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

P. Aram, D.R. Freestone *Graduate Student Member, IEEE*, M. Dewar, K.Scerri *Member, IEEE*, D.B. Grayden *Member, IEEE,* and V. Kadirkamanathan *Member, IEEE*

Abstract

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. State and parameter estimation is performed using the Expectation Maximization (EM) algorithm. A synthetic example is provided to demonstrate the framework.

Index Terms

Neural field model, multi-resolution approximation (MRA), Expectation Maximization (EM) algorithm, wavelets.

P. Aram* and V. Kadirkamanathan are with the Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, S1 3JD, U.K. (e-mail:p.aram@sheffield.ac.uk; visakan@sheffield.ac.uk).

D. R. Freestone and D. B. Grayden are with the Department of Electrical and Electronic Engineering, The University of Melbourne, and The Bionic Ear Institute, VIC, Australia. (e-mail:dfreestone@bionicear.org; grayden@unimelb.edu.au). The Bionic Ear Institute acknowledges the support it receives from the Victorian State Government through the Operational Infrastructure Support Program.

M. Dewar is with the Department of Applied Physics and Applied Mathematics, Columbia University, New York, NY, USA. (e-mail:mike.dewar@columbia.edu).

K. Scerri is with the Department of Systems and Control Engineering, University of Malta, Msida, MSD, Malta. (e-mail:kenneth.scerri@um.edu.mt).

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

I. 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 of the EEG. The key components of these models typically describe the macroscopic dynamics of the human brain, but can also descriptive of finer-scale neurodynamics. For example, the post-synaptic response kernels (the pulse to wave function [**?**]) can be descriptive of inhibitory and excitatory post-synaptic potentials in microscopic single neural models, such as integrate and fire, or an ensemble response to stimulation in macroscopic neural mass models. Furthermore, the activation function (wave to pulse function) can be used used to model spiking statistics across spatial scales from single neurons (where is can be considered a CDF of the probability of firing) to mean firing rates of neural masses. The key factor linking the key functions of the model to particular spatial scales is geometry of the connectivity, hence the ability to infer a multi-resolution connectivity kernel will enable to 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 the infer 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. 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. It is expected that parameters of the neural fields models will be patient-specific and will, therefore, need to be inferred from data.

The first work describing data-driven mesoscopic neural modelling used a neural mass model to fit EEG data [1]. This approach was extended to coupled neural masses through a Bayesian estimation scheme dubbed Dynamic Causal Modelling (DCM) [2]. Following this work, data-driven modelling was extended to continuum field equations that explained the richer dynamics of spatiotemporal neural fields [3]–[5]. Most recently, a framework was developed where a finite element model (FEM) of the neural field (via a global Galerkin projection) was formed, using a basis function decomposition, to transform the PDE neural field equations into a finite dimension system to facilitate efficient state and parameter estimation [6]. 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 behaviour of the system can be represented simultaneously and completely.

II. MULTI-RESOLUTION CORTICAL ARCHITECTURE

The human neocortex has an inherent multi-resolution architecture. On finer scales... where any measurable 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 multi-electrode array or voltage sensitive dye, $\mathbf{r} \in \mathbb{R}^n$ (where $n \leq 3$) describes the spatial location with the field, t denotes time, and j indexes the different levels of resolution in the field.

This multi-resolution structure has been described by numerous methods [7]. This idea of multi-resolution can be thought of a generalisation of modelling the layers of the cortex separately, where the connectivity kernel, or neural foot-print, has varying widths.

Typically, the experimentalist only has access the net field generated by the multi-resolution architecture. Follow 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 [8] in using a wavelet decomposition.

III. IDE NEURAL FIELD MODEL

The stochastic IDE form of the Amari neural field formulation [9] is given by (see [6] for a full derivation)

$$v_{t+1}\left(\mathbf{r}\right) = \xi v_t\left(\mathbf{r}\right) + T_s \int_\Omega w\left(\mathbf{r}, \mathbf{r}'\right) f\left(v_t\left(\mathbf{r}'\right)\right) \, d\mathbf{r}' + e_t\left(\mathbf{r}\right), \tag{2}$$
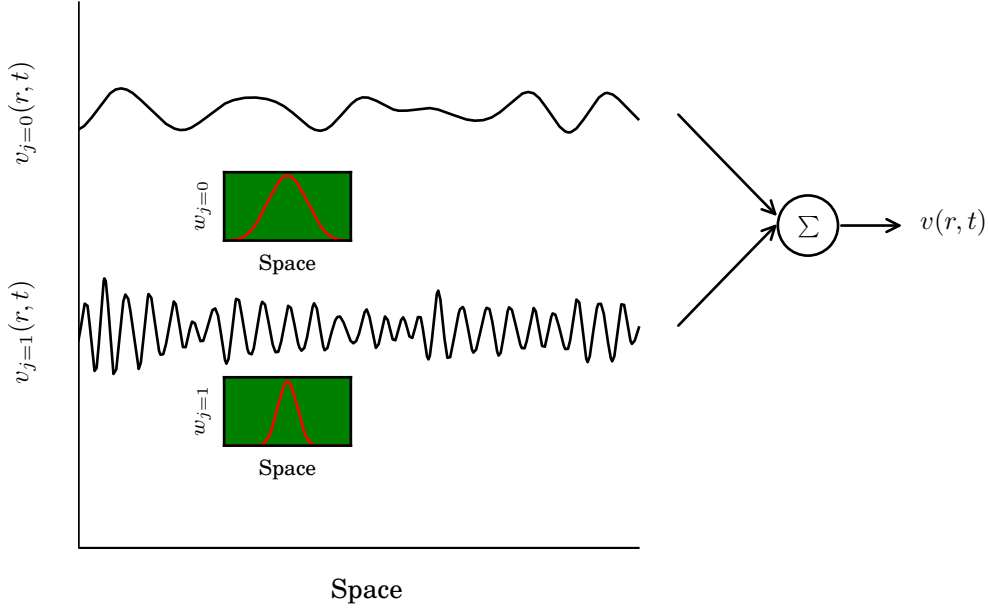
Fig. 1. **Multi-layer neural field model**. The inset figures show the shape of the connectivity kernel components at each spatial resolution.

where the post-synaptic membrane voltage at time t of a population of neurons at position $\mathbf{r}$ is denoted $v_t(\mathbf{r})$. Membrane dynamics are included through the parameter $\xi = 1 - T_s/\tau$, where τ is the membrane time constant and T_s is the sampling time. The connectivity strength between neural populations at a distance $\mid \mathbf{r} - \mathbf{r}' \mid$ is described by the connectivity kernel $w(\mathbf{r} - \mathbf{r}')$. The connectivity kernel is taken as a "Mexican hat" function, which describes local excitation and lateral inhibition [9], [10]. The term $e_t(\mathbf{r})$ is a zero mean Gaussian disturbance, spatially coloured but temporally white, with covariance function

$$cov\left(e_t\left(\mathbf{r}\right), e_{t+t'}\left(\mathbf{r}'\right)\right) = \eta(\mathbf{r} - \mathbf{r}')\delta(t - t'), \tag{3}$$

where $\delta(\cdot)$ is the Dirac-delta function. The firing rate of the presynaptic neurons is related to the post-synaptic membrane potential by the activation function $f(v_t(\mathbf{r})) = \varsigma v_t(\mathbf{r})$ [11], [12].

The observation equation describing the intracranial electrophysiological recordings is given by

$$y_t(\mathbf{r}_{n_y}) = \int_\Omega m\left(\mathbf{r}_{n_y} - \mathbf{r}'\right) v_t\left(\mathbf{r}'\right)\, d\mathbf{r}' + \boldsymbol{\epsilon}_t(\mathbf{r}_{n_y}), \tag{4}$$

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 i.i.d. Gaussian white noise process and the covariance matrix $\boldsymbol{\Sigma}_\epsilon = \sigma_\epsilon^2 \mathbf{I}_{n_y}$, where $\mathbf{I}$ denotes the identity matrix.

IV. MRA OF THE IDE IN STATE-SPACE

The purpose of forming the multi-resolution approximation (MRA) of the integro-differential 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. Therefore, the goal is to form a parameterised equation for coefficients of the field basis functions $\mathbf{x}_t$, which becomes the state vector.

The multi-resolution approximation [13] 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 an n-dimensional field, the static connectivity kernel is decomposed as,

$$w\left(\mathbf{r}\right) = \sum_{\mathbf{l} \in \mathbb{Z}^n} \alpha_{j_0,\mathbf{l}} \underbrace{\phi_{j_0,\mathbf{l}}\left(\mathbf{r}\right)}_{\text{scaling}} + \sum_{j \geq j_0}^{\infty} \sum_{\mathbf{l} \in \mathbb{Z}^n} \beta_{j,\mathbf{l}} \underbrace{\psi_{j,\mathbf{l}}\left(\mathbf{r}\right)}_{\text{wavelet}}, \tag{5}$$

where $\alpha_{j_0,\mathbf{l}}$ are the approximation coefficients at the lowest scale j_0, and $\beta_{j,\mathbf{l}}$ are the detail coefficients at different scales j, with

$$\phi_{j,\mathbf{l}}\left(\mathbf{r}\right) = 2^{\frac{j}{2}} \phi\left(2^j \mathbf{r} - \mathbf{l}\right) \tag{6}$$

$$\psi_{j,\mathbf{l}}\left(\mathbf{r}\right) = 2^{\frac{j}{2}} \psi\left(2^j \mathbf{r} - \mathbf{l}\right). \tag{7}$$

The parameters j and $\mathbf{l}$ control the scale (dilation) and translation (spatial shift), respectively. NEED SOMETHING ABOUT HOW j AND l ARE SET HERE - AS IN THAT THEY ARE KNOWN - AND THEAT WE ONLY NEED TO ESTIMATE THE COEFFICIENTS. 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

$$v_t\left(\mathbf{r}\right) = \sum_{\mathbf{l} \in \mathbb{Z}^n} x_{t,j_0,\mathbf{l}} \underbrace{\phi_{j_0,\mathbf{l}}\left(\mathbf{r}\right)}_{\text{scaling}} + \sum_{j \geq j_0}^{\infty} \sum_{\mathbf{l} \in \mathbb{Z}^n} \check{x}_{t,j,\mathbf{l}} \underbrace{\psi_{j,\mathbf{l}}\left(\mathbf{r}\right)}_{\text{wavelet}}, \tag{8}$$

where $x_{t,j_0,\mathbf{l}}$ and $\check{x}_{t,j,\mathbf{l}}$ are the dynamic coefficients of the expansion, constituting the state vector at time t. The decomposition enables a separation of the static spatial characteristics, which are considered static functions, that are weighted by dynamic coefficients.

Equations 5 and 8 are infinite series expansions and must be truncated to some level $j = J$ 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 such that

$$\boldsymbol{\alpha}_k \triangleq \{\alpha_{k,\mathbf{l}} : \mathbf{l} \in \mathbb{Z}^n\} \tag{9}$$

$$\boldsymbol{\beta}_k \triangleq \tag{10}$$

$$\mathbf{x}_{t,k} \triangleq \tag{11}$$

$$\check{\mathbf{x}}_{t,k} \triangleq \tag{12}$$

$$\boldsymbol{\phi}_{j_0}^\top(\mathbf{r}) \triangleq \tag{13}$$

$$\boldsymbol{\psi}_k^\top(\mathbf{r}) \triangleq, \tag{14}$$

where $k = j_0, j$. ARE THESE EACH n_θ or n_x DIMENSIONAL VECTORS? Following this we can define the vectors or matrices

$$\boldsymbol{\theta}^\top \triangleq [\ \boldsymbol{\alpha}_{j_0}^\top \quad \boldsymbol{\beta}_{j_0}^\top \quad \boldsymbol{\beta}_{j_0+1}^\top \quad \cdots \quad \boldsymbol{\beta}_J^\top \ ] \tag{15}$$

$$\mathbf{x}_t^\top \triangleq [\ \mathbf{x}_{t,j_0}^\top \quad \check{\mathbf{x}}_{t,j_0}^\top \quad \check{\mathbf{x}}_{t,j_0+1}^\top \quad \cdots \quad \check{\mathbf{x}}_{t,J}^\top \ ] \tag{16}$$

$$\boldsymbol{\lambda}^\top(\mathbf{r}) \triangleq \left[\ \boldsymbol{\phi}_{j_0}^\top(\mathbf{r}) \quad \boldsymbol{\psi}_{j_0}^\top(\mathbf{r}) \quad \boldsymbol{\psi}_{j_0+1}^\top(\mathbf{r}) \quad \cdots \quad \boldsymbol{\psi}_J^\top(\mathbf{r}) \ \right] \tag{17}$$

$$\boldsymbol{\mu}^\top(\mathbf{r}) \triangleq \left[\ \boldsymbol{\phi}_{j_0}^\top(\mathbf{r}) \quad \boldsymbol{\psi}_{j_0}^\top(\mathbf{r}) \quad \boldsymbol{\psi}_{j_0+1}^\top(\mathbf{r}) \quad \cdots \quad \boldsymbol{\psi}_J^\top(\mathbf{r}) \ \right], \tag{18}$$

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

$$w(\mathbf{r}) \approx \boldsymbol{\theta}^\top \boldsymbol{\lambda}(\mathbf{r}) \tag{19}$$

$$v_t(\mathbf{r}) \approx \boldsymbol{\mu}^\top(\mathbf{r}) \mathbf{x}_t. \tag{20}$$

Now we have the compact for of the decomposition we can substitute this into the neural field equation (21) giving

$$\boldsymbol{\mu}^\top(\mathbf{r})\mathbf{x}_t = \xi\boldsymbol{\mu}^\top(\mathbf{r})\mathbf{x}_t + T_s \int_\Omega \boldsymbol{\theta}^\top \boldsymbol{\lambda}(\mathbf{r} - \mathbf{r}') f\left(\boldsymbol{\mu}^\top(\mathbf{r}')\mathbf{x}_t\right) \mathrm{d}\mathbf{r}' + e_t(\mathbf{r}), \tag{21}$$

To derive the state-space model, we first define the matrices

$$\boldsymbol{\Lambda}_x \triangleq \int_\Omega \boldsymbol{\mu}(r)\boldsymbol{\mu}^\top(r)\, dr \tag{22}$$

$$\boldsymbol{\Lambda}_\theta \triangleq \int_\Omega \boldsymbol{\mu}(r) \int_\Omega \boldsymbol{\theta}^\top \boldsymbol{\lambda}(r - r')\boldsymbol{\mu}^\top(r)\, dr' dr. \tag{23}$$

Substituting (19) and (20) in (21), pre-multiplying by $\boldsymbol{\mu}(r)$, and integrating over the space gives

$$\boldsymbol{\Lambda}_x \mathbf{x}_{t+1} = \xi \boldsymbol{\Lambda}_x \mathbf{x}_t + T_s \varsigma \boldsymbol{\Lambda}_\theta \mathbf{x}_t + \int_\Omega \boldsymbol{\mu}(r) e_t(r) \, dr. \tag{24}$$

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

$$\mathbf{x}_{t+1} = \mathbf{A}(\boldsymbol{\theta}) \mathbf{x}_t + \mathbf{w}_t \tag{25}$$

$$\mathbf{A}(\boldsymbol{\theta}) = T_s \varsigma \boldsymbol{\Lambda}_x^{-1} \boldsymbol{\Lambda}_\theta + \xi \mathbf{I}. \tag{26}$$

The disturbance becomes

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

which is a vector valued, zero-mean normally distributed, white noise process with covariance (see [6] for proof)

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

The observation equation of the state-space model is found by substituting decomposition (20) into (4) giving

$$\mathbf{y}_t = \mathbf{C} \mathbf{x}_t + \boldsymbol{\varepsilon}_t, \tag{29}$$

where each element of the observation matrix is given by

$$\mathbf{C}_{ij} \triangleq \int_\Omega m(r_i - r') \boldsymbol{\mu}_j(r') \, d\mathbf{r}'. \tag{30}$$

V. B-SPLINE MRAIDE IN STATE-SPACE

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 [14] where the higher the order, the smoother the resulting B-spline becomes (see Figure 2).

B-spline wavelet and scaling functions are formulated independently by Chui and Wang, and Unser et al [15]–[17]. 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 localisation 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 [18]. B-spline wavelets have a better approximation rate as other wavelets with the same number of vanishing moments [19]. B-spline wavelet and scaling functions are particularly suitable in this work due to their ability to analytically
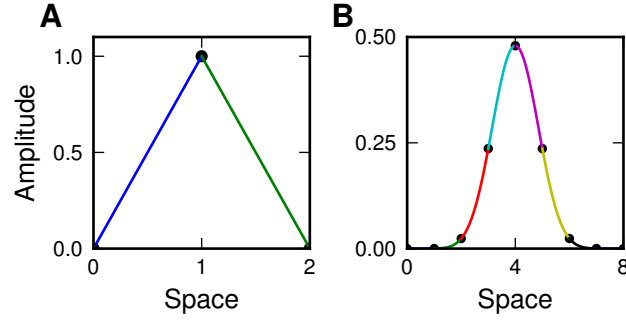
Fig. 2. **Examples of cardinal B-spline functions**. Piecewise polynomial functions (coloured lines) are joined at the break points (black dots). (**A**) B-spline function of order 2 (linear B-spline). (**B**) B-spline function of order 8.

define the convolution and inner product to form MRAIDE components. The following summarises some important characteristics of B-splines relevant to the proposed modelling framework.

The m-th order cardinal B-spline function is defined by the recurrence relation [16]

$$N_m (r) = (N_{m-1} * N_1) (r) = \int_0^1 N_{m-1} (r - r') \, dr',$$ (31)

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

$$N_1 (r) = \begin{cases} 1 & \text{if } 0 \leq r < 1, \\ 0 & \text{elsewhere.} \end{cases}$$ (32)

then a B-spline of any order, $N_m(r)$, can be computed using the recursive expression defined by [20]

$$N_m (r) = \frac{r}{m-1} N_{m-1} (r) + \frac{m - r}{m - 1} N_{m-1} (r - 1) \quad m > 1.$$ (33)

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

$$N_4 (r) = \begin{cases} r^3 & \text{if } 0 \leq r \leq 1, \\ 4 - 12r + 12r^2 - 3r^3 & \text{if } 1 \leq r \leq 2, \\ -44 + 60r - 24r^2 + 3r^3 & \text{if } 2 \leq r \leq 3, \\ 64 - 48r + 12r^2 - r^3 & \text{if } 3 \leq r \leq 4, \\ 0 & \text{elsewhere.} \end{cases}$$ (34)

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 $\mathbf{\Lambda}_x$, $\mathbf{\Lambda}_\theta$ and $\mathbf{\Sigma}_w$). To show how the convolution of two B-splines is calculated, (31) can be rewritten as $(m-1)$ convolutions of the indicator function with itself

$$N_m\left(r\right) = \underbrace{\left(N_1 * N_1 * \cdots * N_1\right)}_{m-1 \quad \text{convolutions}}\left(r\right). \tag{35}$$

Using the associativity property of convolution, we have

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

$$= N_{m+m'}\left(r\right). \tag{36}$$

A direct consequence of (36) is the inner product between two B-splines is another translated B-spline such that

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

$$= N_{m+m'}\left(m'+l_2-l_1\right), \tag{37}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product. This holds as the support of $N_m\left(r\right)$ 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 IX-A for explicit derivation). The values of $N_{m+m'}$ in (37) can be easily determined recursively by evaluating (33) at integer points, i.e.

$$\begin{cases} N_2(k) = \delta(k-1) & k \in \mathbb{Z}, \\ N_m\left(k\right) = \frac{k}{m-1}N_{m-1}\left(k\right) + \frac{m-k}{m-1}N_{m-1}\left(k-1\right) & k = 1, 2, \ldots, m-1 \end{cases} \tag{38}$$

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

To determine the level of approximation in the model, detailed in the previous section, that is the number of basis functions required to reconstruct the field, $v_t(r)$, the frequency response of the B-spline function needs to be computed by applying the Fourier transform to (35). To proceed the Fourier transform of $N_1(r)$ is first calculated,

$$\mathcal{F}(N_1(r)) = \int_{-\infty}^{+\infty} N_1(r)\exp(-2\pi i\nu r)dr$$

$$= \int_0^1 \exp(-2\pi i\nu r)dr$$

$$= \frac{1-\exp(-2\pi i\nu)}{2\pi i\nu}. \tag{39}$$

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 (35) and substituting (39) for $\mathcal{F}(N_1(r))$ yields

$$\mathcal{F}(N_m(r)) = \left(\frac{1 - \exp(-2\pi i \nu)}{2\pi i \nu} \right)^m. \tag{40}$$

The multi-resolution approximation using B-spline functions is completed by defining a two-scale relation pair which for cardinal B-spline scaling and wavelet functions of order m, takes the form of [16]

$$N_m(r) = \sum_{n=0}^{m} [\mathbf{p}]_n N_m(2r - n) \tag{41}$$

$$\varphi_m(r) = \sum_{n=0}^{3m-2} \frac{(-1)^n}{2^{m-1}} [\mathbf{q}]_n N_m(2r - n), \tag{42}$$

where $[\cdot]_n$ denotes the n-th element of the vector coefficients, and

$$[\mathbf{p}]_n = 2^{-m+1} \binom{m}{n} \quad 0 \le n \le m \tag{43}$$

$$[\mathbf{q}]_n = \sum_{l=0}^{m} \binom{m}{l} N_{2m}(n - l + 1), \quad 0 \le n \le 3m - 2. \tag{44}$$

Figure 3 shows the cubic B-spline scaling function and its associated wavelet function with dilation and translation parameters set to zero.
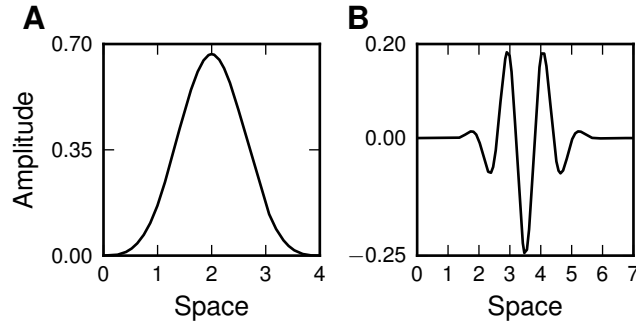

Fig. 3. **Examples of B-spline scaling and wavelet functions**. (**A**) The cubic B-spline scaling function. (**B**) The corresponding wavelet function.

By exploiting (36) and (37), the integrals in (22), (23) and (28) 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 [17]:

$$\langle \varphi_{m;j_1,l_1}(r), \varphi_{m;j_2,l_2}(r) \rangle = 0 \quad \text{for} \quad j_1 \ne j_2 \tag{45}$$

$$\langle N_{m;j_1,l_1}(r), \varphi_{m;j_2,l_2}(r) \rangle = 0 \quad \text{for} \quad j_1 \leq j_2. \tag{46}$$

In order to analytically calculate elements of $\mathbf{\Lambda}_x$ and $\mathbf{\Lambda}_\theta$, the scaling and wavelet basis functions must be expanded in terms of N_m at the appropriate scale. For scaling functions this can be done by using (41) recursively, giving

$$N_m(r) =$$
$$\sum_{n_1,n_2,...n_j=0}^{m} [\mathbf{p}]_{n_1} [\mathbf{p}]_{n_2} \cdots [\mathbf{p}]_{n_j} N_m(2^j r - l_{n_1,n_2,...,n_j}), \tag{47}$$

where

$$l_{n_1,n_2,...,n_j} = 2^{j-1} n_1 + 2^{j-2} n_2 + \cdots + 2^0 n_j. \tag{48}$$

The relation (47) can be also proven by induction (see Appendix ...). A similar method can be used for wavelet functions noting that they must be written first in terms of scaling functions using (42). Examples of such expansions are given in Figure 4 where the scaling and wavelet functions at $j = 0$ are written in terms of 625 and 1250 scaling functions at $j = 4$ respectively. In this paper, 4th − order cardinal
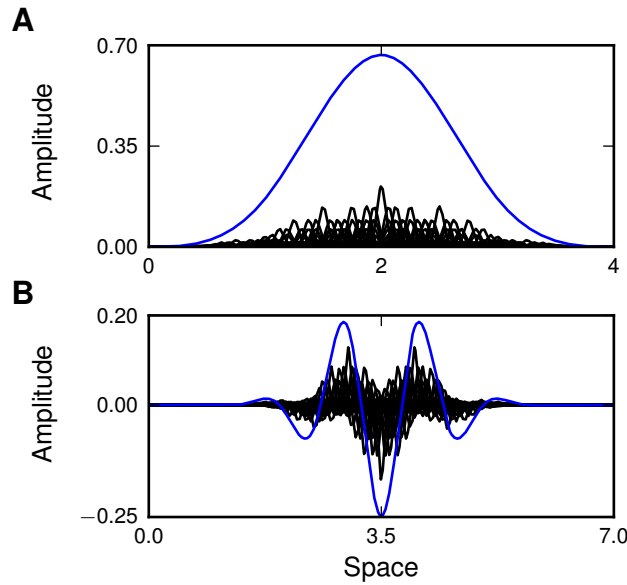
Fig. 4.   **The scaling and the wavelet function decomposition**. (**A**) The scaling function (blue) at $j = 0$ is obtained using 625 scaling functions (black) at $j = 4$. (**B**) The wavelet function (blue) at $j = 0$ is obtained using 1250 scaling functions (black) at $j = 4$.

B-spline scaling and wavelet functions are used, resulting in 8th and 12th order B-spline functions when calculating (22), (23) and (28) whose values at integer points are given in Table I.

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

TABLE I

Cardinal B-Splines at the knot sequence. TABLE ENTRIES ARE CALCULATED USING (38) AND CAN BE ALSO FOUND IN [14].

VI. STATE AND PARAMETER ESTIMATION

The state-space representation of the MRAIDE allows the use of the well known Expectation Maximization (EM) algorithm [21] to infer both the kernel and the field from electrophysiological data. The EM algorithm, when used in this context [22], yields the maximum likelihood kernel estimate and the posterior distribution of the field over time. We use the standard EM algorithm for linear dynamic systems [23] and hence only describe aspects of the algorithm that are specific to MRA neural field equations. EM essentially finds increasingly tighter lower bounds on the likelihood of the kernel $p(\mathbf{y}_1 \ldots \mathbf{y}_T | w)$ so that, at convergence, the maximum of the bound corresponds to the maximum of the likelihood. For the linear state-space model given by (25–30), the joint probability distribution $p(\mathbf{X}, \mathbf{Y}; \boldsymbol{\Theta})$ can be written as

$$p(\mathbf{X}, \mathbf{Y}; \boldsymbol{\Theta}) = \prod_{t=0}^{T-1} p(\mathbf{y}_{t+1} | \mathbf{x}_{t+1}) p(\mathbf{x}_{t+1} | \mathbf{x}_t; \boldsymbol{\theta}) p(\mathbf{x}_0). \tag{49}$$

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

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

$$= \mathbf{E}_{\boldsymbol{\Theta}'} \left[\sum_{t=0}^{T-1} 2 \ln p(\mathbf{y}_{t+1} | \mathbf{x}_{t+1}) + \sum_{t=0}^{T-1} 2 \ln p(\mathbf{x}_{t+1} | \mathbf{x}_t; \boldsymbol{\theta}) + \sum_{t=0}^{T-1} 2 \ln p(\mathbf{x}_0) \right], \tag{50}$$

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

$$\mathcal{Q}(\mathbf{\Theta}, \mathbf{\Theta}') = \mathbf{E}_{\Theta'} \left[\sum_{t=0}^{T-1} 2 \ln p(\mathbf{x}_{t+1}|\mathbf{x}_t; \boldsymbol{\theta}) \right] + \text{const.} \tag{51}$$

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 normalising term, $1/(2\pi)^{\frac{n_x}{2}} \mid \mathbf{\Sigma}_w \mid^{-\frac{1}{2}}$ — the conditional distribution $p(\mathbf{x}_{t+1}|\mathbf{x}_t; \mathbf{\Theta})$ can be written as

$$p(\mathbf{x}_{t+1}|\mathbf{x}_t; \mathbf{\Theta}) = \exp \left(-\frac{1}{2} \left(\mathbf{x}_{t+1} - \mathbf{A}(\boldsymbol{\theta}) \mathbf{x}_t \right)^{\top} \mathbf{\Sigma}_w^{-1} \left(\mathbf{x}_{t+1} - \mathbf{A}(\boldsymbol{\theta}) \mathbf{x}_t \right) \right). \tag{52}$$

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

$$2 \ln p(\mathbf{x}_{t+1}, \mathbf{y}_{t+1}; \mathbf{\Theta}) = - \mathbf{x}_{t+1}^{\top} \mathbf{\Sigma}_w^{-1} \mathbf{x}_{t+1} + 2\mathbf{x}_{t+1}^{\top} \mathbf{\Sigma}_w^{-1} \mathbf{A}(\boldsymbol{\theta}) \mathbf{x}_t$$
$$- \mathbf{x}_t^{\top} \mathbf{A}^{\top}(\boldsymbol{\theta}) \mathbf{\Sigma}_w^{-1} \mathbf{A}(\boldsymbol{\theta}) \mathbf{x}_t. \tag{53}$$

Substituting $\mathbf{A}(\boldsymbol{\theta})$ from (26) into (53) gives

$$2 \ln p(\mathbf{x}_{t+1}, \mathbf{y}_{t+1}; \mathbf{\Theta}) = \beta + 2T_s \varsigma \mathbf{x}_{t+1}^{\top} \mathbf{\Sigma}_w^{-1} \mathbf{\Lambda}_x^{-1} \mathbf{\Lambda}_\theta \mathbf{x}_t$$
$$- T_s^2 \varsigma^2 \mathbf{x}_t^{\top} \mathbf{\Lambda}_\theta^{\top} \mathbf{\Lambda}_x^{-1} \mathbf{\Sigma}_w^{-1} \mathbf{\Lambda}_x^{-1} \mathbf{\Lambda}_\theta \mathbf{x}_t - 2\xi \varsigma T_s \mathbf{x}_t^{\top} \mathbf{\Sigma}_w^{-1} \mathbf{\Lambda}_x^{-1} \mathbf{\Lambda}_\theta \mathbf{x}_t, \tag{54}$$

where

$$\beta = -\mathbf{x}_{t+1}^{\top} \mathbf{\Sigma}_w^{-1} \mathbf{x}_{t+1} + 2\xi \mathbf{x}_{t+1}^{\top} \mathbf{\Sigma}_w^{-1} \mathbf{x}_t - \xi^2 \mathbf{x}_t^{\top} \mathbf{\Sigma}_w^{-1} \mathbf{x}_t, \tag{55}$$

is constant with respect to 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

$$2 \ln p(\mathbf{x}_{t+1}, \mathbf{y}_{t+1}; \mathbf{\Theta}) = \beta + 2T_s \varsigma \text{tr} \left\{ \mathbf{x}_t \mathbf{x}_{t+1}^{\top} \mathbf{\Sigma}_w^{-1} \mathbf{\Lambda}_x^{-1} \mathbf{\Lambda}_\theta \right\}$$
$$- T_s^2 \varsigma^2 \text{tr} \left\{ \mathbf{x}_t \mathbf{x}_t^{\top} \mathbf{\Lambda}_\theta^{\top} \mathbf{\Lambda}_x^{-1} \mathbf{\Sigma}_w^{-1} \mathbf{\Lambda}_x^{-1} \mathbf{\Lambda}_\theta \right\}$$
$$- 2\xi \varsigma T_s \text{tr} \left\{ \mathbf{x}_t \mathbf{x}_t^{\top} \mathbf{\Sigma}_w^{-1} \mathbf{\Lambda}_x^{-1} \mathbf{\Lambda}_\theta \right\}. \tag{56}$$

Rearranging and taking the expectation of the log-likelihood function over all time instants gives the required lower-bound which is to be maximised for the optimal parameter estimates. Therefore

$$\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}') = \beta + 2T_s \varsigma \text{tr} \left\{ \mathbf{\Xi}_0 \mathbf{\Sigma}_w^{-1} \mathbf{\Lambda}_x^{-1} \mathbf{\Lambda}_\theta \right\} - 2\xi \varsigma T_s \text{tr} \left\{ \mathbf{\Xi}_1 \mathbf{\Sigma}_w^{-1} \mathbf{\Lambda}_x^{-1} \mathbf{\Lambda}_\theta \right\}$$
$$- T_s^2 \varsigma^2 \text{tr} \left\{ \mathbf{\Xi}_1 \mathbf{\Lambda}_\theta^{\top} \mathbf{\Lambda}_x^{-1} \mathbf{\Sigma}_w^{-1} \mathbf{\Lambda}_x^{-1} \mathbf{\Lambda}_\theta \right\}, \tag{57}$$

where

$$\mathbf{\Xi}_0 = \mathbf{E}_{\Theta'} \left[\sum_{t=0}^{T-1} \mathbf{x}_t \mathbf{x}_{t+1}^\top \right] \tag{58}$$

$$\mathbf{\Xi}_1 = \mathbf{E}_{\Theta'} \left[\sum_{t=0}^{T-1} \mathbf{x}_t \mathbf{x}_t^\top \right]. \tag{59}$$

Note that the expectation distributes over the trace sum. The three traces that form (57) are dealt with in turn. To begin, the first trace is written in terms of element-wise summations

$$\begin{aligned}
\text{tr} \left\{ \mathbf{\Xi}_0 \mathbf{\Sigma}_w^{-1} \mathbf{\Lambda}_x^{-1} \mathbf{\Lambda}_\theta \right\} &= \sum_{i=1}^{n_x} \left[\mathbf{\Xi}_0 \mathbf{\Sigma}_w^{-1} \mathbf{\Lambda}_x^{-1} \mathbf{\Lambda}_\theta \right]_{ii} \\
&= \sum_{i,j=1}^{n_x} \left[\mathbf{\Xi}_0 \right]_{ij} \left[\mathbf{\Sigma}_w^{-1} \mathbf{\Lambda}_x^{-1} \mathbf{\Lambda}_\theta \right]_{ji} \\
&= \sum_{i,j=1}^{n_x} \left[\mathbf{\Xi}_0 \right]_{ij} \sum_{k=1}^{n_x} \left[\mathbf{\Sigma}_w^{-1} \mathbf{\Lambda}_x^{-1} \right]_{jk} \left[\mathbf{\Lambda}_\theta \right]_{ki},
\end{aligned} \tag{60}$$

where $[\cdot]_{p,q}$ denotes the (p,q)-element of the matrix. Each element of $\mathbf{\Lambda}_\theta$ can be calculated using

$$\left[\mathbf{\Lambda}_\theta \right]_{k,i} = \left[\mathbf{U} \right]^{k,i} \boldsymbol{\theta}, \tag{61}$$

where $[\cdot]^{p,q}$ denotes the (p,q)-block of the block matrix, and where each $1 \times n_\theta$ block of the $n_x \times n_x n_\theta$ block matrix $\mathbf{U}$ is

$$\left[\mathbf{U} \right]^{k,i} = \int_{\mathbf{\Omega}} \left[\boldsymbol{\mu}(r) \right]_k \left[\int_{\mathbf{\Omega}} \boldsymbol{\mu}\left(r'\right) \boldsymbol{\lambda}^\top \left(r - r'\right) dr' \right]_{i:} dr, \tag{62}$$

where $[\cdot]_{i:}$ denotes the i-th row. By substituting for $\left[\mathbf{\Lambda}_\theta \right]_{k,i}$ from (61) the trace given in (60) can be written as

$$\begin{aligned}
\text{tr} \left\{ \mathbf{\Xi}_0 \mathbf{\Sigma}_w^{-1} \mathbf{\Lambda}_x^{-1} \mathbf{\Lambda}_\theta \right\} &= \sum_{i,j=1}^{n_x} \left[\mathbf{\Xi}_0 \right]_{ij} \sum_{k=1}^{n_x} \left[\mathbf{\Sigma}_w^{-1} \mathbf{\Lambda}_x^{-1} \right]_{jk} \left[\mathbf{U} \right]^{k,i} \boldsymbol{\theta} \\
&= \sum_{i,j=1}^{n_x} \left[\mathbf{\Xi}_0 \right]_{ij} \left[\Gamma_1 \right]^{j,i} \boldsymbol{\theta}
\end{aligned} \tag{63}$$

where

$$\left[\Gamma_1 \right]^{j,i} = \sum_{k=1}^{n_x} \left[\mathbf{\Sigma}_w^{-1} \mathbf{\Lambda}_x^{-1} \right]_{jk} \left[\mathbf{U} \right]^{k,i}. \tag{64}$$

The second trace of (57) is dealt with in a similar manner, giving

$$\text{tr} \left\{ \mathbf{\Xi}_1 \mathbf{\Sigma}_w^{-1} \mathbf{\Lambda}_x^{-1} \mathbf{\Lambda}_\theta \right\} = \sum_{i,j=1}^{n_x} \left[\mathbf{\Xi}_1 \right]_{ij} \left[\Gamma_1 \right]^{j,i} \boldsymbol{\theta}. \tag{65}$$

Likewise, the third trace of (57) is broken down into element-wise summations

$$\text{tr}\left\{\mathbf{\Xi}_1\mathbf{\Lambda}_\theta^\top\mathbf{\Lambda}_x^{-1}\mathbf{\Sigma}_w^{-1}\mathbf{\Lambda}_x^{-1}\mathbf{\Lambda}_\theta\right\} = \sum_{i,j,k,m=1}^{n_x} [\mathbf{\Xi}_1]_{i,j} [\mathbf{\Lambda}_\theta^\top]_{jk} \left[\mathbf{\Lambda}_x^{-1}\mathbf{\Sigma}_w^{-1}\mathbf{\Lambda}_x^{-1}\right]_{km} [\mathbf{\Lambda}_\theta]_{mi}$$

$$= \boldsymbol{\theta}^\top \sum_{i,j=1}^{n_x} [\mathbf{\Xi}_1]_{i,j} \sum_{k,m=1}^{n_x} [\mathbf{U}^\top]^{jk} \left[\mathbf{\Lambda}_x^{-1}\mathbf{\Sigma}_w^{-1}\mathbf{\Lambda}_x^{-1}\right]_{km} [\mathbf{U}]^{mi}\, \boldsymbol{\theta}$$

$$= \boldsymbol{\theta}^\top \sum_{i,j=1}^{n_x} [\mathbf{\Xi}_1]_{i,j} [\Gamma_2]^{j,i}\, \boldsymbol{\theta}, \tag{66}$$

where

$$[\Gamma_2]^{j,i} = \sum_{k,m=1}^{n_x} [\mathbf{U}^\top]^{jk} \left[\mathbf{\Lambda}_x^{-1}\mathbf{\Sigma}_w^{-1}\mathbf{\Lambda}_x^{-1}\right]_{km} [\mathbf{U}]^{mi}. \tag{67}$$

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

$$\mathcal{Q}\left(\boldsymbol{\theta},\boldsymbol{\theta}'\right) = \beta + 2T_s\varsigma\left(\boldsymbol{v}_0 - \boldsymbol{v}_1\right)\boldsymbol{\theta} - T_s\varsigma\boldsymbol{\theta}^\top\mathbf{\Upsilon}\boldsymbol{\theta}, \tag{68}$$

where

$$\boldsymbol{v}_0 = \sum_{i,j=1}^{n_x} [\mathbf{\Xi}_0]_{i,j}[\Gamma_1]_{j,i} \tag{69}$$

and

$$\boldsymbol{v}_1 = \xi \sum_{i,j=1}^{n_x} [\mathbf{\Xi}_1]_{i,j}[\Gamma_1]^{j,i} \tag{70}$$

$$\mathbf{\Upsilon} = T_s\varsigma \sum_{i,j=1}^{n_x} [\mathbf{\Xi}_1]_{i,j}[\Gamma_2]^{j,i} \tag{71}$$

All n_x^2 blocks of Γ_1 and Γ_2 can be computed as a one-off before the commencement of the EM iterations, which increases the speed of the M-step significantly compared to the implementation in [22]. The complexity of each step of the algorithm is summarised in Table II where for ease of comparison the computational complexity of the method in [22] is also identified.

By differentiating the $\mathcal{Q}$-function with respect to $\boldsymbol{\theta}$ we have

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

$$= 2T_s\varsigma(\boldsymbol{v}_0 - \boldsymbol{v}_1)^\top - 2T_s\varsigma\mathbf{\Upsilon}^\top\boldsymbol{\theta}. \tag{72}$$

Equating (72) to $\mathbf{0}$ yields

$$\boldsymbol{\theta} = \mathbf{\Upsilon}^{-\top}(\boldsymbol{v}_0 - \boldsymbol{v}_1)^\top. \tag{73}$$

Variable	Equation	Order	Order obtained from [22]
Ξ_0, Ξ_1	(58), (59)	$O(Tn_x^2)$	-
$\boldsymbol{v}_0, \boldsymbol{v}_1$	(69),(70)	$O(n_x^2 n_\theta)$	$O(n_x^2 n_\theta) + O(n_x^3)$
$\boldsymbol{\Upsilon}$	(71)	$O(n_x^2 n_\theta^2)$	$O(n_x^4 n_\theta^2)$
$\mathbf{A}(\boldsymbol{\theta})$	(26)	$O(n_x^3)$	-
$\boldsymbol{\Lambda}_\theta$	(23)	$O(n_x^2 n_\theta)$	-
$\hat{\boldsymbol{\theta}}$	(73)	$O(n_\theta^3)$	-

TABLE II

The Computational complexity of the estimation algorithm. A COMPARISON BETWEEN THE ALGORITHM PROPOSED HEREIN AND THAT OF [22].

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

$$\frac{\partial^2 \mathcal{Q}}{\partial \boldsymbol{\theta}^2} = -2T_s \varsigma \boldsymbol{\Upsilon}. \tag{74}$$

The matrix $\boldsymbol{\Upsilon}$ is positive definite if $n_x^2 \times n_\theta$ matrix $\text{vec}(\mathbf{U})$ is of rank n_θ ([22], Lemma 3), where $\text{vec}(\cdot)$ denotes the vectorisation operator. Under this condition $\boldsymbol{\Upsilon}$ is invertible and the second derivative is negative definite representing a maximum of the $\mathcal{Q}$-function.

The matrices $\boldsymbol{\Xi}_0$ and $\boldsymbol{\Xi}_1$ are calculated using the Rauch Tung Streibel smoother [24] outputs: state estimates, $\hat{\mathbf{x}}_t$, covariance, $\mathbf{P}_t = \text{cov}(\mathbf{x_t})$, and cross-covariance matrix, $\mathbf{M}_t = \text{cov}(\mathbf{x}_t, \mathbf{x}_{t+1})$ [25],

$$\boldsymbol{\Xi}_0 = \sum_{t=0}^{T-1} \left(\mathbf{M}_{t+1} + \hat{\mathbf{x}}_t \hat{\mathbf{x}}_{t+1}^\top \right) \tag{75}$$

$$\boldsymbol{\Xi}_1 = \sum_{t=0}^{T-1} \left(\mathbf{P}_t + \hat{\mathbf{x}}_t \hat{\mathbf{x}}_t^\top \right). \tag{76}$$

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 $Q(\theta, \theta')$. The EM algorithm iterates between these two steps until the parameter estimates converge. The RTS algorithm is given in Algorithm 1 for completeness. The stopping rule adopted herein is

$$\left(\parallel \mathbf{A} \parallel_F^{(i)} - \parallel \mathbf{A} \parallel_F^{(i-1)} \right) < \epsilon, \tag{77}$$

where ϵ is a threshold value and $\parallel \mathbf{A} \parallel_F^{(i)}$ and $\parallel \mathbf{A} \parallel_F^{(i-1)}$ are the Frobenius norms of the successive

---

Algorithm 1 Summary of the Rauch-Tung-Striebel smoother

---

1) Forward initialisation:

$$\hat{\mathbf{x}}_0, \mathbf{P}_0$$

2) Forward iteration: for $t \in \{0, \cdots, T\}$, calculate the predicted state and the predicted covariance matrix

$$\hat{\mathbf{x}}_{t+1}^{f-} = \mathbf{A}\hat{\mathbf{x}}_t^f$$

$$\mathbf{P}_{t+1}^{f-} = \mathbf{A}\mathbf{P}_t^f\mathbf{A}^\top + \boldsymbol{\Sigma}_e$$

3) Compute the filter gain, the filtered state and the filtered covariance matrix

$$\mathcal{K}_{t+1} = \mathbf{P}_{t+1}^{f-}\mathbf{C}^\top(\mathbf{C}\mathbf{P}_{t+1}^{f-}\mathbf{C}^\top + \boldsymbol{\Sigma}_\varepsilon)^{-1}$$

$$\hat{\mathbf{x}}_{t+1}^f = \hat{\mathbf{x}}_{t+1}^{f-} + \mathcal{K}_{t+1}(\mathbf{y}_{t+1} - \mathbf{C}\hat{\mathbf{x}}_{t+1}^{f-})$$

$$\mathbf{P}_{t+1}^f = (\mathbf{I} - \mathcal{K}_{t+1}\mathbf{C})\mathbf{P}_{t+1}^{f-}$$

4) Backward initialisation:

$$\mathbf{P}_T^b = \mathbf{P}_T^f, \quad \hat{\mathbf{x}}_T^b = \hat{\mathbf{x}}_T^f$$

5) Backward iteration: for $t \in \{T-1, \cdots, 0\}$ compute the smoother gain, the smoothed state and the smoothed covariance matrix

$$\mathbf{S}_t = \mathbf{P}_t^f\mathbf{A}^\top\left[\mathbf{P}_{t+1}^{f-}\right]^{-1}$$

$$\hat{\mathbf{x}}_t^b = \hat{\mathbf{x}}_t^f + \mathbf{S}_t(\hat{\mathbf{x}}_{t+1}^b - \hat{\mathbf{x}}_{t+1}^{f-})$$

$$\mathbf{P}_t^b = \mathbf{P}_t^f + \mathbf{S}_t(\mathbf{P}_{t+1}^b - \mathbf{P}_{t+1}^{f-})\mathbf{S}_t^\top$$

6) compute the smoothed cross covariance matrix

initialisation:

$$\mathbf{M}_T = (\mathbf{I} - \mathcal{K}_T\mathbf{C})\mathbf{A}\mathbf{P}_{T-1}^b$$

Backward iteration: for $t \in \{T-1, \cdots, 0\}$

$$\mathbf{M}_t = \mathbf{P}_t^f\mathbf{S}_{t-1}^\top + \mathbf{S}_t(\mathbf{M}_{t+1} - \mathbf{A}\mathbf{P}_t^f)\mathbf{S}_{t-1}^\top$$

---

estimates of $\mathbf{A}$ matrices which is defined by [26]

$$\| \mathbf{A} \|_F = \sqrt{\sum_{i,j=1}^{n_x} | a_{i,j} |^2} = \sqrt{\mathrm{tr}(\mathbf{A}^\top \mathbf{A})} \tag{78}$$

VII. RESULTS

To demonstrate the performance of the MRAIDE estimation framework, data was generated synthetically using (21) and (4), allowing a comparison between true and estimated parameters. In doing so, the synaptic time constant was set to $\tau = 10$ ms [2], and the sampling time was chosen ten times faster [27], i.e. $T_s = 1$ ms, ensuring the discrepancy between the continuous neural field model and its discrete approximation is small.

Two different experiments were designed where in the first one, 200 runs of one second of data were used, where the field disturbance, e_t, and the observation noise, ε_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 realisation. In the second experiment, a single data set was generated and the field estimations are compared for five models, each accounting for a different level of approximation, i.e. $j = 0$ to $j = 4$.

For each of the experiment, the estimation was applied to the final 900 ms, allowing the model's dynamics to stabilise 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 $\mathbf{\Sigma}_\varepsilon = 0.1 \times \mathbf{I}_{n_y}$ and the disturbance covariance function was set to $\gamma(r - r') = \varphi_{3,-2}(r - r')$ (B-spline, $j = 3$). The firing rate slope, ς was set to $0.56 \ mV^{-1}$ [28]. The observation kernel was modelled using a B-spline function, with a width of 0.8 mm at half the maximum amplitude where the distance between adjacent sensors was 0.5 mm, resulting into $n_y = 161$ observations. 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 (45-46)).

A. 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 Figure 5 where the graph obtained by averaging the spectral power of the spatial frequency (over time) from the observations [29]. The spatial bandwidths of wavelets was calculated analytically using (40). 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 at $j = 0$ and 8, 16, 32 and 66 wavelet functions respectively at $j = 0$, $j = 1$, $j = 2$ and $j = 3$.
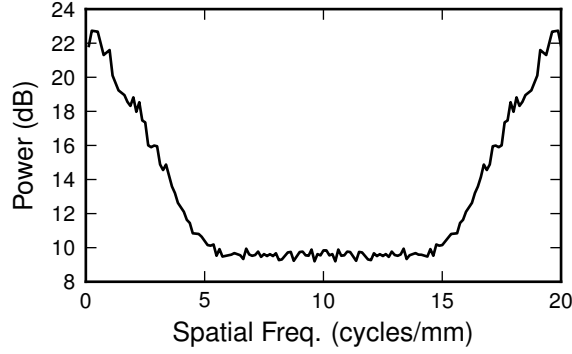

Fig. 5.   **Spatial frequency analysis**. The average (over time) power in dB of the spatial frequency of the observations.

The EM algorithm was allowed to run until the 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 Figure 6.
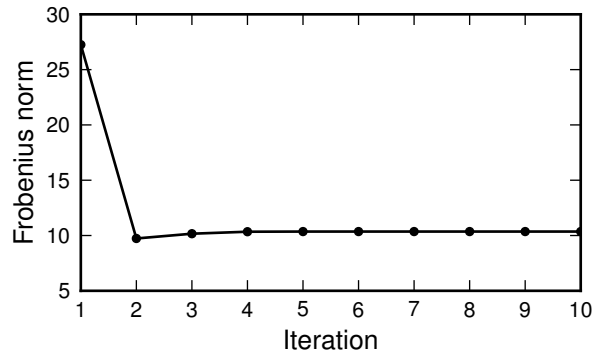
Fig. 6.   **Convergence of the EM algorithm**. Representative plot of the $\mathbf{A}(\boldsymbol{\theta})$ Frobenius norm vs iterations of EM based algorithm. The change in the stopping criterion falls below 10^{-6} after less than 10 iterations.

The actual connectivity kernel and the decomposition is plotted in Figure 7(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 need to be estimated. The small error in the estimate is attributed to

the MRA of the system used to form the estimator. Figure 7(b) and (c) illustrate the kernel scaling and wavelet functions, respectively. A total number of 25 basis functions was used to reconstruct the kernel, comprising 13 and 12 scaling and wavelet functions at $j = 1$ accordingly.
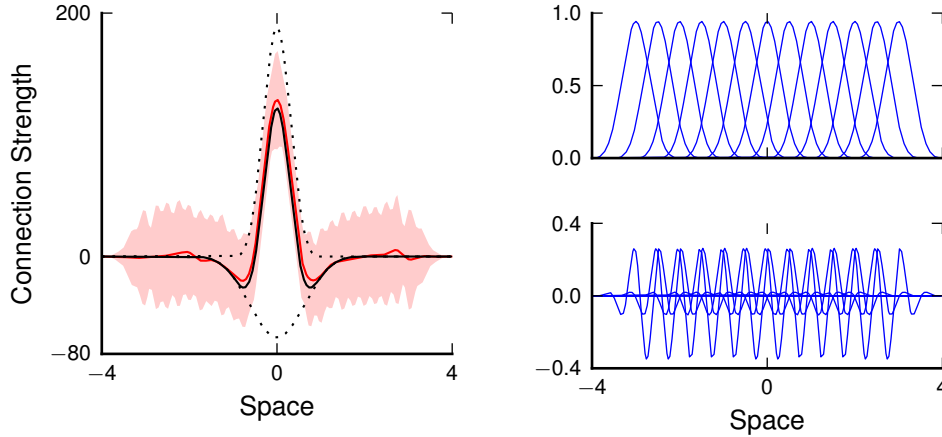

Fig. 7.   **Connectivity kernel estimate**. (**a**) The actual kernel and its components are shown by solid and dashed black lines, respectively. The mean kernel estimates over 200 realisations and confidence interval are shown by the red line and red shaded region (± 2 std.). (**b**) Kernel scaling and wavelets functions, respectively, at $j = 1$ used in the estimation algorithm.

One snap shot of the field reconstruction, at different levels of approximation is given in Figures 8, 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.

B. Experiment II

To study the performance of different models with various MRA capabilities, a single realisation was obtained using (21) and (4) under the same circumstances of the previous experiment. The generated 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 RMSE of the field estimation at different levels of approximation is shown in Figure 10. By increasing the value of j, error in the estimation decreases and converges to a steady value of 0.71, confirming that $j = 3$ is in fact adequate to capture the dynamics of the underlying field. The estimated field at a single time instant is shown in Figure 11. The experiment was performed several times and the results were consistent over each run.
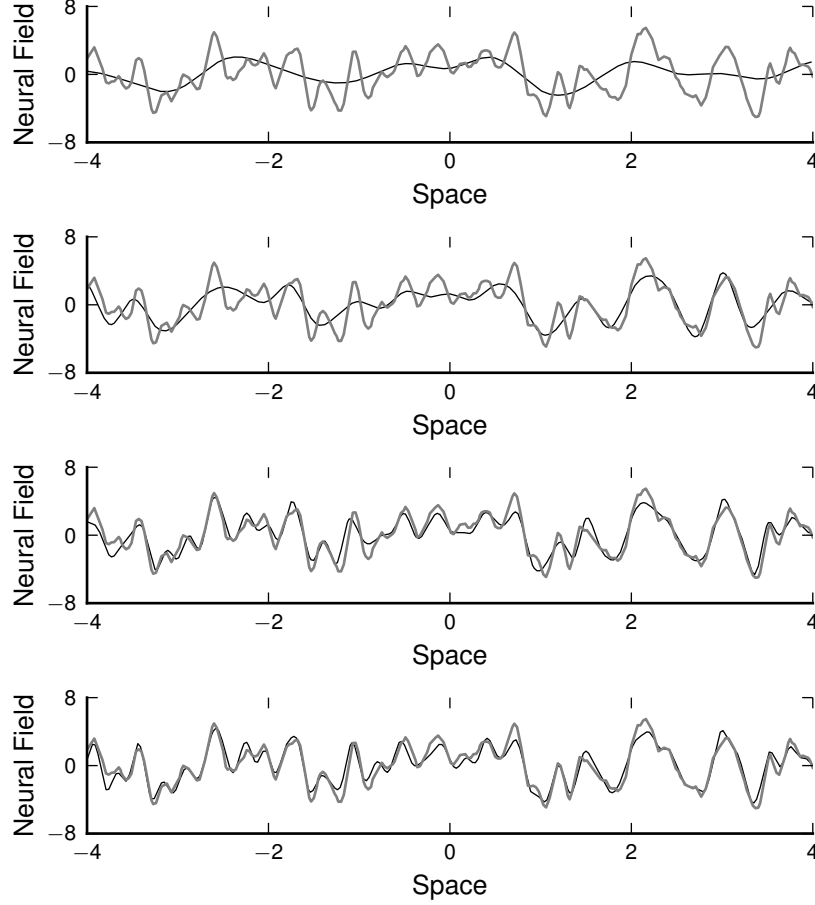
Fig. 8. **Examples of the actual and estimated neural fields for experiment I**. Actual (grey) and estimated (black) neural fields at a single time instant for different spatial resolutions. (**a**) $j = 0$, $n_x = 17$. (**b**) $j = 1$, $n_x = 33$. (**c**) $j = 2$, $n_x = 65$. (**d**) $j = 3$, $n_x = 131$.

VIII. 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 FEM representation of the neural field, and the estimator for an intracortical connectivity kernel with an arbitrary shape. 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, such as epilepsy. Other groups have previously highlighted the importance of a multi-resolution approach in neural field modeling. It is thought that the dynamics and connectivity structure differs at different spatial scales [8], [30].
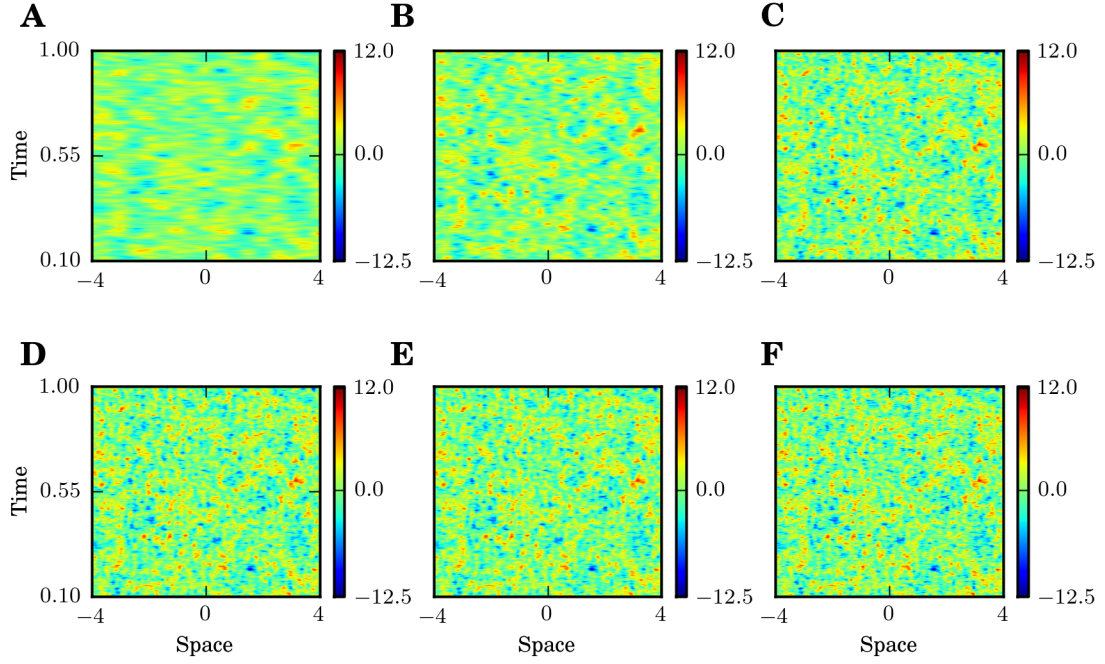
Fig. 9. **Actual and estimated spatiotemporal neural field at different spatial resolutions**. (**A**) The reconstructed field of the reduced model using at $j = 0$. (**B**) The reconstructed field at $j = 1$. (**C**) The reconstructed field at $j = 2$. (**D**) The reconstructed field at $j = 3$. (**E**) The reconstructed at $j = 4$. (**F**) The actual field.
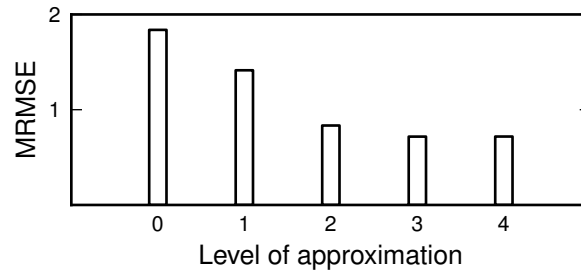

Fig. 10. **Error in the field reconstruction for experiment II**. MRMSE of the estimated field of models with different level of approximations. $j = 0$, $RMSE = 1.83$; $j = 1$, $RMSE = 1.41$; $j = 2$, $RMSE = 0.83$; $j = 3$, $RMSE = 0.71$; $j = 4$, $RMSE = 0.71$

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 maximisation (ECM) [31], [32] which replaces the M-step by a series
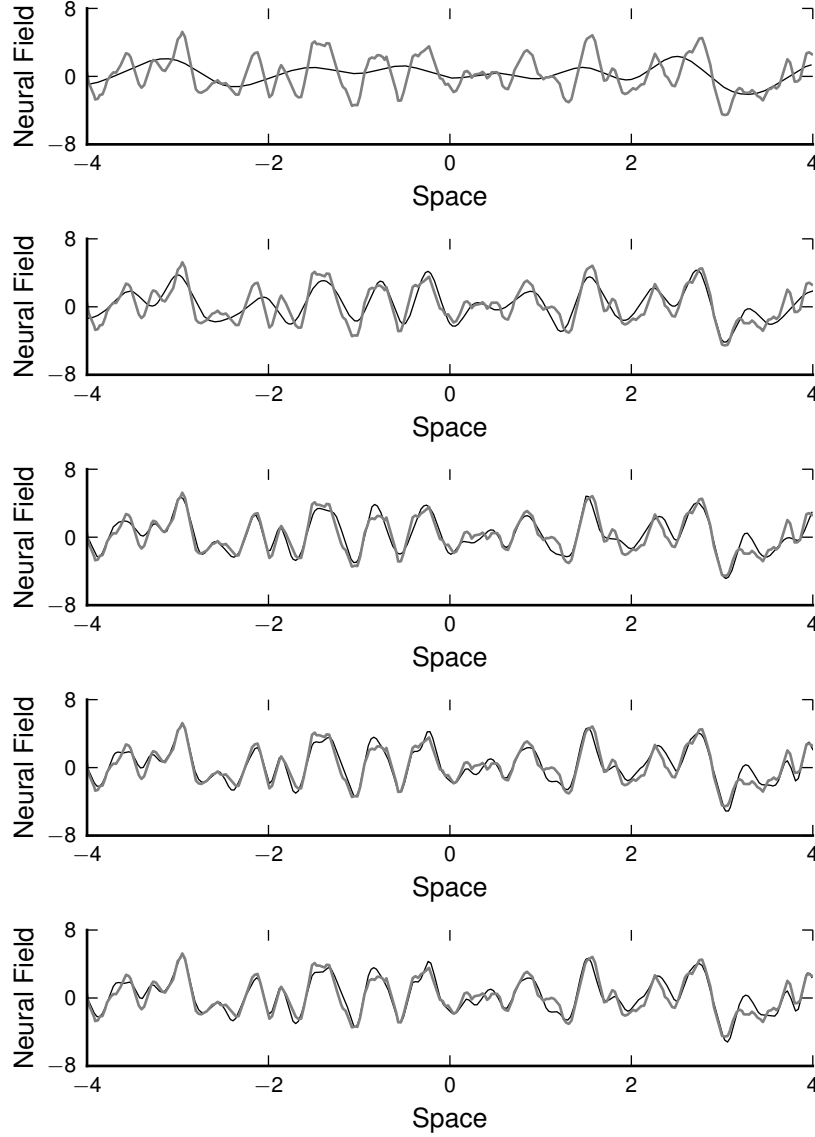
Fig. 11. **Examples of the actual and estimated neural fields for experiment II**. Actual (grey) and estimated (black) neural fields at a single time instant for models with different spatial resolutions. (**a**) $j = 0$, $n_x = 17$. (**b**) $j = 1$, $n_x = 33$. (**c**) $j = 2$, $n_x = 65$. (**d**) $j = 3$, $n_x = 131$. (**e**) $j = 4$, $n_x = 263$.

of computationally simplified conditional maximization (CM) steps. The ECM is a class of generalised EM (GEM) algorithms in which the $\mathcal{Q}$-function is increased rather than being maximised [33].

For systems with a high number of states, the ensemble Kalman smoother (EnKS) [34]–[36] 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. 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 [37]. In general, convergence of the EM algorithm to any stationary point depends on the initialisation. In the examples presented in this chapter, a bounded sequence of the state vectors was used to initialise the algorithm, which gave a satisfactory convergence and good state and parameter estimations.

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 [38], [39].

An assumption was made where the firing rate behaves linearly. The implementation of the framework in the non-linear case can be addressed as future work. Another future avenue would be the modification of the estimation framework to perform efficiently at higher dimensions. The reconstruction of the neural field 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 [40]. In this case three wavelet functions are required to extract fine features of the field at vertical, horizontal and diagonal orientations. Although provided formulations can be easily extended into two dimensions the computational load of the estimation algorithm will be significant. It is important to investigate methods for choosing basis functions to balance the complexity with the reconstruction capability of the model.

The authors acknowledge that there is, and will always be, a discrepancy between the model and cortex. Nevertheless, the model-based framework proposed in this paper may enable meaningful state tracking and connectivity estimation. 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 avenues for future work include extending the model to capture a 2nd-order synaptic response kernel, and time delays according to spatial resolutions. The estimation techniques should also be extended to deal with spatiotemporal heterogeneity in the kernel. Finally, future work should be directed towards applying and validating the framework on real data.

IX. 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 $N_m(s)$ is $[0, m]$ and is symmetric with respect to $r = \frac{m}{2}$, i.e.

$$N_m\left(\frac{m}{2} + r\right) = N_m\left(\frac{m}{2} - r\right). \tag{79}$$

A direct consequence of (79) is

$$N_m(r) = N_m(m - r). \tag{80}$$

Therefore

$$
\begin{aligned}
\int_{-\infty}^{+\infty} N_m(r - l_1) N_{m'}(r - l_2)\, ds &= \int_{-\infty}^{+\infty} N_m(m - r + l_1) N_{m'}(r - l_2)\, dr \\
&= \int_{-\infty}^{+\infty} N_m(m + l_1 - l_2 - u) N_{m'}(u)\, du \\
&= (N_m * N_{m'})(m + l_1 - l_2) \\
&= N_{m+m'}(m + l_1 - l_2) \\
&= N_{m+m'}(m' + l_2 - l_1).
\end{aligned} \tag{81}
$$

B. Induction

For n_1 in equation (47) we have

$$N_m(r) = \sum_{n_1=0}^{m} [\mathbf{p}]_{n_1} N_m(2r - n_1), \tag{82}$$

which clearly holds (see (41)). Now assuming (47) for n_{j-1} we have

$$N_m(r) = \sum_{n_1, n_2, \ldots n_{j-1}=0}^{m} [\mathbf{p}]_{n_1} [\mathbf{p}]_{n_2} \cdots [\mathbf{p}]_{n_{j-1}} N_m(2^{j-1}r - k_{n_1, n_2, \ldots, n_{j-1}}), \tag{83}$$

$N_m(2^{j-1}r - k_{n_1, n_2, \ldots, n_{j-1}})$ can be written in terms of $N_m(2^j r)$ as

$$
\begin{aligned}
N_m(2^{j-1}r - k_{n_1, n_2, \ldots, n_{j-1}}) &= \sum_{n_j=0}^{m} [\mathbf{p}]_{n_j} N_m(2^j r - 2 \times k_{n_1, n_2, \ldots, n_{j-1}} - n_j) \\
&= \sum_{n_j=0}^{m} [\mathbf{p}]_{n_j} N_m(2^j r - k_{n_1, n_2, \ldots, n_j})
\end{aligned} \tag{84}
$$

Substituting (84) in to (47) completes the proof.

REFERENCES

[1] P. Valdes, J. Jimenez, J. Riera, R. Biscay, and T. Ozaki, "Nonlinear EEG analysis based on a neural mass model," *Biological Cybernetics*, vol. 81, no. 5, pp. 415–424, 1999.

[2] O. David and K. Friston, "A neural mass model for MEG/EEG: coupling and neuronal dynamics," *NeuroImage*, vol. 20, no. 3, pp. 1743–1755, 2003.

[3] A. Galka, T. Ozaki, H. Muhle, U. Stephani, and M. Siniatchkin, "A data-driven model of the generation of human EEG based on a spatially distributed stochastic wave equation," *Cognitive Neurodynamics*, vol. 2, no. 2, pp. 101–113, 2008.

[4] S. Schiff and T. Sauer, "Kalman filter control of a model of spatiotemporal cortical dynamics," *Journal of Neural Engineering*, vol. 5, pp. 1–8, 2008.

[5] J. Daunizeau, S. Kiebel, and K. Friston, "Dynamic causal modelling of distributed electromagnetic responses," *NeuroImage*, vol. 47, no. 2, pp. 590–601, 2009.

[6] D. Freestone, P. Aram, M. Dewar, K. Scerri, D. Grayden, and V. Kadirkamanathan, "A data-driven framework for neural field modeling," *NeuroImage*, vol. 56, no. 3, pp. 1043–1058, June 2011.

[7] H. Wilson and J. Cowan, "A mathematical theory of the functional dynamics of cortical and thalamic nervous tissue," *Biological Cybernetics*, vol. 13, no. 2, pp. 55–80, 1973.

[8] M. Breakspear and C. Stam, "Dynamics of a neural system with a multiscale architecture," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 360, no. 1457, p. 1051, 2005.

[9] S. Amari, "Dynamics of pattern formation in lateral-inhibition type neural fields," *Biological Cybernetics*, vol. 27, no. 2, pp. 77–87, 1977.

[10] F. Atay and A. Hutt, "Stability and bifurcations in neural fields with finite propagation speed and general connectivity," *SIAM Journal on Applied Mathematics*, vol. 65, no. 2, pp. 644–666, 2005.

[11] A. Van Rotterdam, F. Lopes da Silva, J. Van den Ende, M. Viergever, and A. Hermans, "A model of the spatial-temporal characteristics of the alpha rhythm," *Bulletin of Mathematical Biology*, vol. 44, no. 2, pp. 283–305, 1982.

[12] B. Murphy and K. Miller, "Balanced amplification: a new mechanism of selective amplification of neural activity patterns," *Neuron*, vol. 61, no. 4, pp. 635–648, 2009.

[13] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 2, no. 7, pp. 674–693, Jul. 1989.

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

[15] C. Chui and J. Wang, "A general framework of compactly supported splines and wavelets*1," *Journal of Approximation Theory*, vol. 71, no. 3, pp. 263–304, 1992.

[16] C. K. Chui and J. Wang, "On compactly supported spline wavelets and a duality principle," *American Mathematical Society*, vol. 330, no. 2, pp. 903–915, 1992.

[17] M. Unser, A. Aldroubi, and M. Eden, "A family of polynomial spline wavelet transforms," *Signal Processing*, vol. 30, no. 2, pp. 141–162, Jan. 1993.

[18] ——, "On the asymptotic convergence of B-spline wavelets to Gabor functions," *Information Theory, IEEE Transactions on*, vol. 38, no. 2, pp. 864–872, 1992.

[19] M. Unser, "Splines: a perfect fit for signal and image processing," *Signal Processing Magazine, IEEE*, vol. 16, no. 6, pp. 22–38, 1999.

[20] C. De Boor, *A Practical Guide to Splines*. Springer Verlag, 2001.

[21] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[22] M. Dewar, K. Scerri, and V. Kadirkamanathan, "Data-driven spatio-temporal modeling using the integro-difference equation," *Signal Processing, IEEE Transactions on*, vol. 57, no. 1, pp. 83–91, Jan. 2009.

[23] R. Shumway and D. Stoffer, *Time Series Analysis and Its Applications*. Springer Verlag, 2000.

[24] H. E. Rauch, F. Tung, and C. T. Striebel, "Maximum likelihood estimates of linear dynamic systems," *AIAA Journal*, vol. 3, no. 8, pp. 1445–1450, 1965.

[25] S. Gibson and B. Ninness, "Robust maximum-likelihood estimation of multivariable dynamic systems," *Automatica*, vol. 41, no. 10, pp. 1667–1682, Oct. 2005.

[26] C. Meyer, *Matrix analysis and applied linear algebra: solutions manual*. Society for Industrial and Applied Mathematics, 2000, vol. 2.

[27] K. Stephan, L. Kasper, L. Harrison, J. Daunizeau, H. den Ouden, M. Breakspear, and K. Friston, "Nonlinear dynamic causal models for fMRI," *NeuroImage*, vol. 42, no. 2, pp. 649–662, 2008.

[28] F. Wendling, "Neurocomputational models in the study of epileptic phenomena." *Journal of Clinical Neurophysiology*, vol. 22, no. 5, p. 285, 2005.

[29] K. Scerri, M. Dewar, and V. Kadirkamanathan, "Estimation and model selection for an IDE-based spatio-temporal model," *Signal Processing, IEEE Transactions on*, vol. 57, no. 2, pp. 482–492, Feb. 2009.

[30] M. Qubbaj and V. Jirsa, "Neural field dynamics under variation of local and global connectivity and finite transmission speed," *Physica D: Nonlinear Phenomena*, vol. 238, no. 23-24, pp. 2331–2346, 2009.

[31] X. Meng and D. Rubin, "Maximum likelihood estimation via the ecm algorithm: A general framework," *Biometrika*, vol. 80, no. 2, pp. 267–278, 1993.

[32] X. Meng, "On the rate of convergence of the ecm algorithm," *The Annals of Statistics*, pp. 326–339, 1994.

[33] J. Fessler and A. Hero, "Space-alternating generalized expectation-maximization algorithm," *Signal Processing, IEEE Transactions on*, vol. 42, no. 10, pp. 2664–2677, 1994.

[34] G. Evensen, "The ensemble kalman filter: theoretical formulation and practical implementation," *Ocean Dynamics*, vol. 53, pp. 343–367, 2003.

[35] ——, "The ensemble kalman filter for combined state and parameter estimation," *IEEE Control Systems Magazine*, pp. 83–104, 2009.

[36] ——, *Data assimilation: The ensemble Kalman filter*. Springer Verlag, 2009.

[37] G. McLachlan and T. Krishnan, *The EM algorithm and extensions*, 2nd ed. Wiley New York, 1997, vol. 274.

[38] C. K. Wikle, "A kernel-based spectral model for non-Gaussian spatio-temporal processes," *Statistical Modelling*, vol. 2, pp. 299–314, 2004.

[39] K. Xu, C. Wikle, and N. Fox, "A kernel-based spatio-temporal dynamical model for nowcasting weather radar reflectivities," *Journal of the American Statistical Association*, vol. 100, no. 472, pp. 1133–1144, 2005.

[40] Y. Meyer, *Wavelets and Operators*. New York: Cambridge University Press, 1992.