eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk/

Spatiotemporal Multi-Resolution Approximation of the Amari Type Neural Field Model

P. Aram[a,e,*], D. R. Freestone[b,f], M. Dewar[c], K. Scerri[d], V. Jirsa[a], D. B. Grayden[b,f], V. Kadirkamanathan[e]

[a]*Theoretical Neuroscience Group, UMR 1106, Institut de Neurosciences des Systemes, 13385 Marseille, France*
[b]*NeuroEngineering Laboratory, Department of Electrical and Electronic Engineering, University of Melbourne, Melbourne, VIC, Australia*
[c]*bit.ly, New York City, USA*
[d]*Department of Systems and Control Engineering, University of Malta, Msida, MSD, Malta*
[e]*Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, UK*
[f]*The Bionics Institute, East Melbourne, VIC, Australia*

Abstract

Neural fields are spatially continuous state variables described by integro-differential equations, which are well suited to describe the spatiotemporal evolution of cortical activations on multiple scales. Here we develop a multi-resolution approximation (MRA) framework for the integro-difference equation (IDE) neural field model based on semi-orthogonal cardinal B-spline wavelets. In this way, a flexible framework is created, whereby both macroscopic and microscopic behavior of the system can be represented simultaneously. State and parameter estimation is performed using the expectation maximization (EM) algorithm. A synthetic example is provided to demonstrate the framework.

Keywords: Neural field model, multi-resolution approximation (MRA), expectation maximization (EM) algorithm

1. Introduction

The human cerebral cortex has a multi-resolution architecture, where spatial scales for information processing range from ion channels, to single neurons, to networks of millions of neurons. This multi-resolution cortical architecture poses major modeling challenges to efficiently describe the brain's dynamics. This paper introduces a multi-resolution data-driven framework for neural field modeling, the multi-resolution approximate integro-difference equation (MRAIDE) to address this challenge.

Neural field models describe the mass action of the central nervous system and are a critical link in our understanding of the biophysics leading to the generation of electroencephalography (EEG) and magnetoencephalography (MEG) signals. The key components of these models typically describe the macroscopic

---

*Corresponding author
Email addresses:* `parham.aram@univ-amu.fr` (P. Aram), `deanrf@unimelb.edu.au` (D. R. Freestone), `md@bit.ly` (M. Dewar), `kenneth.scerri@um.edu.mt` (K. Scerri), `viktor.jirsa@univ-amu.fr` (V. Jirsa), `grayden@unimelb.edu.au` (D. B. Grayden), `visakan@sheffield.ac.uk` (V. Kadirkamanathan)

dynamics of the human brain, but can also be descriptive of finer-scale neurodynamics. Examples are the post-synaptic response kernels (pulse to wave conversion) and the activation function (wave to pulse conversion) (Jirsa and Haken, 1996, 1997). The former can be descriptive of inhibitory and excitatory post-synaptic potentials in populations of microscopic single neural models, such as integrate and fire neurons, or an ensemble response to stimulation in macroscopic neural mass models (Stefanescu and Jirsa, 2008). The latter can be used to model spiking statistics across spatial scales from single neurons (where it can be considered a cumulative distribution function (CDF) of the probability of firing) as mean firing rates of neural masses. The key factor linking the functions of neural mass models to particular spatial scales is the connectivity, hence the ability to infer a multi-resolution connectivity kernel will enable the formulation of a multi-resolution neural field model.

The utility of these models for answering questions in clinical neuroscience and neurology lies in the ability to infer physiological parameters from electrophysiological data. The ability to create patient-specific neural models will contribute to our knowledge base of diseases such as epilepsy and will enable the development of new treatment strategies. Large-scale neuroinformatic efforts, such as the Virtual Brain Project (www.thevirtualbrain.org (Jirsa et al., 2010)), are on the way to enable patient-specific full brain modeling. This is particularly relevant to the advent of new devices that use therapeutic electrical stimulation. Stimulation strategies for devices currently operate in an open loop, where stimulation parameters are chosen using a process of trial and error. Therefore, there exists enormous potential to improve the performance of these devices using systems theory, which in turn requires a suitable dynamic model. Neural mass and neural field models are ideal candidates for estimation algorithms due to their strong links with physiology and their parsimony. The main difference between these two classes is that neural field models represent a cortical sheet, making a continuum approximation to describe the spatiotemporal evolution of neural dynamics (such as mean firing rates or mean membrane potentials), while neural mass models describe ensembles of neurons at a localized point, which may be networked to multiple points. Both neural field and neural mass models make a mean-field approximation to describe the mass action of the brain. It is expected that parameters of these models will be patient-specific and will, therefore, need to be inferred from data.

The multi-resolution architecture of the brain and connectivity in particular will play an important role in this context for multiple reasons. First, there is an on-going debate about the organizational level of the brain at which cognition and behavior is best represented. While clearly single neurons show response changes that map back to learning, memory and motor behavior, much of the recent work in neuroscience suggest that these elementary features must be organized at a higher level to provide the richness and diversity of the behaving system. For example, functional neuroimaging work has confirmed that specific parts of visual cortex are more responsive to particular types of stimuli (e.g. faces), but the specificity of this response is dependent on the large-scale pattern of activity across most visual cortical areas (Haxby et al., 2001). Further, the evolution of diffusion magnetic resonance imaging (dMRI) has led to the development of tensor-

based tractography, which allows for *in vivo* images of the white matter skeleton of any brain. Improvements of the MR sequences (e.g. diffusion spectrum imaging) and tractography analysis have made it possible to use tractography as a standard to define connectivity (Hagmann et al., 2008). Hence individualized connectivity matrices can be developed for brain models enabling patient-specific functional studies and need to be estimated from the experimental data.

The first work (to the authors best knowledge) describing data-driven mesoscopic neural modeling used a neural mass model to fit EEG data (Valdes et al., 1999). This approach was extended to coupled neural masses through a Bayesian estimation scheme called dynamic causal modeling (DCM) (David and Friston, 2003). Following this work, data-driven modeling was extended to continuum field equations that explained the richer dynamics of spatiotemporal neural fields (Galka et al., 2008; Schiff and Sauer, 2008; Daunizeau et al., 2009; Pinotsis et al., 2012). Most recently, a framework was developed where a finite-element model of the neural field (via a dynamic Galerkin projection) was formed, using a basis function decomposition, to transform the PDE neural field equations into a finite-dimensional system to facilitate efficient state and parameter estimation (Freestone et al., 2011). This paper is an extension to the aforementioned study, where we derive a neural field state-space model that accounts for the multi-resolution architecture and spatial dynamics of the human cortex. In this way, a flexible framework is created, whereby both macroscopic and microscopic behavior of the system can be represented simultaneously and completely. Table 1 summarizes all frequently used symbols in the following sections.

2. Multi-Resolution Cortical Architecture

The human neocortex has an inherent multi-resolution architecture. The spatiotemporal neural field can be written as

$$v\left(\mathbf{r}, t\right) = \sum_{j} v_j\left(\mathbf{r}, t\right), \tag{1}$$

where $v\left(\mathbf{r}, t\right)$ is a voltage field that may be measured with a high-resolution multi-electrode array or voltage sensitive dye, $\mathbf{r} \in \mathbb{R}^n$ (where $n \leq 3$) describes the spatial location within the field, t denotes time, and j indexes the different levels of resolution in the field. This multi-resolution structure is depicted in Fig. 1.

The multi-resolution structure has been described by numerous methods. For example, this idea of multi-resolution can be thought of as the generalization of modeling the layers of the cortex separately, where the connectivity kernel, or neural foot-print, has varying widths (Wilson and Cowan, 1973).

Typically, the experimentalist only has access to the net field generated by the multi-resolution architecture of the cortex. Following this, it is natural to ask how do we best describe the multi-resolution structure of the neural system. The solution to this problem was proposed by Breakspear and Stam (2005) in using a wavelet decomposition. This paper is directed at developing a framework for patient-specific multi-resolution neural field models, forming a fundamental link between data and theory.

Symbol	Quantity	Units
	Domain and indices	
Ω	Spatial domain	n.a.
$\mathbb{R}$	The set of all real numbers	n.a.
$\mathbb{Z}$	The set of all integers	n.a.
$\mathbf{r}$	Spatial location	[mm, mm]
t	Time	s
	Model	
y_t	Observation	mV
$v(\mathbf{r}, t)$	Mean membrane potential	mV
$f(v(\mathbf{r}, t))$	Firing rate function	spikes.s^{-1}
$e_t(\mathbf{r})$	Field disturbance, with covariance function $\eta(\mathbf{r} - \mathbf{r}')$	mV
$\boldsymbol{\epsilon}_t$	Observation noise, with covariance matrix Σ_ϵ	mV
$m(\mathbf{r}_n)$	Observation kernel where n is sensor index $n = 1, .., n_y$	n.a.
$w(\mathbf{r})$	Spatial connectivity kernel	n.a.
$h(t)$	Post-synaptic response kernel	n.a.
τ, ξ	Membrane time constant & time constant parameter	s, n.a.
$\delta(t)$	Dirac-delta function	n.a.
	Multi-resolution Approximation	
$\phi_{j,l}(r)$	Scaling function	n.a
$\psi_{j,l}(r)$	Wavelet function	n.a
$a_{j_0,l}, c_{j,l}$	Connectivity kernel approximation and detail coefficients	mV spike^{-1}
$x_{t,j_0,l}, \breve{x}_{t,j,l}$	Field approximation and detail coefficients at a given time	mV spike^{-1}
$N_m(r)$	m^{th} order B-spline function	n.a.
$\varphi_m(r)$	m^{th} order B-spline wavelet function	n.a.
	State-Space Model	
$\boldsymbol{\mu}(r)$	Vector of n_x field scaling and wavelet functions	n.a.
$\boldsymbol{\lambda}(r)$	Vector of $n_{\boldsymbol{\theta}}$ connectivity kernel scaling and wavelet functions	n.a.
$\mathbf{x}_t$	State vector at time t composed of $x_{t,j_0,l}$ and $\breve{x}_{t,j,l}$	mV
$\boldsymbol{\theta}$	Vector of connectivity kernel parameters composed of $a_{j_0,l}$ and $c_{j,l}$	mV.spike^{-1}
$\boldsymbol{\Lambda}_x$	Inner product of field basis functions	n.a.
$\boldsymbol{\Gamma}, \boldsymbol{\Lambda}_{\boldsymbol{\theta}}$	Decomposition connectivity matrices	s, mV.spike^{-1}
$\mathbf{w}_t$	State disturbance vector, with covariance Σ_w	mV
$\mathbf{A}(\boldsymbol{\theta})$	State transition matrix	n.a.
$\mathbf{C}$	Observation matrix	n.a.
	Frequency Analysis	
$\boldsymbol{\nu}$	spatial frequency	cycles/mm
	Estimation	
$\hat{\mathbf{x}}_t^{f-}, \hat{\mathbf{x}}_t^{f}$	Forward prior and posterior state estimates	mV
$\hat{\mathbf{x}}_t^{b}$	Backward posterior state estimates	mV
$\mathbf{P}_t^{f-}, \mathbf{P}_t^{f}$	Forward prior and posterior covariance matrices	mV^2
$\mathbf{P}_t^{b}$	Backward posterior covariance matrices	mV^2
$\mathbf{M}_t$	Cross-covariance matrix	mV^2
$\mathcal{K}_t$	Filter gain	n.a.
$\mathcal{S}_t$	Smoother gain	n.a.
$\boldsymbol{\Xi}_1, \boldsymbol{\Xi}_0$	Expected covariance and cross-covariance matrices	mV^2
$\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}')$	Lower bound on log-likelihood function	n.a.
$\boldsymbol{\nu}_0, \boldsymbol{\nu}_1$	Lower bound coefficient matrices	n.a.
$\boldsymbol{\Upsilon}$	Lower bound coefficient matrix	mV^{-1}.spike

Table 1: **Notation.** The first column of the table gives the symbols for all the notation used in the paper. The second column provides a brief description of the quantity, and the third column provides the SI units where relevant.
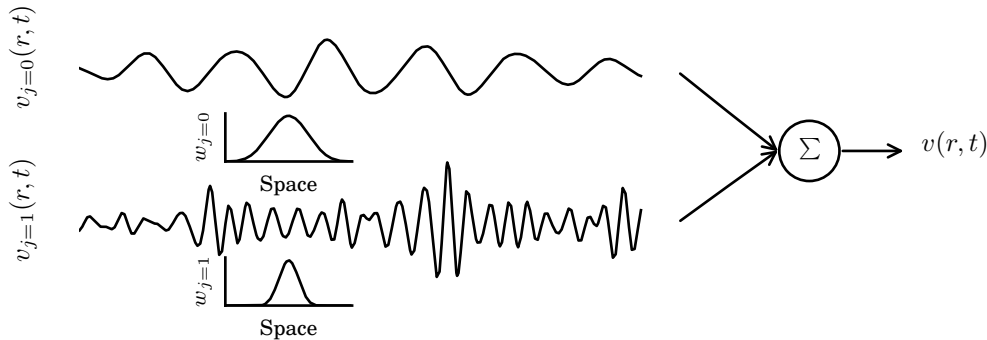
Figure 1: **Multi-layer neural field structure**. The inset figures show the shape of the connectivity kernel components at each spatial resolution.

3. IDE Neural Field Model

In a small network with no time delays the single layer homogeneous neural field model, in the most general form, is described by the equation

$$\underbrace{v\left(\mathbf{r},t\right)}_{\text{field}} = \int_{-\infty}^{t} \underbrace{h\left(t-t'\right)}_{\text{synapse}} \int_{\Omega} \underbrace{w\left(\mathbf{r},\mathbf{r}'\right)}_{\text{connectivity}} \underbrace{f\left(v\left(\mathbf{r}',t'\right)\right)}_{\text{firing}} \mathrm{d}\mathbf{r}'\mathrm{d}t', \tag{2}$$

where the post-synaptic membrane voltage at time t of a population of neurons at position $\mathbf{r}$ within the spatial domain Ω is denoted by $v\left(\mathbf{r},t\right)$. The function $h(t)$ describes the post-synaptic response kernel, $w\left(\mathbf{r},\mathbf{r}'\right)$ describes the connectivity kernel and $f(v(\mathbf{r},t))$ describes the relationship between the mean membrane potential and the mean firing rate. There are various forms of $h(t)$ that have been proposed in the literature, with different levels of complexity, leading to first or second-order differential equations. These are plotted in Fig. 2, showing the second-order (finite rise time) and first-order (instantaneous response) kernels. The second-order synaptic kernel reproduces experimental excitatory and inhibitory post-synaptic potentials (EPSPs and IPSPs) features (Van Rotterdam et al., 1982). Similarly, both the connectivity kernel and the activity function (firing rate) can take various forms. These are also shown in Fig. 2. The form of $w(\mathbf{r},\mathbf{r}')$ typically describes a Laplacian (exponential) or Gaussian decay in connectivity strength from any point in the field (Breakspear et al., 2010; Atay and Hutt, 2005). By assuming the form of $h(t)$ is the same across layers (scales) the Amari style neural field model is formed, incorporating a multi-layer structure into the connectivity kernel. The connectivity kernel then can be written as a sum of weighted (Gaussian, Laplacian or B-Spline) basis functions, where the weights can be thought of as the synaptic efficiency or coupling strength and positive and negative weights describe excitation and inhibition, respectively. In this way, the

5

connectivity kernel typically takes the form of a Mexican hat (Gaussian shape within layer connectivity) or Wizard hat (Laplacian shape within layer) as depicted in Fig. 2.
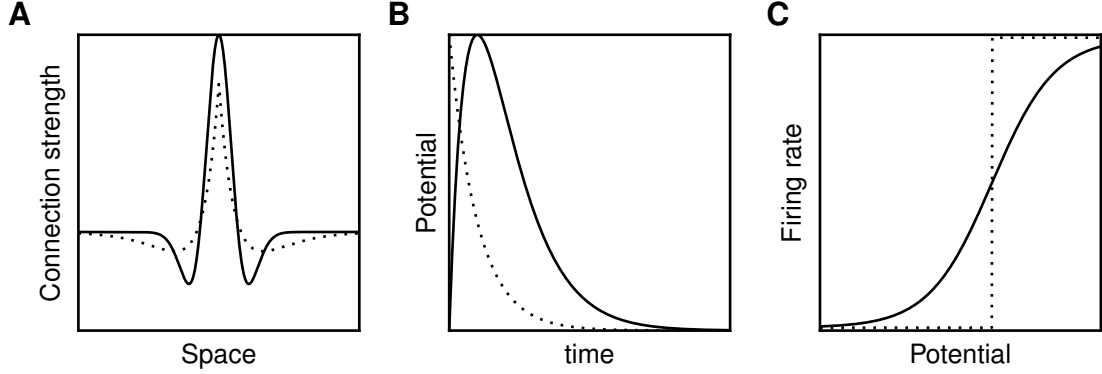

Figure 2: **The IDE neural field model components**. (**A**) The Mexican hat (solid line) and Wizard hat (dotted line) connectivity kernels composed of Gaussian and Laplacian basis functions respectively. (**B**) First order (dotted line) and second order (solid line) synaptic kernels. (**C**) The sigmoidal (solid line) and the Heaviside (dotted line) activation functions.

The stochastic IDE form of the Amari neural field formulation Amari (1977) is given by (see Freestone et al. (2011) for a full derivation)

$$v_{t+1}\left(\mathbf{r}\right) = \xi v_t\left(\mathbf{r}\right) + T_s \int_\Omega w\left(\mathbf{r},\mathbf{r}'\right) f\left(v_t\left(\mathbf{r}'\right)\right) \mathrm{d}\mathbf{r}' + e_t\left(\mathbf{r}\right), \tag{3}$$

where in this first-order (in time) formulation of the model the membrane dynamics are included through the parameter $\xi = 1 - T_s/\tau$, τ is the membrane time constant and T_s is the sampling time step. The term $e_t(\mathbf{r})$ is a zero mean Gaussian disturbance, spatially colored but temporally white, with covariance function

$$cov\left(e_t\left(\mathbf{r}\right), e_{t+t'}\left(\mathbf{r}'\right)\right) = \sigma_d \eta(\mathbf{r} - \mathbf{r}')\delta(t - t'), \tag{4}$$

where $\eta(\mathbf{r} - \mathbf{r}')$ is a spatially homogeneous covariance function, σ_d is disturbance variance and $\delta(\cdot)$ is the Dirac-delta function.

The observation equation describing the intracranial electrophysiological recordings is given by

$$y_t(\mathbf{r}_{n_y}) = \int_\Omega m\left(\mathbf{r}_{n_y} - \mathbf{r}'\right) v_t\left(\mathbf{r}'\right) \mathrm{d}\mathbf{r}' + \epsilon_t(\mathbf{r}_{n_y}), \tag{5}$$

where $m\left(\mathbf{r}_{n_y} - \mathbf{r}'\right)$ is the observation kernel at location $\mathbf{r}_{n_y}$ and $\boldsymbol{\epsilon}_t \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Sigma}_\epsilon\right)$ is an independent and identically distributed (i.i.d.) Gaussian white noise process with the covariance matrix $\boldsymbol{\Sigma}_\epsilon = \sigma_\epsilon^2 \mathbf{I}_{n_y}$, where $\mathbf{I}$ denotes the identity matrix. The observation kernel models the pickup range of the electrical fields of the brain. The lead field parameters, which incorporate information about a realistic head model (Jirsa and McIntosh, 2007), are not included in the observation equation as the intracranial recordings are only

6

considered here.

4. MRA of the IDE in State-Space

The purpose of forming the multi-resolution approximation (MRA) of the integro-difference equation (IDE) is to form a multi-resolution finite-dimensional state-space model that best captures the spatiotemporal characteristics of the neural field. Given such a model, it is then possible to perform state and parameter estimation in an efficient and robust manner, thus allowing for subject-specific models. The goal of this section is to demonstrate how to formulate a state-space model in a canonical form such that standard algorithms may be applied to estimate the multi-resolution connectivity structure.

The multi-resolution approximation (Mallat, 1989) of the neural IDE (MRAIDE) is obtained by decomposing both the field, $v_t(\cdot)$, and the connectivity kernel, $w(\cdot)$, (assuming square-integrable functions) using translations and dilations of a scaling function $\phi(\mathbf{r})$ and a mother wavelet $\psi(\mathbf{r})$. Considering a one-dimensional field, the static connectivity kernel is decomposed as,

$$w\left(r\right) = \sum_{l \in \mathbb{Z}} a_{j_0,l} \underbrace{\phi_{j_0,l}\left(r\right)}_{\text{scaling functions}} + \sum_{j \geq j_0}^{\infty} \sum_{l \in \mathbb{Z}} c_{j,l} \underbrace{\psi_{j,l}\left(r\right)}_{\text{wavelets}}, \tag{6}$$

where $a_{j_0,l}$ are the approximation coefficients at the lowest scale j_0, and $c_{j,l}$ are the detail coefficients at different scales j, with

$$\phi_{j,l}\left(r\right) = 2^{\frac{j}{2}} \phi\left(2^j r - l\right) \tag{7}$$

$$\psi_{j,l}\left(r\right) = 2^{\frac{j}{2}} \psi\left(2^j r - l\right). \tag{8}$$

The parameters j and l control the scale (dilation) and translation (spatial shift), respectively. The number of spatial shifts, l, is dependent on the scale, j, and can be chosen such that the basis functions cover the spatial domain of interest. For example, at scales corresponding to higher spatial frequencies a greater number of wavelets, and thus spatial shifts are required to adequately sample the spatial characteristics of the field. The number of scales depends on the spectral characteristics of the field, where higher spatial frequencies require more scales. Spatial frequency analysis will be described later in Section 5.2 in greater detail. Scaling functions retain the lowest frequency components, due to their low-pass properties, while the wavelet functions extract successively higher frequency components due to their band-pass characteristics.

The dynamic neural field is decomposed using

$$v_t\left(r\right) = \sum_{l \in \mathbb{Z}} x_{t,j_0,l} \underbrace{\phi_{j_0,l}\left(r\right)}_{\text{scaling functions}} + \sum_{j \geq j_0}^{\infty} \sum_{l \in \mathbb{Z}} \check{x}_{t,j,l} \underbrace{\psi_{j,l}\left(r\right)}_{\text{wavelets}}, \tag{9}$$

7

where $x_{t,j_0,l}$ and $\check{x}_{t,j,l}$ are the dynamic coefficients of the expansion, constituting the state vector at time t. The state vector has the same units as the neural field (volts). However, the states are weights for particular levels of resolution, where $v_t(r)$ is the total net field. The decomposition enables a separation of the spatial characteristics, which are considered static functions that are weighted by dynamic coefficients.

Eqs. (6) and (9) are infinite series expansions but must be truncated to some level j, in order to solve the estimation problem. In other words, the neural field must be band-limited up to a given spatial frequency.

Now we will simplify the notation by defining vectors that contain all the translations of the wavelets and scaling functions

$$\boldsymbol{\phi}_{j_0}(r) \triangleq \{\phi_{j_0,l}(r) : l \in \mathbb{Z}\} \tag{10}$$

$$\boldsymbol{\psi}_j(r) \triangleq \{\psi_{j,l}(r) : l \in \mathbb{Z}\} \tag{11}$$

$$\mathbf{a}_{j_0} \triangleq \{a_{j_0,l} : l \in \mathbb{Z}\} \tag{12}$$

$$\mathbf{c}_j \triangleq \{c_{j,l} : l \in \mathbb{Z}\} \tag{13}$$

$$\mathbf{x}_{t,j_0} \triangleq \{x_{t,j_0,l} : l \in \mathbb{Z}\} \tag{14}$$

$$\check{\mathbf{x}}_{t,j} \triangleq \{\check{x}_{t,j,l} : l \in \mathbb{Z}\} . \tag{15}$$

Following this we can define the vectors

$$\boldsymbol{\theta}^\top \triangleq \begin{bmatrix} \mathbf{a}_{j_0}^\top & \mathbf{c}_{j_0}^\top & \mathbf{c}_{j_0+1}^\top & \cdots & \mathbf{c}_j^\top \end{bmatrix} \tag{16}$$

$$\mathbf{x}_t^\top \triangleq \begin{bmatrix} \mathbf{x}_{t,j_0}^\top & \check{\mathbf{x}}_{t,j_0}^\top & \check{\mathbf{x}}_{t,j_0+1}^\top & \cdots & \check{\mathbf{x}}_{t,j}^\top \end{bmatrix} \tag{17}$$

$$\boldsymbol{\lambda}^\top(r) \triangleq \begin{bmatrix} \boldsymbol{\phi}_{j_0}^\top(r) & \boldsymbol{\psi}_{j_0}^\top(r) & \boldsymbol{\psi}_{j_0+1}^\top(r) & \cdots & \boldsymbol{\psi}_j^\top(r) \end{bmatrix} \tag{18}$$

$$\boldsymbol{\mu}^\top(r) \triangleq \begin{bmatrix} \boldsymbol{\phi}_{j_0}^\top(r) & \boldsymbol{\psi}_{j_0}^\top(r) & \boldsymbol{\psi}_{j_0+1}^\top(r) & \cdots & \boldsymbol{\psi}_j^\top(r) \end{bmatrix} , \tag{19}$$

which allow us to write the connectivity kernel and neural field decomposition in the compact form of

$$w(r) \approx \boldsymbol{\theta}^\top \boldsymbol{\lambda}(r) \tag{20}$$

$$v_t(r) \approx \boldsymbol{\mu}^\top(r) \mathbf{x}_t. \tag{21}$$

Now we have the compact form of the decomposition we can substitute this into the neural field Eq. (3) giving

$$\boldsymbol{\mu}^\top(r) \mathbf{x}_{t+1} = \xi \boldsymbol{\mu}^\top(r) \mathbf{x}_t + T_s \int_\Omega \boldsymbol{\theta}^\top \boldsymbol{\lambda}(r - r') f\left(\boldsymbol{\mu}^\top(r') \mathbf{x}_t\right) \mathrm{d}r' + e_t(r). \tag{22}$$

The aim of the next part of the derivation is to isolate the state vector on the left hand side of the equation. The conceptual framework and the key results can be interpreted without following this derivation in detail.

To isolate the state vector we first multiply both sides of Eq. (22) by $\boldsymbol{\mu}(r)$ and integrate over space giving

$$
\int_{\Omega} \boldsymbol{\mu}(r) \boldsymbol{\mu}^{\top}(r) \, dr \mathbf{x}_{t+1} = \xi \int_{\Omega} \boldsymbol{\mu}(r) \boldsymbol{\mu}^{\top}(r) \, dr \mathbf{x}_t + T_s \int_{\Omega} \boldsymbol{\mu}(r) \int_{\Omega} \boldsymbol{\theta}^{\top} \boldsymbol{\lambda}(r - r') f\left(\boldsymbol{\mu}^{\top}(r') \mathbf{x}_t\right) \, dr' dr
$$
$$
+ \int_{\Omega} \boldsymbol{\mu}(r) e_t(r) \, dr. \tag{23}
$$

To isolate the state vector, we now define the matrix

$$
\boldsymbol{\Lambda}_x \triangleq \int_{\Omega} \boldsymbol{\mu}(r) \boldsymbol{\mu}^{\top}(r) \, dr, \tag{24}
$$

and substitute into Eq. (23) giving

$$
\boldsymbol{\Lambda}_x \mathbf{x}_{t+1} = \xi \boldsymbol{\Lambda}_x \mathbf{x}_t + T_s \int_{\Omega} \boldsymbol{\mu}(r) \int_{\Omega} \boldsymbol{\theta}^{\top} \boldsymbol{\lambda}(r - r') f\left(\boldsymbol{\mu}^{\top}(r') \mathbf{x}_t\right) \, dr' dr + \int_{\Omega} \boldsymbol{\mu}(r) e_t(r) \, dr. \tag{25}
$$

Cross-multiplying by $\boldsymbol{\Lambda}_x^{-1}$ gives the state transition equation

$$
\mathbf{x}_{t+1} = \xi \mathbf{x}_t + T_s \boldsymbol{\Lambda}_x^{-1} \int_{\Omega} \boldsymbol{\mu}(r) \int_{\Omega} \boldsymbol{\theta}^{\top} \boldsymbol{\lambda}(r - r') f\left(\boldsymbol{\mu}^{\top}(r') \mathbf{x}_t\right) \, dr' dr + \boldsymbol{\Lambda}_x^{-1} \int_{\Omega} \boldsymbol{\mu}(r) e_t(r) \, dr. \tag{26}
$$

A very nice property of the MRAIDE framework is the disturbance term becomes a vector valued, zero-mean normally distributed, white noise process

$$
\mathbf{w}_t = \boldsymbol{\Lambda}_x^{-1} \int_{\Omega} \boldsymbol{\mu}(r) e_t(r) \, dr, \tag{27}
$$

with covariance (see Freestone et al. (2011) for proof)

$$
\boldsymbol{\Sigma}_w = \boldsymbol{\Lambda}_x^{-1} \iint_{\boldsymbol{\Omega}} \boldsymbol{\mu}(r) \eta(r - r') \boldsymbol{\mu}^{\top}(r') \, dr' dr \boldsymbol{\Lambda}_x^{-\top}. \tag{28}
$$

The state-transition equation can be simplified if the chosen family of scaling and wavelet functions are isotropic where

$$
\boldsymbol{\lambda}_i(r - r') = \boldsymbol{\lambda}(2\mathbf{c}_i + r' - r), \tag{29}
$$

where c_i is the centre of the ith basis function of the kernel decomposition. Following this, we can switch the order of operations with the inner product and convolution in Eq. (26). With this switch the convolution is now between the scaling functions and wavelets for the field and the scaling functions and wavelets for the connectivity kernel. This convolution can be calculated analytically when using B-spline scaling and wavelet functions and can be predefined, since these are static functions. This has major computational advantages

when solving the estimation problem. The predefined convolution can be written as

$$[\Gamma(r')]_{:i} = T_s \Lambda_x^{-1} \int_\Omega \boldsymbol{\mu}(r) \boldsymbol{\lambda}(2c_i + r' - r) \, dr, \tag{30}$$

where $[\Gamma(r')]_{:i}$ denotes the ith column of $[\Gamma(r')]$ giving

$$\mathbf{x}_{t+1} = \xi\mathbf{x}_t + \int_\Omega \Gamma(r') f\left(\boldsymbol{\mu}^\top(r') \mathbf{x}_t\right) dr' \boldsymbol{\theta} + \mathbf{w}_t. \tag{31}$$

The observation equation of the state-space model is found by substituting decomposition (21) into Eq. (5) giving

$$\mathbf{y}_t = \mathbf{C}\mathbf{x}_t + \boldsymbol{\epsilon}_t, \tag{32}$$

where each element of the observation matrix, $\mathbf{C}$, is given by

$$\mathbf{C}_{ik} \triangleq \int_\Omega m(r_i - r') \boldsymbol{\mu}_k(r') \, dr'. \tag{33}$$

5. B-Spline MRAIDE in State-Space

5.1. A Brief Introduction to B-Spline Wavelets

This section is included to provide the interested reader with an overview of the theory used in constructing the wavelets that form the basis of the multi-resolution model. Although a deep understanding of the theory presented in this section is not required for an understanding of the key results of the paper, it serves as a reference for implementing the methods.

B-spline functions are one of the most basic elements for wavelet construction. These functions are composed of piecewise polynomials combined smoothly at some breaking points (knots), which are equally spaced in the case of cardinal B-splines. The degree of smoothness is determined by the order of splines (Goswami and Chan, 1999), where the higher the order the smoother the resulting B-spline becomes (see Fig. 3).

B-spline wavelet and scaling functions were formulated independently by Chui and Wang (1992a), Chui and Wang (1992b) and Unser et al. (1993). The scaling functions are m^{th} order B-splines and the compact support wavelet functions are a linear combination of scaling functions. B-spline wavelet and scaling functions approach optimal time/frequency localization as the order of the spline increases. In fact, for the cubic B-spline scaling and wavelet functions it is already close to the optimal limit for Gaussian functions (Unser, 1999).

B-spline wavelet and scaling functions are particularly suited in this framework due to the property of being able to analytically define the convolution and inner product to form MRAIDE components. The
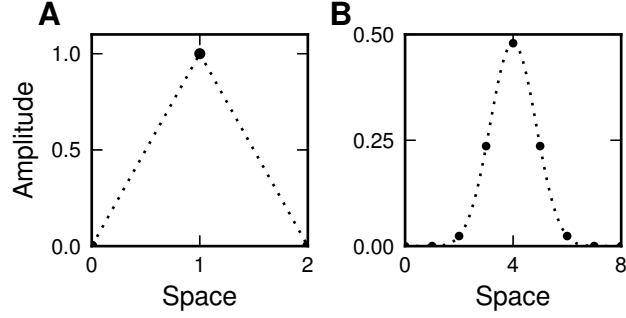
Figure 3: **Examples of cardinal B-spline functions**. Piecewise polynomial functions (dotted lines) are joined at the break points (black circles). (**A**) B-spline function of order 2 (linear B-spline). (**B**) B-spline function of order 8.

following summarizes some important characteristics of B-splines relevant to the proposed modeling framework.

The mth order cardinal B-spline function is defined by the recurrence relation (Chui and Wang, 1992b)

$$N_m\left(r\right) = \left(N_{m-1} * N_1\right)\left(r\right) = \int_0^1 N_{m-1}\left(r - r'\right) \, \mathrm{d}r', \tag{34}$$

where $m > 1$, $*$ denotes convolution, and $N_1\left(r\right)$ is the characteristic function of the unit interval $[0, 1)$, also known as indicator function, i.e.

$$N_1\left(r\right) = \begin{cases} 1 & \text{if } 0 \leq r < 1, \\ 0 & \text{elsewhere.} \end{cases} \tag{35}$$

Following this, a B-spline of any order, $N_m(r)$, can be computed using the recursive expression defined by De Boor (2001)

$$N_m\left(r\right) = \frac{r}{m-1}N_{m-1}\left(r\right) + \frac{m-r}{m-1}N_{m-1}\left(r-1\right) \quad m > 1. \tag{36}$$

The most commonly used form of splines is the cubic spline ($m = 4$), comprised of third degree polynomials added together at the joining points, with an explicit expression given by

$$3!N_4\left(r\right) = \begin{cases} r^3 & \text{if } 0 \leq r \leq 1, \\ 4 - 12r + 12r^2 - 3r^3 & \text{if } 1 \leq r \leq 2, \\ -44 + 60r - 24r^2 + 3r^3 & \text{if } 2 \leq r \leq 3, \\ 64 - 48r + 12r^2 - r^3 & \text{if } 3 \leq r \leq 4, \\ 0 & \text{elsewhere.} \end{cases} \tag{37}$$

One important feature of the B-spline functions is the partition of unity property, meaning that the unity

11

(zero spatial frequency) can be expressed as a linear sum of B-splines, i.e

$$\sum_l N_m(r+l) \equiv 1. \tag{38}$$

This is important as the underlying field can be reconstructed when it is saturated without introducing numerical errors which is not the case for example when Gaussian basis functions are used.

The convolution and inner product of B-spline functions are required in order to formulate the multi-resolution framework described in the previous section (construction of $\mathbf{\Lambda}_x$ and $\mathbf{\Gamma}$). To show how the convolution of two B-splines is calculated, Eq. (34) can be rewritten as $(m-1)$ convolutions of the indicator function with itself

$$N_m(r) = \underbrace{(N_1 * N_1 * \cdots * N_1)}_{m-1 \quad \text{convolutions}}(r). \tag{39}$$

Using the associativity property of convolution, we have

$$N_m(r) * N_{m'}(r) = \underbrace{\overbrace{(N_1 * \cdots * N_1)}^{m-1 \quad \text{convolutions}}(r) * \overbrace{(N_1 * \cdots * N_1)}^{m'-1 \quad \text{convolutions}}(r)}_{m+m'-1 \quad \text{convolutions}}$$

$$= N_{m+m'}(r). \tag{40}$$

A direct consequence of Eq. (40) is the inner product between two B-splines is another translated B-spline such that

$$\langle N_m(r-l_1), N_{m'}(r-l_2) \rangle = N_{m+m'}(m+l_1-l_2)$$
$$= N_{m+m'}(m'+l_2-l_1), \tag{41}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product. This holds as the support of $N_m(r)$ is $[0, m]$ and is symmetric with respect to $r = \frac{m}{2}$, i.e. $N_m\left(\frac{m}{2}+r\right) = N_m\left(\frac{m}{2}-r\right)$ (see Appendix A for explicit derivation). The values of $N_{m+m'}$ in Eq. (41) can be easily determined recursively by evaluating Eq. (36) at integer points, i.e.

$$\begin{cases} N_2(k) = \delta(k-1) & k \in \mathbb{Z}, \\ N_m(k) = \frac{k}{m-1}N_{m-1}(k) + \frac{m-k}{m-1}N_{m-1}(k-1) & k = 1, 2, \ldots, m-1. \end{cases} \tag{42}$$

Note $N_m(k) = 0$ for $k \leq 0$ or $k \geq m$.

The multi-resolution approximation using B-spline functions is completed by defining a two-scale relation pair, which relates the scaling functions and the wavelets at a given scale with the scaling function at the next higher scale. For B-spline scaling and wavelet functions of order m the two-scale relation pair takes

the form of (Chui and Wang, 1992b)

$$N_m(r) = \sum_{n=0}^{m} p_n N_m(2r - n) \tag{43}$$

$$\varphi_m(r) = \sum_{n=0}^{3m-2} q_n N_m(2r - n), \tag{44}$$

where

$$p_n = 2^{-m+1} \binom{m}{n} \quad 0 \leq n \leq m \tag{45}$$

$$q_n = (-1)^n 2^{-m+1} \sum_{l=0}^{m} \binom{m}{l} N_{2m}(n - l + 1), \ \ 0 \leq n \leq 3m - 2 \tag{46}$$

Fig. 4 shows the cubic B-spline scaling function and its associated wavelet function with dilation and translation parameters set to zero. Note in general the supports of B-spline scaling and wavelet functions are given by
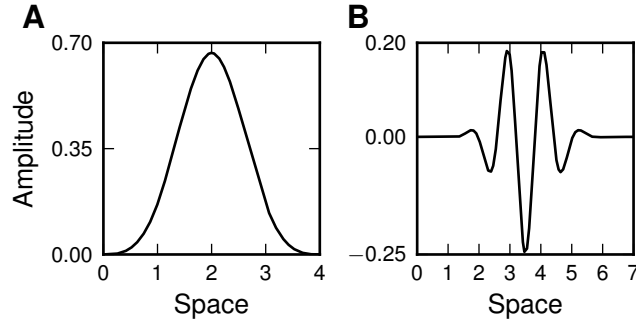

Figure 4: **Examples of B-spline scaling and wavelet functions**. (**A**) The cubic B-spline scaling function. (**B**) The corresponding wavelet function.

$$\text{supp}(N_{m;j,l}) = \left[\frac{l}{2^j}, \frac{m+l}{2^j} \right] \tag{47}$$

$$\text{supp}(\varphi_{m;j,l}) = \left[\frac{l}{2^j}, \frac{2m-1+l}{2^j} \right], \tag{48}$$

where subscripts j and l in Eqs. (47) and (48) are introduced to $N_m(r)$ and $\varphi_m(r)$ to indicate the scale and translation parameters as defined in Eqs (7) and (8).

By exploiting Eqs. (40) and (41), the integrals in Eqs. (24) and (30) can be computed analytically. In constructing $\mathbf{\Lambda}_x$, it is important to note that B-spline scaling and wavelet functions possess the following

orthogonality properties (Unser et al., 1993):

$$\langle \varphi_{m;j_1,l_1}(r), \varphi_{m;j_2,l_2}(r) \rangle = 0 \quad \text{for} \quad j_1 \neq j_2 \tag{49}$$

$$\langle N_{m;j_1,l_1}(r), \varphi_{m;j_2,l_2}(r) \rangle = 0 \quad \text{for} \quad j_1 \leq j_2. \tag{50}$$

Note from Eq. (49) B-spline wavelets are semi-orthogonal as they are not orthogonal with respect to translation on a given scale.

In order to analytically calculate elements of $\mathbf{\Lambda}_x$ and $\mathbf{\Gamma}$, the scaling and wavelet basis functions must be expanded in terms of N_m at the appropriate scale using the two-scale relation pair defined in Eqs. ((43)-(46)). In this paper, 4th order cardinal B-spline scaling and wavelet functions are used, resulting in 8th and 12th order B-spline functions when calculating Eq. (24) and Eq. (67) respectively whose values at integer points are given in Table 2.

k	$3!N_4(k)$	$7!N_8(k)$	$11!N_{12}(k)$
1	1	1	1
2	4	120	$2,036$
3		$1,191$	$152,637$
4		$2,416$	$2,203,488$
5			$9,738,114$
6			$15,724,248$

Table 2: **Cardinal B-Splines at the knot sequence**. Table entries are calculated using Eq. (42) and can be also found in Goswami and Chan (1999).

5.2. Frequency Analysis

To determine the number of basis functions required to reconstruct the field and the level of approximation in the model, j, the frequency response of the B-spline wavelets is required. This is given by (see Appendix B)

$$\mathcal{F}(\varphi_m(r)) = \frac{1}{2} \left(\frac{1 - \exp(-\pi i \nu)}{\pi i \nu} \right)^m \sum_{n=0}^{3m-2} q_n \exp(-\pi i n \nu), \tag{51}$$

where ν is the spatial frequency and $i = \sqrt{-1}$. From Eq. (51), the frequency response of the wavelets and hence the level of approximation, j, can be determined. In this way, we choose wavelets up to level j that can fully describe the spatial frequency contents of the underlying dynamics from the observed field.

6. State and Parameter Estimation

The aim of this section is to describe the joint state-parameter estimation method used in this study, the expectation maximization (EM) algorithm. Here we provide a general description of the algorithm and also discuss the specific formulation for MRAIDE framework.

There exist many variations of state and parameter estimation methods for state-space models. The choice of algorithm is normally determined by the dimension of the system, whether the system is linear or nonlinear and whether or not the parameters are considered static or dynamic. Up to this point of the paper, we have not made any assumptions on any of these criteria for choosing an estimation algorithm. The MRAIDE neural field model formulation may be used in any of the above scenarios.

Nevertheless, when describing a system, a choice needs to be made regarding the complexity of the mathematical description to be adopted. Such a choice is usually based on the principle of parsimony (Burnham and Anderson, 2002), that is; choosing the simplest model capable of capturing the system behavior of interest. Such a choice in favor of a simpler model has the added advantages of a smaller parameter space and therefore more accurate estimates and computationally efficient methods.

To demonstrate the MRAIDE estimation framework we shall consider the parameters to be static and we shall also assume that the relationship between the mean membrane potential and the mean firing rate is predominantly linear. Note that by using a linear activation function in the estimator we are not assuming that there is necessarily a linear relationship between the membrane voltage and firing rate, but that the statistics of the neural masses fall within a linear operating region over the estimation time period. This allows for the adoption of the EM algorithm and thus results in computationally efficient and accurate estimates. Consequently, any loss in model detail may be offset by the estimation accuracy.

By assuming a linear activation function of the form

$$f(v_t(\mathbf{r})) = \varsigma v_t(\mathbf{r}),\tag{52}$$

the state transition equation (26) simplifies to

$$\mathbf{x}_{t+1} = \xi \mathbf{x}_t + \varsigma T_s \mathbf{\Lambda}_x^{-1} \int_\Omega \boldsymbol{\mu}(\mathbf{r}) \int_\Omega \boldsymbol{\theta}^\top \boldsymbol{\lambda}(\mathbf{r} - \mathbf{r}') \boldsymbol{\mu}^\top(\mathbf{r}') \, \mathrm{d}\mathbf{r}' \mathrm{d}\mathbf{r} \mathbf{x}_t + \mathbf{w}_t.\tag{53}$$

Now we can further simplify the system by defining

$$\mathbf{\Lambda}_\theta \triangleq \int_\Omega \boldsymbol{\mu}(r) \int_\Omega \boldsymbol{\theta}^\top \boldsymbol{\lambda}(r - r') \boldsymbol{\mu}^\top(r) \, \mathrm{d}r'\mathrm{d}r,\tag{54}$$

to give the state equation

$$\mathbf{x}_{t+1} = \mathbf{A}(\boldsymbol{\theta})\mathbf{x}_t + \mathbf{w}_t,\tag{55}$$

15

with the parameterized transition matrix

$$\mathbf{A}(\boldsymbol{\theta}) = T_s \varsigma \boldsymbol{\Lambda}_x^{-1} \boldsymbol{\Lambda}_\theta + \xi \mathbf{I}. \tag{56}$$

The problem of joint state and parameter estimation for the state-space model given by Eq. (32) and Eq. (55) can now be formulated as

$$\hat{\mathbf{X}}, \hat{\boldsymbol{\theta}} = \arg \max_{\mathbf{X}, \boldsymbol{\theta}} p(\mathbf{Y}; \boldsymbol{\theta}), \tag{57}$$

where $\mathbf{X}$ and $\mathbf{Y}$ denote the entire sequences of the states and observations respectively, $\boldsymbol{\theta}$ is the parameter set and $p(\mathbf{Y}; \boldsymbol{\theta})$ denotes the probability density function of the observations parameterized by $\boldsymbol{\theta}$, known as the likelihood function. This dual estimation problem cannot be solved separately, as for the state estimation the knowledge of the system's parameters is required and for the parameter estimation the states of the system need to be known. One approach is to iteratively estimate the states and parameters of the system and monitor the convergence of the algorithm. A natural framework to perform this is the well known EM algorithm (Dempster et al., 1977; Shumway and Stoffer, 2000) to infer both the parameter set and the states from the observations corresponding to the connectivity kernel, the neural field and electrophysiological data in our application. The EM algorithm is applicable in two main situations: first is when the observations are incomplete due to a faulty observation process or existing limitations, and the second is when the direct maximization of the likelihood function is difficult but can be simplified by assuming the existence of an additional data referred to as hidden data (Bilmes, 1998), which are the weights of the field basis functions in this case. The EM algorithm, when used in the latter context (Dewar et al., 2009), yields the maximum likelihood kernel estimate, i.e.

$$\boldsymbol{\theta}_{\mathrm{ML}} = \arg \max_{\boldsymbol{\theta}} \ p(\mathbf{Y}; \boldsymbol{\theta}), \tag{58}$$

and the posterior distribution of the field over time.

Both the E and the M steps of the EM algorithm find increasingly tighter lower bounds on the likelihood of the parameters so that, at convergence, the maximum of the bound corresponds to the maximum of the likelihood. For state-space models, the E-step corresponds to the smoothing problem (Haykin, 2001), which can be solved using the Rauch Tung Streibel (RTS) smoother (Rauch et al., 1965). For the M-step, a lower bound on the likelihood function is constructed, where the steps needed to compute the required quantities are very dependent on the particular application. This will be described in detail for our problem.

In summary, to solve the joint state and parameter estimation for linear state-space models one can adapt the EM algorithm, solving the smoothing problem for the E-step and constructing and maximizing the lower bound in the M-step. A schematic representation of the EM algorithm is represented in Fig. 5. Note the algorithm can be initialized by either a random parameter set or a sequence of the random bounded state vectors. Here the latter is chosen as shown in the block diagram ensuring the resultant initial kernel
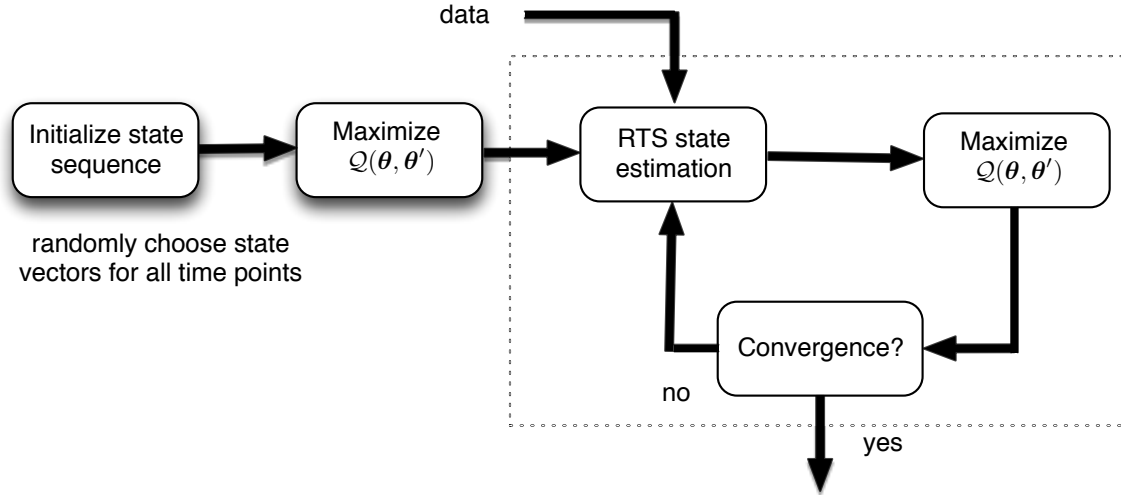
Figure 5: **Schematic of the EM algorithm**. The RTS smoother (E-step) and the maximization of the lower bound, $\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}')$, (M-step) are shown. The dashed box shows the iterative two step algorithm once it is initialized.

estimate leads to a stable system.

6.1. E-Step

As mentioned earlier, the E-step for state-space models is equivalent to the smoothing problem. We use the RTS smoother, given in Table 3 for completeness, in order to infer the states of the system. The RTS smoother uses the Kalman filter (Kalman, 1960) in forward pass (Steps 1 and 2 in Table 3), along with a backward iteration to calculate its outputs (Steps 3 and 4 in Table 3): smoothed states $\hat{\mathbf{x}}_t^b$ and covariances, $\mathbf{P}_t^b = \text{cov}(\mathbf{x}_t)$. Following Gibson and Ninness (2005), an additional backwards recursion is also included (Step 5 in Table 3) to calculate cross-covariance matrix, $\mathbf{M}_t = \text{cov}(\mathbf{x}_t, \mathbf{x}_{t+1})$, which is also required in forming the lower bound on the likelihood function. Note that the following quantities from the forward iteration should be stored when the RTS smoother is implemented: predicted and corrected estimates for both state estimates $(\hat{\mathbf{x}}_t^{f-}, \hat{\mathbf{x}}_t^f)$ and covariance matrices $(\mathbf{P}_t^{f-}, \mathbf{P}_t^f)$. Also, the final value of the Kalman gain, $\mathcal{K}_T$, should be stored in order to initialize the backward cross-covariance computation.

6.2. M-Step

The lower bound on the log-likelihood function is given by the expectation of the complete (hidden and observable) data log-likelihood (Bishop, 2006)

$$\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}') = \mathbf{E}_{\boldsymbol{\theta}'} \left[2 \ln p(\mathbf{X}, \mathbf{Y}; \boldsymbol{\theta}) \right], \tag{59}$$

where $p(\mathbf{X}, \mathbf{Y}; \boldsymbol{\theta})$ is the joint probability distribution of the states and the observations and $\mathbf{E}_{\boldsymbol{\theta}'}[\cdot]$ denotes expectation taken with respect to the marginal distribution of the states conditioned on the observed field

17

1. Forward initialization $$\hat{\mathbf{x}}_0, \mathbf{P}_0$$
2. Forward iteration: for $t \in \{0, \cdots, T-1\}$, calculate the predicted state and the predicted covariance matrix $$\hat{\mathbf{x}}_{t+1}^{f-} = \mathbf{A}\hat{\mathbf{x}}_t^f$$ $$\mathbf{P}_{t+1}^{f-} = \mathbf{A}\mathbf{P}_t^f\mathbf{A}^\top + \mathbf{\Sigma}_e$$ compute the filter gain, the filtered state and the filtered covariance matrix $$\mathcal{K}_{t+1} = \mathbf{P}_{t+1}^{f-}\mathbf{C}^\top(\mathbf{C}\mathbf{P}_{t+1}^{f-}\mathbf{C}^\top + \mathbf{\Sigma}_\varepsilon)^{-1}$$ $$\hat{\mathbf{x}}_{t+1}^f = \hat{\mathbf{x}}_{t+1}^{f-} + \mathcal{K}_{t+1}(\mathbf{y}_{t+1} - \mathbf{C}\hat{\mathbf{x}}_{t+1}^{f-})$$ $$\mathbf{P}_{t+1}^f = (\mathbf{I} - \mathcal{K}_{t+1}\mathbf{C})\mathbf{P}_{t+1}^{f-}$$
3. Backward initialization $$\mathbf{P}_T^b = \mathbf{P}_T^f, \quad \hat{\mathbf{x}}_T^b = \hat{\mathbf{x}}_T^f$$
4. Backward iteration: for $t \in \{T-1, \cdots, 0\}$ compute the smoother gain, the smoothed state and the smoothed covariance matrix $$\mathcal{S}_t = \mathbf{P}_t^f\mathbf{A}^\top\left[\mathbf{P}_{t+1}^{f-}\right]^{-1}$$ $$\hat{\mathbf{x}}_t^b = \hat{\mathbf{x}}_t^f + \mathcal{S}_t(\hat{\mathbf{x}}_{t+1}^b - \hat{\mathbf{x}}_{t+1}^{f-})$$ $$\mathbf{P}_t^b = \mathbf{P}_t^f + \mathcal{S}_t(\mathbf{P}_{t+1}^b - \mathbf{P}_{t+1}^{f-})\mathcal{S}_t^\top$$
5. Compute the smoothed cross covariance matrix, initialization $$\mathbf{M}_T = (\mathbf{I} - \mathcal{K}_T\mathbf{C})\mathbf{A}\mathbf{P}_{T-1}^b$$ for $t \in \{T-1, \cdots, 1\}$ compute $$\mathbf{M}_t = \mathbf{P}_t^f\mathcal{S}_{t-1}^\top + \mathcal{S}_t(\mathbf{M}_{t+1} - \mathbf{A}\mathbf{P}_t^f)\mathcal{S}_{t-1}^\top$$

Table 3: **Algorithm for the RTS smoother (E-step).** This table shows the steps in the Rauch-Tung-Striebel smoother algorithm.

and the current estimate of the parameter set, $\boldsymbol{\theta}'$ i.e. $p(\mathbf{X} \mid \mathbf{Y}; \boldsymbol{\theta}')$. Note that $p(\mathbf{X} \mid \mathbf{Y}; \boldsymbol{\theta}')$, is in fact the smoother output which is obtained at the E-step.

For the linear state-space model given by Eqs. (32) and (55), the bound defined by Eq. (59) is a quadratic of the form (see Appendix C for derivation)

$$\mathcal{Q}\left(\boldsymbol{\theta}, \boldsymbol{\theta}'\right) = \beta + 2T_s\varsigma\left(\boldsymbol{v}_0 - \boldsymbol{v}_1\right)\boldsymbol{\theta} - T_s\varsigma\boldsymbol{\theta}^\top\boldsymbol{\Upsilon}\boldsymbol{\theta}, \tag{60}$$

where β is constant with respect to $\boldsymbol{\theta}$. Equation (60) is maximum at

$$\boldsymbol{\theta} = \boldsymbol{\Upsilon}^{-\top}\left(\boldsymbol{v}_0 - \boldsymbol{v}_1\right)^\top, \tag{61}$$

where

$$\boldsymbol{v}_0 = \sum_{i,j=1}^{n_x} [\boldsymbol{\Xi}_0]_{i,j}[\boldsymbol{\Gamma}_1]_{j,i} \tag{62}$$

$$\boldsymbol{v}_1 = \xi \sum_{i,j=1}^{n_x} [\boldsymbol{\Xi}_1]_{i,j}[\boldsymbol{\Gamma}_1]^{j,i} \tag{63}$$

$$\boldsymbol{\Upsilon} = T_s\varsigma \sum_{i,j=1}^{n_x} [\boldsymbol{\Xi}_1]_{i,j}[\boldsymbol{\Gamma}_2]^{j,i}, \tag{64}$$

and $[\cdot]_{p,q}$ denotes the (p,q)-element of the matrix, $[\cdot]^{p,q}$ denotes the (p,q)-block of the block matrix and where

$$[\boldsymbol{\Gamma}_1]^{j,i} = \sum_{k=1}^{n_x} \left[\boldsymbol{\Sigma}_w^{-1}\boldsymbol{\Lambda}_x^{-1}\right]_{jk}[\mathbf{U}]^{k,i}, \tag{65}$$

$$[\boldsymbol{\Gamma}_2]^{j,i} = \sum_{k,m=1}^{n_x} [\mathbf{U}^\top]^{jk} \left[\boldsymbol{\Lambda}_x^{-1}\boldsymbol{\Sigma}_w^{-1}\boldsymbol{\Lambda}_x^{-1}\right]_{km}[\mathbf{U}]^{mi}. \tag{66}$$

Each $1 \times n_\theta$ block of the $n_x \times n_x n_\theta$ block matrix $\mathbf{U}$ is

$$[\mathbf{U}]^{k,i} = \int_{\boldsymbol{\Omega}} [\boldsymbol{\mu}(r)]_k \left[\int_{\boldsymbol{\Omega}} \boldsymbol{\mu}\left(r'\right)\boldsymbol{\lambda}^\top\left(r - r'\right)\mathrm{d}r'\right]_{i:}\mathrm{d}r, \tag{67}$$

where $[\cdot]_k$ denotes the kth element of the vector and $[\cdot]_{i:}$ denotes the ith row. The matrices $\boldsymbol{\Xi}_0$ and $\boldsymbol{\Xi}_1$ in Eqs. (62)-(64) are calculated using the RTS smoother outputs: $\hat{\mathbf{x}}_t^b$, covariance, $\mathbf{P}_t^b$, and cross-covariance

matrix, $\mathbf{M}_t$ (Gibson and Ninness, 2005) (see Appendix D for details),

$$\boldsymbol{\Xi}_0 = \sum_{t=0}^{T-1} \left(\mathbf{M}_{t+1} + \hat{\mathbf{x}}_t^b \hat{\mathbf{x}}_{t+1}^{b\top} \right) \tag{68}$$

$$\boldsymbol{\Xi}_1 = \sum_{t=0}^{T-1} \left(\mathbf{P}_t^b + \hat{\mathbf{x}}_t^b \hat{\mathbf{x}}_t^{b\top} \right). \tag{69}$$

The algorithm has two steps: the E-step, which computes $\boldsymbol{\Xi}_0$ and $\boldsymbol{\Xi}_1$ based on the most recent parameter estimates using the RTS smoother, and the M-step, which updates the parameter estimates by computing the (analytic) maximum of $Q(\boldsymbol{\theta}, \boldsymbol{\theta}')$. The EM algorithm iterates between these two steps until the parameter estimates converge. A summary of the joint state parameter estimation is given in Table 4.

6.3. Stopping Rule

The stopping criterion is usually associated with either the change in the parameter estimates or the log-likelihood variation (McLachlan and Krishnan, 1997). The stopping rule adopted herein is

$$\left(\| \mathbf{A} \|_F^{(i)} - \| \mathbf{A} \|_F^{(i-1)} \right) < \epsilon, \tag{70}$$

where ϵ is a threshold value and $\| \mathbf{A} \|_F^{(i)}$ and $\| \mathbf{A} \|_F^{(i-1)}$ are the Frobenius norms of the successive estimates of $\mathbf{A}$ matrices which is defined by Meyer (2000)

$$\| \mathbf{A} \|_F = \sqrt{\sum_{i,j=1}^{n_x} | a_{i,j} |^2} = \sqrt{\text{tr}(\mathbf{A}^\top \mathbf{A})}. \tag{71}$$

6.4. Computational Complexity

All n_x^2 blocks of $\mathbf{U}$, $\boldsymbol{\Gamma}_1$ and $\boldsymbol{\Gamma}_2$ can be computed and stored prior to the commencement of the EM iterations (Step 1 in Table 4), which increases the speed of the M-step significantly compared to the implementation of Dewar et al. (2009). The complexity of each step of the algorithm is summarized in Table 5 where for ease of comparison the computational complexity of the method of Dewar et al. (2009) is also identified. It can be seen from the table that the state dimension is an important factor at each iteration of the estimation algorithm, emphasizing the advantage of the proposed algorithm compared to the method proposed in Dewar et al. (2009). This also suggests the use of an efficient algorithm to choose the minimum number of basis functions (states of the system) while achieving the desired details of the underlying field.

7. Results

To demonstrate the performance of the MRAIDE estimation framework, data was generated synthetically using Amari neural field model (Eqs. (3) and (5)), allowing a comparison between true and estimated

1. Initialization:

Choose a bounded random state vector sequence $\hat{\mathbf{X}}$, and set $\mathbf{P}_t$ and $\mathbf{M}_t$ large

compute

$$\mathbf{\Lambda}_x = \int_\Omega \boldsymbol{\mu}\left(r\right) \boldsymbol{\mu}^\top\left(r\right) \mathrm{d}r$$

compute all the n_x^2 blocks of block matrices $\mathbf{U}$, Γ_1 and Γ_2 using

$$[\mathbf{U}]^{k,i} = \int_{\boldsymbol{\Omega}} [\boldsymbol{\mu}(r)]_k \left[\int_{\boldsymbol{\Omega}} \boldsymbol{\mu}\left(r'\right) \boldsymbol{\lambda}^\top\left(r - r'\right) dr'\right]_{i:} dr$$

$$[\Gamma_1]^{j,i} = \sum_{k=1}^{n_x} \left[\boldsymbol{\Sigma}_w^{-1} \mathbf{\Lambda}_x^{-1}\right]_{jk} [\mathbf{U}]^{k,i}$$

$$[\Gamma_2]^{j,i} = \sum_{k,m=1}^{n_x} [\mathbf{U}^\top]^{jk} \left[\mathbf{\Lambda}_x^{-1} \boldsymbol{\Sigma}_w^{-1} \mathbf{\Lambda}_x^{-1}\right]_{km} [\mathbf{U}]^{mi}$$

2. Estimate the connectivity kernel parameters

$$\mathbf{\Xi}_0 = \sum_{t=0}^{T-1} \left(\mathbf{M}_{t+1} + \hat{\mathbf{x}}_t \hat{\mathbf{x}}_{t+1}^\top\right)$$

$$\mathbf{\Xi}_1 = \sum_{t=0}^{T-1} \left(\mathbf{P}_t + \hat{\mathbf{x}}_t \hat{\mathbf{x}}_t^\top\right)$$

$$\boldsymbol{v}_0 = \sum_{i,j=1}^{n_x} [\mathbf{\Xi}_0]_{i,j} [\Gamma_1]_{j,i}$$

$$\boldsymbol{v}_1 = \xi \sum_{i,j=1}^{n_x} [\mathbf{\Xi}_1]_{i,j} [\Gamma_1]^{j,i}$$

$$\mathbf{\Upsilon} = T_s \varsigma \sum_{i,j=1}^{n_x} [\mathbf{\Xi}_1]_{i,j} [\Gamma_2]^{j,i}$$

$$\boldsymbol{\theta} = \mathbf{\Upsilon}^{-\top} (\boldsymbol{v}_0 - \boldsymbol{v}_1)^\top$$

3. Form a state-space representation

compute all n_x^2 elements of $\mathbf{\Lambda}_\theta$ using

$$[\mathbf{\Lambda}_\theta]_{k,i} = [\mathbf{U}]^{k,i} \boldsymbol{\theta}$$

compute transition matrix $\mathbf{A}(\boldsymbol{\theta})$

$$\mathbf{A}(\boldsymbol{\theta}) = T_s \varsigma \mathbf{\Lambda}_x^{-1} \mathbf{\Lambda}_\theta + \xi \mathbf{I}$$

4. Compute the smoothed state $\hat{\mathbf{x}}_t^b$, the smoothed covariance, $\mathbf{P}_t^b$ and cross-covariance, $\mathbf{M}_t$ for $t \in \{0, \cdots, T\}$ using the RTS smoother given in Table 3.

5. Convergence: return to Step 2 unless

$$\left(\| \mathbf{A} \|_F^{(i)} - \| \mathbf{A} \|_F^{(i-1)}\right) < \epsilon$$

Table 4: **State and parameter estimation for the MRAIDE model.** This table shows the steps for the joint state and parameter estimation using the EM based algorithm.

Variable	Equation	Order	Order obtained from Dewar et al. (2009)
Ξ_0, Ξ_1	Eqs. (68) and (69)	$O(Tn_x^2)$	—
$\boldsymbol{v}_0, \boldsymbol{v}_1$	Eqs. (62) and (63)	$O(n_x^2 n_\theta)$	$O(n_x^2 n_\theta) + O(n_x^3)$
$\boldsymbol{\Upsilon}$	Eq. (64)	$O(n_x^2 n_\theta^2)$	$O(n_x^4 n_\theta^2)$
$\mathbf{A}(\boldsymbol{\theta})$	Eq. (56)	$O(n_x^3)$	—
$\boldsymbol{\Lambda}_\theta$	Eq. (C.13)	$O(n_x^2 n_\theta)$	—
$\hat{\boldsymbol{\theta}}$	Eq. (61)	$O(n_\theta^3)$	—

Table 5: **The Computational complexity of the estimation algorithm**. A comparison between the algorithm proposed herein and that of Dewar et al. (2009) is also provided. "—" indicate equal complexities for both algorithms.

parameters. In doing so, the membrane time constant was set to $\tau = 10$ ms (David and Friston, 2003), and the sampling time was chosen ten times faster (Stephan et al., 2008), i.e. $T_s = 1$ ms, ensuring the discrepancy between the continuous neural field model and its discrete approximation is small. The connectivity kernel used for data generation composed of two B-spline scaling functions at $j = 0$ and $j = 1$ forming a compact support Mexican hat shape

$$w(r - r') = \theta_1 \times N_{4;1,2}(r - r') + \theta_2 \times N_{4;0,2}(r - r'), \tag{72}$$

with $\theta_1 = 200$ and $\theta_2 = -100$.

Two different experiments were designed to test the framework. In the first one, 200 realizations of 1 second of data were used, where the field disturbance, e_t, and the observation noise, $\boldsymbol{\epsilon}_t$, were regenerated each run. This way the net shape of the connectivity kernel could be formed by averaging over the parameters estimates from each realization. In the second experiment, a single data set was generated and the field estimations are compared for five state-space models, each accounting up to a certain level of approximation, i.e. $j = 0$ to $j = 4$. We show it is possible to determine the level of approximation to accurately estimate the underlying field prior to the EM-based algorithm. All parameters for the model are given in Table 6.

For each of the experiments, the estimation was applied to the final 900 ms, allowing the forward model's dynamics to stabilize from the initial conditions. The initial state, parameters and the support of the connectivity kernel were unknown to the estimator. The observation noise was set to $\boldsymbol{\Sigma}_\epsilon = 0.1 \times \mathbf{I}_{n_y}$ and the disturbance covariance function was set to $\eta(r - r') = \sigma_d \times \varphi_{4;3,2}(r - r')$ (B-spline, $j = 3$) with $\sigma_d = 0.53$. The firing rate slope, ς was set to $0.56\ mV^{-1}$ (Wendling, 2005). The observation kernel was modeled using a B-spline function, with a width of 0.08 mm at half the maximum amplitude where the distance between

Symbol	Quantity	Value	Units
	Domain and indices		
Δ	Spatial discretisation step	0.01	mm
T_s	Time step	0.001	s
T	Number of time steps used in estimation	900	n.a.
	Model		
τ	Membrane time constant	0.01	s^{-1}
ς	Slope of activation function	0.56 (Wendling, 2005)	$mV^{-1}.spike.s^{-1}$
$\boldsymbol{\theta}$	Vector of connectivity kernel parameters (data generation)	$\begin{bmatrix} 200 & -100 \end{bmatrix}^{\top}$	$mV.spike^{-1}$
n_y	Number of sensors	161	n.a.
Δ_y	Distance between adjacent sensors	0.05	mm
σ_m	Observation kernel full width at half maximum	0.08	mm
Σ_ε	Observation noise covariance matrix	$0.1 \times I_{n_y}$	mm^2
σ_η	Disturbance covariance function full width at half maximum	0.18	mm
σ_d	Disturbance variance	0.53	mV
	Reduced Model		
n_x	Number of field basis functions		n.a.
	Experiment I, $j = 3$	131	
	Experiment II, $j = 0, 1, 2, 3, 4$	17, 33, 65, 131, 263	
	Estimation		
n_θ	Number of connectivity kernel basis functions for estimation	25	n.a.

Table 6: **Parameters.** Parameters for the neural field model and the reduced model and the estimation procedure.

adjacent sensors was 0.05 mm, resulting into $n_y = 161$ observations. The choice of B-spline function to model the observation kernel is justified by the experiment performed in Freestone et al. (2011). The spacing and bandwidth of the sensors allowed the full spatial bandwidth of the field to be observed. With this number of observations, the estimation problem was well-posed, i.e. the number of states, $n_x < n_y$ at each level of the decomposition. This is because the scaling functions are orthogonal to the wavelets at the same and higher levels of decomposition, and the wavelets are orthogonal to each other across different scales (see Eqs. (49) and (50)).

7.1. Experiment I

The spatial frequency response of the observed field was used to specify the level of decomposition required to represent the field using the wavelets. This is shown in Fig. 6 where the graph obtained by averaging the spectral power of the spatial frequency (over time) from the observations (Scerri et al., 2009). The spatial bandwidths of wavelets were calculated analytically using Eq. (51). The result suggested that wavelets up to level $j = 3$ (with the bandwidth $\approx [5, 8]$ cycles/mm) can represent the significant spatial characteristics of the field which yields $n_x = 131$ states, corresponding to 9 scaling functions for $j = 0$ and 8, 16, 32 and 66 wavelet functions, respectively for $j = 0$, $j = 1$, $j = 2$ and $j = 3$.

The EM algorithm was allowed to run until a maximum number of 20 iterations was reached, though typically the change in the transition matrix Frobenius norm dropped below 10^{-6} after less than 10 iterations. The rate of convergence of the EM based algorithm is depicted in Fig. 7.
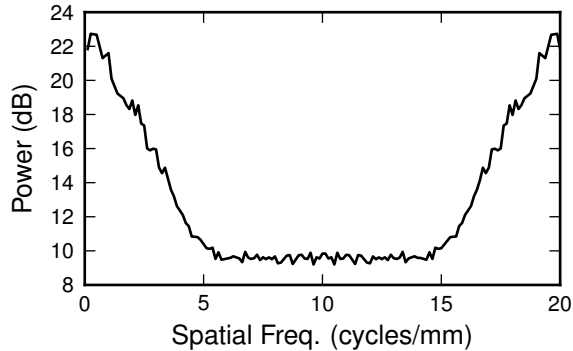
Figure 6: **Spatial frequency analysis**. The average (over time) power in dB of the spatial frequency of the observations.
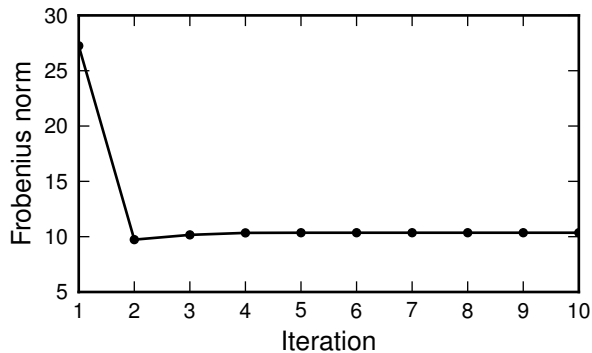

Figure 7: **Convergence of the EM based algorithm**. The mean of the $\mathbf{A}(\boldsymbol{\theta})$ Frobenius norm over 200 realizations vs iterations of the EM based algorithm. The change in the stopping criterion falls below 10^{-6} after less than 10 iterations.

The actual connectivity kernel and the decomposition is plotted in Fig. 8(A). No assumptions where made about the shape of the kernel. The reconstructed kernel is in good accordance with the actual kernel, where the actual kernel lies inside the confidence interval. The large standard deviation is due to the high number of parameters needed to be estimated. The small error in the estimate is attributed to the MRA of the system used to form the estimator. Figs. 8(B) and (C) illustrate the kernel scaling and wavelet functions, respectively. A total number of 25 basis functions were used to reconstruct the kernel, comprising 13 and 12 scaling and wavelet functions at $j = 1$, respectively.

One snap shot of the field reconstruction, at different levels of approximation is given in Fig. 9, showing small discrepancy between the actual and estimated field at $j = 3$. From this figure, it can be observed that the overall trend of the underlying field can be also captured at the coarser levels.

7.2. Experiment II

To study the performance of different models with various MRA capabilities, a single realization was obtained using Eqs. (3) and (5) under the same circumstances of the previous experiment. The generated
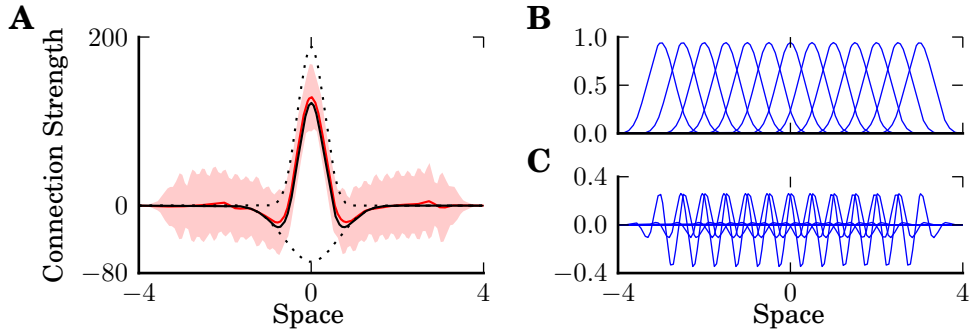
24

Figure 8: **Connectivity kernel estimate**. (**A**) The actual kernel and its components are shown by solid and dashed black lines, respectively. The mean kernel estimates over 200 realizations and confidence interval are shown by the red line and red shaded region (± 2 std.). (**B**,**C**) Kernel scaling and wavelets functions, respectively, at $j = 1$ used in the estimation algorithm.
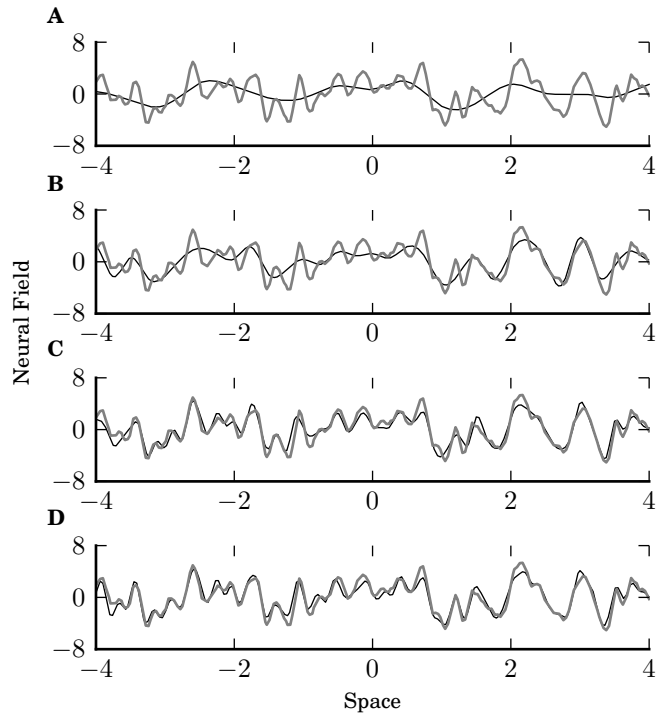

Figure 9: **Examples of the actual and estimated neural fields for experiment I**. Actual (grey) and estimated (black) neural fields at a single time instant for different spatial resolutions. (**A**) $j = 0$, $n_x = 17$. (**B**) $j = 1$, $n_x = 33$. (**C**) $j = 2$, $n_x = 65$. (**D**) $j = 3$, $n_x = 131$.

data set was then employed to estimate the underlying field and the connectivity kernel parameters using different state-space models, $j = 0$ up to $j = 4$. The mean over time of the root mean square error (MRMSE) over space of the field estimation for different models is shown in Fig. 10. By increasing the value of j, error in the estimation decreases and converges to a steady value of 0.71, confirming that the model with $j = 3$
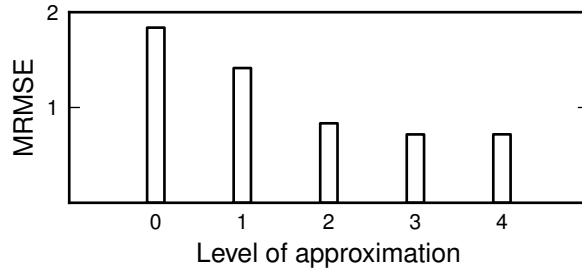
Figure 10: **Error in the field reconstruction for experiment II**. The mean (over time) of average (over space) RMSE of the estimated field of state-space models with different MRA capabilities. $j = 0$, $RMSE = 1.83$; $j = 1$, $RMSE = 1.41$; $j = 2$, $RMSE = 0.83$; $j = 3$, $RMSE = 0.71$; $j = 4$, $RMSE = 0.71$

(used in the first experiment) is in fact adequate to capture the dynamics of the underlying field. The spatiotemporal characteristics of the reconstructed fields (A-E) along with the actual field (F) are shown in Fig. 11. The experiment was performed several times and the results were consistent over each run.

8. Discussion

In this paper, we have presented a novel model-based framework for estimating cortical dynamics from electrophysiological measurements. The novel and key developments of the paper include the multi-resolution finite element model representation of the neural field, and the estimator for an intracortical connectivity kernel with an arbitrary shape. By decomposing both the spatial field and the connectivity kernel using a wavelet decomposition, it becomes possible to represent small scale details in the field at the same time as large scale details. This work is significant, as the ability to create patient-specific neural field models has the potential to contribute to our understanding and improve treatment of diseases resulting from abnormal neurodynamics. Other groups have previously highlighted the importance of a multi-resolution approach in neural field modeling, where the dynamics and connectivity structure differs at different spatial scales (Jirsa et al., 2002; Jirsa, 2004; Breakspear and Stam, 2005; Qubbaj and Jirsa, 2009; Schultze-Kraft et al., 2011).

8.1. State-Space Representation

In order to apply estimation algorithms to neural field models a finite-dimensional state-space model must be formed. The problem of representing a continuous field with a finite number of states is solved by the basis function decomposition. An alternative to using the B-spline wavelets are Gaussian basis functions, which were adopted by Freestone et al. (2011) to decompose the field and connectivity kernel. One clear advantage of B-spline scaling and wavelet functions is the MRA capability, allowing the simultaneous representation of the neural field at different spatial scales. B-spline scaling and wavelet functions are compactly supported, eliminating numerical errors caused by the Gaussian basis function. The partition of unity (ability to represent a zero spatial frequency field) of these basis functions enable the reconstruction of the neural
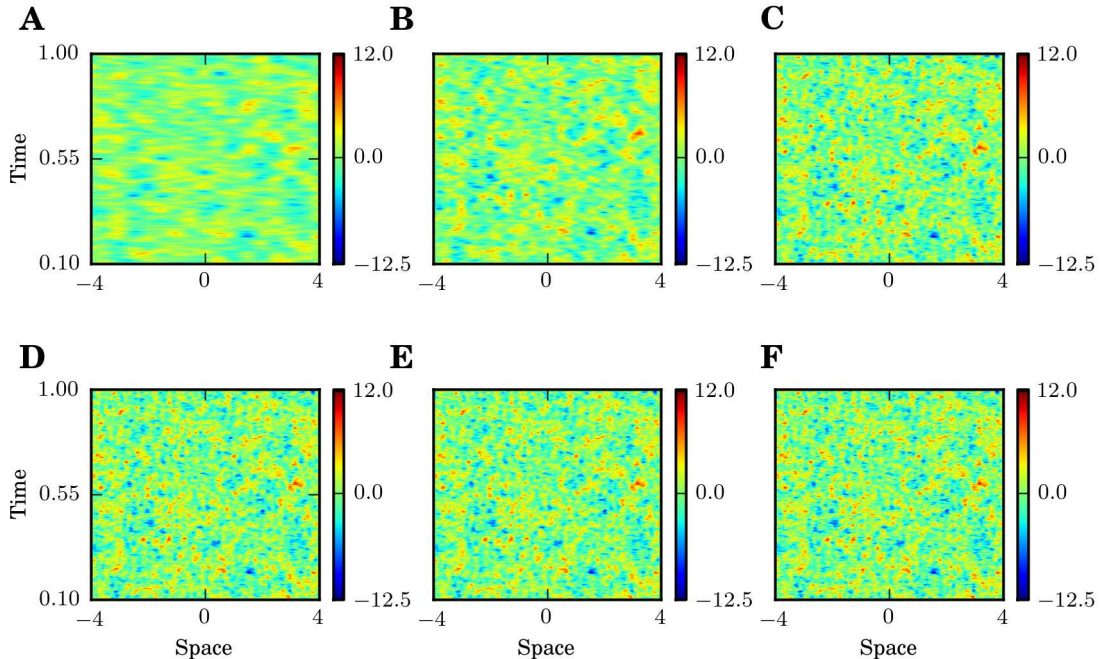
Figure 11: **Actual and estimated spatiotemporal neural field using state-space models with different MRA capabilities**. (**A**) The reconstructed field of the reduced model using wavelet upto $j = 0$ with $n_x = 17$. (**B**) The reconstructed field using wavelets upto $j = 1$ with $n_x = 33$. (**C**) The reconstructed field using wavelets upto $j = 2$ with $n_x = 65$. (**D**) The reconstructed field using wavelets upto $j = 3$ with $n_x = 131$. (**E**) The reconstructed field using wavelets upto $j = 4$ with $n_x = 263$. (**F**) The actual field.

field when it saturates. This may be beneficial to model neurological disorders such as epilepsy. Attaining fine-scale approximation of the neural field using Gaussian basis functions results in an ill-posed problem as the number of states is increased significantly. However, this is not the case when using B-spline scaling and wavelet functions due to their semi-orthogonality property.

Freestone et al. (2011) showed experimentally that Gaussian functions are a suitable choice for modeling the observation kernel when intracranial measurements are considered. B-splines converge to Gaussian functions as the order of the B-spline, m, tends to infinity. The cubic B-spline is already a very close approximation to the Gaussian function. Therefore, these should also be suitable for modeling the observation kernel while having the advantage of being compactly supported.

8.2. Estimation Algorithm

The implementation of the M-step in this work significantly simplifies the computational complexity of the EM based algorithm proposed by Dewar et al. (2009), as shown in Table 5 ($O(n_x^4 n_\theta^2)$ to $O(n_x^2 n_\theta^2)$). This is crucial for high-dimensional systems, and more specifically, for implementing the multi-resolution framework

27

introduced herein. Furthermore, choosing isotropic scaling and wavelet functions allows switching the order of operations with the spatial convolution and inner product in Eq. (26). This simplifies the computational complexity of the estimation algorithm significantly.

For a larger space, or in a case where the connectivity kernel decomposition comprises of many scale levels, the number of parameters to be estimated increases significantly. One solution to this problem could be expectation-conditional maximization (ECM) (Meng and Rubin, 1993; Meng, 1994), which replaces the M-step by a series of computationally simplified conditional maximization (CM) steps. The ECM is a class of generalized EM (GEM) algorithms in which the $\mathcal{Q}$-function is increased rather than being maximized (Fessler and Hero, 1994). For systems with a high number of states, the ensemble Kalman smoother (EnKS) (Evensen, 2003, 2009b,a) provides an alternative to the RTS smoother used in this work. The EnKS is a sequential Monte Carlo (MC) method where the state covariance matrix is approximated by a large stochastic ensemble of model states, thus alleviating computational load due to the storage and forward integration of the state covariance matrix. However, combining the EnKS with the EM algorithm suffers from the loss of monotonicity property — increase in the likelihood function— as MC error will be introduced at the E-step.

Typically, a sequence of likelihood values will converge to a stationary point, i.e. a global (local) maximum or a saddle point. If the sequence of the EM is trapped in a local maximum or a saddle point, a small random perturbation diverges the algorithm from such stationary values (McLachlan and Krishnan, 1997). In general, convergence of the EM algorithm to any stationary point depends on the initialization. In the examples presented in this paper, a bounded sequence of the state vectors was used to initialize the algorithm, which gave a satisfactory convergence and good state and parameter estimations.

8.3. Modification to Higher Dimensions

The reconstruction of the neural field in higher dimensions requires $2^n - 1$ wavelets, where n is the dimension of the space. For example a two-dimensional MRA can be implemented using 2-D scaling and wavelet functions built up using the tensor-product approach (Meyer, 1992). In this case three wavelet functions are required to extract fine features of the field at vertical, horizontal and diagonal orientations. Theoretically, the formulations provided in this paper can be easily extended to two dimensions. However, the increase in computational load of the estimation algorithm will be significant. This is not only due to the increase in number of wavelets, but also the switch of convolution and inner product does not hold, as the resulting higher-dimensional B-spline wavelets are not isotropic. A possible solution could be to use isotropic polyharmonic B-splines scaling functions and wavelets as an alternative (Van De Ville et al., 2005a).

It is important to investigate methods for choosing basis functions to balance the complexity with the reconstruction capability of the model. It is also possible to apply other methods than dyadic subsampling schemes in multiple dimensions, which leads to less wavelet subbands. For example, in 2D, the quincunx scheme is well-known to have better isotropic properties. Unfortunately, in 3D, it is not possible to have a

subsampling operator (acting on a Cartesian sampling lattice) that is performing subsampling at equal rate in each dimension at each iteration (Van De Ville et al., 2005b).

8.4. Extensions to the Framework

Clearly there is, and will always be, a discrepancy between the dynamics of the model and the experimental observation. Nevertheless, the model-based framework proposed in this paper may enable meaningful state tracking and connectivity estimation. The MRAIDE approach is not limited to neural fields; the framework can be applied to modeling other multi-resolution spatiotemporal dynamical systems such as weather systems, ecological systems, and others (Wikle, 2004; Xu et al., 2005). The key development is the multi-resolution decomposition forming the state-space model. While the decomposition holds for more sophisticated models, efficiently performing nonlinear smoothing in the large state-space remains a challenge.

Additional extension for future work include extending the model to capture a second order synaptic response kernel. The second-order postsynaptic kernel proposed in Van Rotterdam et al. (1982) leads to dynamics that are more reminiscent to real postsynaptic potentials (see Spiegler et al. (2010, 2011) for example). Another extension of this framework is the inclusion of finite distant-dependent action potential propagation velocity into the parametric neural field equations. In small networks the resulting delays via propagation can be neglected. However, delays must be accounted for in large scale networks. For instance, Ghosh et al. (2008) demonstrated that inclusion of time delays arising from propagation along connecting fibres are essential for the understanding of the fluctuations in resting state networks. Finally, future work should be directed towards applying and validating the framework on real data.

9. Acknowledgements

Appendix A. Inner product of two B-spline scaling functions

In this section, an analytic formula is derived for the inner product of two B-spline scaling functions. Consider two B-spline scaling functions of order m and m'. The support of $N_m(r)$ is $[0, m]$ and is symmetric with respect to $r = \frac{m}{2}$, i.e.

$$N_m\left(\frac{m}{2} + r\right) = N_m\left(\frac{m}{2} - r\right). \tag{A.1}$$

A direct consequence of Eq. (A.1) is

$$N_m\left(r\right) = N_m\left(m - r\right). \tag{A.2}$$

Therefore

$$
\begin{aligned}
\int_{-\infty}^{+\infty} N_m\left(r - l_1\right) N_{m'}\left(r - l_2\right) dr &= \int_{-\infty}^{+\infty} N_m\left(m - r + l_1\right) N_{m'}\left(r - l_2\right) dr \\
&= \int_{-\infty}^{+\infty} N_m\left(m + l_1 - l_2 - u\right) N_{m'}\left(u\right) du \\
&= \left(N_m * N_{m'}\right)\left(m + l_1 - l_2\right) \\
&= N_{m+m'}\left(m + l_1 - l_2\right) \\
&= N_{m+m'}\left(m' + l_2 - l_1\right).
\end{aligned} \tag{A.3}
$$

Appendix B. Fourier transform of B-spline wavelets

To calculate the frequency response of the B-spline wavelet the Fourier transform of $N_1(r)$ is first calculated,

$$
\begin{aligned}
\mathcal{F}(N_1(r)) &= \int_{-\infty}^{+\infty} N_1(r)\exp(-2\pi i\nu r)dr \\
&= \int_0^1 \exp(-2\pi i\nu r)dr \\
&= \frac{1 - \exp(-2\pi i\nu)}{2\pi i\nu},
\end{aligned} \tag{B.1}
$$

The convolution theorem states that the Fourier transform of the convolution of functions is equal to the product of their Fourier transforms. Therefore, taking Fourier transform of Eq. (39) and substituting Eq. (B.1) for $\mathcal{F}(N_1(r))$ yields

$$\mathcal{F}(N_m(r)) = \left(\frac{1 - \exp(-2\pi i\nu)}{2\pi i\nu}\right)^m. \tag{B.2}$$

The Fourier transform of the mth order B-spline allows the calculation of the spatial frequency response of the wavelets using Eq. (44) giving

$$
\begin{aligned}
\mathcal{F}(\varphi_m\left(r\right)) &= \sum_{n=0}^{3m-2} q_n \mathcal{F}\left(N_m\left(2r - n\right)\right) \\
&= \frac{1}{2}\left(\frac{1 - \exp(-\pi i\nu)}{\pi i\nu}\right)^m \sum_{n=0}^{3m-2} q_n\exp(-\pi in\nu).
\end{aligned} \tag{B.3}
$$

Appendix C. Construction and maximization of the lower bound for the state-space representation of the MRAIDE

For the linear state-space model given by Eqs. (55) and (32). The joint probability distribution $p(\mathbf{X}, \mathbf{Y}; \boldsymbol{\theta})$ can be written as

$$p(\mathbf{X}, \mathbf{Y}; \boldsymbol{\theta}) = \prod_{t=0}^{T-1} p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1}) p(\mathbf{x}_{t+1}|\mathbf{x}_t; \boldsymbol{\theta}) p(\mathbf{x}_0). \tag{C.1}$$

The $\mathcal{Q}$-function, or lower-bound on the likelihood, can be expressed in terms of the joint distribution components

$$\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}') = \mathbf{E}_{\boldsymbol{\theta}'} \left[2 \ln p(\mathbf{X}, \mathbf{Y}; \boldsymbol{\theta}) \right]$$

$$= \mathbf{E}_{\boldsymbol{\theta}'} \left[\sum_{t=0}^{T-1} 2 \ln p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1}) + \sum_{t=0}^{T-1} 2 \ln p(\mathbf{x}_{t+1}|\mathbf{x}_t; \boldsymbol{\theta}) + \sum_{t=0}^{T-1} 2 \ln p(\mathbf{x}_0) \right]. \tag{C.2}$$

It should be noted that neither $p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1})$ nor $p(\mathbf{x}_0)$ are functions of the parameter set and therefore the $\mathcal{Q}$-function can be rewritten as

$$\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}') = \mathbf{E}_{\Theta'} \left[\sum_{t=0}^{T-1} 2 \ln p(\mathbf{x}_{t+1}|\mathbf{x}_t; \boldsymbol{\theta}) \right] + \text{constant}, \tag{C.3}$$

where the constant term can be ignored as it is independent of the parameters. Under the condition that the state distribution is Gaussian — ignoring the normalizing term, $1/(2\pi)^{\frac{n_x}{2}} \mid \boldsymbol{\Sigma}_w \mid^{-\frac{1}{2}}$ — the conditional distribution $p(\mathbf{x}_{t+1}|\mathbf{x}_t; \boldsymbol{\theta})$ can be written as

$$p(\mathbf{x}_{t+1}|\mathbf{x}_t; \boldsymbol{\theta}) = \exp\left(-\frac{1}{2} \left(\mathbf{x}_{t+1} - \mathbf{A}(\boldsymbol{\theta}) \mathbf{x}_t \right)^\top \boldsymbol{\Sigma}_w^{-1} \left(\mathbf{x}_{t+1} - \mathbf{A}(\boldsymbol{\theta}) \mathbf{x}_t \right) \right). \tag{C.4}$$

The logarithm of the above distribution, once expanded, is given by

$$2 \ln p(\mathbf{x}_{t+1}, \mathbf{y}_{t+1}; \boldsymbol{\theta}) = - \mathbf{x}_{t+1}^\top \boldsymbol{\Sigma}_w^{-1} \mathbf{x}_{t+1} + 2 \mathbf{x}_{t+1}^\top \boldsymbol{\Sigma}_w^{-1} \mathbf{A}(\boldsymbol{\theta}) \mathbf{x}_t$$

$$- \mathbf{x}_t^\top \mathbf{A}^\top(\boldsymbol{\theta}) \boldsymbol{\Sigma}_w^{-1} \mathbf{A}(\boldsymbol{\theta}) \mathbf{x}_t. \tag{C.5}$$

The factor of 2 in front of the logarithm is included to simplify the derivation in the later steps. Substituting $\mathbf{A}(\boldsymbol{\theta})$ from Eq. (56) into Eq. (C.5) gives

$$2 \ln p(\mathbf{x}_{t+1}, \mathbf{y}_{t+1}; \boldsymbol{\theta}) = \beta + 2 T_s \varsigma \mathbf{x}_{t+1}^\top \boldsymbol{\Sigma}_w^{-1} \boldsymbol{\Lambda}_x^{-1} \boldsymbol{\Lambda}_\theta \mathbf{x}_t$$

$$- T_s^2 \varsigma^2 \mathbf{x}_t^\top \boldsymbol{\Lambda}_\theta^\top \boldsymbol{\Lambda}_x^{-1} \boldsymbol{\Sigma}_w^{-1} \boldsymbol{\Lambda}_x^{-1} \boldsymbol{\Lambda}_\theta \mathbf{x}_t - 2\xi \varsigma T_s \mathbf{x}_t^\top \boldsymbol{\Sigma}_w^{-1} \boldsymbol{\Lambda}_x^{-1} \boldsymbol{\Lambda}_\theta \mathbf{x}_t,$$

$$\tag{C.6}$$

where

$$\beta = -\mathbf{x}_{t+1}^\top \mathbf{\Sigma}_w^{-1} \mathbf{x}_{t+1} + 2\xi \mathbf{x}_{t+1}^\top \mathbf{\Sigma}_w^{-1} \mathbf{x}_t - \xi^2 \mathbf{x}_t^\top \mathbf{\Sigma}_w^{-1} \mathbf{x}_t, \tag{C.7}$$

is constant with respect to the parameter $\boldsymbol{\theta}$, and will disappear subject to differentiation with respect to $\boldsymbol{\theta}$. Taking the trace and rearranging, using the invariant cyclic permutations property of the trace, this distribution can be written as

$$\begin{aligned}
2 \ln p(\mathbf{x}_{t+1}, \mathbf{y}_{t+1}; \boldsymbol{\theta}) = {} & \beta + 2T_s\varsigma \mathrm{tr} \left\{ \mathbf{x}_t \mathbf{x}_{t+1}^\top \mathbf{\Sigma}_w^{-1} \mathbf{\Lambda}_x^{-1} \mathbf{\Lambda}_\theta \right\} \\
& - T_s^2 \varsigma^2 \mathrm{tr} \left\{ \mathbf{x}_t \mathbf{x}_t^\top \mathbf{\Lambda}_\theta^\top \mathbf{\Lambda}_x^{-1} \mathbf{\Sigma}_w^{-1} \mathbf{\Lambda}_x^{-1} \mathbf{\Lambda}_\theta \right\} \\
& - 2\xi\varsigma T_s \mathrm{tr} \left\{ \mathbf{x}_t \mathbf{x}_t^\top \mathbf{\Sigma}_w^{-1} \mathbf{\Lambda}_x^{-1} \mathbf{\Lambda}_\theta \right\}.
\end{aligned} \tag{C.8}$$

Rearranging and taking the expectation of the log-likelihood function over all time instants gives the required lower-bound, which is to be maximized for the optimal parameter estimates. Note that the expectation distributes over the trace sum. Therefore,

$$\begin{aligned}
\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}') = {} & \beta + 2T_s\varsigma \mathrm{tr} \left\{ \mathbf{\Xi}_0 \mathbf{\Sigma}_w^{-1} \mathbf{\Lambda}_x^{-1} \mathbf{\Lambda}_\theta \right\} - 2\xi\varsigma T_s \mathrm{tr} \left\{ \mathbf{\Xi}_1 \mathbf{\Sigma}_w^{-1} \mathbf{\Lambda}_x^{-1} \mathbf{\Lambda}_\theta \right\} \\
& - T_s^2 \varsigma^2 \mathrm{tr} \left\{ \mathbf{\Xi}_1 \mathbf{\Lambda}_\theta^\top \mathbf{\Lambda}_x^{-1} \mathbf{\Sigma}_w^{-1} \mathbf{\Lambda}_x^{-1} \mathbf{\Lambda}_\theta \right\},
\end{aligned} \tag{C.9}$$

where

$$\mathbf{\Xi}_0 = \mathbf{E}_{\Theta'} \left[\sum_{t=0}^{T-1} \mathbf{x}_t \mathbf{x}_{t+1}^\top \right] \tag{C.10}$$

$$\mathbf{\Xi}_1 = \mathbf{E}_{\Theta'} \left[\sum_{t=0}^{T-1} \mathbf{x}_t \mathbf{x}_t^\top \right]. \tag{C.11}$$

The matrices $\mathbf{\Xi}_0$ and $\mathbf{\Xi}_1$ are calculated using the RTS smoother outputs (see Appendix D for more details).

The goal of the remainder of the derivation is to isolate $\boldsymbol{\theta}$ from the terms forming $\mathcal{Q}$. The three traces that form Eq. (C.9) are dealt with in turn. To begin, the first trace is written in terms of element-wise summations

$$\begin{aligned}
\mathrm{tr} \left\{ \mathbf{\Xi}_0 \mathbf{\Sigma}_w^{-1} \mathbf{\Lambda}_x^{-1} \mathbf{\Lambda}_\theta \right\} & = \sum_{i=1}^{n_x} \left[\mathbf{\Xi}_0 \mathbf{\Sigma}_w^{-1} \mathbf{\Lambda}_x^{-1} \mathbf{\Lambda}_\theta \right]_{ii} \\
& = \sum_{i,j=1}^{n_x} \left[\mathbf{\Xi}_0 \right]_{ij} \left[\mathbf{\Sigma}_w^{-1} \mathbf{\Lambda}_x^{-1} \mathbf{\Lambda}_\theta \right]_{ji} \\
& = \sum_{i,j=1}^{n_x} \left[\mathbf{\Xi}_0 \right]_{ij} \sum_{k=1}^{n_x} \left[\mathbf{\Sigma}_w^{-1} \mathbf{\Lambda}_x^{-1} \right]_{jk} \left[\mathbf{\Lambda}_\theta \right]_{ki},
\end{aligned} \tag{C.12}$$

Each element of $\mathbf{\Lambda}_\theta$ can be calculated using

$$[\mathbf{\Lambda}_\theta]_{k,i} = [\mathbf{U}]^{k,i}\,\boldsymbol{\theta}, \tag{C.13}$$

where $[\mathbf{U}]^{k,i}$ can be computed using Eq. (67). By substituting for $[\mathbf{\Lambda}_\theta]_{k,i}$ from Eq. (C.13) the trace given in Eq. (C.12) can be written as

$$
\begin{aligned}
\operatorname{tr}\left\{\mathbf{\Xi}_0\mathbf{\Sigma}_w^{-1}\mathbf{\Lambda}_x^{-1}\mathbf{\Lambda}_\theta\right\} &= \sum_{i,j=1}^{n_x}[\mathbf{\Xi}_0]_{ij}\sum_{k=1}^{n_x}\left[\mathbf{\Sigma}_w^{-1}\mathbf{\Lambda}_x^{-1}\right]_{jk}[\mathbf{U}]^{k,i}\,\boldsymbol{\theta} \\
&= \sum_{i,j=1}^{n_x}[\mathbf{\Xi}_0]_{ij}\,[\Gamma_1]^{j,i}\,\boldsymbol{\theta},
\end{aligned} \tag{C.14}
$$

where

$$[\Gamma_1]^{j,i} = \sum_{k=1}^{n_x}\left[\mathbf{\Sigma}_w^{-1}\mathbf{\Lambda}_x^{-1}\right]_{jk}[\mathbf{U}]^{k,i}. \tag{C.15}$$

The second trace of Eq. (C.9) is dealt with in a similar manner, giving

$$\operatorname{tr}\left\{\mathbf{\Xi}_1\mathbf{\Sigma}_w^{-1}\mathbf{\Lambda}_x^{-1}\mathbf{\Lambda}_\theta\right\} = \sum_{i,j=1}^{n_x}[\mathbf{\Xi}_1]_{ij}\,[\Gamma_1]^{j,i}\,\boldsymbol{\theta}. \tag{C.16}$$

Likewise, the third trace of Eq. (C.9) is broken down into element-wise summations

$$
\begin{aligned}
\operatorname{tr}\left\{\mathbf{\Xi}_1\mathbf{\Lambda}_\theta^\top\mathbf{\Lambda}_x^{-1}\mathbf{\Sigma}_w^{-1}\mathbf{\Lambda}_x^{-1}\mathbf{\Lambda}_\theta\right\} &= \sum_{i,j,k,m=1}^{n_x}[\mathbf{\Xi}_1]_{i,j}\,[\mathbf{\Lambda}_\theta^\top]_{jk}\left[\mathbf{\Lambda}_x^{-1}\mathbf{\Sigma}_w^{-1}\mathbf{\Lambda}_x^{-1}\right]_{km}[\mathbf{\Lambda}_\theta]_{mi} \\
&= \boldsymbol{\theta}^\top\sum_{i,j=1}^{n_x}[\mathbf{\Xi}_1]_{i,j}\,\sum_{k,m=1}^{n_x}[\mathbf{U}^\top]^{jk}\left[\mathbf{\Lambda}_x^{-1}\mathbf{\Sigma}_w^{-1}\mathbf{\Lambda}_x^{-1}\right]_{km}[\mathbf{U}]^{mi}\,\boldsymbol{\theta} \\
&= \boldsymbol{\theta}^\top\sum_{i,j=1}^{n_x}[\mathbf{\Xi}_1]_{i,j}\,[\Gamma_2]^{j,i}\,\boldsymbol{\theta},
\end{aligned} \tag{C.17}
$$

where

$$[\Gamma_2]^{j,i} = \sum_{k,m=1}^{n_x}[\mathbf{U}^\top]^{jk}\left[\mathbf{\Lambda}_x^{-1}\mathbf{\Sigma}_w^{-1}\mathbf{\Lambda}_x^{-1}\right]_{km}[\mathbf{U}]^{mi}. \tag{C.18}$$

Therefore the $\mathcal{Q}$-function can be rewritten as

$$\mathcal{Q}\left(\boldsymbol{\theta},\boldsymbol{\theta}'\right) = \beta + 2T_s\varsigma\left(\boldsymbol{v}_0 - \boldsymbol{v}_1\right)\boldsymbol{\theta} - T_s\varsigma\boldsymbol{\theta}^\top\mathbf{\Upsilon}\boldsymbol{\theta}, \tag{C.19}$$

where $\boldsymbol{v}_0$, $\boldsymbol{v}_1$ and $\mathbf{\Upsilon}$ are defined in Eqs. (62)-(64). By differentiating the $\mathcal{Q}$-function with respect to $\boldsymbol{\theta}$ we

have

$$\frac{\partial \mathcal{Q}}{\partial \boldsymbol{\theta}} = 2T_s \varsigma (\boldsymbol{v}_0 - \boldsymbol{v}_1)^\top - T_s \varsigma (\boldsymbol{\Upsilon}^\top + \boldsymbol{\Upsilon}) \boldsymbol{\theta}$$

$$= 2T_s \varsigma (\boldsymbol{v}_0 - \boldsymbol{v}_1)^\top - 2T_s \varsigma \boldsymbol{\Upsilon}^\top \boldsymbol{\theta}. \tag{C.20}$$

Equating Eq. (C.20) to **0** vector yields

$$\boldsymbol{\theta} = \boldsymbol{\Upsilon}^{-\top} (\boldsymbol{v}_0 - \boldsymbol{v}_1)^\top. \tag{C.21}$$

The second derivative of the $\mathcal{Q}$-function is given by

$$\frac{\partial^2 \mathcal{Q}}{\partial \boldsymbol{\theta}^2} = -2T_s \varsigma \boldsymbol{\Upsilon}. \tag{C.22}$$

The matrix $\boldsymbol{\Upsilon}$ is positive definite if $n_x^2 \times n_\theta$ matrix vec($\mathbf{U}$) is of rank n_θ ((Dewar et al., 2009), Lemma 3), where vec($\cdot$) denotes the vectorization operator. Under this condition $\boldsymbol{\Upsilon}$ is invertible and the second derivative is negative definite representing a maximum of the $\mathcal{Q}$-function.

Appendix D. Calculation of $\boldsymbol{\Xi}_0$ and $\boldsymbol{\Xi}_1$

The expressions for $\boldsymbol{\Xi}_0$ and $\boldsymbol{\Xi}_1$ in terms of smoother outputs can be found in Shumway and Stoffer (2000), here the derivation is presented for completeness. We start with the expression for $\boldsymbol{\Xi}_0$ as defined in Eq. (C.10)

$$\boldsymbol{\Xi}_0 = \mathbf{E}_{\theta'} \left[\sum_{t=0}^{T-1} \mathbf{x}_t \mathbf{x}_{t+1}^\top \right]. \tag{D.1}$$

First we compute the cross-covariance matrix giving

$$\mathbf{M}_{t+1} = \mathbf{E}_{\theta'} \left[(\mathbf{x}_t - \hat{\mathbf{x}}_t^b)(\mathbf{x}_{t+1} - \hat{\mathbf{x}}_{t+1}^b)^\top \right]$$

$$= \mathbf{E}_{\theta'} \left[\mathbf{x}_t \mathbf{x}_{t+1}^\top \right] - \mathbf{E}_{\theta'} \left[\mathbf{x}_t \hat{\mathbf{x}}_{t+1}^{b\top} \right] - \mathbf{E}_{\theta'} \left[\hat{\mathbf{x}}_t^b \mathbf{x}_{t+1}^\top \right] + \mathbf{E}_{\theta'} \left[\hat{\mathbf{x}}_t^b \hat{\mathbf{x}}_{t+1}^{b\top} \right]$$

$$= \mathbf{E}_{\theta'} \left[\mathbf{x}_t \mathbf{x}_{t+1}^\top \right] - \hat{\mathbf{x}}_t^b \hat{\mathbf{x}}_{t+1}^{b\top} - \hat{\mathbf{x}}_t^b \hat{\mathbf{x}}_{t+1}^{b\top} + \hat{\mathbf{x}}_t^b \hat{\mathbf{x}}_{t+1}^{b\top}$$

$$= \mathbf{E}_{\theta'} \left[\mathbf{x}_t \mathbf{x}_{t+1}^\top \right] - \hat{\mathbf{x}}_t^b \hat{\mathbf{x}}_{t+1}^{b\top}. \tag{D.2}$$

Rearranging Eq. (D.2) gives

$$\mathbf{E}_{\theta'} \left[\mathbf{x}_t \mathbf{x}_{t+1}^\top \right] = \mathbf{M}_{t+1} + \hat{\mathbf{x}}_t^b \hat{\mathbf{x}}_{t+1}^{b\top}. \tag{D.3}$$

Substituting Eq. (D.3) into Eq. (D.1), noting the expectation commutes with the sum yields

$$\mathbf{\Xi}_0 = \sum_{t=0}^{T-1} \left(\mathbf{M}_{t+1} + \hat{\mathbf{x}}_t^b \hat{\mathbf{x}}_{t+1}^{b\top} \right). \tag{D.4}$$

An expression for $\mathbf{\Xi}_1$ in terms of the smoothed states can be also found in a similar way by calculating the covariance matrix, $\mathbf{P}_t^b$ and following the above steps.

Amari, S., 1977. Dynamics of pattern formation in lateral-inhibition type neural fields. Biological Cybernetics 27, 77–87.

Atay, F., Hutt, A., 2005. Stability and bifurcations in neural fields with finite propagation speed and general connectivity. SIAM Journal on Applied Mathematics 65, 644–666.

Bilmes, J., 1998. A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. International Computer Science Institute 4, 126.

Bishop, C., 2006. Pattern Recognition and Machine Learning. volume 4. Springer New York.

Breakspear, M., Heitmann, S., Daffertshofer, A., 2010. Generative models of cortical oscillations: neurobiological implications of the kuramoto model. Frontiers in Human Neuroscience 4.

Breakspear, M., Stam, C., 2005. Dynamics of a neural system with a multiscale architecture. Philosophical Transactions of the Royal Society B: Biological Sciences 360, 1051.

Burnham, K.P., Anderson, D.R., 2002. Model selection and multimodel inference: a practical information-theoretic approach. Springer Science & Business Media.

Chui, C., Wang, J., 1992a. A general framework of compactly supported splines and wavelets*1. Journal of Approximation Theory 71, 263–304.

Chui, C.K., Wang, J., 1992b. On compactly supported spline wavelets and a duality principle. American Mathematical Society 330, 903–915.

Daunizeau, J., Kiebel, S., Friston, K., 2009. Dynamic causal modelling of distributed electromagnetic responses. NeuroImage 47, 590–601.

David, O., Friston, K., 2003. A neural mass model for MEG/EEG: coupling and neuronal dynamics. NeuroImage 20, 1743–1755.

De Boor, C., 2001. A Practical Guide to Splines. Springer Verlag.

Dempster, A.P., Laird, N.M., Rubin, D.B., 1977. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society. Series B (Methodological) 39, 1–38.

Dewar, M., Scerri, K., Kadirkamanathan, V., 2009. Data-driven spatio-temporal modeling using the integro-difference equation. Signal Processing, IEEE Transactions on 57, 83–91.

Evensen, G., 2003. The ensemble Kalman filter: theoretical formulation and practical implementation. Ocean Dynamics 53, 343–367.

Evensen, G., 2009a. Data assimilation: The ensemble Kalman filter. Springer Verlag.

Evensen, G., 2009b. The ensemble Kalman filter for combined state and parameter estimation. IEEE Control Systems Magazine , 83–104.

Fessler, J., Hero, A., 1994. Space-alternating generalized expectation-maximization algorithm. Signal Processing, IEEE Transactions on 42, 2664–2677.

Freestone, D., Aram, P., Dewar, M., Scerri, K., Grayden, D., Kadirkamanathan, V., 2011. A data-driven framework for neural field modeling. NeuroImage 56, 1043–1058.

Galka, A., Ozaki, T., Muhle, H., Stephani, U., Siniatchkin, M., 2008. A data-driven model of the generation of human EEG based on a spatially distributed stochastic wave equation. Cognitive Neurodynamics 2, 101–113.

Ghosh, A., Rho, Y., McIntosh, A., Kötter, R., Jirsa, V., 2008. Noise during rest enables the exploration of the brain's dynamic repertoire. PLoS Computational Biology 4, e1000196.

Gibson, S., Ninness, B., 2005. Robust maximum-likelihood estimation of multivariable dynamic systems. Automatica 41, 1667–1682.

Goswami, J.C., Chan, A., 1999. Fundamentals of Wavelets: Theory, Algorithms, and Applications. Wiley, New York.

Hagmann, P., Cammoun, L., Gigandet, X., Meuli, R., Honey, C., Wedeen, V., Sporns, O., 2008. Mapping the structural core of human cerebral cortex. PLoS biology 6, e159.

Haxby, J., Gobbini, M., Furey, M., Ishai, A., Schouten, J., Pietrini, P., 2001. Distributed and overlapping representations of

faces and objects in ventral temporal cortex. Science 293, 2425–2430.

Haykin, S., 2001. Kalman filtering and Neural Networks. Wiley Online Library.

Jirsa, V., 2004. Connectivity and dynamics of neural information processing. Neuroinformatics 2, 183–204.

Jirsa, V., Haken, H., 1996. Field theory of electromagnetic brain activity. Physical Review Letters 77, 960–963.

Jirsa, V., Haken, H., 1997. A derivation of a macroscopic field theory of the brain from the quasi-microscopic neural dynamics. Physica D: Nonlinear Phenomena 99, 503–526.

Jirsa, V., Jantzen, K., Fuchs, A., Kelso, J., 2002. Spatiotemporal forward solution of the EEG and MEG using network modeling. Medical Imaging, IEEE Transactions on 21, 493–504.

Jirsa, V., Sporns, O., Breakspear, M., Deco, G., McIntosh, A., 2010. Towards the virtual brain: network modeling of the intact and the damaged brain. Archives italiennes de biologie 148, 189–205.

Jirsa, V.K., McIntosh, A.R., 2007. Handbook of brain connectivity. Springer.

Kalman, R., 1960. A new approach to linear filtering and prediction problems. Transactions of the ASME-Journal of Basic Engineering 82(Series D), 35–45.

Mallat, S.G., 1989. A theory for multiresolution signal decomposition: the wavelet representation. Pattern Analysis and Machine Intelligence, IEEE Transactions on 2, 674–693.

McLachlan, G., Krishnan, T., 1997. The EM algorithm and extensions. volume 274. Wiley New York. 2 edition.

Meng, X., 1994. On the rate of convergence of the ECM algorithm. The Annals of Statistics , 326–339.

Meng, X., Rubin, D., 1993. Maximum likelihood estimation via the ECM algorithm: A general framework. Biometrika 80, 267–278.

Meyer, C., 2000. Matrix analysis and applied linear algebra: solutions manual. volume 2. Society for Industrial and Applied Mathematics.

Meyer, Y., 1992. Wavelets and Operators. Cambridge University Press, New York.

Pinotsis, D., Moran, R., Friston, K., 2012. Dynamic causal modeling with neural fields. NeuroImage 59, 1261–1274.

Qubbaj, M., Jirsa, V., 2009. Neural field dynamics under variation of local and global connectivity and finite transmission speed. Physica D: Nonlinear Phenomena 238, 2331–2346.

Rauch, H.E., Tung, F., Striebel, C.T., 1965. Maximum likelihood estimates of linear dynamic systems. AIAA Journal 3, 1445–1450.

Scerri, K., Dewar, M., Kadirkamanathan, V., 2009. Estimation and model selection for an IDE-based spatio-temporal model. Signal Processing, IEEE Transactions on 57, 482–492.

Schiff, S., Sauer, T., 2008. Kalman filter control of a model of spatiotemporal cortical dynamics. Journal of Neural Engineering 5, 1–8.

Schultze-Kraft, M., Becker, R., Breakspear, M., Ritter, P., 2011. Exploiting the potential of three dimensional spatial wavelet analysis to explore nesting of temporal oscillations and spatial variance in simultaneous EEG-fMRI data. Progress in Biophysics and Molecular Biology 105, 67–79.

Shumway, R., Stoffer, D., 2000. Time Series Analysis and Its Applications. Springer Verlag.

Spiegler, A., Kiebel, S., Atay, F., Knösche, T., 2010. Bifurcation analysis of neural mass models: Impact of extrinsic inputs and dendritic time constants. NeuroImage 52, 1041–1058.

Spiegler, A., Knösche, T., Schwab, K., Haueisen, J., Atay, F., 2011. Modeling brain resonance phenomena using a neural mass model. PLoS Computational Biology 7, e1002298.

Stefanescu, R., Jirsa, V., 2008. A low dimensional description of globally coupled heterogeneous neural networks of excitatory and inhibitory neurons. PLoS Computational Biology 4, e1000219.

Stephan, K., Kasper, L., Harrison, L., Daunizeau, J., den Ouden, H., Breakspear, M., Friston, K., 2008. Nonlinear dynamic causal models for fMRI. NeuroImage 42, 649–662.

Unser, M., 1999. Splines: a perfect fit for signal and image processing. Signal Processing Magazine, IEEE 16, 22–38.

Unser, M., Aldroubi, A., Eden, M., 1993. A family of polynomial spline wavelet transforms. Signal Processing 30, 141–162.

Valdes, P., Jimenez, J., Riera, J., Biscay, R., Ozaki, T., 1999. Nonlinear EEG analysis based on a neural mass model. Biological Cybernetics 81, 415–424.

Van De Ville, D., Blu, T., Unser, M., 2005a. Isotropic polyharmonic B-splines: Scaling functions and wavelets. Image Processing, IEEE Transactions on 14, 1798–1813.

Van De Ville, D., Blu, T., Unser, M., 2005b. On the multidimensional extension of the quincunx subsampling matrix. Signal Processing Letters, IEEE 12, 112–115.

Van Rotterdam, A., Lopes da Silva, F., Van den Ende, J., Viergever, M., Hermans, A., 1982. A model of the spatial-temporal characteristics of the alpha rhythm. Bulletin of Mathematical Biology 44, 283–305.

Wendling, F., 2005. Neurocomputational models in the study of epileptic phenomena. Journal of Clinical Neurophysiology 22, 285.

Wikle, C.K., 2004. A kernel-based spectral model for non-Gaussian spatio-temporal processes. Statistical Modelling 2, 299–314.

Wilson, H., Cowan, J., 1973. A mathematical theory of the functional dynamics of cortical and thalamic nervous tissue. Biological Cybernetics 13, 55–80.

Xu, K., Wikle, C., Fox, N., 2005. A kernel-based spatio-temporal dynamical model for nowcasting weather radar reflectivities. Journal of the American Statistical Association 100, 1133–1144.