Quantifying Statistical Interdependence by Message Passing on Graphs PART II: Multi-Dimensional Point Processes

J. Dauwels [a,b,*,1] F. Vialatte [c] T. Weber [d] T. Musha [e] A. Cichocki [c]

[a] *Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA.*

[b] *Amari Research Unit, RIKEN Brain Science Institute, Saitama, Japan.*

[c] *Laboratory for Advanced Brain Signal Processing, RIKEN Brain Science Institute, Saitama, Japan.*

[d] *Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA.*

[e] *Brain Functions Laboratory, Inc., Yokohama, Japan.*

Abstract

Stochastic event synchrony is a technique to quantify the similarity of pairs of signals. First, "events" are extracted from the two given time series. Next, one tries 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. In Part I, one-dimensional events are considered, this paper (Paper II) concerns multi-dimensional events. Although the basic idea is similar, the extension to multi-dimensional point processes involves a significantly harder combinatorial problem, and therefore, it is non-trivial.

Also in the multi-dimensional, the problem of jointly computing the pairwise alignment and SES parameters is cast as a statistical inference problem. This problem is solved by coordinate descent, more specifically, by alternating the following two steps: (i) one estimates the SES parameters from a given pairwise alignment; (ii) with the resulting estimates, one refines the pairwise alignment. The SES parameters are computed by maximum a posteriori (MAP) estimation (Step 1), in analogy to the one-dimensional case. The pairwise alignment (Step 2) can no longer be obtained through dynamic programming, since the state space becomes too large. Instead it is determined by applying the max-product algorithm on a cyclic graphical model.

The method is first applied to surrogate data in order to test its robustness and reliability. Next it is applied to detect anomalies in EEG synchrony of Mild Cognitive Impairment (MCI) patients. Numerical results suggest that SES is significantly more sensitive to perturbations in EEG synchrony than a large variety of classical synchrony measures.

Preprint submitted to Neural Computation *4 December 2008*

Key words: timing precision, event reliability, stochastic event synchrony, graphical model, max-product algorithm, maximum a posteriori estimation, Morlet wavelet, time-frequency map, bump model, EEG, Alzheimer's disease, mild cognitive impairment

1 Introduction

In the last years, the problem of detecting correlations between neural signals has attracted quite some attention in the neuroscience community. Several studies have related neural synchrony to attention and cognition (e.g., (Buzsáki, 2006)); recently, it has been demonstrated that patterns of neural synchronization flexibly trigger patterns of neural interactions (Womelsdorf *et al.*, 2007). Moreover, it is has frequently been reported that abnormalities in neural synchrony lie at the heart of a variety of brain disorders such as Alzheimer's and Parkinson's disease (e.g., (Matsuda *et al.*, 2001; Jeong, 2004; Uhlhaas *et al.*, 2006)). In response to those findings, quite some efforts have been made to develop novel quantitative methods to detect statistical dependencies in brain signals (see, e.g., (Stam, 2005; Quiroga *et al.*, 2002; Pereda *et al.*, 2005)).

In this paper, we extend stochastic event synchrony (SES) from one-dimensional point processes (Part I) to multi-dimensional processes (Part II). The underlying principle is identical, but the inference algorithm to compute the SES parameters is fundamentally different. The basic idea is again the following: First, we extract "events" from the two given time series. Next, we 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 "non-coincident" events, and average similarity of the aligned events. In this paper, we mostly focus on point processes in time-frequency domain. The average event similarity is in that case described by two parameters: the average frequency offset between events in the time-frequency plane and the variance of the frequency offset ("frequency

* Corresponding author.
Email addresses: `justin@dauwels.com` (J. Dauwels), `fvialatte@brain.riken.jp` (F. Vialatte), `theo_w@mit.edu` (T. Weber), `musha@bfl.co.jp` (T. Musha), `cia@brain.riken.jp` (A. Cichocki).

[1] J.D. was in part supported by post-doctoral fellowships from the Japanese Society for the Promotion of Science (JSPS), the King Baudouin Foundation, and the Belgian American Educational Foundation (BAEF). Part of this work was carried out while J.D. and T.W. were at the RIKEN Brain Science Institute, Saitama, Japan.

jitter"). SES then consists of five parameters in total. Those parameters quantify the synchrony of oscillatory events, and provide an alternative to classical synchrony measures that quantify amplitude or phase synchrony.

The pairwise alignment of point processes is again cast as a statistical inference problem. However, inference in that model cannot be carried out by dynamic programming, since the state space is too large. Instead we apply the max-product algorithm on a cyclic graphical model (Jordan, 1999; Loeliger, 2004; Loeliger *et al.*, 2007); the inference method is now an iterative algorithm. Based on a result in (Bayati *et al.*, 2005) (generalized in (Sanghavi, 2007a,b)), we will show that this algorithm yields the optimal alignment as long as the optimal alignment is unique.

In this paper, we only consider *pairs* of point processes, but the methods may be extended to multiple point processes. That extension, however, is non-trivial and goes beyond the scope of this paper; it will be described in a future report.

As in the one-dimensional case, the method may be applied to any kind of time series (e.g., from finance, oceanography, and seismology). However, we will here only consider EEG signals. More specifically, we will present promising results on the early prediction of Alzheimer's disease based on electroencephalograms (EEG).

This paper is organized as follows. In the next section, we introduce SES for multi-dimensional point processes; we describe the underlying statistical model in Section 3. Inference in that model is carried out by applying the max-product algorithm on a factor graph of that model. That factor graph is discussed in Section 4; the inference method is outlined in Section 5 and derived in detail in Appendix C. In Section 6, we list several extensions of the basic multi-dimensional SES model. In Section 7, we investigate the robustness and reliability of the SES inference method by means of surrogate data. In Section 8, we apply that method to detect abnormalities in the EEG synchrony of MCI disease patients. We offer some concluding remarks in Section 9.

2 Principle

Suppose that we are given a pair of continuous-time signals, e.g., EEG signals recorded from two different channels, and we wish to determine the similarity of those two signals. As a first step, we extract point processes from those signals, which may be achieved in various ways. As an example, we generate point processes in time-frequency domain: first the time-frequency ("wavelet") transform of each signal is computed in a frequency band $f \in [f_{\min}, f_{\max}]$.

Next those maps are approximated as a sum of half-ellipsoid basis functions, referred to as "bumps" (see Fig. 1; we will provide more details on bump modeling in Section 8.2.3). Each bump is described by five parameters: time t, frequency f, width Δt, height Δf, and amplitude w. The resulting bump models $e = ((t_1, f_1, \Delta t_1, \Delta f_1, w_1), \ldots, (t_n, f_n, \Delta t_n, \Delta f_n, w_n))$ and $e' = ((t'_1, f'_1, \Delta t'_1, \Delta f'_1, w'_1), \ldots, (t'_{n'}, f'_{n'}, \Delta t'_{n'}, \Delta f'_{n'}, w'_{n'}))$ represent the most prominent oscillatory activity in the signals at hand. This activity may correspond to various physical or biological phenomena, for example:

- oscillatory events in EEG and other brain signals are believed to occur when assemblies of neurons are spiking in synchrony (Buzsáki, 2006; Nunez *et al.*, 2006),
- oscillatory events in calcium imaging data are due to oscillations of intracellular calcium, which are believed to play an important role in signal transduction between cells (see, e.g., (Völkers *et al.*, 2006)),
- oscillations and waves are of central interest in several fields beyond neuroscience, such as oceanography (e.g., oceanic "normal modes" caused by convection (Kantha *et al.*, 2006)) and seismography (e.g., free earth oscillations and earth oscillations induced by earthquakes, hurricanes, and human activity (Alder *et al.*, 1972)).

In the following, we will develop SES for bump models. In this setting, SES quantifies the synchronous interplay between oscillatory patterns in two given signals, while it ignores the other components in those signals ("background activity"). In contrast, classical synchrony measures such as amplitude or phase synchrony are computed from the entire signal, they make no distinction between oscillatory components and the background activity. As a consequence, SES captures alternative aspects of similarity, and hence, it provides complementary information about synchrony.

Besides bump models, SES may be applied to other sparse representations of signals, for example:

- matching pursuit (Mallat *et al.*, 1993) and refinements such as orthogonal matching pursuit (Tropp *et al.*, 2005), stage-wise orthogonal matching pursuit (Donoho *et al.*, 2006), tree matching pursuit (Duarte *et al.*, 2005) and chaining pursuit (Gilbert *et al.*, 2006),
- chirplets (see, e.g., (O'Neill *et al.*, 2000; Cui *et al.*, 2007, 2005)),
- wave atoms (Demanet *et al.*, 2007),
- curvelets (Candès *et al.*, 2002),
- sparsification by loopy belief propagation (Sarvotham *et al.*, 2006),
- the Hilbert-Huang transform (Huang *et al.*, 1998),
- compressed sensing (Candès *et al.*, 2006; Donoho, 2006).

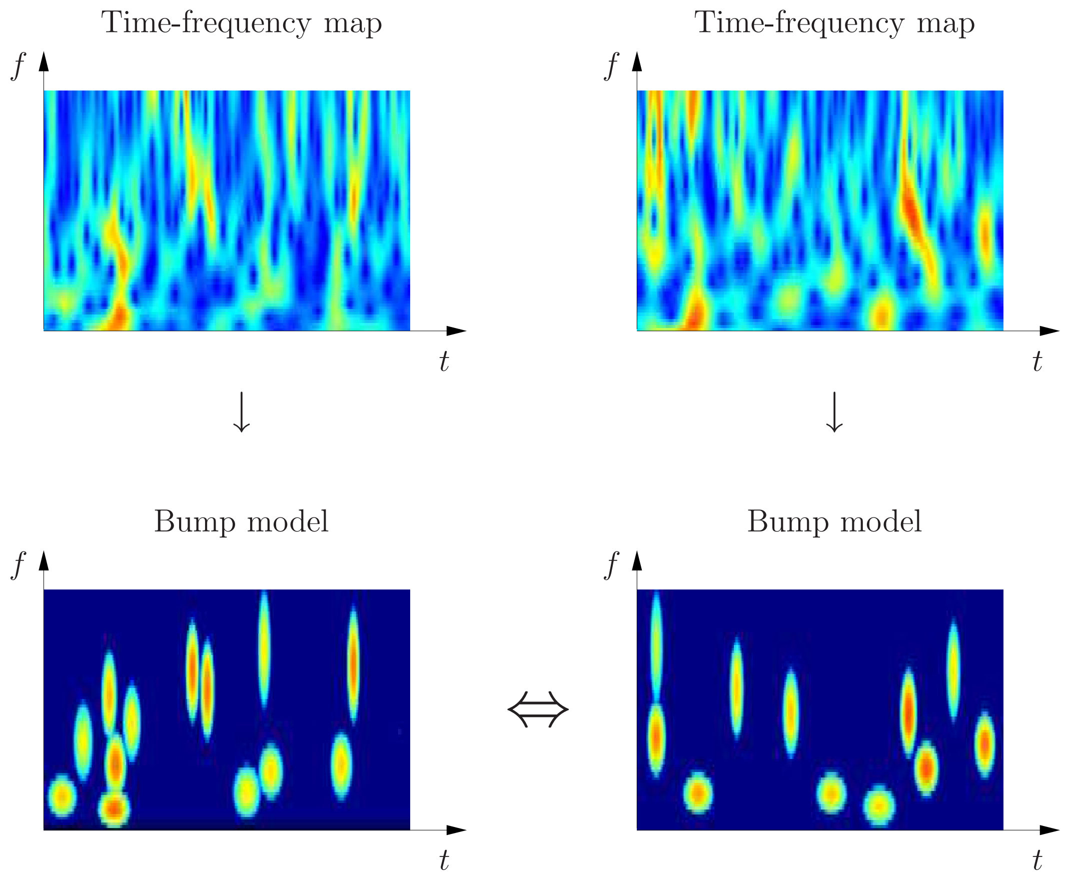
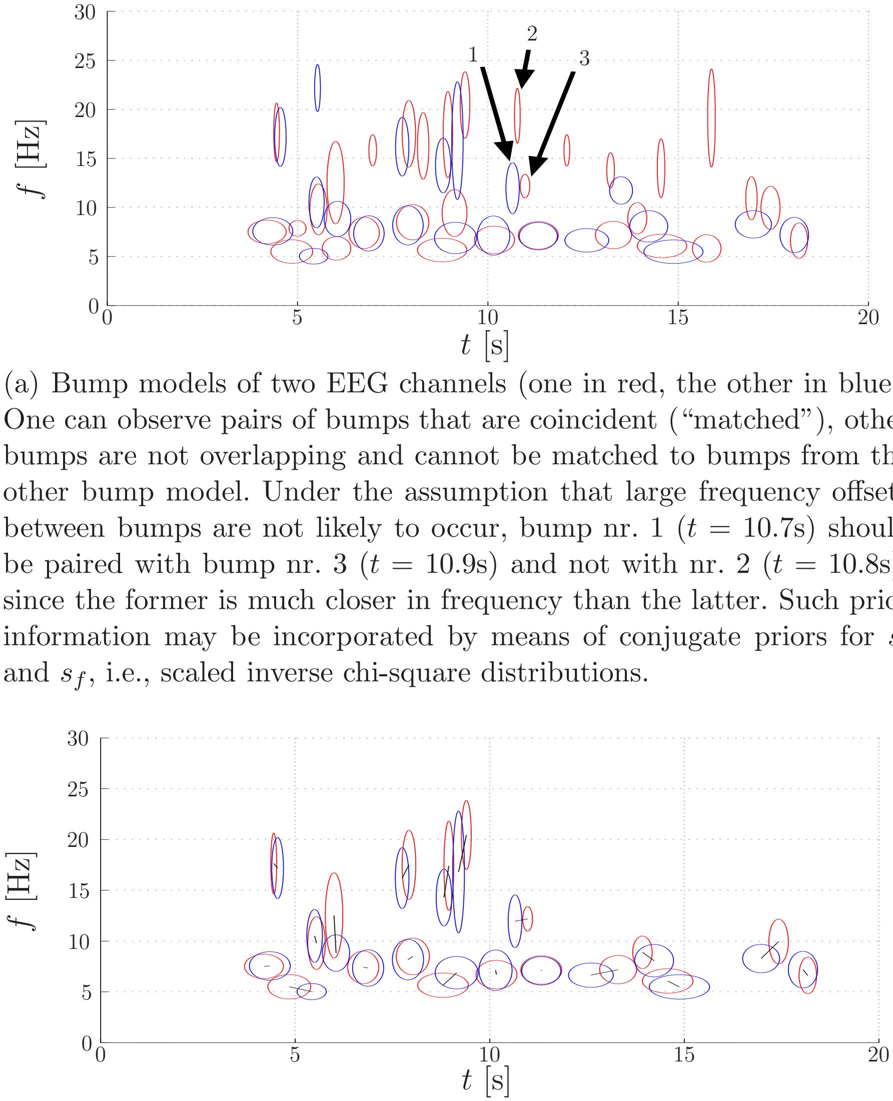


Fig. 1. Two-dimensional stochastic event synchrony. Top: two given EEG signals in time-frequency domain; Bottom: bump models extracted from those time-frequency maps. Stochastic event synchrony quantifies the similarity of two such bump models.

Moreover, the point processes may be defined in other spaces than the time-frequency plane, for example, they may occur in two-dimensional space (e.g., images), space-frequency (e.g., wavelet image coding) or space-time (e.g., movies); they may also be defined on more complicated manifolds, such as curves, surfaces, etc. Such extensions may straightforwardly be derived from the example of bump models. We consider several extensions in Section 6.

Our extension of stochastic event synchrony to multi-dimensional point processes (and bump models in particular) is derived from the following observation (see Fig. 2(a)): bumps in one time-frequency map may not be present in the other map ("non-coincident" bumps); other bumps are present in both maps ("coincident bumps"), but appear at slightly different positions on the maps. The black lines in Fig. 2(b) connect the centers of coincident bumps, and hence, visualize the offsets between pairs of coincident bumps.

Such offsets jeopardize the suitability of classical similarity measures for time-frequency maps. For example, let us consider the Pearson correlation coeffi-



(a) Bump models of two EEG channels (one in red, the other in blue). One can observe pairs of bumps that are coincident ("matched"), other bumps are not overlapping and cannot be matched to bumps from the other bump model. Under the assumption that large frequency offsets between bumps are not likely to occur, bump nr. 1 ($t = 10.7$s) should be paired with bump nr. 3 ($t = 10.9$s) and not with nr. 2 ($t = 10.8$s), since the former is much closer in frequency than the latter. Such prior information may be incorporated by means of conjugate priors for s_t and s_f, i.e., scaled inverse chi-square distributions.

(b) Coincident bumps ($\rho = 27\%$); the black lines connect the centers of coincident bumps.

Fig. 2. Coincident and non-coincident activity.

cient r between two time-frequency maps $x_1(t, f)$ and $x_2(t, f)$:

$$r = \frac{\sum_{t,f}(x_1(t,f) - \bar{x}_1)(x_2(t,f) - \bar{x}_2)}{\sqrt{\sum_{t,f}(x_1(t,f) - \bar{x}_1)^2}\sqrt{\sum_{t,f}(x_2(t,f) - \bar{x}_2)^2}}, \tag{1}$$

where $\bar{x}_i = \sum_{t,f} x_i(t, f)$ $(i = 1, 2)$. Note that r, like many other classical similarity measures, is based on pointwise comparisons, in other words, it compares the activity at instance (t, f) in map x_1 to the activity in x_2 at the *same* instance (t, f). Therefore, if the correlated activity in the maps $x_1(t, f)$ and $x_2(t, f)$ is slightly delayed or a little shifted in frequency, the correlation coefficient r will be small, and as a result, it may not be able to capture the correlated activity. Our approach alleviates this shortcoming, since it explicitly

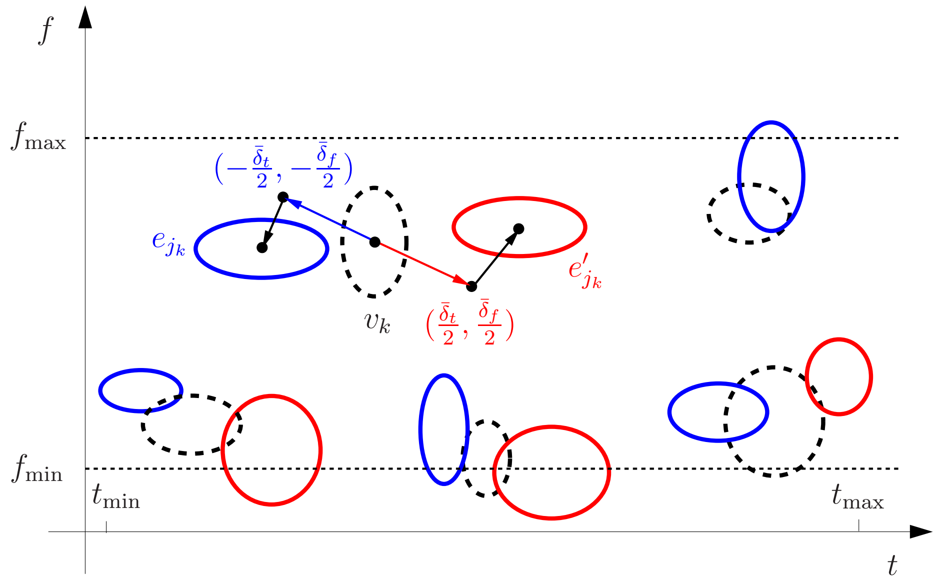

f
$f_{\max}$
$(-\frac{\bar{\delta}_t}{2}, -\frac{\bar{\delta}_f}{2})$
e_{j_k}
v_k
e'_{j_k}
$(\frac{\bar{\delta}_t}{2}, \frac{\bar{\delta}_f}{2})$
$f_{\min}$
$t_{\min}$
$t_{\max}$
t

Fig. 3. Generative model for e and e'. One first generates a hidden process v, next one makes two identical copies of v and shifts those over $(-\delta_t/2, -\delta_f/2)$ and $(\delta_t/2, \delta_f/2)$ respectively; the events of the resulting point process are slightly shifted (with variance (s_t, s_f)), and some of those events are deleted (with probability p_d), resulting in e and e'.

handles delays and frequency offsets.

We quantify the interdependence between two bump models by five parameters, i.e., the parameters ρ, δ_t, and s_t introduced in Part I, in addition to:

- δ_f: the average frequency offset between coincident bumps,
- s_f: the variance of the frequency offset between coincident bumps.

We determine those 5 parameters and the pairwise alignment of e and e' by *statistical inference*, as in the one-dimensional case (cf. Section 3 and 4 in Part I). We start by constructing a statistical model that captures the relation between the two bump models e and e'; that model contains the 5 SES parameters, besides variables related to the pairwise alignment of the bumps of e and e'. Next we perform inference in that model, resulting in estimates for the SES parameters and the pairwise alignment. More concretely, we apply coordinate descent, as in the case of one-dimensional point processes. In the following section, we outline our statistical model. In Section 4, we describe the factor graph of that model. From that factor graph, we derive the inference algorithm for multi-dimensional SES; in Section 5, we outline that inference algorithm. We refer to Appendix C for the detailed derivations. In Section 6, we suggest various extensions of our statistical model.

Symbol	Explanation
e and e'	the two given bump models
t and t'	occurrence time of the bumps of e and e'
f and f'	frequencies of the bumps of e and e'
Δt and $\Delta t'$	width of the bumps of e and e'
Δf and $\Delta f'$	height of the bumps of e and e'
v	hidden bump model from which the observed bump models e and e' are generated
$\tilde{e}$ and $\tilde{e}'$	bump models obtained by shifting v over ($\delta_t/2$, $\delta_f/2$) and (-$\delta_t/2$, -$\delta_f/2$) resp. and randomly perturbing the timing and frequency of the resulting sequences (with variance $s_t/2$ and $s_f/2$ resp.)
b and b'	binary sequences that indicate whether bumps in e and e' resp. are coincident or not
c	binary sequence that indicates whether a particular bump in e is coincident with a particular bump in e', more precisely, $c_{kk'} = 1$ iff e_k is coincident with $e'_{k'}$ and is zero otherwise
j and j'	indices of the coincident bumps in e and e' resp.
n and n'	length of e and e' resp.
ℓ	length of v
δ_t and δ_f	timing and frequency offset resp. between e and e'
s_t and s_f	timing and frequency jitter resp. between e and e'

Table 1
List of variables and parameters associated with models $p(e, e', j, j', \theta)$ (8) and $p(e, e', b, b', c, \theta)$ (16).

3 Statistical Model

In this section, we explain the statistical model that forms the foundation of multivariate SES. For reasons that will be explained in the following, we will represent the model in two different ways (see (8) and (16)). For the sake of clarity, we listed in Table 1 the most important variables and parameters that appear in those representations. We will use the notation $\theta = (\delta_t, s_t, \delta_f, s_f)$.

Fig. 3 illustrates how we extend the generative procedure underlying one-dimensional SES (cf. Part I) to the time-frequency domain. As a first step, one again generates a hidden point process v (dotted bumps in Fig. 3). The number ℓ of bumps v_k is also now described by a geometric prior, in particular:

$$p(\ell) = (1 - \tilde{\lambda})\tilde{\lambda}^{\ell}, \tag{2}$$

with $\tilde{\lambda} = \lambda(t_{\max} - t_{\min})(f_{\max} - f_{\min}) \in (0, 1)$. (We motivate this choice of prior in Part I.) The centers $(\tilde{t}_k, \tilde{f}_k)$ of those bumps are placed uniformly within the rectangle $[t_{\min}, t_{\max}] \times [f_{\min}, f_{\max}]$, and as a consequence:

$$p\Big(\tilde{t}, \tilde{f}|\ell\Big) = \frac{1}{(t_{\max} - t_{\min})^{\ell}(f_{\max} - f_{\min})^{\ell}}. \tag{3}$$

The amplitudes, widths and heights of the bumps v_k are independently and identically distributed according to priors p_w, $p_{\Delta t}$ and $p_{\Delta f}$ respectively. Next, from bump model v, one generates the bump models e and e' as follows:

(1) One starts by making two copies $\tilde{e}$ and $\tilde{e}'$ of bump model v,

(2) One generates new amplitudes w_k, widths Δt_k, and heights Δf_k for the bumps $\tilde{e}_k$ by drawing (independent) samples from the priors p_w, $p_{\Delta t}$ and $p_{\Delta f}$ respectively. Likewise, one generates new amplitudes w'_k and widths $\Delta t'_k$ and $\Delta f'_k$ for the bumps $\tilde{e}'_k$,

(3) One shifts the bumps $\tilde{e}_k$ and $\tilde{e}'_k$ over $(-\frac{\bar{\delta}_t}{2}, -\frac{\bar{\delta}_f}{2})$ and $(\frac{\bar{\delta}_t}{2}, \frac{\bar{\delta}_f}{2})$ respectively with:

$$\bar{\delta}_t = \delta_t\,(\Delta t_k + \Delta t'_k), \tag{4}$$

$$\bar{\delta}_f = \delta_f\,(\Delta f_k + \Delta f'_k). \tag{5}$$

(4) Next, one adds small random perturbations to the position of the bumps $\tilde{e}_k$ and $\tilde{e}'_k$ (cf. Fig. 3), modeled as zero-mean Gaussian random vectors with diagonal covariance matrix $\mathrm{diag}(\frac{\bar{s}_t}{2}, \frac{\bar{s}_f}{2})$:

$$\bar{s}_t = s_t\,(\Delta t_k + \Delta t'_k)^2, \tag{6}$$

$$\bar{s}_f = s_f\,(\Delta f_k + \Delta f'_k)^2. \tag{7}$$

(5) At last, one randomly removes bumps from $\tilde{e}$ and $\tilde{e}'$: each bump is deleted with probability p_d independently of the other bumps, resulting in the bump models e and e'.

As in the one-dimensional case, the above generative procedure (cf. Fig. 3) may straightforwardly be extended from a pair of point processes e and e' to a collection of point processes, but inference in the resulting probabilistic model is intractable; we will present approximate inference algorithms in a future report.

Also in the multi-dimensional case, event synchrony is inherently ambiguous; as an illustration, Fig. 4 shows two procedures to generate the same point processes e and e'. If s_t is large, with high probability events in e and e' will not be ordered in time; for example, the events (1,2) and (3',4') in Fig. 4(a) are reversed in time. Ignoring this fact will result in estimates of s_t that are smaller than the true value s_t. The SES algorithm will most probably correctly infer the coincident event pairs (3,3') and (4,4'), since those pairs are far apart in frequency, and therefore, event 3 is closer to event 3' than it is to event 4'. However, it will most probably treat (1',2) and (1,2') as coincident event pairs instead of (1,2) and (1',2') (see Fig. 4(b)), since event 1 is much closer to event 2' than event 2. As a consequence, SES will underestimate s_t. However, the bias will be smaller than in the one-dimensional case: the SES algorithm will only incorrectly align pairs of events if those pairs of events have about the same frequency (as events 1, 1', 2, and 2' in Fig. 4); if those events are far apart in frequency (as events 3, 3', 4, and 4' in Fig. 4), potential time reversals will be correctly inferred. This observations obviously carries over to any other kind of multi-dimensional point processes.

Besides timing reversals, some event deletions may be ignored: in Fig. 4(a) events 5 and 6 are non-coincident, however, in the procedure of Fig. 4(b) they are both coincident. The latter generative procedure is simpler in the sense that it involves less deletions and the perturbations are slightly smaller. As a result, the parameter ρ (and hence also p_d) is generally underestimated. However, also this bias will be smaller than in the one-dimensional case. Indeed, if the events 5 and 6 had strongly different frequencies, the SES algorithm would probably not treat them as a coincident pair. Obviously, also this observation extends to any kind of multi-dimensional point processes.

The generative procedure of Fig. 3 leads to the two-dimensional extension of the one-dimensional SES model (cf. (26) in Part I):

$$\begin{aligned} p(e, e', j, j', \theta) = \gamma\, \beta^{n^{\text{tot}}_{\text{non-co}}} p(\delta_t) p(s_t) p(\delta_f) p(s_f) \prod_{k=1}^{n^{\text{tot}}_{\text{co}}} & p_w(w_{j_k}) p_w(w'_{j'_k}) \\ & \cdot p_{\Delta t}(\Delta t_{j_k}) p_{\Delta t}(\Delta t'_{j'_k}) p_{\Delta f}(\Delta f_{j_k}) p_{\Delta f}(\Delta f'_{j'_k}) \\ & \cdot \mathcal{N}\Big(t'_{j'_k} - t_{j_k}; \bar{\delta}_t, \bar{s}_t\Big)\, \mathcal{N}\Big(f'_{j'_k} - f_{j_k}; \bar{\delta}_f, \bar{s}_f\Big), \end{aligned} \tag{8}$$

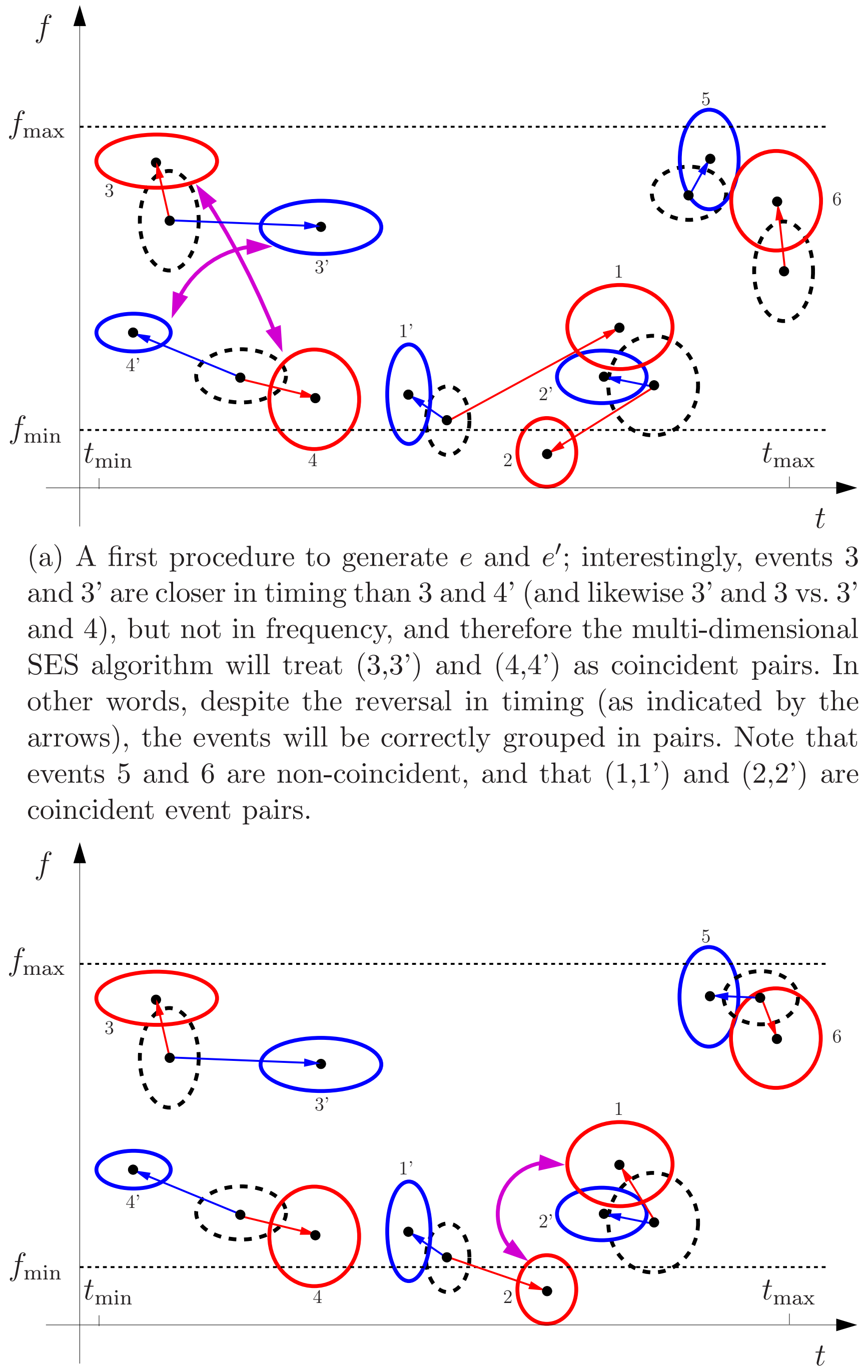
with $\bar{\delta}_t = \delta_t\,(\Delta t_{j_k} + \Delta t'_{j'_k})$, $\bar{\delta}_f = \delta_f\,(\Delta f_{j_k} + \Delta f'_{j'_k})$, $\bar{s}_t = s_t\,(\Delta t_{j_k} + \Delta t'_{j'_k})^2$, $\bar{s}_f = s_f\,(\Delta f_{j_k} + \Delta f'_{j'_k})^2$, and where the constant β is again given by:

$$\beta = p_d\,\sqrt{\lambda}, \tag{9}$$

and

$$\gamma = \Big(\sqrt{\lambda}\,(1 - p_d)\Big)^{n+n'} (1 - \tilde{\lambda})\, \frac{1}{1 - p_d^2\,\tilde{\lambda}}, \tag{10}$$

with $\tilde{\lambda} = \lambda(t_{\max} - t_{\min})(f_{\max} - f_{\min})$. In model (8) the priors p_w, $p_{\Delta t}$ and $p_{\Delta f}$ for the bump parameters w_{j_k}, $w'_{j'_k}$, Δt_{j_k}, $\Delta t'_{j'_k}$, Δf_{j_k}, and $\Delta f'_{j'_k}$ respectively are



(a) A first procedure to generate e and e'; interestingly, events 3 and 3' are closer in timing than 3 and 4' (and likewise 3' and 3 vs. 3' and 4), but not in frequency, and therefore the multi-dimensional SES algorithm will treat (3,3') and (4,4') as coincident pairs. In other words, despite the reversal in timing (as indicated by the arrows), the events will be correctly grouped in pairs. Note that events 5 and 6 are non-coincident, and that (1,1') and (2,2') are coincident event pairs.

(b) A second procedure to generate the same point processes e and e'. Event pairs (1',2) and (1,2') are now coincident, or equivalently, event 1 plays now the role of event 2 in Fig. 4(a) (as indicated by the arrow). Since event 1 is much closer to event 2' than event 1', the SES inference algorithm will most probably prefer the alignment (1,2') and (1',2) instead of (1,1') and (2,2'). Note that (5,6) are now coincident event pairs; since events 5 and 6 are close, the SES algorithm would consider them as coincident.

Fig. 4. Inherent ambiguity in event synchrony: two equivalent procedures to generate the multi-dimensional point processes e and e'.

irrelevant for what follows; we will therefore discard them from now on. We will also discard the constant γ in (8), since it does not depend on j and j'; as a consequence, it is not relevant for determining j, j' and the SES parameters. The parameter β, however, clearly affects the inference of j, j' and the SES parameters, since the exponent of β in (8) does depend on j and j'. We will elaborate on the priors of the parameters $\theta = (\delta_t, s_t, \delta_f, s_f)$ later on.

As in the one-dimensional case, it is instructive to consider the negative logarithm of (8):

$$\begin{aligned} -\log p(e,e',j,j',\theta) = \sum_{k=1}^{n_{\text{co}}^{\text{tot}}} \Big[&\frac{(t'_{j'_k} - t_{j_k} - \delta_t)^2}{2s_t(\Delta t_{j_k} + \Delta t'_{j'_k})^2} + \frac{(f'_{j'_k} - f_{j_k} - \delta_f)^2}{2s_f(\Delta f_{j_k} + \Delta f'_{j'_k})^2} \\ &+ \frac{1}{4} \log 4\pi^2 s_t(\Delta t_{j_k} + \Delta t'_{j'_k})^2 s_f(\Delta f_{j_k} + \Delta f'_{j'_k})^2 \Big] - n_{\text{non-co}}^{\text{tot}} \log \beta \\ &- \log p(\delta_t)p(s_t)p(\delta_f)p(s_f) + \zeta, \end{aligned} \tag{11}$$

with ζ is an irrelevant constant. The expression (11) may be considered as a cost function, along the lines of the one-dimensional case; the unit cost $d(s_t)$ associated to each non-coincident event equals:

$$d(s_t) = -\log \beta. \tag{12}$$

The unit cost $d(e_{j_k}, e'_{j'_k})$ of each event pair $(e_{j_k}, e'_{j'_k})$ is given by:

$$\begin{aligned} d(e_{j_k}, e'_{j'_k}) = &\frac{(t'_{j'_k} - t_{j_k} - \delta_t)^2}{2s_t(\Delta t_{j_k} + \Delta t'_{j'_k})^2} + \frac{(f'_{j'_k} - f_{j_k} - \delta_f)^2}{2s_f(\Delta f_{j_k} + \Delta f'_{j'_k})^2} \\ &+ \frac{1}{4} \log \Big(4\pi^2 s_t(\Delta t_{j_k} + \Delta t'_{j'_k})^2 s_f(\Delta f_{j_k} + \Delta f'_{j'_k})^2 \Big). \end{aligned} \tag{13}$$

Interestingly, the first two terms in (13) may be viewed as an Euclidian distance. Since the point processes e and e' of Fig. 1 are defined on the time-frequency plane, the Euclidean distance is indeed a natural metric. Note that the Euclidian distance is normalized: the timing and frequency offsets are normalized by the bump width and height, due to the particular choices (4)–(7). Fig. 5 explains why normalization is crucial.

In the one-dimensional model proposed in Part I, the third term in (13) is absorbed into the unit cost $d(s_t)$. In the multi-dimensional case, however, that is no longer possible: that term depends on the width and height of the two events e_{j_k} and $e'_{j'_k}$, and cannot be decomposed in a term that *only* depends on the parameters of e_{j_k} and a second term that *only* depends on the parameters of $e'_{j'_k}$; in other words, the third term in (13) cannot be interpreted as unit costs of single events.

In our model of one-dimensional SES, we did not consider the width of events, instead we only incorporated the occurrence time, since that suffices for most

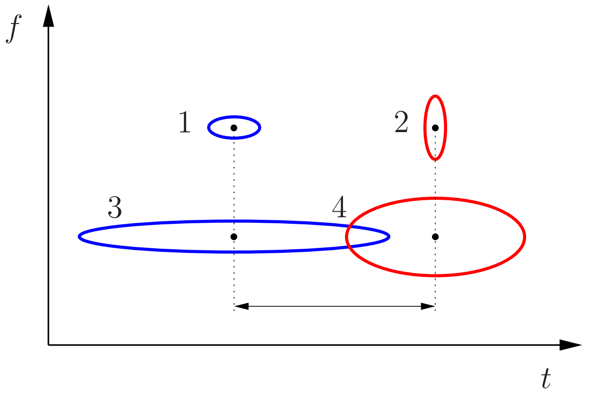


Fig. 5. Bumps 1 and 2 are farther apart than 3 and 4, although the distance between their centers is identical. Therefore, in order to quantify the distance between bumps, it is necessary to normalize the distance between the bump centers by the bump widths (cf. (11)).

common applications. However, the model could easily be extended to include event widths, if necessary.

We wish to point out that the unit costs $d(s_t)$ (12) and $d(e_{j_k}, e'_{j'_k})$ (13) are in general not dimensionless; the total cost (11), however, is dimensionless, in the sense that a change in units will affect the total cost by a constant, irrelevant term. In the one-dimensional case considered in Part I, the unit costs are dimensionless, since the third term in (13) is there absorbed into the unit cost $d(s_t)$.

In principle, one may determine the sequences j and j' and the parameters θ by coordinate descent along the lines of the algorithm of one-dimensional SES. In the multi-dimensional case, however, the alignment cannot be solved by the Viterbi algorithm (or equivalently, the max-product algorithm applied on a cycle-free factor graph of model (8)): one needs to allow timing reversals (see Fig. 4), therefore, the indices j_k and $j'_{k'}$ are no longer necessarily monotonically increasing; as a consequence, the state space becomes substantially larger.

Instead of applying the max-product algorithm on a cycle-free factor graph of model (8), we apply that algorithm on a *cyclic* factor graph, which will amount to a practical procedure to obtain pairwise alignments of multi-dimensional point processes (and bump models in particular); we will show that it finds the optimal solution under very mild conditions. In order to derive this procedure, we introduce a parametrization of model (8) that is naturally represented by a cyclic graph. For each pair of events e_k and $e'_{k'}$, we introduce a binary variable $c_{kk'}$ that equals one if e_k and $e'_{k'}$ form a pair of coincident events and is zero otherwise. Since each event in e associated to at most one event in e', we have the constraints:

$$\sum_{k'=1}^{n'} c_{1k'} \triangleq s_1 \in \{0,1\}, \sum_{k'=1}^{n'} c_{2k'} \triangleq s_2 \in \{0,1\}, \ldots, \sum_{k'=1}^{n'} c_{nk'} \triangleq s_n \in \{0,1\}. \quad (14)$$

Similarly, each event in e' is associated to at most one event in e, which may be expressed by a similar set of constraints. The sequences s and s' are related to the sequences b and b' (cf. Part I) as follows:

$$b_k = 1 - s_k \qquad \text{and} \qquad b'_k = 1 - s'_k. \quad (15)$$

From the variables $c_{kk'}$ (with $k = 1, \ldots, n$ and $k' = 1, \ldots, n'$), one can also easily determine the sequences j and j'. Indeed, if $c_{kk'} = 1$, the index k and k' appear in j and j' respectively.

In this representation, the global statistical model (8) can be cast as:

$$\begin{aligned} p(e, e', b, b', c, \theta) \propto & \prod_{k=1}^{n} (\beta\, \delta[b_k - 1] + \delta[b_k]) \prod_{k'=1}^{n'} (\beta\, \delta[b'_k - 1] + \delta[b'_k]) \\ & \cdot \prod_{k=1}^{n} \prod_{k'=1}^{n'} \Big(\mathcal{N}\Big(t'_{k'} - t_k; \bar{\delta}_t, \bar{s}_t\Big) \mathcal{N}\Big(f'_{k'} - f_k; \bar{\delta}_f, \bar{s}_f\Big)\Big)^{c_{kk'}} \\ & \cdot p(\delta_t) p(s_t) p(\delta_f) p(s_f) \prod_{k=1}^{n} \Big(\delta[b_k + \sum_{k'=1}^{n'} c_{kk'} - 1]\Big) \\ & \cdot \prod_{k'=1}^{n'} \Big(\delta[b'_{k'} + \sum_{k=1}^{n} c_{kk'} - 1]\Big), \end{aligned} \quad (16)$$

where $\delta[\cdot]$ is the Kronecker delta, the variables $c_{kk'}$, b_k, and $b_{k'}$ are binary, and $\bar{\delta}_t = \delta_t\,(\Delta t_k + \Delta t'_{k'})$, $\bar{\delta}_f = \delta_f\,(\Delta f_k + \Delta f'_{k'})$, $\bar{s}_t = s_t\,(\Delta t_k + \Delta t'_{k'})^2$, $\bar{s}_f = s_f\,(\Delta f_k + \Delta f'_{k'})^2$. The last two factors in (16) encode the expressions (15).

We now comment on the priors of the parameters $\theta = (\delta_t, s_t, \delta_f, s_f)$. Since we (usually) do not need to encode prior information about δ_t and δ_f, we choose improper priors $p(\delta_t) = 1 = p(\delta_f)$. On the other hand, one may have prior knowledge about s_t and s_f. For example, in the case of spontaneous EEG (see Section 8), we a priori expect the frequency jitter s_f to be small: frequency shifts can only be caused by non-linear transformations, which are hard to justify from a physiological perspective, therefore, we expect bumps to appear at about the same frequency in both time-frequency maps. On the other hand, the timing jitter s_t may be larger, since signals often propagate over significant distances in the brain, and therefore, timing jitter arises quite naturally. For example, bump nr. 1 in Fig. 2(a) ($t = 10.7$s) should then be paired with bump nr. 3 ($t = 10.9$s) and not with nr. 2 ($t = 10.8$s), since the former is much closer in frequency than the latter. One may encode such prior information by means

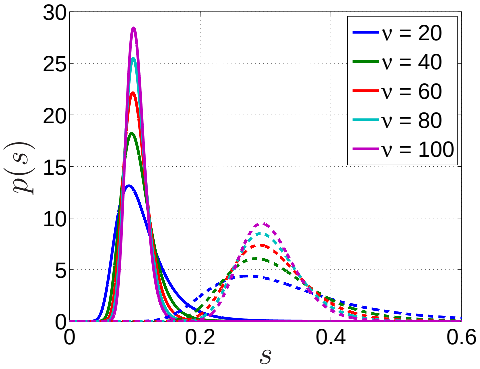


Fig. 6. Scaled inverse chi-square distributions for various values of the degrees of freedom ν and widths s_0; $s_0 = 0.1$ (solid) and 0.3 (dashed).

of conjugate priors for s_t and s_f, i.e., scaled inverse chi-square distributions:

$$p(s_t) = \frac{(s_{0,t}\nu_t/2)^{\nu_t/2}}{\Gamma(\nu_t/2)} \frac{e^{-\nu_t s_{0,t}/2s_t}}{s_t^{1+\nu_t/2}} \tag{17}$$

$$p(s_f) = \frac{(s_{0,f}\nu_f/2)^{\nu_f/2}}{\Gamma(\nu_f/2)} \frac{e^{-\nu_f s_{0,f}/2s_f}}{s_f^{1+\nu_f/2}}, \tag{18}$$

where ν_t and ν_f are the degrees of freedom and $\Gamma(x)$ is the Gamma function. In the example of spontaneous EEG, the widths $s_{0,t}$ and $s_{0,f}$ are chosen such that $s_{0,t} < s_{0,f}$, since s_f is expected to be smaller than s_t. Fig. 6 shows the scaled inverse chi-square distribution with $\nu = 20, 40, \ldots, 100$ and $s_0 = 0.1$ and 0.3.

4 Factor Graph

To perform inference in model (16), we use a *factor graph* of that model (see Fig. 7); each edge represents a variable, each node corresponds to a factor of (16), as indicated by the arrows at the right hand side—we refer to Appendix A for an introduction to factor graphs. We omitted the edges for the (observed) variables t_k, $t'_{k'}$, f_k, $f'_{k'}$, w_k, $w'_{k'}$, Δt_k, $\Delta t'_{k'}$, Δf_k, and $\Delta f'_{k'}$ in order not to clutter the figure. In the following, we discuss the nodes in Fig. 7 (from top to bottom):

- The nodes denoted by β correspond to the factors $(\beta\delta[b_k - 1] + \delta[b_k])$ and $(\beta\delta[b'_k - 1] + \delta[b'_k])$.
- The nodes denoted by $\bar{\Sigma}$ represent the factors $\left(\delta[b_k + \sum_{k'=1}^{n'} c_{kk'} - 1]\right)$ (blue) and $\left(\delta[b'_{k'} + \sum_{k=1}^{n} c_{kk'} - 1]\right)$ (red); see also row 4 in Table C.1.
- The equality constraint nodes (marked by "=") enforce the equality of the incident variables; see also row 3 in Table C.1.

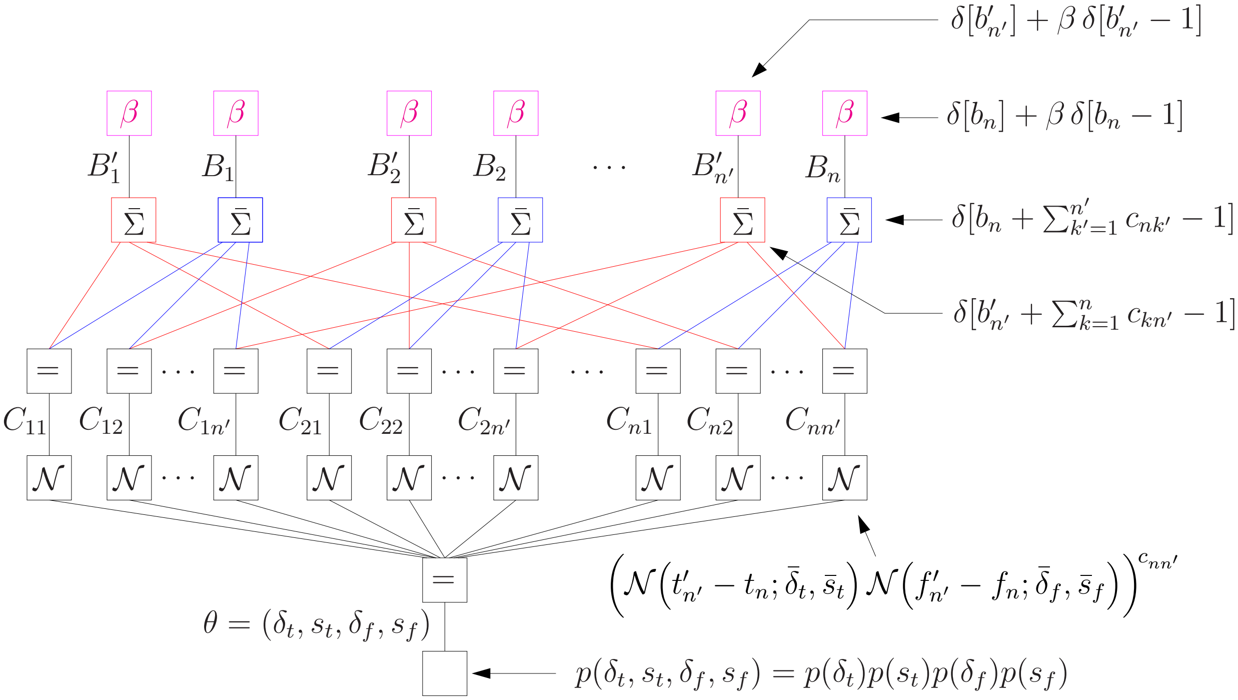

$\delta[b'_{n'}] + \beta\,\delta[b'_{n'} - 1]$
β
β
β
β
β
β
$\delta[b_n] + \beta\,\delta[b_n - 1]$
B'_1
B_1
B'_2
B_2
$\cdots$
$B'_{n'}$
B_n
$\bar{\Sigma}$
$\delta[b_n + \sum_{k'=1}^{n'} c_{nk'} - 1]$
$\delta[b'_{n'} + \sum_{k=1}^{n} c_{kn'} - 1]$
$=$
C_{11}
C_{12}
$C_{1n'}$
C_{21}
C_{22}
$C_{2n'}$
C_{n1}
C_{n2}
$C_{nn'}$
$\mathcal{N}$
$\Big(\mathcal{N}\big(t'_{n'} - t_n; \bar{\delta}_t, \bar{s}_t\big)\,\mathcal{N}\big(f'_{n'} - f_n; \bar{\delta}_f, \bar{s}_f\big)\Big)^{c_{nn'}}$
$\theta = (\delta_t, s_t, \delta_f, s_f)$
$p(\delta_t, s_t, \delta_f, s_f) = p(\delta_t)p(s_t)p(\delta_f)p(s_f)$

Fig. 7. Factor graph of model (16); each edge represents a variable, each node corresponds to a factor of (16), as indicated by the arrows at the right hand side. More details on factor graphs can be found in Appendix A.

- The nodes $\mathcal{N}$ corresponds to the Gaussian distributions in (16), more precisely, they correspond to the factors

$$g_{\mathcal{N}}(c_{kk'}; \theta) = \Big(\mathcal{N}\big(t'_{k'} - t_k; \bar{\delta}_t, \bar{s}_t\big)\,\mathcal{N}\big(f'_{k'} - f_k; \bar{\delta}_f, \bar{s}_f\big)\Big)^{c_{kk'}}, \tag{19}$$

  where $\bar{\delta}_t = \delta_t\,(\Delta t_k + \Delta t'_{k'})$, $\bar{\delta}_f = \delta_f\,(\Delta f_k + \Delta f'_{k'})$, $\bar{s}_t = s_t\,(\Delta t_k + \Delta t'_{k'})^2$, $\bar{s}_f = s_f\,(\Delta f_k + \Delta f'_{k'})^2$.
- The bottom node stands for the prior:

$$p(\theta) = p(\delta_t, s_t, \delta_f, s_f) = p(\delta_t)p(s_t)p(\delta_f)p(s_f). \tag{20}$$

5 Statistical Inference

We determine the alignment $c = (c_{11}, c_{12}, \ldots, c_{nn'})$ and the parameters $\theta = (\delta_t, s_t, \delta_f, s_f)$ by maximum a posteriori (MAP) estimation:

$$(\hat{c}, \hat{\theta}) = \operatorname*{argmax}_{c,\theta} p(e, e', c, \theta), \tag{21}$$

where $p(e, e', c, \theta)$ is obtained from $p(e, e', b, b', c, \theta)$ (16) by marginalizing over b and b':

$$\begin{aligned} p(e, e', c, \theta) \propto & \prod_{k=1}^{n} \Big(\beta\delta\big[\sum_{k'=1}^{n'} c_{kk'}\big] + \delta\big[\sum_{k'=1}^{n'} c_{kk'} - 1\big]\Big) \\ & \cdot \prod_{k=1}^{n} \prod_{k'=1}^{n'} \Big(\mathcal{N}\Big(t'_{k'} - t_k; \bar{\delta}_t, \bar{s}_t\Big)\, \mathcal{N}\Big(f'_{k'} - f_k; \bar{\delta}_f, \bar{s}_f\Big)\Big)^{c_{kk'}} \\ & \cdot p(\delta_t) p(s_t) p(\delta_f) p(s_f). \end{aligned} \tag{22}$$

From $\hat{c}$, we obtain the estimate $\hat{\rho}$ as:

$$\hat{\rho} = \frac{n + n' - 2\sum_{k=1}^{n} \sum_{k'=1}^{n'} \hat{c}_{kk'}}{n + n'} = \frac{\sum_{k=1}^{n} \hat{b}_k + \sum_{k=1}^{n'} \hat{b}'_{k'}}{n + n'}. \tag{23}$$

The MAP estimate (21) is intractable, and we try to obtain (21) by coordinate descent: first, the parameters θ are initialized (e.g., $\hat{\delta}_t^{(0)} = 0 = \delta_f^{(0)}$, $\hat{s}_t^{(0)} = s_{0,t}$, and $\hat{s}_f^{(0)} = s_{0,f}$), then one alternates the following two update rules until convergence (or until the available time has elapsed):

$$\hat{c}^{(i+1)} = \operatorname*{argmax}_{c}\, p(e, e', c, \hat{\theta}^{(i)}) \tag{24}$$

$$\hat{\theta}^{(i+1)} = \operatorname*{argmax}_{\theta}\, p(e, e', \hat{c}^{(i+1)}, \theta). \tag{25}$$

The estimate $\hat{\theta}^{(i+1)}$ (25) is available in closed-form, as we show in Appendix C. In that appendix, we also show that the alignment (24) is equivalent to a classical problem in combinatorial optimization, known as max-weight bipartite matching (see, e.g., (Gerards, 1995; Pulleyblank, 1995; Bayati *et al.*, 2005, 2007; Huang *et al.*, 2007; Sanghavi, 2007a,b)). There is a variety of ways to solve that problem. We will describe one of them in more detail, i.e., the max-product algorithm, since it is arguably the simplest approach. That algorithm can be derived by means of the graph of Fig. 7. It operates by sending information ("messages") along the edges of that graph, as illustrated in Fig. 8. The "messages", depicted by arrows, contain (probabilistic) information about which pairs of bumps are coincident and which are not; they are computed according to a generic rule, i.e., the max-product rule. Intuitively, the nodes may be viewed as computing elements that iteratively update their opinion about the bump matching, based on the opinions ("messages") they receive from neighboring nodes. When the max-product algorithm eventually has converged and the nodes have found a "consensus", the messages are combined to obtain a decision on c, b and b', and an estimate of ρ. In Appendix C, we derive the algorithm (24)(25) in detail; it is summarized in Table 2.

The computational complexity of this algorithm is in principle proportional to nn', i.e., the product of both sequence lengths. If one excludes pairs of events

INPUT: Models e and e', parameters β, ν_t, ν_f, $s_{0,t}$, $s_{0,f}$, $\hat{\delta}_t^{(0)}$, $\hat{\delta}_f^{(0)}$, $\hat{s}_t^{(0)}$, and $\hat{s}_f^{(0)}$

ALGORITHM: Iterate the following two steps until convergence:

(1) <u>Update the alignment $\hat{c}$ by max-product message passing</u>

Initialize messages $\mu\!\downarrow'(c_{kk'}) = 1 = \mu\!\downarrow''(c_{kk'})$

Iterative until convergence:

a. Upward messages:

$$\begin{aligned}\mu\!\uparrow'(c_{kk'}) &\propto \mu\!\downarrow''(c_{kk'})g_{\mathcal{N}}(c_{kk'};\hat{\theta}^{(i)})\\ \mu\!\uparrow''(c_{kk'}) &\propto \mu\!\downarrow'(c_{kk'})g_{\mathcal{N}}(c_{kk'};\hat{\theta}^{(i)}),\end{aligned}$$

where

$$g_{\mathcal{N}}(c_{kk'};\hat{\theta}^{(i)}) = \left(\mathcal{N}\Big(t'_{k'} - t_k; \bar{\delta}_t^{(i)}, \bar{s}_t^{(i)}\Big)\,\mathcal{N}\Big(f'_{k'} - f_k; \bar{\delta}_f^{(i)}, \bar{s}_f^{(i)}\Big)\right)^{c_{kk'}},$$

with $\bar{\delta}_t^{(i)} = \hat{\delta}_t^{(i)}(\Delta t_k + \Delta t'_{k'})$, $\bar{\delta}_f^{(i)} = \hat{\delta}_f^{(i)}(\Delta f_k + \Delta f'_{k'})$, $\bar{s}_t^{(i)} = \hat{s}_t^{(i)}(\Delta t_k + \Delta t'_{k'})^2$, $\bar{s}_f^{(i)} = \hat{s}_f^{(i)}(\Delta f_k + \Delta f'_{k'})^2$

b. Downward messages:

$$\begin{pmatrix}\mu\!\downarrow'(c_{kk'}=0)\\ \mu\!\downarrow'(c_{kk'}=1)\end{pmatrix} \propto \begin{pmatrix}\max\big(\beta, \max_{\ell'\neq k'} \mu\!\uparrow'(c_{k\ell'}=1)/\mu\!\uparrow'(c_{k\ell'}=0)\big)\\ 1\end{pmatrix}$$

$$\begin{pmatrix}\mu\!\downarrow''(c_{kk'}=0)\\ \mu\!\downarrow''(c_{kk'}=1)\end{pmatrix} \propto \begin{pmatrix}\max\big(\beta, \max_{\ell\neq k} \mu\!\uparrow''(c_{\ell k'}=1)/\mu\!\uparrow''(c_{\ell k'}=0)\big)\\ 1\end{pmatrix}$$

Compute marginals $p(c_{kk'}) \propto \mu\!\downarrow'(c_{kk'})\mu\!\downarrow''(c_{kk'})g_{\mathcal{N}}(c_{kk'};\hat{\theta}^{(i)})$

Compute decisions $\hat{c}_{kk'} = \mathrm{argmax}_{c_{kk'}}\, p(c_{kk'})$

(2) <u>Update the SES parameters:</u>

$$\begin{aligned}\hat{\delta}_t^{(i+1)} &= \frac{1}{n^{(i+1)}}\sum_{k=1}^{n^{(i+1)}} \frac{\hat{t}_k'^{(i+1)} - \hat{t}_k^{(i+1)}}{\Delta\hat{t}_k^{(i+1)} + \Delta\hat{t}_k'^{(i+1)}}\\ \hat{\delta}_f^{(i+1)} &= \frac{1}{n^{(i+1)}}\sum_{k=1}^{n^{(i+1)}} \frac{\hat{f}_k'^{(i+1)} - \hat{f}_k^{(i+1)}}{\Delta\hat{f}_k^{(i+1)} + \Delta\hat{f}_k'^{(i+1)}}\\ \hat{s}_t^{(i+1)} &= \frac{\nu_t s_{0,t} + n^{(i+1)}\hat{s}_{t,\mathrm{sample}}^{(i+1)}}{\nu_t + n^{(i+1)} + 2}\\ \hat{s}_f^{(i+1)} &= \frac{\nu_f s_{0,f} + n^{(i+1)}\hat{s}_{f,\mathrm{sample}}^{(i+1)}}{\nu_f + n^{(i+1)} + 2},\end{aligned}$$

OUTPUT: Alignment $\hat{c}$ and SES parameters $\hat{\rho}$, $\hat{\delta}_t$, $\hat{\delta}_f$, $\hat{s}_t$, $\hat{s}_f$

Table 2
Inference algorithm for multi-dimensional SES.

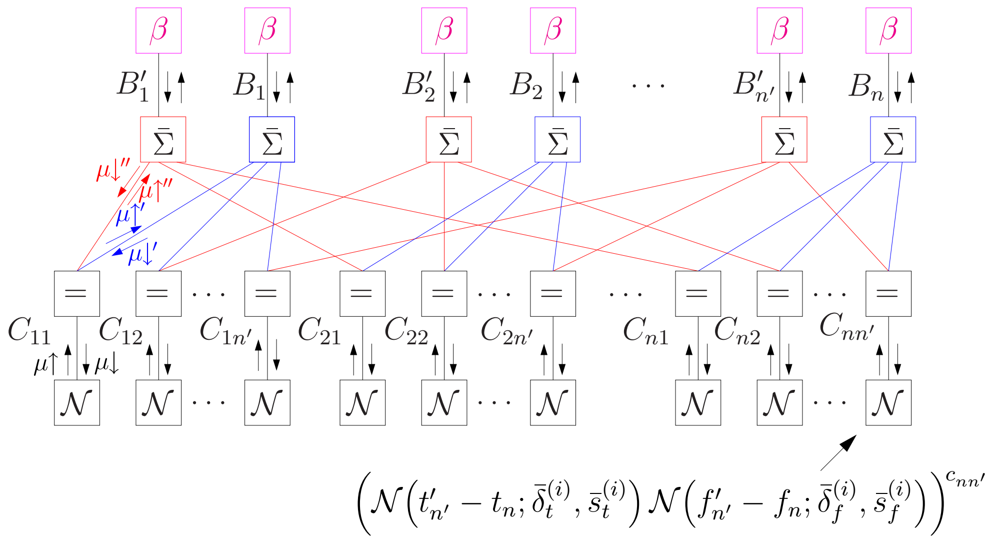


Fig. 8. Max-product message passing; the messages indicate the max-product messages, computed according to the max-product update rule (see Appendix C).

that are too far apart on the time-frequency map, one obtains an algorithm with linear complexity (as in the one-dimensional case).

The fixed points of the algorithm can be characterized as follows: the fixed point $\hat{\theta}$ is a stationary point of (22), and the alignment $\hat{c}$ is "neighborhood maximum", i.e., the posterior probability (22) of $\hat{c}$ is guaranteed to be greater than all other assignments in region around that assignment $\hat{c}$ (Freeman *et al.*, 1999).

The algorithm is an instance of coordinate descent, and is therefore guaranteed to converge if the conditional maximizations (24)(25) have unique solutions (Bezdek *et al.*, 2002, 1987). The conditional maximization (25) always has a unique solution (cf. (C.1)–(C.4)). If the alignment (24) has a unique solution, the max-product algorithm is guaranteed to find that unique optimum (Bayati *et al.*, 2005; Huang *et al.*, 2007; Bayati *et al.*, 2007; Sanghavi, 2007a,b). Therefore, as long as (24) has a unique solution, the algorithm of Table 2 is guaranteed to converge. In many applications, the optimum of (24) is unique with probability one, and as a consequence, the proposed algorithm converges. We will provide numerical results in Section 8.

6 Extensions

So far, we have developed multi-dimensional SES for the particular example of bump models in the time-frequency domain. Here we consider alternative SES models, both in time-frequency domain and in other domain. Those models are straightforward extensions of (8). We will also outline how the SES inference algorithm can be modified accordingly.

6.1 Bump Parameters

One may incorporate differences in amplitude, width and height between the bumps of e and e' in model (8). In the generative process of Fig. 3, those parameters are then no longer drawn independently from certain prior distributions; instead they are obtained by perturbing the parameters of the hidden bump model v. In particular, one may again consider Gaussian perturbations, as for the timing t, t' and frequency f, f' of the bumps. This leads to additional parameters δ_w, $\delta_{\Delta t}$, $\delta_{\Delta f}$, s_w, $s_{\Delta t}$, and $s_{\Delta f}$, which stand for the average offset and jitter between the bump amplitudes, widths and heights of e and e' respectively; it also leads to additional Gaussian factors in model (8). Moreover, priors for those additional parameters can be included in that model, leading to the expression:

$$\begin{aligned} p(e, e', j, j', \theta) = \gamma\, \beta^{n^{\text{tot}}_{\text{non-co}}} & p(\delta_t)p(s_t)p(\delta_f)p(s_f)p(\delta_w)p(s_w)p(\delta_{\Delta t})p(s_{\Delta t})p(\delta_{\Delta f})p(s_{\Delta f}) \\ & \cdot \mathcal{N}\Big(t'_{j'_k} - t_{j_k}; \bar{\delta}_t, \bar{s}_t\Big)\, \mathcal{N}\Big(f'_{j'_k} - f_{j_k}; \bar{\delta}_f, \bar{s}_f\Big)\, \mathcal{N}\Big(w'_{j'_k} - w_{j_k}; \delta_w, s_w\Big) \\ & \cdot \mathcal{N}\Big(\Delta t'_{j'_k} - \Delta t_{j_k}; \delta_{\Delta_t}, s_{\Delta_t}\Big)\, \mathcal{N}\Big(\Delta f'_{j'_k} - \Delta f_{j_k}; \delta_{\Delta_f}, s_{\Delta_f}\Big), \end{aligned} \tag{26}$$

where the parameters β and γ are again given by (9) and (10) respectively.

The inference algorithm for this model is very similar to the one of model (8) (cf. Table 2). The additional parameters δ_w, $\delta_{\Delta t}$, $\delta_{\Delta f}$, s_w, $s_{\Delta t}$, and $s_{\Delta f}$ are updated similarly as δ_t, δ_f, s_t, and s_f. The alignment procedure is almost identical, one merely needs to modify the upward message $g_{\mathcal{N}}$ (see Step 1 in Table 2): besides the Gaussian factors for the timing and frequency offsets, that message contains similar factors for the offsets in bump amplitude, width, and height (cf. 26).

6.2 Oblique Bumps

Alternatively, one may consider *oblique* bumps, i.e., bumps that are not necessarily parallel to the time and frequency axes (see Fig. 9). Such bumps correspond to chirps (see, e.g., (O'Neill *et al.*, 2000; Cui *et al.*, 2007, 2005)). The rotation angle of each bump e_k and $e'_{k'}$ is denoted by α_k and $\alpha'_{k'}$ respectively (with $\alpha_k, \alpha'_{k'} \in [0, \pi/2]$ for all k and k'). The model (8) can take those rotations into account: first, the normalization of the timing and frequency offsets needs to be modified accordingly, second, one may wish to incorporate the difference in rotation angles α_k and $\alpha'_{k'}$ of bump e_k and $e'_{k'}$ respectively. In the generative process of Fig. 3, one may include Gaussian perturbations

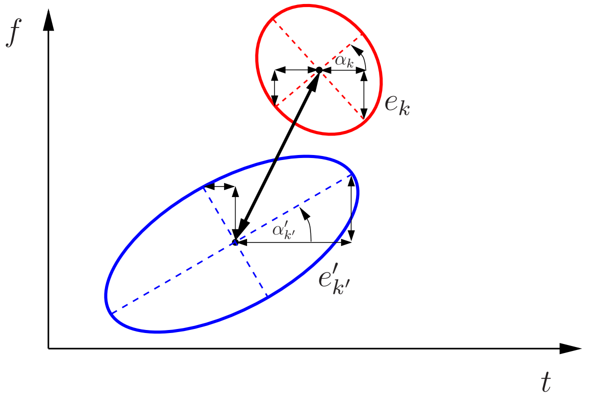


Fig. 9. Two oblique bumps e_k and $e'_{k'}$ with rotation angle α_k and $\alpha'_{k'}$.

for the rotation angles. This leads to the following extension of model (8):

$$
\begin{aligned}
p(e, e', j, j', \theta) = \gamma\, \beta^{n_{\text{non-co}}^{\text{tot}}} p(\delta_t) p(s_t) p(\delta_f) p(s_f) p(\delta_\alpha) p(s_\alpha) \\
\cdot \prod_{k=1}^{n_{\text{co}}^{\text{tot}}} p_w(w_{j_k}) p_w(w'_{j'_k}) p_{\Delta t}(\Delta t_{j_k}) p_{\Delta t}(\Delta t'_{j'_k}) p_{\Delta f}(\Delta f_{j_k}) p_{\Delta f}(\Delta f'_{j'_k}) \\
\cdot \mathcal{N}\Big(t'_{j'_k} - t_{j_k}; \bar{\delta}_t, \bar{s}_t\Big) \mathcal{N}\Big(f'_{j'_k} - f_{j_k}; \bar{\delta}_f, \bar{s}_f\Big) \mathcal{N}\Big(\alpha'_{j'_k} - \alpha_{j_k}; \delta_\alpha, s_\alpha\Big),
\end{aligned}
\tag{27}
$$

where δ_α and s_α are the average offset and jitter respectively between the rotation angles, the parameters β and γ are again given by (9) and (10) respectively, $\bar{\delta}_t = \delta_t\, (\tilde{\Delta} t_{j_k} + \tilde{\Delta} t'_{j'_k})$, $\bar{\delta}_f = \delta_f\, (\tilde{\Delta} f_{j_k} + \tilde{\Delta} f'_{j'_k})$, $\bar{s}_t = s_t\, (\tilde{\Delta} t_{j_k} + \tilde{\Delta} t'_{j'_k})^2$, $\bar{s}_f = s_f\, (\tilde{\Delta} f_{j_k} + \tilde{\Delta} f'_{j'_k})^2$, with

$$\tilde{\Delta} t_{j_k} = \cos \alpha_k\, \Delta t_{j_k} + \sin \alpha_k\, \Delta f_{j_k} \tag{28}$$

$$\tilde{\Delta} f_{j_k} = \sin \alpha_k\, \Delta t_{j_k} + \cos \alpha_k\, \Delta f_{j_k} \tag{29}$$

$$\tilde{\Delta} t'_{j'_k} = \cos \alpha'_{k'}\, \Delta t'_{j'_k} + \sin \alpha'_{k'}\, \Delta f'_{j'_k} \tag{30}$$

$$\tilde{\Delta} f'_{j'_k} = \sin \alpha'_{k'}\, \Delta t'_{j'_k} + \cos \alpha'_{k'}\, \Delta f'_{j'_k}. \tag{31}$$

Note that that model (27) reduces to (8) if $\alpha_k = 0 = \alpha'_{k'}$ for all k and k'. The model (27) does not incorporate differences in the amplitude, width, and height of the bumps, but it could easily be extended if necessary.

In order to extend the SES algorithm of Table 2 to oblique bumps, one needs to make three modifications:

- In the upward message $g_{\mathcal{N}}$ (Step 1 in Table 2) and in the update of the SES parameters (Step 2), the parameters Δt_k, $\Delta t'_{k'}$, Δf_k, and $\Delta f'_{k'}$ are replaced by $\tilde{\Delta} t_k$, $\tilde{\Delta} t'_{k'}$, $\tilde{\Delta} f_k$, and $\tilde{\Delta} f'_{k'}$.
- If the prior on the parameters δ_α and s_α is the improper prior $p(\delta_\alpha) = 1 =$

$p(s_\alpha)$, those parameters are updated similarly as δ_t and s_t:

$$\hat{\delta}_\alpha^{(i+1)} = \frac{1}{n^{(i+1)}} \sum_{k=1}^{n^{(i+1)}} \hat{\alpha}_k'^{(i+1)} - \hat{\alpha}_k^{(i+1)} \tag{32}$$

$$\hat{s}_\alpha^{(i+1)} = \frac{1}{n^{(i+1)}} \sum_{k=1}^{n^{(i+1)}} \left(\hat{\alpha}_k'^{(i+1)} - \hat{\alpha}_k^{(i+1)} - \hat{\delta}_\alpha^{(i+1)} \right)^2. \tag{33}$$

Note that since $\alpha_k, \alpha_k' \in [0, \pi/2]$, one can simply add the angle differences.

- The alignment procedure is again almost identical: the upward message $g_{\mathcal{N}}$ (Step 1 in Table 2) contains, in addition to the Gaussian factors for the timing and frequency offsets, a similar factor for the offsets in bump rotation angle (cf. (27)).

6.3 Point Processes in Other Spaces

6.3.1 Euclidean Spaces

Until now we have considered bump models in the time-frequency domain. However, the statistical model (8) directly applies to point processes in other domains, for example three-dimensional space. Indeed, one can easily verify that the generative procedure depicted in Fig. 3 is not restricted to time-frequency domain, since at no point, the procedure relies on particularities of time-frequency. In general, the constants β and γ in (8) are still defined by (9) and (10) respectively. The constant $\tilde{\lambda}$ in (10) is in general defined as $\tilde{\lambda} = \lambda\, \mathrm{vol}(S)$, where S is the space in which the point processes are defined and $\mathrm{vol}(S)$ is the volume of that space. The SES algorithm can straightforwardly be extended to more general models, along the lines of the extensions we considered in Section 6.1 and 6.2.

For example, in time and three-dimensional space, SES may be described by the following statistical model:

$$\begin{aligned} p(e, e', j, j', \theta) &= \gamma\, \beta^{n^{\text{tot}}_{\text{non-co}}} p(\delta_t) p(s_t) p(\delta_x) p(s_x) p(\delta_y) p(s_y) p(\delta_z) p(s_z) \\ &\quad \cdot \prod_{k=1}^{n^{\text{tot}}_{\text{co}}} p_w(w_{j_k}) p_w(w'_{j'_k}) p_{\Delta t}(\Delta t_{j_k}) p_{\Delta t}(\Delta t'_{j'_k}) p_{\Delta x}(\Delta x_{j_k}) p_{\Delta x}(\Delta x'_{j'_k}) \\ &\quad \cdot p_{\Delta y}(\Delta y_{j_k}) p_{\Delta y}(\Delta y'_{j'_k}) p_{\Delta z}(\Delta z_{j_k}) p_{\Delta z}(\Delta z'_{j'_k})\, \mathcal{N}\Big(t'_{j'_k} - t_{j_k}; \bar{\delta}_t, \bar{s}_t\Big) \\ &\quad \cdot \mathcal{N}\Big(x'_{j'_k} - x_{j_k}; \bar{\delta}_x, \bar{s}_x\Big)\, \mathcal{N}\Big(y'_{j'_k} - y_{j_k}; \bar{\delta}_y, \bar{s}_y\Big)\, \mathcal{N}\Big(z'_{j'_k} - z_{j_k}; \bar{\delta}_z, \bar{s}_z\Big), \end{aligned} \tag{34}$$

where $\bar{\delta}_t = \delta_t\, (\Delta t_{j_k} + \Delta t'_{j'_k})$, $\bar{\delta}_x = \delta_x\, (\Delta x_{j_k} + \Delta x'_{j'_k})$, $\bar{\delta}_y = \delta_y\, (\Delta y_{j_k} + \Delta y'_{j'_k})$, $\bar{\delta}_z = \delta_z\, (\Delta z_{j_k} + \Delta z'_{j'_k})$, the parameters $\bar{s}_t$, $\bar{s}_x$, $\bar{s}_y$, and $\bar{s}_z$ are defined similarly, and the parameters β and γ are again given by (9) and (10) respectively. The parameter $\tilde{\lambda}$ in (10) is now defined as:

$$\tilde{\lambda} = \lambda(t_{\max} - t_{\min})(x_{\max} - x_{\min})(y_{\max} - y_{\min})(z_{\max} - z_{\min}). \tag{35}$$

In model (34), the bumps have dispersion in time and space. In some applications, however, the bumps may only have dispersion in space and no dispersion in time. In that case, one would need to replace $\bar{\delta}_t$ and $\bar{s}_t$ by δ_t and s_t respectively, and there would be no factors $p_{\Delta t}(\Delta t_{j_k})$ and $p_{\Delta t}(\Delta t'_{j'_k})$.

Note that an SES model for point processes in three-dimensional space may be directly obtained from model (34); one simply needs to remove the factors $p(\delta_t)$, $p(s_t)$, $p_{\Delta t}(\Delta t_{j_k})$, $p_{\Delta t}(\Delta t'_{j'_k})$, and $\mathcal{N}\Big(t'_{j'_k} - t_{j_k}; \bar{\delta}_t, \bar{s}_t\Big)$.

The inference algorithm for model (34) can be readily obtained from the algorithm of Table 2. The parameter updates are very similar, and the same holds for the pairwise alignment procedure: the upward message $g_{\mathcal{N}}$ (Step 1 in Table 2) contains a Gaussian factors for the timing offsets and similar factors for the offsets in the three spatial dimensions (cf. (34)).

Interestingly, one can easily *combine* the above extensions. For example, one may consider oblique bumps in time and three-dimensional space; that model may take changes in bump orientation, amplitude and width into account.

6.3.2 Non-Euclidean Spaces

So far, we have considered Gaussian perturbations, or equivalently, Euclidean distances. In some applications, however, the point processes may be defined on *curved* manifolds, and non-Euclidean distances are then more natural. For instance, the two point processes may be defined on a planar closed curve. We consider such example in (Dauwels *et al.*, 2008), which concerns the synchrony of morphological and molecular events in cell migration. More specifically, those events are extracted from time-lapse fluorescence resonance energy transfer (FRET) images of Rac1 activity; the protein Rac1 is well known to induce filamentous structures that enable cells to migrate. The morphological and molecular events take place along the cell boundary, and since we consider images, that boundary is a closed planar curve. We do not take the dispersion of the events into account, since it is not relevant for the application at hand. The morphological and molecular events are denoted by $e = ((t_1, u_1, w_1), \ldots, (t_n, u_n, w_n))$ and $e' = ((t'_1, u'_1, w'_1), \ldots, (t'_{n'}, u'_{n'}, w'_{n'}))$ respectively, where t_k and $t'_{k'}$, u_k and $u'_{k'}$, and w_k and $w'_{k'}$ denote the occurrence time, position along the boundary, and the amplitude respectively of the morphological and molecular events. The distance between morphological and molecular events is non-euclidean. We adopt the following statistical model for morphological and molecular events (Dauwels *et al.*, 2008):

$$p(e, e', j, j', \theta) = \gamma\, \beta^{n^{\text{tot}}_{\text{non-co}}} p(\delta_t) p(s_t) p(\delta_u) p(s_u) \prod_{k=1}^{n^{\text{tot}}_{\text{co}}} p_w(w_{j_k}) p_w(w'_{j'_k}) \\ \cdot \mathcal{N}\Big(g_t(t_{j_k}, t'_{j'_k}); \delta_t, s_t\Big) \mathcal{N}\Big(g_u(u_{j_k}, u'_{j'_k}); \delta_u, s_u\Big), \qquad (36)$$

where g_t and g_u are non-linear functions that take the shape of the cell boundary into account. Due to those non-linearities, the factors $\mathcal{N}(g_t(t_{j_k}, t'_{j'_k}); \delta_t, s_t)$ and $\mathcal{N}(g_u(u_{j_k}, u'_{j'_k}); \delta_u, s_u)$ are not Gaussian distributions, and the distance between events is non-euclidean. The parameters β and γ are again given by (9) and (10) respectively. The parameter $\tilde{\lambda}$ in (10) is now defined as:

$$\tilde{\lambda} = \lambda(t_{\max} - t_{\min})L, \qquad (37)$$

where L is the length of the cell boundary. Extending the algorithm of Table 2 to model (36) is straightforward. The parameter updates (Step 2 in Table 2) are now given by:

$$\hat{\delta}_t^{(i+1)} = \frac{1}{n^{(i+1)}} \sum_{k=1}^{n^{(i+1)}} g_t\Big(\hat{t}_k^{(i+1)}, \hat{t'}_k^{(i+1)}\Big) \qquad (38)$$

$$\hat{s}_t^{(i+1)} = \frac{1}{n^{(i+1)}} \sum_{k=1}^{n^{(i+1)}} \Big(g_t\Big(\hat{t}_k^{(i+1)}, \hat{t'}_k^{(i+1)}\Big) - \hat{\delta}_t^{(i+1)}\Big)^2 \qquad (39)$$

$$\hat{\delta}_u^{(i+1)} = \frac{1}{n^{(i+1)}} \sum_{k=1}^{n^{(i+1)}} g_u\Big(\hat{u}_k^{(i+1)}, \hat{u'}_k^{(i+1)}\Big) \qquad (40)$$

$$\hat{s}_u^{(i+1)} = \frac{1}{n^{(i+1)}} \sum_{k=1}^{n^{(i+1)}} \Big(g_u\Big(\hat{u}_k^{(i+1)}, \hat{u'}_k^{(i+1)}\Big) - \hat{\delta}_u^{(i+1)}\Big)^2. \qquad (41)$$

The pairwise alignment procedure is almost identical (Step 1 in Table 2), we again only need to modify the upward message $g_{\mathcal{N}}$:

$$g_{\mathcal{N}}(c_{kk'}; \hat{\theta}^{(i)}) = \left(\mathcal{N}\Big(g_t(t_k, t'_{k'}); \hat{\delta}_t^{(i)}, \hat{s}_t^{(i)}\Big) \mathcal{N}\Big(g_u(u_k, u'_{k'}); \hat{\delta}_u^{(i)}, \hat{s}_u^{(i)}\Big)\right)^{c_{kk'}}.$$

In the next sections, we will use the basic model (8), since that suffices for our purposes.

7 Analysis of Surrogate Data

As in the one-dimensional case (Part I, Section 6), we investigate the robustness and reliability of multi-dimensional SES by means of surrogate data.

We randomly generated 1'000 pairs of two-dimensional point processes (e, e') according to the symmetric procedure depicted in Fig. 3.

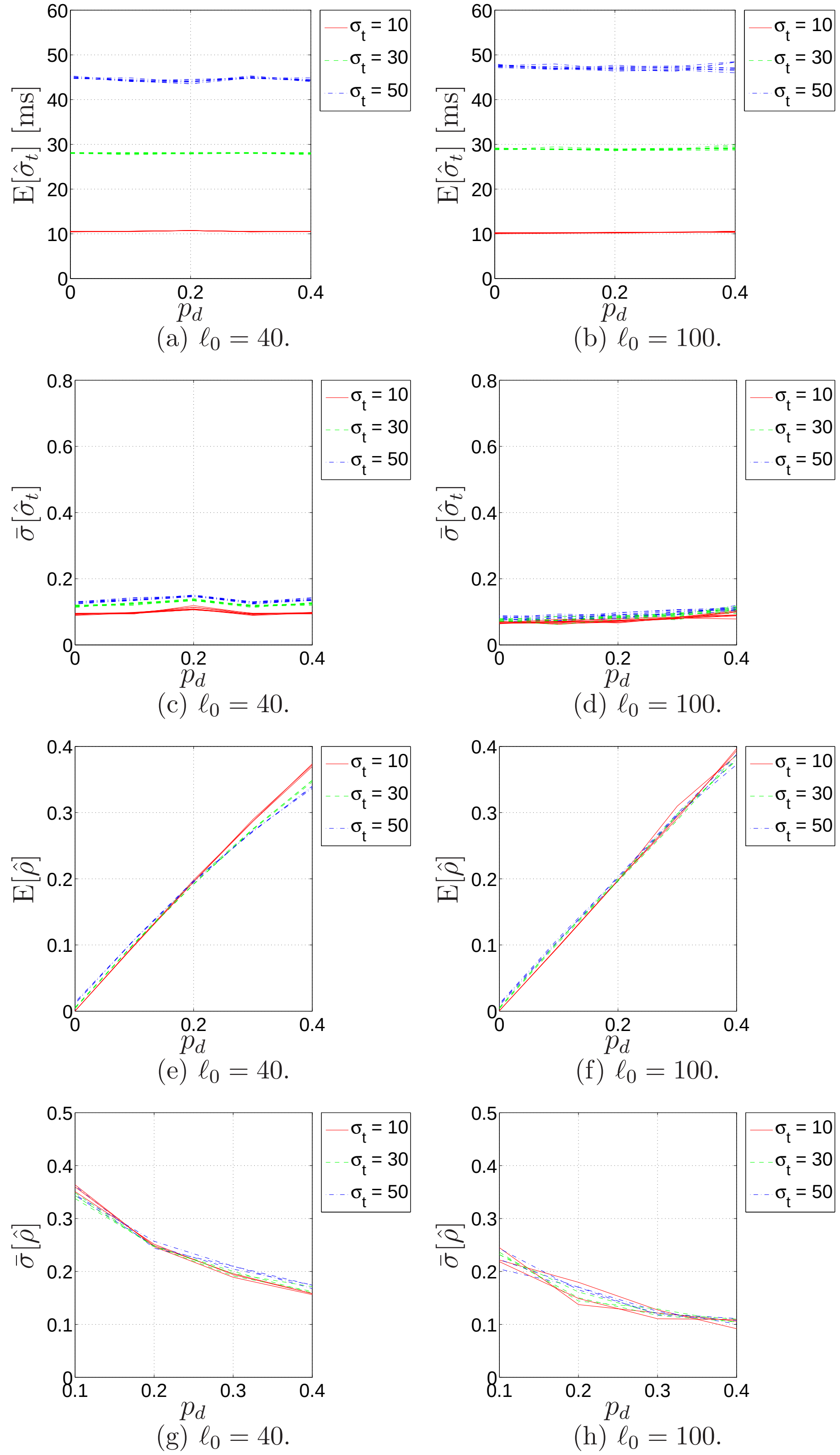
We considered several values of the parameters ℓ, p_d, δ_t, δ_f, s_t, (σ_t) and s_f (σ_f). More specifically, the length ℓ was chosen as $\ell = \ell_0/(1 - p_d)$, where $\ell_0 \in \mathbb{N}_0$ is a constant. With this choice, the expected length of e and e' is ℓ_0, independently of p_d. We considered the values $\ell_0 = 40$ and 100, $p_d = 0$, 0.1, $\ldots$, 0.4, $\delta_t = 0$ms, 25ms, 50ms, $\sigma_t = 10$ms, 30ms, and 50ms, $\delta_f = 0$Hz, 2.5Hz, 5Hz, $\sigma_f = 1$Hz, 2.5Hz, and 5Hz, $t_{\min} = 0$s, $f_{\min} = 0$Hz, $t_{\max} = \ell_0 \cdot 100$ms and $f_{\max} = \ell_0 \cdot 1$Hz. With this choice, the average event occurrence rate is about 10Hz, for all ℓ_0 and p_d. The width Δt_k and height Δf_k of all bumps is set equal to 0.5, so that $(\Delta t_k + \Delta t'_{k'}) = 1 = (\Delta f_k + \Delta f'_{k'})$ for all k and k', and hence $\bar{\delta}_t = \delta_t$, $\bar{\delta}_f = \delta_f$, $\bar{s}_t = s_t$, and $\bar{s}_f = s_f$ (cf. (4), (5), (6), (7), and Table 2).

We used the initial values $\hat{\delta}_t^{(0)} = 0$, 30, and 70ms, $\hat{\delta}_f^{(0)} = 0$Hz, $\hat{s}_t^{(0)} = (30\text{ms})^2$, and $\hat{s}_f^{(0)} = (3\text{Hz})^2$. The parameter β was identical for all parameter settings, i.e., $\beta = 0.005$; it was optimized to yield the best overall results. We used an uninformative prior for δ_t, δ_f, s_t, and s_f,, i.e., $p(\delta_t) = p(\delta_f) = p(s_t) = p(s_f) = 1$.

In order to assess the SES measures $S = s_t, \rho$, we compute for each above mentioned parameter setting the expectation E$[S]$ and normalized standard deviation $\overline{\sigma}[S] = \sigma[S]/\mathrm{E}[S]$. Those statistics are computed by averaging over 1'000 pairs of point processes (e,e'), randomly generated according to the symmetric procedure depicted in Fig. 3.

The results are summarized in Fig. 10 to 12. From those figures we can make the following observations:

- The estimates of s_t and p_d are slightly biased, especially for small ℓ_0, i.e., $\ell_0 = 40$, $s_t \geq (30\text{ms})^2$ and $p_d > 0.2$; more specifically, the expected value of those estimates is slightly smaller than the true value, which is due to ambiguity inherent in event synchrony (cf. Fig. 4). However, the bias is significantly smaller than in the one-dimensional case (cf. Part I, Section 6); the bias increases with s_f, which is in agreement with our expectations: the more frequency jitter, the more likely that some events are reversed in frequency, and hence are aligned incorrectly.
- As in the one-dimensional case, the estimates of δ_t are unbiased for all considered values of δ_t, δ_f, s_t, s_f, and p_d, likewise the estimates of δ_f (not shown here).
- The estimates of s_t do only weakly depend on p_d, and vice versa.
- The estimates of s_t and p_d do not depend on δ_t and δ_f, i.e., they are robust to lags δ_t and frequency offsets δ_f, since the latter can be estimated reliably.
- The normalized standard deviation of the estimates of δ_t, s_t and p_d grows with s_t and p_d, but it remains below 30%. Those estimates are therefore



(a) $\ell_0 = 40$.

(b) $\ell_0 = 100$.

(c) $\ell_0 = 40$.

(d) $\ell_0 = 100$.

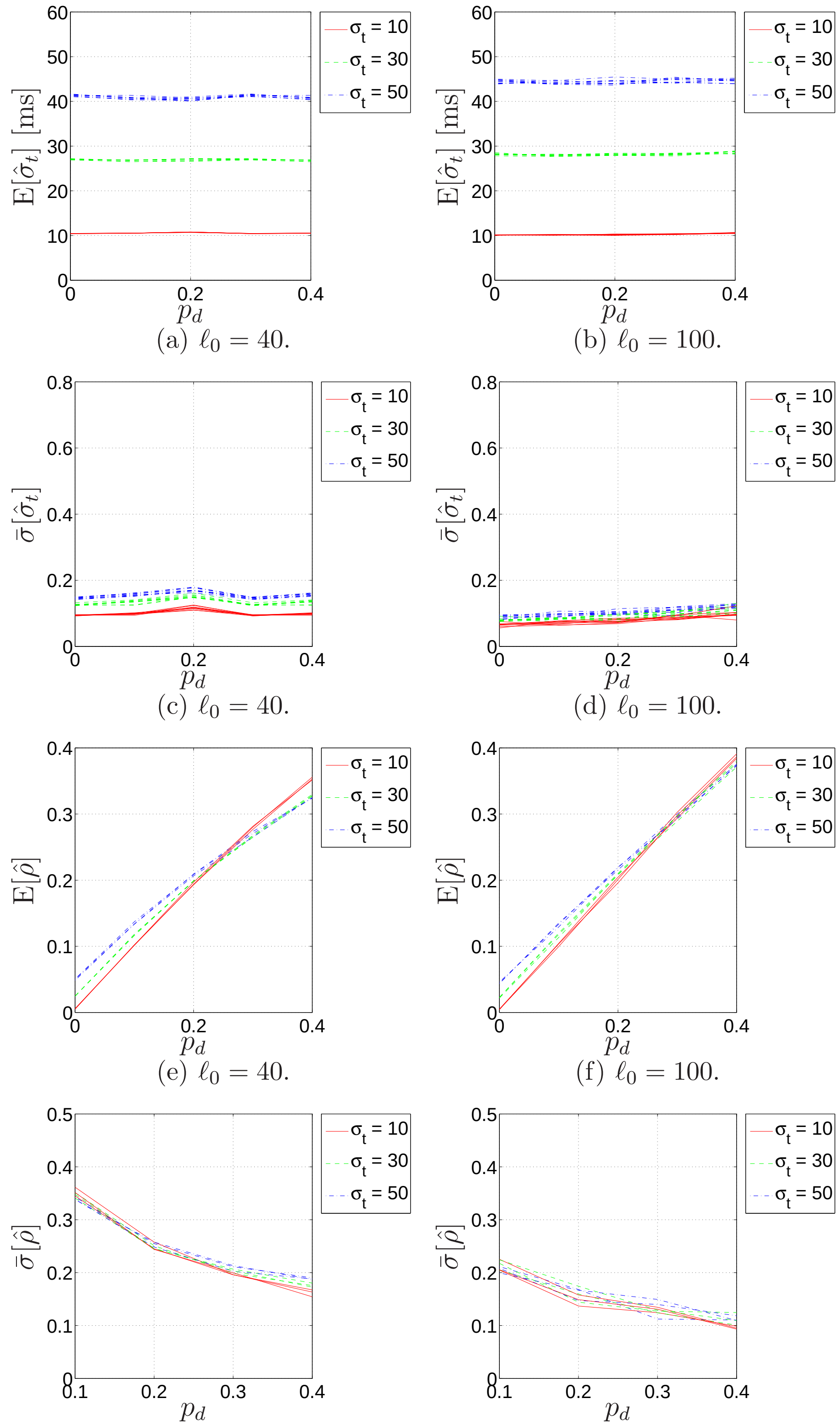
(e) $\ell_0 = 40$.

(f) $\ell_0 = 100$.

(g) $\ell_0 = 40$.

(h) $\ell_0 = 100$.

Fig. 10. Results for surrogate data: the figure shows the expected value E[$\hat{\sigma}_t$] and E[$\hat{\rho}$] and the normalized standard deviation $\bar{\sigma}[\hat{\sigma}_t]$ and $\bar{\sigma}[\hat{\rho}]$ for the parameter settings $\ell_0 = 40$ and 100, $\delta_t = 0, 25, 50$ms, $\delta_f = 0, 2.5, 5$Hz, $\sigma_t = 10, 30, 50$ms, $\sigma_f = 1$Hz and $p_d = 0, 0.1, \ldots, 0.4$. The curves for different δ_t and δ_t are practically coinciding.



(a) $\ell_0 = 40$.

(b) $\ell_0 = 100$.

(c) $\ell_0 = 40$.

(d) $\ell_0 = 100$.

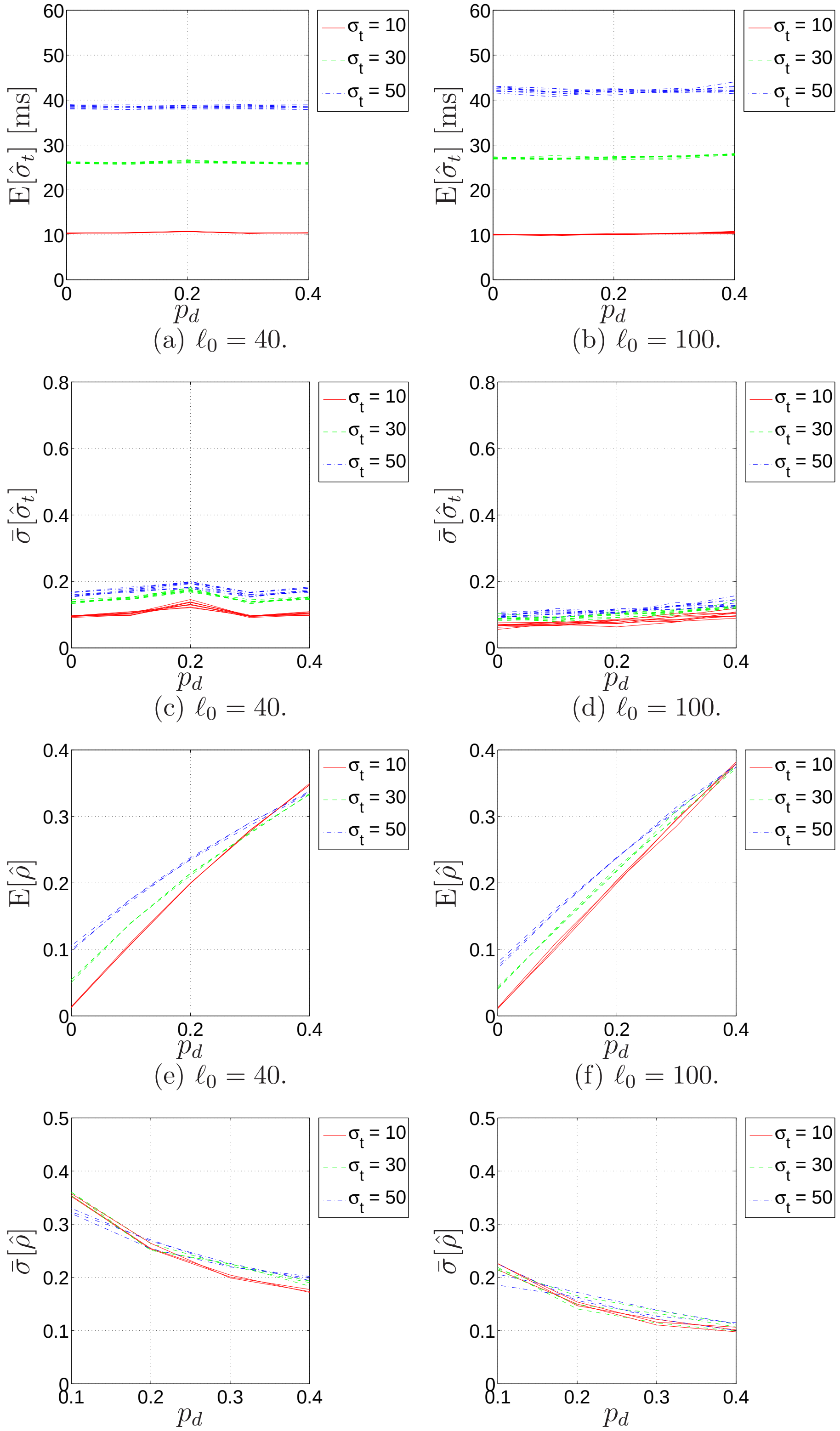
(e) $\ell_0 = 40$.

(f) $\ell_0 = 100$.

(g) $\ell_0 = 40$.

(h) $\ell_0 = 100$.

Fig. 11. Results for surrogate data: the figure shows the expected value $\mathrm{E}[\hat{\sigma}_t]$ and $\mathrm{E}[\hat{\rho}]$ and the normalized standard deviation $\bar{\sigma}[\hat{\sigma}_t]$ and $\bar{\sigma}[\hat{\rho}]$ for same the parameter settings as in Fig. 10, but now with $\sigma_f = 2.5$Hz. Again, the curves for different δ_t and δ_f are practically coinciding.



(a) $\ell_0 = 40$.

(b) $\ell_0 = 100$.

(c) $\ell_0 = 40$.

(d) $\ell_0 = 100$.

(e) $\ell_0 = 40$.

(f) $\ell_0 = 100$.

(g) $\ell_0 = 40$.

(h) $\ell_0 = 100$.

Fig. 12. Results for surrogate data: the figure shows the expected value $\mathrm{E}[\hat{\sigma}_t]$ and $\mathrm{E}[\hat{\rho}]$ and the normalized standard deviation $\bar{\sigma}[\hat{\sigma}_t]$ and $\bar{\sigma}[\hat{\rho}]$ for same the parameter settings as in Fig. 10, but now with $\sigma_f = 5$Hz. Again, the curves for different δ_t and δ_f are practically coinciding.

reliable.

- The expected value of s_t and p_d does hardly depend on the length ℓ_0. On the other hand, the estimates of s_t and p_d are less biased for larger ℓ_0. The normalized standard deviation of the SES parameters decreases as the length ℓ_0 increases, as expected.

In summary, by means of the SES inference method, one may reliably and robustly determine the timing dispersion s_t and event reliability ρ of pairs of multi-dimensional point processes. We wish to reiterate, however, that it slightly underestimates the timing dispersion and the number of event deletions due to the ambiguity inherent in event synchrony (cf. Fig. 4). Moreover, similarly as in the one-dimensional case, it is critical to choose an appropriate set of initial values $\hat{\delta}_t^{(0)}$, $\hat{\delta}_f^{(0)}$, $\hat{s}_t^{(0)}$, and $\hat{s}_f^{(0)}$.

8 Application: Diagnosis of MCI from EEG

Several clinical studies have shown that the EEG of Alzheimer's disease (AD) patients is generally less coherent than of age-matched control subjects; this is also the case for patients suffering from mild cognitive impairment (see (Jeong, 2004) for a review). In this section, we apply SES to detect subtle perturbations in EEG synchrony of MCI patients.

First we describe the EEG data at hand (Section 8.1), then we describe how we preprocess the EEG, extract bump models, and apply SES (Section 8.2); at last, we present our results (Section 8.3).

8.1 EEG Data

The EEG data used here have been analyzed in previous studies concerning early diagnosis of Alzheimer's disease (AD) (Chapman *et al.*, 2007; Cichocki *et al.*, 2005; Hogan *et al.*, 2003; Musha *et al.*, 2002; Vialatte *et al.*, 2005).

Ag/AgCl electrodes (disks of diameter 8mm) were placed on 21 sites according to 10-20 international system, with the reference electrode on the right earlobe. EEG was recorded with Biotop 6R12 (NEC San-ei, Tokyo, Japan) using analog bandpass filtering in the frequency range 0.5-250Hz at a sampling rate of 200Hz. As in (Chapman *et al.*, 2007; Cichocki *et al.*, 2005; Hogan *et al.*, 2003; Musha *et al.*, 2002; Vialatte *et al.*, 2005), the signals were then digitally band pass filtered between 4 and 30Hz using a third-order Butterworth filter.

The subjects comprised two study groups. The first consisted of a group of 25 patients who had complained of memory problems. These subjects were then diagnosed as suffering from mild cognitive impairment (MCI) and subsequently developed mild AD. The criteria for inclusion into the MCI group were a mini mental state exam (MMSE) score = 24, though the average score in the MCI group was 26 (SD of 1.8). The other group was a control set consisting of 56 age-matched, healthy subjects who had no memory or other cognitive impairments. The average MMSE of this control group was 28.5 (SD of 1.6). The ages of the two groups were 71.9 ± 10.2 and 71.7 ± 8.3, respectively. Finally, it should be noted that the MMSE scores of the MCI subjects studied here are quite high compared to a number of other studies. For example, in (Hogan *et al.*, 2003) the inclusion criterion was MMSE = 20, with a mean value of 23.7, while in (Chapman *et al.*, 2007), the criterion was MMSE = 22; the mean value was not provided. The disparity in cognitive ability between the MCI and control subjects was thus comparatively small, making the present classification task relatively difficult.

All recording sessions were conducted with the subjects in an awake but resting state with eyes closed; the EEG technicians prevented the subjects from falling asleep (vigilance control). After recording, the EEG data has been carefully inspected. Indeed, EEG recordings are prone to a variety of artifacts, for example due to electronic smog, head movements, and muscular activity. The EEG data has been investigated by three EEG experts independently. EEG segments were considered as artifact-free if all three experts agreed. Only those subjects were retained in the analysis whose EEG recordings contained at least 20s of artifact-free data. Based on this requirement, the number of subjects in the two groups described above was further reduced to 22 and 38, respectively. From each subject, one EEG segment of 20s was analyzed (for each of the 21 channels).

8.2 Methods

We successively apply the following transformations to the EEG signals:

(1) wavelet transform,
(2) normalization of the wavelet coefficients,
(3) bump modeling of the normalized wavelet representation,
(4) aggregation of the resulting bump models in several regions.

Eventually, we compute the SES parameters for each pair of aggregated bump models. In the following, we detail each of those five operations.

8.2.1 Wavelet Transform

In order to extract the oscillatory patterns in the EEG, we apply a wavelet transform. More specifically, we use the complex Morlet wavelets (Goupillaud *et al.*, 1984; Delprat *et al.*, 1992):

$$\psi(t) = A \exp\Big(-t^2/2\sigma_0^2\Big) \exp(2i\pi f_0 t), \tag{42}$$

where t is time, f_0 is frequency, σ_0 is a (positive) real parameter, and A is a (positive) normalization factor. The Morlet wavelet (42) has proven to be well suited for the time-frequency analysis of EEG (see (Tallon-Baudry *et al.*, 1996; Herrmann *et al.*, 2005)). The product $w_0 = 2\pi f_0 \cdot \sigma_0$ determines the number of periods in the wavelet ("wavenumber"). This number should be sufficiently large (≥ 5), otherwise the wavelet $\psi(t)$ does not fulfill the admissibility condition:

$$\int \frac{|\psi(t)|^2}{t} dt < \infty, \tag{43}$$

and as a result, the temporal localization of the wavelet becomes unsatisfactory (Goupillaud *et al.*, 1984; Delprat *et al.*, 1992). In the present study, we choose a wavenumber $w_0 = 7$, as in the earlier studies (Tallon-Baudry *et al.*, 1996; Vialatte *et al.*, 2007); this choice yields good temporal resolution in the frequency range we consider in this study.

The wavelet transform $x(t, s)$ of an EEG signal $x(t)$ is obtained as:

$$x(t, s) \triangleq \sum_{t'=1}^{K} x(t')\, \psi^*\Big(\frac{t'-t}{s}\Big), \tag{44}$$

where $\psi(t)$ is the Morlet "mother" wavelet (42), s is a scaling factor, and $K = f_s T$, with f_s the sampling frequency and T the length of the signal. For the EEG data at hand, we have $T = 20s$ and $f_s = 200\text{Hz}$ and hence $K = 4000$. The scaled and shifted "daughter" wavelet in (44) has center frequency $f \triangleq f_0/s$. In the following, we will use the notation $x(t, f)$ instead of $x(t, s)$.

Next we compute the squared magnitude $s(t, f)$ of the coefficients $x(t, f)$:

$$s(t, f) \triangleq |x(t, f)|^2. \tag{45}$$

Intuitively speaking, the time-frequency coefficients $s(t, f)$ represents the energy of oscillatory components with frequency f at time instances t. It is noteworthy that $s(t, f)$ contains no information about the phase of that component.

It is well known that EEG signals have very non-flat spectrum with an overall 1/f shape, besides state-dependent peaks at specific frequencies. Therefore, the map $s(t, f)$ contains most energy at low frequencies f. If we directly apply bump modeling to the map $s(t, f)$, most bumps would be located in the low-frequency range, in other words, the high-frequency range would be under-represented. Since relevant information might be contained at high frequency, we normalize the map $s(t, f)$ before extracting the bump models.

We wish to point out that the time-frequency map $s(t, f)$ may be determined by alternative methods. For example, one may compute $s(t, f)$ by the multi-taper method (Thomson *et al.*, 1982) or by filterbanks (Harris *et al.*, 2004). We decided to use the Morlet wavelet transformation for two reasons:

- Morlet wavelets have the optimal joint time-frequency resolution. We remind the reader of the fact that the joint time-frequency resolution is fundamentally limited by the uncertainty principle: the resolution in both time and frequency cannot be arbitrarily high *simultaneously*. It is well known that the Morlet wavelets achieve the uncertainty relation with *equality* (Goupillaud *et al.*, 1984; Delprat *et al.*, 1992; Mallat, 1999).
- EEG signals are typically highly non-stationary; the wavelet transform is ideally suited for non-stationary signals (Mallat, 1999), in contrast to approaches based on multitapers and filterbanks.

8.2.2 Normalization

The coefficients $s(t, f)$ are centered and normalized, resulting in the coefficients $\tilde{z}(t, f)$:

$$\tilde{z}(t, f) \triangleq \frac{s(t, f) - m_s(f)}{\sigma_s(f)}, \tag{46}$$

where $m_s(f)$ is obtained by averaging $s(t, f)$ over the whole length of the EEG signal:

$$m_s(f) = \frac{1}{K} \sum_{t=1}^{K} s(t, f). \tag{47}$$

Likewise, $\sigma_s^2(f)$ is the variance of $s(t, f)$:

$$\sigma_s^2(f) = \frac{1}{K} \sum_{t=1}^{K} \Big(s(t, f) - m_s(f) \Big)^2. \tag{48}$$

In words: the coefficients $\tilde{z}(t, f)$ encode fluctuations from the baseline EEG power at time t and frequency f. The normalization (46) is known as z-score (see, e.g., (Buzsáki, 2006)), and is commonly applied (Matthew *et al.*, 2002; Martin *et al.*, 2004; Ohara *et al.*, 2004; Vialatte *et al.*, 2007; Chen *et al.*, 2007). The coefficients $\tilde{z}(t, f)$ are positive when the activity at t and f is stronger than the baseline $m_s(f)$ and negative otherwise.

There are various approaches to apply bump modeling to the z-score $\tilde{z}(t, f)$.

One may first set the negative coefficients to zero, and next apply bump modeling. The bump models in that case represent peak activity. Alternatively, one may first set the positive coefficients equal to zero, reverse the sign of the negative coefficients, and then apply bump modeling. In that case, the bump models represent dips in the energy maps $s(t, f)$.

In the application of diagnosing AD (see Section 8), we will follow yet another approach. In order to extract bump models, we wish to exploit as much information as possible from the $\tilde{z}$ maps. Therefore we will set only a small fraction of the coefficients $\tilde{z}(t, f)$ equal to zero, i.e., the 1% smallest coefficients. This approach was also followed in (Vialatte *et al.*, 2007), and is equivalent to the following transformation: we shift the coefficients (46) in the positive direction by adding a constant α, the remaining negative coefficients are set to zero:

$$z(t, f) \triangleq \left\lceil \tilde{z}(t, f) + \alpha \right\rceil^{+} = \left\lceil \frac{s(t, f) - m_s(f)}{\sigma_s(f)} + \alpha \right\rceil^{+}, \tag{49}$$

where $\lceil x \rceil^{+} = x$ if $x \geq 0$ and $\lceil x \rceil^{+} = 0$ otherwise. The constant α is chosen such that only 1% of the coefficients remains negative after addition with α; this corresponds to $\alpha = 3.5$ in the application of diagnosing AD (see Section 8). (In the study of (Vialatte *et al.*, 2007), it corresponds to $\alpha = 2$.) The top row of Fig. 1 shows the normalized wavelet map z (49) of two EEG signals.

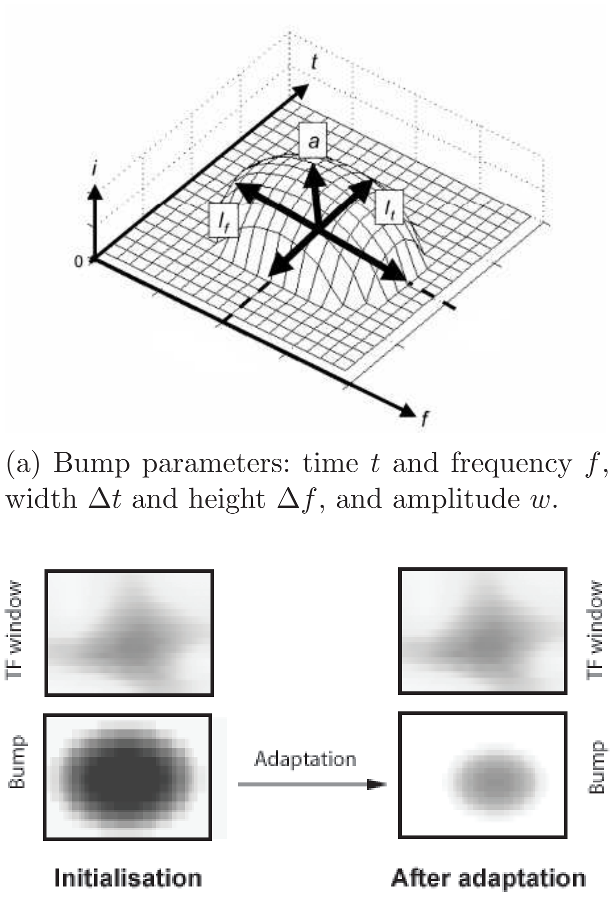
8.2.3 Bump Modeling

Next, bump models are extracted from the coefficient maps z (see Fig. 1 and (Vialatte *et al.*, 2007)). We approximate the map $z(t, f)$ as a sum $z_{\text{bump}}(t, f, \theta)$ of a "small" number of smooth basis functions or "bumps" (denoted by f_{bump}):

$$z(t, f) \approx z_{\text{bump}}(t, f, \theta) \triangleq \sum_{k=1}^{N_b} f_{\text{bump}}(t, f, \theta_k), \tag{50}$$

where θ_k are vectors of bump parameters and $\theta \triangleq (\theta_1, \theta_2, \ldots, \theta_{N_b})$. The sparse bump approximation $z_{\text{bump}}(t, f, \theta)$ represents regions in the time-frequency plane where the EEG contains more power than the baseline; in other words, it captures the most significant oscillatory activities in the EEG signal.

We choose half-ellipsoid bumps since they are well suited for our purposes (Vialatte, 2005; Vialatte *et al.*, 2007) (see Fig. 13). Since we wish to keep the number of bump parameters as low as possible, the principal axes of the half ellipsoid bumps are restricted to be parallel to the time-frequency axes. As a result, each bump is described by five parameters (see Fig. 13(a)): the coordinates of its center (i.e., time t_k and frequency f_k), its amplitude $w_k > 0$, and the extension Δt_k and Δf_k in time and frequency respectively, in other



(a) Bump parameters: time t and frequency f, width Δt and height Δf, and amplitude w.

(b) Learning the bump parameters by minimizing the quadratic cost function (51); Top (left and right): a given patch of the time-frequency map. Bottom left: initial bump; Bottom right: bump obtained after adaptation.

Fig. 13. Half ellipsoid bump.

words, $\theta_k = (t_k, f_k, w_k, \Delta t_k, \Delta f_k)$. More precisely, the ellipsoid bump function $f_{\text{bump}}(t, f, \theta_k)$ is defined as:

$$f_{\text{bump}}(t, f, \theta_k) = \begin{cases} w_k \sqrt{1 - \kappa(t, f, \theta_k)} & \text{for } 0 \leq \kappa(t, f, \theta_k) \leq 1 \\ 0 & \text{for } \kappa(t, f, \theta_k) > 1, \end{cases} \tag{51}$$

where

$$\kappa(t, f, \theta_k) = \frac{(t - t_k)^2}{(\Delta t_k)^2} + \frac{(f - f_k)^2}{(\Delta f_k)^2}. \tag{52}$$

For the EEG data described in Section 8.1, the number of bumps N_b (cf. (50)) is typically between 50 and 100, and therefore, $z_{\text{bump}}(t, f, \theta)$ is fully specified by a few hundred parameters. On the other hand, the time-frequency map $z(t, f)$ consists of between 10^4 and 10^5 coefficients; the bump model $z_{\text{bump}}(t, f, \theta)$ is thus a sparse (but approximate) representation of $z(t, f)$.

The bump model $z_{\text{bump}}(t, f, \theta)$ is extracted from $z(t, f)$ by the following algorithm (Vialatte, 2005; Vialatte *et al.*, 2007):

(1) Define appropriate boundaries for the map $z(t, f)$ in order to avoid finite-size effects.
(2) Partition the map $z(t, f)$ into small zones. The size of these zones depends on the time-frequency ratio of the wavelets, and are optimized to model oscillatory activities lasting 4 to 5 oscillation periods. Larger oscillatory patterns are modeled by multiple bumps.
(3) Find the zone $\mathcal{Z}$ that contains the most energy.
(4) Adapt a bump to that zone; the bump parameters are determined by minimizing the quadratic cost function (see Fig. 13(b)):

$$\mathcal{E}(\theta_k) \triangleq \sum_{t,f \in \mathcal{Z}} \Big(z(t, f) - f_{\text{bump}}(t, f, \theta_k) \Big)^2. \tag{53}$$

Next withdraw the bump from the original map.

(5) The fraction of total intensity contained in that bump is computed:

$$F = \frac{\sum_{t,f \in \mathcal{Z}} f_{\text{bump}}(t, f, \theta_k)}{\sum_{t,f \in \mathcal{Z}} z(t, f)}. \tag{54}$$

If $F < G$ for three consecutive bumps (and hence those bumps contain only a small fraction of the energy of map $z(t, f)$), stop modeling and proceed to (6), otherwise iterate (3).

(6) After all signals have been modeled, define a threshold $T \geq G$, and remove the bumps for which $F < T$. This allows us to trade off the information loss and modeling of background noise: when too few bumps are generated, information about the oscillatory activity of the brain is lost. On the other hand, if too many bumps are generated, the bump model also contains low-amplitude oscillatory components; since the measurement process typically introduces a substantial amount of noise, it is likely that the low-amplitude oscillatory components do not stem from organized brain oscillations but are instead due measurement noise. By adjusting the threshold T, we try to find an appropriate number of bumps.

In the present application, we used a threshold $G = 0.05$. With this threshold, each bump model contains many bumps. Some of those bumps may actually model background noise. Therefore, we further pruned the bump models (cf. Step 6). We tested various values of the threshold $T \in [0.2, 0.25]$; as we will show, the results depend on the specific choice of T: the optimal separation between MCI and age-matched control subjects is obtained for $T = 0.22$, the separation gradually diminishes for increasing and decreasing values of T. We refer to (Vialatte, 2005; Vialatte *et al.*, 2007) for more information on bump

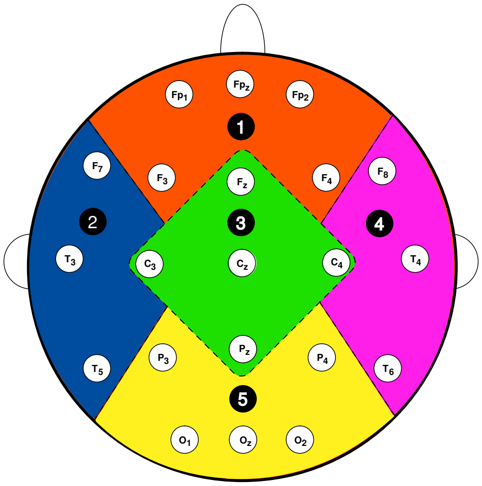


Fig. 14. The 21 electrodes used for EEG recording, distributed according to the 10–20 international placement system (Nunez *et al.*, 2006). The clustering into N_R = 5 zones is indicated by the colors and dashed lines (1 = frontal, 2 = left temporal, 3 = central, 4 = right temporal and 5 = occipital).

modeling. In particular, we used the same choice of boundaries (Step 1) and partitions (Step 2) as in those references.

Eventually, we obtain 21 bump models, i.e., one per EEG channel. In the following, we describe how those models are further processed.

8.2.4 Aggregation

As a next step, we group the 21 electrodes into a small number N_R of regions, as illustrated in Fig. 14 for $N_R = 5$; we will report results for $N_R = 3$, 5, and 7. From the 21 bump models obtained by sparsification (cf. Section 8.2.3), we extract a single bump model for each of the zones by means of the aggregation algorithm described in (Vialatte *et al.*, 2007).

8.2.5 Stochastic Event Synchrony

Aggregation vastly reduces the computational complexity: instead of computing the SES parameters between all possible pairs of 21 electrodes (210 in total), we compute those parameters for all pairs of regions, i.e., $N_R(N_R-1)/2$ pairs in total. In addition, in order to obtain measures for *average* synchrony, we average the SES parameters over all region pairs, resulting in one set of average SES parameters per subject. It is noteworthy that in this setting, the SES parameters quantify *large-scale* synchrony, since each region spans several tens of millimeters. In the following, we will only consider ρ and s_t, since those two parameters are the most relevant.

We choose the parameters of the SES algorithm as follows. Since we are dealing with spontaneous EEG, it is unlikely that the EEG signals from certain channels are delayed w.r.t. other channels; moreover, systematic frequency offsets are unrealistic. Therefore, we choose the initialization $\hat{\delta}_t^{(0)} = 0 = \hat{\delta}_f^{(0)}$. We used the parameter settings $\hat{s}_t^{(0)} = s_{0,t} = 0.15, 0.175, \ldots, 0.25$ and $\hat{s}_f^{(0)} = s_{0,f} = 0.025, 0.050, \ldots, 0.15$. We will show results for all those parameter values. The parameters ν_t and ν_f are set equal to 100, which corresponds to priors for s_t and s_f that have a sufficiently wide support (cf. Fig. 6). We have observed that smaller values of ν_t and ν_f are not satisfactory (e.g., $\nu_t = 50 = \nu_f$): the prior takes non-negligible values for large values of s_t and s_f, which leads to prohibitively large and unrealistic offsets in time and frequency. Larger values of ν_t and ν_f are not satisfactory either, since the priors for s_t and s_f then become too informative and would strongly bias the parameter estimates.

8.3 Results

The main results are summarized in Fig. 15 and 16; they contain p-values obtained by the Mann-Whitney test for the parameters ρ and s_t respectively. This test indicates whether the parameters take different values for the two subject populations. More precisely, low p-values indicate large difference in the medians of the two populations. The p-values are shown for $\hat{s}_t^{(0)} = s_{0,t} = 0.15, 0.175, \ldots, 0.25$, $\hat{s}_f^{(0)} = s_{0,f} = 0.025, 0.050, \ldots, 0.15$, $\beta = 0.01$, 0.001, 0.0001, $T = 0.2$, 0.21, …, 0.25, and the number of zones $N_R = 3$, 5, and 7, with $\nu_t = 100 = \nu_f$.

The lowest p-values for ρ are obtained for $T = 0.22$ and $N_R = 5$ (see Fig. 19(e)). In particular, the smallest value is p $= 1.2 \cdot 10^{-4}$, which occurs for $\beta = 0.001$, $\hat{s}_t^{(0)} = s_{0,t} = 0.225$, and $\hat{s}_f^{(0)} = s_{0,f} = 0.05$.

It is interesting that the results depend on T (cf. Section 8.2.3). That parameter allows us to balance the information loss and modeling of background noise: when too few bumps are generated, information about the oscillatory activity of the brain is lost. On the other hand, if too many bumps are generated, the bump model also contains low-amplitude oscillatory components. The p-values are the lowest for $T = 0.22$, and become gradually larger as T decreases from $T = 0.22$ to 0.2 and as T increases from $T = 0.22$ to 0.25. One explanation could be that the number of bumps in each bump model is significantly smaller for MCI patients than in control subjects, with the maximum difference at $T = 0.22$; if the bump models of MCI patients contained fewer bumps, it would be intrinsically harder to align those models. However, as Fig. 17 shows, this is not the case: On average, the bump models of MCI patients contain fewer bumps than the models of control subjects, but the difference is only weakly significant at best. Moreover, the largest difference does not consistently occur at $T = 0.22$. In other words, the difference in number of bumps between both subject populations cannot explain the dependency of the p-values on T.

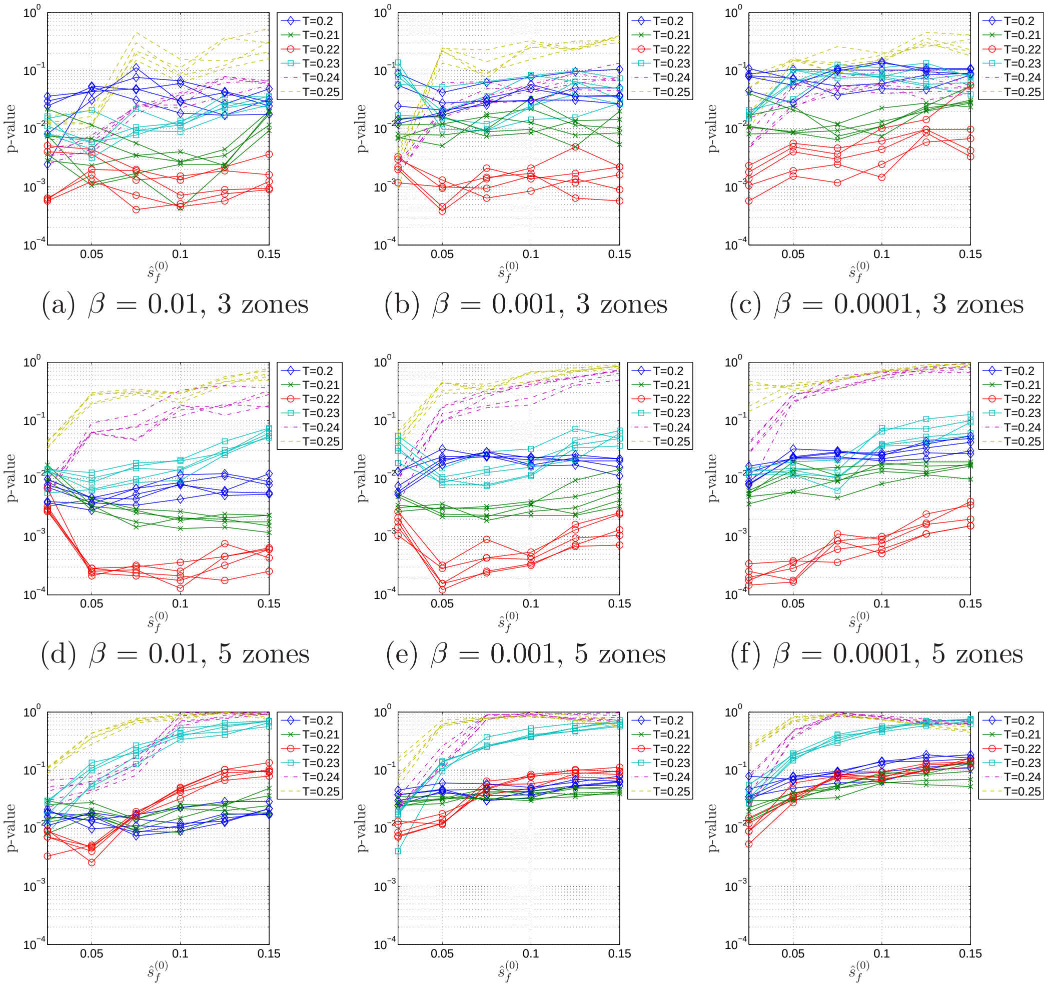
This seems to suggest an alternative explanation: at $T = 0.22$, the optimal trade off between information loss and modeling of background noise occurs. At lower values of T, the bump models contain more background noise, i.e., components that are unrelated to oscillatory events in the brain signals, and therefore, the statistical differences between both populations decrease. At higher values of T, the models capture fewer oscillatory events in the brain signals, and therefore, important information to distinguish both populations is discarded; the estimated parameters become less reliable.

From Fig. 15, we can conclude that the statistical differences in ρ are highly significant, especially for $T = 0.22$ and $N_R = 5$: There is a significantly higher degree of non-correlated activity in MCI patients, more specifically, a high number of non-coincident, non-synchronous oscillatory events. Interestingly, we did not observe a significant effect on the timing jitter s_t of the coincident events (see Fig. 16): very few p-values for s_t are smaller than 0.01, which suggests there are no significant differences in s_t. In other words, MCI seems to be associated with a significant increase of non-coincident background activity, while the *coincident* activity remains well synchronized. For the sake of clarity, Fig. 18 shows boxplots for ρ and s_t, for the parameter setting that leads to the lowest p-values for ρ, i.e., $T = 0.22$, $N_R = 5$, $\beta = 0.001$, $\hat{s}_t^{(0)} = s_{0,t} = 0.225$, and $\hat{s}_f^{(0)} = s_{0,f} = 0.05$.

We now discuss how the p-values for ρ depend on $\hat{s}_t^{(0)} = s_{0,t}$ and $\hat{s}_f^{(0)} = s_{0,f}$. Fig. 19 shows those p-values for $\hat{s}_t^{(0)} = s_{0,t} = 0.025, 0.050, \ldots, 0.25$, $\hat{s}_f^{(0)} = s_{0,f} = 0.025, 0.050, \ldots, 0.15$ $\beta = 0.01$, 0.001, 0.0001, with $T = 0.22$, $N_R = 5$, and $\nu_t = 100 = \nu_f$. Note that, in order not to clutter the figures, we only show results for $\hat{s}_t^{(0)} = s_{0,t} = 0.15, 0.175, \ldots, 0.25$ in Fig. 15 and 16; Fig. 19 shows in addition results for $\hat{s}_t^{(0)} = s_{0,t} = 0.025, 0.050, \ldots, 0.125$.

When both $\hat{s}_t^{(0)} = s_{0,t}$ and $\hat{s}_f^{(0)} = s_{0,f}$ are smaller or equal 0.75, the fraction of non-matched events is usually about 70-80% (not shown here), and pairs of events that are close in time and frequency are not always matched. In other words, the obtained solutions are not satisfactory for those values of $\hat{s}_t^{(0)} = s_{0,t}$ and $\hat{s}_f^{(0)} = s_{0,f}$.

The smallest p-values occur typically for $\hat{s}_t^{(0)} > 0.15$ and $\hat{s}_f^{(0)} < 0.1$. This is in agreement with our expectations: As we argued in Section 3, we expect bumps to appear at about the same frequency in both time-frequency maps,



(a) $\beta = 0.01$, 3 zones (b) $\beta = 0.001$, 3 zones (c) $\beta = 0.0001$, 3 zones

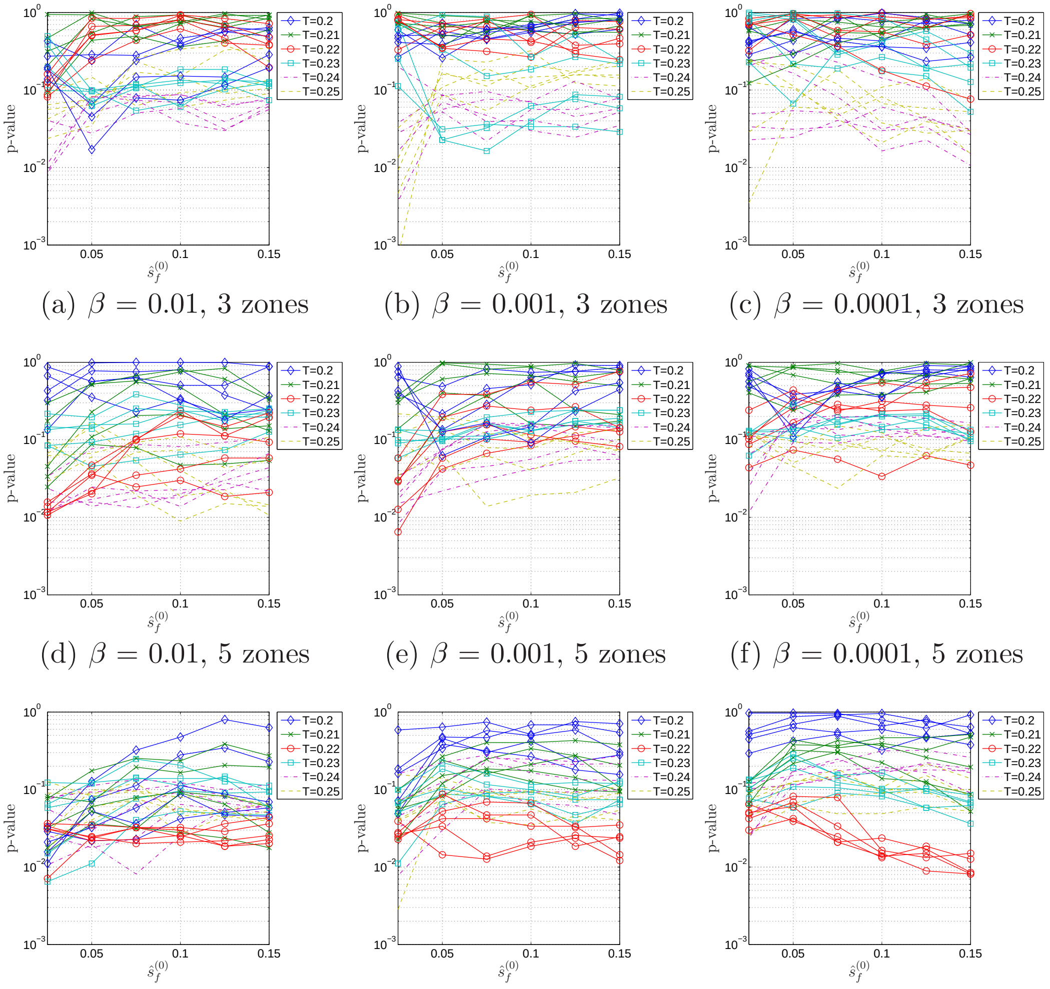
(d) $\beta = 0.01$, 5 zones (e) $\beta = 0.001$, 5 zones (f) $\beta = 0.0001$, 5 zones

(g) $\beta = 0.01$, 7 zones (h) $\beta = 0.001$, 7 zones (i) $\beta = 0.0001$, 7 zones

Fig. 15. p-values obtained by the Mann-Whitney test for the parameter ρ for $\hat{s}_t^{(0)} = s_{0,t} = 0.15, 0.175, \ldots, 0.25$, $\hat{s}_f^{(0)} = s_{0,f} = 0.025, 0.050, \ldots, 0.15$, $\beta = 0.01$, 0.001, 0.0001, $T = 0.2$, 0.21, ..., 0.25 and the number of zones $N_R = 3$, 5, and 7, with $\nu_t = 100 = \nu_f$. The p-values seem to vary little with $s_t^{(0)}$, $s_f^{(0)}$ and β, but are more dependent on T and the number of zones. The lowest p-values are obtained for $T = 0.22$ and 5 zones; the corresponding statistical differences are highly significant.

since frequency shifts are hard to justify from a physiological perspective, whereas timing jitter arises quite naturally.

We verified that the SES measures ρ and s_t are not correlated with other synchrony measures, e.g., Pearson correlation coefficient, magnitude and phase coherence, phase synchrony etc. (Pearson r, $p > 0.10$; see (Dauwels *et al.*, 2008) for more details). In contrast to the classical measures, SES quantifies the synchrony of oscillatory events instead of more conventional amplitude or phase synchrony, therefore, it provides complementary information about EEG synchrony.



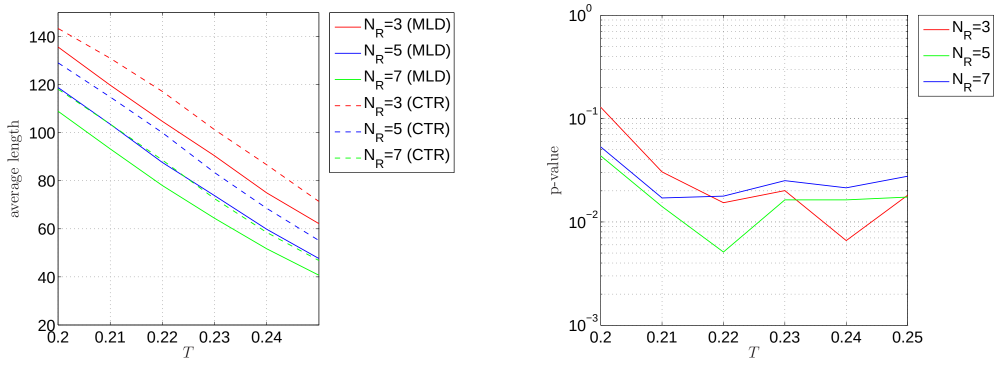
(a) $\beta = 0.01$, 3 zones (b) $\beta = 0.001$, 3 zones (c) $\beta = 0.0001$, 3 zones

(d) $\beta = 0.01$, 5 zones (e) $\beta = 0.001$, 5 zones (f) $\beta = 0.0001$, 5 zones

(g) $\beta = 0.01$, 7 zones (h) $\beta = 0.001$, 7 zones (i) $\beta = 0.0001$, 7 zones

Fig. 16. p-values obtained by the Mann-Whitney test for the parameter s_t for $\hat{s}_t^{(0)} = s_{0,t} = 0.15, 0.175, \ldots, 0.25$, $\hat{s}_f^{(0)} = s_{0,f} = 0.025, 0.050, \ldots, 0.15$, $\beta = 0.01$, 0.001, 0.0001, $T = 0.2$, 0.21, $\ldots$, 0.25 and the number of zones $N_R = 3$, 5, and 7, with $\nu_t = 100 = \nu_f$. Very few p-values are smaller than 0.01, which suggests there are no significant differences in s_t.

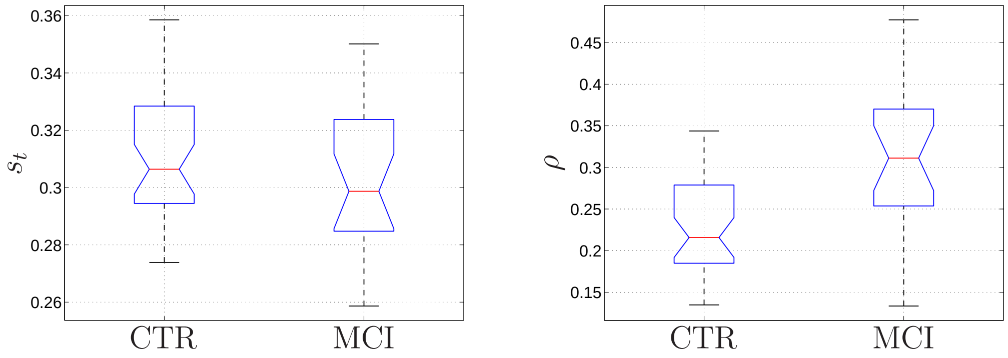
We applied a variety of classical synchrony measures to the same EEG data set (Dauwels *et al.*, 2008). Most measures yield (weakly) significantly different values for the MCI and control subjects, some differences are highly significant; the most significant results were obtained with the the full-frequency direct transfer function (ffDTF), which is a Granger measure (Pereda *et al.*, 2005), resulting in a p-value of about 10^{-3} (Mann-Whitney test). We combined ρ with ffDTF as features to distinguish MCI from control subjects (see Fig.20). We used the parameter setting of the SES algorithm that leads to the smallest p-value for ρ (p $= 1.2 \cdot 10^{-4}$); we verified that all parameter settings with $T = 0.22$ and $N_R = 5$ yields about the same classification results. About 85% of the subjects are correctly classified, which is a promising result. However, it is too weak to allow us to predict AD reliably. To this end, we would need



(a) Average number of bumps

(b) p-values

Fig. 17. Average number of bumps in each bump model, for MCI and control subjects; average number (left) and p-values obtained by the Mann-Whitney test (right).



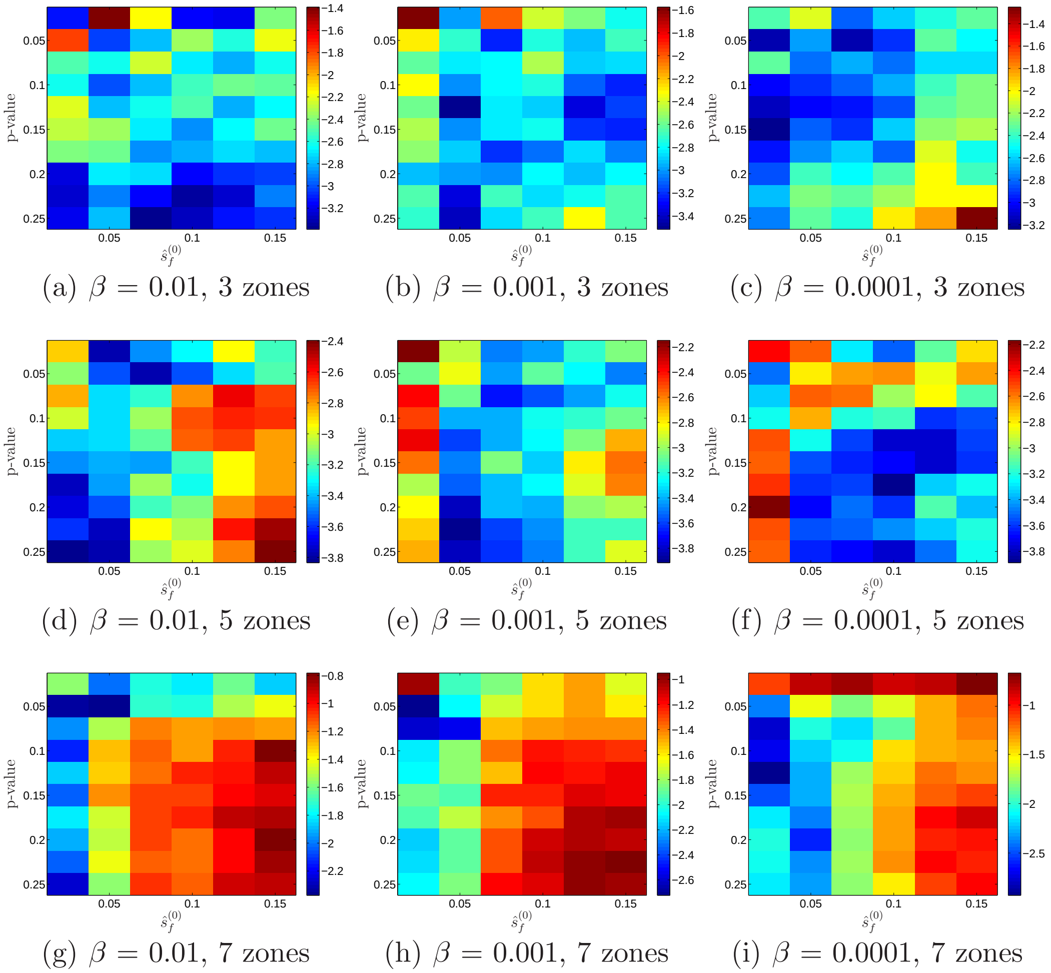
(a) s_t (p = 0.19)

(b) ρ (p = 0.00012)

Fig. 18. Box plots of s_t and ρ , for MCI and control subjects, with $T = 0.22$, $N_R = 5$, $\beta = 0.001$, $\hat{s}_t^{(0)} = s_{0,t} = 0.225$, and $\hat{s}_f^{(0)} = s_{0,f} = 0.05$. Interestingly, the parameter ρ leads to highly significant differences (p = 0.00012), in contrast to the parameter s_t (p = 0.19).

to combine those two synchrony measures with complementary features, for example, derived from the slowing effect of MCI on EEG, or perhaps from different modalities such as PET, MRI, or biochemical indicators. We wish to point out, however, that in the data set at hand, patients did not carry out any specific task. In addition, we considered recordings of 20s, which are rather short. It is plausible that the sensitivity of EEG synchrony could be further improved by increasing the length of the recordings and by recording the EEG before, while, and after patients carry out specific tasks, e.g., working memory tasks. As such, the classifier displayed in Fig. 20 might be applied to screen a population for MCI, since it only requires an EEG recording system. The latter is a relatively simple and low-cost technology, at present available in most hospitals.

We tried to verify whether the small p-value of ρ is due to a decrease in



(a) $\beta = 0.01$, 3 zones (b) $\beta = 0.001$, 3 zones (c) $\beta = 0.0001$, 3 zones

(d) $\beta = 0.01$, 5 zones (e) $\beta = 0.001$, 5 zones (f) $\beta = 0.0001$, 5 zones

(g) $\beta = 0.01$, 7 zones (h) $\beta = 0.001$, 7 zones (i) $\beta = 0.0001$, 7 zones

Fig. 19. p-values (Mann-Whitney test) for the parameter ρ for $\hat{s}_t^{(0)} = s_{0,t} = 0.025, 0.050, \ldots, 0.25$, $\hat{s}_f^{(0)} = s_{0,f} = 0.025, 0.050, \ldots, 0.15$ $\beta = 0.01$, 0.001, 0.0001, with $T = 0.22$, $N_R = 5$, and $\nu_t = 100 = \nu_f$.

coincident oscillatory events or whether it can be attributed to an effect not related to synchrony or perhaps to an artifact. To this end, we generated and investigated surrogate data. From a given bump model, we obtain a surrogate bump model by shuffling the bumps over time: the center t_k of the bumps is chosen randomly, more precisely, it is drawn uniformly over the support of the bump model, the other bump parameters are kept fixed. We created 1000 such bump models for each subject, and obtained as a result 1000 surrogate EEG data sets. The distribution of the p-values of ρ for those 1000 surrogates is shown in Fig. 21. The p-value of ρ for the actual EEG data set (p = 0.00012) is indicated by a cross. All the surrogates yielded p-values larger than 0.00012. We interpret this result as follows. If the p-values of the surrogate data were on average about 0.00012, we would be able to conclude that synchrony alone cannot explain the observed significant decrease in ρ. Since the p-values of the surrogates are on average much larger than 0.00012, it is less likely that other effects besides decrease of coincident neural activity result in the lower ρ in

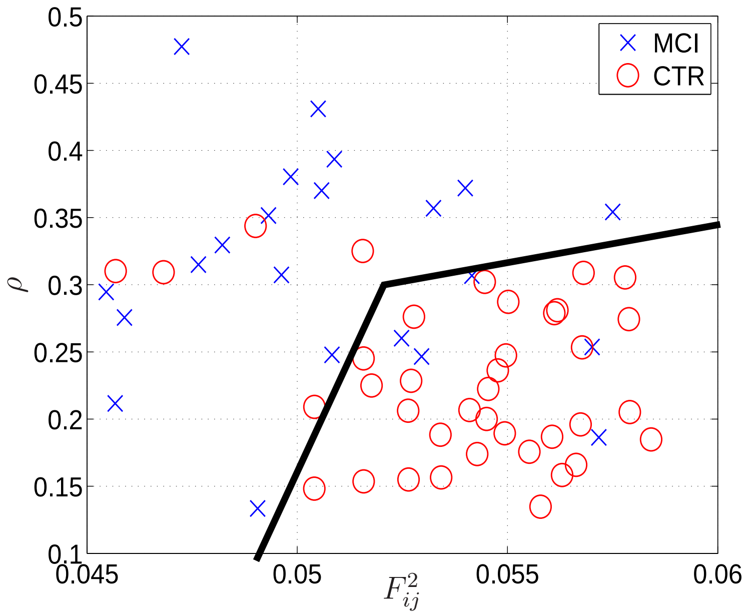
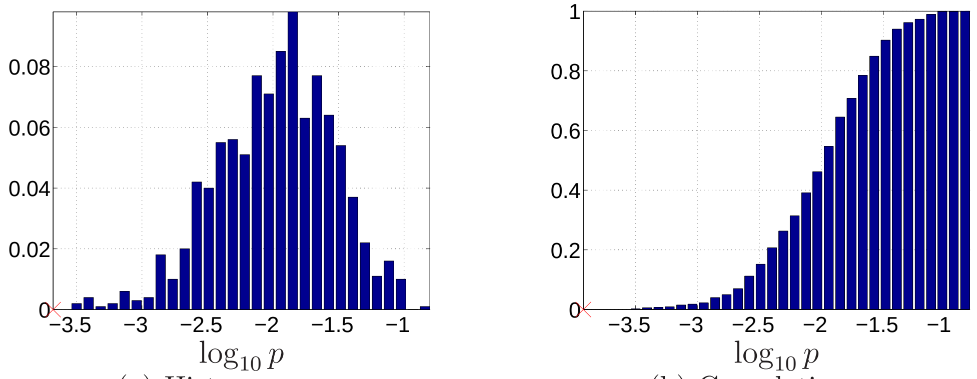


Fig. 20. Combining ρ with ffDTF as features to distinguish MCI from age-matched control subjects. Note that ffDTF is a similarity measure whereas ρ is a dissimilarity measure. The (ffDTF, ρ) pairs of the MCI and control subjects tend towards the left top corner and bottom right corner respectively. The smooth curve (solid) yields a classification rate of 85%.



(a) Histogram

(b) Cumulative

Fig. 21. Distribution of the p-value of parameter ρ for 1000 surrogate EEG data sets. The p-value of ρ for the actual EEG data set (p = 0.00012) is indicated by a cross. All surrogates yielded p-values larger than 0.00012.

MCI patients.

We analyzed the convergence of the proposed inference algorithm (cf. Table 2). A histogram of the number of iterations (Step 1 and 2 in Table 2) required for convergence is shown in Fig. 22, computed over all subjects, all pairs of regions, and all parameter settings. The algorithm converged after at most 23 iterations, and on average, after about four iterations. We allowed a maximum number of 50 iterations, and therefore, Fig. 22 indicates that the algorithm

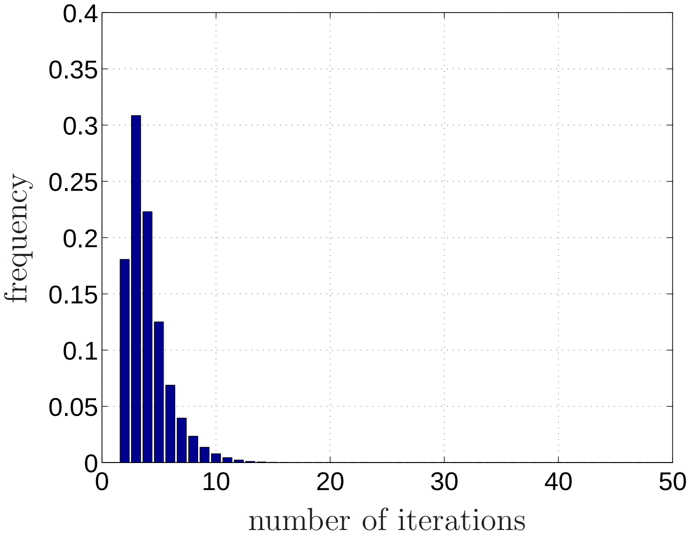


Fig. 22. Histogram of the number of iterations (Step 1 and 2 in Table 2) required for convergence, computed over all subjects and all pairs of regions. The algorithm converged after at most 23 iterations, and on average, after about four iterations. We allowed a maximum number of 50 iterations, and the histogram shows that the algorithm always converged for the EEG data set at hand.

always converged for the EEG data set at hand, as suggested by the theory of Section 5.

Besides the algorithm of Table 2, we also implemented an algorithm in which the alignment (24) is carried out by a linear programming relaxation instead of the max-product algorithm. Since that algorithm is more complicated, we will not describe it here. We observed that both algorithms always converged to the same results. Moreover, since the max-product algorithm always converged in our experiments, we can deduce that the optimal solution of the linear programming relaxation of (24) was every time unique (Bayati *et al.*, 2005; Huang *et al.*, 2007; Bayati *et al.*, 2007; Sanghavi, 2007a,b). Since it is well-known that the linear programming relaxation is tight for bipartite max-weight matching (Gerards, 1995; Pulleyblank, 1995), we can conclude that in our experiments, both the max-product algorithm and linear programming relaxation of (24) resulted in the unique optimal alignment (24).

9 Conclusions

We have presented an alternative method to quantify the similarity of two time series, referred to as stochastic event synchrony (SES). As a first step, one extracts 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 paper

(Part II), we focussed on multi-dimensional point processes.

Through the analysis of surrogate data, we verified that also in the multi-dimensional case, SES can distinguish timing dispersion from event reliability. However, it typically underestimates the timing dispersion and overestimates event reliability; this is due to the ambiguous nature of the synchrony of point processes. The bias tends to be smaller for multi-dimensional point processes than for one-dimensional point processes.

Also in the multi-dimensional case, it is crucial to extract suitable events from the given time series. Only if those events are characteristic for the time series, SES may yield meaningful results. As we have shown, for spontaneous EEG signals, it is natural to consider oscillatory events from the time-frequency representation; in particular, we considered bump models extracted from time-frequency maps of the EEG. However, depending on the nature of the EEG, there might be interesting alternatives, for example based on matching pursuit or chirplets.

Since the proposed similarity measure does not take the entire time series into account but focusses exclusively on certain events, it provides complementary information about synchrony. Therefore, we believe that it may prove to be useful to blend our similarity measure with classical measures such as the Pearson correlation coefficient, Granger causality, or phase synchrony indices. We have shown that such combined approach yields interesting results for the concrete application of diagnosing MCI from EEG: we computed ρ, the fraction of non-matched oscillatory events, and full-frequency directed transfer function (ffDTF) from spontaneous EEG and used those two (dis)similarity measures as features to distinguish MCI from control subjects, resulting in a classification rate of about 85%. Moreover, we observed that there are significantly more non-matched oscillatory events in the EEG of MCI subjects than in control subjects. The timing jitter s_t of the matched oscillatory events, however, is not different in the two subject groups. In future work, we will analyze additional data sets, and incorporate other modalities such as fMRI and DTI into the analysis.

We wish to underline that the SES measures proposed in this paper are only applicable to pairs of signals. However, extensions to an arbitrary number of signals are feasible. Moreover, in the present study, the SES parameters are assumed to be constant; SES may be extended to time-varying parameters. Such extensions will be the subject of future reports.

At last, we wish to outline another potential extension. In the generative process of SES, the events of the hidden point process are sampled independently and uniformly in the space at hand. However, in some applications, those events may naturally occur in clusters. More generally, the events may be statistically dependent. For example, it has been shown that specific frequency bands in EEG are sometimes coupled. Such couplings lead to correlations between the bumps in time-frequency domain. Our current analysis ignores such correlations. By taken those dependencies into account, we may be able to further improve our classification results; moreover, it may lead to further insights about MCI and AD.

Acknowledgments

Results of this paper were in part reported in (Dauwels *et al.*, 2007). The authors wish to 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. They are grateful to participants of the Retreat of the MIT Picower Center for Learning and Memory (May 2007, Cape Cod, MA, US), the RIKEN Symposium "Brain Activity and Information Integration" (September 2007, RIKEN, Saitama, Japan), and the NIPS Workshop "Large-Scale Brain Dynamics" (December 2007, Whistler, Canada) for numerous inspiring questions and comments. The authors also acknowledge the anonymous reviewers for helpful comments and Dr. Monique Maurice (RIKEN) for designing several figures in this paper. 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.

A Appendix: Factor Graphs

In this appendix, we provide some basic information on graphical models, in particular, factor graphs. We will closely follow (Loeliger, 2004; Loeliger *et al.*, 2007). Graphical models are graphical representations of multivariate functions. Examples of graphical models are Markov random fields (or "Markov networks"), Bayesian networks (or "belief networks"), and factor graphs (Loeliger, 2004; Jordan, 1999; Loeliger *et al.*, 2007). We use factor graphs in this paper, more specifically, Forney-style factor graph or "normal" graphs, since they are more flexible than other types of graphical models; moreover, the sum-product and max-product algorithm can be formulated most easily in the factor graph notation (see (Loeliger, 2004) for a more detailed argumentation).

As already mentioned before, graphical models (and factor graphs in particular) represent functions. Let us have a look at some examples.

Example A.1 (Factor graph of a function without structure)
The factor graph of the function $f(x_1, x_2, x_3)$ is shown in Fig. A.1 (left): edges represent variables, and nodes represent factors. An edge is connected to a node if and only if the corresponding variable is an argument of the corresponding function. □

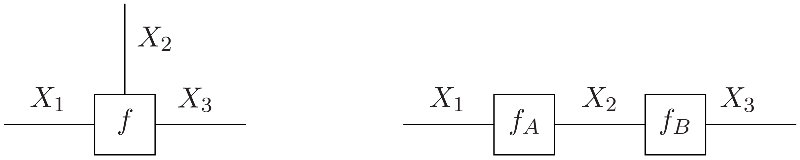


Fig. A.1. Factor graph of function without structure, i.e., $f(x_1, x_2, x_3)$ (left) and a function with structure, i.e., $f(x_1, x_2, x_3) \triangleq f_A(x_1, x_2) f_B(x_2, x_3)$ (right).

The concept of factor graphs becomes interesting as soon as the function has structure, i.e., when it factors.

Example A.2 (Factor graph of a function with structure)
Let us assume that the function $f(x_1, x_2, x_3)$ of Example A.1 factors as $f(x_1, x_2, x_3) \triangleq f_A(x_1, x_2) f_B(x_2, x_3)$; the factor graph of Fig. A.1 (right) represents this factorization. We call f the global *function and f_A and f_B* local *functions.* □

Example A.3 *The (global) function*

$$f(x_1, x_2, x_3, x_4, x_5, x_6) \triangleq f_A(x_1, x_2) f_B(x_3, x_4) f_C(x_2, x_4, x_5) f_D(x_5, x_6) \tag{A.1}$$

is represented by the factor graph in Fig. A.2 □

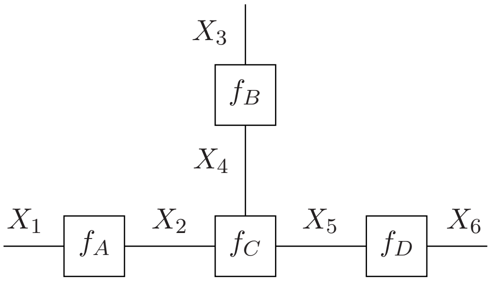


Fig. A.2. An example factor graph, representing the function (A.1). Each node corresponds to a factor in that function, i.e., f_A, f_B, f_C, and f_D, each edge corresponds to a variable, i.e., X_1, X_2, ..., X_6.

More formally, a Forney-style factor graph (FFG) is defined as follows:

- **Factor graph:** An FFG represents a function f and consists of nodes and edges. We assume that f can be written as a product of factors.
- **Global functions:** The function f is called the global function.
- **Nodes/local functions:** There is a node for every factor, also called local function.
- **Edges/variables:** There is an edge or half-edge for every variable.
- **Connections:** An edge (or half-edge) representing some variable X is connected to a node representing some factor f if and only if f is a function of X.
- **Configuration:** A configuration is a particular assignment of values to all variables. We use capital letters for unknown variables and small letters for particular values. This imitates the notation used in probability theory to denote chance/random variables and realizations thereof.
- **Configuration space:** The configuration space Ω is the set of all configurations: it is the domain of the global function f. One may regard the variables as functions of the configuration ω, just as we would with random/chance variables.
- **Valid configuration:** A configuration $\omega \in \Omega$ will be called valid if $f(\omega) \neq 0$.

Implicit in the previous definition is the assumption that no more than two edges are connected to one node. This restriction is easily circumvented by introducing variable replication nodes (also referred to as "equality constraint nodes"). An equality constraint node represents the factorization $\delta(x-x')\delta(x'-x'')$, and is depicted in Fig. A.4 (left). It enforces the equality of the variables X, X', and X''. The (single) equality constraint node generates two replicas of X, i.e., X' and X''. If more replicas are required, one can concatenate single nodes as shown in Fig. A.4 (middle); combining those single nodes leads to a compound equality constraint node (see Fig. A.4 (right)).

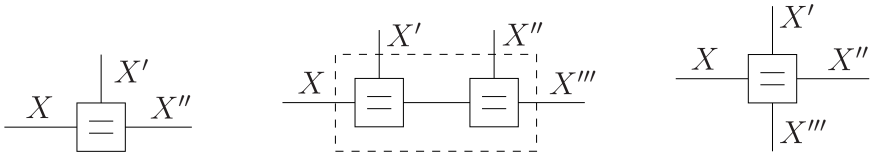


Fig. A.4. Equality constraint node used for variable replication. (left) single node; (right) compound node; (middle) the compound node as concatenation of single nodes.

B Appendix: Summary Propagation Algorithm

This appendix aims at giving a brief review of the summary-propagation algorithm on a generic level (also here we will closely follow (Loeliger, 2004; Loeliger *et al.*, 2007)). One of the most important operations that can be performed on factor graphs is marginalization, i.e., the computation of marginals of probability functions. Marginalization lies at the heart of many algorithms in signal processing, coding and machine learning. As we will show, computing marginals amounts to passing messages ("summaries") along the edges in the factor graph of the system at hand. We will now describe this generic message-passing algorithm, called the sum(mary)-product algorithm (SPA).

B.1 Summary Propagation on Factor Trees

Example B.1 (Marginalization of a factored function)
Let us consider again the global function $f(x_1, x_2, x_3, x_4, x_5, x_6)$ of Example A.3. Suppose we are interested in the marginal function

$$f(x_5) \triangleq \sum_{x_1,x_2,x_3,x_4,x_6} f(x_1, x_2, x_3, x_4, x_5, x_6). \tag{B.1}$$

With the factorization (A.1), we have:

$$\begin{aligned} f(x_5) &= \sum_{x_1,x_2,x_3,x_4,x_6} f_A(x_1, x_2) \cdot f_B(x_3, x_4) \cdot f_C(x_2, x_4, x_5) \cdot f_D(x_5, x_6) \\ &= \underbrace{\sum_{x_2,x_4} f_C(x_2, x_4, x_5) \underbrace{\Big(\sum_{x_1} f_A(x_1, x_2) \Big)}_{\mu_{f_A \to x_2}(x_2)} \cdot \underbrace{\Big(\sum_{x_3} f_B(x_3, x_4) \Big)}_{\mu_{f_B \to x_4}(x_4)}}_{\mu_{f_C \to x_5}(x_5)} \\ &\qquad \cdot \underbrace{\Big(\sum_{x_6} f_D(x_5, x_6) \Big)}_{\mu_{f_D \to x_5}(x_5)} . \end{aligned} \tag{B.2}$$

□

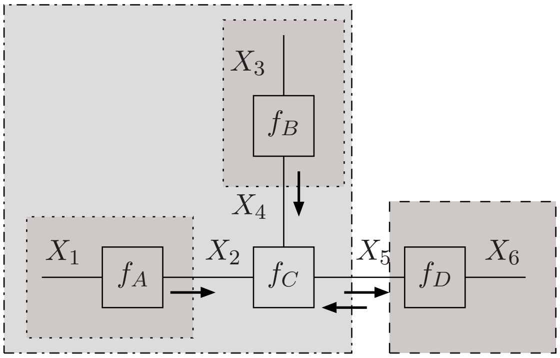


Fig. B.1. Summary-propagation for computing $f(x_5)$. The arrows correspond to intermediate results, referred to as "messages" (see (B.2)).

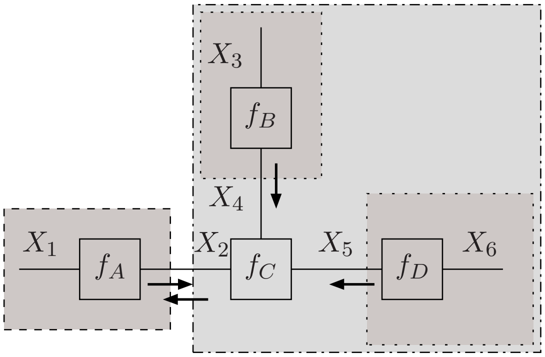


Fig. B.2. Summary-propagation for computing $f(x_2)$. Note that we have already computed the messages $\mu_{f_A \to x_2}(x_2)$, $\mu_{f_B \to x_4}(x_4)$, and $\mu_{f_D \to x_5}(x_5)$ for computing $f(x_5)$ (see Fig. B.1); they can be "re-used" for computing $f(x_2)$.

The idea behind (B.2) is to "push" the summations as much right as possible. For example, when summing w.r.t. X_6, we can push the summation sign to the right side of every factor except $f_D(x_5, x_6)$, since this factor depends on X_6. As a result, instead of carrying out a high-dimensional sum, it suffices to carry out simpler ones (one- and two-dimensional in our example). The intermediate terms $\mu_{f_j \to x_i}(x_i)$ are functions of X_i. The domain of such a functions is the alphabet of X_i. Their meaning becomes obvious when looking at Fig. B.1.

The intermediate results can be interpreted as "messages" flowing along the edges of the graph. For example, the message $\mu_{f_A \to x_2}(x_2)$, which is the sum $\sum_{x_1} f_A(x_1, x_2)$, can be interpreted as a message leaving node f_A along edge X_2. If both $\mu_{f_A \to x_2}(x_2)$ and $\mu_{f_B \to x_4}(x_4)$ are available, the message $\mu_{f_C \to x_5}(x_5)$ can be computed as the output message of node f_C towards edge X_5. The final result of (B.2) is

$$f(x_5) = \mu_{f_C \to x_5}(x_5) \cdot \mu_{f_D \to x_5}(x_5). \tag{B.3}$$

It is the product of the two messages along the same edge.

Each message can be regarded as a "summary" of what lies "behind" it, as illustrated by the boxes in Fig. B.1. Computing a message means "closing" a

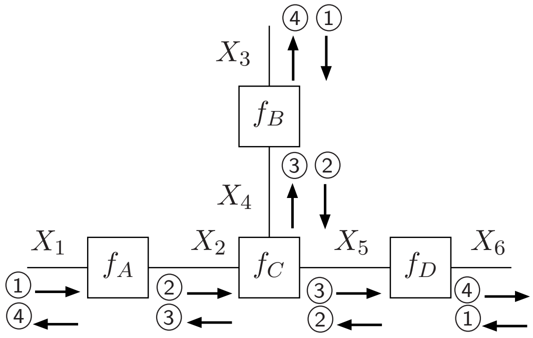


Fig. B.3. The SPA computes two messages along each edge. Those messages are required for calculating the marginal functions $f(x_1)$, $f(x_2)$, $f(x_3)$, $f(x_4)$, $f(x_5)$ and $f(x_6)$. The circled numbers indicate the order of the message computations.

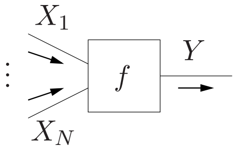


Fig. B.4. Message along a generic edge.

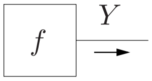


Fig. B.5. Message out of a leaf node.

part of the graph ("box"). The details inside such a box are "summed out", only a summary is propagated (hence the name summary-propagation). In the first step, the dark shaded areas in Fig. B.1 are summarized (resulting in $\mu_{f_A \to x_2}(x_2)$ and $\mu_{f_D \to x_5}(x_5)$). Afterwards, the lighter shaded box is closed (amounting to $\mu_{f_C \to x_2}(x_2)$), until we arrive at (B.3).

Half-edges (such as X_1) do not carry a message towards the connected node; alternatively, the edge may be thought of as carrying a message representing a neutral factor 1. With this in mind, we notice that every message (i.e., every intermediate result) of (B.2) is computed in the same way. Consider the generic node depicted in Fig. B.4 with messages arriving along its edges $X_1, \ldots, X_N$. The message towards edge e is computed by the following rule.

Sum-product rule:

$$\mu_{f \to y}(y) \triangleq \sum_{x_1,\ldots,x_N} f(y, x_1, \ldots, x_N)\mu_{x_1 \to f}(x_1) \cdots \mu_{x_N \to f}(x_N). \tag{B.4}$$

In words: The message out of a node f along the edge Y is the product of the function f and all messages towards f along all other edges, summarized over all variables except Y. This is the sum-product rule. In general, messages are computed out of any edge, there is no preferential direction. The message out of a leaf node f along edge Y is the function f itself, as illustrated in Fig. B.5.

Example B.2 (Maximization of a factored function)
Let us consider again the global function $f(x_1, x_2, x_3, x_4, x_5, x_6)$ of Example B.1. Assume we are now interested in the function ("max-marginal")

$$f(x_5) \triangleq \max_{x_1,x_2,x_3,x_4,x_6} f(x_1, x_2, x_3, x_4, x_5, x_6). \tag{B.5}$$

With the factorization (A.1), we have:

$$\begin{aligned} f(x_5) &= \max_{x_1,x_2,x_3,x_4,x_6} f_A(x_1, x_2) \cdot f_B(x_3, x_4) \cdot f_C(x_2, x_4, x_5) \cdot f_D(x_5, x_6) \\ &= \underbrace{\max_{x_2,x_4} f_C(x_2, x_4, x_5) \underbrace{\Big(\max_{x_1} f_A(x_1, x_2)\Big)}_{\mu_{f_A \to x_2}(x_2)} \cdot \underbrace{\Big(\max_{x_3} f_B(x_3, x_4)\Big)}_{\mu_{f_B \to x_4}(x_4)}}_{\mu_{f_C \to x_5}(x_5)} \\ &\qquad \cdot \underbrace{\Big(\max_{x_6} f_D(x_5, x_6)\Big)}_{\mu_{f_D \to x_5}(x_5)}. \end{aligned} \tag{B.6}$$

□

It is noteworthy that every message of (B.6) is computed according to the same rule.

Max-product rule:

$$\mu_{f \to y}(y) \triangleq \max_{x_1,\ldots,x_N} f(y, x_1, \ldots, x_N)\mu_{x_1 \to f}(x_1) \cdots \mu_{x_N \to f}(x_N) \tag{B.7}$$

The sum-product and max-product rules can be considered as instances of the following single rule.

> **Summary-product rule:** The message $\mu_{f \to y}(y)$ out of a factor node $f(y, \ldots)$ along the edge Y is the product of $f(y, \ldots)$ and all messages towards f along all edges except Y, summarized over all variables except Y.

The following example shows how several marginals can be obtained simultaneously in an efficient manner.

Example B.3 (Recycling messages)
Suppose we are also interested in the max-marginal function $f(x_2)$ of the global function $f(x_1, x_2, x_3, x_4, x_5, x_6)$ of Example B.1:

$$f(x_2) \triangleq \max_{x_1,x_3,x_4,x_5,x_6} f(x_1, x_3, x_4, x_5, x_6).$$

This max-marginal can be computed by the max-product algorithm depicted in Fig. B.2. Note that we have already computed the messages $\mu_{f_A \to x_2}(x_2)$, $\mu_{f_B \to x_4}(x_4)$, and $\mu_{f_D \to x_5}(x_5)$ in (B.6)*; they can be "re-used" for computing $f(x_2)$. Eventually, $f(x_2)$ is obtained as*

$$f(x_2) = \mu_{f_A \to x_2}(x_2)\mu_{f_C \to x_2}(x_2). \tag{B.8}$$

□

From this last example, we learn that the two messages associated to an edge are for the computation of each (max-)marginal the same. It is therefore sufficient to compute each message once. The (max-)marginal $f(y)$ of a certain variable Y is the product of the two messages on the corresponding edge, such as (B.3) and (B.8). In general, it is

$$f(y) = \mu_{f_A \to y}(y) \cdot \mu_{f_B \to y}(y) \tag{B.9}$$

where f_A and f_B are the two nodes attached to edge Y. For half edges, the message coming from the open end carries a neutral factor "1". Therefore, the message from the node towards the edge is already the marginal of the corresponding variable.

In its general form, the summary-propagation algorithm (SPA) computes two messages on every edge. For factor graphs without loops (factor trees), the marginals can obtained in an optimal number of computations as follows.[2] One starts the message computation from the leaves and proceeds with nodes whose input messages become available. In this way, each message is computed exactly once, as illustrated in Fig. B.3. When the algorithm stops, exact marginals, such as (B.9), are available for all variables *simultaneously*.

In summary:

- Marginals such as (B.5) can be computed as the product of two messages as in (B.9).
- Such messages are summaries of the subgraph behind them.
- All messages (except those out of terminal nodes) are computed from other messages according to the summary-product rule.

If the summaries are computed by the sum-product rule, the above algorithm is referred to as "sum-product algorithm" or "belief propagation". On the other hand, if the summaries are computed according to the max-product rule, it is known as the "max-product algorithm".

[2] The number of computations may be reduced by additional information about the structure of the local node functions. This is the case when the factor nodes themselves may be expressed by (non-trivial) factor trees.

If one applies the rules (B.4) or (B.7), the values of the messages often quickly tend to zero and the algorithm becomes instable. Therefore, it is advisable to scale the message: instead of the message $\mu(.)$, a modified message $\tilde{\mu}(.) \triangleq \gamma\mu(.)$ is computed, where the scale factor γ may be chosen as one wishes. The final result (B.9) will then be known up to a scaling factor, which is often not a problem.

A message update schedule says when one has to calculate what message. For factor trees, there is an optimal message update schedule, as we explained previously; for cyclic factor graphs, this is not the case.

B.2 Summary Propagation on Cyclic Factor Graphs

The situation becomes quite different when the graph has cycles. In this case, the summary-propagation algorithm becomes iterative: a new output message at some node can influence the inputs of the same node through another path in the graph. The algorithm does not amount to the exact marginal functions. In fact, there is even no guarantee that the algorithm converges! Astonishingly, applying the summary-product algorithm on cyclic graphs works excellently in the context of coding and signal processing, and machine learning. In many practical cases, the algorithm reaches a stable point and the obtained marginal functions are satisfactory: decisions based on those marginals are often close enough to the "optimal" decisions.

Summary-propagation on cyclic-graphs consists of the following steps

(1) First, all edges are initialized with a neutral message, i.e., a factor $\mu(.) = 1$.
(2) All messages are then recursively updated according to some schedule. This schedule may vary from step to step.
(3) After each step, the marginal functions are computed according to (B.9).
(4) One takes decisions based on the current marginal functions.
(5) The algorithm is halted when the available time is over or when some stopping criterion is satisfied (e.g., when all messages varied less than some small ε over the last iterations).

C Appendix: Derivation of the SES Inference Algorithm

In this appendix, we derive the inference algorithm for multi-variate SES, summarized in Table 2.

The estimate $\hat{\theta}^{(i+1)}$ (25) is available in closed-form; indeed, it is easily verified that the point estimates $\hat{\delta}_t^{(i+1)}$ and $\hat{\delta}_f^{(i+1)}$ are the (sample) mean of the timing and frequency offset respectively, computed over all pairs of coincident events:

$$\hat{\delta}_t^{(i+1)} \triangleq \frac{1}{n^{(i+1)}} \sum_{k=1}^{n^{(i+1)}} \frac{\hat{t}_k'^{(i+1)} - \hat{t}_k^{(i+1)}}{(\Delta\hat{t}_k^{(i+1)} + \Delta\hat{t}_k'^{(i+1)})^2} \tag{C.1}$$

$$\hat{\delta}_f^{(i+1)} \triangleq \frac{1}{n^{(i+1)}} \sum_{k=1}^{n^{(i+1)}} \frac{\hat{f}_k'^{(i+1)} - \hat{f}_k^{(i+1)}}{(\Delta\hat{f}_k^{(i+1)} + \Delta\hat{f}_k'^{(i+1)})^2}, \tag{C.2}$$

where $n^{(i+1)}$ is the number of coincident bump pairs in alignment $\hat{c}^{(i+1)}$, and where we used the shorthand notation $\hat{t}_k^{(i+1)} = t_{\hat{j}_k^{(i+1)}}$, $\hat{f}_k^{(i+1)} = f_{\hat{j}_k^{(i+1)}}$, $\Delta\hat{t}_k^{(i+1)} = \Delta t_{\hat{j}_k^{(i+1)}}$, $\Delta\hat{f}_k^{(i+1)} = \Delta f_{\hat{j}_k^{(i+1)}}$, and likewise $\hat{t'}_k^{(i+1)}$, $\hat{f'}_k^{(i+1)}$, $\Delta\hat{t'}_k^{(i+1)}$, $\Delta\hat{f'}_k^{(i+1)}$.

The estimates $\hat{s}_t^{(i+1)}$ and $\hat{s}_f^{(i+1)}$ are obtained as:

$$\hat{s}_t^{(i+1)} = \frac{\nu_t s_{0,t} + n^{(i+1)} \hat{s}_{t,\text{sample}}^{(i+1)}}{\nu_t + n^{(i+1)} + 2} \tag{C.3}$$

$$\hat{s}_f^{(i+1)} = \frac{\nu_f s_{0,f} + n^{(i+1)} \hat{s}_{f,\text{sample}}^{(i+1)}}{\nu_f + n^{(i+1)} + 2}, \tag{C.4}$$

where ν_t, ν_f, $s_{0,t}$ and $s_{0,f}$ are the parameters of the conjugate priors (17) and (18), and $s_{t,\text{sample}}$ and $s_{f,\text{sample}}$ are the (sample) variance of the timing and frequency offset respectively, computed over all pairs of coincident events:

$$\hat{s}_{t,\text{sample}}^{(i+1)} \triangleq \frac{1}{n^{(i+1)}} \sum_{k=1}^{n^{(i+1)}} \frac{(\hat{t}_k'^{(i+1)} - \hat{t}_k^{(i+1)} - \hat{\delta}_t^{(i+1)})^2}{(\Delta\hat{t}_k^{(i+1)} + \Delta\hat{t}_k'^{(i+1)})^2} \tag{C.5}$$

$$\hat{s}_{f,\text{sample}}^{(i+1)} \triangleq \frac{1}{n^{(i+1)}} \sum_{k=1}^{n^{(i+1)}} \frac{(\hat{f}_k'^{(i+1)} - \hat{f}_k^{(i+1)} - \hat{\delta}_f^{(i+1)})^2}{(\Delta\hat{f}_k^{(i+1)} + \Delta\hat{f}_k'^{(i+1)})^2}. \tag{C.6}$$

Now we address the update (24), i.e., finding the optimal pairwise alignment c for *given* values $\hat{\theta}^{(i)}$ of the parameters θ. In the following, we will show that it is equivalent to a standard problem in combinatorial optimization, i.e., max-weight bipartite matching (see, e.g., (Gerards, 1995; Pulleyblank, 1995; Bayati *et al.*, 2005, 2007; Huang *et al.*, 2007; Sanghavi, 2007a,b)). First, let us point out that in (22), there is a factor β for every non-coincident bump; the total number of factors β is hence $n_{\text{non-co}} = n + n' - 2n_{\text{co}}$, where n_{co} is the number of coincident bump pairs. On the other hand, for each pair of coincident bumps, there is a factor $\mathcal{N}(\cdot; \delta_t, s_t)\,\mathcal{N}(\cdot; \delta_f, s_f)$; in total there are n_{co} such factors. Therefore, we can rewrite (22) as:

$$p(e, e', c, \theta) \propto \prod_{k=1}^{n} \prod_{k'=1}^{n'} \Big(\mathcal{N}\big(t'_{k'} - t_k; \bar{\delta}_t, \bar{s}_t\big)\, \mathcal{N}\big(f'_{k'} - f_k; \bar{\delta}_f, \bar{s}_f\big) \beta^{-2} \Big)^{c_{kk'}} \cdot I(c) p(\delta_t) p(s_t) p(\delta_f) p(s_f), \tag{C.7}$$

where we omitted the factor $\beta^{n+n'}$ since it is an irrelevant constant, and

$$I(c) = \prod_{k=1}^{n} \Big(\delta\Big[\sum_{k'=1}^{n'} c_{kk'} \Big] + \delta\Big[\sum_{k'=1}^{n'} c_{kk'} - 1 \Big] \Big) \cdot \prod_{k'=1}^{n'} \Big(\delta\Big[\sum_{k=1}^{n} c_{kk'} \Big] + \delta\Big[\sum_{k=1}^{n} c_{kk'} - 1 \Big] \Big). \tag{C.8}$$

The factor $I(c)$ encodes the constraints (14). The maximization (24) is equivalent to:

$$\hat{c}^{(i+1)} = \underset{c}{\operatorname{argmax}} \log p(e, e', c, \hat{\theta}^{(i)}). \tag{C.9}$$

Using (C.7), we can rewrite (C.9) as:

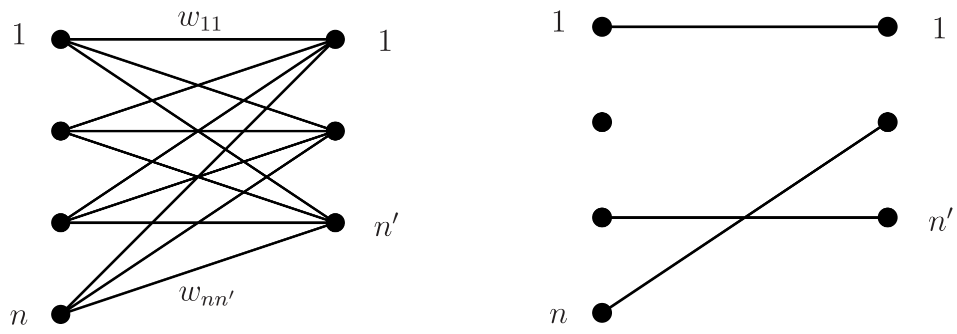
$$\hat{c}^{(i+1)} = \underset{c}{\operatorname{argmax}} \sum_{kk'} w_{kk'}\, c_{kk'} + \log I(c) + \zeta, \tag{C.10}$$

where ζ is an irrelevant constant and

$$w_{kk'} = -\frac{\Big(t'_{k'} - t_k - \hat{\delta}_t^{(i)}\Big)^2}{2s_t(\Delta t_k + \Delta t'_{k'})^2} - \frac{\Big(f'_{k'} - f_k - \hat{\delta}_f^{(i)}\Big)^2}{2s_f(\Delta f_k + \Delta f'_{k'})^2} - 2\log\beta - 1/2 \log 2\pi s_t(\Delta t_k + \Delta t'_{k'})^2 - 1/2 \log 2\pi s_f(\Delta f_k + \Delta f'_{k'})^2, \tag{C.11}$$

where the weights $w_{kk'}$ can be positive or negative. If weight $w_{kk'}$ is negative, then $c_{kk'} = 0$. Indeed, setting $c_{kk'}$ equal to one would decrease $\log p(e, e', c, \hat{\theta}^{(i)})$. Bump pairs $(e_k, e'_{k'})$ with large weights $w_{kk'}$ are likely to be coincident. The closer the bumps $(e_k, e'_{k'})$ on the time-frequency plane, the larger their weight $w_{kk'}$. From the definition of β (9), we can also see that the weights increase as the prior for a deletion p_d decreases. Indeed, the fewer deletions, the more likely that a bump e_k is coincident with a bump e'_k. In addition, the weights $w_{kk'}$ grow as the concentration λ of bumps on the time-frequency plane decreases. Indeed, if there are few bumps in each model (per square unit) and a bump e_k of e happens to be close to a bump $e'_{k'}$ of e', they are most probably a coincident bump pair, since most likely, there are only few other bumps in e' that are close to e_k.

One can naturally associate a bipartite graph with the optimization problem (C.10). The latter is a graph whose nodes can be divided into two disjoint sets $\mathcal{V}_1$ and $\mathcal{V}_2$ such that every edge connects a node in $\mathcal{V}_1$ and one in $\mathcal{V}_2$, i.e., there is no edge between two vertices in the same set. As a first step, one associates a node to each bump in e, resulting in the set of nodes $\mathcal{V}_1$, and likewise, one associates a node to each bump in e', resulting in the set of nodes $\mathcal{V}_2$. Next one draws edges between each node of $\mathcal{V}_1$ and $\mathcal{V}_2$, resulting in the bipartite graph depicted in Fig. 1(a). At last, one assigns a weight to every edge, more precisely, the edge between node k of $\mathcal{V}_1$ and node k' of $\mathcal{V}_2$ has weight $w_{kk'}$.



(a) Weighted bipartite graph.

(b) Non-perfect matching.

Fig. C.1. Bipartite max-weight matching. The weighted bipartite graph (left) is obtained as follows: first one associates a node to each bump in e, resulting in the set of nodes $\mathcal{V}_1$ (nodes at the left), and likewise, one associates a node to each bump in e', resulting in the set of nodes $\mathcal{V}_2$ (nodes at the right). Next one draws edges between each node of $\mathcal{V}_1$ and $\mathcal{V}_2$, and associates a weight $w_{kk'}$ to each edge. Problem (C.10) is equivalent to finding the heaviest disjoint set of edges in that weighted bipartite graph. Note that some nodes may not be connected to edges of that subset, i.e., the matching may be non-perfect (right).

Let us now look back at problem (C.10): one maximizes a sum of weights $w_{kk'}$ subject to the constraints (14). This problem is equivalent to finding the heaviest disjoint set of edges in the bipartite graph of Fig. 1(a). This set of edges does not need to be connected to every node, some nodes may not be matched. For example, in Fig. 1(b) the second node of $\mathcal{V}_1$ is not matched. The latter problem is known as imperfect max-weight bipartite matching, and can be solved in at least three different ways:

- by the Edmond-Karp (Edmonds *et al.*, 1972) or auction algorithm (Bertsekas *et al.*, 1989),
- by using the tight LP relaxation to the integer programming formulation of bipartite max-weight matching (Gerards, 1995; Pulleyblank, 1995),
- by applying the max-product algorithm (Bayati *et al.*, 2005, 2007; Huang *et al.*, 2007; Sanghavi, 2007a,b).

The Edmond-Karp (Edmonds *et al.*, 1972) and auction algorithm (Bertsekas *et al.*, 1989) both result in the optimum solution of (C.10). The same holds for the linear programming relaxation approach and the max-product algorithm as long as the optimum solution is unique. If the latter is not unique, the linear programming relaxation method may result in non-integer solutions and the max-product algorithm will not converge, as shown in (Sanghavi, 2007a,b). Note that in many practical problems, the optimum matching (C.10) is unique with probability one. This is in particular the case for the bump models described in the above. Since the max-product algorithm is arguably the simplest algorithm in the above list, we will in the following only describe that algorithm.

Before we can apply the max-product algorithm on the graph Fig. 7 in order to find the optimal alignment (24), we first need to slightly modify that graph. Indeed, the alignment c is computed for given $\theta = \hat{\theta}^{(i)}$, i.e., one computes c conditioned on $\theta = \hat{\theta}^{(i)}$. Generally, if one performs inference conditioned on a variable X, the edge(s) X need to be removed from the factor graph of the statistical model at hand. Therefore, for the purpose of computing (24), one needs to remove the θ edges (and the two bottom nodes in Fig. 7), resulting in the factor graph depicted in Fig. 8. It is noteworthy that the $\mathcal{N}$-nodes have become leaf nodes, and that θ in $g_{\mathcal{N}}$ (19) is replaced by the estimate $\hat{\theta}^{(i)}$.

Before applying the max-product algorithm to Fig. 8, we briefly describe it in general terms. The max-product algorithm is an optimization procedure that operates on a factor graph (or any other kind of graphical model) (Jordan, 1999; Loeliger, 2004; Loeliger *et al.*, 2007); local information (referred to as "messages") propagates along the edges in the graph, and is computed at each node according to the generic max-product computation rule. After convergence or after a fixed number of iterations, one combines the messages in order to obtain decisions (Loeliger, 2004; Loeliger *et al.*, 2007). If the graph is cycle-free, one obtains optimal solution of the optimization problem, on the other hand, if the graph is cyclic, the max-product algorithm may not converge, and if it converges, the resulting decisions are not necessarily optimal (Loeliger, 2004; Loeliger *et al.*, 2007). However, for certain problems that involve cyclic graphs, it has been shown that the max-product algorithm is guaranteed to find the optimum solution. As we pointed out earlier, this is in particular the case for the max weight matching problem (Bayati *et al.*, 2005, 2007; Huang *et al.*, 2007; Sanghavi, 2007a,b). We refer to the Appendix B for more information on the max-product algorithm.

As mentioned in the above, the messages in the graph of Fig. 8 are iteratively updated according to the max-product update rule, which is stated in row 1 of Table C.1 for a generic node g. We now apply that generic rule to the nodes in Fig. 8. Let us first consider the β- and $\mathcal{N}$-nodes, which are leaf nodes. The max-product message leaving a leaf node is nothing but the node function itself (see row 2 of Table C.1). Therefore, the messages $\mu{\downarrow}(b_k)$ and $\mu{\downarrow}(b'_k)$, propagating downward along the edges B_k and B'_k respectively, are given by:

$$\mu{\downarrow}(b_k) = g_\beta(b_k) = \beta\delta[b_k - 1] + \delta[b_k] \tag{C.12}$$

$$\mu{\downarrow}(b'_k) = g_\beta(b'_k) = \beta\delta[b'_k - 1] + \delta[b'_k], \tag{C.13}$$

and similarly, the messages $\mu{\uparrow}(c_{kk'})$, propagating upward along the edges $C_{kk'}$:

$$\mu{\uparrow}(c_{kk'}) = g_{\mathcal{N}}(c_{kk'}; \hat{\theta}^{(i)}) \tag{C.14}$$

$$= \Big(\mathcal{N}\Big(t'_{k'} - t_k; \bar{\delta}_t^{(i)}, \bar{s}_t^{(i)}\Big)\,\mathcal{N}\Big(f'_{k'} - f_k; \bar{\delta}_f^{(i)}, \bar{s}_f^{(i)}\Big)\Big)^{c_{kk'}}, \tag{C.15}$$

where $\bar{\delta}_t^{(i)} = \hat{\delta}_t^{(i)}(\Delta t_k + \Delta t'_{k'})$, $\bar{\delta}_f^{(i)} = \hat{\delta}_f^{(i)}(\Delta f_k + \Delta f'_{k'})$, $\bar{s}_t^{(i)} = \hat{s}_t^{(i)}(\Delta t_k + \Delta t'_{k'})^2$,

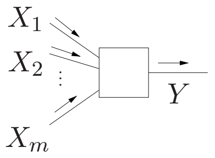
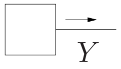
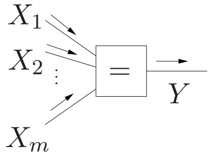
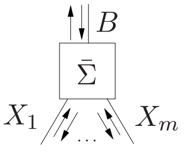
1	 $g(x_1, x_2, \ldots x_m, y)$	$\mu(y) \propto \max_{x_1,x_2,\ldots,x_m} g(x_1, \ldots, x_m, y)\mu(x_1) \ldots \mu(x_m)$
2	 $g(y)$	$\mu(y) \propto g(y)$
3	 $\delta[x_1 - x_2] \ldots \delta[x_m - y]$ $X_1, \ldots, X_m, Y \in \{0, 1\}$	$\begin{pmatrix} \mu(y=0) \\ \mu(y=1) \end{pmatrix} \propto \begin{pmatrix} \mu(x_1=0)\mu(x_2=0) \ldots \mu(x_m=0) \\ \mu(x_1=1)\mu(x_2=1) \ldots \mu(x_m=1) \end{pmatrix}$
4	 $\delta\big[b + \sum_{k=1}^{m} x_k - 1\big]$ $B, X_1, \ldots, X_m \in \{0, 1\}$	$\begin{pmatrix} \mu{\downarrow}(x_k=0) \\ \mu{\downarrow}(x_k=1) \end{pmatrix} \propto \begin{pmatrix} \max\left(\frac{\mu\downarrow(b=1)}{\mu\downarrow(b=0)}, \max_{\ell \neq k} \frac{\mu\uparrow(x_\ell=1)}{\mu\uparrow(x_\ell=0)}\right) \\ 1 \end{pmatrix}$ $\begin{pmatrix} \mu{\uparrow}(b=0) \\ \mu{\uparrow}(b=1) \end{pmatrix} \propto \begin{pmatrix} \max_k \frac{\mu\uparrow(x_k=1)}{\mu\uparrow(x_k=0)} \\ 1 \end{pmatrix}$

Table C.1
Max-product computation rules for the nodes in the factor graph of Fig. 8.

and $\bar{s}_f^{(i)} = \hat{s}_f^{(i)}(\Delta f_k + \Delta f'_{k'})^2$. Note that the messages $\mu\downarrow(b_k)$ and $\mu\downarrow(b'_k)$ never change, i.e., they do not need to be recomputed in the course of the max-product algorithm.

We now turn to the messages at the $\bar{\Sigma}$-nodes; row 4 of Table C.1 considers a generic $\bar{\Sigma}$-node. Its (generic) incident edges $X_1, \ldots, X_m$ are replaced by $C_{1k'}, \ldots, C_{nk'}$ or $C_{k1}, \ldots, C_{kn'}$ in Fig. 8; for convenience, we now compute the messages in terms of $X_1, \ldots, X_m$, later we will formulate them in terms of

$C_{kk'}$ (more specifically, in (C.22) (C.23) (C.30) (C.32)). The message $\mu\!\uparrow\!(b)$, propagating upward along the edge B, is computed as:

$$\begin{aligned}
\mu\!\uparrow\!(b) &= \max_{x_1,\ldots,x_m} \delta\Big[b + \sum_{k=1}^{m} x_k - 1\Big]\, \mu\!\uparrow\!(x_1)\ldots\mu\!\uparrow\!(x_m) && \text{(C.16)}\\
&= \mu\!\uparrow\!(x_1=0)\ldots\mu\!\uparrow\!(x_m=0)\\
&\quad\cdot \Big(\delta[b-1] + \delta[b]\max_k \mu\!\uparrow\!(x_k=1)/\mu\!\uparrow\!(x_k=0)\Big), && \text{(C.17)}
\end{aligned}$$

where $X_1,\ldots,X_m,B \in \{0,1\}$ and $\mu\!\uparrow\!(x_k)$ are the messages propagating upward along the edges X_k. In row 4 of Table C.1, we have written the message $\mu\!\uparrow\!(b)$ componentwise. The messages $\mu\!\downarrow\!(x_k)$, propagating downward along the edges X_k, are computed similarly:

$$\begin{aligned}
\mu\!\downarrow\!(x_k) &= \max_{b,x_1,\ldots,x_{k-1},x_{k+1}\cdots,x_m} \delta\Big[b + \sum_{k=1}^{m} x_k - 1\Big]\mu\!\downarrow\!(b)\mu\!\uparrow\!(x_1)\ldots\mu\!\uparrow\!(x_{k-1})\\
&\qquad\cdot\mu\!\uparrow\!(x_{k+1})\ldots\mu\!\uparrow\!(x_m) && \text{(C.18)}\\
&= \mu\!\uparrow\!(b=0)\mu\!\uparrow\!(x_1=0)\ldots\mu\!\uparrow\!(x_{k-1}=0)\mu\!\uparrow\!(x_{k+1}=0)\ldots\mu\!\uparrow\!(x_m=0)\\
&\quad\cdot\Big[\delta[x_k-1] + \delta[x_k]\max\Big(\mu\!\downarrow\!(b=1)/\mu\!\downarrow\!(b=0),\\
&\qquad\max_{\ell\neq k}\mu\!\uparrow\!(x_\ell=1)/\mu\!\uparrow\!(x_\ell=0)\Big)\Big], && \text{(C.19)}
\end{aligned}$$

where $\mu\!\uparrow\!(b)$ is the message propagating upward along the edge B. The componentwise formulation of $\mu\!\downarrow\!(x_k)$ is also listed in row 4 of Table C.1.

At last, we turn to the messages computed at the equality constraint nodes in Fig. 8. The (generic) equality constraint node is considered in row 3 of Table C.1. The message $\mu(y)$, leaving this node along the edge Y, is computed as follows:

$$\begin{aligned}
\mu(y) &= \max_{x_1,\ldots,x_m} \delta[x_1-x_2]\ldots\delta[x_{m-1}-x_m]\delta[x_m-y]\,\mu(x_1)\ldots\mu(x_m) && \text{(C.20)}\\
&= \mu(x_1=y)\ldots\mu(x_m=y), && \text{(C.21)}
\end{aligned}$$

where $X_1,\ldots,X_m,Y \in \{0,1\}$. Since the equality constraint node is symmetric, the other messages leaving the equality constraint node (along the edges X_1, $\ldots$, X_m) are computed analogously.

We now use (C.12)–(C.21) to derive the update rules for the messages $\mu\!\uparrow\!(b_k)$, $\mu\!\uparrow\!(b'_k)$, $\mu\!\downarrow\!(c_{kk'})$, $\mu\!\uparrow'\!(c_{kk'})$, $\mu\!\downarrow'\!(c_{kk'})$, $\mu\!\uparrow''\!(c_{kk'})$, and $\mu\!\downarrow''\!(c_{kk'})$ in Fig. 8:

- the messages $\mu\!\uparrow\!(b_k)$ and $\mu\!\uparrow\!(b'_k)$ propagate *upward* along the edges b_k and b'_k respectively, towards the β-nodes,
- the messages $\mu\!\downarrow\!(c_{kk'})$ propagate *downward* along the edges $C_{kk'}$, leaving the equality constraint nodes,

- the messages $\mu\!\uparrow'(c_{kk'})$ and $\mu\!\uparrow''(c_{kk'})$ propagate *upward* along the edges $C_{kk'}$, towards the $\bar{\Sigma}$-nodes connected to the edges B_k and $B'_{k'}$ respectively (see Fig. 8, left hand side),
- the messages $\mu\!\downarrow'(c_{kk'})$ and $\mu\!\downarrow''(c_{kk'})$ propagate *downward* along the edges $C_{kk'}$, leaving the $\bar{\Sigma}$-nodes connected to the edges B_k and $B'_{k'}$ respectively.

We start with the messages $\mu\!\uparrow(b_k)$:

$$\begin{aligned}\mu\!\uparrow(b_k) &= \mu\!\uparrow'(c_{k1}=0)\ldots\mu\!\uparrow'(c_{kn'}=0)\\ &\quad\cdot\Big(\delta[b_k-1]+\delta[b_k]\max_{k'}\mu\!\uparrow'(c_{kk'}=1)/\mu\!\uparrow'(c_{kk'}=0)\Big),\end{aligned}\tag{C.22}$$

where we used (C.17), and likewise:

$$\begin{aligned}\mu\!\uparrow(b'_{k'}) &= \mu\!\uparrow''(c_{1k'}=0)\ldots\mu\!\uparrow''(c_{nk'}=0)\\ &\quad\cdot\Big(\delta[b'_{k'}-1]+\delta[b'_{k'}]\max_{k}\mu\!\uparrow''(c_{kk'}=1)/\mu\!\uparrow''(c_{kk'}=0)\Big).\end{aligned}\tag{C.23}$$

The messages $\mu\!\downarrow(c_{kk'})$ are derived as follows:

$$\mu\!\downarrow(c_{kk'}) = \mu\!\downarrow'(c_{kk'})\mu\!\downarrow''(c_{kk'}),\tag{C.24}$$

where we used (C.21).

The messages $\mu\!\uparrow'(c_{kk'})$ are derived as follows:

$$\begin{aligned}\mu\!\uparrow'(c_{kk'}) &\propto \mu\!\downarrow''(c_{kk'})\mu\!\uparrow(c_{kk'}) = \mu\!\downarrow''(c_{kk'})g_{\mathcal{N}}(c_{kk'};\hat{\theta}^{(i)}) &(C.25)\\ &= \mu\!\downarrow''(c_{kk'})\Big(\mathcal{N}\big(t'_{k'}-t_k;\bar{\delta}_t^{(i)},\bar{s}_t^{(i)}\big)\,\mathcal{N}\big(f'_{k'}-f_k;\bar{\delta}_f^{(i)},\bar{s}_f^{(i)}\big)\Big)^{c_{kk'}}, &(C.26)\end{aligned}$$

where we used (C.15) and (C.21), and where $\bar{\delta}_t^{(i)} = \hat{\delta}_t^{(i)}(\Delta t_k+\Delta t'_{k'})$, $\bar{\delta}_f^{(i)} = \hat{\delta}_f^{(i)}(\Delta f_k+\Delta f'_{k'})$, $\bar{s}_t^{(i)} = \hat{s}_t^{(i)}(\Delta t_k+\Delta t'_{k'})^2$, and $\bar{s}_f^{(i)} = \hat{s}_f^{(i)}(\Delta f_k+\Delta f'_{k'})^2$. Similarly, we have:

$$\begin{aligned}\mu\!\uparrow''(c_{kk'}) &\propto \mu\!\downarrow'(c_{kk'})\mu\!\uparrow(c_{kk'}) = \mu\!\downarrow'(c_{kk'})g_{\mathcal{N}}(c_{kk'};\hat{\theta}^{(i)}) &(C.27)\\ &= \mu\!\downarrow'(c_{kk'})\Big(\mathcal{N}\big(t'_{k'}-t_k;\bar{\delta}_t^{(i)},\bar{s}_t^{(i)}\big)\,\mathcal{N}\big(f'_{k'}-f_k;\bar{\delta}_f^{(i)},\bar{s}_f^{(i)}\big)\Big)^{c_{kk'}}. &(C.28)\end{aligned}$$

The messages $\mu\!\uparrow'(c_{kk'})$ and $\mu\!\uparrow''(c_{kk'})$ depend on the messages $\mu\!\downarrow''(c_{kk'})$ and $\mu\!\downarrow'(c_{kk'})$ respectively. The latter are computed as follows:

$$\begin{aligned}\mu\!\downarrow'(c_{kk'}) &\propto \Big(\delta[c_{kk'}-1]+\delta[c_{kk'}]\max\Big(\mu\!\downarrow(b_k=1)/\mu\!\downarrow(b_k=0),\\ &\qquad\qquad \max_{\ell'\neq k'}\mu\!\uparrow'(c_{k\ell'}=1)/\mu\!\uparrow'(c_{k\ell'}=0)\Big)\Big) &(C.29)\\ &= \Big(\delta[c_{kk'}-1]+\delta[c_{kk'}]\max\Big(\beta,\\ &\qquad\qquad \max_{\ell'\neq k'}\mu\!\uparrow'(c_{k\ell'}=1)/\mu\!\uparrow'(c_{k\ell'}=0)\Big)\Big), &(C.30)\end{aligned}$$

where we used (C.12) and (C.19), and likewise

$$\begin{aligned}\mu{\downarrow}''(c_{kk'}) &\propto \Big(\delta[c_{kk'}-1]+\delta[c_{kk'}]\max\Big(\mu{\downarrow}(b'_k=1)/\mu{\downarrow}(b'_k=0),\\ &\qquad\qquad \max_{\ell\neq k}\mu{\uparrow}''(c_{\ell k'}=1)/\mu{\uparrow}''(c_{\ell k'}=0)\Big)\Big) \quad\text{(C.31)}\\ &=\Big(\delta[c_{kk'}-1]+\delta[c_{kk'}]\max\Big(\beta,\\ &\qquad\qquad \max_{\ell\neq k}\mu{\uparrow}''(c_{\ell k'}=1)/\mu{\uparrow}''(c_{\ell k'}=0)\Big)\Big). \quad\text{(C.32)}\end{aligned}$$

The messages $\mu{\downarrow}''(c_{kk'})$ and $\mu{\downarrow}'(c_{kk'})$ depend on $\mu{\uparrow}'(c_{kk'})$ and $\mu{\uparrow}''(c_{kk'})$ and vice versa (as we pointed out earlier). Therefore, a natural way to determine all those messages is to first initialize $\mu{\downarrow}'(c_{kk'})=1=\mu{\downarrow}''(c_{kk'})$ and then to iterate the updates (C.26)–(C.30) until convergence. This can also be understood from Fig. 8: since the graph is cyclic, the max-product algorithm becomes an iterative procedure.

After convergence or after a fixed number of iterations, we compute the marginals $p(c_{kk'})$ as follows:

$$\begin{aligned}p(c_{kk'}) &\propto \mu{\downarrow}(c_{kk'})\mu{\uparrow}(c_{kk'}) \quad\text{(C.33)}\\ &= \mu{\downarrow}'(c_{kk'})\mu{\downarrow}''(c_{kk'})\\ &\quad\cdot\Big(\mathcal{N}\Big(t'_{k'}-t_k;\bar{\delta}_t^{(i)},\bar{s}_t^{(i)}\Big)\mathcal{N}\Big(f'_{k'}-f_k;\bar{\delta}_f^{(i)},\bar{s}_f^{(i)}\Big)\Big)^{c_{kk'}}, \quad\text{(C.34)}\end{aligned}$$

where we used (C.15) and (C.24). The decisions $\hat{c}_{kk'}$ are then obtained as:

$$\hat{c}_{kk'}=\operatorname*{argmax}_{c_{kk'}} p(c_{kk'}). \quad\text{(C.35)}$$

References

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

Alder B., Fernbach S., and Rotenberg M.(eds.), 1972. *Seismology: Surface Waves and Earth Oscillations,* Methods in Computational Physics 11, Academic Press, New York.

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

Bayati M., Shah D., and Sharma M., 2005. Maximum weight matching via max-product belief propagation. Proc. 2005 IEEE International Symposium on Information Theory (ISIT 2005), 1763–1767.

Bayati M., Borgs C., Chayes J., and Zecchina R., 2007. Belief-propagation for weighted b-matchings on arbitrary graphs and its relation to linear programs with integer solutions. preprint: arXiv:0709.1190v1.

Bertsekas D. and Tsitsiklis J., 1999. *Parallel and Distributed Computation:Numerical Methods,* Englewood Cliffs NJ: Prentice Hall.

Bezdek J. and Hathaway R., 2002. Some notes on alternating optimization. Proc. AFSS Int. Conference on Fuzzy Systems.

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

Buzsáki G., 2006. Rhythms of the brain. Oxford University Press.

Candès E. J. and Donoho D. L., 2002. New tight frames of curvelets and optimal representations of objects with piecewise-C2 singularities. Comm. Pure Appl. Math. 57, 219–266.

Candès E., Romberg J., and Tao T., 2006. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency estimation. IEEE Trans. Information Theory 52, no. 2, 489–509.

Chapman R., Nowlis G., McCrary J., Chapman J., Sandoval T., Guillily M., Gardner M., Reilly L., 2007. Brain event-related potentials: diagnosing early-stage Alzheimer's disease. Neurobiol. Aging 28, 194–201.

Chen Z., Ohara S., Cao J., Vialatte F., Lenz F., and Cichocki A, 2007. Statistical modeling and analysis of laser-evoked potentials of electrocorticogram recordings from awake humans, Computational Intelligence and Neuroscience, 10479.

Cichocki A., Shishkin S., Musha T., Leonowicz Z., Asada T., and Kurachi T., 2005. EEG filtering based on blind source separation (BSS) for early diagnosis of Alzheimer's disease. Clin. Neurophys 116, 729–37.

Cui J. and Wong W., 2006. The adaptive chirplet transform and visual evoked potentials. IEEE Transactions on Biomedical Engineering 53(7), 1378–1384.

Cui J., Wong W. and Mann S., 2005. Time-frequency analysis of visual evoked potentials using chirplet transform. Electronic Letters 41(4), 217–218.

Dauwels J., Vialatte F., Rutkowski T., and Cichocki A., 2007. Measuring neural synchrony by message passing, Advances in Neural Information Processing Systems 20 (NIPS 20), in press.

Dauwels J., Vialatte F., and Cichocki A., 2008. A comparative study of synchrony measures for the early diagnosis of Alzheimer's disease based on EEG. NeuroImage (submitted).

Dauwels J., Tsukada Y., Sakumura Y., Ishii S., Aoki K., Nakamura T., Matsuda M., Vialatte F., and Cichocki A., 2008. On the synchrony of morphological and molecular signaling events in cell migration. Lecture Notes on Computer Science, submitted.

Delprat N., Escudié B., Guillemain P., Kronland-Martinet R., Tchamitchian P., and Torrésani B., 1992. Asymptotic wavelet and Gabor analysis: extraction of instantaneous frequencies. IEEE Trans. Information Theory 38, 644–664.

Demanet L. and Ying L., 2007. Wave atoms and sparsity of oscillatory patterns. to appear in Appl. Comput. Harmon. Anal.

Donoho D., 2006. Compressed sensing. IEEE Trans. Information Theory 52(4), 1289–1306.

Donoho D., Tsaig I., Drori I., and Stark J-C, 2006. Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit. preprint.

Duarte M. F., Wakin M. B., and Baraniuk R. G., 2005. Fast reconstruction of piecewise smooth signals from random projections. Proc. SPARS05.

Edmonds J. and Karp R., 1972. Theoretical improvements in algorithmic efficiency for network flow problems. Journal of the ACM 19(2), 248–264.

Freeman W. T. and Weiss Y., 1999. On the fixed points of the max-product algorithm. MERL TR1999-039.

Gerards A. M. H., 1995. *Matching,* Handbooks in Operations Research and Management Science 7, Chapter 3, 135–224, North-Holland.

Gilbert A. C., Strauss M. J., Tropp J., and Vershynin R., 2006. Algorithmic linear dimension reduction in the ℓ_1 norm for sparse vectors. submitted.

Goupillaud P., Grossman A., and Morlet J., 1984. Cycle-octave and related transforms in seismic signal analysis. Geoexploration 23, 85–102.

Harris F.J., 2004. *Multirate signal processing for communication systems.* Upper Saddle River, NJ: Prentice Hall PTR.

Herrmann C. S., Grigutsch M., and Busch N. A., 2005. EEG oscillations and wavelet analysis. In: Handy, T. (ed.), Event-Related Potentials: a Methods Handbook, Cambridge, MIT Press, 229–259.

Hogan M. J., Swanwick G. R. J., Kaiser J., Rowan M., and Lawlor B., 2003. Memory-related EEG power and coherence reductions in mild Alzheimer's disease. Int. J. Psychophysiol. 49.

Huang B. and Jebara T., 2007. Loopy belief propagation for bipartite maximum weight b-matching. Proc. Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS).

Huang N. E., Shen Z., Long S. R., Wu M. C., Shih H. H., Zheng Q., Yen N.-C., Tung C. C., and Liu H. H., 1998. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 454(1971), 903–995.

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

Jordan M. I. (ed.), 1999. *Learning in Graphical Models,* MIT Press.

Kantha L. and Clayson C., 2000. *Numerical Models of Oceans and Oceanic Processes,* International Geophysics Series 66, Academic Press.

Koenig T., Lehmann D., Saito N., Kuginuki T., Kinoshita T., and Koukkou M., 2001. Decreased functional connectivity of EEG theta-frequency activity in first-episode, neuroleptic-naive patients with schizophrenia: preliminary results. Schizophrenia Res. 1–2, 55–60.

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

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

Mallat S. and Zhang Z., 1993. Matching pursuit with time-frequency dictionaries. IEEE Transactions on Signal Processing 41(12), 3397–3415.

Mallat S., 1999. *A Wavelet Tour of Signal Processing,* Academic Press.

Martin C., Gervais R., Hugues E., Messaoudi B., and Ravel N., 2004. Learning modulation of odor-induced oscillatory responses in the rat olfactory bulb: A correlate of odor recognition? The Journal of Neuroscience, 24(2):389-397.

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

Matthew B. and Cutmore T., 2002. Low-probability event-detection and separation via statistical wavelet thresholding: an application to psychophysiological denoising. Clinical Neurophysiology 113, 1403–1411.

Musha T., Asada T., Yamashita F., Kinoshita T., Chen Z., Matsuda H., Uno M., Shankle W.R., 2002. A new EEG method for estimating cortical neuronal impairment that is sensitive to early stage Alzheimers disease. Clin. Neurophysiol 113, 1052–8.

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

Nunez P. and Srinivasan R., 2006. *Electric Fields of the Brain: The Neurophysics of EEG.* Oxford University Press.

Ohara S., Crone N.E., Weiss N., and Lenz F.A., 2004. Attention to a painful cutaneous laser stimulus modulates electrocorticographic event-related desynchronization in humans Clinical Neurophysiology 115:1641-1652.

O'Neill J. C., Flandrin P., and Karl W. C., 2000. Sparse representations with chirplets via maximum likelihood estimation. submitted.

Pereda E., Quiroga R. Q., and Bhattacharya J., 2005. Nonlinear multivariate analysis of neurophsyiological signals. Progress in Neurobiology 77, 1–37.

Pulleyblank W., 1995. *Matchings and Extension,* Handbook of Combinatorics 1, Chapter 3, 179–232, North-Holland.

Quiroga R. Q., Kraskov A., Kreuz T., and Grassberger P., 2002. Performance of different synchronization measures in real data: a case study on EEG signals. Physical Review E 65.

Sanghavi S., 2007. Equivalence of LP relaxation and max-product for weighted matching in general graphs. Proc. IEEE Information Theory Workshop, Sept. 2–6, Lake Tahoe, California, USA. preprint:arXiv:0705.0760.

Sanghavi S., 2008. Linear programming analysis of loopy belief propagation for weighted matching. Advances in Neural Information Processing Systems 20 (NIPS 20), MIT Press.

Sarvotham S., Baron D., and Baraniuk R. G., 2006. Compressed sensing reconstruction via belief propagation. submitted.

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,* D. Reidel Pub. Co.

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.

Tallon-Baudry C., Bertrand O., Delpuech C., and Pernier J., 1996. Stimulus specificity of phase-locked and non-phase-locked 40Hz visual responses in human. Journal of Neuroscience 16, 4240–4249.

Thomson, D. J., 1982. Spectrum estimation and harmonic analysis. Proceedings of the IEEE 70, 1055–1096.

Tropp J. and Gilbert A. C., 2005. Signal recovery from partial information via orthogonal matching pursuit. preprint.

P. Uhlhaas and W. Singer, "Neural synchrony in brain disorders: relevance for cognitive dysfunctions and pathophysiology," *Neuron,* 52:155–168, 2006.

Varela F., Lachaux J. P., Rodriguez E., and Martinerie J., 2001. The brainweb: phase synchronization and large-scale integration. Nature Reviews Neuroscience 2(4), 229–39.

Vialatte F., Cichocki A., Dreyfus G., Musha T., Rutkowski T. M., and Gervais R., 2005. Blind source separation and sparse bump modelling of time-frequency representation of EEG signals: new tools for early detection of Alzheimer's disease. Proc. IEEE Workshop on Machine Learning for Signal Processing, 27–32.

Vialatte F., 2005. *Modélisation en bosses pour l'analyse des motifs oscillatoires reproductibles dans l'activité de populations neuronales : applications à l'apprentissage olfactif chez l'animal et à la détection précoce de la maladie d'Alzheimer,* PhD Thesis, Paris VI University, Paris.

Vialatte F., Martin C., Dubois R., Haddad J., Quenet B., Gervais R., and Dreyfus G., 2007. A machine learning approach to the analysis of time-frequency maps, and its application to neural dynamics. Neural Networks 20, 194–209.

Völkers M., Loughrey C. M., MacQuaide N., Remppis A., de George B., Koch W. J., Wegner F., Friedrich O., Fink R. H. A., Smith G., and Most P., 2006. S100A1 decreases calcium spark frequency and alters their spatial characteristics in permeabilized adult ventricular cardiomyocytes. Cell Calcium.

T. Womelsdorf, J.M. Schoffelen, R. Oostenveld, W. Singer, R. Desimone, A.K. Engel, and P. Fries, “Modulation of neuronal interactions through neuronal synchronization,” *Science,* 316:1609–1612.