Algorithms for estimating the states and parameters of neural mass models for epilepsy

Michelle S. Chong
Department of Automatic Control, Lund University

Joint work with Romain Postoyan (CNRS, Nancy, France), Dragan Nesic (University of Melbourne-UoM), Levin Kuhlmann (UoM)
Outline

- Motivation
- Neural mass models
- Background and overview of our recent results
- Supervisory observer
- The big picture
Motivation

EEG measurement

Estimation algorithm

Seizure / No Seizure?
Motivation

EEG measurement

Model-based estimation algorithm

Seizure / No Seizure?
Motivation

EEG measurement

Model-based estimation algorithm

Seizure / No Seizure?

Which mathematical model to use?
Motivation

Which mathematical model to use?

Neural mass models
Neural mass models
Average behaviour of neuronal populations: excitatory and inhibitory interneurons

Reproduce EEG patterns related to seizures

A class of neural mass models convenient for seizure prediction

A network of neural mass models
Neural mass models

- See Deco et. al. (2008) for an overview.

Model by Stam et. al. (1999)

Model by Jansen and Rit (1995)

Model by Wendling et. al. (2005)
Neural mass model (System)

\[ \dot{x} = Ax + G(\theta^*) \gamma(Hx) + \sigma(u, y, \theta^*) \]
\[ y = Cx \]
Background and overview
Parameter and state estimation - Background

- Rich literature: adaptive control and system identification
- Stochastic and deterministic methods
Parameter and state estimation - Background

- Rich literature: adaptive control and system identification
- Stochastic and deterministic methods
- Specific to classes of systems

System:
\[
\dot{x} = f(x, u, \theta^*)
y = h(x)
\]

Goal:
\[
\hat{x}(t) \to x(t) \\
\hat{\theta}(t) \to \theta^* \\
as \ t \to \infty
\]
Parameter and state estimation - Background

- Rich literature: adaptive control and system identification
- Stochastic and deterministic methods
- Specific to classes of systems
- Our objective: a multi-model framework for estimation inspired by works in supervisory control (Vu & Liberzon; Hespanha, Morse, et al)
Overview of our estimation algorithms

- **Supervisory observer**
  - Static sampling policy
  - Dynamic sampling policy
  - DIviding RECTangles (DIRECT) sampling policy

- An adaptive observer

\( \theta^* \) is unknown & constant

\[
\dot{x} = f(x, u, \theta^*) \\
y = h(x)
\]

\[
\hat{x}(t) \rightarrow x(t) \\
\hat{\theta}(t) \rightarrow \theta^* \\
as \ t \rightarrow \infty
\]
Overview of our estimation algorithms

- **Supervisory observer**
  - Static sampling policy
  - Dynamic sampling policy
  - DIviding RECTangles (DIRECT) sampling policy

- **An adaptive observer**

  See book chapter in *Recent Advances in Predicting and Preventing Epileptic Seizures*, pp 63-82

$\theta^*$ is **unknown & constant**

\[ \text{System} \\
\dot{x} = f(x, u, \theta^*) \\
y = h(x) \]

\[ \text{Goal:} \\
\hat{x}(t) \to x(t) \\
\hat{\theta}(t) \to \theta^* \\
as t \to \infty \]
Overview of our estimation algorithms

- **Supervisory observer**
  - Static sampling policy
  - Dynamic sampling policy
  - DIviding RECTangles (DIRECT) sampling policy

- **An adaptive observer**

M. Chong, D. Nesic, R. Postoyan, L. Kuhlmann

\[ \theta^* \text{ is unknown & constant} \]

\[ \dot{x} = f(x, u, \theta^*) \]
\[ y = h(x) \]

\[ \hat{x}(t) \rightarrow x(t) \]
\[ \hat{\theta}(t) \rightarrow \theta^* \quad \text{as } t \rightarrow \infty \]
Overview of our estimation algorithms

- **Supervisory observer**
  - Static sampling policy
  - Dynamic sampling policy
  - Dlviding RECTangles (DIRECT) sampling policy

- **An adaptive observer**

θ* is unknown & constant

System:
\[
\dot{x} = f(x, u, \theta^*)
\]
\[
y = h(x)
\]

Estimation Algorithm:

Goal:
\[
\hat{x}(t) \rightarrow x(t)
\]
\[
\hat{\theta}(t) \rightarrow \theta^* \quad \text{as } t \rightarrow \infty
\]

To appear IEEE Conference on Decision and Control 2017
M. Chong, R. Postoyan, D. Nesic
Supervisory observer
Let $\theta^* \in \Theta$. 

**System**

$\dot{x} = f(x, u, \theta^*)$

$y = h(x)$
Multiple-model architecture

Let $\theta^* \in \Theta$.

System:

$\dot{x} = f(x, u, \theta^*)$

$y = h(x)$
Let $\theta^* \in \Theta$. 

System
\[ \dot{x} = f(x, u, \theta^*) \]
\[ y = h(x) \]

State observer 1
State observer 2
... 
State observer N
Let $\theta^* \in \Theta$. 

Let $\theta^* \in \Theta$. 

**System**

$$\dot{x} = f(x, u, \theta^*)$$

$$y = h(x)$$

**Some scheme to provide**

$\hat{x}$, $\hat{\theta}$

**State observer 1**

$\hat{x}_1$

**State observer 2**

$\hat{x}_2$

**State observer N**

$\hat{x}_N$
Supervisory observer

System
\[ \dot{x} = f(x, \theta^*, u) \]
\[ y = h(x, \theta^*) \]

Multiobserver
State observer
\[ \hat{x}_1 \]
State observer
\[ \hat{x}_2 \]
\[ \vdots \]
State observer
\[ \hat{x}_N \]

Supervisor
Monitoring signals
Selection criterion, \( \sigma(t) \)
Estimator
\[ \hat{x}(t) := \hat{x}_{\sigma(t)} \]
\[ \hat{\theta}(t) := \theta_{\sigma(t)} \]
Supervisory observer

Assumptions:

1. System:
   Uniformly bounded solutions.

2. Multiobserver:
   Observers are robust w.r.t parameter mismatch.

3. Supervisor:
   - Monitoring signals
     \[ \hat{\mu}_i(t) := \int_0^t e^{-\lambda(t-s)} |y(s) - \hat{y}_i(s)|^2 \, ds \]
     satisfy a PE condition, i.e.
     \[ \exists T_f > 0 \text{ and } \alpha_y \in \mathcal{K}_\infty \text{ s.t. } \forall \text{init. cond.} \]
     \[ \int_{t-T_f}^t |y(s) - \hat{y}_i(s)|^2 \, ds \geq \alpha_y (\|\tilde{\theta}_i\|), \forall t \geq T_f \]
     where \( \tilde{\theta}_i := \theta^* - \theta_i \).

4. The parameter set \( \Theta \) is sampled appropriately.
Suppose all assumptions hold. For all \( \Delta_x > 0 \) and any margins \( \nu_{\tilde{x}}, \nu_{\tilde{\nu}} > 0 \), there exist \( T > 0 \) and sufficiently large \( N^* > 0 \) such that for any \( N \geq N^* \)

\[
\left| \tilde{p}_{\sigma(t)}(t) \right| \leq \nu_{\tilde{p}}, \quad \forall t \geq T^*,
\]

\[
\limsup_{t \to \infty} \left| \tilde{x}_{\sigma(t)}(t) \right| \leq \nu_{\tilde{x}}
\]

for any initial conditions \( x(0), \hat{x}_i(0) \in B_{\Delta_i} \).

Recall \( \tilde{p}_{\sigma(t)} := \theta^* - \theta_{\sigma(t)}; \quad \tilde{x}_{\sigma(t)} = x - \hat{x}_{\sigma(t)}; \quad N = \text{no. of observers} \)
Theorem (Convergence guaranteed!)

Let $\theta^* \in \Theta$.

Suppose all assumptions hold. For all $\Delta_x > 0$ and any margins $\nu_{\tilde{x}}, \nu_{\tilde{p}} > 0$, there exist $T > 0$ and sufficiently large $N^* > 0$ such that for any $N \geq N^*$ needs large $N$!

\[
\left| \tilde{p}_{\sigma(t)}(t) \right| \leq \nu_{\tilde{p}}, \quad \forall t \geq T^*,
\]

\[
\limsup_{t \to \infty} \left| \tilde{x}_{\sigma(t)}(t) \right| \leq \nu_{\tilde{x}}
\]

for any initial conditions $x(0), \hat{x}_i(0) \in B_{\Delta_i}$.

Recall $\tilde{p}_{\sigma(t)} := \theta^* - \theta_{\sigma(t)}$; $\tilde{x}_{\sigma(t)} = x - \hat{x}_{\sigma(t)}$; $N = \text{no. of observers}$

See Chong et. al. (2015) IEEE Transactions of Automatic Control
Supervisory observer

Recall the monitoring signals
\[ \mu_i(t) := \int_0^t e^{-\lambda(t-s)} |y(s) - \hat{y}_i(s)|^2 \, ds \]
and the selection criterion
\[ \sigma(t) := \arg \min_{i \in \{1, \ldots, N\}} \mu_i(t) \]
Recall the monitoring signals
\[ \mu_i(t) := \int_0^t e^{-\lambda(t-s)} |y(s) - \hat{y}_i(s)|^2 \, ds \]

or
\[ \mu(\theta, t) := \int_0^t e^{\lambda(t-s)} |y(\theta^*, s) - \hat{y}(\theta, s)|^2 \, ds \]

and the selection criterion
\[ \sigma(t) := \arg \min_{i \in \{1, \ldots, N\}} \mu_i(t) \]

Equivalently, find \( \theta^* \in \Theta \) such that
\[ \mu(\theta^*, t) = \min_{\theta \in \Theta} \mu(\theta, t), \]
for all \( t \geq 0 \).
DIViding RECTangles (DIRECT)

- A Lipschitz optimisation method (Jones et. al. 93):
  - Find $\mu^*$ and $p^*$ such that
    \[ \mu^* = \mu(p^*) = \min_{p \in \Theta} \mu(p) \]
    $\Theta$ compact
  - $\mu$ is a Lipschitz function
  - i.e. there exists $L > 0$ such that
    \[ |\mu(a) - \mu(b)| \leq L|a - b| \]
DIViding RECTangles (DIRECT)

- A Lipschitz optimisation method (Jones et. al. 93):
  Find $\mu^*$ and $p^*$ such that
  \[ \mu^* = \mu(p^*) = \min_{p \in \Theta} \mu(p) \]
  \( \Theta \) compact
  \( \mu \) is a Lipschitz function
  i.e.
  \[ |\mu(a) - \mu(b)| \leq L |a - b| \]

- The DIRECT algorithm
DIViding RECTangles (DIRECT)

- A Lipschitz optimisation method (Jones et. al. 93):
  Find $\mu^*$ and $p^*$ such that
  $$\mu^* = \mu(p^*) = \min_{p \in \Theta} \mu(p)$$
  $\Theta$ compact
  $\mu$ is a Lipschitz function
  i.e. there exists $L > 0$ such that
  $$|\mu(a) - \mu(b)| \leq L|a - b|$$

- The DIRECT algorithm

At $k = 0$: $\hat{\mu}_0 = \min_p \mu(p)$
**Dividing RECTangles (DIRECT)**

- A Lipschitz optimisation method (Jones et. al. 93):
  Find $\mu^*$ and $p^*$ such that
  \[
  \mu^* = \mu(p^*) = \min_{p \in \Theta} \mu(p) \\
  \Theta \text{ compact} \quad \mu \text{ is a Lipschitz function}
  \]

- The DIRECT algorithm

At $k = 0$: \[\hat{\mu}_0 = \min_p \mu(p) = \mu(p_1)\]
**Dividing RECTangles (DIRECT)**

A Lipschitz optimisation method (Jones et. al. 93):
Find $\mu^*$ and $p^*$ such that
\[
\mu^* = \mu(p^*) = \min_{p \in \Theta} \mu(p)
\]
$\Theta$ compact $\mu$ is a Lipschitz function

**The DIRECT algorithm**

<table>
<thead>
<tr>
<th>$\Theta$</th>
<th>$p_3$</th>
<th>$p_5$</th>
<th>$p_1$</th>
<th>$p_2$</th>
<th>$p_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$+$</td>
<td>$+$</td>
<td>$+$</td>
<td>$+$</td>
<td>$+$</td>
</tr>
</tbody>
</table>

At $k = 0$: $\hat{\mu}_0 = \mu(p_1)$
At $k = 1$:
Dividing RECTangles (DIRECT)

- A Lipschitz optimisation method (Jones et. al. 93):
  Find $\mu^*$ and $p^*$ such that
  \[
  \mu^* = \mu(p^*) = \min_{p \in \Theta} \mu(p)
  \]
  $\Theta$ compact

  $\mu$ is a Lipschitz function

- The DIRECT algorithm
  At $k = 0$: $\hat{\mu}_0 = \mu(p_1)$
  At $k = 1$: $\mu(p_3)$, $\mu(p_5)$, $\mu(p_4)$
DIViding RECTangles (DIRECT)

- A Lipschitz optimisation method (Jones et. al. 93):
  Find $\mu^*$ and $p^*$ such that
  $$\mu^* = \mu(p^*) = \min_{p \in \Theta} \mu(p)$$
  $\Theta$ compact
  $\mu$ is a Lipschitz function

- The DIRECT algorithm
  
  At $k = 0$: $\hat{\mu}_0 = \mu(p_1)$
  At $k = 1$:
  
  E.g.
  $$\min\{\mu(p_2), \mu(p_5)\} < \min\{\mu(p_3), \mu(p_4)\}$$
DIViding RECTangles (DIRECT)

- A Lipschitz optimisation method (Jones et. al. 93):
  - Find $\mu^*$ and $p^*$ such that
    $$\mu^* = \mu(p^*) = \min_{p \in \Theta} \mu(p)$$
    $\Theta$ compact $\mu$ is a Lipschitz function

- The DIRECT algorithm

<table>
<thead>
<tr>
<th>$\Theta$</th>
<th>$p_1$</th>
<th>$p_2$</th>
<th>$p_3$</th>
<th>$p_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_4$</td>
<td>$p_*$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

At $k = 0$: $\hat{\mu}_0 = \mu(p_1)$
At $k = 1$:

E.g.
$$\min\{\mu(p_2), \mu(p_5)\} < \min\{\mu(p_3), \mu(p_4)\}$$

Divide along the horizontal dimension first.
A Lipschitz optimisation method (Jones et. al. 93):
Find $\mu^*$ and $p^*$ such that

$$\mu^* = \mu(p^*) = \min_{p \in \Theta} \mu(p)$$

$\Theta$ compact

$\mu$ is a Lipschitz function

The DIRECT algorithm

At $k = 0$: $\hat{\mu}_0 = \mu(p_1)$

At $k = 1$:

E.g.

$$\min\{\mu(p_2), \mu(p_5)\} < \min\{\mu(p_3), \mu(p_4)\}$$

Next, divide along the vertical dimension.
DIViding RECTangles (DIRECT)

A Lipschitz optimisation method (Jones et. al. 93):
Find $\mu^*$ and $p^*$ such that

$$
\mu^* = \mu(p^*) = \min_{p \in \Theta} \mu(p)
$$

$\theta$ compact $\mu$ is a Lipschitz function

The DIRECT algorithm

At $k = 0$: $\hat{\mu}_0 = \mu(p_1)$
At $k = 1$:

Now, identify the potentially optimal boxes:
There exists $L > 0$ such that
1. $\mu(p_i) - Ld_i \leq \mu(p_j) - Ld_j$
2. $\mu(p_i) - Ld_i \leq \hat{\mu}_{k-1} - \epsilon |\hat{\mu}_{k-1}|$, $\epsilon > 0$. 

\[ \begin{array}{ccc}
\theta & p_3 & p^* \\
\hline
p_5 & p_1 & p_2 \\
\hline
p_4 & & \\
\end{array} \]
A Lipschitz optimisation method (Jones et. al. 93):
Find $\mu^*$ and $p^*$ such that
$$
\mu^* = \mu(p^*) = \min_{p \in \Theta} \mu(p)
$$
$\Theta$ compact

$\mu$ is a Lipschitz function

The DIRECT algorithm

At $k = 0$: $\hat{\mu}_0 = \mu(p_1)$

At $k = 1$:

Now, identify the potentially optimal boxes:
There exists $L > 0$ such that
1. $\mu(p_i) - Ld_i \leq \mu(p_j) - Ld_j$
2. $\mu(p_i) - Ld_i \leq \hat{\mu}_{k-1} - \epsilon|\hat{\mu}_{k-1}|$, $\epsilon > 0$. 

DIViding RECTangles (DIRECT)

A Lipschitz optimisation method (Jones et. al. 93):
Find $\mu^*$ and $p^*$ such that
$$\mu^* = \mu(p^*) = \min_{p \in \Theta} \mu(p)$$
$\Theta$ compact
$\mu$ is a Lipschitz function

The DIRECT algorithm

At $k = 0$: $\hat{\mu}_0 = \mu(p_1)$
At $k = 1$:
Further divide each of the potentially optimal boxes according to the dividing procedure before.
DIviding RECTangles (DIRECT)

- A Lipschitz optimisation method (Jones et. al. 93):
  Find $\mu^*$ and $p^*$ such that
  \[ \mu^* = \mu(p^*) = \min_{p \in \Theta} \mu(p) \]
  \[ \Theta \text{ compact} \]
  $\mu$ is a Lipschitz function

- The DIRECT algorithm

  At $k = 0$: $\hat{\mu}_0 = \mu(p_1)$
  At $k = 1$:

  Further divide each of the potentially optimal boxes according to the dividing procedure before.
DIViding RECTangles (DIRECT)

- A Lipschitz optimisation method (Jones et. al. 93):
  Find $\mu^*$ and $p^*$ such that
  \[
  \mu^* = \mu(p^*) = \min_{p \in \Theta} \mu(p)
  \]
  $\Theta$ compact $\mu$ is a Lipschitz function

- The DIRECT algorithm

  At $k = 0$: $\hat{\mu}_0 = \mu(p_1)$
  At $k = 1$:

  Further divide each of the potentially optimal boxes according to the dividing procedure before.
DIViding RECTangles (DIRECT)

A Lipschitz optimisation method (Jones et. al. 93):
Find $\mu^*$ and $p^*$ such that
$$
\mu^* = \min_{p \in \Theta} \mu(p)
$$
$\Theta$ compact
$\mu$ is a Lipschitz function

The DIRECT algorithm

At $k = 0$:
$$
\hat{\mu}_0 = \mu(p_1)
$$

At $k = 1$:
Further divide each of the potentially optimal boxes according to the dividing procedure before.

Then identify $\hat{\mu}_k = \min_{p} \mu(p)$
DIViding RECTangles (DIRECT)

A Lipschitz optimisation method (Jones et. al. 93):
Find $\mu^*$ and $p^*$ such that
$$\mu^* = \mu(p^*) = \min_{p \in \Theta} \mu(p)$$
$\Theta$ compact
$\mu$ is a Lipschitz function

The DIRECT algorithm

At $k = 0$: $\hat{\mu}_0 = \mu(p_1)$
At $k = 1$:

Further divide each of the potentially optimal boxes according to the dividing procedure before.

Then identify $\hat{\mu}_k = \min_{p} \mu(p)$

Repeat for a specified number of iterations... until all the boxes are small enough
DIViding RECTangles (DIRECT)

- A Lipschitz optimisation method (Jones et. al. 93):
  Find \( \mu^* \) and \( p^* \) such that
  \[
  \mu^* = \mu(p^*) = \min_{p \in \Theta} \mu(p)
  \]
  \( \Theta \) compact \( \mu \) is a Lipschitz function

- The DIRECT algorithm

  | \( \Theta \) | + | + | + | + |
  | + | + | + | + | + |
  | + | + + + | + | + |
  | + | + | + | + | + |
  | + | + | + | + | + |

  Two questions:

  1. Let \( \Theta(k) \) be the set of sampled parameters at iteration \( k \).
     \( d(p^*, \Theta(k)) \to 0 \) as \( k \to \infty \)?
DIViding RECTangles (DIRECT)

- A Lipschitz optimisation method (Jones et. al. 93):
  Find $\mu^*$ and $p^*$ such that
  $$\mu^* = \mu(p^*) = \min_{p \in \Theta} \mu(p)$$
  $\Theta$ compact $\mu$ is a Lipschitz function

- The DIRECT algorithm

Two questions:

1. Let $\widehat{\Theta}(k)$ be the set of sampled parameters at iteration $k$.
   $d(p^*, \widehat{\Theta}(k)) \to 0$ as $k \to \infty$? YES!
   (shown in Jones et. al. 93)
**DIViding RECTangles (DIRECT)**

- A Lipschitz optimisation method (Jones et. al. 93):
  Find $\mu^*$ and $p^*$ such that
  \[
  \mu^* = \mu(p^*) = \min_{p \in \Theta} \mu(p) \quad \theta \text{ compact} 
  \]
  $\mu$ is a Lipschitz function

- The DIRECT algorithm

<table>
<thead>
<tr>
<th>$\theta$</th>
<th>$+$</th>
<th>$+$</th>
<th>$+$</th>
<th>$+$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$+$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$+$</td>
<td>$+$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$+$</td>
<td>$+$</td>
<td>$+$</td>
<td>$+$</td>
<td>$+$</td>
</tr>
<tr>
<td>$+$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Two questions:

1. Let $\hat{\Theta}(k)$ be the set of sampled parameters at iteration $k$.
   
   \[
   d(p^*, \hat{\Theta}(k)) \to 0 \text{ as } k \to \infty \quad \text{YES!}
   \]
   
   (shown in Jones et. Al. 93)

2. When do we terminate?
DIViding RECTangles (DIRECT)

- A Lipschitz optimisation method (Jones et. al. 93): Find \( \mu^* \) and \( p^* \) such that
  \[
  \mu^* = \mu(p^*) = \min_{p \in \Theta} \mu(p) \\
  \Theta \text{ compact}
  \]
  \( \mu \) is a Lipschitz function

- The DIRECT algorithm

2. When do we terminate?

**Lemma:** Given any \( d^* > 0 \),

\[
\text{Let } k^* = 3^{m-1} \left( \frac{3^{m(i+1)} - 1}{3^{m-1} - 1} \right), \text{ where } i \text{ satisfies } \\
\frac{(m3^{-2i})^{0.5}}{2} \leq d^*. \\
\text{Then, DIRECT samples } \Theta \subset \mathbb{R}^m \text{ such that}
\]

\[
\min_{p \in \Theta(k)} |p - p^*| \leq d^*, \quad \forall k \geq k^*.
\]
Combining DIRECT with the supervisory observer

System
\[ \dot{x} = f(x, \theta^*, u) \]
\[ y = h(x, \theta^*) \]

Multiobserver
- State observer
- State observer
- State observer

Supervisor
- Monitoring signals
- Selection criterion, \( \sigma(t) \)

Estimator
\[ \hat{x}(t) := \hat{x}_{\sigma(t)} \]
\[ \hat{\theta}(t) := \theta_{\sigma(t)} \]
Combining DIRECT with the supervisory observer

System
\[ \dot{x} = f(x, \theta^*, u) \]
\[ y = h(x, \theta^*) \]

At \( k=0: \)
\[ \theta \]
\[ p_1 \]

Multiobserver
\( p_1 \)
State observer
\( \hat{x}_1 \)

Supervisor
\( \hat{x}(t) := \hat{x}_{\sigma(t)} \)
\( \hat{\theta}(t) := \theta_{\sigma(t)} \)

Estimator
Combining DIRECT with the supervisory observer

At $k=1$:

\[ \theta \begin{array}{c} p_3 \\ + \\ p_5 \\ + \\ p_1 \\ + \\ p_2 \end{array} \]

At each update time $t_k$, an observer is designed for each of the newly generated samples $p_i^k \in \hat{\Theta}(k)$:

\[
\hat{x}_i(t) = f(\hat{x}_i(t), p_i^k, u, y) \\
\hat{y}_i(t) = h(\hat{x}_i(t), p_i^k), \quad \forall t \in [t_k, t_{k+1}) \\
\text{and } \hat{x}_i(t_k^+) = \hat{x}_{\sigma(t_k)}(t_k).
\]
Combining DIRECT with the supervisory observer

\[ \dot{x} = f(x, \theta^*, u) \]
\[ y = h(x, \theta^*) \]

Supervisor Multiobserver Monitoring signals State observer State observer State observer

Selection criterion, \( \sigma(t) \)

Estimator

At \( k=2 \):

\[ \theta \]
\[ + \]
\[ + \]
\[ p^* \]

At each update time \( t_k \), an observer is designed for each of the newly generated samples \( p_i^k \in \hat{\Theta}(k) \):

\[ \hat{x}_i(t) = f(\hat{x}_i(t), p_i^k, u, y) \]
\[ \hat{y}_i(t) = h(\hat{x}_i(t), p_i^k), \quad \forall t \in [t_k, t_{k+1}) \]

and \( \hat{x}_i(t_k^+) = \hat{x}_{\sigma(t_k)}(t_k) \).
Theorem: Convergence is guaranteed!

Given any \( \Delta_x, \Delta_u > 0 \), accuracy margins \( d^*, \eta > 0 \), there exist a class \( K_\infty \) function \( \gamma \), a large enough \( T > 0 \) s.t. for any sampling interval \( T_d \geq T \), there exist a class \( K_\infty \) function \( \nu_{\tilde{\rho}} \) and a constant \( T^* > 0 \) s.t.

\[
\left| \tilde{p}_{\sigma}(t)(t) \right| \leq \nu_{\tilde{\rho}}(d^*) + \eta, \quad \forall t \geq T^*
\]

\[
\limsup_{t \to \infty} \left| \tilde{x}_{\sigma}(t)(t) \right| \leq \gamma_{\tilde{x}}(d^*) + \eta
\]

for all initial conditions \( x(0) \in B_{\Delta_x} \) and inputs \( u \) bounded by \( \Delta_u \) satisfying the PE condition

\[
\exists T_f > 0 \text{ and } \alpha_y \in K_\infty \text{ s.t. } \forall \text{ init. cond.} \quad \int_{t-T_f}^t \left| y(s) - \hat{y}_i(s) \right|^2 ds \geq \alpha_y (|\tilde{p}_i|), \quad \forall t \geq T_f
\]

where \( \tilde{p}_i := \theta^* - \theta_i \).
Simulation example: Jansen and Rit ’95 neural mass model

- **States:**
  - mean membrane potential of each population

- **Parameters** \((p_1, p_2)\):
  - synaptic gain of excitatory and inhibitory populations
Simulation example: Jansen and Rit ’95 neural mass model

- **States:** mean membrane potential of each population
- **Parameters** $(p_1, p_2)$: synaptic gain of excitatory and inhibitory populations
Simulation example: Jansen and Rit ’95 neural mass model

- **States:**
  - mean membrane potential of each population

- **Parameters \((p_1, p_2)\):**
  - synaptic gain of excitatory and inhibitory populations
Simulation example: Jansen and Rit ’95 neural mass model

- **States:**
  mean membrane potential of each population

- **Parameters** \((p_1, p_2)\):
  synaptic gain of excitatory and inhibitory populations
Simulation example: Jansen and Rit ’95 neural mass model

- **States:** mean membrane potential of each population
- **Parameters** \((p_1, p_2)\): synaptic gain of excitatory and inhibitory populations
Simulation example: Jansen and Rit ’95 neural mass model

- **States:**
  mean membrane potential of each population

- **Parameters** $(p_1, p_2)$:
  synaptic gain of excitatory and inhibitory populations
Simulation example: Jansen and Rit ’95 neural mass model

- States:
  mean membrane potential of each population

- Parameters \((p_1, p_2)\):
  synaptic gain of excitatory and inhibitory populations
States:
mean membrane potential of each population

Parameters \((p_1, p_2)\):
synaptic gain of excitatory and inhibitory populations
Summary

- A framework for state and parameter estimation adapted for neural mass models.
- Convergence of estimates is guaranteed.
- Computationally tractable.
- Future work includes application to EEG data

Big picture: