Estimation of Stochastic Wilson and Cowan Neural Field Model using Expectation Maximisation Algorithm

In no specific order: Parham Aram, Dean R. Freestone, Kenneth Scerri, Michael Dewar, Andrew Zammit Mangion, David B. Grayden and Visakan Kadirkamanathan

2010-07-06

Abstract

- Background
- Method
- Result
- Conclusion

1 Introduction

- Understanding of macroscopic neurodynamics
- Neural field model
- Improvements in electrode technology with great spatiotemporal resolution

The human brain brain is arguably the most complicated system known to man. The development of a complete theory for its function is one of the greatest challenges faced by scientists today. To address this challenge, researchers have naturally used both theoretical and experimental frameworks to develop and test hypotheses. Experimental frameworks are typically designed to uncover causal relationships between the systems properties or parameters and its function (reverse engineering the brain). Theoretical studies are typically used in one of two ways. The first is to explain data acquired in an experiment and the second is to predict system behaviour.

Presently, it is almost half a century since Hodgkin and Huxley shared the Nobel prize (in 1963 with Eccles for Physiology and Medicine) for influential model that established the field of theoretical neuroscience. Over this time period, both experimental and theoretical methods have developed considerably. Experimental approaches have been able to isolate function using various forms or manipulation in greater detail. In parallel, theoretical/computational models have been able to explain and provide new hypotheses for an ever increasing assortment of neural phenomena. Through the development of these models a fundamental set a parametric equations has been established that explain neuronal responses to input a varying spatiotemporal scales from small patches of membrane to networks neural ensembles.

Although the predictive explanatory power of the theoretical models is rather vast when they are tuned to mimic experimental conditions, a major limitation has been in establishing a rigorous method for choosing model parameters in more general situations. For example, over the last decade it has come to light that there is significant variability in neural systems across subjects despite seemingly similar network activity (ref Marder 2006). It has been shown that a target network behaviour may be achieved by continuum of parameter combinations (Tail wagging the ref). We expect an analogous situation in disease states and pathological activity such as epileptic seizures. Therefore, for models to be clinically useful they must be subject-specific.

By assimilating experimental or clinical measurements with mathematical models one can infer unmeasured or latent system properties or parameters. Inference of patient-specific parameters has the the potential to revolutionise the treatment of disease. Typically, one can measure the blood-oxygen level dependent signal through fMRI or the electromagnetic fields of the brain through EEG or MEG. In clinical neurology, these measurement modalities give a kind or a 'where and when' indication of the presence of a pathology, but provide little information on the causative mechanisms for disease. A classic example of this is in epilepsy management, where the disease is diagnosed using EEG by identifying electrographic seizures. If electrical seizures are recorded with EEG, a medication plan is prescribed. The choice of medication is not based on the data acquired with the EEG, but based on the clinician's experience and the patient's circumstance, then the treatment plan is often modified based on trial and error. This process of searching for the best drug combination is often required similar electrographic seizures can have fundamentally different mechanisms of initiation (ref JTs paper). Ideally in epilepsy monitoring we would like to image the concentration of ions, the synaptic dynamics, the connectivity strength and structure etc. to better inform a treatment plan by understand physiological changes that lead to seizures. This is also highly desirable for epileptic seizure prediction.

The aim of this paper is to contribute towards to goal of being able to infer patient-specific parameters from electrophysiological data.

A key development in making this framework possible is the unscented Kalman filter. blah. Contributions from other groups...

Another recent approach uses graph theory [**?**, **?**] for large-scale neural networks and employs realistic anatomical connectivity from the CoCoMac database [**?**] to model the brain's resting state activity [**?**, **?**]. Each node in the network is described by a physiologically reasonable neural population model of their internal dynamics. The authors perform stability analysis on the network based on total connectivity and time delays in the presence of noise. Such an approach has the potential to model a virtual brain which can incorporate data from individual patients, giving insights into how patients' specific features can affect its large-scale neural dynamics [**?**, **?**].

2 Method

2.1 Neural Field Model

The neural field equations used as the parametric form of the model are based on the influential papers from Wilson and Cowan [**?**], and Amari [**?**]. This class of model describes a continuous cortical sheet or surface, relating mean firing rates of pre and post-synaptic neural populations. The neural field, $v\left(\mathbf{r},t\right)$, at position $\mathbf{r}$ and time t, is the aggregated post-synaptic potentials described by the **(temporal convolution - I'd rather say integral equation)** integral equation

$$v\left(\mathbf{r},t\right) = e^{-\zeta t} v\left(\mathbf{r},0\right) + \int_0^t h\left(t-t'\right) g\left(\mathbf{r},t'\right) \mathrm{d}t', \tag{1}$$

where $v\left(\mathbf{r},0\right)$ is the membrane voltage at the initial time $t=0$, $h(\cdot)$ is the synaptic response kernel assumed to be first order and of the form

$$h(t) = \eta(t)\exp\left(-\zeta t\right), \tag{2}$$

small thing: are we using $\exp(x)$ or e^x to represent exponential? $\zeta = \tau^{-1}$, τ is the synaptic time constant and $\eta(t)$ is the Heaviside step function **(you can keep it here if you wish as it doesn't change the integral equation because of the upper limit. I would remove it)**. $g(\cdot)$ describes the mean firing rate. In practice, the firing rate can rarely be considered as a purely deterministic process because **(can someone write down why?)**. We thus choose to introduce a spatially coloured space-time Wiener process $W(\mathbf{r},t)$ to model the fluctuations in the mean firing rate and decompose $g(\cdot)$ as

$$g(\mathbf{r},t)\mathrm{d}t = \tilde{g}(\mathbf{r},t)\mathrm{d}t + \sigma_W \mathrm{d}W(\mathbf{r},t) \tag{3}$$

where $\sigma_W \geq 0$ is a measure of the introduced space-time randomness. For $\sigma_W = 0$ a pure deterministic process is recovered. As a result of the random component, for $\sigma_W > 0$, the membrane voltage $v(\mathbf{r}, t)$ is itself a stochastic process. Assuming an infinite propagation velocity for action potentials within the field, the deterministic incoming firing rate is described by a spatial convolution **citation?**

$$\tilde{g}\left(\mathbf{r}, t\right) = \int_\Omega w\left(\mathbf{r}, \mathbf{r}'\right) f\left(v\left(\mathbf{r}', t\right)\right) \mathrm{d}\mathbf{r}', \tag{4}$$

where $w(\cdot)$ is the spatial connectivity kernel and Ω is the spatial domain representing the cortical sheet or surface. The function $f(\cdot)$ relates mean post-synaptic potentials to mean firing rates and follows a sigmoid described by

$$f\left(v\left(\mathbf{r}', t\right)\right) = \frac{1}{1 + \exp\left(\varsigma\left(v_0 - v\left(\mathbf{r}', t\right)\right)\right)}, \tag{5}$$

where v_0 and ς describe the firing threshold and the slope of the sigmoid respectively. By substituting equation **??** into equation **??** we obtain the stochastic integral equation model

$$v\left(\mathbf{r}, t\right) = e^{-\zeta t} v\left(\mathbf{r}, 0\right) + \int_0^t h\left(t - t'\right) \tilde{g}\left(\mathbf{r}, t'\right) \mathrm{d}t' + \sigma_W \int_0^t h\left(t - t'\right) \mathrm{d}W\left(\mathbf{r}, t'\right). \tag{6}$$

Here we assume that $v\left(\mathbf{r}, 0\right)$ is independent of the disturbance on the action potentials and assumed to be generated by a known distribution. We stress that σ_W does not depend on the field $v(\mathbf{r}, t)$ and hence the noise in equation **??** is strictly additive. The general integral-differential equation is given by

$$\mathrm{d}v(\mathbf{r}, t) + \zeta v(\mathbf{r}, t)\mathrm{d}t = \tilde{g}(\mathbf{r}, t)\mathrm{d}t + \mathrm{d}W(\mathbf{r}, t) \qquad t \geq 0, v(\mathbf{r}, 0) = GP? \tag{7}$$

To show that this is indeed the case consider the function $\kappa(v(\mathbf{r}, t), t) = v(\mathbf{r}, t)e^{\zeta t}$. We note that $\kappa(v(\mathbf{r}, t), t)$ is twice differentiable so that we can apply Ito's formula to obtain

$$\mathrm{d}\kappa = e^{\zeta t} g(\mathbf{r}, t)\mathrm{d}t + e^{\zeta t}\sigma \mathrm{d}W(\mathbf{r}, t) \tag{8}$$

Integrating over $[0, t]$ and multiplying throughout by $e^{-\zeta t}$ gives the required result. The neural field equations must be written as a discrete-time finite-dimensional model in order to relate it to patient-specific data. The discrete-time model is found applying the a first-order Euler-Maruyama method on equation **??** giving

$$v\left(\mathbf{r}, t + T_s\right) - v\left(\mathbf{r}, t\right) + \zeta T_s v\left(\mathbf{r}, t\right) = T_s \int_\Omega w\left(\mathbf{r}, \mathbf{r}'\right) f\left(v\left(\mathbf{r}', t\right)\right) \mathrm{d}\mathbf{r}' + \sigma_W [W\left(\mathbf{r}, t + T_s\right) - W\left(\mathbf{r}, t\right)], \tag{9}$$

where T_s is the time step or sampling period. To simplify the notation, the sample at the the current time step shall be indexed by t and the future time step by $t+1$ for the rest of the paper. Rearranging equation **??** gives the integro-difference equation (IDE) form

$$v_{t+1}\left(\mathbf{r}\right) = \xi v_t\left(\mathbf{r}\right) + T_s \int_\Omega w\left(\mathbf{r}, \mathbf{r}'\right) f\left(v_t\left(\mathbf{r}'\right)\right) \mathrm{d}\mathbf{r}' + e_t\left(\mathbf{r}\right), \tag{10}$$

where $e_t(\mathbf{r}) = \sigma_W [W\left(\mathbf{r}, t + T_s\right) - W\left(\mathbf{r}, t\right)]$ is the increment of a space-time Wiener process, *i.i.d.* with zero spatial mean such that $e_t(\mathbf{r}) \sim \mathcal{GP}(\mathbf{0}, \sigma_d^2 \gamma(\mathbf{r} - \mathbf{r}'))$, where $\sigma_d^2 = T_s \sigma_W^2$. Here $\mathcal{GP}(\mathbf{0}, \sigma_d^2 \gamma(\mathbf{r} - \mathbf{r}'))$ denotes a zero mean spatial Gaussian process with covariance function $\gamma(\mathbf{r} - \mathbf{r}')$ [**?**]. To reduce the model to a finite-dimensional system the neural field is approximated by a basis function decomposition where

$$v_t\left(\mathbf{r}\right) \approx \boldsymbol{\phi}^\top\left(\mathbf{r}\right) \mathbf{x}_t \tag{11}$$

and $\mathbf{x}_t$ is a state vector that weights a vector of two-dimension Gaussian basis functions, $\boldsymbol{\phi}(\mathbf{r})$, described by

$$\boldsymbol{\phi}\left(\mathbf{r} - \mathbf{r}'\right) = \exp\left(-\frac{\left(\mathbf{r} - \mathbf{r}'\right)^\top \left(\mathbf{r} - \mathbf{r}'\right)}{\sigma_\phi^2}\right). \tag{12}$$

The parameter σ_ϕ controls the basis function width and is inferred by frequency analysis. Substituting equation **??** into **??** gives to approximate model

$$\boldsymbol{\phi}^\top(\mathbf{r})\mathbf{x}_{t+1} = T_s \int_\Omega f(\boldsymbol{\phi}^\top(\mathbf{r}')\mathbf{x}_t) w(\mathbf{r},\mathbf{r}')\mathrm{d}\mathbf{r}' + \xi \boldsymbol{\phi}^\top(\mathbf{r}) x_t + e_t(\mathbf{r}). \tag{13}$$

Next we multiply equation **??** by $\boldsymbol{\phi}(\mathbf{r})$ and integrate over the spatial domain, Ω, to get

$$\int_\Omega \boldsymbol{\phi}(\mathbf{r})\boldsymbol{\phi}^\top(\mathbf{r})\,\mathrm{d}\mathbf{r}\mathbf{x}_{t+1} = T_s \int_\Omega \boldsymbol{\phi}(\mathbf{r}) \int_\Omega w(\mathbf{r},\mathbf{r}') f(\boldsymbol{\phi}^\top(\mathbf{r}')\mathbf{x}_t)\mathrm{d}\mathbf{r}'\mathrm{d}\mathbf{r} + \xi \int_\Omega \boldsymbol{\phi}(\mathbf{r})\boldsymbol{\phi}^\top(\mathbf{r})\mathrm{d}\mathbf{r}\mathbf{x}_t + \int_\Omega \boldsymbol{\phi}(\mathbf{r}) e_t(\mathbf{r})\mathrm{d}\mathbf{r}. \tag{14}$$

Now by defining the matrix

$$\boldsymbol{\Gamma} \triangleq \int_\Omega \boldsymbol{\phi}(\mathbf{r})\boldsymbol{\phi}^\top(\mathbf{r})\,\mathrm{d}\mathbf{r} \tag{15}$$

enables the pseudo inverse of $\boldsymbol{\phi}(\mathbf{r})$ to be taken, such that the state vector is isolated on the left-hand-side as

$$\mathbf{x}_{t+1} = T_s \boldsymbol{\Gamma}^{-1} \int_\Omega \boldsymbol{\phi}(\mathbf{r}) \int_\Omega w(\mathbf{r},\mathbf{r}') f(\boldsymbol{\phi}^\top(\mathbf{r}')\mathbf{x}_t)\mathrm{d}\mathbf{r}'\mathrm{d}\mathbf{r} + \xi \mathbf{x}_t + \boldsymbol{\Gamma}^{-1} \int_\Omega \boldsymbol{\phi}(\mathbf{r}) e_t(\mathbf{r})\mathrm{d}\mathbf{r}. \tag{16}$$

A property of the basis function decomposition it the preservation of the Gaussianality of the disturbance term. The decomposition disturbance term is defined as

$$\mathbf{e}_t \triangleq \boldsymbol{\Gamma}^{-1} \int_\Omega \boldsymbol{\phi}(\mathbf{r}) e_t(\mathbf{r})\mathrm{d}\mathbf{r}, \tag{17}$$

where $\mathbf{e}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_e)$. The covariance matrix is defined as (see the Supporting Information, S1, for the derivation)

$$\boldsymbol{\Sigma}_e \triangleq \sigma_d^2 \boldsymbol{\Gamma}^{-1} \int_\Omega \int_\Omega \boldsymbol{\phi}(\mathbf{r})\,\gamma(\mathbf{r}-\mathbf{r}')\,\boldsymbol{\phi}(\mathbf{r}')^\top d\mathbf{r}' d\mathbf{r} \boldsymbol{\Gamma}^{-\top}. \tag{18}$$

To relate the model to the data the electrodes are also modelled. The electrodes have a pick-up range (for LFPs) of approximately 400 μm. Therefore, we model the electrodes as a spatial convolution of a Gaussian shaped sensor kernel (using a width parameters of 400 μm) with the neural field. The observation equation is

$$\mathbf{y}_t = \int_\Omega m(\mathbf{r}_n - \mathbf{r}')\, v_t(\mathbf{r}')\,\mathrm{d}\mathbf{r}' + \boldsymbol{\varepsilon}_t, \tag{19}$$

where $\mathbf{r}_n$ defines the location of the sensors in the field, $n = 1, ..., N$ indexes the sensors and $\boldsymbol{\varepsilon}_t \sim \mathcal{N}(0, \boldsymbol{\Sigma}_\varepsilon)$. $\mathcal{N}(0, \boldsymbol{\Sigma}_\varepsilon)$ denotes the multivariate normal distribution with mean zero and covariance matrix $\boldsymbol{\Sigma}_\varepsilon$. The sensor kernel, $m(\mathbf{r}-\mathbf{r}')$, is defined by

$$m(\mathbf{r}-\mathbf{r}') = \exp\left(-\frac{(\mathbf{r}-\mathbf{r}')^\top(\mathbf{r}-\mathbf{r}')}{\sigma_m^2}\right), \tag{20}$$

where σ_m sets the sensor width. Under the assumption that the basis function decomposition is accurate, the observation equation of the reduced model is found by substituting equation **??** into **??** giving

$$\mathbf{y}_t = \int_\Omega m(\mathbf{r}_n - \mathbf{r}')\,\boldsymbol{\phi}^\top(\mathbf{r}')\,\mathbf{x}_t\mathrm{d}\mathbf{r}' + \boldsymbol{\varepsilon}_t. \tag{21}$$

The observation equation is linear with respect to the state and can be written in the more compact form

$$\mathbf{y}_t = \mathbf{C}\mathbf{x}_t + \boldsymbol{\varepsilon}_t, \tag{22}$$

where the observation matrix is

$$\mathbf{C} = \begin{bmatrix} c_{1,1} & \cdots & c_{1,L} \\ \vdots & \ddots & \vdots \\ c_{N,1} & \cdots & c_{N,L} \end{bmatrix} \tag{23}$$

and

$$c_{i,j} = \int_\Omega m(\mathbf{r}_i - \mathbf{r}')\boldsymbol{\phi}_j(\mathbf{r}')\mathrm{d}\mathbf{r}'. \tag{24}$$

I will do the connectivity kernel decomposition bit after the correlation analysis, because we can use it to say something about heterogeneity and isotropy.

2.2 Correlation analysis

Under the assumption that the sensors are not spatially band-limiting the spectral content of the field and the connectivity kernel is homogeneous, the support for the connectivity kernel can be inferred by studying the spatial cross-correlation between consecutive observations. The deterministic component of the spatial mapping between consecutive fields is due to the convolution of the connectivity kernel with the firing rates. Although the relationship between the field and the firing rate is nonlinear, the cross-correlation analysis still extracts meaningful results. This can be seen by considering the three operating regions of the sigmoid. When the mean membrane voltage is below the active region of the sigmoid, the firing rate will be approximately zero and only very small spatial correlations will observed. Therefore, this will not contribute to the results. Within the active region the sigmoid can be approximated by a linear function between consecutive samples, given a sufficiently high sampling rate. Therefore, linear cross-correlation analysis is appropriate. When the sigmoid is in the saturated region, the correlation coefficients will be higher, but not provide a meaningful measure of the kernel weight or gain. Therefore, we can not estimate the gain for the kernel using cross-correlation analysis, but the spatial extent can still be inferred. The gain for the kernel will be inferred using the EM algorithm described below.

2.2.1 Estimation of Connectivity Kernel Support

For the derivation of the spatial properties estimator of the neural field equations a more compact notation is employed to define convolution and correlation operators. The spatial convolution shall be denoted as

$$\int_\Omega a(\mathbf{r} - \mathbf{r}')b(\mathbf{r}')d\mathbf{r}' = (a * b)(\mathbf{r}), \tag{25}$$

and the spatial cross-correlation shall be denoted as

$$\int_\Omega a(\mathbf{r})b(\mathbf{r} + \boldsymbol{\iota})d\mathbf{r} = (a \star b)(\boldsymbol{\iota}), \tag{26}$$

where ι is the spatial shift. The spatial relationship between consecutive observations is governed by the shape of the connectivity kernel. Therefore, the spatial cross-correlation between consecutive observations is used to estimate the kernel's support and shape. To begin the derivation the spatial cross-correlation between consecutive observations (in time) is defined as

$$R_{y_{t+1}y_t}(\boldsymbol{\iota}) = (y_{t+1} \star y_t)(\boldsymbol{\iota}), \tag{27}$$

The aim of the derivation is to make the necessary substitutions and simplifications to get an expression of the cross-correlation as a function of the connectivity kernel. In the next step equation (**??**) is substituted for $y_{t+1}(\mathbf{r})$ and expanded to give

$$R_{y_{t+1}y_t}(\boldsymbol{\iota}) = ((m * v_{t+1}) \star y_t)(\boldsymbol{\iota}) + (\varepsilon_{t+1} \star y_t)(\boldsymbol{\iota}). \tag{28}$$

Next equation (**??**) is substituted for $v_{t+1}(\mathbf{r})$ giving

$$\begin{aligned} R_{y_{t+1}y_t}(\boldsymbol{\iota}) &= ((m * (\xi v_t + T_s g_t + e_t)) \star y_t)(\boldsymbol{\iota}) \qquad (29) \\ &= \xi\left((m * v_t) \star y_t\right)(\boldsymbol{\iota}) \\ &\quad + T_s\left((m * g_t) \star y_t\right)(\boldsymbol{\iota}) \\ &\quad + ((m * e_t) \star y_t)(\boldsymbol{\iota}) \\ &\quad + (\varepsilon_{t+1} \star y_t)(\boldsymbol{\iota}). \qquad (30) \end{aligned}$$

Now the expectation over time is taken, so that the observation noise and process disturbance terms have minimal effect on the result, giving

$$\mathbf{E}[R_{y_{t+1}y_t}(\boldsymbol{\iota})] = \mathbf{E}[\xi\left((m * v_t) \star y_t\right)(\boldsymbol{\iota})] + T_s\mathbf{E}[\left((m * g_t) \star y_t\right)(\boldsymbol{\iota})], \tag{31}$$

since the disturbance and measurement noise are assumed to be independent of the observations and temporally white. The cross-correlation is further simplified by recognising that the first term on right hand side of equation (**??**) can be written as

$$\begin{aligned}\mathbf{E}[\xi\left((m * v_t) \star y_t\right)(\boldsymbol{\iota})] &= \mathbf{E}\left[\xi\left((y_t - \varepsilon_t) \star y_t\right)(\boldsymbol{\iota})\right] \\ &= \xi\mathbf{E}\left[(y_t \star y_t)(\boldsymbol{\iota}) - (\varepsilon_t \star y_t)(\boldsymbol{\iota})\right] \\ &= \xi\mathbf{E}[R_{y_ty_t}(\boldsymbol{\iota}) - (\varepsilon_t \star (m * v_t + \varepsilon_t))(\boldsymbol{\iota})] \\ &= \xi\mathbf{E}[R_{y_ty_t}(\boldsymbol{\iota}) - (\varepsilon_t \star (m * v_t))(\boldsymbol{\iota}) - (\varepsilon_t \star \varepsilon_t)(\boldsymbol{\iota})] \\ &= \xi\left(\mathbf{E}[R_{y_ty_t}(\boldsymbol{\iota})] - \sigma_\varepsilon^2\delta(\boldsymbol{\iota})\right), \end{aligned} \tag{32}$$

where $\delta(\cdot)$ denotes Kronecker delta. Substituting equation (**??**) back into equation (**??**) gives

$$\mathbf{E}[R_{y_{t+1}y_t}(\boldsymbol{\iota})] = \xi\left(\mathbf{E}[R_{y_ty_t}(\boldsymbol{\iota})] - \sigma_\varepsilon^2\delta(\boldsymbol{\iota})\right) + T_s\mathbf{E}[\left((m * g_t) \star y_t\right)(\boldsymbol{\iota})]. \tag{33}$$

Next we simplify second term in equation (**??**) to get an expression involving the connectivity kernel. By substituting equation (**??**) for g_t we can write

$$T_s((m * g_t) \star y_t)(\boldsymbol{\iota}) = T_s((w * m * f(v_t)) \star y_t)(\boldsymbol{\iota}). \tag{34}$$

The remainder of the derivation is focused on isolating the connectivity kernel. To show how this is done, the activation function is assumed to be in the linear region, i.e

$$f\left(v_t(\mathbf{r})\right) \approx \varsigma v_t(\mathbf{r}). \tag{35}$$

I think we should use this activation function (this will fix the distortion across the $x = 0, y = 0$ axis): The function linearised about the firing threshold is

$$\begin{aligned}\hat{f}(v_t(\mathbf{r})) &= f(v_0) + f'(v_0)(v_t(\mathbf{r}) - v_0) \qquad (36)\\ &= \frac{2 + \varsigma(v_t(\mathbf{r}) - v_0)}{4}. \qquad (37)\end{aligned}$$

Substituting the linear activation function back into equation (**??**) yields

$$T_s((m * g_t) \star y_t)(\boldsymbol{\iota}) \approx T_s\varsigma((w * m * v_t) \star y_t)(\boldsymbol{\iota}). \tag{38}$$

Now by substituting $y_t - \varepsilon_t$ in for $m * v_t$ equation (**??**) can be written as

$$T_s((m * g_t) \star y_t)(\boldsymbol{\iota}) \approx T_s\varsigma(w * (y_t - \varepsilon_t)) \star y_t(\boldsymbol{\iota}) \tag{39}$$

To isolate the kernel, the order of the convolution and cross-correlation is reversed by recognising that a property $(a * b)(\boldsymbol{\iota}) \star c(\boldsymbol{\iota}) = a(-\boldsymbol{\iota}) * (b \star c)(\boldsymbol{\iota})$ **(see Appendix for proof)**. Therefore,

$$T_s((m * g_t) \star y_t)(\boldsymbol{\iota}) \approx T_s\varsigma w(-\boldsymbol{\iota}) * ((y_t - \varepsilon_t)) \star y_t)(\boldsymbol{\iota}), \tag{40}$$

Now using the identity established to get equation (**??**) gives

$$T_s\mathbf{E}\left[((m * g_t) \star y_t)(\boldsymbol{\iota})\right] \approx T_s\varsigma w(-\boldsymbol{\iota}) * \left(\mathbf{E}\left[R_{y_ty_t}(\boldsymbol{\iota})\right] - \sigma_\varepsilon^2\delta(\boldsymbol{\iota})\right) \tag{41}$$

Now substituting back equation (**??**) into equation (**??**) gives

$$\begin{aligned}\mathbf{E}[R_{y_{t+1}y_t}(\boldsymbol{\iota})] = \xi&\left(\mathbf{E}[R_{y_ty_t}(\boldsymbol{\iota})] - \sigma_\varepsilon^2\delta(\boldsymbol{\iota})\right) \\ &+ T_s\varsigma w(-\boldsymbol{\iota}) * \left(\mathbf{E}\left[R_{y_ty_t}(\boldsymbol{\iota})\right] - \sigma_\varepsilon^2\delta(\boldsymbol{\iota})\right)\end{aligned} \tag{42}$$

Rearranging equation (**??**) gives

$$w(-\boldsymbol{\iota}) * \left(\mathbf{E}\left[R_{y_t y_t}(\boldsymbol{\iota})\right] - \sigma_\varepsilon^2 \delta(\boldsymbol{\iota})\right) = \frac{1}{T_s \varsigma}\mathbf{E}[R_{y_{t+1} y_t}(\boldsymbol{\iota})] - \frac{\xi}{T_s \varsigma}\big(\mathbf{E}[R_{y_t y_t}(\boldsymbol{\iota})] - \sigma_\varepsilon^2 \delta(\boldsymbol{\iota})\big) \tag{43}$$

The solution of the above equation for the connectivity kernel is a deconvolution. This can be approached from a number of different standpoints. The simplest solution is to use the convolution theorem and to solve for the kernel algebraically in the frequency domain.

$$\begin{aligned}\mathcal{F}\{w(-\boldsymbol{\iota})\}\mathcal{F}\{(\mathbf{E}\left[R_{y_t y_t}(\boldsymbol{\iota})\right] - \sigma_\varepsilon^2 \delta(\boldsymbol{\iota}))\} \approx \frac{1}{T_s \varsigma}&\mathcal{F}\{\mathbf{E}[R_{y_{t+1} y_t}(\boldsymbol{\iota})]\} \\ &- \frac{\xi}{T_s \varsigma}\mathcal{F}\{\big(\mathbf{E}[R_{y_t y_t}(\boldsymbol{\iota})] - \sigma_\varepsilon^2 \delta(\boldsymbol{\iota})\big)\}.\end{aligned} \tag{44}$$

Now rearranging (**??**) gives

$$\mathcal{F}\left(w(-\boldsymbol{\iota})\right) = \frac{1}{T_s \varsigma}\left[\frac{\mathcal{F}\{\mathbf{E}[R_{y_{t+1} y_t}(\boldsymbol{\iota})]\}}{\mathcal{F}\{(\mathbf{E}\left[R_{y_t y_t}(\boldsymbol{\iota})\right] - \sigma_\varepsilon^2 \delta(\boldsymbol{\iota}))\}} - \xi\right]. \tag{45}$$

From (**??**) it can be observed that an error in the initial estimate of ξ will result in a dilation or contraction of the estimated kernel support and an incorrect guess of the observation noise will result in a distortion in the zero frequency (offset) component of the kernel, however if the signal-to-noise ratio is high such a distortion becomes insignificant.

The denominator of equation (**??**) is equivalent to the power spectral density (PSD) of the noise-free observations. By the Weiner-Khintchine theorem (WKT) the PSD is non-negative and a real quantity. The WKT states that the power spectral density of the wide sense stationary process is the Fourier transform of its autocorrelation function [**?**]. **The WKT allows us establish the bounds**

$$\mathcal{F}\{(\mathbf{E}\left[R_{y_t y_t}(\boldsymbol{\iota})\right] - \sigma_\varepsilon^2 \delta(\boldsymbol{\iota}))\} \geq 0 \tag{46}$$

$$\implies \mathcal{F}\{(\mathbf{E}\left[R_{y_t y_t}(\boldsymbol{\iota})\right]\} \geq \mathcal{F}\{\sigma_\varepsilon^2 \delta(\boldsymbol{\iota}))\} \geq 0 \tag{47}$$

$$\implies \min_f \mathcal{F}\{(\mathbf{E}\left[R_{y_t y_t}(\boldsymbol{\iota})\right]\} \geq \sigma_\varepsilon^2 \geq 0. \tag{48}$$

The numerator is also known as the cross-spectrum or cross-spectral density.

Since the connectivity kernel is a real function, the following relation holds [**?**]

$$\mathcal{F}\left(w(\boldsymbol{\iota})\right) = \overline{\mathcal{F}\left(w(-\boldsymbol{\iota})\right)}, \tag{49}$$

where over-bar denotes complex conjugate operator. Finally, an expression for the kernel is obtained by taking the inverse Fourier transform

$$w(\boldsymbol{\iota}) = \frac{1}{T_s \varsigma}\mathcal{F}^{-1}\overline{\left\{\frac{\mathcal{F}\{\mathbf{E}[R_{y_{t+1} y_t}(\boldsymbol{\iota})]\}}{\mathcal{F}\{(\mathbf{E}\left[R_{y_t y_t}(\boldsymbol{\iota})\right]\} - \sigma_\varepsilon^2} - \xi\right\}}. \tag{50}$$

The complex conjugate operator essentially reflects the connectivity kernel through the origin in the spatial domain, therefore, if the connectivity kernel is isotropic, the complex conjugate operator can be dropped.

2.2.2 Estimation of Disturbance Covariance Support

The observation auto-correlation at time $t+1$ can be expressed as

$$R_{y_{t+1} y_{t+1}}(\boldsymbol{\iota}) = (y_{t+1} \star y_{t+1})(\boldsymbol{\iota}). \tag{51}$$

Substituting equation (**??**) in for y_{t+1} and expanding gives

$$\begin{aligned} R_{y_{t+1}y_{t+1}}(\boldsymbol{\iota}) = \xi((m * v_t) \star y_{t+1})(\boldsymbol{\iota}) + T_s((m * g_t) \star y_{t+1})(\boldsymbol{\iota}) \\ + ((m * e_t) \star y_{t+1})(\boldsymbol{\iota}) + (\varepsilon_{t+1} \star y_{t+1})(\boldsymbol{\iota}). \end{aligned} \tag{52}$$

By using similar arguments that were used in the derivation for the connectivity kernel support, the auto-correlation can be simplified by recognising that

$$\mathbf{E}[\xi((m * v_t) \star y_{t+1})(\boldsymbol{\iota})] = \xi \mathbf{E}[R_{y_t y_{t+1}}(\boldsymbol{\iota})], \tag{53}$$

and

$$\begin{aligned} T_s \mathbf{E}[((m * g_t) \star y_{t+1})(\boldsymbol{\iota})] &\approx T_s \varsigma \mathbf{E}[((m * w) \star y_{t+1})(\boldsymbol{\iota})] \\ &= T_s \varsigma w(-\boldsymbol{\iota}) * (\mathbf{E}\left[R_{y_t y_{t+1}}(\boldsymbol{\iota})\right]). \end{aligned} \tag{54}$$

$$\mathbf{E}[(\varepsilon_{t+1} \star y_{t+1})(\boldsymbol{\iota})] = \sigma_\varepsilon^2 \delta(\boldsymbol{\iota}), \tag{55}$$

Substituting equations (**??**), (**??**), and (**??**) back into equation (**??**) gives

$$\begin{aligned} \mathbf{E}[R_{y_{t+1}y_{t+1}}(\boldsymbol{\iota})] = \xi \mathbf{E}[R_{y_t y_{t+1}}(\boldsymbol{\iota})] + T_s \varsigma w(-\boldsymbol{\iota}) * (\mathbf{E}\left[R_{y_t y_{t+1}}(\boldsymbol{\iota})\right]) \\ + \mathbf{E}[((m * e_t) \star y_{t+1})(\boldsymbol{\iota})] + \sigma_\epsilon^2 \delta(\boldsymbol{\iota}). \end{aligned} \tag{56}$$

The third term in (**??**) can be simplified as

$$\begin{aligned} \mathbf{E}[((m * e_t) \star y_{t+1})(\boldsymbol{\iota})] &= \mathbf{E}[((m * e_t) \star (m * v_{t+1} + \varepsilon_{t+1}))(\boldsymbol{\iota})] \\ &= \mathbf{E}[((m * e_t) \star (m * [\xi v_t + T_s g_t + e_t] + \varepsilon_{t+1}))(\boldsymbol{\iota})] \\ &= \mathbf{E}[(m * e_t) \star (m * e_t)\,(\boldsymbol{\iota})] \end{aligned} \tag{57}$$

A property of cross-correlation and convolution is $(a*b)(\boldsymbol{\tau}) \star (a*b)(\boldsymbol{\tau}) = (a \star a)(\boldsymbol{\tau}) * (b \star b)(\boldsymbol{\tau})$ **(see Appendix II for a proof)**, using the isotropy property of the observation kernel we can write

$$\begin{aligned} \mathbf{E}[((m * e_t) \star (m * e_t))(\boldsymbol{\iota})] &= \mathbf{E}[((m \star m) * (e_t \star e_t))(\boldsymbol{\iota})] \\ &= (m * m * \gamma)(\boldsymbol{\iota}). \end{aligned} \tag{58}$$

Substituting this back into equation (**??**) gives

$$\begin{aligned} \mathbf{E}[R_{y_{t+1}y_{t+1}}(\boldsymbol{\iota})] = \xi \mathbf{E}[R_{y_t y_{t+1}}(\boldsymbol{\iota})] + T_s \varsigma w(-\boldsymbol{\iota}) * (\mathbf{E}\left[R_{y_t y_{t+1}}(\boldsymbol{\iota})\right]) \\ + (m * m * \gamma)(\boldsymbol{\iota}) + \sigma_\epsilon^2 \delta(\boldsymbol{\iota}). \end{aligned} \tag{59}$$

To solve for the disturbance covariance, again the convolution theorem is applied to perform the deconvolution algebraically. Taking the Fourier transform and rearranging gives

$$\begin{aligned} \mathcal{F}\{(m * m * \gamma)(\boldsymbol{\iota})\} = \mathcal{F}\{\mathbf{E}[R_{y_{t+1}y_{t+1}}(\boldsymbol{\iota})]\} - \xi \mathcal{F}\{\mathbf{E}[R_{y_t y_{t+1}}(\boldsymbol{\iota})]\} \\ - T_s \varsigma \mathcal{F}\{w(-\boldsymbol{\iota})\} \mathcal{F}\left\{\mathbf{E}\left[R_{y_t y_{t+1}}(\boldsymbol{\iota})\right]\right\} - \sigma_\varepsilon^2 \end{aligned} \tag{60}$$

Now substituting in equation (**??**) for $\mathcal{F}\{w(-\boldsymbol{\iota})\}$ results in

$$\begin{aligned} \mathcal{F}\{(m * m * \gamma)(\boldsymbol{\iota})\} = \mathcal{F}\{\mathbf{E}[R_{y_{t+1}y_{t+1}}(\boldsymbol{\iota})]\} - \xi \mathcal{F}\{\mathbf{E}[R_{y_t y_{t+1}}(\boldsymbol{\iota})]\} \\ - T_s \varsigma \times \frac{1}{T_s \varsigma} \left[\frac{\mathcal{F}\{\mathbf{E}[R_{y_{t+1} y_t}(\boldsymbol{\iota})]\}}{\mathcal{F}\{(\mathbf{E}\left[R_{y_t y_t}(\boldsymbol{\iota})\right] - \sigma_\varepsilon^2 \delta(\boldsymbol{\iota}))\}} - \xi \right] \mathcal{F}\left\{\mathbf{E}\left[R_{y_t y_{t+1}}(\boldsymbol{\iota})\right]\right\} - \sigma_\varepsilon^2 \end{aligned} \tag{61}$$

Now simplifying gives

$$\mathcal{F}\{m(\boldsymbol{\iota})\}\mathcal{F}\{m(\boldsymbol{\iota})\}\mathcal{F}\{\gamma(\boldsymbol{\iota})\} = \\ \mathcal{F}\{\mathbf{E}[R_{y_{t+1}y_{t+1}}(\boldsymbol{\iota})]\} - \left[\frac{\mathcal{F}\{\mathbf{E}[R_{y_{t+1}y_t}(\boldsymbol{\iota})]\}\mathcal{F}\left\{\mathbf{E}\left[R_{y_ty_{t+1}}(\boldsymbol{\iota})\right]\right\}}{\mathcal{F}\{(\mathbf{E}\left[R_{y_ty_t}(\boldsymbol{\iota})\right]\} - \sigma_\varepsilon^2}\right] - \sigma_\varepsilon^2, \tag{62}$$

rearranging and taking inverse Fourier transform gives the final result

$$\gamma(\boldsymbol{\iota}) = \\ \mathcal{F}^{-1}\left\{\frac{1}{\tilde{m}(\boldsymbol{\iota})}\left[\left[\mathcal{F}\{\mathbf{E}[R_{y_{t+1}y_{t+1}}(\boldsymbol{\iota})]\} - \sigma_\varepsilon^2\right] - \left[\frac{\mathcal{F}\{\mathbf{E}[R_{y_{t+1}y_t}(\boldsymbol{\iota})]\}\mathcal{F}\left\{\mathbf{E}\left[R_{y_ty_{t+1}}(\boldsymbol{\iota})\right]\right\}}{\mathcal{F}\{(\mathbf{E}\left[R_{y_ty_t}(\boldsymbol{\iota})\right]\} - \sigma_\varepsilon^2}\right]\right]\right\}, \tag{63}$$

where

$$\tilde{m}(\boldsymbol{\iota}) = \mathcal{F}\{m(\boldsymbol{\iota})\}\mathcal{F}\{m(\boldsymbol{\iota})\}. \tag{64}$$

From (**??**), to calculate the exact shape of the disturbance covariance function, observation noise variance and the sensor kernel support are required. Again if the signal to noise ratio is high the effect of σ_ε would be small. Note the first bracket and the denominator in (**??**) are non-negative power spectral denisties of the noise free observations at time $t+1$ and t respectively. If m is a point sensor then the spatial support of γ can be calculated directly, when considering a sensor with a spatial extent, dividing by $\tilde{m}(\boldsymbol{\iota})$ in (**??**) is then required. Alternatively, assuming a Gaussian shape for disturbance function and sensor kernel (with known support), the support of the disturbance covariance function, σ_γ^2 can be determined analytically using the expression for Gaussian basis functions' convolution, giving

$$\sigma_\gamma^2 = \sigma_{mm\gamma}^2 - 2\sigma_m^2. \tag{65}$$

where $\sigma_{mm\gamma}^2$ is the width of fitted Gaussian to the estimated support of $(m * m * \gamma)(\boldsymbol{\iota})$.

2.3 Parametrisation of Connectivity Kernel

The connectivity kernel can be decomposed as

$$w\left(\mathbf{r},\mathbf{r}'\right) \approx \boldsymbol{\psi}^\top\left(\mathbf{r},\mathbf{r}'\right)\boldsymbol{\theta}, \tag{66}$$

where $\boldsymbol{\psi}(\mathbf{r},\mathbf{r}')$ is a vector of Gaussian basis functions and $\boldsymbol{\theta}$ is a vector of scaling parameters. By assuming a Gaussian isotropic connectivity structure, the kernel basis functions can be written as $\boldsymbol{\psi}(\mathbf{r}-\mathbf{r}')$. We will assume that we know the parametric form of the connectivity basis functions from the correlation analysis, where the scaling parameters $\boldsymbol{\theta}$ are unknown. Substituting equation **??** into equation**??** gives

$$\mathbf{x}_{t+1} = T_s\boldsymbol{\Gamma}^{-1}\int_\Omega \boldsymbol{\phi}(\mathbf{r})\int_\Omega \boldsymbol{\psi}^\top\left(\mathbf{r},\mathbf{r}'\right)\boldsymbol{\theta} f(\boldsymbol{\phi}^\top(\mathbf{r}')\mathbf{x}_t)\mathrm{d}\mathbf{r}'\mathrm{d}\mathbf{r} + \xi\mathbf{x}_t + \mathbf{e}_t(\mathbf{r}). \tag{67}$$

The connectivity kernel weights can be shifted outside the integral giving

$$\mathbf{x}_{t+1} = T_s\boldsymbol{\Gamma}^{-1}\int_\Omega \boldsymbol{\phi}(\mathbf{r})\int_\Omega \boldsymbol{\psi}^\top(\mathbf{r}-\mathbf{r}') f(\boldsymbol{\phi}^\top(\mathbf{r}')\mathbf{x}_t)\mathrm{d}\mathbf{r}'\mathrm{d}\mathbf{r}\boldsymbol{\theta} + \xi\mathbf{x}_t + \mathbf{e}_t(\mathbf{r}). \tag{68}$$

Now taking the basis function vector inside the inner integrand gives

$$\mathbf{x}_{t+1} = T_s\boldsymbol{\Gamma}^{-1}\int_\Omega\int_\Omega \boldsymbol{\phi}(\mathbf{r})\boldsymbol{\psi}^\top(\mathbf{r}-\mathbf{r}') f(\boldsymbol{\phi}^\top(\mathbf{r}')\mathbf{x}_t)\mathrm{d}\mathbf{r}'\mathrm{d}\mathbf{r}\boldsymbol{\theta} + \xi\mathbf{x}_t + \mathbf{e}_t(\mathbf{r}). \tag{69}$$

This equation can be simplified by exploiting the isotropy of the basis functions used to represent the kernel, and hence,

$$\psi_i(\mathbf{r}-\mathbf{r}') = \psi_i(2c_i + \mathbf{r}' - \mathbf{r}). \tag{70}$$

where c_i is the center of the i^{th} basis function of the kernel decomposition. To make the simplification, we define

$$[\mathbf{\Psi}(\mathbf{r}')]_{:i} \triangleq T_s \mathbf{\Gamma}^{-1} \int_\Omega \boldsymbol{\phi}(\mathbf{r})\psi_i(2c_i + \mathbf{r}' - \mathbf{r})\mathrm{d}\mathbf{r}, \tag{71}$$

where $[\mathbf{\Psi}(\mathbf{r}')]_{:i}$ denotes the i^{th}column of $\mathbf{\Psi}(\mathbf{r}')$ which is a constant $L \times n_\theta$ matrix that is defined analytically, where L is the number of basis functions (and states) and n_θ is the number of connectivity kernel basis functions. Substituting equation **??** into **??** gives

$$\mathbf{x}_{t+1} = \int_\Omega \mathbf{\Psi}(\mathbf{r}') f(\boldsymbol{\phi}^\top(\mathbf{r}')\mathbf{x}_t)\mathrm{d}\mathbf{r}'\boldsymbol{\theta} + \xi\mathbf{x}_t + \mathbf{e}_t(\mathbf{r}). \tag{72}$$

This substitution exploits the isotropy of the connectivity kernel basis functions, allowing us to swap the convolution in equation **??**, which is computationally demanding, with the convolution in equation **??**. This provides a dramatic increase in estimation speed, since the matrix $\mathbf{\Psi}(\mathbf{r}')$ can be calculated analytically. This is of great importance as it lowers the computational demands of the estimation algorithm.

As a result of the model reduction via the basis function decomposition and the decomposition of the connectivity kernel the state space model is formed

$$\mathbf{x}_{t+1} = q(\mathbf{x}_t)\boldsymbol{\theta} + \xi\mathbf{x}_t + \boldsymbol{e}_t(\mathbf{r}) \tag{73}$$

$$\mathbf{y}_t = \mathbf{C}\mathbf{x}_t + \boldsymbol{\epsilon}_t, \tag{74}$$

where the nonlinear function, $q(\cdot)$, is defined as

$$q(\mathbf{x}_t) = \int_\Omega \mathbf{\Psi}(\mathbf{r}') f(\boldsymbol{\phi}^\top(\mathbf{r}')\mathbf{x}_t) d\mathbf{r}'. \tag{75}$$

3 Estimation Algorithm

Here we describe the problem of estimating non-linear state space neural field model using expectation-maximisation (EM) based algorithm [**?**, **?**] is described. EM is an iterative algorithm that converges on the maximum-likelihood estimates of the parameters. Both the E and the M steps of the EM algorithm increase a lower bound on the log-likelihood function of the parameters given the observed field. In the E-step we estimate the state $\mathbf{x}_t$, of the system given the observed field and current estimate of the model parameters to calculate expected log-likelihood. In the M-step we estimate the system parameters by maximising the expected log-likelihood computed in the E-step. The resulting parameters are then used to estimate the distribution of the state vector sequence for the next iteration. The stopping criterion is usually associated with either the change in the parameter estimates or the log-likelihood variation [**?**].

3.1 E-Step

In state space models, E-step corresponds to solving the smoothing problem, i.e. estimating the states of the system, $\mathbf{x}_t$ given all observations and the parameters of the system. The smoother output is then used to calculate the expected log-likelihood function required in M-step. The non-linear behaviour of the state-space neural field described in (**??**) requires a non-linear smoother to be employed for state estimation in the E-step.

3.1.1 State estimation

The additive form of the unscented Rauch-Tung-Striebel smoother (URTSS) [**?**] was used for the state estimation. The URTSS incorporates an unscented Kalman filter (UKF) [**?**, **?**] in a forward step to estimate posterior states, $\hat{\mathbf{x}}_t^f$, followed by a backward step to compute the smoothed state estimates, $\hat{\mathbf{x}}_t^b$. The first and second order moments of the predicted state are captured by propagating the so-called sigma points through the state equation. The sigma points, $\mathcal{X}_i$, are calculated using the unscented transform as follows:

$$\mathcal{X}_0 = \bar{\mathbf{x}} \tag{76}$$

$$\mathcal{X}_i = \bar{\mathbf{x}} + \left(\sqrt{(n_x+\lambda)\mathbf{P}_x}\right)_i, \quad i = 1, \ldots, n_x \tag{77}$$

$$\mathcal{X}_i = \bar{\mathbf{x}} - \left(\sqrt{(n_x+\lambda)\mathbf{P}_x}\right)_{i-n_x}, \quad i = n_x+1, \ldots, 2n_x \tag{78}$$

where $\bar{\mathbf{x}}$ represents either $\hat{\mathbf{x}}_t^f$ or $\hat{\mathbf{x}}_t^b$, $\mathbf{P}_x$ is the corresponding covariance matrix for filtering or smoothing, $\left(\sqrt{(n_x+\lambda)\mathbf{P}_x}\right)_i$ is the i^{th} column of the scaled matrix square root of $\mathbf{P}_x$, and n_x is the dimension of the state space. The total number of sigma points is $2n_x+1$. The scaling parameter, λ, is defined as

$$\lambda = \alpha^2(n_x+\kappa) - n_x, \tag{79}$$

The positive constant α, that determines the spread of the sigma points around $\bar{\mathbf{x}}$, was set to 10^{-3}. It is typically set arbitrary small to minimise higher-order effects [?]. The other scaling parameter, κ, was set to $3-n_x$ in accordance to [?].

The sigma vectors are propagated through the system equations and weighted to form the predicted mean and covariance. The weights are calculated by

$$\mathbf{W}_0^{(m)} = \frac{\lambda}{n_x+\lambda} \tag{80}$$

$$\mathbf{W}_0^{(c)} = \frac{\lambda}{n_x+\lambda} + (1-\alpha^2+\beta) \tag{81}$$

$$\mathbf{W}_i^{(m)} = \mathbf{W}_i^{(c)} = \frac{1}{2(n_x+\lambda)} \quad i = 1, \ldots, 2n_x, \tag{82}$$

where the superscripts m and c denote mean and covariance and β incorporates prior knowledge of the distribution of the states, $\mathbf{x}$ (for a Gaussian disturbance, β should be set to 2 [?]). Since the observation equation is linear (equation **??**), the standard Kalman Filter update equations are used to correct the predicted states. The state estimates from forward filtering are used to form a new set of sigma points for the smoother, as described above. The predicted and the smoothed cross-covariance matrices of the states, $\mathbf{M}_t^{b-}$ and $\mathbf{M}_t^b$, are also required. The former is used for computing the smoother gain and the latter is used to compute expected log-likelihood function. The formulation of the cross-covariance matrices are given in the summary of the steps in the URTSS algorithm in Table **??**.

3.1.2 Expected log-likelihood

It is well known that the expected joint data log likelihood $\ln p(\mathbf{X},\mathbf{Y};\boldsymbol{\Theta})$, forms a lower bound on the log-likelihood function $\ln p(\mathbf{Y};\boldsymbol{\Theta})$, where $\mathbf{X}$ and $\mathbf{Y}$ are set of all states and the set of all observations respectively and $\boldsymbol{\Theta}$ denotes the set of all system parameters, i.e.

$$\boldsymbol{\Theta} = \{\boldsymbol{\theta}, \xi, \sigma_d, \sigma_\epsilon\} \tag{83}$$

Therefore we can write

$$\mathcal{Q}(\boldsymbol{\Theta},\boldsymbol{\Theta}') = \mathbf{E}_{\boldsymbol{\Theta}'}\left[2\ln p(\mathbf{X},\mathbf{Y};\boldsymbol{\Theta})\right], \tag{84}$$

where $\mathbf{E}_{\boldsymbol{\Theta}'}[.]$ denotes expectation taken with respect to the distribution $p(\mathbf{X} \mid \mathbf{Y};\boldsymbol{\Theta}')$ which is marginal distribution of the state estimates conditioned on observed field and current estimate of the parameter set $\boldsymbol{\Theta}'$, i.e. smoother output. The factor of 2 is included to simplify the derivation in the later steps. We proceed by expanding the joint distribution $p(\mathbf{X},\mathbf{Y};\boldsymbol{\Theta})$ as

$$p(\mathbf{X},\mathbf{Y};\boldsymbol{\Theta}) = \prod_{t=0}^{T-1} p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1};\sigma_\epsilon)p(\mathbf{x}_{t+1}|\mathbf{x}_t;\boldsymbol{\theta},\xi,\sigma_d)p(\mathbf{x}_0). \tag{85}$$

Substituting equations (**??**) in (**??**) gives

$$\mathcal{Q}(\boldsymbol{\Theta}, \boldsymbol{\Theta}') = \mathbf{E}_{\boldsymbol{\Theta}'} \left[\sum_{t=0}^{T-1} 2 \ln p(\mathbf{y}_{t+1} | \mathbf{x}_{t+1}; \sigma_\epsilon) + \sum_{t=0}^{T-1} 2 \ln p(\mathbf{x}_{t+1} | \mathbf{x}_t; \boldsymbol{\theta}, \xi, \sigma_d) \right. \\ \left. + \sum_{t=0}^{T-1} 2 \ln p(\mathbf{x}_0) \right]. \tag{86}$$

The last term on the right-hand side of (**??**) can be omitted as it does not depend on the parameter set, therefore $\mathcal{Q}$-function can be rewritten as

$$\mathcal{Q}(\boldsymbol{\Theta}, \boldsymbol{\Theta}') = \mathbf{E}_{\Theta'} \left[\sum_{t=0}^{T-1} 2 \ln p(\mathbf{y}_{t+1} | \mathbf{x}_{t+1}; \sigma_\epsilon) + \sum_{t=0}^{T-1} 2 \ln p(\mathbf{x}_{t+1} | \mathbf{x}_t; \boldsymbol{\theta}, \xi, \sigma_d) \right]. \tag{87}$$

The distribution of the initial states is assumed to be Gaussian and known. Therefore, the last term on the right-hand side of (**??**) can be omitted as it does not depend on the parameter set, $\mathcal{Q}$-function can be then rewritten as

$$\mathcal{Q}(\boldsymbol{\Theta}, \boldsymbol{\Theta}') = \mathbf{E}_{\Theta'} \left[\sum_{t=0}^{T-1} 2 \ln p(\mathbf{y}_{t+1} | \mathbf{x}_{t+1}; \sigma_\epsilon) + \sum_{t=0}^{T-1} 2 \ln p(\mathbf{x}_{t+1} | \mathbf{x}_t; \boldsymbol{\theta}, \xi, \sigma_d) \right]. \tag{88}$$

Gaussian distributions of the field disturbance and the observation noise result into the following conditional distributions, where the constant normalising factors are dropped as multiplying by a constant does not alter the location of the maximum.

$$p\left(\mathbf{y}_{t+1} | \mathbf{x}_{t+1}; \sigma_\epsilon\right) = \mid \boldsymbol{\Sigma}_\epsilon \mid^{-\frac{1}{2}} \exp\left(-\frac{1}{2} \left(\mathbf{y}_{t+1} - \mathbf{C}\mathbf{x}_{t+1}\right)^\top \boldsymbol{\Sigma}_\epsilon^{-1} \left(\mathbf{y}_{t+1} - \mathbf{C}\mathbf{x}_{t+1}\right) \right), \tag{89}$$

$$p(\mathbf{x}_{t+1} | \mathbf{x}_t; \boldsymbol{\theta}, \xi, \sigma_d) = \mid \boldsymbol{\Sigma}_e \mid^{-\frac{1}{2}} \\ \times \exp\left(-\frac{1}{2} (\mathbf{x}_{t+1} - \mathbf{q}(\mathbf{x}_t)\boldsymbol{\theta} - \xi \mathbf{x}_t)^\top \boldsymbol{\Sigma}_e^{-1} (\mathbf{x}_{t+1} - \mathbf{q}(\mathbf{x}_t)\boldsymbol{\theta} - \xi \mathbf{x}_t) \right). \tag{90}$$

Note the disturbance covariance matrix, $\boldsymbol{\Sigma}_e$, can be written as a product of the unknown parameter, σ_d^2, with the constant matrix, $\tilde{\boldsymbol{\Sigma}}_e$, where the constant part can be calculated from (**??**) using the estimated support of the disturbance covariance function (see (**??**)). This simplifies the M-step significantly as the estimation of the entire covariance matrix reduces to the estimation of the single scalar σ_d^2. By expanding the exponent in (**??**) and (**??**), twice the logarithm of the results give

$$2 \ln p\left(\mathbf{y}_{t+1} | \mathbf{x}_{t+1}; \sigma_\epsilon\right) = -n_y \ln \sigma_\epsilon^2 - \frac{1}{\sigma_\epsilon^2} \left[(\mathbf{y}_{t+1} - \mathbf{C}\mathbf{x}_{t+1})^\top (\mathbf{y}_{t+1} - \mathbf{C}\mathbf{x}_{t+1}) \right], \tag{91}$$

$$2 \ln p(\mathbf{x}_{t+1} | \mathbf{x}_t; \boldsymbol{\theta}, \xi, \sigma_d) = -n_x \ln \sigma_d^2 - \ln \mid \tilde{\boldsymbol{\Sigma}}_e \mid - \frac{1}{\sigma_d^2} \mathbf{x}_{t+1}^\top \tilde{\boldsymbol{\Sigma}}_e^{-1} \mathbf{x}_{t+1} - \frac{\xi^2}{\sigma_d^2} \mathbf{x}_t^\top \tilde{\boldsymbol{\Sigma}}_e^{-1} \mathbf{x}_t + \\ \frac{2\xi}{\sigma_d^2} \mathbf{x}_{t+1}^\top \tilde{\boldsymbol{\Sigma}}_e^{-1} \mathbf{x}_t - \frac{1}{\sigma_e^2} \boldsymbol{\theta}^\top \mathbf{q}^\top(\mathbf{x}_t) \tilde{\boldsymbol{\Sigma}}_e^{-1} \mathbf{q}(\mathbf{x}_t) \boldsymbol{\theta} - \frac{2\xi}{\sigma_d^2} \mathbf{x}_t^\top \tilde{\boldsymbol{\Sigma}}_e^{-1} \mathbf{q}(\mathbf{x}_t) \boldsymbol{\theta} + \frac{2}{\sigma_d^2} \mathbf{x}_{t+1}^\top \tilde{\boldsymbol{\Sigma}}_e^{-1} \mathbf{q}(\mathbf{x}_t) \boldsymbol{\theta}. \tag{92}$$

In the next step, the above equations are rearranged to isolate the covariance and cross-covariance terms such that the smoother outputs can be used. In order to permute the matrices, the scalar nature of the likelihood is exploited, this way the trace operator can be applied to express the likelihood in a more compact form. Taking trace and rearranging, applying the invariant cyclic permutations property of the trace, equations

(**??**) and (**??**) become

$$
\begin{aligned}
&2\ln p\left(\mathbf{y}_{t+1}|\mathbf{x}_{t+1};\sigma_\epsilon\right) = -n_y \ln\sigma_\epsilon^2 - \frac{1}{\sigma_\epsilon^2}\mathrm{tr}\left\{(\mathbf{y}_{t+1}-\mathbf{C}\mathbf{x}_{t+1})(\mathbf{y}_{t+1}-\mathbf{C}\mathbf{x}_{t+1})^\top\right\} && (93)\\
&2\ln p(\mathbf{x}_{t+1}|\mathbf{x}_t;\boldsymbol{\theta},\xi,\sigma_d) = -n_x\ln\sigma_d^2 - \frac{1}{\sigma_d^2}\mathrm{tr}\left\{\mathbf{x}_{t+1}\mathbf{x}_{t+1}^\top\tilde{\boldsymbol{\Sigma}}_e^{-1}\right\} - \frac{\xi^2}{\sigma_d^2}\mathrm{tr}\left\{\mathbf{x}_t\mathbf{x}_t^\top\tilde{\boldsymbol{\Sigma}}_e^{-1}\right\}\\
&+\frac{2\xi}{\sigma_d^2}\mathrm{tr}\left\{\mathbf{x}_t\mathbf{x}_{t+1}^\top\tilde{\boldsymbol{\Sigma}}_e^{-1}\right\} - \frac{1}{\sigma_d^2}\boldsymbol{\theta}^\top\mathbf{q}^\top(\mathbf{x}_t)\tilde{\boldsymbol{\Sigma}}_e^{-1}\mathbf{q}(\mathbf{x}_t)\boldsymbol{\theta} - \frac{2\xi}{\sigma_d^2}\mathbf{x}_t^\top\tilde{\boldsymbol{\Sigma}}_e^{-1}\mathbf{q}(\mathbf{x}_t)\boldsymbol{\theta}\\
&+\frac{2}{\sigma_d^2}\mathbf{x}_{t+1}^\top\tilde{\boldsymbol{\Sigma}}_e^{-1}\mathbf{q}(\mathbf{x}_t)\boldsymbol{\theta}. && (94)
\end{aligned}
$$

Substituting (**??**) and (**??**) back into (**??**) and taking the expectation of the log-likelihood function at all time instants gives the required lower-bound. Note that the expectation distributes over the trace sum.

$$
\begin{aligned}
\mathcal{Q}(\boldsymbol{\Theta},\boldsymbol{\Theta}') &= -Tn_y\ln\sigma_\epsilon^2 - Tn_x\ln\sigma_d^2\\
&-\frac{1}{\sigma_\epsilon^2}\mathrm{tr}\left\{\sum\nolimits_{t=0}^{T-1}\left[(\mathbf{y}_{t+1}-\mathbf{C}\hat{\mathbf{x}}_{t+1}^b)(\mathbf{y}_{t+1}-\mathbf{C}\hat{\mathbf{x}}_{t+1}^b)^\top + \mathbf{C}\mathbf{P}_{t+1}^b\mathbf{C}^\top\right]\right\}\\
&-\frac{1}{\sigma_d^2}\mathrm{tr}\left\{\mathbf{E}_{\Theta'}\left[\sum_{t=0}^{T-1}\mathbf{x}_{t+1}\mathbf{x}_{t+1}^\top\right]\tilde{\boldsymbol{\Sigma}}_e^{-1}\right\} + \frac{2\xi}{\sigma_d^2}\mathrm{tr}\left\{\mathbf{E}_{\Theta'}\left[\sum_{t=0}^{T-1}\mathbf{x}_t\mathbf{x}_{t+1}^\top\right]\tilde{\boldsymbol{\Sigma}}_e^{-1}\right\}\\
&-\frac{\xi^2}{\sigma_d^2}\mathrm{tr}\left\{\mathbf{E}_{\Theta'}\left[\sum_{t=0}^{T-1}\mathbf{x}_t\mathbf{x}_t^\top\right]\tilde{\boldsymbol{\Sigma}}_e^{-1}\right\} + \frac{2}{\sigma_d^2}\mathbf{E}_{\Theta'}\left[\sum_{t=0}^{T-1}\mathbf{x}_{t+1}^\top\tilde{\boldsymbol{\Sigma}}_e^{-1}\mathbf{q}(\mathbf{x}_t)\right]\boldsymbol{\theta}\\
&-\frac{1}{\sigma_d^2}\boldsymbol{\theta}^\top\mathbf{E}_{\Theta'}\left[\sum_{t=0}^{T-1}\mathbf{q}^\top(\mathbf{x}_t)\tilde{\boldsymbol{\Sigma}}_e^{-1}\mathbf{q}(\mathbf{x}_t)\right]\boldsymbol{\theta} - \frac{2\xi}{\sigma_d^2}\mathbf{E}_{\Theta'}\left[\sum_{t=0}^{T-1}\mathbf{x}_t^\top\tilde{\boldsymbol{\Sigma}}_e^{-1}\mathbf{q}(\mathbf{x}_t)\right]\boldsymbol{\theta}. \qquad (95)
\end{aligned}
$$

The third term in (**??**) is obtained using the same method in [**?**], i.e. by adding and subtracting $\mathbf{C}\hat{\mathbf{x}}_{t+1}^b\hat{\mathbf{x}}_{t+1}^{b\top}\mathbf{C}^\top$ to the expectation of the second term in (**??**) and then exploiting the expression given in (**??**) . By defining $\boldsymbol{\Xi}$-variables as

$$\boldsymbol{\Xi}_0 = \mathbf{E}_{\Theta'}\left[\sum_{t=0}^{T-1}\mathbf{x}_{t+1}\mathbf{x}_{t+1}^\top\right] \qquad (96)$$

$$\boldsymbol{\Xi}_1 = \mathbf{E}_{\Theta'}\left[\sum_{t=0}^{T-1}\mathbf{x}_t\mathbf{x}_{t+1}^\top\right] \qquad (97)$$

$$\boldsymbol{\Xi}_2 = \mathbf{E}_{\Theta'}\left[\sum_{t=0}^{T-1}\mathbf{x}_t\mathbf{x}_t^\top\right] \qquad (98)$$

$$\boldsymbol{\Xi}_3 = \mathbf{E}_{\Theta'}\left[\sum_{t=0}^{T-1}\mathbf{x}_{t+1}^\top\tilde{\boldsymbol{\Sigma}}_e^{-1}\mathbf{q}(\mathbf{x}_t)\right] \qquad (99)$$

$$\boldsymbol{\Xi}_4 = \mathbf{E}_{\Theta'}\left[\sum_{t=0}^{T-1}\mathbf{q}^\top(\mathbf{x}_t)\tilde{\boldsymbol{\Sigma}}_e^{-1}\mathbf{q}(\mathbf{x}_t)\right] \qquad (100)$$

$$\boldsymbol{\Xi}_5 = \mathbf{E}_{\Theta'}\left[\sum_{t=0}^{T-1}\mathbf{x}_t^\top\tilde{\boldsymbol{\Sigma}}_e^{-1}\mathbf{q}(\mathbf{x}_t)\right] \qquad (101)$$

equation (**??**) can be written in a more compact form

$$
\begin{aligned}
\mathcal{Q}(\boldsymbol{\Theta},\boldsymbol{\Theta}') = &-\frac{1}{\sigma_\epsilon^2}\mathrm{tr}\left\{\sum\nolimits_{t=0}^{T-1}\left[(\mathbf{y}_{t+1}-\mathbf{C}\hat{\mathbf{x}}^b_{t+1})(\mathbf{y}_{t+1}-\mathbf{C}\hat{\mathbf{x}}^b_{t+1})^\top+\mathbf{C}\mathbf{P}^b_{t+1}\mathbf{C}^\top\right]\right\} \\
&-Tn_y\ln\sigma_\epsilon^2-Tn_x\ln\sigma_d^2-\frac{1}{\sigma_d^2}\mathrm{tr}\left\{\boldsymbol{\Xi}_0\tilde{\boldsymbol{\Sigma}}_e^{-1}\right\}+\frac{2\xi}{\sigma_d^2}\mathrm{tr}\left\{\boldsymbol{\Xi}_1\tilde{\boldsymbol{\Sigma}}_e^{-1}\right\} \\
&-\frac{\xi^2}{\sigma_d^2}\mathrm{tr}\left\{\boldsymbol{\Xi}_2\tilde{\boldsymbol{\Sigma}}_e^{-1}\right\}+\frac{2}{\sigma_d^2}\boldsymbol{\Xi}_3\boldsymbol{\theta}-\frac{1}{\sigma_d^2}\boldsymbol{\theta}^\top\boldsymbol{\Xi}_4\boldsymbol{\theta}-\frac{2\xi}{\sigma_d^2}\boldsymbol{\Xi}_5\boldsymbol{\theta}. \qquad (102)
\end{aligned}
$$

The covariance and cross-covariance terms in (**??**) consist of both the linear terms, $\boldsymbol{\Xi}_0$, $\boldsymbol{\Xi}_1$ and $\boldsymbol{\Xi}_2$ and the non-linear terms, $\boldsymbol{\Xi}_3$, $\boldsymbol{\Xi}_4$ and $\boldsymbol{\Xi}_5$. Finding the linear $\boldsymbol{\Xi}$-variables is straightforward as they can be computed directly from the smoother outputs, i.e. $\hat{\mathbf{x}}^b_t$, $\mathbf{P}^b_t$ and $\mathbf{M}^b_t$ [**?**]. To show how it can be done, the smoothed covariance matrix at time $t+1$ can be written as

$$
\mathbf{P}^b_{t+1} = \mathbf{E}_{\Theta'}\left[(\mathbf{x}_{t+1}-\hat{\mathbf{x}}^b_{t+1})(\mathbf{x}_{t+1}-\hat{\mathbf{x}}^b_{t+1})^\top\right] = \mathbf{E}_{\theta'}\left[\mathbf{x}_{t+1}\mathbf{x}_{t+1}^\top\right]-\hat{\mathbf{x}}^b_{t+1}\hat{\mathbf{x}}^{b\top}_{t+1} \qquad (103)
$$

$$
\mathbf{M}^b_{t+1} = \mathbf{E}_{\Theta'}\left[(\mathbf{x}_{t}-\hat{\mathbf{x}}^b_{t})(\mathbf{x}_{t+1}-\hat{\mathbf{x}}^b_{t+1})^\top\right] = \mathbf{E}_{\Theta'}\left[\mathbf{x}_{t}\mathbf{x}_{t+1}^\top\right]-\hat{\mathbf{x}}^b_{t}\hat{\mathbf{x}}^{b\top}_{t+1}, \qquad (104)
$$

consequently,

$$
\mathbf{E}_{\Theta'}\left[\mathbf{x}_{t+1}\mathbf{x}_{t+1}^\top\right] = \mathbf{P}^b_{t+1}+\hat{\mathbf{x}}^b_{t+1}\hat{\mathbf{x}}^{b\top}_{t+1} \qquad (105)
$$

$$
\mathbf{E}_{\Theta'}\left[\mathbf{x}_{t}\mathbf{x}_{t+1}^\top\right] = \mathbf{M}^b_{t+1}+\hat{\mathbf{x}}^b_{t}\hat{\mathbf{x}}^{b\top}_{t+1}, \qquad (106)
$$

substituting (**??**) and (**??**) in (**??**) and (**??**) respectively , gives the final formula for $\boldsymbol{\Xi}_0$ and $\boldsymbol{\Xi}_1$, i.e.

$$
\boldsymbol{\Xi}_0 = \sum_{t=0}^{T-1}\left(\mathbf{P}^b_{t+1}+\hat{\mathbf{x}}^b_{t+1}\hat{\mathbf{x}}^{b\top}_{t+1}\right), \qquad (107)
$$

$$
\boldsymbol{\Xi}_1 = \sum_{t=0}^{T-1}\left(\mathbf{M}^b_{t+1}+\hat{\mathbf{x}}^b_{t}\hat{\mathbf{x}}^{b\top}_{t+1}\right). \qquad (108)
$$

The expression for $\boldsymbol{\Xi}_2$ can be derived in a similar manner, giving

$$
\boldsymbol{\Xi}_2 = \sum_{t=0}^{T-1}\left(\mathbf{P}^b_{t}+\hat{\mathbf{x}}^b_{t}\hat{\mathbf{x}}^{b\top}_{t}\right). \qquad (109)
$$

However, calculation of the non-linear $\boldsymbol{\Xi}$-variables is not straightforward as in the linear case. Therefore, an additional Taylor expansion step to the URTSS algorithm is also created to linearise $\mathbf{q}(\mathbf{x}_t)$, enabling the calculations of $\boldsymbol{\Xi}_3$, $\boldsymbol{\Xi}_4$ and $\boldsymbol{\Xi}_5$ from the smoother outputs. This is accomplished by taking the first-order truncated Taylor-series expansion of the sigmoidal activation function at each time instant around the smoothed state, $\hat{\mathbf{x}}^b_t$, i.e.

$$
f\left(\boldsymbol{\phi}^\top(\mathbf{r}')\,\mathbf{x}_t\right) \approx f\left(\boldsymbol{\phi}^\top(\mathbf{r}')\hat{\mathbf{x}}^b_t\right)+\boldsymbol{\phi}^\top(\mathbf{r}')\left(\mathbf{x}_t-\hat{\mathbf{x}}^b_t\right)f'\left(\boldsymbol{\phi}^\top(\mathbf{r}')\,\hat{\mathbf{x}}^b_t\right), \qquad (110)
$$

where

$$
\begin{aligned}
f'(\boldsymbol{\phi}^\top(\mathbf{r}')\hat{\mathbf{x}}^b_t) = &\frac{\varsigma}{\left(1+\exp\left(\varsigma\left(v_0-\boldsymbol{\phi}^\top(\mathbf{r}')\,\hat{\mathbf{x}}^b_t\right)\right)\right)^2}\times\exp\left(\varsigma\left(v_0-\boldsymbol{\phi}^\top(\mathbf{r}')\,\hat{\mathbf{x}}^b_t\right)\right) \\
=&\frac{\varsigma}{1+\exp\left(\varsigma\left(v_0-\boldsymbol{\phi}^\top(\mathbf{r}')\,\hat{\mathbf{x}}^b_t\right)\right)}\times\left(1-\frac{1}{1+\exp\left(\varsigma\left(v_0-\boldsymbol{\phi}^\top(\mathbf{r}')\,\hat{\mathbf{x}}^b_t\right)\right)}\right) \\
=&\varsigma f(\boldsymbol{\phi}^\top(\mathbf{r}')\,\hat{\mathbf{x}}^b_t)\left(1-f(\boldsymbol{\phi}^\top(\mathbf{r}')\,\hat{\mathbf{x}}^b_t\right). \qquad (111)
\end{aligned}
$$

substituting (**??**) in (**??**), an approximation for $\mathbf{q}(\mathbf{x}_t)$ around the smoothed state $\hat{\mathbf{x}}_t^b$ can be obtained

$$\mathbf{q}(\mathbf{x}_t) \approx \mathbf{q}(\hat{\mathbf{x}}_t^b) + \int_\Omega \boldsymbol{\Psi}\left(\mathbf{r}'\right) \boldsymbol{\phi}^\top\left(\mathbf{r}'\right)\left(\mathbf{x}_t - \hat{\mathbf{x}}_t^b\right) f'\left(\boldsymbol{\phi}^\top\left(\mathbf{r}'\right)\hat{\mathbf{x}}_t^b\right) d\mathbf{r}'. \tag{112}$$

Having approximated $\mathbf{q}\left(\mathbf{x}_t\right)$, by substituting (**??**) into equations (**??**-**??**) an approximation to each of the non-linear $\boldsymbol{\Xi}$-variable can be made. The three non-linear variables are dealt with in turn. To start, first consider (**??**), substituting in (**??**) for $\mathbf{q}\left(\mathbf{x}_t\right)$, the term under the summation sign can be approximated by

$$\mathbf{x}_{t+1}^\top \tilde{\boldsymbol{\Sigma}}_e^{-1}\mathbf{q}(\mathbf{x}_t) \approx \mathbf{x}_{t+1}^\top \tilde{\boldsymbol{\Sigma}}_e^{-1}\mathbf{q}(\hat{\mathbf{x}}_t^b) + \mathbf{x}_{t+1}^\top \tilde{\boldsymbol{\Sigma}}_e^{-1} \int_\Omega \boldsymbol{\Psi}(\mathbf{r}')\boldsymbol{\phi}^\top(\mathbf{r}')(\mathbf{x}_t - \hat{\mathbf{x}}_t^b) f'(\boldsymbol{\phi}^\top(\mathbf{r}')\hat{\mathbf{x}}_t^b) d\mathbf{r}', \tag{113}$$

noting that $\phi^\top(\mathbf{r}')(\mathbf{x}_t - \hat{\mathbf{x}}_t^b)$ is scalar, (**??**) can be rearranged in the form

$$\mathbf{x}_{t+1}^\top \tilde{\boldsymbol{\Sigma}}_e^{-1}\mathbf{q}(\mathbf{x}_t) \approx \mathbf{x}_{t+1}^\top \tilde{\boldsymbol{\Sigma}}_e^{-1}\mathbf{q}(\hat{\mathbf{x}}_t^b) + \int_\Omega \boldsymbol{\phi}^\top(\mathbf{r}')(\mathbf{x}_t - \hat{\mathbf{x}}_t^b)\mathbf{x}_{t+1}^\top \tilde{\boldsymbol{\Sigma}}_e^{-1} \boldsymbol{\Psi}(\mathbf{r}') f'(\boldsymbol{\phi}^\top(\mathbf{r}')\hat{\mathbf{x}}_t^b) d\mathbf{r}', \tag{114}$$

taking expected value of (**??**) gives

$$\begin{aligned}\mathbf{E}_{\boldsymbol{\Theta}'}\left[\mathbf{x}_{t+1}^\top \tilde{\boldsymbol{\Sigma}}_e^{-1}\mathbf{q}(\mathbf{x}_t)\right] &\approx \mathbf{E}_{\boldsymbol{\Theta}'}\left[\mathbf{x}_{t+1}^\top\right] \tilde{\boldsymbol{\Sigma}}_e^{-1}\mathbf{q}(\hat{\mathbf{x}}_t^b) \\ &\quad + \int_\Omega \boldsymbol{\phi}^\top(\mathbf{r}')\mathbf{E}_{\boldsymbol{\Theta}'}\left[(\mathbf{x}_t - \hat{\mathbf{x}}_t^b)\mathbf{x}_{t+1}^\top\right] \tilde{\boldsymbol{\Sigma}}_e^{-1} \boldsymbol{\Psi}(\mathbf{r}') f'(\boldsymbol{\phi}^\top(\mathbf{r}')\hat{\mathbf{x}}_t^b) d\mathbf{r}',\end{aligned} \tag{115}$$

Substituting for $\mathbf{E}_{\boldsymbol{\Theta}'}\left[(\mathbf{x}_t - \hat{\mathbf{x}}_t^b)\mathbf{x}_{t+1}^\top\right]$ from (**??**) and summing over $t \in \{0, \ldots, T-1\}$ yields

$$\boldsymbol{\Xi}_3 \approx \sum_{t=0}^{T-1}\left[\hat{\mathbf{x}}_{t+1}^{b\top} \tilde{\boldsymbol{\Sigma}}_e^{-1}\mathbf{q}(\hat{\mathbf{x}}_t^b) + \int_\Omega \boldsymbol{\phi}^\top(\mathbf{r}')\mathbf{M}_{t+1}^b \tilde{\boldsymbol{\Sigma}}_e^{-1} \boldsymbol{\Psi}(\mathbf{r}') f'(\boldsymbol{\phi}^\top(\mathbf{r}')\hat{\mathbf{x}}_t^b) d\mathbf{r}'\right]. \tag{116}$$

To find an approximation for $\boldsymbol{\Xi}_4$, a similar approach is used where the non-linear expression in (**??**) is replaced by (**??**) giving

$$\begin{aligned}\mathbf{q}^\top(\mathbf{x}_t)\tilde{\boldsymbol{\Sigma}}_e^{-1}\mathbf{q}(\mathbf{x}_t) &\approx \left[\mathbf{q}(\hat{\mathbf{x}}_t^b) + \int_\Omega \boldsymbol{\Psi}(\mathbf{r}')\boldsymbol{\phi}^\top(\mathbf{r}')(\mathbf{x}_t - \hat{\mathbf{x}}_t^b) f'(\boldsymbol{\phi}^\top(\mathbf{r}')\hat{\mathbf{x}}_t^b) d\mathbf{r}'\right]^\top \times \tilde{\boldsymbol{\Sigma}}_e^{-1} \\ &\quad \times \left[\mathbf{q}(\hat{\mathbf{x}}_t^b) + \int_\Omega \boldsymbol{\Psi}(\mathbf{r}')\boldsymbol{\phi}^\top(\mathbf{r}')(\mathbf{x}_t - \hat{\mathbf{x}}_t^b) f'(\boldsymbol{\phi}^\top(\mathbf{r}')\hat{\mathbf{x}}_t^b) d\mathbf{r}'\right] \\ &= \text{term1} + \text{term2} + \text{term3} + \text{term4},\end{aligned} \tag{117}$$

where

$$\text{term1} = \mathbf{q}^\top(\hat{\mathbf{x}}_t^b)\tilde{\boldsymbol{\Sigma}}_e^{-1}\mathbf{q}(\hat{\mathbf{x}}_t^b) \tag{118}$$

$$\text{term2} = \mathbf{q}^\top(\hat{\mathbf{x}}_t^b)\tilde{\boldsymbol{\Sigma}}_e^{-1}\int_\Omega \boldsymbol{\Psi}(\mathbf{r}')\boldsymbol{\phi}^\top(\mathbf{r}')(\mathbf{x}_t - \hat{\mathbf{x}}_t^b) f'(\boldsymbol{\phi}^\top(\mathbf{r}')\hat{\mathbf{x}}_t^b) d\mathbf{r}' \tag{119}$$

$$\text{term3} = \int_\Omega \boldsymbol{\Psi}^\top(\mathbf{r}')\boldsymbol{\phi}^\top(\mathbf{r}')(\mathbf{x}_t - \hat{\mathbf{x}}_t^b) f'(\boldsymbol{\phi}^\top(\mathbf{r}')\hat{\mathbf{x}}_t^b) d\mathbf{r}' \tilde{\boldsymbol{\Sigma}}_e^{-1}\mathbf{q}(\hat{\mathbf{x}}_t^b) \tag{120}$$

$$\begin{aligned}\text{term4} &= \int_\Omega \boldsymbol{\Psi}^\top(\mathbf{r}')\boldsymbol{\phi}^\top(\mathbf{r}')(\mathbf{x}_t - \hat{\mathbf{x}}_t^b) f'(\boldsymbol{\phi}^\top(\mathbf{r}')\hat{\mathbf{x}}_t^b) d\mathbf{r}' \times \tilde{\boldsymbol{\Sigma}}_e^{-1} \\ &\quad \times \int_\Omega \boldsymbol{\Psi}(\mathbf{r}')\boldsymbol{\phi}^\top(\mathbf{r}')(\mathbf{x}_t - \hat{\mathbf{x}}_t^b) f'(\boldsymbol{\phi}^\top(\mathbf{r}')\hat{\mathbf{x}}_t^b) d\mathbf{r}'\end{aligned} \tag{121}$$

It should be noted that when applying expectation, term1 is constant, and term2 and term3 vanish as $\mathbf{E}_{\Theta'}\left[(\mathbf{x}_t - \hat{\mathbf{x}}_t^b)\right] = \mathbf{0}$. However in term4 to use the covariance estimate from the smoother the $\mathbf{x}_t$ terms should be permuted. To do this (**??**) can be rewritten as

$$\begin{aligned}
term4 &= \int_\Omega \boldsymbol{\Psi}^\top(\mathbf{r}')f'(\boldsymbol{\phi}^\top(\mathbf{r}')\hat{\mathbf{x}}_t^b)\boldsymbol{\phi}^\top(\mathbf{r}')(\mathbf{x}_t - \hat{\mathbf{x}}_t^b)d\mathbf{r}' \times \tilde{\boldsymbol{\Sigma}}_e^{-1} \\
&\times \int_\Omega (\mathbf{x}_t - \hat{\mathbf{x}}_t^b)^\top \boldsymbol{\phi}(\mathbf{r}')\boldsymbol{\Psi}(\mathbf{r}')f'(\boldsymbol{\phi}^\top(\mathbf{r}')\hat{\mathbf{x}}_t^b)d\mathbf{r}' \\
&= \sum_{i,j=1}^{n_x} \boldsymbol{\Lambda}_t^{(i)\top}(\mathbf{x}_{i,t} - \hat{\mathbf{x}}_{i,t}^b)(\mathbf{x}_{j,t} - \hat{\mathbf{x}}_{j,t}^b)\tilde{\boldsymbol{\Sigma}}_e^{-1}\boldsymbol{\Lambda}_t^{(j)},
\end{aligned} \tag{122}$$

where

$$\boldsymbol{\Lambda}_t^{(i)} = \int_\Omega \boldsymbol{\Psi}(\mathbf{r}')\phi_i(\mathbf{r}')f'(\boldsymbol{\phi}^\top(\mathbf{r}')\hat{\mathbf{x}}_t^b)d\mathbf{r}'. \tag{123}$$

Now taking expectation from (**??**) gives

$$\begin{aligned}
\mathbf{E}_{\boldsymbol{\Theta}'}\left[\mathbf{q}^\top(\mathbf{x}_t)\tilde{\boldsymbol{\Sigma}}_e^{-1}\mathbf{q}(\mathbf{x}_t)\right] &\approx \mathbf{q}^\top(\hat{\mathbf{x}}_t^b)\tilde{\boldsymbol{\Sigma}}_e^{-1}\mathbf{q}(\hat{\mathbf{x}}_t^b) \\
&+ \sum_{i,j=1}^{n_x} \boldsymbol{\Lambda}_t^{(i)\top}\mathbf{E}_{\boldsymbol{\Theta}'}\left[(\mathbf{x}_{i,t} - \hat{\mathbf{x}}_{i,t}^b)(\mathbf{x}_{j,t} - \hat{\mathbf{x}}_{j,t}^b)\right]\tilde{\boldsymbol{\Sigma}}_e^{-1}\boldsymbol{\Lambda}_t^{(j)},
\end{aligned} \tag{124}$$

and noting $\mathbf{E}_{\boldsymbol{\theta}'}\left[(\mathbf{x}_{i,t} - \hat{\mathbf{x}}_{i,t}^b)(\mathbf{x}_{j,t} - \hat{\mathbf{x}}_{j,t}^b)\right] = \left[\mathbf{P}_t^b\right]_{i,j}$ gives

$$\mathbf{E}_{\boldsymbol{\Theta}'}\left[\mathbf{q}^\top(\mathbf{x}_t)\tilde{\boldsymbol{\Sigma}}_e^{-1}\mathbf{q}(\mathbf{x}_t)\right] = \mathbf{q}^\top(\hat{\mathbf{x}}_t)\tilde{\boldsymbol{\Sigma}}_e^{-1}\mathbf{q}(\hat{\mathbf{x}}_t) + \boldsymbol{\mathit{\Lambda}}_t \tag{125}$$

where

$$\boldsymbol{\mathit{\Lambda}}_t = \sum_{i,j=1}^{n_x}[\mathbf{P}_t^b]_{ij}\boldsymbol{\Lambda}_t^{(i)\top}\tilde{\boldsymbol{\Sigma}}_e^{-1}\boldsymbol{\Lambda}_t^{(j)}, \tag{126}$$

summing over $t \in \{0, \ldots, T-1\}$ yields

$$\boldsymbol{\Xi}_4 \approx \sum_{t=0}^{T-1}\left[\mathbf{q}^\top(\hat{\mathbf{x}}_t^b)\tilde{\boldsymbol{\Sigma}}_e^{-1}\mathbf{q}(\hat{\mathbf{x}}_t^b) + \boldsymbol{\mathit{\Lambda}}_t\right]. \tag{127}$$

Finally, to calculate $\boldsymbol{\Xi}_5$, the non-linear term can be treated in a similar way, using the expression for $\mathbf{q}(\mathbf{x}_t)$ in (**??**) and rearranging gives

$$\mathbf{x}_t^\top\tilde{\boldsymbol{\Sigma}}_e^{-1}\mathbf{q}(\mathbf{x}_t) \approx \mathbf{x}_t^\top\tilde{\boldsymbol{\Sigma}}_e^{-1}\mathbf{q}(\hat{\mathbf{x}}_t^b) + \int_\Omega \boldsymbol{\phi}^\top(\mathbf{r}')(\mathbf{x}_t - \hat{\mathbf{x}}_t^b)\mathbf{x}_t^\top\tilde{\boldsymbol{\Sigma}}_e^{-1}\boldsymbol{\Psi}(\mathbf{r}')f'(\boldsymbol{\phi}^\top(\mathbf{r}')\hat{\mathbf{x}}_t^b)d\mathbf{r}'. \tag{128}$$

Taking expected value of (**??**) yields

$$\begin{aligned}
\mathbf{E}_{\Theta'}\left[\mathbf{x}_t^\top\boldsymbol{\Sigma}_e^{-1}\mathbf{q}(\mathbf{x}_t)\right] &\approx \mathbf{E}_{\Theta'}\left[\mathbf{x}_t^\top\right]\tilde{\boldsymbol{\Sigma}}_e^{-1}\mathbf{q}(\hat{\mathbf{x}}_t^b) \\
&+ \int_\Omega \boldsymbol{\phi}^\top(\mathbf{r}')\mathbf{E}_{\Theta'}\left[(\mathbf{x}_t - \hat{\mathbf{x}}_t^b)\mathbf{x}_t^\top\right]\tilde{\boldsymbol{\Sigma}}_e^{-1}\boldsymbol{\Psi}(\mathbf{r}')f'(\boldsymbol{\phi}^\top(\mathbf{r}')\hat{\mathbf{x}}_t^b)d\mathbf{r}'.
\end{aligned} \tag{129}$$

Substituting for $\mathbf{E}_{\Theta'}\left[(\mathbf{x}_t - \hat{\mathbf{x}}_t^b)\mathbf{x}_t^\top\right]$ from (**??**) at time t, and summing over $t \in \{0, \ldots, T-1\}$ gives

$$\boldsymbol{\Xi}_5 \approx \sum_{t=0}^{T-1}\left[\hat{\mathbf{x}}_t^{b\top}\tilde{\boldsymbol{\Sigma}}_e^{-1}\mathbf{q}(\hat{\mathbf{x}}_t^b) + \int_\Omega \boldsymbol{\phi}^\top(\mathbf{r}')\mathbf{P}_t^b\tilde{\boldsymbol{\Sigma}}_e^{-1}\boldsymbol{\Psi}(\mathbf{r}')f'(\boldsymbol{\phi}^\top(\mathbf{r}')\hat{\mathbf{x}}_t^b)d\mathbf{r}'\right] \tag{130}$$

By computing $\boldsymbol{\Xi}$-variables, the approximation of the lower bound is complete. The lower bound now can be used in the M-step to find a set of new parameters.

3.1.3 M-Step

The objective of the M-step is to maximise the computed lower bound in the E-step, which consequently increases the log-likelihood function $\ln p(\mathbf{Y};\boldsymbol{\Theta})$. Consider first the maximisation of the $\mathcal{Q}$-function with respect to ξ, differentiating (**??**) with respect to the synaptic parameter gives

$$\frac{\partial \mathcal{Q}}{\partial \xi} = \frac{2}{\sigma_d^2}\mathrm{tr}\left\{\boldsymbol{\Xi}_1\tilde{\boldsymbol{\Sigma}}_e^{-1}\right\} - \frac{2\xi}{\sigma_d^2}\mathrm{tr}\left\{\boldsymbol{\Xi}_2\tilde{\boldsymbol{\Sigma}}_e^{-1}\right\} - \frac{2}{\sigma_d^2}\boldsymbol{\Xi}_5\boldsymbol{\theta}, \tag{131}$$

Substituting for $\boldsymbol{\theta}$ from (**??**) and then setting the result to zero and solving for ξ gives

$$\hat{\xi} = \frac{\mathrm{tr}\left\{\boldsymbol{\Xi}_1\tilde{\boldsymbol{\Sigma}}_e^{-1}\right\} - \boldsymbol{\Xi}_5\boldsymbol{\Xi}_4^{-1}\boldsymbol{\Xi}_3^\top}{\mathrm{tr}\left\{\boldsymbol{\Xi}_2\tilde{\boldsymbol{\Sigma}}_e^{-1}\right\} - \boldsymbol{\Xi}_5\boldsymbol{\Xi}_4^{-1}\boldsymbol{\Xi}_5^\top}. \tag{132}$$

In order to maximise the $\mathcal{Q}$-function with respect to the connectivity kernel parameters, $\boldsymbol{\theta}$, differentiation is taken as follows

$$\frac{\partial \mathcal{Q}}{\partial \boldsymbol{\theta}} = \frac{2}{\sigma_d^2}\boldsymbol{\Xi}_3 - \frac{1}{\sigma_d^2}\boldsymbol{\theta}^\top(\boldsymbol{\Xi}_4^\top + \boldsymbol{\Xi}_4) - \frac{2\xi}{\sigma_d^2}\boldsymbol{\Xi}_5, \tag{133}$$

noting that the variable $\boldsymbol{\Xi}_4$ is symmetric by construction, equating (**??**) to the zero vector yields

$$\hat{\boldsymbol{\theta}} = \boldsymbol{\Xi}_4^{-1}\left(\boldsymbol{\Xi}_3^\top - \xi\boldsymbol{\Xi}_5^\top\right). \tag{134}$$

The condition under which $\boldsymbol{\Xi}_4$ is invertible will be discussed later in this section (see below (**??**)). Now maximising the lower bound with respect to σ_ϵ^2 gives

$$\frac{\partial \mathcal{Q}}{\partial \sigma_\epsilon^2} = -Tn_y\frac{1}{\sigma_\epsilon^2} + \frac{1}{\sigma_\epsilon^4}\mathrm{tr}\left\{\sum_{t=0}^{T-1}\left[(\mathbf{y}_{t+1} - \mathbf{C}\hat{\mathbf{x}}_{t+1})(\mathbf{y}_{t+1} - \mathbf{C}\hat{\mathbf{x}}_{t+1})^\top + \mathbf{C}\mathbf{P}_{t+1}\mathbf{C}^\top\right]\right\}, \tag{135}$$

and setting the above equation to zero and solving for σ_ϵ^2 yields

$$\hat{\sigma}_\epsilon^2 = \frac{1}{Tn_y}\sum_{t=0}^{T-1}\mathrm{tr}\left\{(\mathbf{y}_{t+1} - \mathbf{C}\hat{\mathbf{x}}_{t+1})(\mathbf{y}_{t+1} - \mathbf{C}\hat{\mathbf{x}}_{t+1})^\top + \mathbf{C}\mathbf{P}_{t+1}\mathbf{C}^\top\right\}. \tag{136}$$

Finally differentiating the lower bound with respect to σ_d^2 gives

$$\begin{aligned}\frac{\partial \mathcal{Q}}{\partial \sigma_d^2} = &-Tn_x\frac{1}{\sigma_d^2} + \frac{1}{\sigma_d^4}\mathrm{tr}\left\{\boldsymbol{\Xi}_0\tilde{\boldsymbol{\Sigma}}_e^{-1}\right\} - \frac{2\xi}{\sigma_d^4}\mathrm{tr}\left\{\boldsymbol{\Xi}_1\tilde{\boldsymbol{\Sigma}}_e^{-1}\right\} \\ &+ \frac{\xi^2}{\sigma_d^4}\mathrm{tr}\left\{\boldsymbol{\Xi}_2\tilde{\boldsymbol{\Sigma}}_e^{-1}\right\} - \frac{2}{\sigma_d^4}\boldsymbol{\Xi}_3\boldsymbol{\theta} + \frac{1}{\sigma_d^4}\boldsymbol{\theta}^\top\boldsymbol{\Xi}_4\boldsymbol{\theta} + \frac{2\xi}{\sigma_d^4}\boldsymbol{\Xi}_5\boldsymbol{\theta},\end{aligned} \tag{137}$$

setting the above to zero and solving for σ_d^2 completes the maximisation step.

$$\begin{aligned}\hat{\sigma}_d^2 = \frac{1}{Tn_x}&\left[\mathrm{tr}\left\{\boldsymbol{\Xi}_0\tilde{\boldsymbol{\Sigma}}_e^{-1}\right\} - 2\xi\mathrm{tr}\left\{\boldsymbol{\Xi}_1\tilde{\boldsymbol{\Sigma}}_e^{-1}\right\} + \xi^2\mathrm{tr}\left\{\boldsymbol{\Xi}_2\tilde{\boldsymbol{\Sigma}}_e^{-1}\right\}\right. \\ &\left.-2\boldsymbol{\Xi}_3\boldsymbol{\theta} + \boldsymbol{\theta}^\top\boldsymbol{\Xi}_4\boldsymbol{\theta} + 2\xi\boldsymbol{\Xi}_5\boldsymbol{\theta}\right].\end{aligned} \tag{138}$$

In order to confirm that the calculated solutions maximise the lower bound, a second derivative test is performed for each of the parameter giving

$$\left.\frac{\partial \mathcal{Q}^2}{\partial^2 \xi}\right|_{\xi=\hat{\xi}} = -\frac{2}{\sigma_d^2}\mathrm{tr}\left\{\boldsymbol{\Xi}_2\tilde{\boldsymbol{\Sigma}}_e^{-1}\right\} \tag{139}$$

$$\left.\frac{\partial \mathcal{Q}^2}{\partial^2 \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}} = -\frac{2}{\sigma_d^2}\boldsymbol{\Xi}_4 \tag{140}$$

$$\left.\frac{\partial \mathcal{Q}^2}{\partial^2 \sigma_\epsilon^2}\right|_{\sigma_\epsilon=\hat{\sigma}_\epsilon} = -\frac{Tn_y}{\hat{\sigma}_\epsilon^4} \tag{141}$$

$$\left.\frac{\partial \mathcal{Q}^2}{\partial^2 \sigma_d^2}\right|_{\sigma_d=\hat{\sigma}_d} = -\frac{Tn_x}{\hat{\sigma}_d^4} \tag{142}$$

$\boldsymbol{\Xi}_2$ is positive definite under the persistent excitation condition of the state vectors, which is guaranteed by the disturbance, $\mathbf{e}_t(\mathbf{r})$, and $\tilde{\boldsymbol{\Sigma}}_e^{-1}$ is positive definite by construction (Note that the inverse of a positive definite matrix is also positive definite [?]), therefore by [?]

$$\mathrm{tr}\left\{\boldsymbol{\Xi}_2\tilde{\boldsymbol{\Sigma}}_e^{-1}\right\} > 0, \tag{143}$$

meaning (**??**) is negative and $\hat{\xi}$ represents a maximum. Equation (**??**) is negative where $\boldsymbol{\Xi}_4$ is positive definite requiring $\mathbf{q}(\mathbf{x}_t)$ to be of rank n_θ [?]. However, $\mathbf{q}(\mathbf{x}_t)$ defined in (**??**) cannot be computed analytically as the inner product of a Gaussian and a sigmoid function cannot be found analytically [?], making it difficult to calculate the rank of $\mathbf{q}(\mathbf{x}_t)$. Therefore, the rank of $\mathbf{q}(\mathbf{x}_t)$ at each iteration of the algorithm should be examined numerically. In the experiments of this work the problem of rank deficiency was never encountered. From (**??**) and (**??**), the observation noise and disturbance variance estimates clearly represent maxima of the $\mathcal{Q}$-function. The joint state and parameter estimation algorithm is summarised in Algorithm **??**.

For initialisation a bounded random state sequence is employed to ensure that the initial parameter estimates describe a stable kernel, although the quality of such initial estimate is poor, it provides an initialisation that typically ensues a satisfactory convergence of the algorithm, as illustrated numerically in §**??**. The stopping criterion is usually associated with either the change in the parameter estimates or the log-likelihood variation [?]. Here the change in parameter estimates, L^2-norm in the case of the connectivity kernel weights, is used to halt the execution of the algorithm. Besides, the lower bound on the likelihood function is also monitored to further investigate the convergence of the EM algorithm.

An approximate for the computational complexity of the algorithm is given in Table **??**. In this table only the computational complexity of the variables specific to the presented algorithm are shown, and the computational complexity of the URTSS can be found in [?]. Note that the computational complexity of the non-linear terms depend on the number of points, n_r, used for the spatial discretisation. Calculation of the non-linear terms is clearly the slowest part of the algorithm where n_x is the critical dimension, more specifically the matrix $\boldsymbol{\Xi}_4$ has the complexity that grows with the order of four in n_x. Although the number of states is decoupled from the number of observation locations (recording electrodes in this case) , and the number of basis functions for the connectivity kernel decomposition, it grows quadratically in two dimensional space for the equally spaced basis functions.

4 Results

In this section, a demonstration of the estimation framework for the proposed IDE neural field model is provided. To demonstrate the estimation procedure, simulated data is used to establish a ground truth for a comparison. The parameters of the model that are estimated include the synaptic time constant, the shape of the connectivity kernel, the field disturbance properties and the observation noise properties. All the parameters for the simulated data given in Table **??**. To demonstrate the estimation procedure, a Monte Carlo simulation was employed where 100 realisations of 500 ms of data were generated. The first 100 ms of each realisation was discarded allowing the model's dynamics to stabilise from the initial conditions.

Variable	Equation	Order
$\mathbf{q}(\mathbf{x}_t)$	(??)	$O(n_x n_\theta n_r)$
$\mathbf{\Xi}_0, \mathbf{\Xi}_1, \mathbf{\Xi}_2$	(??), (??), (??)	$O(Tn_x^2)$
$\mathbf{\Xi}_3, \mathbf{\Xi}_5$	(??),(??)	$O(Tn_x^2 n_r) + O(Tn_x n_\theta n_r)$
$\mathbf{\Xi}_4$	(??)	$O(Tn_x^4 n_\theta) + O(Tn_x^3 n_\theta^2) + O(Tn_x^3 n_\theta n_r)$
$\hat{\xi}$	(??)	$O(n_x^3 + n_\theta^3)$
$\hat{\boldsymbol{\theta}}$	(??)	$O(n_\theta^3)$
$\hat{\sigma}_\epsilon^2$	(??)	$O(Tn_x^2 n_y) + O(Tn_x n_y^2)$
$\hat{\sigma}_d^2$	(??)	$O(n_x^3) + O(n_\theta^2)$

Table 1: Computational complexity of the EM algorithm

Algorithm 1 Joint state and parameter estimation for the IDE model.

1. Initialise: Generate a bounded, random sequence of dimension n_x and use as the initial state sequence estimate.
2. Approximate Ξ-variables

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

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

$$\mathbf{\Xi}_2 = \sum_{t=0}^{T-1} \left(\mathbf{P}_{t}^b + \hat{\mathbf{x}}_{t}^b \hat{\mathbf{x}}_{t}^{b\top} \right)$$

$$\mathbf{\Xi}_3 \approx \sum_{t=0}^{T-1} \left[\hat{\mathbf{x}}_{t+1}^{b\top} \tilde{\mathbf{\Sigma}}_e^{-1} \mathbf{q}(\hat{\mathbf{x}}_t^b) + \int_\Omega \boldsymbol{\phi}^\top(\mathbf{r}') \mathbf{M}_{t+1}^b \tilde{\mathbf{\Sigma}}_e^{-1} \mathbf{\Psi}(\mathbf{r}') f'(\boldsymbol{\phi}^\top(\mathbf{r}')\hat{\mathbf{x}}_t^b) d\mathbf{r}' \right]$$

$$\mathbf{\Xi}_4 \approx \sum_{t=0}^{T-1} \left[\mathbf{q}^\top(\hat{\mathbf{x}}_t^b) \tilde{\mathbf{\Sigma}}_e^{-1} \mathbf{q}(\hat{\mathbf{x}}_t^b) + \sum_{i,j=1}^{n_x} [\mathbf{P}_t^b]_{ij} \mathbf{\Lambda}_t^{(i)\top} \tilde{\mathbf{\Sigma}}_e^{-1} \mathbf{\Lambda}_t^{(j)} \right]$$

$$\mathbf{\Xi}_5 \approx \sum_{t=0}^{T-1} \left[\hat{\mathbf{x}}_{t}^{b\top} \tilde{\mathbf{\Sigma}}_e^{-1} \mathbf{q}(\hat{\mathbf{x}}_t^b) + \int_\Omega \boldsymbol{\phi}^\top(\mathbf{r}') \mathbf{P}_{t}^b \tilde{\mathbf{\Sigma}}_e^{-1} \mathbf{\Psi}(\mathbf{r}') f'(\boldsymbol{\phi}^\top(\mathbf{r}')\hat{\mathbf{x}}_t^b) d\mathbf{r}' \right]$$

3. Calculate the parameter estimates

$$\hat{\xi} = \frac{\mathrm{tr}\left\{\mathbf{\Xi}_1 \tilde{\mathbf{\Sigma}}_e^{-1}\right\} - \mathbf{\Xi}_5 \mathbf{\Xi}_4^{-1} \mathbf{\Xi}_3^\top}{\mathrm{tr}\left\{\mathbf{\Xi}_2 \tilde{\mathbf{\Sigma}}_e^{-1}\right\} - \mathbf{\Xi}_5 \mathbf{\Xi}_4^{-1} \mathbf{\Xi}_5^\top}$$

$$\hat{\boldsymbol{\theta}} = \mathbf{\Xi}_4^{-1} \left(\mathbf{\Xi}_3^\top - \hat{\xi} \mathbf{\Xi}_5^\top \right)$$

$$\hat{\sigma}_\epsilon^2 = \frac{1}{Tn_y} \sum_{t=0}^{T-1} \mathrm{tr}\left\{ (\mathbf{y}_{t+1} - \mathbf{C}\hat{\mathbf{x}}_{t+1})(\mathbf{y}_{t+1} - \mathbf{C}\hat{\mathbf{x}}_{t+1})^\top + \mathbf{C}\mathbf{P}_{t+1}\mathbf{C}^\top \right\}$$

$$\begin{aligned}\hat{\sigma}_d^2 = \frac{1}{Tn_x} & \left[\mathrm{tr}\left\{\mathbf{\Xi}_0 \tilde{\mathbf{\Sigma}}_e^{-1}\right\} - 2\hat{\xi}\mathrm{tr}\left\{\mathbf{\Xi}_1 \tilde{\mathbf{\Sigma}}_e^{-1}\right\} + \hat{\xi}^2 \mathrm{tr}\left\{\mathbf{\Xi}_2 \tilde{\mathbf{\Sigma}}_e^{-1}\right\} \right. \\ & \left. -2\mathbf{\Xi}_3 \hat{\boldsymbol{\theta}} + \hat{\boldsymbol{\theta}}^\top \mathbf{\Xi}_4 \hat{\boldsymbol{\theta}} + 2\hat{\xi}\mathbf{\Xi}_5 \hat{\boldsymbol{\theta}} \right]\end{aligned}$$

4. E step: Execute Algorithm 2.1 of chapter 2, and perform the additional step

$$\mathbf{M}_{t+1}^b = \mathbf{S}_t \mathbf{P}_{t+1}^b$$

   and compute the Ξ-variables using step 2.
5. M step: execute step 3.
6. Calculate stopping criterion.

4.1 Estimation of Connectivity Kernel and Field Disturbance Support

The correlation analysis, described in §**??**, was applied to the simulated data to estimate the unknown support of both the connectivity kernel and the disturbance covariance function. The goal of the estimation procedure for the kernel support is to constrain to positioning and number of kernel basis functions.

From equation **??**, it is clear that the an initial estimate of the synaptic time constant is required to estimate the shape of the connectivity kernel from data. **Maybe we can say something like:** From equation **??**, it can be seen the an error in the estimate of the value of the synaptic time constant will yield either shift up or down in the frequency domain, which leads to a constriction or dial. Since the goal of the procedure is to constrain the positioning of connectivity kernel basis functions, the initial estimate of the time constant should be conservatively **higher (or lower?)**, which will result in a dilation of shape of the kernel (shift lower in the frequency domain). If the shape of the kernel estimate is wider than the actual kernel, then we can be sure that the positioning of the basis functions will capture the connectivity structure once the basis function coefficients have been estimated. If the kernel shape is too narrow, then the parameter values associated with the basis function can not compensate.

In addition to the synaptic time constant, an initial guess for the observation noise variance, σ_ϵ^2, is also required. From equation (**??**), the value of σ_ϵ^2 is bounded. Using the artificial data, the upper bound on σ_ϵ^2 (the minimum value of the power spectral densities) was found to be approximately 0.23.

To explored the sensitivity of the initial guess of σ_ϵ^2, several tests were performed. **The connectivity kernel should come first when presenting the results, since the derivation is presented first in the method section.** First consider (**??**) for estimating the disturbance covariance function.

Figure **??** shows the support estimates for varying σ_ε^2 around the calculated limit. This figure shows that for the values less than 0.23 the estimated support is close to the true one (see Figure **??**(b)) while for the values above 0.23 the shape of the disturbance covariance function is distorted. The estimated support of the covariance function convolved twice with the sensor kernel for σ_ε^2 <0.23 is shown in Figure **??** (a), where the results were normalised and thresholded for values less than 0.01. By fitting a Gaussian basis function to the result, and under the assumption that the width of the sensor, σ_m^2, was known the width of the disturbance covariance function, σ_γ^2, was analytically calculated using (**??**). Alternatively deconvolution can be performed numerically to directly find the support as shown in Figure **??**. The effect of the observation noise variance on the connectivity kernel support is very much like that of the disturbance covariance function, where an initial guess can be made in a similar manner. To study the influence of the synaptic parameter, $\xi = 1 - \frac{T_s}{\tau}$, on the resultant support of the connectivity kernel different values of the synaptic time constant, τ, in a physiologically plausible range were considered, while noting that the chosen values should be higher than the sampling time, T_s. In fact synaptic parameter in (**??**) acts as a scaling factor — dilating or shrinking the estimated support — since subtracting a constant in the frequency domain only affects the value at the origin in the spatial domain. The results of the correlation analysis for the connectivity kernel are shown in Figure **??** where each subplot is the normalised kernel estimate for varying synaptic time constants while holding the observation noise fixed. Note for clarification the amplitude at the origin was negated when the chosen values of τ and σ_ε^2 resulted in a negative value at the origin. The result suggested that a support of approximately $[-5, 5]$ is a reasonable choice for the spatial extent of the connectivity kernel.

4.2 State and parameter estimation

The estimated supports were employed in the EM algorithm described in §**??** to estimate the smoothed states $\hat{\mathbf{x}}_t^b$, the connectivity kernel weights $\boldsymbol{\theta}$, the synaptic dynamic ξ, and the parameters σ_ϵ^2 and σ_d^2. A square grid of $n_\theta = 25$ (5 by 5) basis functions with 2.25 mm spacing were used to cover the estimated kernel support where $\sigma_\psi^2 = 3.5$ mm^2. This is shown in Figure.**??** by blue circles, each of which representing full width at half maximum of the connectivity kernel basis functions. The depicted support was obtained by normalising and thresholding the result for the values less than 0.01. The red circle shows the estimated support which encompasses the non-zero values (black area) after thresholding. The true and estimated connectivity kernel for 100 Monte Carlo simulations are shown in Figure **??**. The cross-sections of the connectivity kernel estimates for each run are also shown in Figure **??**. The mean reconstructed kernel is

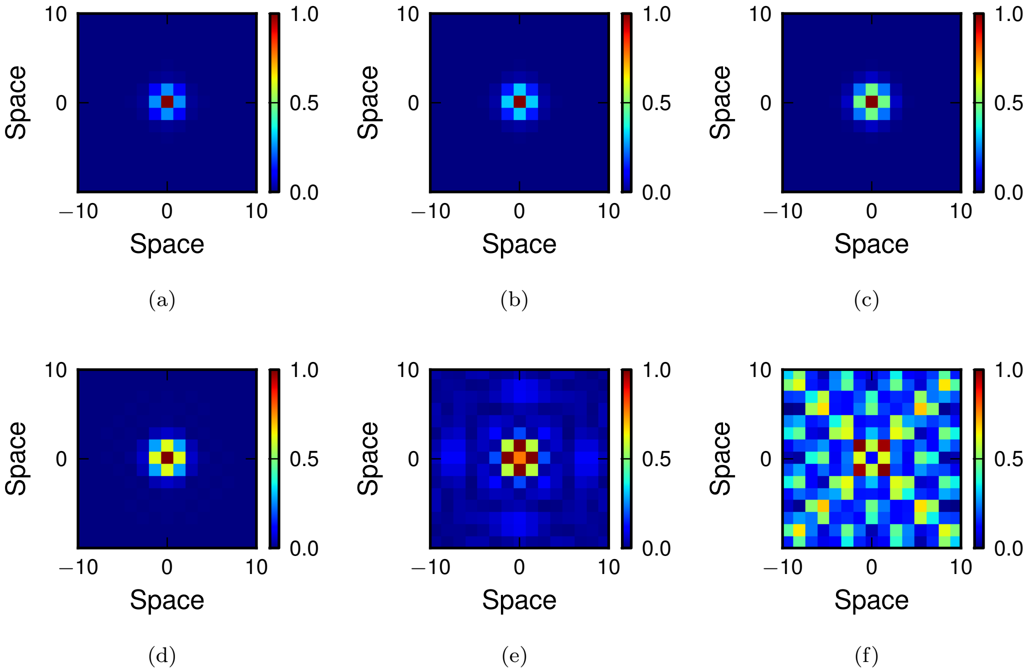


Figure 1: **Disturbance support estimates for different values of observation noise variance.** Each subplot shows the support of the disturbance covariance function convolved twice with the sensor kernel. (**a**) σ_ε^2=0. (**b**) σ_ε^2=0.1 . (**c**) σ_ε^2=0.2 . (**d**) σ_ε^2=0.23 . (**e**) σ_ε^2=0.3 . (**f**) σ_ε^2=0.4 .

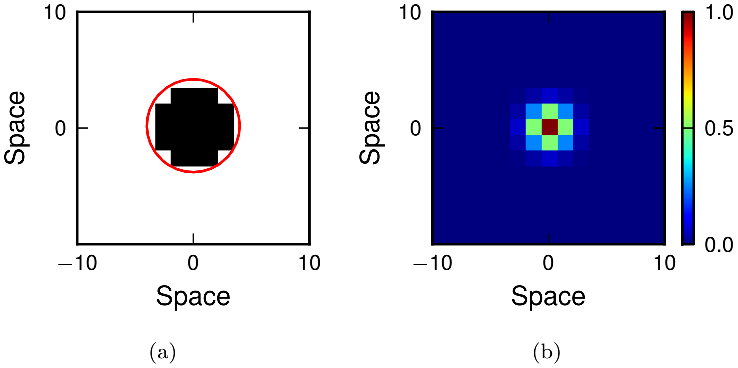


Figure 2: **Disturbance support approximation for** σ_ε^2 <0.23. Each subplot shows the support of the disturbance covariance function convolved twice with the sensor kernel. (**a**) Estimated support is shown by red circle, non-zero values after thresholding are shown by the black area. (**b**) True support.

reasonably accurate, where the actual kernel is inside the confidence interval. The large standard deviation is due to the high number of unknown parameters associated with the connectivity kernel decomposition. Figure **??** depicts the histogram of the non-kernel parameter estimates from 100 simulations where the true parameter values are shown by the solid lines and the mean parameter estimates are shown by the dotted lines. These histograms indicate small bias in the parameter estimates with small standard deviations of

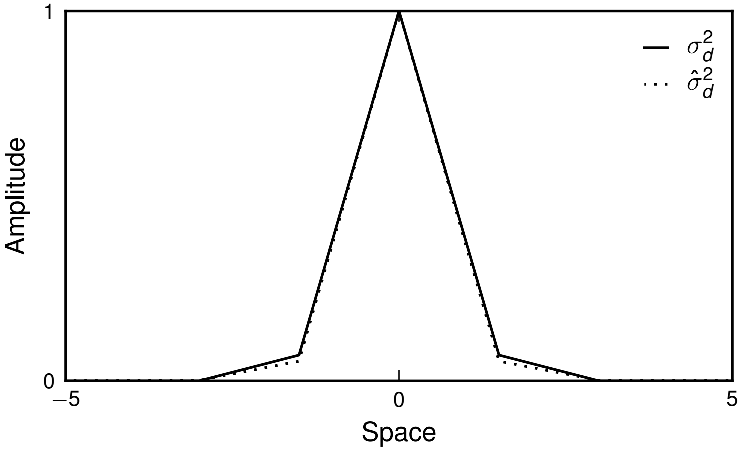
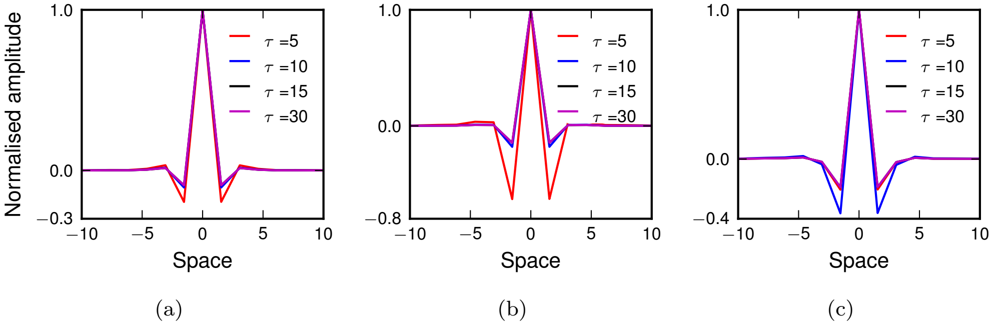


Figure 3: **Cross-section of normalised disturbance covariance function estimation**. The actual and the estimated covariance functions (obtained by numerical deconvolution via (**??**)) are shown by solid and dashed lines respectively.



(a)

(b)

(c)

Figure 4: **Kernel support approximation for different values of the synaptic time constant, τ and the observation noise variance, σ_ε^2.** Each subplot shows the normalised connectivity kernel estimate. (**a**) σ_ε^2=0. (**b**) σ_ε^2=0.1 . (**c**) σ_ε^2=0.18.

0.004, 0.0013 and 0.0012 for Figure **??**(a), (b) and (c) respectively. Examples of the neural field reconstruction are given in Figure **??**. The figure shows snapshots of the true and estimated neural fields at three different time instants. In each subplot the neural activity is plotted against the spatial index in one dimension to provide a better comparison between the true and the estimated underlying fields. Plot of the lower bound on the log-likelihood function versus iterations of the EM is shown in Figure **??**. In general, for non-linear dynamical systems the actual lower bound cannot be calculated analytically, therefore an approximate of the lower bound obtained by the URTSS is shown. The mean change in the lower bound in consecutive iterations dropped below 0.009 percent after 15 iterations, confirming that the algorithm converged to a steady value. To further investigate the convergence property of the algorithm the parameter estimates averaged over 100 realisations are also shown in Figure **??**, demonstrating the convergence of the EM algorithm.

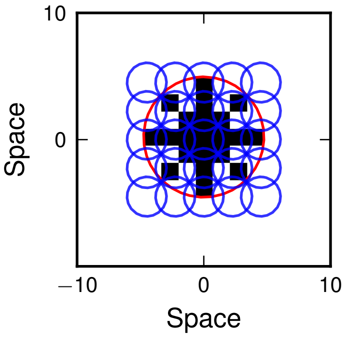
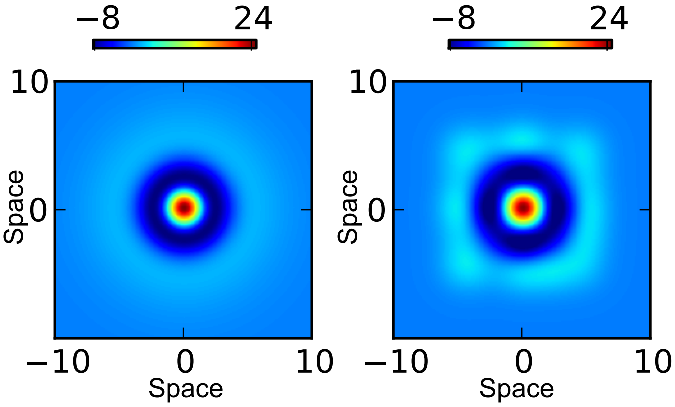


Figure 5: **Connectivity kernel decomposition.** The approximate support of the kernel is shown by the red circle. Black area shows the non-zero values of the support after thresholding. A 5 × 5 grid of basis functions (blue circles) used to reconstruct the connectivity kernel.



(a)

Figure 6: **Connectivity kernel estimates.** (A) Actual kernel. (B) The mean kernel estimate over 100 realisations.

5 Discussion

This paper is an extension of previous work in developing a framework for subject-specific neural field modelling. The extensions of the previous work include a method for estimating the support of the connectivity kernel and disturbance covariance, and the development of an EM framework for the neural field equations using the URTSS. The EM framework combined with the estimation of the support of the disturbance enables estimation of the temporal disturbance and noise characteristics of the system. Furthermore, by estimating the support of the connectivity kernel we are able to relax the strong assumption of isotropy in the estimated connectivity structure.

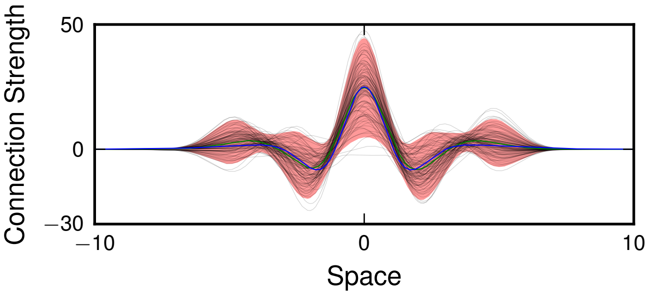
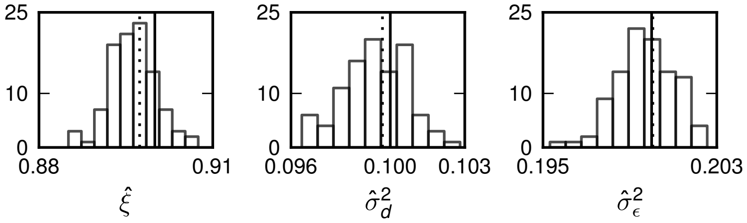


Figure 7: **Cross-sections of connectivity kernel estimates.** Actual kernel and the mean kernel estimates are shown with blue and green lines respectively. Each of the estimate from Monte Carlo simulation is shown with black line. The 95% confidence intervals are shown by the shaded red area.



(a)

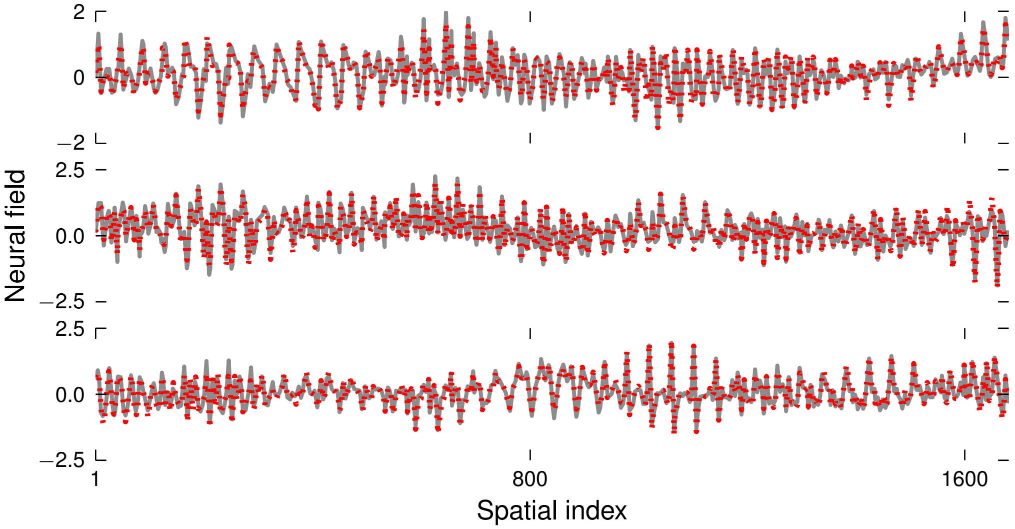
Figure 8: **Histograms of the non-kernel parameter estimates over 100 realizations**. The actual and mean of the estimated parameter values are shown by solid and dotted lines respectively. (**a**) The histogram of the synaptic dynamic parameter, ξ. (**b**) The histogram of the observation noise variance, σ_ϵ. (**c**) The histogram of the disturbance noise variance, σ_d^2.

5.1 Correlation Analysis

The correlation analysis specifies an approximation for the support of both the disturbance covariance function and the connectivity kernel. Although the unknown parameters ξ and σ_ε^2 can be set as described in §**??**, the chosen values influence the exact shape of the kernel significantly.

The derivation for the closed-form solution of the kernel requires linearisation of the sigmoidal activation function about the firing threshold. Therefore, the resulting kernel estimate is simply used as a guide to constrain the placement of basis function and the corresponding coefficients must still be inferred using the EM algorithm for an accurate representation of the connectivity structure. In fact, the correlation analysis can be considered a prerequisite for the EM algorithm described in §**??**, where the approximated supports are used to estimate the gain of the kernel and the field disturbance variance.

An alternative to estimating the support of the kernel is to place a regular grid of connectivity kernel basis function over the entire spatial domain of the field. However, the estimate of the spatial extent of the connectivity kernel greatly improves the speed of the estimation algorithm as the number of basis functions can be significantly reduced. This also improves the uncertainty in the estimated coefficients, as a lower number of unknown parameters are estimated. The approximated support of the disturbance covariance function facilitates the application of the EM algorithm for estimating the temporal variance in the disturbance signal. This way the disturbance covariance matrix can be decomposed into two parts: an unknown scalar and a constant matrix depending only on the inferred spatial support. Therefore, the estimation problem of the disturbance covariance matrix breaks down to the estimation of a single scalar



(a)

Figure 9: **Neural field estimates at different time instants**. In each subplot, true (grey line) and reconstructed neural field (red dots) are plotted versus the spatial index.

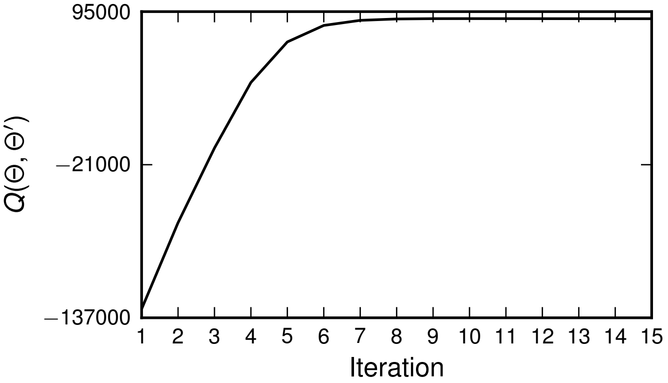
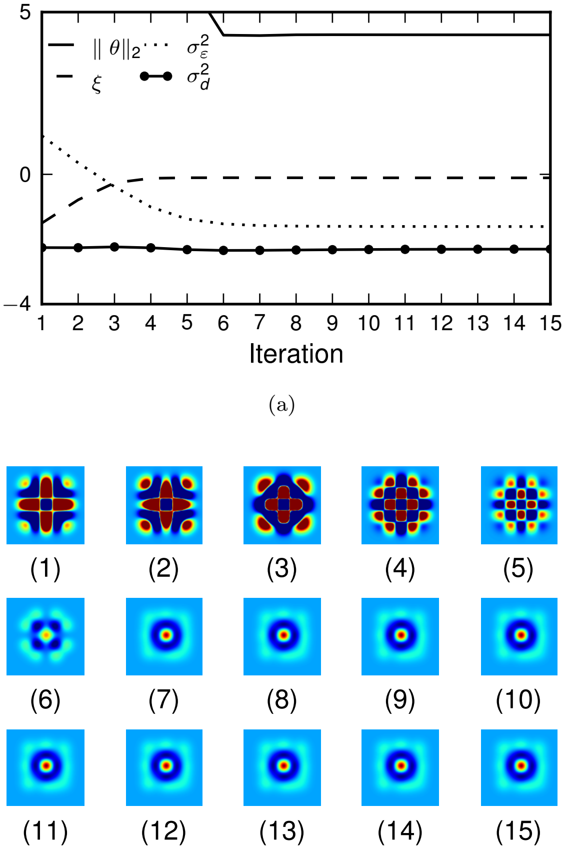


Figure 10: **Convergence of the EM algorithm.** The mean of the lower bound of the log-likelihood function over 100 realisations. The change in lower bound drops below 0.009 percent after 15 iterations; $Q(\mathbf{\Theta}', \mathbf{\Theta}') = -(T-1)\left[n_y(1+\ln \hat{\sigma}_\epsilon^2) + n_x(1+\ln \hat{\sigma}_d^2)\right]$.

parameter, improving the accuracy and the convergence property of the algorithm.

5.2 Development of the EM for the URTSS

The additional Taylor approximation step introduced in the E-step of the algorithm resembles the extended Kalman filter (EKF) [**?**, **?**]. The EKF approximates a stationary nonlinear dynamical system with a linear nonstationary one using the first-order Taylor expansion around the state estimates. This might imply that the extended Kalman smoother (EKS) is a better candidate — in terms of implementation — for the state inference in the proposed algorithm. However, the EKS requires re-linearising about the backward estimates, which in turn necessitates the nonlinear mapping to be uniquely invertible. Although this is indeed the case



(a)

(b)

Figure 11: **Convergence of parameters**. All parameters converge after 9 iterations. (**a**) The logarithm of the mean of synaptic parameter, observation noise variance, field disturbance variance and L_2 norm of the connectivity kernel parameter vector over 100 realisations. (**b**) The mean kernel estimates at each iteration of the EM algorithm.

in the neural field equation employed in this work it will impose more computational complexity in the state estimation. In addition, the URTSS provides better state estimations compared to the EKS [**?**] and therefore the constructed lower bound will be more accurate, resulting in a better parameter estimation.

While the EKS has been applied in the EM algorithm for learning nonlinear models [**?**], this is the first work (to the best author's knowledge) where the URTSS is used in this context. However, as in the case of the EKS, the stability of the algorithm cannot be ensured. This is because the EKS and the URTSS provide an approximation to the exact E-step. Following this, the maximisation of the approximated lower bound is not guaranteed to increase the log-likelihood function. Nevertheless, it is envisaged that the use of the EM algorithm using the URTSS will provide a more stable algorithm due to the superior performance of the UKF over the EKF.

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 saddle point, a small random perturbation diverges the algorithm from the local maxima [**?**]. In general, convergence of the EM algorithm to any stationary point (ie. global maxima, local maxima or saddle point) depends on the initialisation. In the examples presented in this paper, 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 first-order Taylor expansion provides a good local approximation of the sigmoid function, thus the EM provides a good approximation to the lower bound of the likelihood function. However, in a general settings with a more complicated nonlinearity, higher-order terms may be required to provide a good approximation. However, using higher-order terms introduces intractable computational complexity.

6 Conclusion

In this chapter, a method has been developed to estimate the states and parameters of a spatially continuous neural field model from electrophysiological data. The novel aspects of this chapter include an estimator for the kernel spatial support and the field disturbance characteristics. The derivations and computational results demonstrate that by choosing the synaptic time constant and the observation noise variance within a plausible range, the unknown supports can be identified. The inclusion of this step facilitates the use of the EM algorithm, where the complexity is decreased and the accuracy is improved by reducing the number of unknown variables. In addition to the connectivity kernel weights and the synaptic parameter, the observation noise and disturbance variances were identified. The results using simulated data demonstrated good estimation performance, where the distributions of the estimated non-kernel parameters showed small bias with very small standard deviations.

The data-driven modelling framework presented in this chapter provides a much needed link between theoretical and experimental neuroscience. By establishing this link, patient-specific systems neuroscience may eventually be used in a clinical setting, which has the potential to revolutionise neurosciences and neurology.

A Product of two n-dimensional Gaussian functions

In this section, we provide a derivation for the product of two n-dimensional Gaussian basis functions. This derivation is used in the calculation of $\mathbf{\Lambda}_t^{(i)}$ defined in equation (**??**) of the main text. Consider two Gaussian basis functions

$$\varphi_i(\mathbf{r}) = \exp\left(-\frac{1}{\sigma_i^2}(\mathbf{r}-\boldsymbol{\mu}_i)^\top(\mathbf{r}-\boldsymbol{\mu}_i)\right) \tag{144}$$

and

$$\varphi_j(\mathbf{r}) = \exp\left(-\frac{1}{\sigma_j^2}(\mathbf{r}-\boldsymbol{\mu}_j)^\top(\mathbf{r}-\boldsymbol{\mu}_j)\right). \tag{145}$$

the product of two Gauusian basis functions is given by

$$\varphi_i(\mathbf{r})\varphi_j(\mathbf{r}) = \exp\left(-\left[\frac{1}{\sigma_i^2}(\mathbf{r}-\boldsymbol{\mu}_i)^\top(\mathbf{r}-\boldsymbol{\mu}_i) + \frac{1}{\sigma_j^2}(\mathbf{r}-\boldsymbol{\mu}_j)^\top(\mathbf{r}-\boldsymbol{\mu}_j)\right]\right) \tag{146}$$

the product is expanded to give

$$\begin{aligned}
\varphi_i(\mathbf{r})\varphi_j(\mathbf{r}) &= \exp\left(-\left[\frac{(\sigma_i^2+\sigma_j^2)\left[\mathbf{r}^\top\mathbf{r} - 2\mathbf{r}^\top\frac{\sigma_j^2\boldsymbol{\mu}_i+\sigma_i^2\boldsymbol{\mu}_j}{\sigma_i^2+\sigma_j^2} + \frac{(\sigma_j^2\boldsymbol{\mu}_i+\sigma_i^2\boldsymbol{\mu}_j)^\top(\sigma_j^2\boldsymbol{\mu}_i+\sigma_i^2\boldsymbol{\mu}_j)}{(\sigma_i^2+\sigma_j^2)^2}\right]}{\sigma_i^2\sigma_j^2}\right.\right. \\
&\qquad \left.\left.\frac{+\frac{\sigma_i^2\sigma_j^2}{\sigma_i^2+\sigma_j^2}(\boldsymbol{\mu}_i-\boldsymbol{\mu}_j)^\top(\boldsymbol{\mu}_i-\boldsymbol{\mu}_j)}{}\right]\right) \\
&= \exp\left(-\frac{(\sigma_i^2+\sigma_j^2)\left[(\mathbf{r}-\frac{\sigma_j^2\boldsymbol{\mu}_i+\sigma_i^2\boldsymbol{\mu}_j}{\sigma_i^2+\sigma_j^2})^\top(\mathbf{r}-\frac{\sigma_j^2\boldsymbol{\mu}_i+\sigma_i^2\boldsymbol{\mu}_j}{\sigma_i^2+\sigma_j^2})\right]}{\sigma_i^2\sigma_j^2}\right) \\
&\times \exp\left(-\frac{(\boldsymbol{\mu}_i-\boldsymbol{\mu}_j)^\top(\boldsymbol{\mu}_i-\boldsymbol{\mu}_j)}{\sigma_i^2+\sigma_j^2}\right),
\end{aligned} \tag{147}$$

therefore we have

$$\varphi_i(\mathbf{r})\varphi_j(\mathbf{r}) = c_{i,j} \times \exp\left(-\frac{1}{\sigma^2}(\mathbf{r}-\boldsymbol{\mu})^\top(\mathbf{r}-\boldsymbol{\mu})\right) \tag{148}$$

where

$$c_{i,j} = \exp\left(-\frac{(\boldsymbol{\mu}_i-\boldsymbol{\mu}_j)^\top(\boldsymbol{\mu}_i-\boldsymbol{\mu}_j)}{\sigma_i^2+\sigma_j^2}\right) \qquad \sigma^2 = \frac{\sigma_i^2\sigma_j^2}{\sigma_i^2+\sigma_j^2} \tag{149}$$

and

$$\boldsymbol{\mu} = \frac{\sigma_j^2\boldsymbol{\mu}_i+\sigma_i^2\boldsymbol{\mu}_j}{\sigma_i^2+\sigma_j^2} \tag{150}$$

B Cross-correlation and convolution

This appendix provide a proof for the properties of the cross-correlation and convolution used in the Correlation analysis Section of Chapter 2. To show

$$(a * b)(\boldsymbol{\iota}) \star c(\boldsymbol{\iota}) = a(-\boldsymbol{\iota}) * (b \star c)(\boldsymbol{\iota}) \tag{151}$$

and

$$(a * b)(\boldsymbol{\iota}) \star (a * b)(\boldsymbol{\iota}) = (a \star a)(\boldsymbol{\iota}) * (b \star b)(\boldsymbol{\iota}), \tag{152}$$

first note that cross-correlation function is related to the convolution by [?]

$$(a \star b)\left(\boldsymbol{\iota}\right) = a\left(-\boldsymbol{\iota}\right) * b\left(\boldsymbol{\iota}\right). \tag{153}$$

Therefore, equation (**??**) can be wriiten as

$$\begin{aligned}(a * b)\left(\boldsymbol{\iota}\right) \star c\left(\boldsymbol{\iota}\right) &= (a * b)\left(-\boldsymbol{\iota}\right) * c\left(\boldsymbol{\iota}\right) \\ &= a\left(-\boldsymbol{\iota}\right) * \left(b\left(-\boldsymbol{\iota}\right) * c\left(\boldsymbol{\iota}\right)\right) \\ &= a\left(-\boldsymbol{\iota}\right) * (b \star c)\left(\boldsymbol{\iota}\right). \end{aligned} \tag{154}$$

Similarly, equation (**??**) can be written as

$$\begin{aligned}(a * b)(\boldsymbol{\iota}) \star (a * b)(\boldsymbol{\iota}) &= (a * b)(-\boldsymbol{\iota}) * (a * b)(\boldsymbol{\iota}) \\ &= a(-\boldsymbol{\iota}) * a(\boldsymbol{\iota}) * b(-\boldsymbol{\iota}) * b(\boldsymbol{\iota}) \\ &= (a \star a)(\boldsymbol{\iota}) * (b \star b)(\boldsymbol{\iota}). \end{aligned} \tag{155}$$

Table 2: **Algorithm for the Unscented RTS Smoother**

1. Forward initialisation

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

2. Forward iteration: for $t \in \{0, \cdots, T\}$, calculate the sigma points $\mathcal{X}^{f}_{i,t}$ using equations **??**-**??** and propagate through equation **??**

$$\mathcal{X}^{f-}_{i,t+1} = Q(\mathcal{X}^{f}_{i,t}) \quad i = 0, \ldots, 2L$$

Calculate the predicted state and the predicted covariance matrix

$$\hat{\mathbf{x}}^{f-}_{t+1} = \textstyle\sum_{i=0}^{2L} W_i^{(m)} \mathcal{X}^{f-}_{i,t+1}$$

$$\mathbf{P}^{f-}_{t+1} = \textstyle\sum_{i=0}^{2L} W_i^{(c)} (\mathcal{X}^{f-}_{i,t+1} - \hat{\mathbf{x}}^{f-}_{t+1})(\mathcal{X}^{f-}_{i,t+1} - \hat{\mathbf{x}}^{f-}_{t+1})^\top + \mathbf{\Sigma}_e$$

Compute the filter gain, the filtered state and the filtered covariance matrix using the standard Kalman filter update equations

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

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

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

3. Backward initialisation

$$\mathbf{P}^b_T = \mathbf{P}^f_T, \quad \hat{\mathbf{x}}^b_T = \hat{\mathbf{x}}^f_T$$

4. Backward iteration: for $t \in \{T-1, \cdots, 0\}$ calculate the sigma points $\mathcal{X}^{b}_{i,t}$ and propagate them through equation **??**

$$\mathcal{X}^{b-}_{i,t+1} = Q(\mathcal{X}^{b}_{i,t}) \quad i = 0, \ldots, 2L$$

Calculate the predicted state and the predicted covariance matrix

$$\hat{\mathbf{x}}^{b-}_{t+1} = \textstyle\sum_{i=0}^{2L} W_i^{(m)} \mathcal{X}^{b-}_{i,t+1}$$

$$\mathbf{P}^{b-}_{t+1} = \textstyle\sum_{i=0}^{2L} W_i^{(c)} (\mathcal{X}^{b-}_{i,t+1} - \hat{\mathbf{x}}^{b-}_{t+1})(\mathcal{X}^{b-}_{i,t+1} - \hat{\mathbf{x}}^{b-}_{t+1})^\top + \mathbf{\Sigma}_e$$

$$\mathbf{M}^{b-}_{t+1} = \textstyle\sum_{i=0}^{2L} W_i^{(c)} (\mathcal{X}^{b-}_{i,t} - \hat{\mathbf{x}}^{f}_{t})(\mathcal{X}^{b-}_{i,t+1} - \hat{\mathbf{x}}^{b-}_{t+1})^\top$$

Compute the smoother gain, the smoothed state and the smoothed covariance matrix

$$\mathbf{S}_t = \mathbf{M}^{b-}_{t+1}\left[\mathbf{P}^{b-}_{t+1}\right]^{-1}$$

$$\hat{\mathbf{x}}^b_t = \hat{\mathbf{x}}^f_t + \mathbf{S}_t\left[\hat{\mathbf{x}}^b_{t+1} - \hat{\mathbf{x}}^{b-}_{t+1}\right]$$

$$\mathbf{P}^b_t = \mathbf{P}^f_t + \mathbf{S}_t\left[\mathbf{P}^b_{t+1} - \mathbf{P}^{b-}_{t+1}\right]\mathbf{S}_t^\top$$

Additional step to calculate the smoothed cross covariance matrix [**?**]

$$\mathbf{M}^b_{t+1} = \mathbf{S}_t\mathbf{P}^b_{t+1}$$

This table shows the steps in the unscented Rauch-Tung-Striebel smoother algorithm. The steps are iterated 11 times for our state estimation procedure. The least squares algorithm is run after each iteration to update the parameter estimates.