Développements méthodologiques et logiciels pour l'analyse de données neurophysiologiques (bonus)

Christophe Pouzat

Mathématiques Appliquées à Paris 5 (MAP5)

Université Paris-Descartes et CNRS UMR 8145

christophe.pouzat@parisdescartes.fr

Vendredi 14 mars 2014

Où en est-on ?

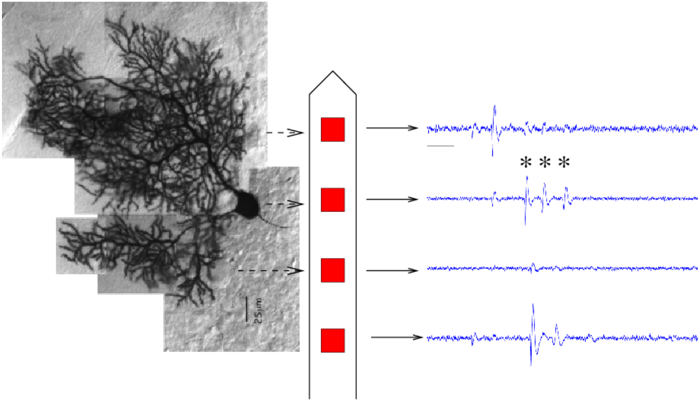
Tri des potentiels d’action dans les cas compliqués

Contexte

Ce travail a été effectué avec :

- Matthieu Delescluse, étudiant en thèse au laboratoire de physiologie cérébrale ;
- Jean Diebolt, DR CNRS, laboratoire d'analyse et de mathématiques appliquées, Université de Marne La Vallée ;
- Pascal Viot, DR CNRS (à présent Professeur), laboratoire de physique théorique de la matière condensée, Université Pierre et Marie Curie.

L'amplitude des PAs émis par un neurone varie



Un exemple d'enregistrement dans le cortex cérébelleux du rat. À gauche une cellule de Purkinje remplie avec de la biocytin (rat de 38 jours). Au milieu une sonde *neuronexus* linéaire. À droite les enregistrements sur les différents sites (Matthieu Delescluse). Les trois PAs marqués par les « * * * » viennent du même neurone.

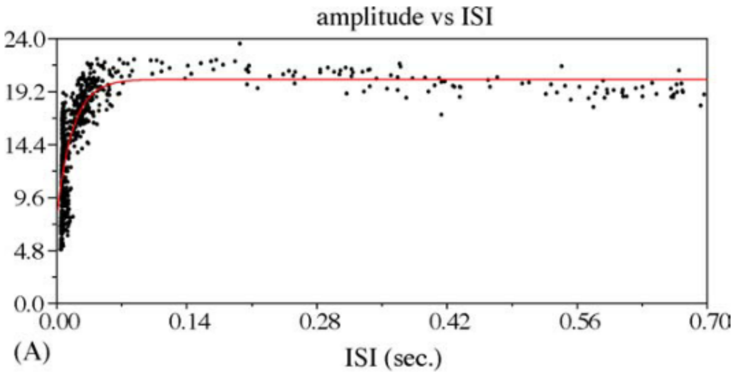
Une description de la dynamique de forme par une relaxation exponentielle

L'événement, **a**, d'un neurone donné, faisant suite au PA précédent du même neurone avec un intervalle, *isi*, devrait être décrit, en l'absence de bruit, par :

$$\mathbf{a}\,(isi) = \mathbf{p} \cdot \big(1 - \delta \cdot \exp\,(-\lambda \cdot isi)\big)\ ,$$

où λ est l'inverse d'un temps de relaxation, **p** est l'événement observé si $isi \gg \lambda^{-1}$ et $\mathbf{p} \cdot (1 - \delta)$ est l'événement observé si $isi \ll \lambda^{-1}$.

Empiriquement on voit :



En noir, les données ; en rouge, le modèle de relaxation exponentielle ajusté. Figure 3 de Delescluse et Pouzat (2006).

La statistique de décharge n'est pas Poisson

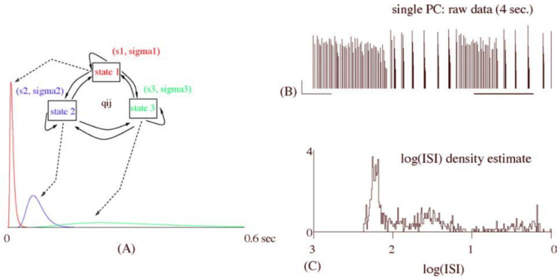


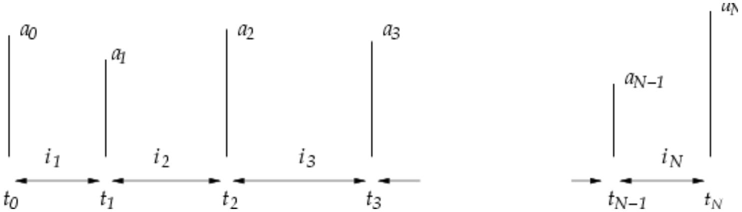
Fig. 1. Model for the ISI density compared to a real spike train. (A) ISI density model: a hidden Markov model with three states. Every state is a log-normal density with two parameters: a scale parameter s (in s) and a shape parameter (dimensionless) σ: (0.01 s, 0.5) state 1 (0.07 s, 0.3) state 2 and (0.3 s, 0.5) state 3. (B) spontaneous activity of a single PC in presence of bath-applied DHPG (40 μM) in loose cell-attached. Normalized peak amplitudes of the detected events are shown (duration: 4 s). The thick horizontal bar on the right indicates the part of the train shown in Fig. 2B. Horizontal scale bar: 0.5 s. Vertical scale bar: 5 (in units of noise S.D.). (C) $\log_{10}$ (isi) histogram of the same spike train as in (A) (1 min, 763 spikes). Bin width: 0.01.

Enregistrement d'une seule cellule de Purkinje en configuration « cellule attachée » — du DHPG (40 µM) a été rajouté à la solution pour accentuer le comportement « en bouffées » des décharges — (Delescluse et Pouzat, 2006). Nous allons modéliser la statistique de décharge par une chaîne de Markov à trois états, chaque état génère des intervalles de loi log-normale.

Modèle de génération de données

- la statistique de décharge de chaque neurone est décrite par un processus (ponctuel) de renouvellement de loi log-normale — afin d'alléger les notations, nous exposons ici le cas simplifié à « un état » — ;
- l'amplitude des PAs générés par chaque neurone dépend de l'intervalle de temps écoulé depuis le dernier PA du neurone. Cette dépendance est décrite par une relaxation exponentielle ;
- le bruit d'enregistrement est gaussien IID et est statistiquement indépendant des PAs.

Calcul de la vraisemblance pour des données d'un seul neurone (1)



Nous avons :

$$L(\mathscr{D} \mid \mathbf{p}, \delta, \lambda, s, f) = \prod_{j=1}^{N} \pi_{isi}(i_j \mid s, f) \cdot \pi_{amp}(\mathbf{a}_j \mid i_j, \mathbf{p}, \delta, \lambda) ,$$

avec :

$$\pi_{amp}(\mathbf{a}_j \mid i_j, \mathbf{p}, \delta, \lambda) = \frac{1}{(2\pi)^{\frac{n_s}{2}}} \cdot \mathrm{e}^{-\frac{1}{2} \| \mathbf{a}_j - \mathbf{p} \cdot (1 - \delta \cdot \exp(-\lambda \cdot i_j)) \|^2} .$$

Calcul de la vraisemblance pour des données d'un seul neurone (2)

La log-vraisemblance peut alors être écrite comme la somme de deux termes :

$$\mathscr{L}\left(\mathscr{D} \mid \mathbf{p}, \delta, \lambda, s, f\right) = \mathscr{L}_{isi}\left(\mathscr{D} \mid s, f\right) + \mathscr{L}_{amp}\left(\mathscr{D} \mid \mathbf{p}, \delta, \lambda\right),$$

où :

$$\mathscr{L}_{isi}\left(\mathscr{D} \mid s, f\right) = -N \cdot \ln f - \sum_{j=1}^{N} \left\{ \ln i_j + \frac{1}{2} \left[\frac{\ln\left(\frac{i_j}{s}\right)}{f} \right]^2 \right\} + \text{Cste}$$

et :

$$\mathscr{L}_{amp}\left(\mathscr{D} \mid \mathbf{p}, \delta, \lambda\right) = -\frac{1}{2} \sum_{j=1}^{N} \left\| \mathbf{a}_j - \mathbf{p} \cdot \left(1 - \delta \cdot \exp\left(-\lambda \cdot i_j\right)\right) \right\|^2 + \text{Cste}.$$

Problèmes avec des données générées par plusieurs neurones (1)

Nous allons désigner par Θ l'ensemble de paramètres de notre modèle. C'est-à-dire pour un modèle à K neurones :

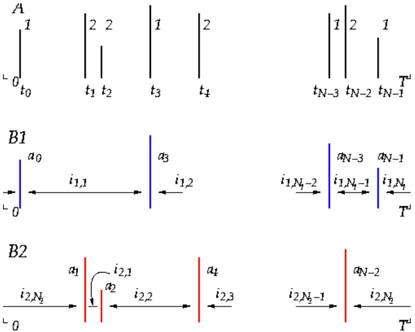
$$\Theta = (\mathbf{P}_1, \Delta_1, \Lambda_1, S_1, F_1, \ldots, \mathbf{P}_K, \Delta_K, \Lambda_K, S_K, F_K) .$$

Nous allons formaliser notre ignorance *a priori* de l'origine de chacun des PAs enregistré en « attachant » à chaque PA, j, un *label*, L_j, à valeurs dans : $\{1, \ldots, K\}$. Si $l_j = 3$, cela veut dire que l'événement j est attribué au neurone 3.
Nous allons désigner par C la configuration des données c'est-à-dire la variable aléatoire suivante :

$$C = (L_1, \ldots, L_N)^T .$$

Avec ce formalisme, le tri des PAs consiste à trouver la distribution de C.

Problèmes avec des données générées par plusieurs neurones (2)



Une fois la configuration introduite, le calcul de la vraisemblance de données générées par plusieurs neurones devient faisable :

$$L(\mathscr{D}, c \mid \theta) = \prod_{q=1}^{K} L_{l_j=q}(\mathscr{D}, c \mid \theta) \, .$$

Densité *a posteriori*

Suivant le formalisme bayesien, nous allons estimer :

$$\pi_{posterior}(\theta, c \mid \mathscr{D}) = \frac{L(\mathscr{D}, c \mid \theta) \cdot \pi_{prior}(\theta)}{Z} \, ,$$

où la constante de normalisation, Z, est donnée par :

$$Z = \sum_{c \in \mathscr{C}} \int_{\theta} d\theta \, L(\mathscr{D}, c \mid \theta) \cdot \pi_{prior}(\theta) \, .$$

Ici $\mathscr{C}$ désigne l'ensemble des configurations et c'est là qu'est le problème puisqu'il y en a K^N et qu'en pratique, $K \sim O(10)$ et $N \sim O(1000)$.

Analogie avec la Physique Statistique

On commence par poser :

$$E(\theta, c \mid \mathscr{D}) = -\log\left[L(\mathscr{D}, c \mid \theta) \cdot \pi_{prior}(\theta)\right].$$

On a alors :

$$\pi_{posterior}(\theta, c \mid \mathscr{D}) = \frac{\exp\left[-\beta E(\theta, c \mid \mathscr{D})\right]}{Z},$$

avec $\beta = 1$. Si E est considérée comme une énergie, β comme « l'inverse d'une température », Z comme une fonction de partition, alors $\pi_{posterior}$ devient une distribution canonique. Une chaîne de Markov évoluant dans l'espace $\Theta \times \mathscr{C}$ suivant une règle de type Metropolis-Hastings peut ainsi être utilisée pour générer des états : (θ, c) distribués selon $\pi_{posterior}$.

Présentation sommaire de l'algorithme

On construit notre matrice de transition :

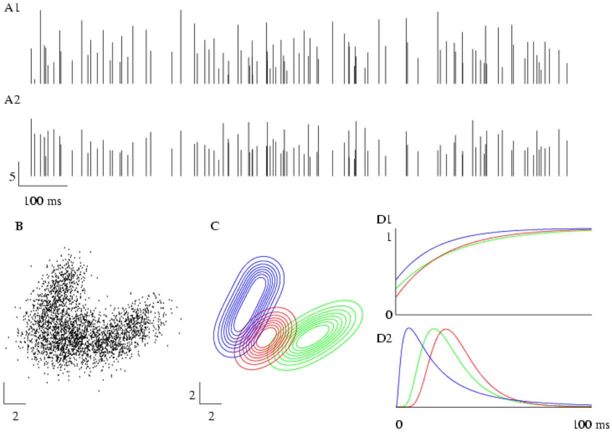
$$T\left(\theta', c' \mid \theta, c\right),$$

comme une séquence de transitions spécifiques à chaque label et à chaque paramètre du modèle :

$$T = T_{l_1} \times \ldots \times T_{l_N} \times T_{P_{1,1}} \times \ldots \times T_{f_1} \times \ldots \times T_{P_{K,1}} \times \ldots \times T_{f_K}.$$

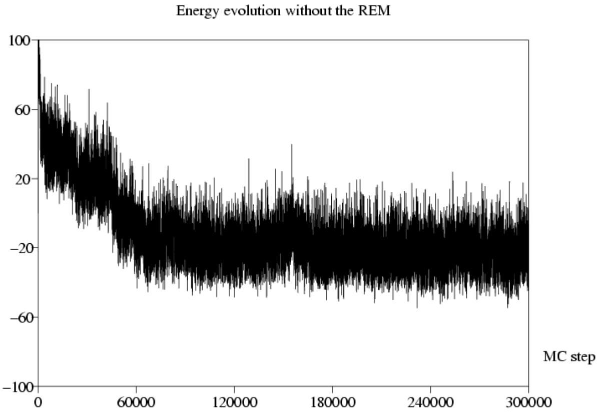
- les T_{l_j} sont des « heat-bath » (Gibbs) ;
- les T_{s_q} sont des Gaussiennes tronquées ;
- les T_{f_q} sont des inverse-Gammas tronquées ;
- pour les paramètres d'amplitude : $\mathbf{P}, \Lambda, \Delta$ on a recourt à un « Metropolis-Hastings ».

Exemple de données simulées



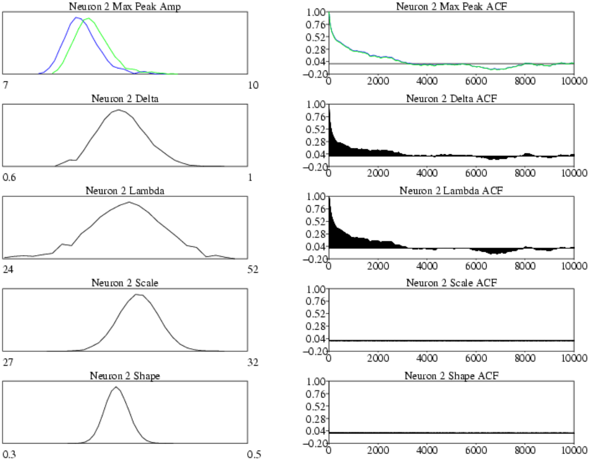
Données simulées avec 3 neurones et 2 sites d'enregistrements.

Évolution de l'énergie : relaxation lente et états méta-stables

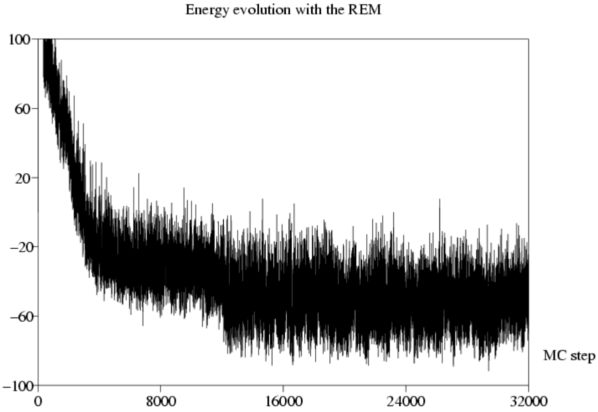


Évolution de l'énergie (opposée de la log-vraisemblance) pendant 300000 pas MC.

Distribution *a posteriori* et autocorrélations des paramètres du neurone « rouge »

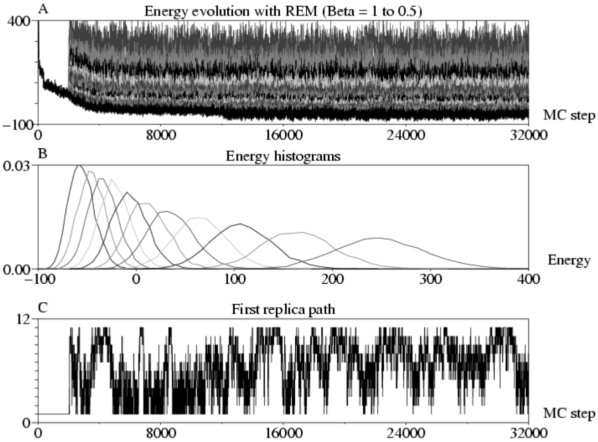


Évolution de l'énergie avec REM



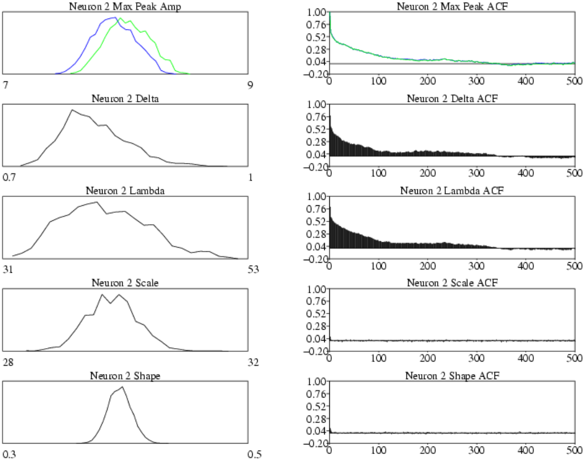
Évolution de l'énergie pendant 32000 pas MC. Le REM est mise en œuvre après le 10000e pas.

Conditions requises pour un bon fonctionnement du REM

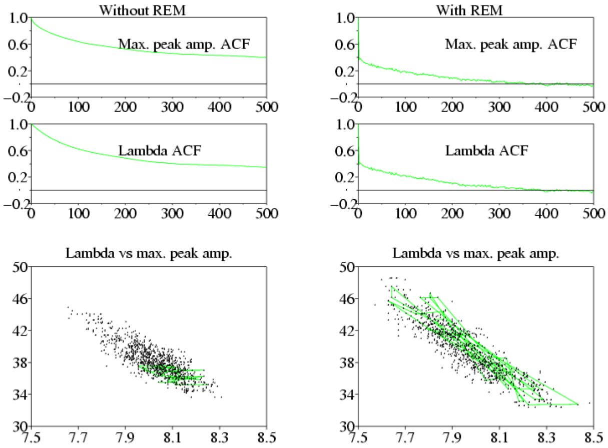


Évolution et distributions de l'énergie pendant 32000 pas MC. Le REM est mise en œuvre après le 10000^{e} pas.

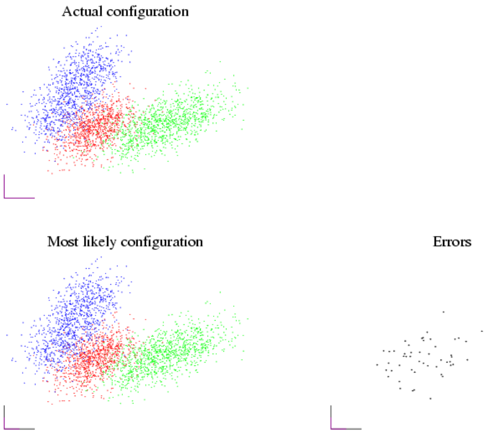
Distribution *a posteriori* et autocorrélations des paramètres du neurone « rouge » avec REM



Détails de la dynamique du REM



La classification



Retour aux données réelles (1)

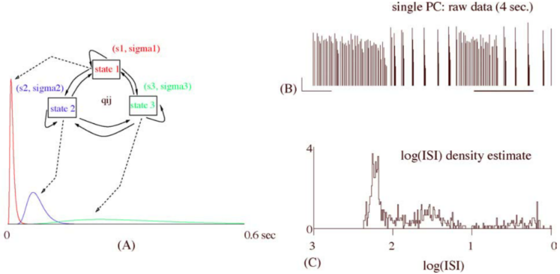


Fig. 1. Model for the ISI density compared to a real spike train. (A) ISI density model: a hidden Markov model with three states. Every state is a log-normal density with two parameters: a scale parameter s (in s) and a shape parameter (dimensionless) σ: (0.01 s, 0.5) state 1 (0.07 s, 0.3) state 2 and (0.3 s, 0.5) state 3. (B) spontaneous activity of a single PC in presence of bath-applied DHPG (40 μM) in loose cell-attached. Normalized peak amplitudes of the detected events are shown (duration: 4 s). The thick horizontal bar on the right indicates the part of the train shown in Fig. 2B. Horizontal scale bar: 0.5 s. Vertical scale bar: 5 (in units of noise S.D.). (C) $\log_{10}$ (isi) histogram of the same spike train as in (A) (1 min, 763 spikes). Bin width: 0.01.

Retour aux données réelles (2)

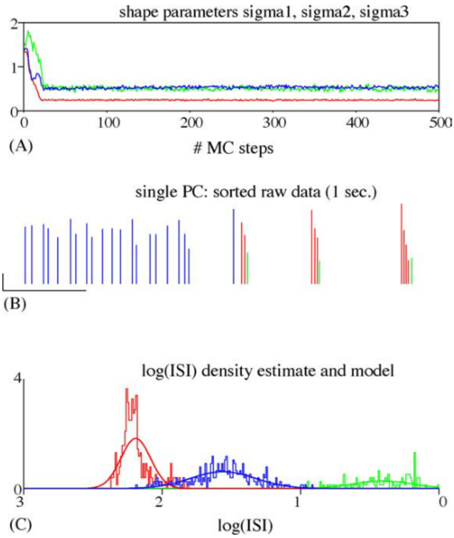


Figure 2 de Delescluse et Pouzat (2006).

Performances avec référence indépendante

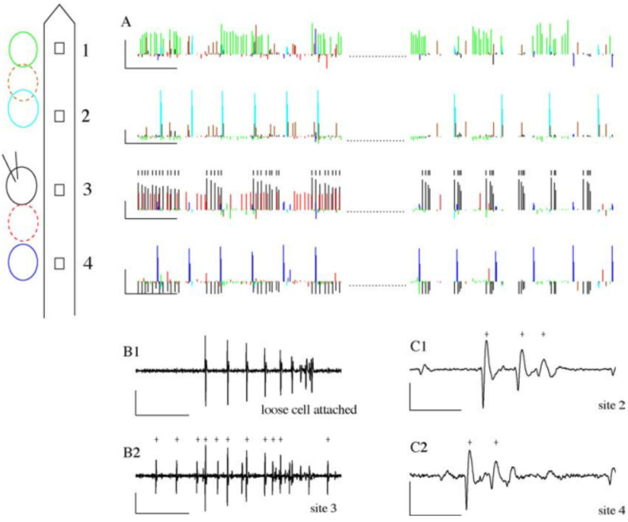


Figure 4 de Delescluse et Pouzat (2006).