{tot}}} \\ &\quad \cdot (1 - \lambda T_0)\lambda^{\ell} T_0^{n_{\text{del,pair}}}. \end{aligned} \tag{3.19}$$

Note that we can marginalize over v analytically (cf. equation 3.19) because of two reasons:

- We have chosen a uniform conditional distribution $p(v \mid \ell)$, equation 3.3.
- We model the offsets between an event v_k and the corresponding events x_ℓ and $x'_{\ell'}$ in x and x', respectively, as gaussian random variables. Therefore, the offset between the two events x_ℓ and $x'_{\ell'}$ is also gaussian distributed.

We wish to point out that more generally, the offset between x_ℓ and $x'_{\ell'}$ may be modeled by any infinite divisible distribution. A probability distribution f on the real line is by definition infinitely divisible if the following holds: if X is any random variable whose distribution is f, then for every positive integer m, there exist m independent identically distributed random variables $X_1, \ldots, X_m$ whose sum is equal in distribution to X. Note that those m other random variables do not usually have the same probability distribution as X. In our setting, the offset between x_ℓ and $x'_{\ell'}$ takes the role of X; we have $m = 2$, and the variables X_1 and X_2 stand for the offset between v_k and x_ℓ and the offset between $x'_{\ell'}$ and v_k, respectively. If the distribution f of the offset between x_ℓ and $x'_{\ell'}$ is infinite divisible, we can decompose that offset as the sum of the offset between v_k and x_ℓ and the offset between $x'_{\ell'}$ and v_k. Indeed, since f is assumed to be infinitely divisible, there exists a distribution $\tilde{f}$ such that the offset between v_k and x_ℓ and the offset between $x'_{\ell'}$ and v_k are independently distributed according to $\tilde{f}$, and their sum is distributed according to f.

Examples of infinitely divisible distributions are the gaussian distribution and the Cauchy distribution, and all other members of the stable distribution family, which is a four-parameter family of continuous probability distributions that has the property of stability. If a number of independent and identically distributed random variables have a stable distribution, then a linear combination of these variables will have the same distribution, except for possibly different shift and scale parameters (Zolotarev, 1986). We will not further consider the extension to general infinite divisible distributions, since gaussian offsets suffice for our purposes.

We now return to model 3.19. Substituting equations 3.12 and 3.13 in 3.19 leads to

$$p(x, x', b, b', \delta_t, s_t, \ell) = \prod_{k=1}^{n_{\mathrm{co}}^{\mathrm{tot}}} \mathcal{N}\big(x'_{j'_k} - x_{j_k}; \delta_t, s_t\big)\big(\sqrt{\lambda}\,(1-p_d)\big)^{n+n'} \cdot \big(\sqrt{\lambda}\,p_d\big)^{n_{\text{non-co}}^{\mathrm{tot}}}(1-\lambda T_0)\big(p_d^2\,\lambda T_0\big)^{n_{\mathrm{del,pair}}}. \tag{3.20}$$

Now we marginalize over the length ℓ. The length can be decomposed according to equation 3.13, where the third term is fixed, since n and n' are the length of the given point processes x and x', respectively. The first term in equation 3.13 is fixed for given b and b'. Therefore, marginalizing $p(x, x', b, b', \delta_t, s_t, \ell)$, equation 3.20, over ℓ is equivalent to marginalizing over $n_{\mathrm{del,pair}}$:

$$p(x, x', b, b', \delta_t, s_t) = \sum_{\ell=0}^{\infty} p(x, x', b, b', \delta_t, s_t, \ell) \tag{3.21}$$

$$= \sum_{n_{\mathrm{del,pair}}=0}^{\infty} p(x, x', b, b', \delta_t, s_t, \ell) \tag{3.22}$$

$$= \prod_{k=1}^{n_{\mathrm{co}}^{\mathrm{tot}}} \mathcal{N}(x'_{j'_k} - x_{j_k}; \delta_t, s_t)\big(\sqrt{\lambda}\,(1-p_d)\big)^{n+n'} \cdot \big(\sqrt{\lambda}\,p_d\big)^{n_{\text{non-co}}^{\mathrm{tot}}}(1-\lambda T_0) \sum_{n_{\mathrm{del,pair}}=0}^{\infty} \big(p_d^2\,\lambda T_0\big)^{n_{\mathrm{del,pair}}} \tag{3.23}$$

$$= \prod_{k=1}^{n_{\mathrm{co}}^{\mathrm{tot}}} \mathcal{N}\big(x'_{j'_k} - x_{j_k}; \delta_t, s_t\big)\big(\sqrt{\lambda}\,(1-p_d)\big)^{n+n'} \cdot \big(\sqrt{\lambda}\,p_d\big)^{n_{\text{non-co}}^{\mathrm{tot}}}(1-\lambda T_0)\,\frac{1}{1-p_d^2\,\lambda T_0}. \tag{3.24}$$

We wish to point out that in equation 3.23, we have a sum of a geometric series; since $|p_d^2\,\lambda T_0| < 1$, we can apply the well-known formula for the sum of a geometric series, resulting in equation 3.24. We can rewrite the latter expression as

$$p(x, x', b, b', \delta_t, s_t) = \gamma\beta^{n_{\text{non-co}}^{\text{tot}}} \prod_{k=1}^{n_{\text{co}}^{\text{tot}}} \mathcal{N}\big(x'_{j'_k} - x_{j_k}; \delta_t, s_t\big), \tag{3.25}$$

with $\beta = p_d\sqrt{\lambda}$ and

$$\gamma = (\sqrt{\lambda}\,(1 - p_d))^{n+n'}(1 - \lambda T_0)\,\frac{1}{1 - p_d^2\,\lambda T_0}. \tag{3.26}$$

The constant γ does not depend on b and b', and therefore, it is irrelevant for estimating b, b', and the SES parameters ρ, δ_t, and s_t. We will discard it in the following. On the other hand, the exponent of β in equation 3.25 clearly depends on b and b' (cf. equations 3.10 and 3.11). Therefore, the parameter β affects the inference of b, b' and the SES parameters. In section 7, we explain how the parameter β may be determined from given sequences x and x'. Moreover, we will interpret the parameter β in terms of cost functions (see equation 3.29). The expression $\log\beta$ is part of the cost associated with each noncoincident event.

After marginalizing with regard to v and ℓ, we obtain a model $p(x, x', b, b', \delta_t, s_t)$ (cf. equation 3.25) that is symmetric in x and x'. In the following, we will denote model 3.25 by $p(x, x', j, j', \delta_t, s_t)$ instead of $p(x, x', b, b', \delta_t, s_t)$, since it is more natural to describe that model in terms of j and j' than in terms of b and b' (cf. the right-hand side of equation 3.25). The sequences b and b' may be obtained directly from j and j': the variables b_k (for all k) equal zero if k appears in the sequence j and one otherwise. The variables b'_k (for all k) may be obtained along the same lines.

It is instructive to consider the negative logarithm of equation 3.25:

$$-\log p(x, x', j, j', \delta_t, s_t) = -n_{\text{non-co}}^{\text{tot}}\log\beta + \frac{1}{2s_t}\sum_{k=1}^{n_{\text{co}}^{\text{tot}}}\big(x'_{j'_k} - x_{j_k} - \delta_t\big)^2 + \frac{n_{\text{co}}^{\text{tot}}}{2}\log 2\pi s_t + \zeta, \tag{3.27}$$

where ζ is an irrelevant constant. As a consequence of equation 3.18, we have

$$n_{\text{co}}^{\text{tot}} = \frac{n + n' - n_{\text{non-co}}^{\text{tot}}}{2}, \tag{3.28}$$

and we can rewrite equation 3.27 as

$$-\log p(x, x', j, j', \delta_t, s_t) = -n^{\text{tot}}_{\text{non-co}}\left(\log\beta + \frac{1}{4}\log 2\pi s_t\right) + \frac{1}{2s_t}\sum_{k=1}^{n^{\text{tot}}_{\text{co}}}\left(x'_{j'_k} - x_{j_k} - \delta_t\right)^2 + \zeta', \tag{3.29}$$

with $\zeta' = \zeta + \frac{n+n'}{4}\log 2\pi s_t$. Note that ζ' does not depend on j and j'; in other words, it is independent of the assignment of coincident and noncoincident events. In the following, we investigate how equation 3.29 depends on the assignment j and j', and ζ' is then irrelevant.

Expression 3.29 may be considered as a cost function that associates certain costs with coincident and noncoincident events. This viewpoint will give us insight into how we can minimize equation 3.29 (equivalently, maximize equation 3.25) with regard to j and j', for fixed δ_t and s_t. The unit cost $d(s_t)$ associated with each noncoincident event equals

$$d(s_t) = -\log\beta - \frac{1}{4}\log 2\pi s_t. \tag{3.30}$$

The unit cost $d(x_{j_k}, x'_{j'_k}; \delta_t, s_t)$ of each event pair $(x_{j_k}, x'_{j'_k})$ is the normalized Euclidean distance:

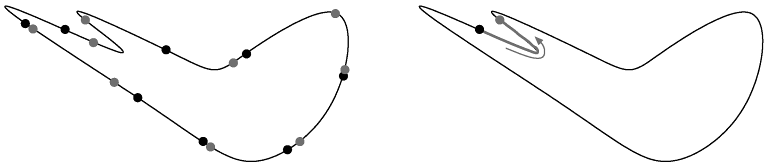
$$d\left(x_{j_k}, x'_{j'_k}; \delta_t, s_t\right) = \frac{\left(x'_{j'_k} - x_{j_k} - \delta_t\right)^2}{2s_t}. \tag{3.31}$$

Since the point processes x and x' of Figure 1b are defined on the real line, the (normalized) Euclidean distance is indeed a natural metric. In some applications, the point process may be defined on more general curves (as illustrated in Figure 4); in such situations, one may adopt non-Euclidean distance measures. We are currently exploring such applications (see Dauwels et al., in press, for preliminary results).

We underline that the unit costs $d(s_t)$, equation 3.30, and $d(x_{j_k}, x'_{j'_k}; \delta_t, s_t)$, equation 3.31, are dimensionless. In other words, they do not depend on the unit (e.g., seconds, milliseconds, meters, or millimeters) in which x and x' are expressed. This property is obvious for $d(x_{j_k}, x'_{j'_k}; \delta_t, s_t)$. For $d(s_t)$, it can be shown as follows:

$$d(s_t) = -\log\beta - \frac{1}{4}\log 2\pi s_t \tag{3.32}$$

$$= -\log p_d - \frac{1}{2}\log\lambda\sqrt{s_t} - \frac{1}{4}\log 2\pi. \tag{3.33}$$



(a) Two point processes defined on a contour (black and gray).

(b) Non-Euclidean distance between two events.

Figure 4: Point processes along a contour. The distance between two events is non-Euclidean.

The parameter p_d is dimensionless; the same holds for the product $\lambda\sqrt{s_t}$, and hence $d(s_t)$ is also dimensionless. Therefore minimizing the cost (see equation 3.29) with regard to the sequences j and j' (for fixed δ_t and s_t), or equivalently, performing maximum a posteriori estimation of j and j' in $p(x, x', j, j', \delta_t, s_t, \ell)$, equation 3.25, will yield the same solutions $\hat{j}$ and $\hat{j}'$, independent of the units of x and x'. In section 4 we explain how one may estimate j and j'.

By interpreting equation 3.29 as a cost function, we also established a connection between SES and the distance metric of Victor and Purpura (1997), Aronov (2003), and Victor et al. (2007). The latter is also formulated in terms of a cost function, more precisely, it is determined as the minimum cost associated with transforming one point process into the other. In this transformation, one is allowed to delete and insert events and move events over time (cf. Figure 1a), and there is a cost associated with each of those three basic operations. For the sake of completeness, we will review the distance metric of Victor and Purpura (1997), Aronov (2003), and Victor et al. (2007) in section 5.1.

Note also that the Poisson prior, equation 3.5, leads to a cost that depends nonlinearly on the number of noncoincident events. The cost per noncoincident event is then not constant but depends on the total number of noncoincident events. Since a constant cost per noncoincident event is easier to interpret and leads to a simpler inference algorithm (see section 4), we decided to use the geometric prior, equation 3.2.

4 Statistical Inference

Given event strings x and x', we wish to infer the parameters δ_t and s_t, and the sequences j and j' (cf. equation 3.25). From decisions $\hat{j}$ and $\hat{j}'$, one can easily determine the corresponding decisions $\hat{b}$ and $\hat{b}'$; the decision $\hat{b}_k$ (for all k) equals zero if k appears in the sequence $\hat{j}$ and is one otherwise. The decisions $\hat{b}'_k$ (for all k) may be obtained along the same lines. The decisions

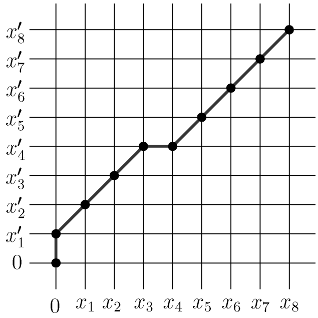


Figure 5: The $(n+1) \times (n'+1)$ grid associated with the point processes x and x' of Figure 1. Each path on that grid corresponds to a pair (j, j'). The path $\mathcal{P}$ shown in this figure corresponds to the alignment of Figure 1b.

$\hat{b}$ and $\hat{b}'$ naturally amount to an estimate of ρ (cf. equation 2.1):

$$\hat{\rho} \triangleq \frac{\sum_{k=1}^{n} \hat{b}_k + \sum_{k=1}^{n'} \hat{b}'_k}{n+n'}. \tag{4.1}$$

Moreover, the parameters T_0, λ, and p_d are unknown and need to be chosen appropriately. Interestingly, they do not need to be specified individually, since they appear in equation 3.25 only through β. The latter serves in practice as a knob to control the number of noncoincident events; we address this issue in section 7.

There are various ways to jointly infer the SES parameters and sequences j and j'; perhaps the most natural solution is coordinate descent. First, one chooses initial values $\hat{\delta}_t^{(0)}$ and $\hat{s}_t^{(0)}$ and then alternates the following two update rules until convergence or until the available time has elapsed:

$$\left(\hat{j}^{(i+1)}, \hat{j}'^{(i+1)}\right) = \underset{j, j'}{\operatorname{argmax}}\, p\left(x, x', j, j', \hat{\delta}_t^{(i)}, \hat{s}_t^{(i)}\right) \tag{4.2}$$

$$\left(\hat{\delta}_t^{(i+1)}, \hat{s}_t^{(i+1)}\right) = \underset{\delta_t, s_t}{\operatorname{argmax}}\, p\left(x, x', \hat{j}^{(i+1)}, \hat{j}'^{(i+1)}, \delta_t, s_t\right). \tag{4.3}$$

In appendix A, we explain how expressions 4.2 and 4.3 can be computed. In particular, we derive a closed-form expression for equation 4.3. We show that equation 4.2 may be obtained by considering paths on a $(n+1) \times (n'+1)$ grid (see Figure 5), where each path corresponds to a pair (j, j'). One can associate a cost M to each path, obtained by adding the costs $d(s_t)$ of each corresponding noncoincident event and the costs $d(x_{j_k}, x'_{j'_k}; \delta_t, s_t)$ associated with each pair of coincident events $(x_{j_k}, x'_{j'_k})$. Finding the pair $(\hat{j}^{(i+1)}, \hat{j}'^{(i+1)})$, equation 4.2, corresponds to the problem of determining the minimum-cost

Table 2: Inference Algorithm for One-Dimensional SES.

Input
One-dimensional point processes x and x' and parameters β, $\hat{\delta}_t^{(0)}$, and $\hat{s}_t^{(0)}$.

Algorithm
Iterate the following two steps until convergence or the available time has elapsed:

1. Update the alignment $(\hat{j}, \hat{j}')$ by dynamic programming:
   Compute the matrix M:
   $M_{k,0} = 0 = M_{0,k'}$ (for $k = 0, 1, \ldots, n$ and $k' = 0, 1, \ldots, n'$). The other elements are computed recursively as

$$M_{k,k'} = \left[M_{k-1,k'} + d\left(\hat{s}_t^{(i)}\right), M_{k,k'-1} + d\left(\hat{s}_t^{(i)}\right), M_{k-1,k'-1} + d\left(x_k, x'_{k'}; \hat{\delta}_t^{(i)}, \hat{s}_t^{(i)}\right)\right].$$

   Determine the min-cost sequence $(\hat{j}, \hat{j}')$ by tracing back the decisions in the recursive updates $M_{k,k'}$.
2. Update the SES parameters:

$$\hat{\delta}_t^{(i+1)} \triangleq \frac{1}{n^{(i+1)}} \sum_{k=1}^{n^{(i+1)}} \hat{x}_k'^{(i+1)} - \hat{x}_k^{(i+1)},$$

$$\hat{s}_t^{(i+1)} \triangleq \frac{1}{n^{(i+1)}} \sum_{k=1}^{n^{(i+1)}} \left(\hat{x}_k'^{(i+1)} - \hat{x}_k^{(i+1)} - \hat{\delta}_t^{(i+1)}\right)^2.$$

Output
Alignment $(\hat{j}, \hat{j}')$ and SES parameters $\hat{\rho}$, $\hat{\delta}_t$, and $\hat{s}_t$.

Note: We refer to appendix A for its derivation.

path, which may be achieved by means of a simple recursion. The resulting algorithm is summarized in Table 2. (Matlab code of the algorithm is available at http://www.dauwels.com/SESToolbox/SES.html.)

Note that if one adopts a Poisson prior, equation 3.5, expression 4.2 may no longer be obtained by determining the minimum-cost path on a grid. In fact, this expression becomes intractable, and one would need to resort to approximative methods.

The algorithm is an instance of coordinate descent, which generally is guaranteed to converge if the iterated conditional maximizations have unique solutions (cf. equations 4.2 and 4.3; Bezdek & Hathaway, 2002; Bezdek, Hathaway, Howard, Wilson, & Windham, 1987). The conditional maximization, equation 4.3, has a unique solution (cf. A.1 and A.3); the solution of equation 4.2 is in most practical situations unique. In our experiments (cf. sections 6 and 7), the algorithm always converged . We provide numerical results on convergence in section 7.

In general, the fixed points of a coordinate descent algorithm are stationary points of the objective function at hand. In particular, alternating equations 4.2 and 4.3 converges to stationary points of $p(x, x', j, j', \delta_t, s_t)$. Since this model may have numerous stationary points, it may be necessary to run these equations with several different initial values $\hat{\delta}_t^{(0)}$ and $\hat{s}_t^{(0)}$, resulting in several fixed points $(\hat{j}, \hat{j}', \hat{\delta}_t, \hat{s}_t)$. Eventually the fixed point with the largest value $p(x, x', \hat{j}, \hat{j}', \hat{\delta}_t, \hat{s}_t)$ is selected. In practice, there is often prior knowledge about δ_t and s_t. For example, in the case of neural spike trains (see section 7), the lag δ_t is usually not larger than 100 ms; similarly, s_t is typically not larger than (100 ms)^2. In most applications, it makes sense to start with the initial value $\hat{\delta}_t^{(0)} = 0$. However, depending on the available computational resources, the algorithm may be run with additional initial values $\hat{\delta}_t^{(0)}$. It may also be run with several values of $\hat{s}_t^{(0)}$ between 0 and the largest plausible value (e.g., $\hat{s}_t^{(0)} = (10\text{ ms})^2, (20\text{ ms})^2, \ldots, (100\text{ ms})^2$). For the sake of completeness, we specify how we selected the initial values $\hat{\delta}_t^{(0)}$ and $\hat{s}_t^{(0)}$ in the applications of section 6 and 7.

In principle, the computational complexity grows proportional to nn' (i.e., the product of both sequence lengths). However, the state space can be restricted to pairs of events that are close to each other—pairs of events $(x_{j_k}, x'_{j'_k})$ that fulfill the constraint $|x_{j_k} - x'_{j'_k}| < \delta_t^{\max}$ (for some $\delta_t^{\max} > 0$, e.g., $\delta_t^{\max} = 100$ ms). The paths on the grid of Figure 5 then remain close to the diagonal, and only the entries $M_{k,k'}$ around the diagonal of M are computed (cf. equation A.4). As a result, the computational complexity becomes linear in the sequence length.

5 Review of Classical Similarity Measures

In this section, we review some of the best-known classical (dis)similarity measures for one-dimensional point processes, including the Victor-Purpura distance metric (Victor & Purpura, 1997; Aronov, 2003; Victor et al., 2007), the van Rossum distance metric (van Rossum, 2001), the Schreiber et al. similarity measure (Schreiber et al., 2003), the Hunter-Milton similarity measure (Hunter & Milton, 2003), and the event synchronization measure proposed in Quian Quiroga, Kreuz, et al. (2002). We have chosen the same selection of classical measures as in Kreuz et al. (2007). Note that this selection is not exhaustive; various other (dis)similarity measures for one-dimensional point processes have recently been proposed, such as the ISI distance in Kreuz et al. (2007). In order to benchmark SES in a meaningful way, however, we have restricted ourselves to existing measures that are the most related to SES.

In this section, we closely follow the exposition in Kreuz et al. (2007). For the sake of definiteness, we will discuss the measures in the context of point processes in time.

5.1 Victor-Purpura Spike Train Metric. The distance metric D_V of (Victor & Purpura, 1997; Aronov, 2003; Victor et al., 2007) is closely related to SES, as we pointed out earlier. It defines the distance between two point processes as the minimum cost of transforming one point process into the other. This transformation is carried out by combining three basic operations: event insertion, event deletion, and event movement (cf. Figure 1a). The cost of deleting or inserting an event is set to one, whereas the cost of moving an event in time is proportional to the time shift; the proportionality constant C_V defines the timescale of the distance metric.

If $C_V = 0$, the distance metric D_V reduces to the difference in number of events. If $C_V \gg 1$, the distance quantifies the number of noncoincident events. Indeed, since for large C_V, it becomes less favorable to move events, one point process is transformed into the other, mostly by inserting and deleting events. The cost D_V associated with this transformation is then (approximately) proportional to the number of noncoincident events. In practice, most events do not perfectly coincide; therefore, in the limit $C_V \gg 1$, virtually all events are either inserted or deleted in the tranformation from one point process to the other. In the intermediate regime $C_V \approx 1$, neighboring events are treated as coincident; they no longer need to occur at precisely the same time. In that regime, D_V is similar to the SES parameter ρ.

It is important to realize that C_V is not dimensionless. Since the cost $C_V \Delta T$ associated with moving an event over ΔT is supposed to be dimensionless, the unit of C_V is the inverse of the unit in which x and x' are expressed. For example, if x and x' are expressed in milliseconds, the condition $C_V \approx 1$ stands for $C_V \approx 10^{-3}$ ms^{-1}. Note that the metric D_V is dimensionless.

If and only if the point processes x and x' are identical, the distance metric $D_V = 0$. The time constant $\tau_V = 1/C_V$, the inverse of C_V, defines the timescale of distance metric D_V.

It is noteworthy that in SES, the unit cost of moving an event in time is quadratic in the time shift (with proportionality constant $1/2s_t$; cf. equation 3.31), in contrast to the Victor-Purpura metric, where the cost is linear in the time shift. Note that the proportionality constant C_V in the Victor-Purpura metric is fixed and needs to be chosen by the user. The proportionality constant $1/2s_t$ in SES is determined adaptively from the given point processes. In both approaches, the minimum cost is computed by the Viterbi algorithm (Forney, 1973; cf. appendix A).

5.2 Van Rossum Similarity Measure. In the approach of van Rossum (2001), the two point processes are converted into continuous time series. In particular, each event of x is convolved with an exponential function $\exp(t - x_k/\tau_R)$ (with $t > x_k$), resulting in the time series $s(t)$. Likewise each event of x' is convolved with this exponential function, leading to the time series $s'(t)$. From the time series $s(t)$ and $s'(t)$, the van Rossum distance measure (van Rossum, 2001) is computed as

$$D_R(\sigma_S) = \frac{1}{\tau_R} \int_t [s(t) - s'(t)]^2 \, dt. \tag{5.1}$$

Note that $D_R(\sigma_S) = 0$ if and only if x and x' are identical. The timescale of this distance measure is determined by the time constant τ_R.

5.3 Schreiber et al. Similarity Measure. In the approach proposed in Haas and White (2002) and Schreiber et al. (2003), the two point processes x and x' are first convolved with a filter, resulting in time series $s(t)$ and $s'(t)$. The filter may, for example, be exponential (Haas & White, 2002) or gaussian (Schreiber et al., 2003), and it has a certain width τ_S. Next the pairwise correlation between the time series $s(t)$ and $s'(t)$ is computed:

$$S_S(\sigma_S) = \frac{\int_t s(t)s'(t)\,dt}{\sqrt{\int_t s^2(t)\,dt}\sqrt{\int_t s'^2(t)\,dt}}. \tag{5.2}$$

In Haas and White (2002), one adjusts the phase lag between the time series, whereas in Schreiber et al. (2003), no phase lag is allowed. In this letter (as in Kreuz et al., 2007), we consider the approach of Schreiber et al. (2003). Note that the width τ_S of the filter defines the timescale of interaction between the two point processes. We also point out that if and only if x and x' are identical do we have $S_S = 1$.

5.4 Hunter-Milton Similarity Measure. An alternative similarity measure was proposed in Hunter and Milton (2003). For each event x_k, one identifies the nearest event $x'_{k'(k)}$ in the point process x'. The degree of coincidence between those two events is determined as $d(x_k) = \exp(-|x_k - x'_{k'(k)}|/\tau_H)$. Along the same lines, one identifies for each $x'_{k'}$ the nearest event $x_{k(k')}$ in the point process x and determines the degree of coincidence $d(x'_{k'})$. The similarity S_H between x and x' is then computed as

$$S_H = \frac{\frac{1}{N}\sum_{k=1}^{N} d(x_k) + \frac{1}{N'}\sum_{k'=1}^{N'} d(x'_k)}{2}. \tag{5.3}$$

The parameter τ_H sets the timescale for event coincidence. If x and x' are identical, we have $S_H = 1$.

5.5 Event Synchronization. Event synchronization (Quian Quiroga, Kreuz, et al., 2002) defines similarity in terms of coincident events. Two events are considered to be coincident if their timing offset is smaller than a maximum lag τ_Q. This lag can be fixed by the user, or it can be extracted automatically from the point processes x and x':

$$\tau_Q(k, k') = \min(x_{k+1} - x_k, x_k - x_{k-1}, x'_{k'+1} - x'_{k'}, x'_{k'} - x'_{k'-1})/2. \tag{5.4}$$

One computes the number of times an event appears in x shortly after an event appears in x':

$$d(x \mid x') = \sum_{k=1}^{N} \sum_{k'=1}^{N'} J_{kk'}, \tag{5.5}$$

where

$$J_{kk'} = \begin{cases} 1 & \text{if } 0 < x_k - x'_{k'} \leq \tau_Q \\ 1/2 & \text{if } x_k = x'_{k'} \\ 0 & \text{else,} \end{cases} \tag{5.6}$$

where τ_Q may be fixed or may be computed according to equation 5.4.

Similarly one can define $d(x' \mid x)$, and eventually event synchronization is determined as

$$S_Q = \frac{d(x \mid x') + d(x'|x)}{\sqrt{NN'}}. \tag{5.7}$$

If and only if all events in x and x' are coincident, we have $S_Q = 1$.

5.6 Discussion

5.6.1 Binning. Interestingly, the classical approaches and SES do not discretize the time (or space) axis, in contrast to other methods (e.g., Johnson, Gruner, Baggerly, & Seshagiri, 2001; Christen, Kohn, Ott, & Stoop, 2006). The latter divide the time (space) axis in bins and then convert the point processes into binary sequences. If an event occurred within a bin, then a one is associated with that bin, otherwise a zero. A critical issue is the choice of bin width, since the results may depend on this parameter. SES and the classical measures avoid that issue since they do not rely on binning. This also holds for other measures, for example, the ISI distance (Kreuz et al., 2007).

5.6.2 Timescale. Several of the above measures depend on a parameter that defines the timescale of the interaction between the point processes, in particular, the Victor-Purpura distance metric (Victor & Purpura, 1997; Aronov, 2003; Victor et al., 2007), the van Rossum similarity measure (van Rossum, 2001), the Schreiber et al. similarity measure (Schreiber et al., 2003), and the Hunter-Milton similarity measure (Hunter & Milton, 2003). Event synchronization, however, adapts its timescale automatically; the user does not need to specify it. The same holds for SES: the timescale is determined by the parameter s_t, which is computed by the algorithm, and does not need to be specified a priori. One just needs to choose initial values $\hat{s}_t^{(0)}$ within the range of plausible values. The SES inference algorithm (cf. Table 2) then refines those initial estimates and selects the most appropriate one. Besides event synchronization and SES, also other existing measures do not rely on a fixed time scale, e.g., the ISI-distance (Kreuz et al., 2007).

In some applications, the user may prefer to use an automatic procedure to determine the timescale; in other applications, one may wish to investigate how the similarity depends on the timescale. For instance, the timescale may be chosen based on optimizing some desired quantity, such as classification fidelity, as in Victor and Purpura (1997). In event synchronization, one can fix the timescale instead of using the adaptive rule, equation 5.4. Likewise, in SES, one may fix s_t instead of estimating it.

5.6.3 Delays. There might be a delay between the two point processes x and x'. Before the classical measures can be applied, potential delays need to be estimated and the point processes shifted accordingly. SES directly handles delays and does not require a separate procedure to estimate delays. As a consequence, the estimates of s_t and ρ are robust against lags between the point processes, as we demonstrate in section 6.

5.6.4 Matching. The van Rossum and Schreiber et al. measures allow matching between a single event in one train and multiple events in the other, since the exponential kernel of an event in one train may overlap with the exponential kernels of multiple events in the other train. Similarly in event synchronization and the Hunter-Milton measure, an event may be matched with multiple events. In SES and the Victor-Purpura metric, each event can be coincident with at most one other event.

6 Analysis of Surrogate Data

Here we investigate the robustness and reliability of SES and the classical (dis)similarity measures reviewed in section 5. In order to benchmark the different measures, we apply them to surrogate data. For these data, the true values of event reliability and timing jitter are known and directly controllable. As far as we know, such comparison using surrogate data has not been carried out yet. In the investigation of Tiesinga et al. (2008), the measures were applied to spike trains generated by a Hogdkin-Huxley type model; for such models, the true values of event reliability and timing jitter are unknown.

We randomly generated 10,000 pairs of one-dimensional point processes (x, x') according to the symmetric procedure depicted in Figure 1b. For definiteness, we assume that x and x' are point processes in time. We considered several values of the parameters ℓ, p_d, δ_t, and s_t (σ_t). More specifically, the length ℓ was chosen as $\ell = \ell_0/(1 - p_d)$, where $\ell_0 \in \mathbb{N}_0$ is a constant. With this choice, the expected length of x and x' is ℓ_0, which is independent of p_d. We considered the values $\ell_0 = 40$ and 100, $p_d = 0, 0.1, \ldots, 0.4, \delta_t = 0$ ms, 25 ms, 50 ms, and $\sigma_t = 10$ ms, 30 ms, and 50 ms. The parameter T_0 was chosen as $\ell_0 \cdot 100$ ms. The average spiking rate therefore is about 10 Hz for all choices of ℓ_0 and p_d.

In the SES approach, we used the initial values $\hat{\delta}_t^{(0)} = 0$, 30, and 70 and $\hat{s}_t^{(0)} = (30\text{ ms})^2$. The parameter β was identical for all settings of ℓ, p_d, δ_t, and s_t: $\beta = 0.02$. It was optimized to yield the best overall results. There are perhaps ways to determine β from a single pair of point processes, which would allow us to determine β for each setting of ℓ, p_d, δ_t, and s_t separately; we leave this issue as a topic for further research. In practice, however, one often considers multiple point processes simultaneously. In section 7 we determine the SES parameters from multiple point processes and describe a method to determine the optimal parameter β.

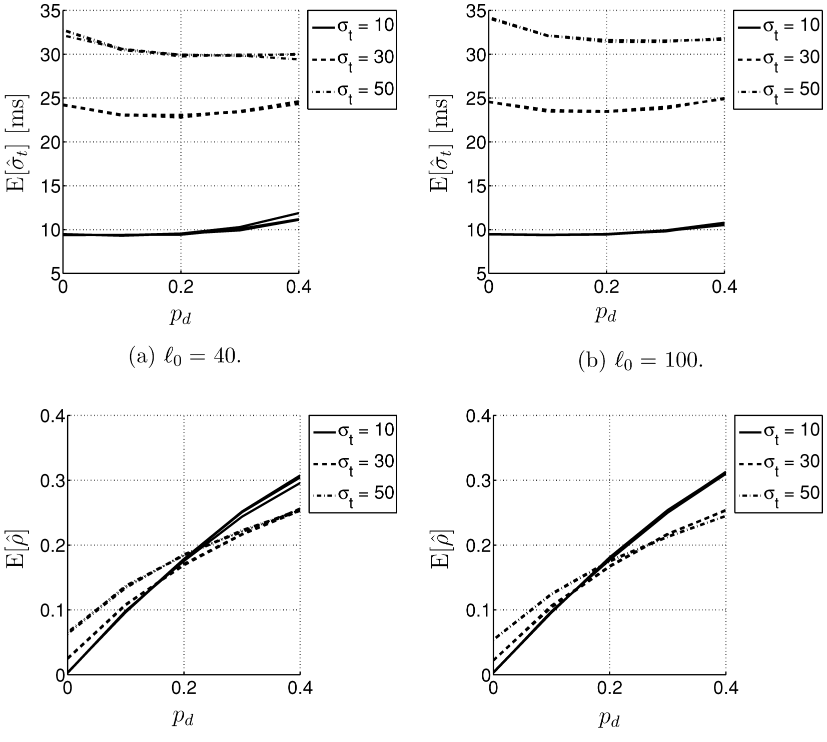
The constant C_V of the Victor-Purpura metric was set to 0.001 and 0.1 ms^{-1}. For the time constants τ_R, τ_S, and τ_H (cf. section 5), we considered the values 10 ms and 20 ms. For the time constant τ_Q (cf. section 5), we chose the values 20 ms and 40 ms. Those values of the different time constants seemed to yield the most reasonable results. Since we consider different lengths ℓ_0, we normalized the Victor-Purpura metric D_V by the number of events in both point processes: that is, we consider the normalized metric $\bar{D}_V$ defined as

$$\bar{D}_V = \frac{D_V}{n + n'}. \tag{6.1}$$

In order to assess the (dis)similarity measures, we compute for each parameter setting and for each measure S the expectation $\mathrm{E}[S]$ and normalized standard deviation $\overline{\sigma}[S] = \sigma[S]/\mathrm{E}[S]$. Those statistics are computed by averaging over 10,000 pairs of point processes (x,x'), randomly generated according to the symmetric procedure depicted in Figure 1b.

6.1 Results. Results for SES are summarized in Figure 6. From this figure, we can make the following observations:

- The estimates of s_t and ρ are biased, especially for small ℓ_0 (in particular $\ell_0 = 40$), $s_t \geq (30\text{ ms})^2$, and $p_d > 0.2$; more specifically, the expected value of those estimates is smaller than the true value, which is due to the ambiguity inherent in event synchrony (cf. Figure 2). On the other hand, the estimates of δ_t are unbiased for all considered values of δ_t, s_t, and p_d (not shown here).
- The estimates of s_t only weakly depend on p_d, and vice versa.
- The estimates of s_t and ρ do not depend on δ_t; they are robust to lags δ_t, since the latter can be estimated reliably.
- The normalized standard deviation of the estimates of δ_t, s_t, and ρ grows with s_t and p_d, but it remains below 30% (not shown here). Those estimates are therefore reliable.



(a) $\ell_0 = 40$.

(b) $\ell_0 = 100$.

(c) $\ell_0 = 40$.

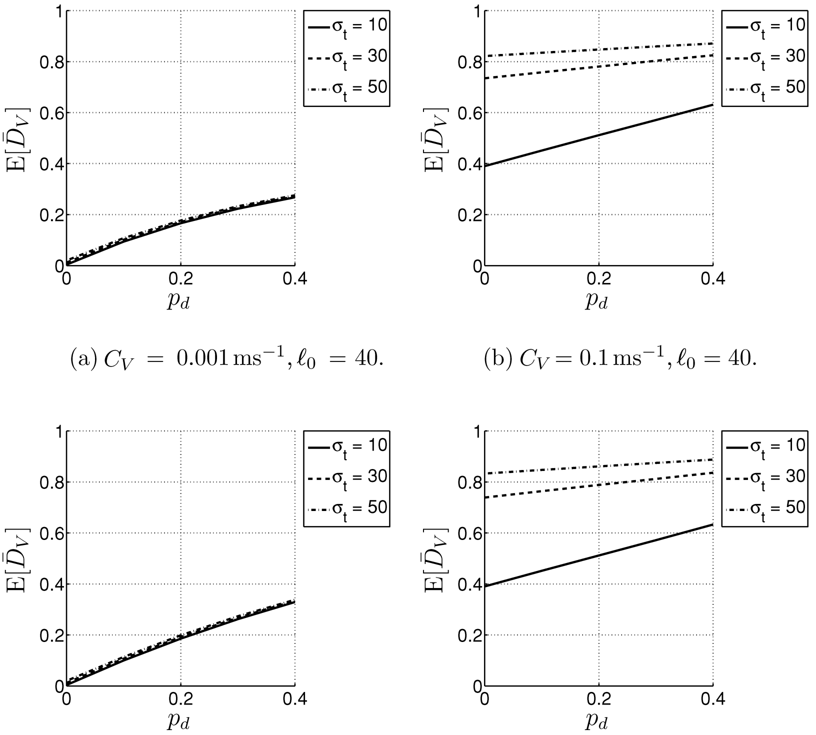
(d) $\ell_0 = 100$.

Figure 6: Results for stochastic event synchrony. The figure shows the expected value $E[\hat{\sigma}_t]$ and $E[\hat{\rho}]$ for the parameter settings $\ell_0 = 40$ and 100, $\delta_t = 0$ ms, 25 ms, 50 ms, $\sigma_t = 10$ ms, 30 ms, 50 ms, and $p_d = 0, 0.1, \ldots, 0.4$. The curves for different δ_t are practically coinciding.

- The expected value of the estimates of s_t and ρ barely depends on the length ℓ_0. The normalized standard deviation of the SES parameters decreases as the length ℓ_0 increases (not shown here), as expected.

In other words, the SES algorithm results in reliable estimates of the SES parameters s_t and ρ.

Results for the classical measures reviewed in section 5 are summarized in Figures 7 and 8. For clarity, we show only the results for $\delta_t = 0$ in those figures. The influence of lags on classical measures will be investigated later in this section (see Figure 9a). We first consider the results for the Victor-Purpura distance metric (Victor & Purpura, 1997; Aronov, 2003; Victor et al., 2007), which are summarized in Figure 7. From that figure we can see the following:



(a) $C_V = 0.001\,\mathrm{ms}^{-1}, \ell_0 = 40.$

(b) $C_V = 0.1\,\mathrm{ms}^{-1}, \ell_0 = 40.$

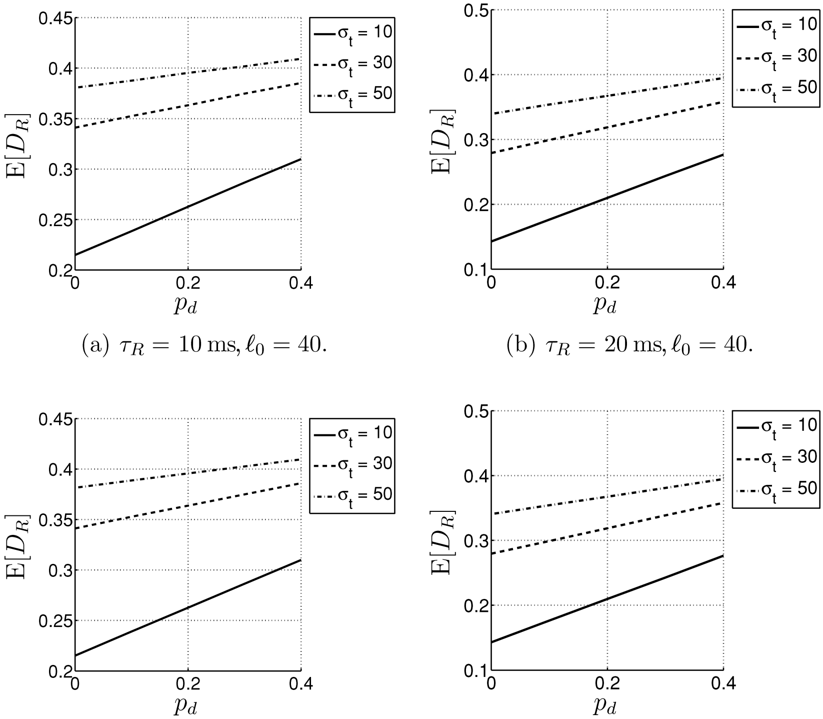
(c) $C_V = 0.001\mathrm{ms}^{-1}, \ell_0 = 100.$

(d) $C_V = 0.1\,\mathrm{ms}^{-1}, \ell_0 = 100.$

Figure 7: Results for (normalized) Victor-Purpura distance metric $\bar{D}_V$. The figure shows the expected value E[$\bar{D}_V$] for the parameter settings $\ell_0 = 40$ and $100, \delta_t = 0\,\mathrm{ms}\, \sigma_t = 10\,\mathrm{ms}, 30\,\mathrm{ms}, 50\,\mathrm{ms}$, $p_d = 0, 0.1, \ldots, 0.4$ and $C_V = 0.001\,\mathrm{ms}^{-1}$ and $0.1\,\mathrm{ms}^{-1}$.

- For $C_V = 0.001\ \mathrm{ms}^{-1}$, the distance metric $\bar{D}_V$ grows with p_d and is practically independent of s_t. (This is the "intermediate" regime mentioned in section 5.1.) In this regime, the metric $\bar{D}_V$ is proportional to the number of noncoincident events, and it behaves similarly to ρ; however, due to ambiguity inherent in event synchrony (cf. Figure 2), for $p_d > 0.2$, it overestimates the number of coincident events and underestimates p_d. For larger C_V, in particular $C_V = 0.1\ \mathrm{ms}^{-1}$, the metric clearly depends on both p_d and s_t. It is noteworthy that the value $C_V = 0.1\ \mathrm{ms}^{-1}$ depends on T_0/ℓ_0 (average distance between events in x and x'), which is 100 ms.

  If $C_V = 0\ \mathrm{ms}^{-1}$ (not shown here), the metric $\bar{D}_V$ is close to zero, since it is equal to the difference in length of both point processes x and x', and in our experiments, both point processes have equal length on



(a) $\tau_R = 10\,\mathrm{ms}, \ell_0 = 40.$

(b) $\tau_R = 20\,\mathrm{ms}, \ell_0 = 40.$

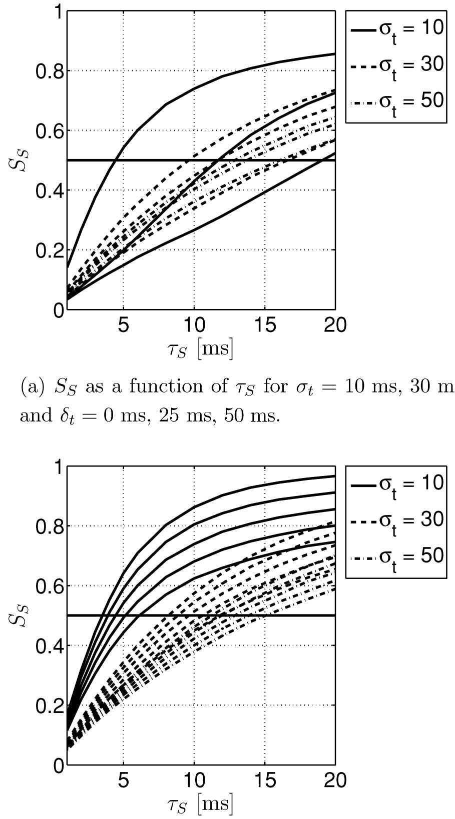
(c) $\tau_R = 10\,\mathrm{ms}, \ell_0 = 100.$

(d) $\tau_R = 20\,\mathrm{ms}, \ell_0 = 100.$

Figure 8: Results for van Rossum distance metric D_R. The figure shows the expected value E[D_R] for the parameter settings $\ell_0 = 40$ and 100, $\delta_t = 0$ ms, $\sigma_t = 10$ ms, 30 ms, 50 ms, $p_d = 0, 0.1, \ldots, 0.4$, and $\tau_R = 10$ ms and 20 ms.

average. However, note that the metric $\bar{D}_V$ is not exactly equal to zero, since the length of the sequences is not identical in every realization but only on average. The difference in length fluctuates more strongly as p_d increases. Therefore, for $C_V = 0$ ms^{-1}, $\bar{D}_V$ increases (weakly) with p_d, independent of s_t, but remains close to zero ($\bar{D}_V < 0.1$). On the other hand, if $C_V \gg 1\,\mathrm{ms}^{-1}$ (not shown here), the metric $\bar{D}_V$ is close to one, independent of p_d and s_t. In the transformation of one point process into the other, (almost) every event is either deleted or inserted, and therefore, a cost of 1 is associated with (almost) every event.

- The normalized standard deviation of $\bar{D}_V$ decreases with p_d and s_t, and it remains below 30% (not shown here). The estimates of $\bar{D}_V$ are therefore reliable.



(a) S_S as a function of τ_S for $\sigma_t = 10$ ms, 30 ms, 50 ms, and $\delta_t = 0$ ms, 25 ms, 50 ms.

(b) S_S as a function of τ_S for $\sigma_t = 10$ ms, 30 ms, 50 ms, and $p_d = 0, 0.1, \ldots, 0.4$.

Figure 9: Sensitivity of the Schreiber et al. measure S_S to δ_t and p_d, with $\ell_0 = 100$ and $\sigma_t = 10$ ms, 30 ms, 50 ms. (top) The parameter settings are $p_d = 0.2$ and $\delta_t = 0$ ms, 25 ms, 50 ms. Note that the similarity S_S decreases with δ_t. (bottom) The parameter settings are $p_d = 0, 0.1, \ldots, 0.4$ and $\delta_t = 0$ ms.

- The expected value of $\bar{D}_V$ does not depend on the length ℓ_0; however, its normalized standard deviation decreases as the length ℓ_0 increases (not shown here).

The results for the van Rossum distance measure D_R (van Rossum, 2001) are summarized in Figure 8. From that figure one can see the following:

- The distance metric D_R grows with both p_d and s_t, as does the distance metric $\bar{D}_V$ (for $C_V \gg 0.001\ \mathrm{ms}^{-1}$).
- The normalized standard deviation of D_R is largely independent of p_d and decreases with s_t (not shown here). Since it remains below 15%, the estimates of D_R are reliable.

- Like the metric $\bar{D}_V$, the expected value of D_R does not depend on the length ℓ_0; however, its normalized standard deviation decreases as the length ℓ_0 increases (not shown here).
- As the time constant τ_R increases, the expected value of D_R and its normalized standard deviation decrease (not shown here). This can be explained as follows: the larger τ_R, the more the time series $s(t)$ and $s'(t)$ overlap, and hence the smaller D_R. Since there are generally also more coincident events for larger D_R, the fluctuations of D_R are smaller. As a result, its normalized standard deviation becomes smaller.

We made similar observations for the Schreiber et al. measure S_S (Schreiber et al., 2003), the Hunter-Milton measure S_H (Hunter & Milton, 2003), and event synchronization S_Q (both with fixed and adaptive time constant $\tau_Q(k, k')$; see equation 5.4).

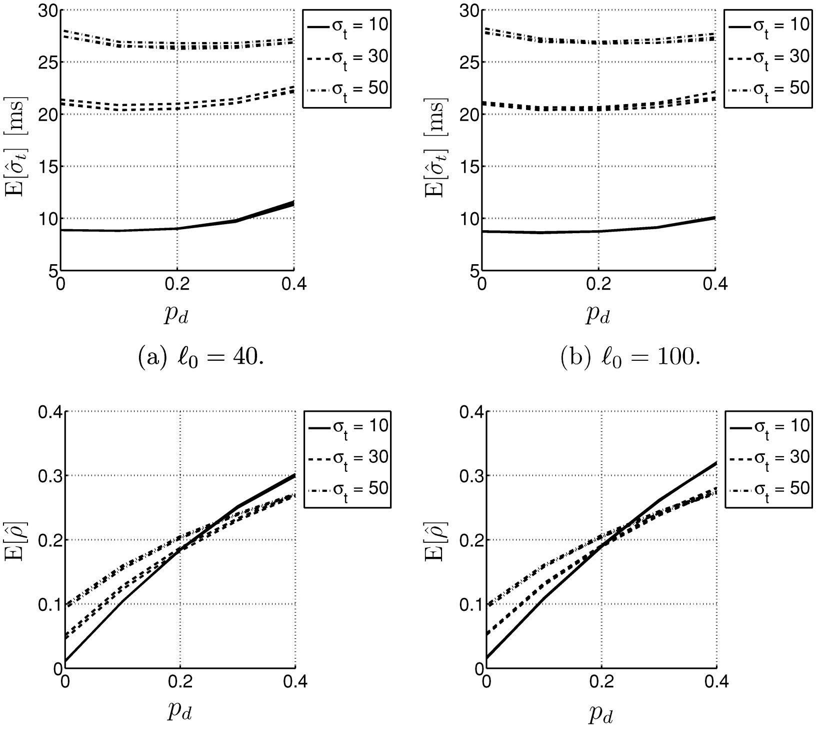
In order to assess the robustness of the SES algorithm, we also analyzed surrogate data generated by an alternative procedure. In the symmetric procedure depicted in Figure 1b, the timing perturbations are drawn not from a gaussian distribution but from a *Laplacian* distribution instead:

$$p(x; m, w) = \frac{1}{2w} \exp\left(-\frac{|x-m|}{w}\right), \tag{6.2}$$

where w is a scale parameter. The variance s of the Laplacian distribution is given by $s = 2w^2$, and hence the parameter w is related to the variance s as $w = \sqrt{s/2}$.

More specifically, the generative process is as follows: First, one generates an event string v of length ℓ, where the events v_k are mutually independent and uniformly distributed in $[0, T_0]$. The strings z and z' are generated by delaying v over $-\delta_t/2$ and $\delta_t/2$, respectively, and by (slightly) perturbing the resulting event occurrence times. Those perturbations are zero-mean Laplacian variables with variance $s_t/2$; the scale parameter w (cf. equation 6.2) is chosen as $w = \sqrt{s_t/4} = \sigma_t/2$. Next, some of the events in z and z' are removed, resulting in the sequences x and x'; each event of z and z' is removed with probability p_d ("deletion"), independent of the other events.

We again considered the values $\ell_0 = 40$ and 100, $p_d = 0, 0.1, \ldots, 0.4$, $\delta_t = 0$ ms, 25 ms, 50 ms, and $\sigma_t = 10$ ms, 30 ms, and 50 ms, and the parameter T_0 was again chosen as $\ell_0 \cdot 100$ ms. The results are summarized in Figure 10. A comparison of Figure 10 with Figure 6 shows that the results for both generative processes are very similar. This suggests that SES can robustly quantify timing precision and event reliability.



(a) $\ell_0 = 40$.

(b) $\ell_0 = 100$.

(c) $\ell_0 = 40$.

(d) $\ell_0 = 100$.

Figure 10: Results for stochastic event synchrony for the surrogate data with Laplacian timing perturbations. The figure shows the expected value $\mathrm{E}[\hat{\sigma}_t]$ and $\mathrm{E}[\hat{\rho}]$ for the parameter settings $\ell_0 = 40$ and 100, $\delta_t = 0$ ms, 25 ms, 50 ms, $\sigma_t =$ 10 ms, 30 ms, 50 ms, and $p_d = 0, 0.1, \ldots, 0.4$. The curves for different δ_t are practically coinciding.

6.2 Discussion. From this study of surrogate data, we can conclude the following:

- SES and the classical measures considered in this letter are reliable in the sense that their statistical fluctuations are relatively small. Their normalized standard deviation is typically below 30%, and often even below 20%.
- The longer the point process, the more reliable the measures become, as expected.
- The classical measures are sensitive to lags. Therefore, one needs to estimate potential lags before they can be applied. As an illustration, Figure 9a shows how the Schreiber et al. measure S_S depends on δ_t; clearly, S_S drops as the delay δ_t increases. On the other hand, SES directly

incorporates lags, and as a result, the estimates s_t and ρ are robust to lags (cf. Figure 6). However, it is critical to choose an appropriate set of initial values $\hat{\delta}_t^{(0)}$. For example, if one uses only $\hat{\delta}_t^{(0)} = 0$ as the initial value, the estimates s_t and ρ become more sensitive to lags (not shown here). In other words, one of the initial values should be sufficiently close to the true lag. Therefore, prior information about potential lags is crucial for the success of the SES algorithm. If no such prior information is available, one needs to choose multiple initial values in a wide range; if the true lag falls within that range, the SES algorithm will most likely yield reliable estimates of s_t and ρ. On the other hand, if the true lag is far from the initial values $\hat{\delta}_t^{(0)}$, the estimates of s_t and ρ may not be reliable.

• Most classical measures depend on both p_d and s_t, and therefore they are unable to separate the two key aspects of synchrony: timing precision and event reliability. There is one exception: the distance metric $\bar{D}_V$ grows with p_d for small cost factors C_V, independent of s_t (cf. Figures 7a and 7c). The same holds for the SES parameter ρ (cf. Figures 6c and 6d); both $\bar{D}_V$ and ρ are measures of event reliability. Note that ρ is robust to lags δ_t, in contrast to $\bar{D}_V$. The SES parameter s_t is largely independent of p_d (cf. Figures 6a and 6b); it is a robust measure for timing dispersion. Interestingly, the parameters s_t and ρ seem to quantify timing precision and event reliability, respectively, even if the data at hand are generated from a model that differs from the SES model (cf. Figure 1b). We wish to point out once more, however, that all (dis)similarity measures for one-dimensional point processes underestimate the timing dispersion and the number of event deletions due to the ambiguity inherent in event synchrony (cf. Figure 2).

• There exists a classical procedure to estimate the timing dispersion based on the Schreiber et al. measure S_S (see, e.g., Tiesinga et al., 2008). One computes S_S for a range of values of τ_S. The value of τ_S at which $S_S = 0.5$ is considered as an estimate σ_S of the timing dispersion. Similarly one may determine timing dispersion from other classical measures, such as the Hunter-Milton similarity measure. It is important to realize, however, that since the classical measures significantly depend on p_d (with the exception of the Victor-Purpura distance for sufficiently small C_V), the resulting estimates of timing dispersion will significantly depend on p_d. This is illustrated in Figure 9b. From the figure, it can be seen that both the similarity measure S_S and the timing dispersion estimate σ_S significantly depend on p_d. For example, σ_S is equal to 12 ms for the parameter settings ($\sigma_t = 30$ ms, $p_d = 0.4$) and ($\sigma_t = 50$ ms, $p_d = 0.1$); in other words, σ_S is not a reliable measure for timing dispersion, and the same holds for similar estimates of timing dispersion, for example, derived from the Hunter-Milton similarity measure. In contrast, the estimate $\hat{s}_t$ of the SES parameter s_t does not suffer from those shortcomings (see Figures 6a and 6b).

• SES is significantly more computationally complex than some classical similarity measures, such as the Hunter-Milton similarity measure. In principle, the complexity of the SES inference algorithm scales quadratically with the sequence length. Without further modifications, the SES algorithm is practical only for sequences of length 100 and less. However, a very reasonable approach to limit the computationally complexity is to consider only pairs of events that are sufficiently close to each other. For example, in the application at hand, it is not likely that two events with a lag of more than 500 ms form an event pair. Therefore, such pairs can be discarded a priori in the SES algorithm. The complexity then becomes linear in the sequence length, and the SES algorithm remains practical for sequences of length 1000 and more.

- The SES algorithm leads to reliable estimates of s_t and ρ only if the parameter β is appropriately chosen. In the application at hand, β was fixed for all parameter settings, and we chose the value of β that resulted in the most reliable estimates. In the application in section 7, we will propose a technique to determine β from multiple given point processes.

7 Application: Firing Reliablity of a Neuron

In this section, we investigate an application related to neuroscience. In particular, we apply SES to quantify the firing reliability of neurons. We consider the Morris-Lecar neuron model (Morris & Lecar, 1981), which exhibits properties of type I and II neurons (Gutkin & Ermentrout, 1998; Tsumoto, Kitajima, Yoshinaga, Aihara, & Kawakami, 2006; Tateno & Pakdaman, 2004). The spiking behavior differs in both neuron types, as illustrated in Figures 12 and 13. In type II neurons, the timing jitter is small, but spikes tend to drop out. In type I neurons, fewer spikes drop out, but the dispersion of spike times is larger. In other words, type II neurons prefer to stay coherent or to be silent; type I neurons follow the middle course between those two extremes (Robinson, 2003).

This difference in spiking behavior is due to the way periodic firing is established (Gutkin & Ermentrout, 1998; Tsumoto et al., 2006; Tateno & Pakdaman, 2004; Robinson, 2003). In type I neurons, periodic firing results from a saddle-node bifurcation of equilibrium points. Such neurons show a continuous transition from zero frequency to arbitrary low frequency of firing. Pyramidal cells are believed to be type I neurons. In type II neurons, periodic firing occurs by a subcritical Hopf bifurcation. Such neurons show an abrupt onset of repetitive firing at a higher firing frequency; they cannot support regular low-frequency firing. Squid giant axons and the Hodgkin-Huxley model are type II.

In the following section, we describe the Morris-Lecar neuron model in more detail. In section 7.2.1, we apply both SES and classical (dis)similarity to quantify the firing reliability of both types of neurons and discuss how the difference in spiking behavior is reflected in those (dis)similarity measures.

Table 3: Parameter Setting for Type I and II Morris-Lecar Neurons.

Parameter	Type I	Type II
g_{Ca} [μS/cm^2]	4.0	4.4
ϕ [s^{-1}]	1/15	1/25
V_3 [mV]	12	2
V_4 [mV]	17.4	30

Table 4: Fixed Parameters for the Morris-Lecar Neuron.

Parameter	Value
C_M	5 [μF/cm^2]
g_{K}	8 [μS/cm^2]
g_{L}	2 [μS/cm^2]
V_{Ca}	120 [mV]
V_{K}	−80 [mV]
V_{L}	−60 [mV]
V_1	−1.2 [mV]
V_2	18 [mV]

Note: This parameter setting is used in both types of neurons.

7.1 Morris-Lecar Neuron Model. The Morris-Lecar neuron model is described by (Morris & Lecar, 1981):

$$C_M \frac{dV}{dt} = -g_{\mathrm{L}}(V - V_L) - g_{\mathrm{Ca}} M_\infty (V - V_{\mathrm{Ca}}) - g_{\mathrm{K}} N (V - V_{\mathrm{K}}) + I_{\mathrm{ext}} \tag{7.1}$$

$$\frac{dN}{dt} = \lambda_N (N_\infty - N), \tag{7.2}$$

where M_∞, N_∞, and λ_N are the following functions:

$$M_\infty = 0.5(1 + \tanh((V - V_1)/V_2)) \tag{7.3}$$

$$N_\infty = 0.5(1 + \tanh((V - V_3)/V_4)) \tag{7.4}$$

$$\lambda_N = \phi \cosh((V - V_3)/2V_4). \tag{7.5}$$

Depending on the parameters of the system, the M-L neuron model behaves as a type I or II neuron. Gutkin and Ermentrout (1998) have determined a setting of the system parameters for each type. Table 3 lists parameter values that are different in the two classes, and Table 4 lists common parameter values. The analysis of Gutkin and Ermentrout (1998) was further refined in Tsumoto et al. (2006) and Tateno and Pakdaman (2004). However, for our purposes, the parameter setting of Tables 3 and 4 suffices.

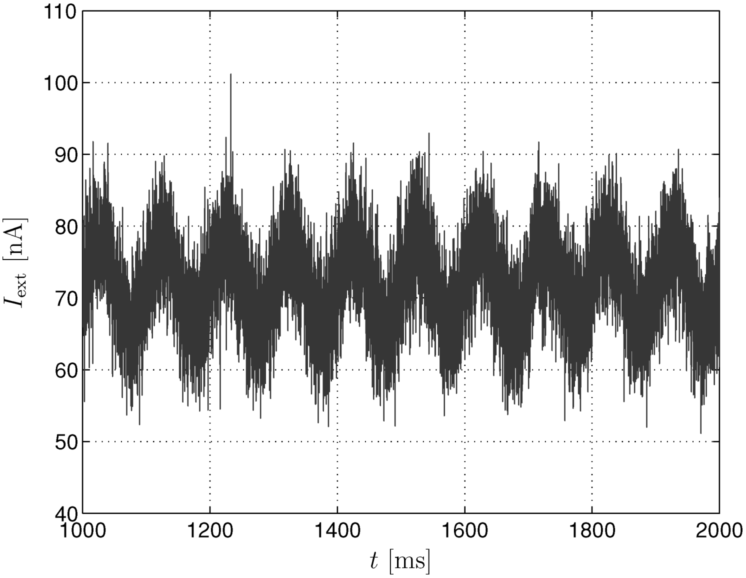


Figure 11: A realization of the input current I_{ext}, equation 7.6. It consists of a baseline B, a sinusoid with amplitude A and frequency f, and additive white gaussian noise with variance σ_n^2.

In our experiments, the input current I_{ext} is equal to

$$I_{\text{ext}} = A\sin(2\pi f t) + B + n(t), \tag{7.6}$$

where $n(t)$ is zero-mean white gaussian noise with variance σ_n^2. Figure 11 shows a realization of I_{ext}. The sinusoidal component forces the neuron to spike regularly; however, the precise timing varies from trial to trial due to the noise $n(t)$. Our objective is to investigate how the noise affects the spike timing and the tendency to drop spikes. We are especially interested in how the effect of noise differs in both neuron types. The parameter settings for the input current I_{ext} are listed in Table 5. We have chosen the parameters such that we obtain the typical spiking behavior of both types of neurons, as described in Robinson (2003). Figure 12 shows the membrane potential V, equation 7.1, for five trials. By thresholding V, we obtain the raster plots of Figure 13; we show 50 trials.

7.2 Results. We present the results for the SES approach in section 7.2.1. In section 7.2.2, we discuss the results for classical methods.

7.2.1 Stochastic Event Synchrony. We computed the SES parameters for each pair of the 50 trials and for different values of β. Next we averaged

Table 5: Parameters of Input Current I_{ext}, Equation 7.6, for Type I and II Morris-Lecar Neurons.

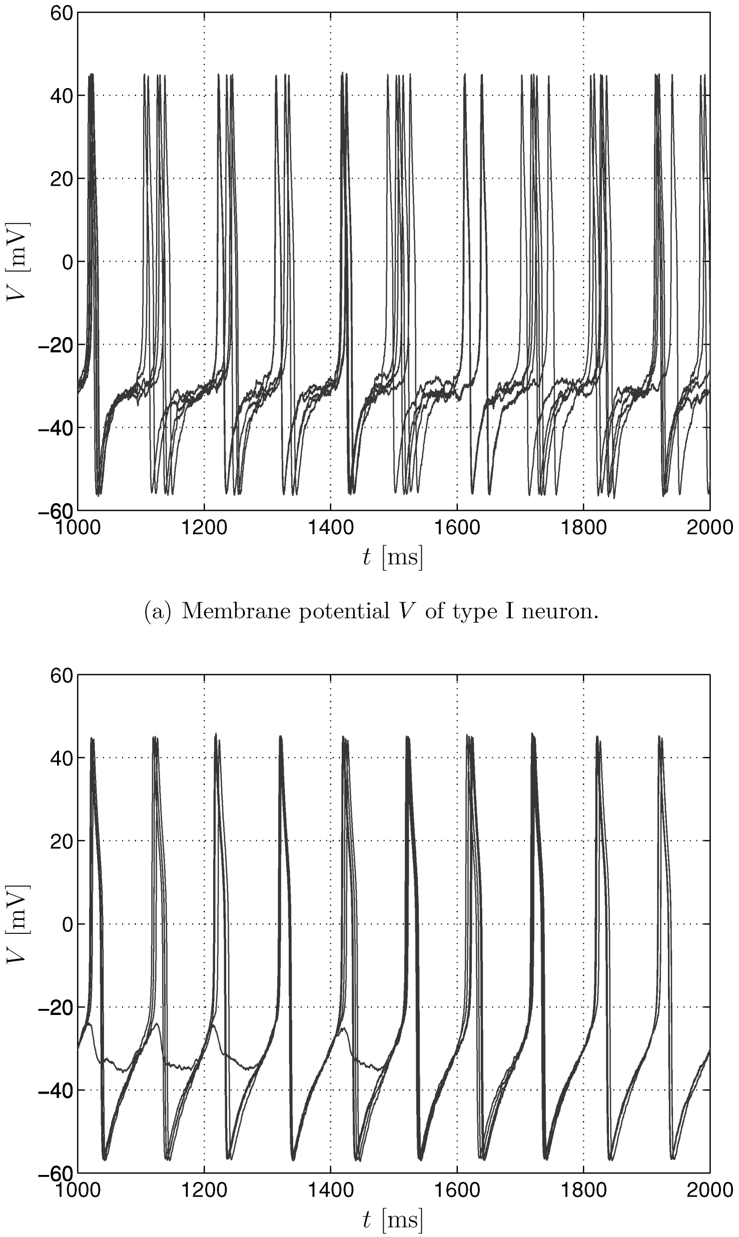
Parameter	Type I	Type II
$A\,[\text{nA/cm}^2]$	40	72
$B\,[\text{nA/cm}^2]$	0.67	6
f [Hz]	10	10
$\sigma\,[\text{nA/cm}^2]$	9	5

those parameters over all pairs; since there are 50 trials, we have 1225 such pairs. A similar approach was followed in Haas and White (2002), Schreiber et al. (2003), and Hunter and Milton (2003). We set $\hat{\delta}^{(0)} = 0$, and in order to overcome local extrema, we use multiple initial values $\hat{s}_t^{(0)} = (1\text{ ms})^2$, $(3\text{ ms})^2$, $(5\text{ ms})^2$, $(7\text{ ms})^2$ and $(9\text{ ms})^2$. Each initialization of $(\hat{\delta}^{(0)}, \hat{s}_t^{(0)})$ may lead to a different solution $(\hat{j}, \hat{j}', \hat{\delta}_t, \hat{s}_t)$. We choose the most probable solution—the one that has the largest value $p(x, x', \hat{j}, \hat{j}', \hat{\delta}_t, \hat{s}_t)$.

Note that instead of considering all 1225 pairs of trials, an arguably more elegant approach would be to consider all 50 trials jointly. As we pointed out earlier, SES can indeed be extended to collections of point processes, but this goes beyond the scope of this letter and the companion letter.

Figures 14a and 14b show how the average s_t (σ_t) and ρ, respectively, depend on β for both neuron types. Figure 14c shows s_t (σ_t) as a function of ρ for several values of β. The "optimal" values of (β, s_t, ρ) are indicated by arrows. Later we will explain how we determined those values. From those three figures, it becomes immediately clear that the parameter ρ is significantly smaller in type I than in type II neurons (for $\beta \in [10^{-10}, 10^{-1}]$). In contrast, s_t is vastly larger. This agrees with our intuition: since in type II neurons, spikes tend to drop out, ρ should be larger. But since the timing dispersion of the spikes in type I is larger, we expect s_t to be larger in those neurons.

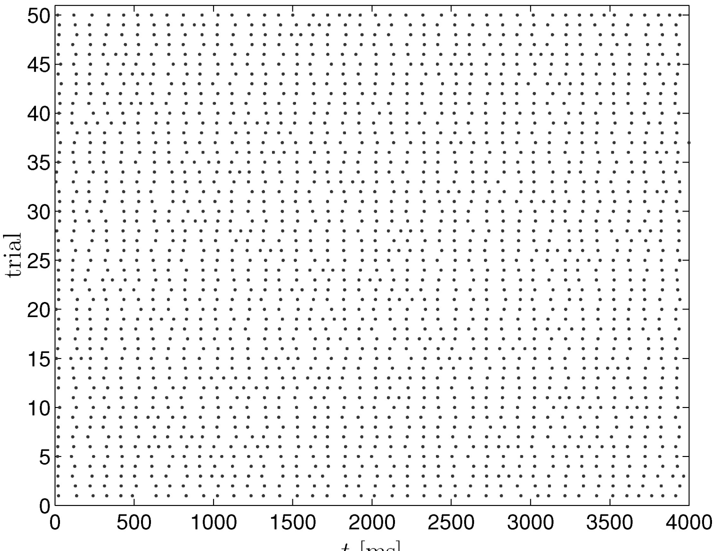
Figures 14a to 14c show the pair (s_t, ρ) for various values of β. Of course, we eventually want to describe the firing reliability by one pair (s_t, ρ), but how should we select β? If we choose β too small, some noncoincident events will be treated as coincident events; they will be matched with other events, resulting in large offsets. As a consequence, the distribution of the offsets will have a significant number of outliers, which leads to an inconsistency: in model $p(x, x', b, b', \delta_t, s_t, \ell)$, equation 3.25, this distribution is supposed to be gaussian, which cannot capture the large number of outliers. In addition, due to those outliers, the parameter s_t will be unreasonably large. As can be seen from Figure 14a, this occurs for type II neurons when $\beta < 10^{-10}$. Figure 14c shows this threshold phenomenon more clearly: there are two distinct regimes in the s_t-ρ curve. This is most obvious for the type II



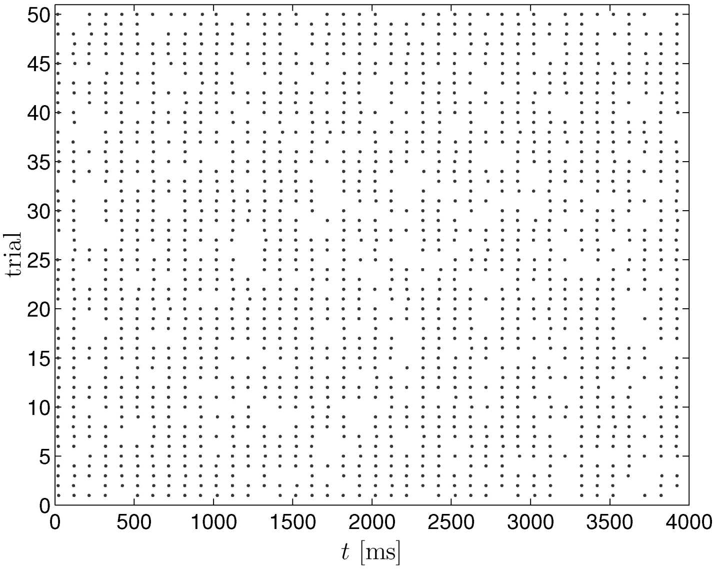
(a) Membrane potential V of type I neuron.

(b) Membrane potential V of type II neuron.

Figure 12: Membrane potential V, equation 7.1, for type I (top) and type II (bottom) neurons. Five realizations are shown.

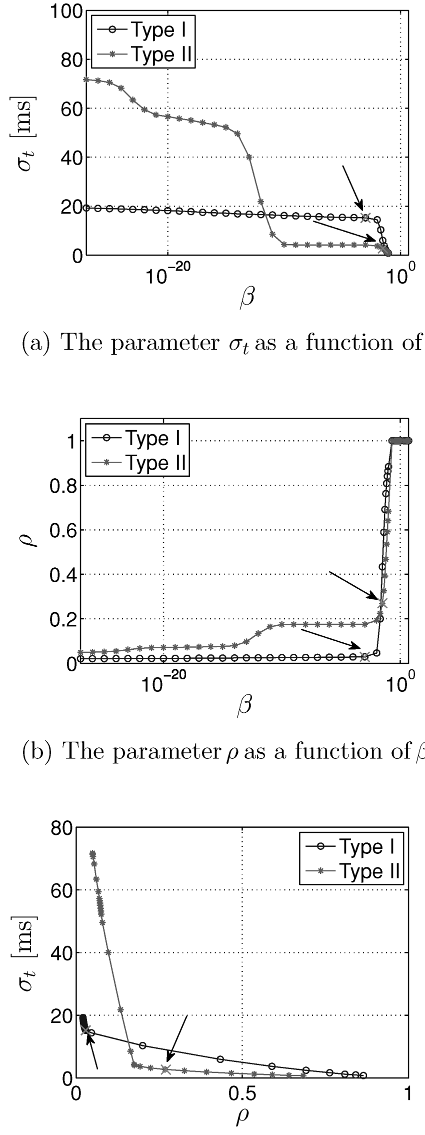


(a) Spike trains from type I neuron.



(b) Spike trains from type II neuron.

Figure 13: Raster plots of spike trains from type I (top) and type II (bottom) neurons. In each case, 50 spike trains are shown.

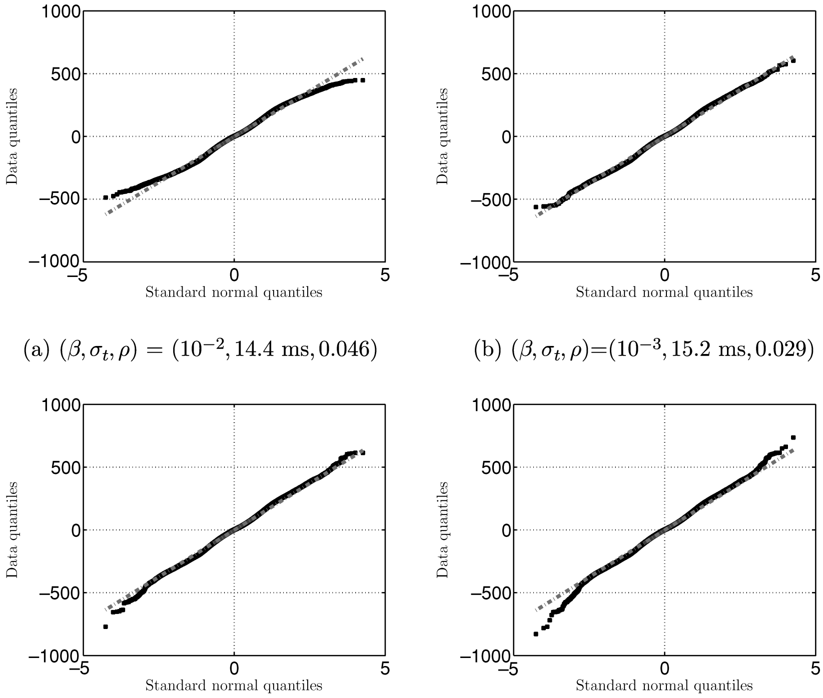


(a) The parameter σ_t as a function of β.

(b) The parameter ρ as a function of β.

(c) The parameter σ_t as a function of ρ.

Figure 14: The parameters σ_t and ρ estimated from spike trains of type I and type II Morris-Lecar neurons (cf. Figure 13). The top and middle panels show how σ_t and ρ, respectively, depend on β. The bottom panel shows how σ_t and ρ jointly evolve with β. The arrows indicate the optimal settings $(\beta, s_t, \rho) = (10^{-3}$, $(15.2 \text{ ms})^2$, 0.029) and $(\beta, s_t, \rho) = (0.03$, $(2.7 \text{ ms})^2$, 0.27) for type I and type II neurons, respectively.



(a) $(\beta, \sigma_t, \rho) = (10^{-2}, 14.4 \text{ ms}, 0.046)$

(b) $(\beta, \sigma_t, \rho) = (10^{-3}, 15.2 \text{ ms}, 0.029)$

(c) $(\beta, \sigma_t, \rho) = (10^{-4}, 15.3 \text{ ms}, 0.028)$

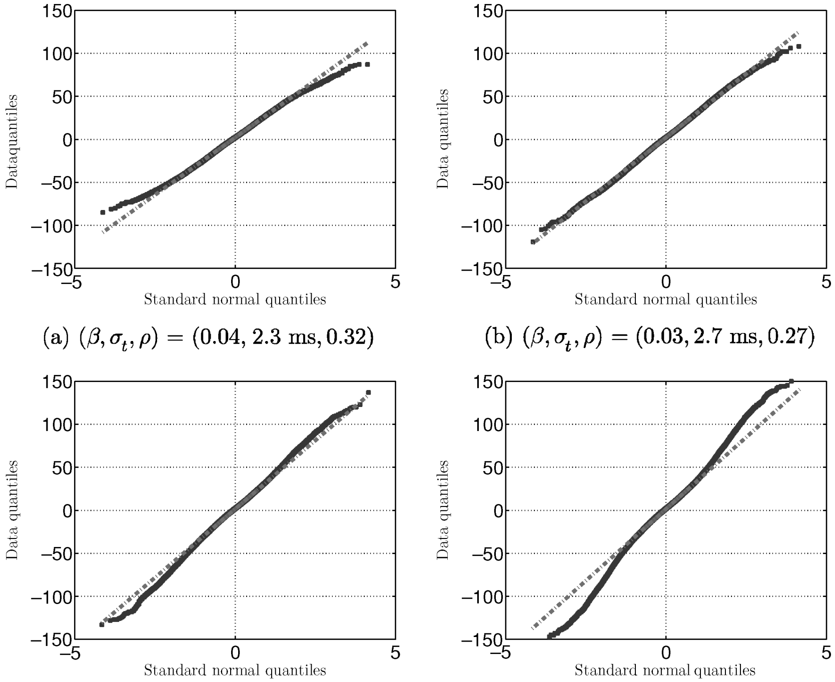
(d) $(\beta, \sigma_t, \rho) = (10^{-5}, 15.4 \text{ ms}, 0.028)$

Figure 15: Quantile-quantile plots for the offset between coincident spikes of the type I neuron. If the (solid) quantile-quantile curve coincides with the (dotted) straight line, the distribution of the offset of coincident events is gaussian. The deviation between both curves is the smallest at $(\beta, s_t, \rho) = (10^{-3}, (15.2 \text{ ms})^2, 0.029)$. They then practically coincide.

neuron, but it also occurs in the type I neuron: the slope of its s_t-ρ curve is larger in the region $\rho < 0.03$ than in the region $\rho > 0.03$.

But if β is too large, some coincident event pairs will no longer be matched, and those events will be treated as noncoincident events. As a result, the distribution of the offsets will have lighter tails than the gaussian distribution; the parameter s_t will then be too small and ρ unreasonably large. This occurs in both neuron types for $\beta > 0.1$ (cf. Figures 14a and 14b).

From this intuitive reasoning, we expect there is an optimal value of β. This is confirmed in Figures 15 and 16, which show quantile-quantile plots of the offset distribution for various values of β. If the offset distribution were exactly gaussian, the data quantiles would lie on the straight dashed lines. One can clearly see deviations from the straight lines for small and large values of β. Figure 17 shows the average deviation from the straight



(a) $(\beta, \sigma_t, \rho) = (0.04, 2.3 \text{ ms}, 0.32)$

(b) $(\beta, \sigma_t, \rho) = (0.03, 2.7 \text{ ms}, 0.27)$

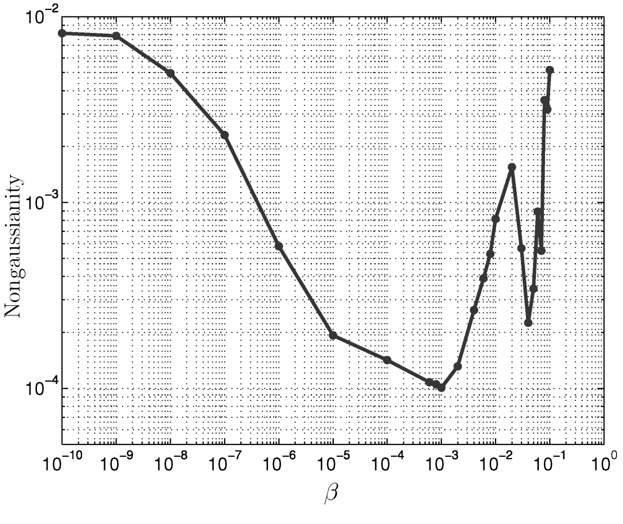
(c) $(\beta, \sigma_t, \rho) = (0.02, 3.2 \text{ ms}, 0.23)$

(d) $(\beta, \sigma_t, \rho) = (0.01, 3.7 \text{ ms}, 0.19)$

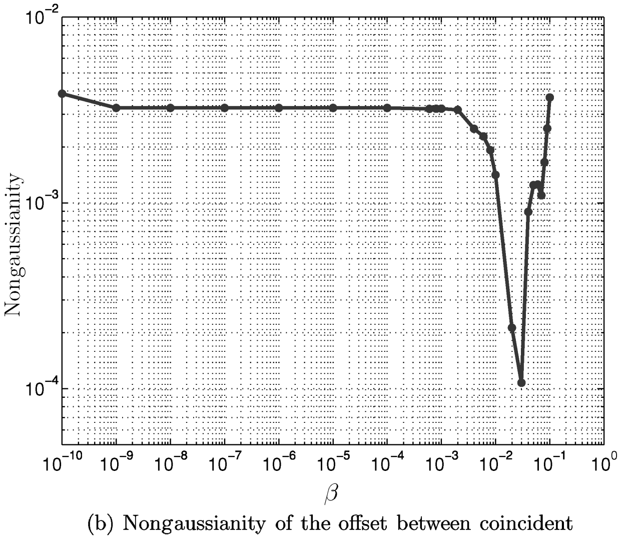
Figure 16: Quantile-quantile plots for the offset between coincident spikes of the type II neuron. If the (solid) quantile-quantile curve coincides with the (dotted) straight line, the distribution of the offset of coincident events is gaussian. The deviation between both curves is the smallest at $(\beta, s_t, \rho) = (0.03, (2.7 \text{ ms})^2, 0.27)$. They then practically coincide.

line as a function of β, which is a measure for how much the offset distribution differs from a gaussian distribution. The value of β with the smallest deviations is 10^{-3} and 0.03 for type I and type II neurons, respectively, which corresponds to $(s_t, \rho) = ((15.2 \text{ ms})^2, 0.029)$ and $(s_t, \rho) = ((2.7 \text{ ms})^2, 0.27)$, respectively. For those values of β, the data quantiles practically coincide with the straight line, and therefore, the offset distribution may be considered gaussian, and model $p(x, x', b, b', \delta_t, s_t, \ell)$, equation 3.25, is then self-consistent.

We also applied this technique for determining β to single pairs of point processes (cf. section 6) but did not obtain satisfactory results. The method needs a sufficient number of (coincident) events in order to be reliable. Therefore, we decided to fix the parameter β in the experiments in section 6 and to optimize over it.



(a) Nongaussianity of the offset between coincident events in type I neurons.



(b) Nongaussianity of the offset between coincident events in type II neurons.

Figure 17: Nongaussianity of the offset between coincident events. This is the deviation of the offset distribution from a gaussian distribution. This quantity is computed as the average distance between the quantile-quantile curve and the straight line shown in the quantile-quantile plots of Figures 15 and 16. The minimum nongaussianity is reached when the distance between both curves is the smallest. This occurs at $\beta = 10^{-3}$ and $\beta = 0.03$ in type I and type II neurons, respectively.

Table 6: Results from the Bootstrapping Analysis of the SES Estimates $(s_t, \rho) = ((15.2\text{ ms})^2, 0.029)$ and $(s_t, \rho) = ((2.7\text{ ms})^2, 0.27)$ for Type I and Type II Neurons, Respectively.

Statistics	Type I	Type II
$\mathrm{E}[s_t]$	15.3	2.70
$\bar{\sigma}[s_t]$	1.8%	1.8%
$\mathrm{E}[\rho]$	0.0283	0.273
$\bar{\sigma}[\rho]$	12%	3.1%

Notes: The table shows the expected values $\mathrm{E}[s_t]$ and $\mathrm{E}[\rho]$, besides the normalized standard deviations $\bar{\sigma}[s_t]$ and $\bar{\sigma}[\rho]$. The expected values practically coincide with the actual estimates, and the normalized standard deviations are small. Therefore, the SES estimates (s_t, ρ) may be considered reliable.

We assessed the estimates of (s_t, ρ) by bootstrapping (Efron & Tibshirani, 1993). More precisely, for both types of neurons, we generated 1000 sets of 50 spike trains. Those sets of spike trains were generated along the lines of the symmetic procedure of Figure 1b. First, we generate a hidden process v with length $\ell = 40/(1 - p_d)$ and equidistant events v_k. Then we generate 50 noisy copies of v by slightly perturbing the timing of the events v_k (with noise variance $s_t/2$) and deleting some of the events (with probability p_d). The delay δ_t was set equal to zero. We carried out this procedure for type I neurons with $(s_t, \rho) = ((15.2\text{ ms})^2, 0.029)$ and type II neurons with $(s_t, \rho) = ((2.7\text{ ms})^2, 0.27)$, which are the estimates obtained by the SES inference procedure. Next we applied the SES algorithm of Table 2 to those sets of point processes; the parameter β was set equal to 10^{-3} and 0.03 for type I and type II neurons, respectively, and we chose the initial values $\delta_t^{(0)} = 0$ ms and $s_t^{(0)} = (30\text{ ms})^2$. The results of this analysis are summarized in Table 6. Since the expected values of s_t and ρ agree very well with the true values and the normalized standard deviations are small (<15%), it is reasonable to believe that the estimates $(s_t, \rho) = ((15.2\text{ ms})^2, 0.029)$ and $(s_t, \rho) = ((2.7\text{ ms})^2, 0.27)$ for type I and type II neurons, respectively, are accurate.

For completeness, we show in Figure 18 a histogram of the number of iterations required for the SES algorithm of Table 2 to converge. In each of those iterations, one updates the sequences (j, j') and the SES parameters (cf. Table 2). The histogram of Figure 18 was computed over all pairs of trials of both types of neurons and for all values of β considered in Figure 14a. From the histogram, we can see that the algorithm converged after at most 19 iterations, and on the average, after about three iterations. We allowed a maximum number of 30 iterations, and therefore, from Figure 18, we can conclude that the algorithm always converged in our experiments.

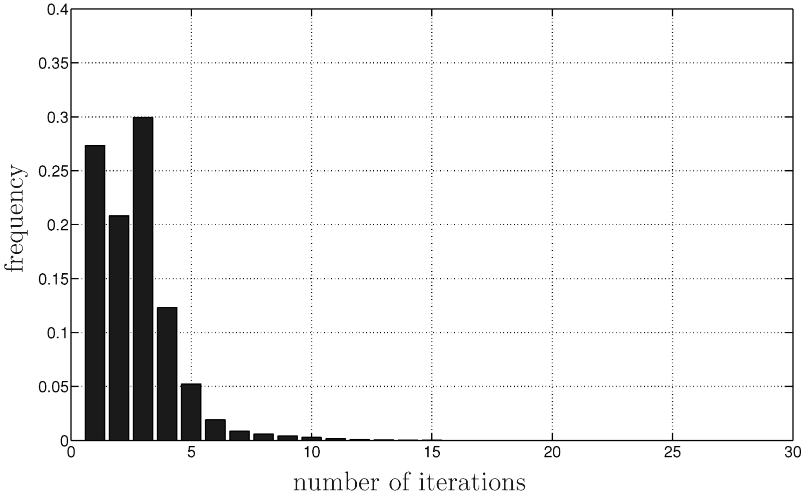
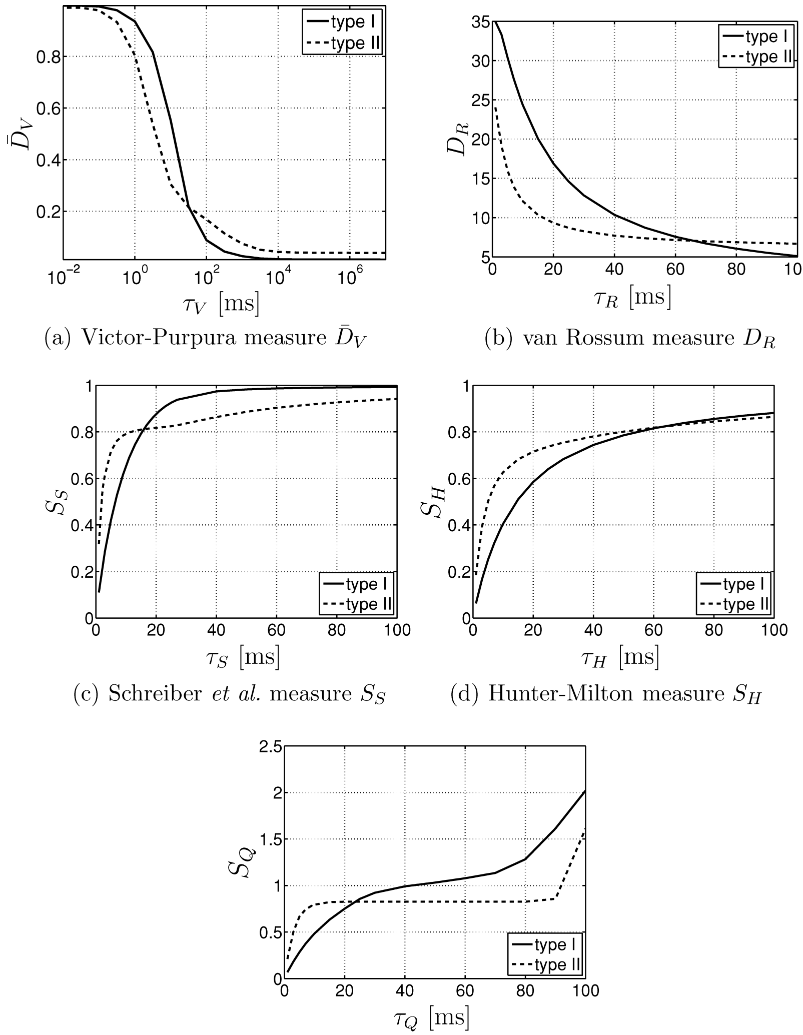


Figure 18: Histogram of the number of iterations required for convergence of the SES inference algorithm of Table 2. In each of those iterations, the sequences (j, j') and the SES parameters are updated (cf. Table 2). The histogram is computed over all pairs of spike trains of both types of neurons (cf. Figure 13) and for all values of β considered in Figure 14a. The maximum number of iterations was set to 30. It can be seen from this histogram that the algorithm converged in all experiments.

7.2.2 Classical Measures. Besides SES, we also applied the classical methods reviewed in section 5. The results are summarized in Figure 19. From the figure, it can be seen that the similarity measures S_S, S_H, and S_Q are larger for type II neurons than for type I neurons if the time constants τ_S, τ_H, and τ_Q are small. For large time constants, the opposite holds. This can be explained as follows. Since the timing dispersion in type I neurons is fairly large, many spikes of type I neurons will be treated as noncoincident (nonoverlapping) if the time constants τ_S, τ_H, and τ_Q are small. If those time constants are large, most spikes of type I neurons will be considered as coincident (overlapping). In contrast, type II neurons have high timing precision, and therefore the similarity measures S_S, S_H, and S_Q grow quickly with the time constants τ_S, τ_H, and τ_Q. However, the measures converge to relatively small values. Due to the large number of dropouts in spike trains of type II neurons, a substantial number of spikes are treated as noncoincident; therefore, as the time constants grow, the similarity measures S_S, S_H, and S_Q attain smaller values than in type I neurons. The results of the (normalized) Victor-Purpura distance metric $\bar{D}_V$ and the van Rossum distance metric D_R can be understood along the same lines.

As we pointed out earlier, SES adjusts its timescale automatically. The same holds for event synchronization (Quian Quiroga, Kreuz, et al., 2002):



(a) Victor-Purpura measure $\bar{D}_V$

(b) van Rossum measure D_R

(c) Schreiber *et al.* measure S_S

(d) Hunter-Milton measure S_H

(e) Event synchronization S_Q

Figure 19: Classical (dis)similarity measures applied to the spike trains of type I and type II Morris-Lecar neurons (cf. Figure 13). The figures show the (normalized) Victor-Purpura distance metric $\bar{D}_V$, the van Rossum distance metric D_R, the Schreiber et al. similarity measure S_S, the Hunter-Milton similarity measure S_H, and the event synchronization measure S_Q as a function of their time constants $\tau_V = 1/C_V$, τ_R, τ_S, τ_H, and τ_Q, respectively. For small values of those constants, the measures indicate that type II neurons fire more synchronously than type I neurons; for larger values of time constants, the opposite holds.

one may adapt the time constant τ_Q according to equation 5.4. With this adaption rule for τ_Q, we obtained $S_Q = 0.96$ for type I neurons and $S_Q = 0.83$ for type II neurons. This can be understood as follows: since the adaptive time constant τ_Q is typically about 50 ms or larger, the value of S_Q is the lowest in type II neurons due to the frequent dropouts in their spike trains.

At last, we consider a classical similarity measure S_{ISI} for multiple point processes, introduced in Tiesinga and Sejnowski (2004) (see also Tiesinga et al., 2008); it is based on interspike intervals (ISI). The first step is to merge the spike times across all trials. Next the interspike intervals of this sequence are calculated, and the coefficient of variation of the aggregated response (CVP) is calculated as the standard deviation of the interspike intervals divided by their mean. The similarity measure S_{ISI} is then eventually obtained by subtracting 1 from the CVP and dividing by the square root of the number of trials. We obtained $S_{\mathrm{ISI}} = 0.25$ and $S_{\mathrm{ISI}} = 0.64$ for type I and type II neurons, respectively. Since S_{ISI} captures mostly the timing precision and is less sensitive to dropouts, we indeed expect that it attains a larger value for type II neurons than for type I neurons.

7.3 Discussion.

This analysis underlines an important issue: most classical measures depend on a time constant, and in some practical situations, it is not obvious how to choose the optimal value of those time constants. Indeed, Figure 19 suggests that one should compute the measures for a range of values of the time constants. As a result, one obtains not just one single measure of similarity but a similarity function $S(\tau)$. Such a function may not always be easy to interpet, compare, or manipulate in practice. Event synchronization and SES are able to automatically determine the appropriate timescale.

However, in some applications, one may wish to investigate how the similarity depends on the timescale. In event synchronization and SES, the timescale can be fixed. Therefore, event synchronization and SES can be computed for a range of timescales.

8 Conclusion

We have presented an alternative method to quantify the similarity of two time series, referred to as stochastic event synchrony (SES). The first step is to extract events from both time series, resulting in two point processes. The events in those point processes are then aligned. The better the alignment, the more similar the original time series are considered to be. In this letter, we focused on one-dimensional point processes.

Obviously it is important to extract meaningful events from the given time series. The proposed method may be expected to produce useful results only if the events characterize the time series in a suitable manner. In the case of spike trains, individual spikes can naturally be considered as events.

Note that for certain neurons, however, it may actually be more appropriate to define a burst of spikes as a single event.

We compared SES to classical (dis)similarity measures for one-dimensional point processes. Through the analysis of surrogate data, we observed that most classical (dis)similarity measures are unable to distinguish timing dispersion from event reliability, they depend on both quantities. In contrast, SES allows quantifying both aspects of synchrony separately. We also wish to reiterate that all (dis)similarity measures, both the classical measures and SES, typically underestimate the timing dispersion and overestimate event reliability; this is due to the ambiguous nature of the synchrony of one-dimensional point processes.

This ambiguity may be resolved by incorporating additional information about the events. For example, in the case of spikes, one may take the shape of the spikes into account. The point processes then become multidimensional. In the companion letter in this issue, we describe how SES may be extended to multidimensional point processes. In that setting, the events pairs are no longer assumed to be ordered, in contrast to the formulation of SES here (see section 3).

Finally, we address an interesting topic for future reseach. The SES parameters are determined by coordinate descent, which is guaranteed to converge to stationary points of the posterior distribution of the SES parameters. However, it does not necessarily converge to the maximum of that distribution, which corresponds to the maximum a posteriori (MAP) estimates of the SES parameters. Instead of trying to obtain the MAP estimates (by coordinate descent or other techniques), one may (approximately) compute the posterior distribution of the SES parameters by means of Monte Carlo algorithms such as Gibbs sampling or Markov chain Monte Carlo methods. From that (approximate) posterior distribution, one may be able to obtain more reliable estimates of the SES parameters. In addition, whereas the proposed approach is mostly practical when the prior for the number of events is a geometric distribution, Monte Carlo methods can easily deal with other priors, such as Poisson distributions. However, such Monte Carlo approaches would be substantially slower than the proposed algorithm based on coordinate descent.

Appendix: Derivation of Inference Algorithm for One-Dimensional SES

Here we derive the algorithm of Table 2. More specifically, we clarify how to carry out the updates 4.2 and 4.3.

We start with update 4.3 since it is the more straightforward. The estimate $\hat{\delta}_t^{(i+1)}$ is the average offset between the coincident events at iteration $i+1$:

$$\hat{\delta}_t^{(i+1)} \triangleq \frac{1}{n^{(i+1)}} \sum_{k=1}^{n^{(i+1)}} \left(\hat{x}_k'^{(i+1)} - \hat{x}_k^{(i+1)}\right), \tag{A.1}$$

where $\hat{x}_k^{(i+1)} \triangleq x_{\hat{j}_k^{(i+1)}}$ is the $\hat{j}_k^{(i+1)}$th event of x, and $\hat{j}^{(i+1)}$ is the estimate of j at iteration $i+1$. Likewise $\hat{x}_k^{\prime(i+1)} \triangleq x'_{\hat{j}_k^{\prime(i+1)}}$ is the $\hat{j}_k^{\prime(i+1)}$th event of x', and $n^{(i+1)}$ is the number of coincident pairs at iteration $i+1$:

$$n^{(i+1)} = n - \sum_{k=1}^{n} \hat{b}_k^{(i+1)} = n' - \sum_{k=1}^{n} \hat{b}_k^{\prime(i+1)}. \tag{A.2}$$

Similarly, the estimate $\hat{s}_t^{(i+1)}$ is the variance of the offset between the coincident events at iteration $i+1$:

$$\hat{s}_t^{(i+1)} \triangleq \frac{1}{n^{(i+1)}} \sum_{k=1}^{n^{(i+1)}} \left(\hat{x}_k^{\prime(i+1)} - \hat{x}_k^{(i+1)} - \hat{\delta}_t^{(i+1)}\right)^2. \tag{A.3}$$

Update 4.2 can readily be carried out by applying the Viterbi algorithm (Forney, 1973) ("dynamic programming") on a trellis with the pairs of coincident events $(x_{j_k}, x'_{j'_k})$ as states, or equivalently, by applying the max-product algorithm on a cycle-free factor graph (Loeliger, 2004; Loeliger et al., 2007) of $p(x, x', j, j', \delta_t, s_t)$. The procedure is equivalent to dynamic time warping (Myers & Rabiner, 1981); it is, for example, used in the context of bio-informatics to compute the distance between genetic sequences (Sellers, 1974, 1979). It is also applied in neuroscience to compute various spike metrics (Victor & Purpura, 1997; Aronov, 2003; Victor et al., 2007).

As a first step in that procedure, one arranges the sequences x and x' on the sides of a $(n+1) \times (n'+1)$ grid (see Figure 5). Note that we assume, without loss of generality, that the sequences x and x' are ordered: $x_k \geq x_{k-1}$ and $x'_k \geq x'_{k-1}$. An alignment $(\hat{j}, \hat{j}')$ corresponds to a path $\mathcal{P} = \{(x_{q_1}, x'_{q'_1}), (x_{q_2}, x'_{q'_2}), \ldots\}$ on the grid; in particular, the alignment 4.2 corresponds to the minimal cost path. Note that each path starts at (0,0) and ends at (n, n'). In addition, it never turns back; in other words, the indices q_k and $q'_{k'}$ never decrease, since the event sequences are assumed to be ordered (cf. section 3). Moreover, those indices increase by at most 1 at each step along the path. As a result, each path contains three kinds of segments, $[(q_{k-1}, q'_{k'-1}), (q_k, q'_{k'})]$, all of length 1:

Horizontal: $(q_k, q'_{k'}) = (q_{k-1}+1, q'_{k'-1})$
Vertical: $(q_k, q'_{k'}) = (q_{k-1}, q'_{k'-1}+1)$
Diagonal: $(q_k, q'_{k'}) = (q_{k-1}+1, q'_{k'-1}+1)$.

The minimal cost path is found by computing an $(n+1) \times (n'+1)$ cost matrix M. The first row and column of M are filled with zeroes—the elements $M_{k,0} = 0 = M_{0,k'}$ (for $k = 0, 1, \ldots, n$ and $k' = 0, 1, \ldots, n'$). The other elements are computed recursively as

$$M_{k,k'} = \min\Big[M_{k-1,k'} + d\big(\hat{s}_t^{(i)}\big), M_{k,k'-1} + d\big(\hat{s}_t^{(i)}\big),$$
$$M_{k-1,k'-1} + d\big(x_k, x'_{k'}; \hat{\delta}_t^{(i)}, \hat{s}_t^{(i)}\big)\Big], \tag{A.4}$$

for $k = 1, \ldots, n$ and $k' = 1, \ldots, n'$. Obviously, in order to compute the cost $M_{k,k'}$, the costs $M_{k-1,k'}$, $M_{k,k'-1}$, and $M_{k-1,k'-1}$ need to have been computed previously. To this end, one may first compute $M_{1,1}$. Then one may gradually fill the rest of the matrix M. The minimal cost is eventually given by $M_{nn'}$. The corresponding path $\mathcal{P}$ and alignment $(\hat{j}, \hat{j}')$ may be traced back from the options chosen at each stage in the recursion (see equation A.4). The first choice corresponds to treating x_k as a noncoincident event ($\hat{b}_k = 1$; horizontal segment), the second choice corresponds to treating $x'_{k'}$ as a noncoincident event ($\hat{b}'_{k'} = 1$; vertical segment), and the third choice corresponds to treating $(x_k, x'_{k'})$ as an event pair ($\hat{b}_k = 0$ and $\hat{b}'_{k'} = 0$; diagonal segment). Combining the updates A.1 and A.3 with the recursion A.4 leads to the algorithm of Table 2.

Note that if the event sequences are not assumed to be ordered, the paths on the grid may return and the minimal-cost path may no longer be found by this simple dynamic programming procedure.

Acknowledgments

Results of this letter were in part reported in Dauwels et al. (2007). The authors thank Zhe ("Sage") Chen (Harvard University), Kenji Morita (RIKEN Brain Science Institute), Yuichi Sakumura (NAIST), Carlos Rodriguez (University at Albany), Sujay Sanghavi (MIT), and Yuki Tsukada (NAIST) for helpful discussions. We are grateful to participants of the Retreat of the MIT Picower Center for Learning and Memory (May 2007), the RIKEN Symposium "Brain Activity and Information Integration" (September 2007), and the NIPS Workshop "Large-Scale Brain Dynamics" (December 2007; Whistler, Canada) for numerous inspiring questions and comments. J.D. is deeply indebted to Shun-ichi Amari (RIKEN Brain Science Institute) and Andi Loeliger (ETH Zurich) for continuing support and encouragement over the last years.

J.D. was supported in part by postdoctoral fellowships from the Japanese Society for the Promotion of Science, the King Baudouin Foundation, and the Belgian American Educational Foundation, T.W. was in part supported by NSF grant DMI-0447766. Part of this work was carried out while J.D. and T.W. were at the RIKEN Brain Science Institute, Saitama, Japan.

References

Abeles, M., Bergman, H., Margalit, E., & Vaadia, E. (1993). Spatiotemporal firing patterns in the frontal cortex of behaving monkeys. *J. Neurophysiol., 70*(4), 1629–1638.

Amari, S., Nakahara, H., Wu, S., & Sakai, Y. (2003). Synchronous firing and higher-order interactions in neuron pool. *Neural Comput., 15*, 127–142.

Aronov, D. (2003). Fast algorithm for the metric-space analysis of simultaneous responses of multiple single neurons. *J. Neuroscience Methods, 124*, 175–179.

Bezdek, J., & Hathaway, R. (2002). Some notes on alternating optimization. In *Proc. AFSS Int. Conference on Fuzzy Systems* (pp. 187–195). Berlin: Springer-Verlag.

Bezdek, J., Hathaway, R., Howard, R., Wilson, C., & Windham, M. (1987). Local convergence analysis of a grouped variable version of coordinate descent. *Journal of Optimization Theory and Applications, 54*(3), 471–477.

Christen, M., Kohn, A., Ott, T., & Stoop, R. (2006). Measuring spike pattern reliability with the Lempel-Ziv distance. *J. Neuroscience Methods, 156*, 342–350.

Crick, F., & Koch, C. (2003). A framework for consciousness. *Nat. Neurosci., 6*(2), 119–126.

Dauwels, J., Tsukada, Y., Sakumura, Y., Ishii, S., Aoki, K., Nakamura, T., et al. (in press). On the synchrony of morphological and molecular signaling events in cell migration. In *Proc. International Conference on Neural Information Processing*. N.p.

Dauwels, J., Vialatte, F., Rutkowski, T., & Cichocki, A. (2007). Measuring neural synchrony by message passing. In J.C. Platt, D. Koller, Y. Singer, & S. Roweis (Eds.), *Advances in neural information processing systems, 20*. Cambridge, MA: MIT Press.

Efron, B., & Tibshirani, R. (1993). *An introduction to the bootstrap*. London: Chapman & Hall/CRC.

Eichhorn, J., Tolias, A., Zien, A., Kuss, M., Rasmussen, C., Weston, J., et al. (2004). Prediction on spike data using kernel algorithms. In S. Thrün, L.K. Saul, & B. Schölkopf (Eds.), *Advances in neural information processing systems, 16*. Cambridge, MA: MIT Press.

Forney, G. D. (1973). The Viterbi algorithm. *Proceedings of the IEEE, 61*(3), 268–278.

Gallager, R. (1996). *Discrete stochastic processes*. Norwell, MA: Kluwer.

Gutkin, B. S., & Ermentrout, G. B. (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.

Haas, J. S., & White, J. A. (2002). Frequency selectivity of layer II stellate cells in the medial entorhinal cortex. *J. Neurophysiology, 88*, 2422–2429.

Hunter, J. D., & Milton, G. (2003). Amplitude and frequency dependence of spike timing: Implications for dynamic regulation. *J. Neurophysiology, 90*, 387–394.

Jeong, J. (2004). EEG dynamics in patients with Alzheimer's disease. *Clinical Neurophysiology, 115*, 1490–1505.

Johnson, D. H., Gruner, C. M., Baggerly, K., & Seshagiri, C. (2001). Information-theoretic analysis of neural coding. *J. Comp. Neuroscience, 10*, 47–69.

Jordan M. I. (Ed.). (1999). *Learning in graphical models*. Cambridge, MA: MIT Press.

Kreuz, T., Haas, J. S., Morelli, A., Abarbanel, H. D. I., & Politi, A. (2007). Measuring spike train synchrony. *Journal of Neuroscience Methods, 165*(1), 151–161.

Loeliger, H.-A. (2004, January). An introduction to factor graphs. *IEEE Signal Processing Magazine*, 28–41.

Loeliger, H.-A., Dauwels, J., Hu, J., Korl, S., Li, Ping, & Kschischang, F. (2007). The factor graph approach to model-based signal processing. *Proceedings of the IEEE, 95*(6), 1295–1322.

Mainen, Z. F., & Sejnowski, T. J. (1995). Reliability of spike timing in neocortical neurons. *Science, 268*(5216), 1503–1506.

Matsuda, H. (2001). Cerebral blood flow and metabolic abnormalities in Alzheimer's disease. *Ann. Nucl. Med., 15*, 85–92.

Morris, C., & Lecar, H. (1981). Voltage oscillations in the barnacle giant muscle fiber. *Biophys. J., 35*, 193–213.

Myers, C. S., & Rabiner, L. R. (1981). A comparative study of several dynamic time-warping algorithms for connected word recognition. *Bell System Technical Journal, 60*(7), 1389–1409.

Pereda, E., Quiroga, R. Q., & Bhattacharya, J. (2005). Nonlinear multivariate analysis of neurophysiological signals. *Progress in Neurobiology, 77*, 1–37.

Quian Quiroga, R. Q., Kraskov, A., Kreuz, T., & Grassberger, P. (2002). Performance of different synchronization measures in real data: A case study on EEG signals. *Physical Review E 65*, 041903.1–041903.14.

Quian Quiroga, R. Q., Kreuz, T., & Grassberger, P. (2002). Event synchronization: A simple and fast method to measure synchronicity and time delay patterns. *Physical Review E 66*, 041904.1–041904.9.

Robinson, H. P. C. (2003). The biophysical basis of firing variability in cortical neurons. In J. Feng (Ed.), *Computational neuroscience: A comprehensive approach*. London: Chapman & Hall/CRC.

Schrauwen, B., & Van Campenhout, J. (2007). Linking non-binned spike train kernels to several existing spike train metrics. *Neurocomputing, 70*(7–9), 1247–1253.

Schreiber, S., Fellous, J. M., Whitmer, J. H., Tiesinga, P. H. E., & Sejnowski, T. J. (2003). A new correlation-based measure of spike timing reliability. *Neurocomputing 52*, 925–931.

Sellers, P. H. (1974). On the theory and computation of evolutionary distances. *SIAM J. Appl. Math., 26*, 787–793.

Sellers, P. H. (1979). *Combinatorial complexes*. Dordrecht: D. Reidel.

Shpigelman, L., Singer, Y., Paz, R., & Vaadia, E. (2005). Spikernels: Predicting arm movements by embedding population spike rate patterns in inner-product spaces. *Neural Comput., 17*(3), 671–690.

Singer, W. (2001). Consciousness and the binding problem, 2001. *Annals of the New York Academy of Sciences, 929*, 123–146.

Stam, C. J. (2005). Nonlinear dynamical analysis of EEG and MEG: Review of an emerging field. *Clinical Neurophysiology, 116*, 2266–2301.

Tateno, T., & Pakdaman, K. (2004). Random dynamics of the Morris-Lecar neural model. *Chaos, 14*(3), 511–530.

Tiesinga, P., Fellous, J.-M., & Sejnowski, T. J. (2008). Regulation of spike timing in visual cortical circuits. *Nature Reviews Neuroscience, 9*, 97–107.

Tiesinga, P., & Sejnowski, T. J. (2004). Rapid temporal modulation of synchrony by competition in cortical interneuron networks. *Neural Comput., 16*, 251–275.

Tsumoto, K., Kitajima, H., Yoshinaga, T., Aihara, K., & Kawakami, H. (2006). Bifurcations in Morris-Lecar neuron model. *Neurocomputing, 69*, 293–316.

van Rossum, M. C. W. (2001). A novel spike distance. *Neural Comput., 13*, 751–763.

Varela, F., Lachaux, J. P., Rodriguez, E., & Martinerie, J. (2001). The brainweb: Phase synchronization and large-scale integration. *Nature Reviews Neuroscience, 2*(4), 229–239.

Victor, J. D., Goldberg, D., & Gardner, D. (2007). Dynamic programming algorithms for comparing multineuronal spike trains via cost-based metrics and alignments. *J. Neurosci. Meth., 161*, 351–360.

Victor, J. D. & Purpura, K. P. (1997). Metric-space analysis of spike trains: Theory, algorithms, & application. *Network: Comput. Neural Systems, 8*(17), 127–164.

von der Malsburg, C. (1981). The correlation theory of brain function. In E. Domany, J. L. van Hemmen, & K. Schulten (Eds.), *Models of neural networks II* (pp. 95–119). Berlin: Springer.

Zolotarev, V. M. (1986). *One-dimensional stable distributions*. Providence, RI: American Mathematical Society.

Received April 3, 2008; accepted December 9, 2008.