Data-driven neural field modelling

Dean R. Freestone
Supervisors: David B. Grayden, Anthony N. Burkitt, Levin Kuhlmann and Mark J. Cook

BEI, SVHM, UoM (DEEE)

25-03-2011

Contents lists available at ScienceDirect

NeuroImage

journal homepage: www.elsevier.com/locate/ynimg

A data-driven framework for neural field modeling

D.R. Freestone [a,b,c,*,1], P. Aram [d,1], M. Dewar [c,e], K. Scerri [f], D.B. Grayden [a,b], V. Kadirkamanathan [d]

[a] Department of Electrical and Electronic Engineering, University of Melbourne, Melbourne, VIC, Australia
[b] The Bionic Ear Institute, East Melbourne, VIC, Australia
[c] Institute for Adaptive and Neural Computation, University of Edinburgh, Edinburgh, UK
[d] Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, UK
[e] Department of Applied Physics and Applied Mathematics, Columbia University, New York, NY, USA
[f] Department of Systems and Control Engineering, University of Malta, Msida, MSD, Malta

ARTICLE INFO

Article history:
Received 23 November 2010
Revised 18 January 2011
Accepted 9 February 2011
Available online xxxx

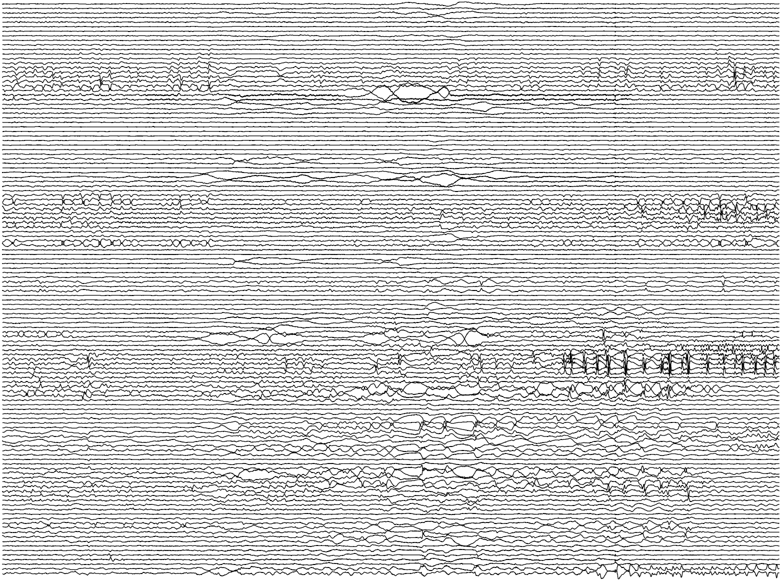
Keywords:
Neural field model
Nonlinear estimation
Intracortical connectivity
Nonlinear dynamics

ABSTRACT

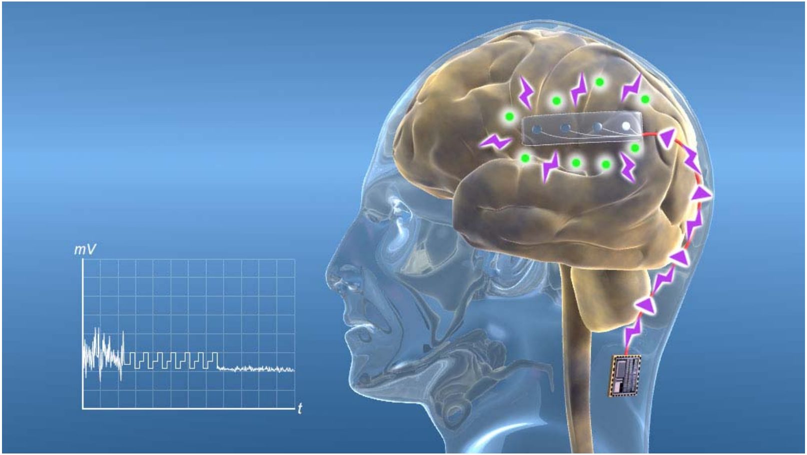
This paper presents a framework for creating neural field models from electrophysiological data. The Wilson and Cowan or Amari style neural field equations are used to form a parametric model, where the parameters are estimated from data. To illustrate the estimation framework, data is generated using the neural field equations incorporating modeled sensors enabling a comparison between the estimated and true parameters. To facilitate state and parameter estimation, we introduce a method to reduce the continuum neural field model using a basis function decomposition to form a finite-dimensional state-space model. Spatial frequency analysis methods are introduced that systematically specify the basis function configuration required to capture the dominant characteristics of the neural field. The estimation procedure consists of a two-stage iterative algorithm incorporating the unscented Rauch–Tung–Striebel smoother for state estimation and a least squares algorithm for parameter estimation. The results show that it is theoretically possible to reconstruct the neural field and estimate intracortical connectivity structure and synaptic dynamics with the proposed framework.

Outline

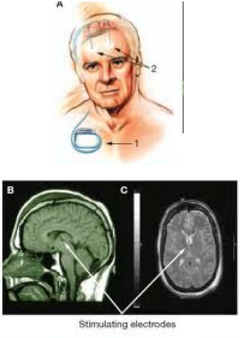
Motivation

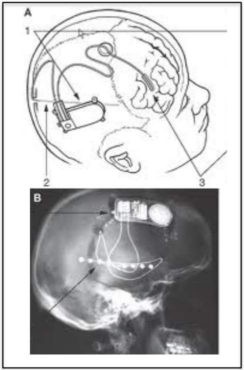


Electrical stimulation as therapy



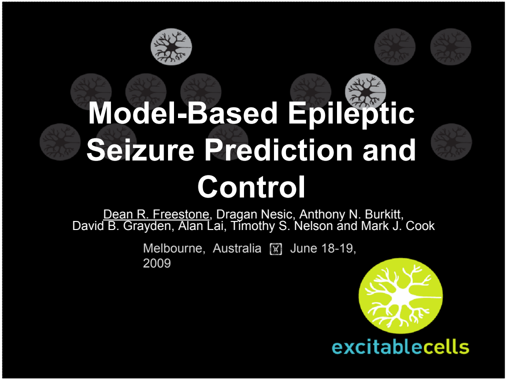
Trial and error (open loop) stimulation strategy

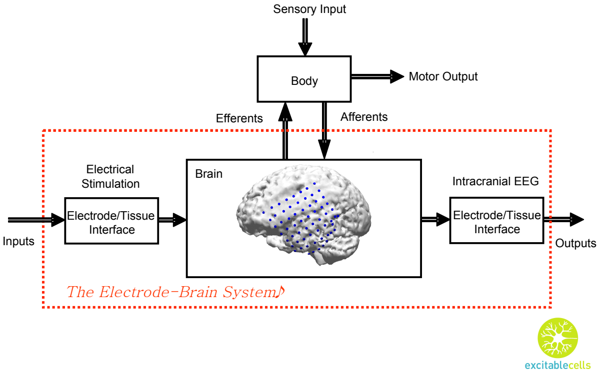


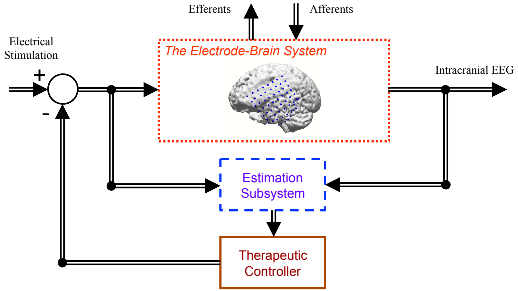


Parkinson’s (persistent) vs. epilepsy (intermittent)

Inception


Model-Based Epileptic Seizure Prediction and Control
Dean R. Freestone, Dragan Nesic, Anthony N. Burkitt, David B. Grayden, Alan Lai, Timothy S. Nelson and Mark J. Cook
Melbourne, Australia June 18-19, 2009
excitablecells

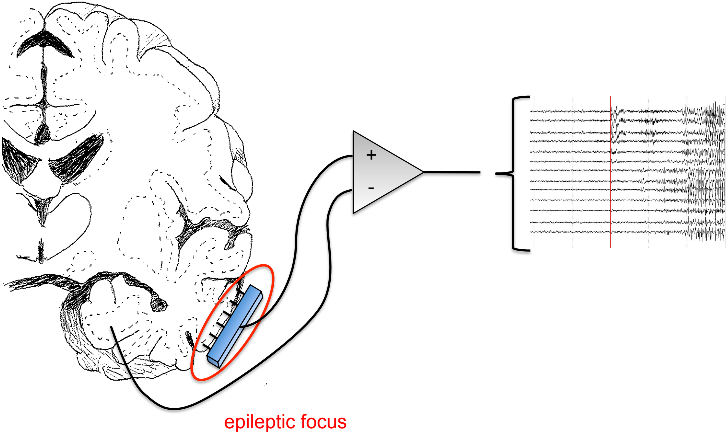

Sensory Input
Body
Motor Output
Efferents
Afferents
Electrical Stimulation
Brain
Intracranial EEG
Inputs
Electrode/Tissue Interface
Electrode/Tissue Interface
Outputs
The Electrode–Brain System
excitablecells



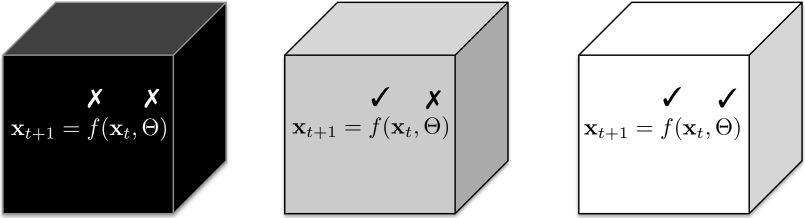
Closed-Loop Therapeutic Feedback
via Electrical Stimulation

Patient specific model

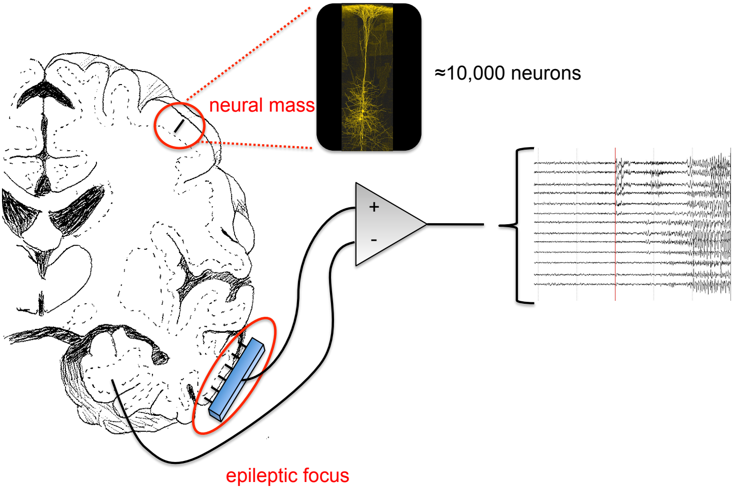
Aim: To track the patient-specific neural dynamics and estimate the parameters that influence seizures from electrophysiological measurements.



Modelling approaches



Neural mass



Neural mass

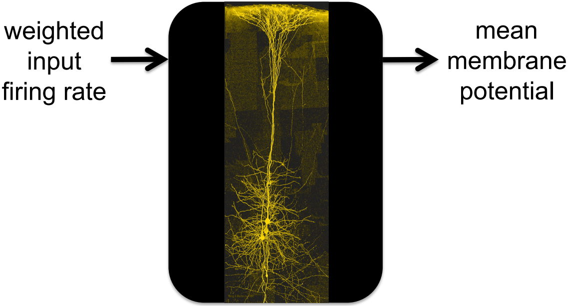
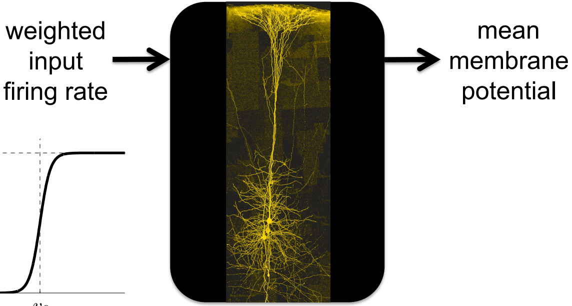
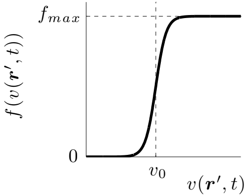


Photo: Ecole Polytechnique Federale de Lausanne, HPC photo gallery, *Golden Pyramids in Columns*

Neural mass



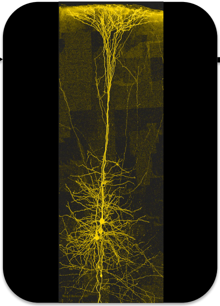


$$g(\mathbf{r}, t) = w_{\mathbf{r},\mathbf{r}'} f\left(v\left(\mathbf{r}', t\right)\right) \tag{1}$$

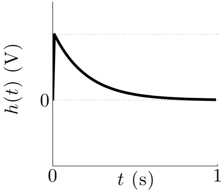
$$f\left(v\left(\mathbf{r}', t\right)\right) = \frac{1}{1 + \exp\left(\varsigma\left(v_0 - v\left(\mathbf{r}', t\right)\right)\right)} \tag{2}$$

Neural mass

weighted
input
firing rate



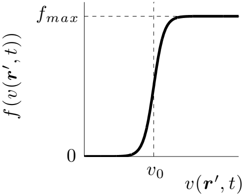
mean
membrane
potential

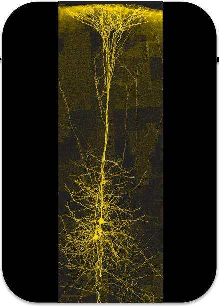


$$h(t) = \eta(t) \exp\left(-\frac{t}{\tau}\right) \qquad (3)$$

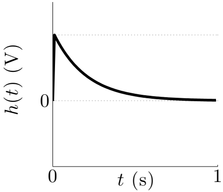
Neural mass

weighted
input
firing rate



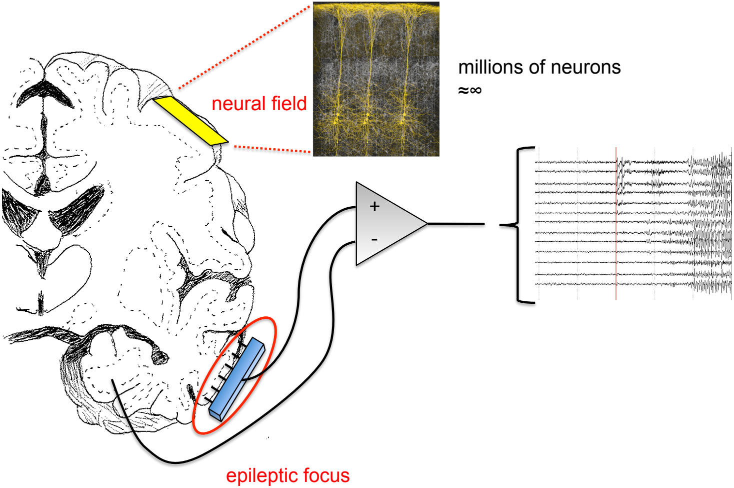


mean
membrane
potential



$$\underbrace{v(\mathbf{r},t)}_{\text{potential}} = \int_{-\infty}^{t} \underbrace{h(t-t')}_{\text{synaptic}}\, \overbrace{w_{\mathbf{r},\mathbf{r}'}}^{\text{weight}}\, \underbrace{f(v(\mathbf{r}',t))}_{\text{firing}}\, dt' \tag{4}$$

Neural field



Neural field model

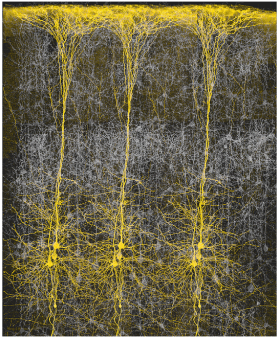
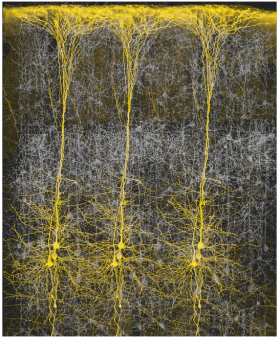
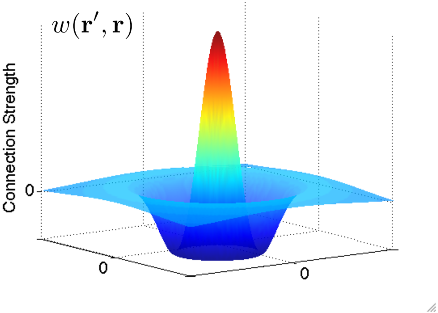


Photo: Ecole Polytechnique Federale de Lausanne, HPC photo gallery, *Golden Pyramids in Columns*

Neural field model

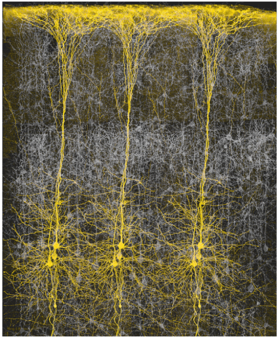


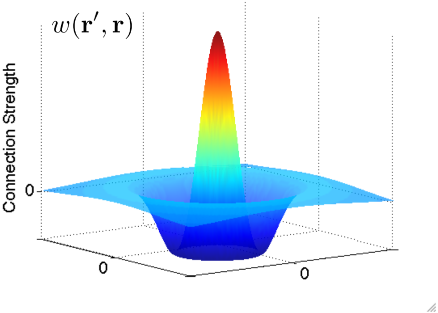


$$\underbrace{v\left(\mathbf{r}, t\right)}_{\text{potential}} = \int_{-\infty}^{t} \underbrace{h\left(t - t'\right)}_{\text{synaptic}} \overbrace{w\left(\mathbf{r}, \mathbf{r}'\right)}^{\text{connectivity}} \underbrace{f\left(v\left(\mathbf{r}', t\right)\right)}_{\text{firing}} dt' \qquad (5)$$

Photo: Ecole Polytechnique Federale de Lausanne, HPC photo gallery, *Golden Pyramids in Columns*

Neural field model





$$\underbrace{v(\mathbf{r},t)}_{\text{potential}} = \int_{-\infty}^{t} \underbrace{h(t-t')}_{\text{synaptic}}\, \overbrace{w(\mathbf{r},\mathbf{r}')}^{\text{connectivity}}\, \underbrace{f(v(\mathbf{r}',t))}_{\text{firing}}\, dt' \qquad (5)$$

$$\frac{dv(\mathbf{r},t)}{dt} = -\frac{1}{\tau} v(\mathbf{r},t) + \int_{\Omega} w(\mathbf{r},\mathbf{r}')\, f(v(\mathbf{r}',t))\, d\mathbf{r}' \qquad (6)$$

Photo: Ecole Polytechnique Federale de Lausanne, HPC photo gallery, *Golden Pyramids in Columns*

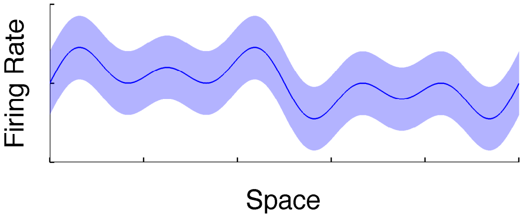
Discrete time stochastic neural field model

$$\underbrace{v_{t+T_s}(\mathbf{r})}_{\text{dynamics}} = \underbrace{\xi v_t(\mathbf{r})}_{\text{synaptic}} + T_s \int_\Omega \underbrace{w(\mathbf{r},\mathbf{r}')}_{\text{connectivity}} \underbrace{f(v_t(\mathbf{r}'))}_{\text{firing}} d\mathbf{r}' + \underbrace{e_t(\mathbf{r})}_{\text{uncertainty}}, \quad (7)$$

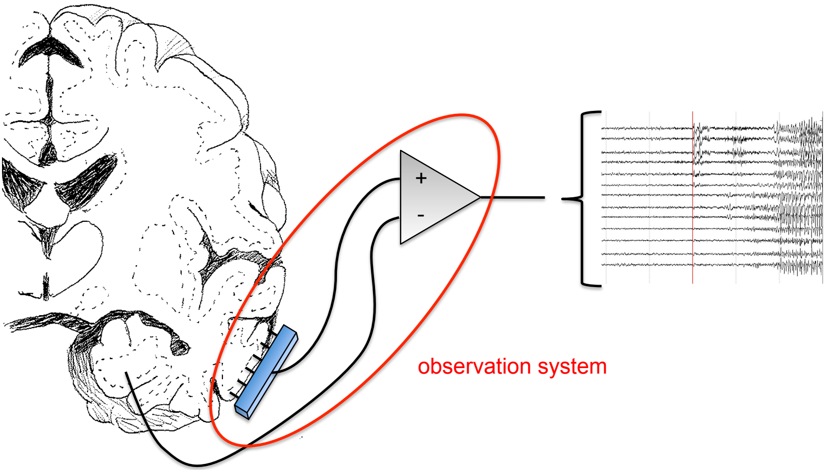
where

$$\xi = 1 - \frac{T_s}{\tau}$$

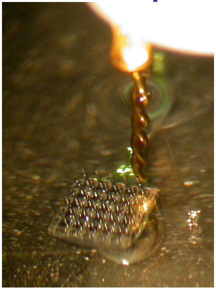
$$e_t(\mathbf{r}) \sim \mathcal{GP}(\mathbf{0}, \gamma(\mathbf{r}-\mathbf{r}'))$$



Observations



Observation equation



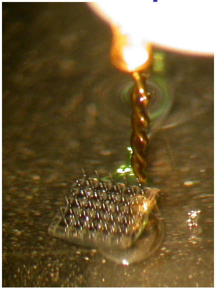
$$\underbrace{y_t(\mathbf{r}_n)}_{\text{observation}} = \int_\Omega \underbrace{m\left(\mathbf{r}_n - \mathbf{r}'\right)}_{\text{sensor}} \underbrace{v_t\left(\mathbf{r}'\right)}_{\text{potential}} d\mathbf{r}' + \underbrace{\varepsilon_t(\mathbf{r}_n)}_{\text{noise}}, \tag{8}$$

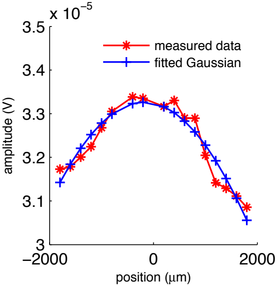
where

$$m\left(\mathbf{r} - \mathbf{r}'\right) = \exp\left(-\frac{(\mathbf{r} - \mathbf{r}')^\top (\mathbf{r} - \mathbf{r}')}{\sigma_m^2}\right) \tag{9}$$

$$\varepsilon_t(\mathbf{r}_n) \sim \mathcal{N}\left(0, \mathbf{\Sigma}_\varepsilon\right). \tag{10}$$

Observation equation





$$\underbrace{y_t(\mathbf{r}_n)}_{\text{observation}} = \int_\Omega \underbrace{m\left(\mathbf{r}_n - \mathbf{r}'\right)}_{\text{sensor}} \underbrace{v_t\left(\mathbf{r}'\right)}_{\text{potential}} d\mathbf{r}' + \underbrace{\varepsilon_t(\mathbf{r}_n)}_{\text{noise}}, \tag{8}$$

where

$$m\left(\mathbf{r} - \mathbf{r}'\right) = \exp\left(-\frac{(\mathbf{r} - \mathbf{r}')^\top (\mathbf{r} - \mathbf{r}')}{\sigma_m^2}\right) \tag{9}$$

$$\varepsilon_t(\mathbf{r}_n) \sim \mathcal{N}\left(0, \mathbf{\Sigma}_\varepsilon\right). \tag{10}$$

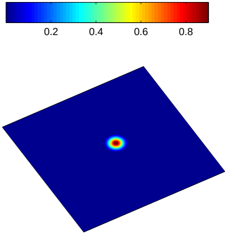
Observation equation

$$\underbrace{y_t(\mathbf{r}_n)}_{\text{observation}} = \int_\Omega \underbrace{m\left(\mathbf{r}_n - \mathbf{r}'\right)}_{\text{sensor}} \underbrace{v_t\left(\mathbf{r}'\right)}_{\text{potential}} d\mathbf{r}' + \underbrace{\varepsilon_t(\mathbf{r}_n)}_{\text{noise}}, \tag{8}$$

where

$$m\left(\mathbf{r} - \mathbf{r}'\right) = \exp\left(-\frac{(\mathbf{r} - \mathbf{r}')^\top(\mathbf{r} - \mathbf{r}')}{\sigma_m^2}\right) \tag{9}$$

$$\varepsilon_t(\mathbf{r}_n) \sim \mathcal{N}\left(0, \mathbf{\Sigma}_\varepsilon\right). \tag{10}$$



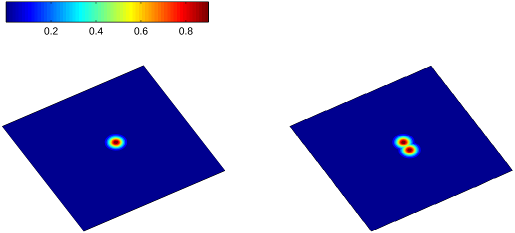
Observation equation

$$\underbrace{y_t(\mathbf{r}_n)}_{\text{observation}} = \int_\Omega \underbrace{m\left(\mathbf{r}_n - \mathbf{r}'\right)}_{\text{sensor}} \underbrace{v_t\left(\mathbf{r}'\right)}_{\text{potential}} d\mathbf{r}' + \underbrace{\varepsilon_t(\mathbf{r}_n)}_{\text{noise}}, \tag{8}$$

where

$$m\left(\mathbf{r} - \mathbf{r}'\right) = \exp\left(-\frac{(\mathbf{r}-\mathbf{r}')^\top(\mathbf{r}-\mathbf{r}')}{\sigma_m^2}\right) \tag{9}$$

$$\varepsilon_t(\mathbf{r}_n) \sim \mathcal{N}\left(0, \mathbf{\Sigma}_\varepsilon\right). \tag{10}$$



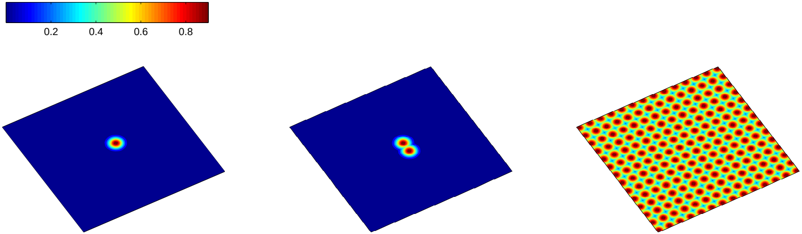
Observation equation

$$\underbrace{y_t(\mathbf{r}_n)}_{\text{observation}} = \int_\Omega \underbrace{m\left(\mathbf{r}_n - \mathbf{r}'\right)}_{\text{sensor}} \underbrace{v_t\left(\mathbf{r}'\right)}_{\text{potential}} d\mathbf{r}' + \underbrace{\varepsilon_t(\mathbf{r}_n)}_{\text{noise}}, \tag{8}$$

where

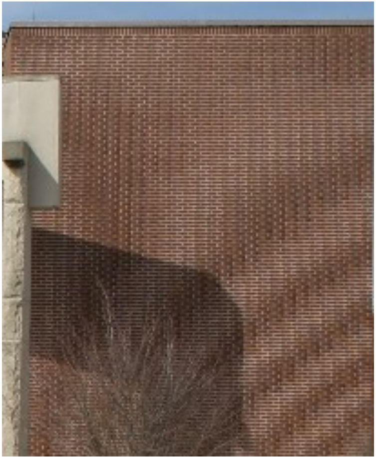
$$m\left(\mathbf{r} - \mathbf{r}'\right) = \exp\left(-\frac{(\mathbf{r} - \mathbf{r}')^\top (\mathbf{r} - \mathbf{r}')}{\sigma_m^2}\right) \tag{9}$$

$$\varepsilon_t(\mathbf{r}_n) \sim \mathcal{N}\left(0, \mathbf{\Sigma}_\varepsilon\right). \tag{10}$$



Spatial aliasing

Spatial aliasing

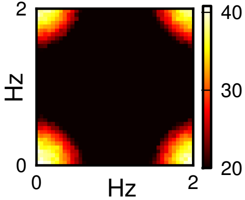


Positioning sensors

Assuming

$$V_t(\nu) \approx 0 \;\; \forall \nu > \nu_c, \tag{11}$$

where ν is the spatial frequency.



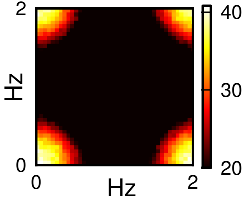
[1] Scerri et al. (2009) IEEE Trans. Sig. Proc. 57

Positioning sensors

Assuming

$$V_t(\nu) \approx 0 \;\; \forall \boldsymbol{\nu} > \boldsymbol{\nu}_c, \tag{11}$$

where ν is the spatial frequency.



$$\Delta_y \leq \frac{1}{2\rho_y \boldsymbol{\nu}_c}, \tag{12}$$

where $\rho_y \in \mathbb{R} \geq 1$. See Scerri et al. for more info[1].

[1] Scerri et al. (2009) IEEE Trans. Sig. Proc. 57

Field spatial frequency

- Freeman[2] estimated ν_c to be approximately 0.25 cycles/mm.

[2] Freeman et al. (2000) J. Neurosci. Meth. 95.

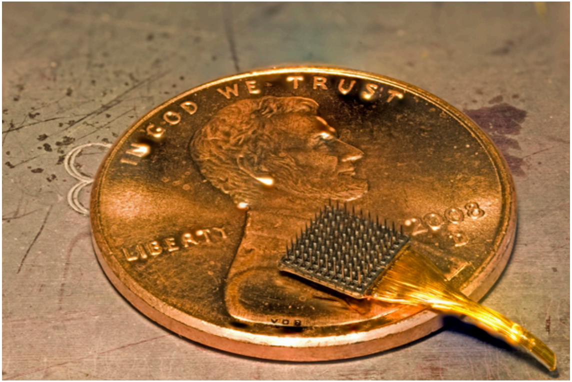
Field spatial frequency

- Freeman[2] estimated ν_c to be approximately 0.25 cycles/mm.
- This implies $\Delta_y \lesssim 1.25$ mm to prevent aliasing.

[2] Freeman et al. (2000) J. Neurosci. Meth. 95.

Field spatial frequency

- Freeman[2] estimated ν_c to be approximately 0.25 cycles/mm.
- This implies $\Delta_y \lesssim 1.25$ mm to prevent aliasing.
- Electrodes satisfying this requirement are in clinical use.

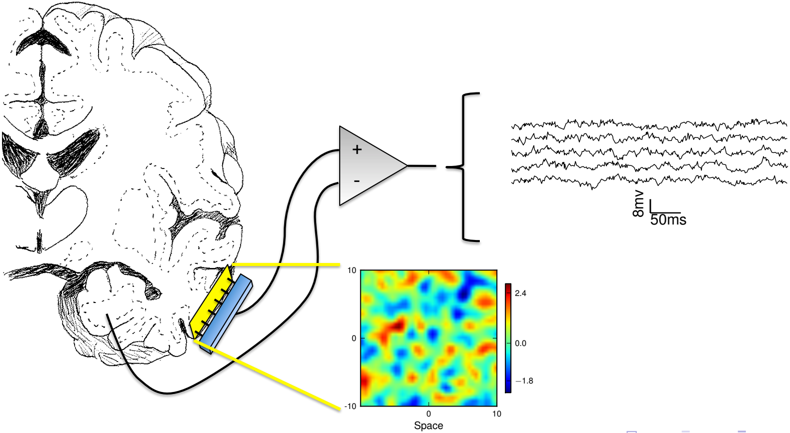


[2]Freeman et al. (2000) J. Neurosci. Meth. 95.

Model output

$$v_{t+T_s}(\mathbf{r}) = \xi v_t(\mathbf{r}) + T_s \int_\Omega w(\mathbf{r}, \mathbf{r}') f(v_t(\mathbf{r}'))\, d\mathbf{r}' + e_t(\mathbf{r})$$

$$y_t(\mathbf{r}_n) = \int_\Omega m(\mathbf{r}_n - \mathbf{r}') v_t(\mathbf{r}')\, d\mathbf{r}' + \varepsilon_t(\mathbf{r}_n)$$



Aim again

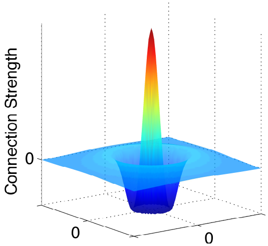
Aim: To track the patient-specific neural dynamics and estimate the parameters that influence seizures from electrophysiological measurements.

$$\underbrace{v_{t+T_s}(\mathbf{r})}_{\text{dynamics}} = \underbrace{\xi v_t(\mathbf{r})}_{\text{synaptic}} + T_s \int_\Omega \underbrace{w(\mathbf{r}, \mathbf{r}')}_{\text{connectivity}} \underbrace{f(v_t(\mathbf{r}'))}_{\text{firing}} d\mathbf{r}' + \underbrace{e_t(\mathbf{r})}_{\text{uncertainty}}$$

$$\underbrace{y_t(\mathbf{r}_n)}_{\text{observation}} = \int_\Omega \underbrace{m(\mathbf{r}_n - \mathbf{r}')}_{\text{sensor}} \underbrace{v_t(\mathbf{r}')}_{\text{potential}} d\mathbf{r}' + \underbrace{\varepsilon_t(\mathbf{r}_n)}_{\text{noise}}$$

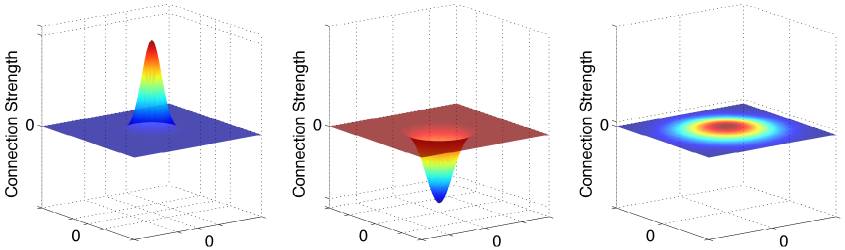
Connectivity kernel decomposition

$$w\left(\mathbf{r},\mathbf{r}'\right) = \psi^{\top}\left(\mathbf{r},\mathbf{r}'\right)\theta \tag{13}$$



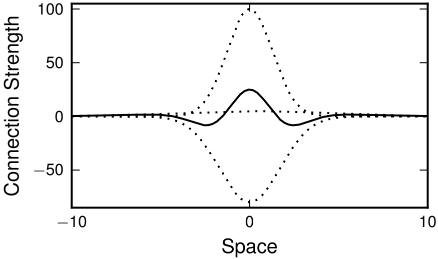
Connectivity kernel decomposition

$$w\left(\mathbf{r},\mathbf{r}'\right) = \psi^{\top}\left(\mathbf{r},\mathbf{r}'\right)\theta \tag{13}$$



Connectivity kernel decomposition

$$w\left(\mathbf{r},\mathbf{r}'\right) = \psi^{\top}\left(\mathbf{r},\mathbf{r}'\right)\theta \tag{13}$$



Aim again, again

Aim: To track the patient-specific neural dynamics and estimate the parameters that influence seizures from electrophysiological measurements.

$$v_{t+T_s}(\mathbf{r}) = \xi v_t(\mathbf{r}) + T_s \int_\Omega f\left(v_t\left(\mathbf{r}'\right)\right)\psi^\top\left(\mathbf{r},\mathbf{r}'\right) d\mathbf{r}'\boldsymbol{\theta} + e_t(\mathbf{r})$$

$$y_t(\mathbf{r}_n) = \int_\Omega m\left(\mathbf{r}_n - \mathbf{r}'\right) v_t\left(\mathbf{r}'\right) d\mathbf{r}' + \varepsilon_t(\mathbf{r}_n)$$

Field basis function decomposition

We can approximate the field by

$$v_t(\mathbf{r}) \approx \phi^\top(\mathbf{r})\,\mathbf{x}_t, \tag{14}$$

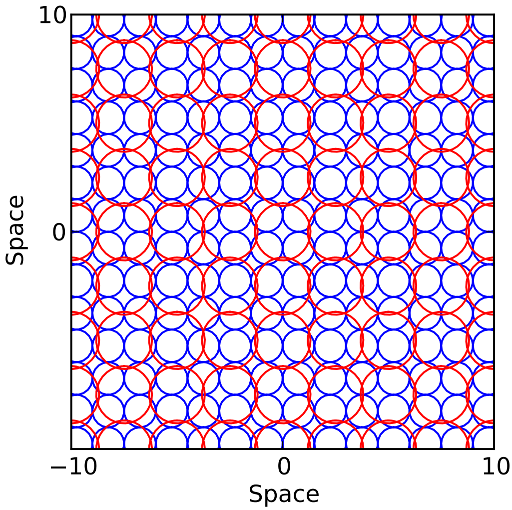
where

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

1D neural field decomposition with state

A movie was here

Sensor and field basis function position



Reduced system equation

We get a reduced system equation

$$\phi^{\top}(\mathbf{r})\mathbf{x}_{t+T_s} = \xi\phi^{\top}(\mathbf{r})\mathbf{x}_t + T_s \int_{\Omega} f\left(\phi^{\top}(\mathbf{r})\mathbf{x}_t\right)\psi^{\top}\left(\mathbf{r},\mathbf{r}'\right) d\mathbf{r}'\boldsymbol{\theta} + e_t\left(\mathbf{r}\right). \tag{15}$$

Reduced system equation

We get a reduced system equation

$$\phi^\top(\mathbf{r})\mathbf{x}_{t+T_s} = \xi\phi^\top(\mathbf{r})\mathbf{x}_t + T_s \int_\Omega f\left(\phi^\top(\mathbf{r})\mathbf{x}_t\right)\psi^\top\left(\mathbf{r},\mathbf{r}'\right) d\mathbf{r}'\boldsymbol{\theta} + e_t\left(\mathbf{r}\right). \tag{15}$$

This simplifies to

$$\mathbf{x}_{t+1} = \xi\mathbf{x}_t + \int_\Omega \boldsymbol{\Psi}(\mathbf{r}')f(\phi^\top(\mathbf{r}')\mathbf{x}_t)\, d\mathbf{r}'\boldsymbol{\theta} + \mathbf{e}_t, \tag{16}$$

Reduced system equation

We get a reduced system equation

$$\phi^\top(\mathbf{r})\mathbf{x}_{t+T_s} = \xi\phi^\top(\mathbf{r})\mathbf{x}_t + T_s\int_\Omega f\left(\phi^\top(\mathbf{r})\mathbf{x}_t\right)\psi^\top\left(\mathbf{r},\mathbf{r}'\right)d\mathbf{r}'\theta + e_t\left(\mathbf{r}\right). \tag{15}$$

This simplifies to

$$\mathbf{x}_{t+1} = \xi\mathbf{x}_t + \int_\Omega \mathbf{\Psi}(\mathbf{r}')f(\phi^\top(\mathbf{r}')\mathbf{x}_t)\,d\mathbf{r}'\theta + \mathbf{e}_t, \tag{16}$$

where

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

$$\mathbf{\Gamma} \triangleq \int_\Omega \phi\left(\mathbf{r}\right)\phi^\top\left(\mathbf{r}\right)\,d\mathbf{r} \tag{18}$$

$$\left[\mathbf{\Psi}(\mathbf{r}')\right]_{:i} \triangleq T_s\mathbf{\Gamma}^{-1}\int_\Omega \phi(\mathbf{r})\psi_i(2\mathbf{c}_i + \mathbf{r}' - \mathbf{r})\mathrm{d}\mathbf{r}. \tag{19}$$

Reduced observation equation

$$y_t(\mathbf{r}_n) = \int_\Omega m\left(\mathbf{r}_n - \mathbf{r}'\right)\phi^\top(\mathbf{r})\mathbf{x}_t \, d\mathbf{r}' + \varepsilon_t(\mathbf{r}_n) \tag{20}$$

Reduced observation equation

$$y_t(\mathbf{r}_n) = \int_\Omega m\left(\mathbf{r}_n - \mathbf{r}'\right)\phi^\top(\mathbf{r})\mathbf{x}_t \, d\mathbf{r}' + \varepsilon_t(\mathbf{r}_n) \tag{20}$$

In the more compact form

$$\mathbf{y}_t = \mathbf{C}\mathbf{x}_t + \varepsilon_t, \tag{21}$$

Reduced observation equation

$$y_t(\mathbf{r}_n) = \int_\Omega m\left(\mathbf{r}_n - \mathbf{r}'\right)\phi^\top(\mathbf{r})\mathbf{x}_t \, d\mathbf{r}' + \varepsilon_t(\mathbf{r}_n) \tag{20}$$

In the more compact form

$$\mathbf{y}_t = \mathbf{C}\mathbf{x}_t + \boldsymbol{\varepsilon}_t, \tag{21}$$

where each element of the observation matrix, **C**, is

$$\mathbf{C}_{ij} \triangleq \int_\Omega m(\mathbf{r}_i - \mathbf{r}')\phi_j(\mathbf{r}') \, d\mathbf{r}'. \tag{22}$$

Aim again, again, again

Aim: To track the patient-specific neural dynamics and estimate the parameters that influence seizures from electrophysiological measurements.

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

$$\mathbf{y}_t = \mathbf{C}\mathbf{x}_t + \varepsilon_t$$

Aim again, again, again

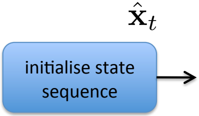
Aim: To track the patient-specific neural dynamics and estimate the parameters that influence seizures from electrophysiological measurements.

$$\mathbf{x}_{t+1} = \xi \mathbf{x}_t + \int_\Omega \mathbf{\Psi}(\mathbf{r}') f(\phi^\top(\mathbf{r}')\mathbf{x}_t)\, d\mathbf{r}' \theta + \mathbf{e}_t$$
$$\mathbf{y}_t = \mathbf{C}\mathbf{x}_t + \varepsilon_t$$

We need to estimate the states, $\mathbf{x}_t$, and parameters, θ and ξ, given the prerecorded observations, $\mathbf{y}_t$.

Estimation algorithm

We need to estimate the states, $\mathbf{x}_t$, and parameters, θ and ξ, given the prerecorded observations, $\mathbf{y}_t$. (see refs[3])

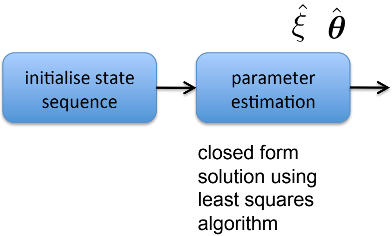


randomly
choose state
vector for all
time points

[3] Ljung, L., (1999); Haykin, S., (2001); Sarkka, S., (2010); Julier, S. and Uhlmann, J., (1997).

Estimation algorithm

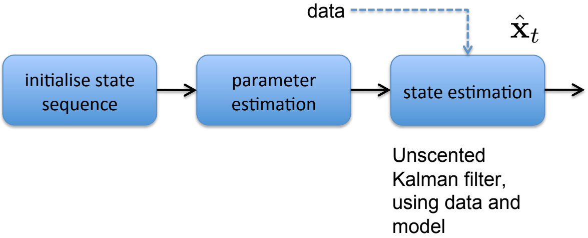
We need to estimate the states, $\mathbf{x}_t$, and parameters, $\boldsymbol{\theta}$ and ξ, given the prerecorded observations, $\mathbf{y}_t$. (see refs[3])



[3]Ljung, L., (1999); Haykin, S., (2001); Sarkka, S., (2010); Julier, S. and Uhlmann, J., (1997).

Estimation algorithm

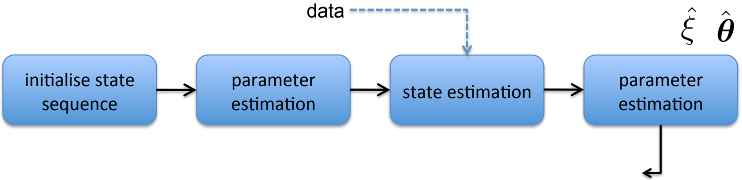
We need to estimate the states, $\mathbf{x}_t$, and parameters, θ and ξ, given the prerecorded observations, $\mathbf{y}_t$. (see refs[3])



[3] Ljung, L., (1999); Haykin, S., (2001); Sarkka, S., (2010); Julier, S. and Uhlmann, J., (1997).

Estimation algorithm

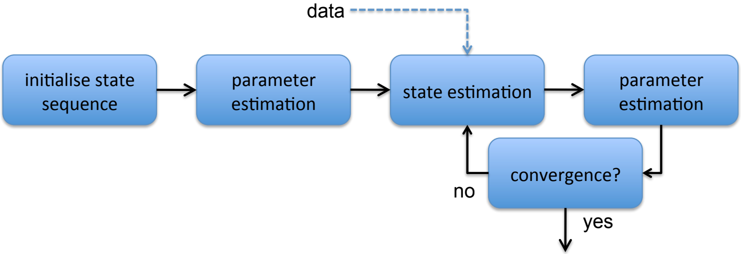
We need to estimate the states, $\mathbf{x}_t$, and parameters, θ and ξ, given the prerecorded observations, $\mathbf{y}_t$. (see refs[3])



[3] Ljung, L., (1999); Haykin, S., (2001); Sarkka, S., (2010); Julier, S. and Uhlmann, J., (1997).

Estimation algorithm

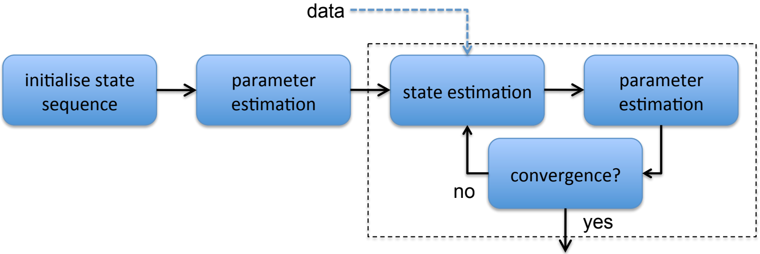
We need to estimate the states, $\mathbf{x}_t$, and parameters, θ and ξ, given the prerecorded observations, $\mathbf{y}_t$. (see refs[3])



[3] Ljung, L., (1999); Haykin, S., (2001); Sarkka, S., (2010); Julier, S. and Uhlmann, J., (1997).

Estimation algorithm

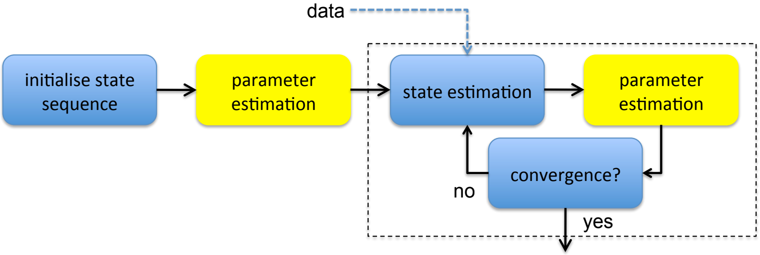
We need to estimate the states, $\mathbf{x}_t$, and parameters, θ and ξ, given the prerecorded observations, $\mathbf{y}_t$. (see refs[3])



[3] Ljung, L., (1999); Haykin, S., (2001); Sarkka, S., (2010); Julier, S. and Uhlmann, J., (1997).

Estimation algorithm

We need to estimate the states, $\mathbf{x}_t$, and parameters, θ and ξ, given the prerecorded observations, $\mathbf{y}_t$. (see refs[3])



[3] Ljung, L., (1999); Haykin, S., (2001); Sarkka, S., (2010); Julier, S. and Uhlmann, J., (1997).

Parameter estimation - least squares

Assume we have an estimate (guess) for $\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_0, \hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \ldots, \hat{\mathbf{x}}_T$

Parameter estimation - least squares

Assume we have an estimate (guess) for $\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_0, \hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \ldots, \hat{\mathbf{x}}_T$ and

$$\mathbf{x}_{t+1} = \xi \mathbf{x}_t + \underbrace{\int_\Omega \boldsymbol{\Psi}(\mathbf{r}') f(\phi^\top(\mathbf{r}')\mathbf{x}_t)\, d\mathbf{r}'}_{\mathbf{q}(\hat{\mathbf{x}}_t)} \theta + \mathbf{e}_t. \tag{23}$$

Parameter estimation - least squares

Assume we have an estimate (guess) for $\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_0, \hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \ldots, \hat{\mathbf{x}}_T$ and

$$\mathbf{x}_{t+1} = \xi \mathbf{x}_t + \underbrace{\int_\Omega \boldsymbol{\Psi}(\mathbf{r}') f(\phi^\top(\mathbf{r}')\mathbf{x}_t)\, d\mathbf{r}'}_{\mathbf{q}(\hat{\mathbf{x}}_t)} \theta + \mathbf{e}_t. \tag{23}$$

Given the estimated state sequence we can write

$$\begin{aligned}
\hat{\mathbf{x}}_1 &= \mathbf{q}(\hat{\mathbf{x}}_0)\theta + \xi\hat{\mathbf{x}}_0 + \mathbf{e}_0 \\
\hat{\mathbf{x}}_2 &= \mathbf{q}(\hat{\mathbf{x}}_1)\theta + \xi\hat{\mathbf{x}}_1 + \mathbf{e}_1 \\
&\vdots \\
\hat{\mathbf{x}}_T &= \mathbf{q}(\hat{\mathbf{x}}_{T-1})\theta + \xi\hat{\mathbf{x}}_{T-1} + \mathbf{e}_{T-1}.
\end{aligned}$$

Parameter estimation - least squares

Assume we have an estimate (guess) for $\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_0, \hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \ldots, \hat{\mathbf{x}}_T$ and

$$\mathbf{x}_{t+1} = \xi \mathbf{x}_t + \underbrace{\int_\Omega \boldsymbol{\Psi}(\mathbf{r}') f(\phi^\top(\mathbf{r}')\mathbf{x}_t)\, d\mathbf{r}'}_{\mathbf{q}(\hat{\mathbf{x}}_t)} \theta + \mathbf{e}_t. \tag{23}$$

Given the estimated state sequence we can write

$$\begin{aligned}
\hat{\mathbf{x}}_1 &= \mathbf{q}(\hat{\mathbf{x}}_0)\theta + \xi\hat{\mathbf{x}}_0 + \mathbf{e}_0 \\
\hat{\mathbf{x}}_2 &= \mathbf{q}(\hat{\mathbf{x}}_1)\theta + \xi\hat{\mathbf{x}}_1 + \mathbf{e}_1 \\
&\vdots \\
\hat{\mathbf{x}}_T &= \mathbf{q}(\hat{\mathbf{x}}_{T-1})\theta + \xi\hat{\mathbf{x}}_{T-1} + \mathbf{e}_{T-1}.
\end{aligned}$$

This can be written in the compact form

$$\mathbf{Z} = \mathbf{X}\mathcal{W} + \mathbf{e}, \tag{24}$$

Parameter estimation continued

$$\mathbf{Z} = \mathbf{X}\mathcal{W} + \mathbf{e}, \tag{25}$$

Parameter estimation continued

$$\mathbf{Z} = \mathbf{X}\mathcal{W} + \mathbf{e}, \tag{25}$$

where

$$\mathbf{Z} = \begin{bmatrix} \hat{\mathbf{x}}_1 \\ \hat{\mathbf{x}}_2 \\ \vdots \\ \hat{\mathbf{x}}_T \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} \mathbf{q}(\hat{\mathbf{x}}_0) & \hat{\mathbf{x}}_0 \\ \mathbf{q}(\hat{\mathbf{x}}_1) & \hat{\mathbf{x}}_1 \\ \vdots & \vdots \\ \mathbf{q}(\hat{\mathbf{x}}_{T-1}) & \hat{\mathbf{x}}_{T-1} \end{bmatrix}$$

and

$$\mathcal{W} = \begin{bmatrix} \boldsymbol{\theta} \\ \xi \end{bmatrix}, \quad \mathbf{e} = \begin{bmatrix} \mathbf{e}_0 \\ \mathbf{e}_1 \\ \vdots \\ \mathbf{e}_{T-1} \end{bmatrix}.$$

Parameter estimation continued

$$\mathbf{Z} = \mathbf{X}\mathcal{W} + \mathbf{e}, \tag{25}$$

where

$$\mathbf{Z} = \begin{bmatrix} \hat{\mathbf{x}}_1 \\ \hat{\mathbf{x}}_2 \\ \vdots \\ \hat{\mathbf{x}}_T \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} \mathbf{q}(\hat{\mathbf{x}}_0) & \hat{\mathbf{x}}_0 \\ \mathbf{q}(\hat{\mathbf{x}}_1) & \hat{\mathbf{x}}_1 \\ \vdots & \vdots \\ \mathbf{q}(\hat{\mathbf{x}}_{T-1}) & \hat{\mathbf{x}}_{T-1} \end{bmatrix}$$

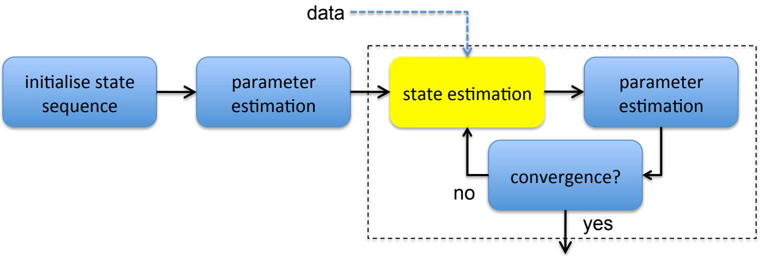
and

$$\mathcal{W} = \begin{bmatrix} \boldsymbol{\theta} \\ \xi \end{bmatrix}, \quad \mathbf{e} = \begin{bmatrix} \mathbf{e}_0 \\ \mathbf{e}_1 \\ \vdots \\ \mathbf{e}_{T-1} \end{bmatrix}.$$

The closed-form solution is

$$\hat{\mathcal{W}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Z}. \tag{26}$$

Estimation algorithm



State estimation - Kalman filter

$$\boxed{\hat{\mathbf{x}}_0}$$

State estimation - Kalman filter

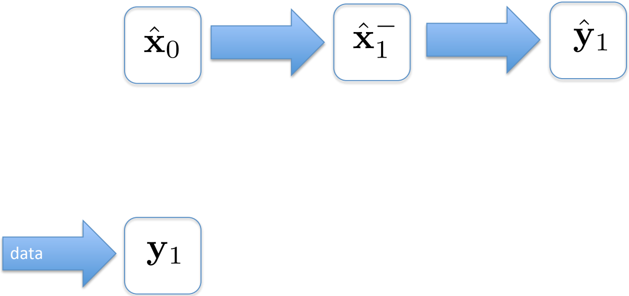
$$\hat{\mathbf{x}}_1^- = \hat{\xi}\hat{\mathbf{x}}_0 + \int_\Omega \boldsymbol{\Psi}\left(\mathbf{r}'\right) f\left(\boldsymbol{\phi}^\top(\mathbf{r}')\hat{\mathbf{x}}_0\right) d\mathbf{r}'\hat{\boldsymbol{\theta}}$$

$$\boxed{\hat{\mathbf{x}}_0} \longrightarrow \boxed{\hat{\mathbf{x}}_1^-}$$

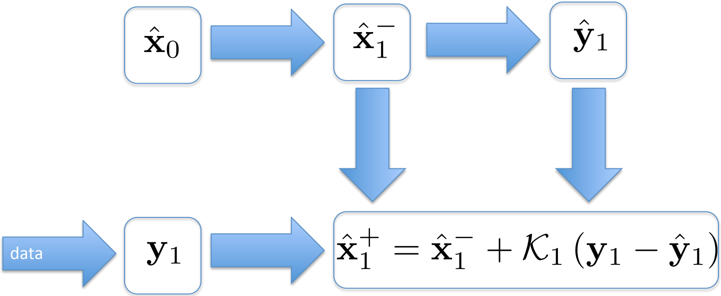
State estimation - Kalman filter

$$\hat{\mathbf{x}}_0 \longrightarrow \hat{\mathbf{x}}_1^- \xrightarrow{\hat{\mathbf{y}}_1 = \mathbf{C}\hat{\mathbf{x}}_1^-} \hat{\mathbf{y}}_1$$

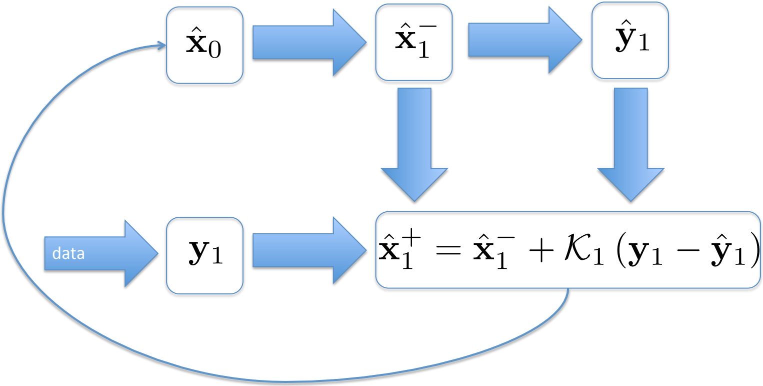
State estimation - Kalman filter



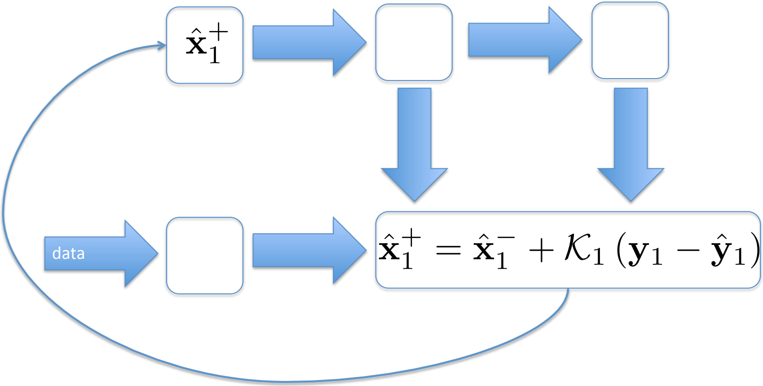
State estimation - Kalman filter



State estimation - Kalman filter

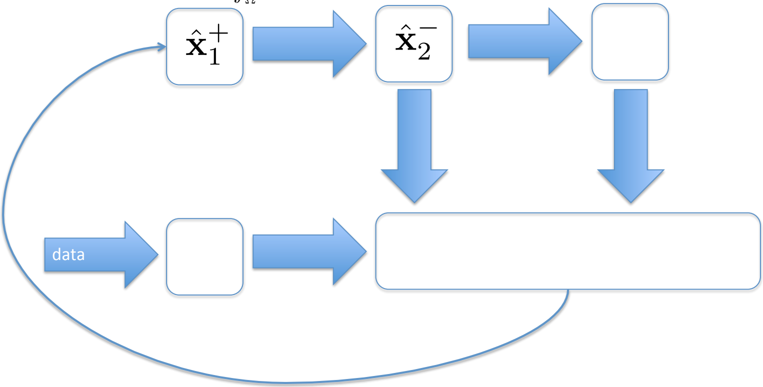


State estimation - Kalman filter

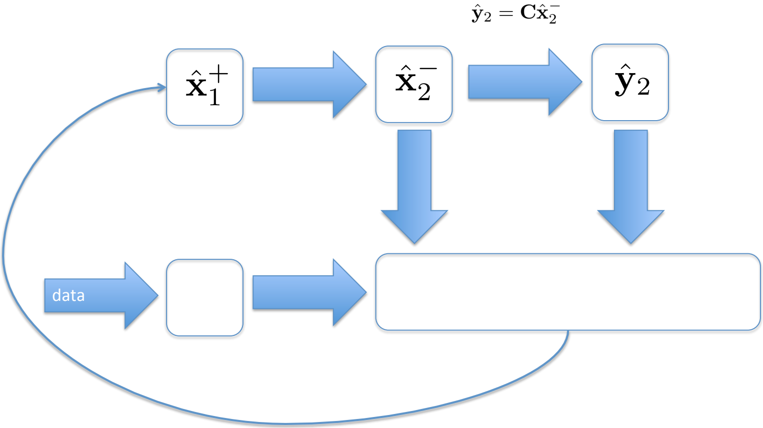


State estimation - Kalman filter

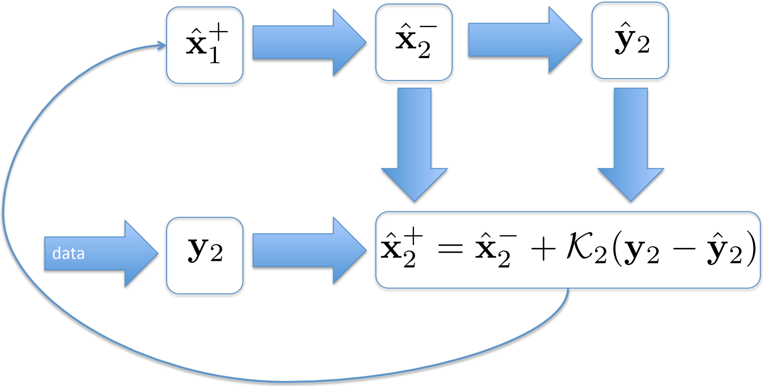
$$\hat{\mathbf{x}}_2^- = \hat{\xi}\hat{\mathbf{x}}_1^+ + \int_\Omega \mathbf{\Psi}(\mathbf{r}') f\left(\boldsymbol{\phi}^\top(\mathbf{r}')\hat{\mathbf{x}}_1^+\right) d\mathbf{r}'\hat{\boldsymbol{\theta}}$$



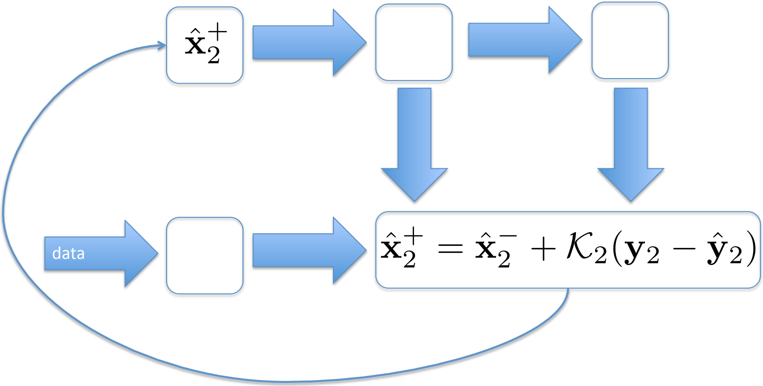
State estimation - Kalman filter



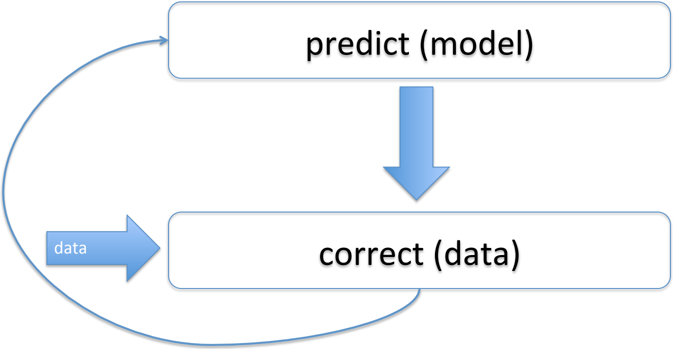
State estimation - Kalman filter



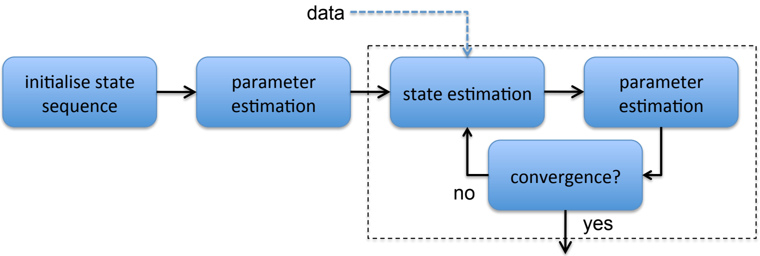
State estimation - Kalman filter



State estimation - Kalman filter



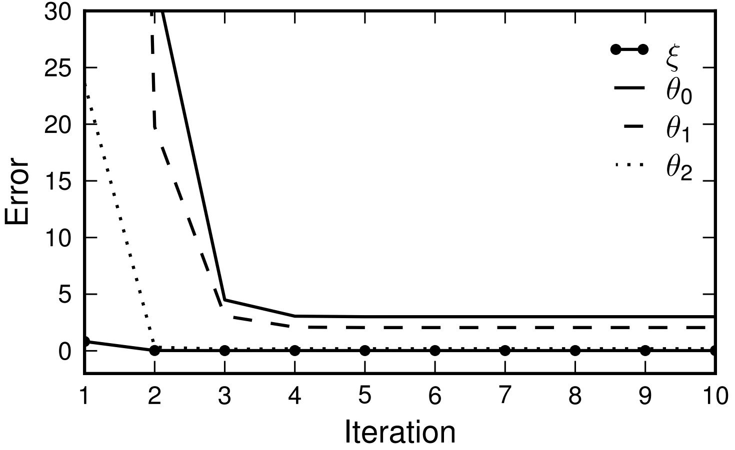
Estimation algorithm



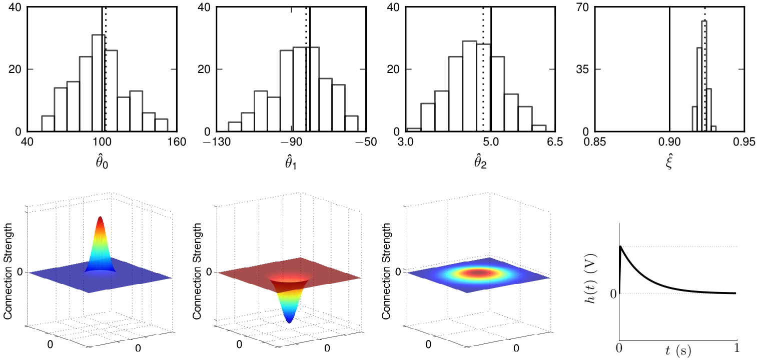
Experimental setup

- 150 realisations of data using model (Monte Carlo)
- Set the model parameters to be realistic
- Use the data-driven framework to perform state and parameters estimation
- Compare estimated parameters to actual parameters
- Compare the estimated neural field to actual field

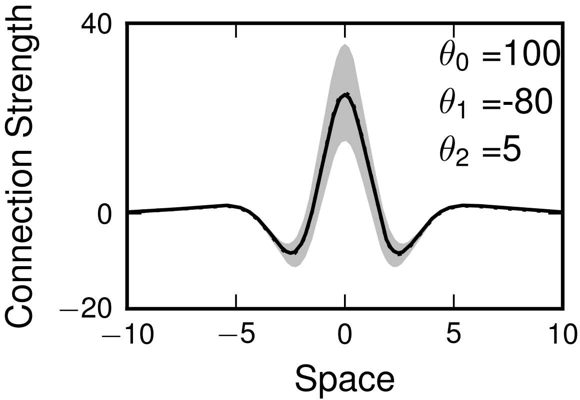
Monte Carlo simulation results - convergence



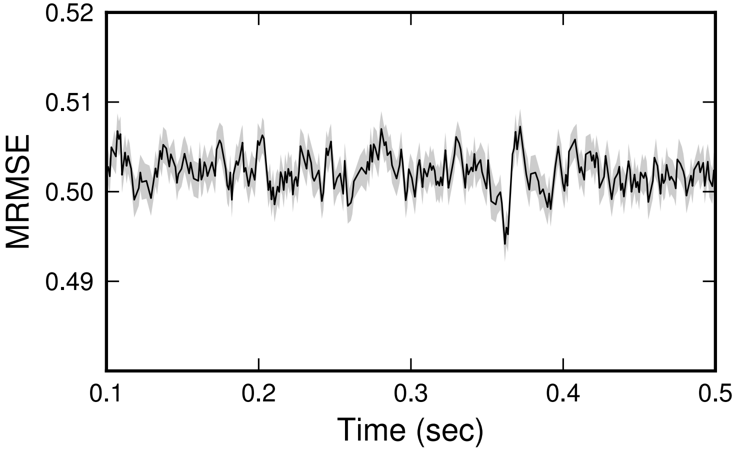
Monte Carlo simulation results - distributions



Monte Carlo simulation results - kernel estimation



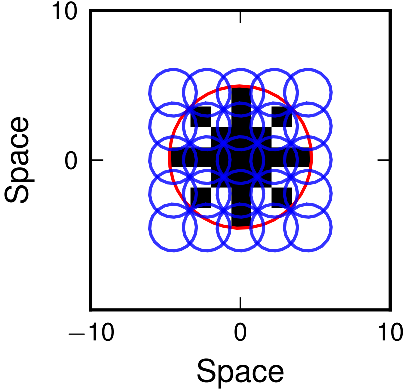
Monte Carlo simulation results - state estimation



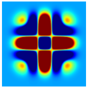
Field reconstruction

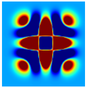
A movie was here

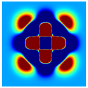
Unknown Connectivity Kernel Support (and Isotropy)

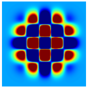


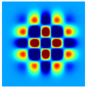
Unknown Connectivity Kernel Support (and Isotropy)

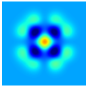

(1)

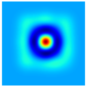

(2)

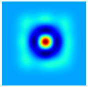

(3)

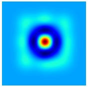

(4)

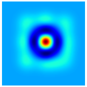

(5)

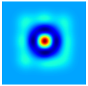

(6)

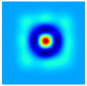

(7)

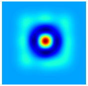

(8)

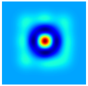

(9)

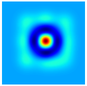

(10)


(11)

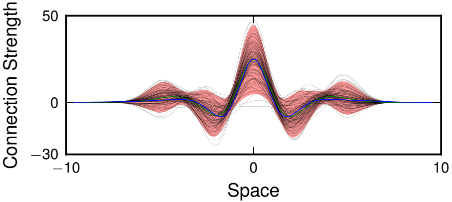

(12)


(13)

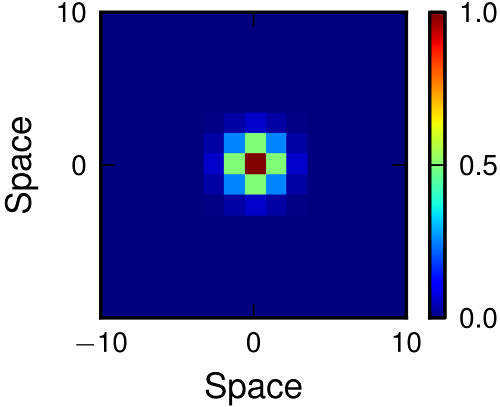

(14)


(15)

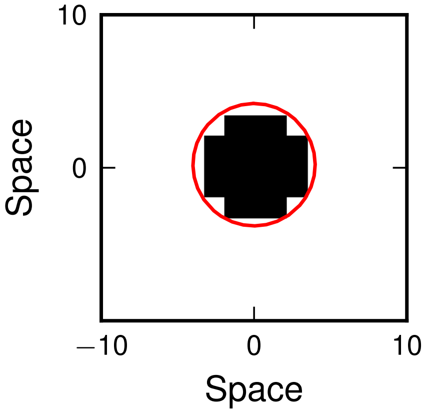
Unknown Connectivity Kernel Support (and Isotropy)



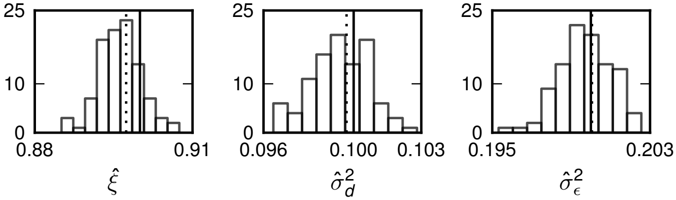
Unknown ‘Noise’ Characteristics



Unknown 'Noise' Characteristics

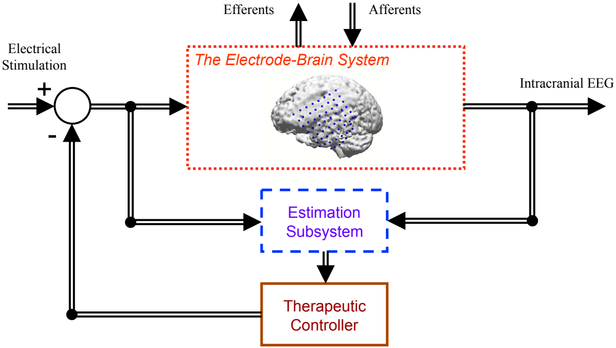


Unknown ‘Noise’ Characteristics



Summary and Discussion

- Recap
- Assumptions
- Future work



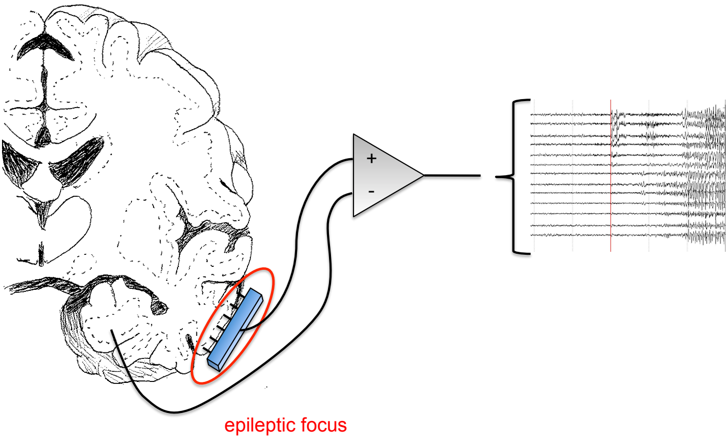
Closed-Loop Therapeutic Feedback via Electrical Stimulation

excitablecells

t

Summary and Discussion

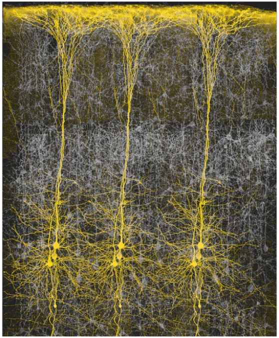
- Recap
- Assumptions
- Future work

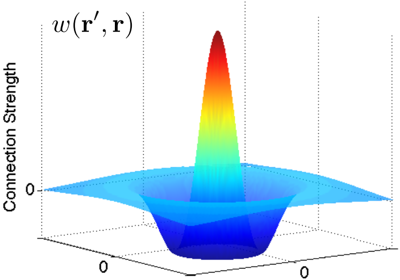


t

Summary and Discussion

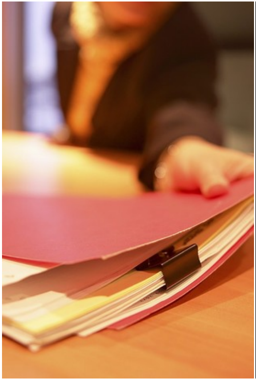
- Recap
- Assumptions
- Future work

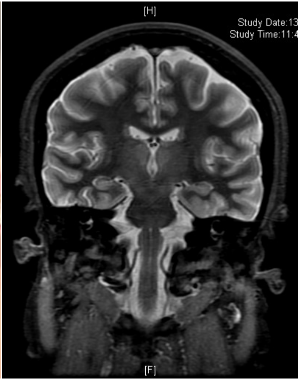


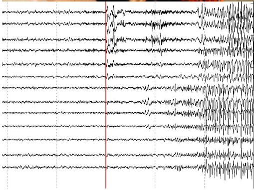


t

Patient-specific modelling - drug therapy







That's it

Thanks!