

filtrage particulaire  
et  
assimilation de données séquentielle

Fabien Campillo

# filtrage non linéaire

# le modèle

- l'évolution de l'état est modélisée par

$$\mathbf{x}_k = f_k(\mathbf{x}_{k-1}) + \mathbf{w}_k \quad \mathbf{w}_k \stackrel{\text{iid}}{\sim} N(0, Q_k^w)$$

- l'observation est décrite par

$$y_k = h_k(\mathbf{x}_k) + v_k \quad v_k \stackrel{\text{iid}}{\sim} N(0, Q_k^v)$$

- $\mathbf{w}_k, v_k, \mathbf{x}_0$  sont  $\perp$

# le problème

- on veut calculer **récurivement** la loi conditionnelle  $\pi_k$  de l'état  $x_k$  sachant les observations présentes et passées  $y_{1:k} = \{y_1, \dots, y_k\}$

$$\pi_k(dx) = \mathbb{P}(x_k \in dx | y_{1:k}) \quad \text{loi}(x_k | y_{1:k})$$

$$\pi_{k-}(dx) = \mathbb{P}(x_k \in dx | y_{1:k-1}) \quad \text{loi}(x_k | y_{1:k-1})$$

- la loi conditionnelle  $\pi_k$  représente tout ce que l'on sait sur  $x_k$  sachant le modèle a priori et les observations  $y_{1:k}$

# le filtre non linéaire

méthode récursive pour calculer  $\pi_k$  à partir de  $\pi_{k-1}$  et  $y_k$ :

- **prédiction:** calculer  $\pi_{k-}$  à partir de  $\pi_{k-1}$  est du modèle d'état

$$\pi_{k-}(dx') = \pi_{k-1} Q_k(dx) \stackrel{\text{déf}}{=} \int_x \pi_{k-1}(dx) Q_k(x, dx')$$

- noyau markovien  $Q_k(x, dx') = \mathbb{P}(x_k \in dx' | x_{k-1} = x)$
- **correction:** calculer  $\pi_k$  à partir de  $\pi_{k-}$  et de  $y_k$  (formule de Bayes)

$$\underbrace{\pi_k(dx)}_{\text{posteriori}} \propto \underbrace{\psi_k(x)}_{\text{vraisemblance}} \times \underbrace{\pi_{k-}(dx)}_{\text{priori}}$$

- $\psi_k(x) = \psi_k(x, y_k)$  et  $\psi_k(x, y) dy = \mathbb{P}(y_k \in dy | x_k = x)$
- équation non linéaire (coef. de normalisation  $\int \psi_k(x) \pi_{k-}(dx)$ )

## le filtre non linéaire (suite)

ici

- $Q_k(x, dx') = \mathcal{N}(f_k(x), Q_k^w)$
- $\psi_k(x) \propto \exp(-\frac{1}{2}[y_k - h_k(x)] [Q_k^v]^{-1} [y_k - h_k(x)]^*)$

## en pratique

- le FNL est une équation dans un espace infini-dimensionnel
- à chaque itération: intégrales
- peu d'intérêt en pratique: peut se résoudre comme une EDP
  - différences finies
  - méthodes multigrilles

# filtre de Kalman et extensions

## modèle linéaire/gaussien

c'est le seul cas où (1) la solution est connue explicitement (2) elle peut être calculée à l'aide d'un algorithme

$$x_k = F_k x_{k-1} + w_k$$

$$y_k = H_k x_k + v_k$$

alors  $\pi_k(dx') = N(\hat{x}_k, P_k)$  et  $\pi_{k-}(dx') = N(\hat{x}_{k-}, P_{k-})$  avec

$$\hat{x}_{k-} = F_k \hat{x}_{k-1}$$

$$P_{k-} = F_k P_{k-1} F_k^* + Q_k^w$$

$$K_k = P_{k-} H_k^* [H_k P_{k-} H_k^* + Q_k^v]^{-1}$$

$$\hat{x}_k = \hat{x}_{k-} + K_k [y_k - H_k \hat{x}_{k-}]$$

$$P_k = [I - K_k H_k] P_{k-}$$

## filtre de Kalman étendu

linéariser les fonctions autour des estimées courantes et supposer que toutes les lois sont gaussiennes  
on obtient

$$\pi_k(dx') \simeq N(\hat{x}_k, P_k), \quad \pi_{k-}(dx') \simeq N(\hat{x}_{k-}, P_{k-})$$

avec

$$\hat{x}_{k-} = f_k(\hat{x}_{k-1})$$

$$P_{k-} = F_k P_{k-1} F_k^* + Q_k^w \quad \text{où } F_k = \partial f_k(\hat{x}_{k-1})$$

$$K_k = P_{k-} H_k^* [H_k P_{k-} H_k^* + Q_k^v]^{-1}$$

$$\hat{x}_k = \hat{x}_{k-} + K_k [y_k - h_k(\hat{x}_{k-})]$$

$$P_k = [I - K_k H_k] P_{k-} \quad \text{où } H_k = \partial h_k(\hat{x}_{k-})$$

# Unscented Kalman filter

- but: ne pas calculer de gradient.
- Idée: transformation non linéaire de formules de quadrature
  - Julier, J. Uhlmann. *A general method for approximating nonlinear transformations of probability distributions*, Technical report, University of Oxford, 1996.
  - Julier, Uhlmann, Durrant-Whyte. *A new method for the nonlinear transformation of means and covariances in filters and estimators*, *IEEE Transactions on Automatic Control*, 2000.
  - Julier, Uhlmann. *Unscented filtering and nonlinear estimation*. *IEEE Review*, 2004.

## Unscented transform

- on dispose d'une formule de quadrature  $(\omega^i, x^i)_{i=1:n}$  au second ordre pour  $\mathbf{x}$ , i.e.

$$\sum_{i=1}^n \omega^i x^i = \mathbb{E}\mathbf{x} \quad \sum_{i=1}^n \omega^i (x^i - \mathbb{E}\mathbf{x})(x^i - \mathbb{E}\mathbf{x})^* = \text{cov}(\mathbf{x})$$

then  $(\omega^i, y^i)_{i=1:n}$  with  $y^i = f(x^i)$  can be used for a quadrature formula for  $\mathbf{y} = f(\mathbf{x})$ .

- les auteurs proposent une méthode simple de calcul des  $\sigma$ -points

$$\begin{aligned} x^i &\leftarrow \mathbb{E}(\mathbf{x}) + \left[ \sqrt{(d + \lambda) \text{cov}(\mathbf{x})} \right]_i & \omega^i &\leftarrow \frac{1}{2(d + \lambda)} & i &= 1 \cdots d \\ x^{i+d} &\leftarrow \mathbb{E}(\mathbf{x}) - \left[ \sqrt{(d + \lambda) \text{cov}(\mathbf{x})} \right]_i & \omega^{i+d} &\leftarrow \frac{1}{2(d + \lambda)} & i &= 1 \cdots d \\ x^{2d+1} &\leftarrow \mathbb{E}(\mathbf{x}) & \omega^{2d+1} &\leftarrow \frac{\lambda}{d + \lambda} \end{aligned}$$

( $\lambda$  paramètre d'échelle libre)

# Unscented Kalman filter

$$\sigma_{k-1} = (\omega_{k-1}^i, x_{k-1}^i) \text{ } \sigma\text{-points pour } N(\hat{x}_{k-1}, P_{k-1})$$

$$\hat{x}_{k-} = \sum_{\sigma_{k-1}} \omega_{k-1}^i f_k(x_{k-1}^i)$$

$$P_{k-} = \sum_{\sigma_{k-1}} \omega_{k-1}^i [f_k(x_{k-1}^i) - \hat{x}_{k-}] [f_k(x_{k-1}^i) - \hat{x}_{k-}]^* + Q_k^w$$

$$\sigma_{k-} = (\omega_{k-}^i, x_{k-}^i) \text{ } \sigma\text{-points pour } N(\hat{x}_{k-}, P_{k-})$$

$$\hat{y}_{k-} = \sum_{\sigma_{k-}} \omega_{k-}^i h_k(x_{k-}^i)$$

$$S_{k-} = \sum_{\sigma_{k-}} \omega_{k-}^i [h_k(x_{k-}^i) - \hat{y}_{k-}] [h_k(x_{k-}^i) - \hat{y}_{k-}]^* + Q_k^v$$

$$U_{k-} = \sum_{\sigma_{k-}} \omega_{k-}^i [x_{k-}^i - \hat{x}_{k-}] [h_k(x_{k-}^i) - \hat{y}_{k-}]^*$$

$$\hat{x}_k = \hat{x}_{k-} + U_{k-} [S_{k-}]^{-1} [y_k - \hat{y}_{k-}]$$

$$P_k = P_{k-} - U_{k-} [S_{k-}]^{-1} [U_{k-}]^*$$

# approximation particulière

# approximation particulière

- méthode de Monte Carlo séquentielle
- biblio
  - Gordon, Salmond, Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation, IEE Proceedings, 1993.
  - Doucet, de Freitas, Gordon (eds). Sequential Monte Carlo Methods in Practice. Springer-Verlag, 2001.

# Monte Carlo

$$\mathbb{E}\phi(\mathbf{x}) \simeq \frac{1}{N} \sum_{i=1}^N \phi(\xi^i) \quad \text{où} \quad \xi^{1:N} \stackrel{\text{iid}}{\sim} \text{loi}(\mathbf{x})$$

- valable pour tout  $\phi$

$$\text{loi}(\mathbf{x}) \simeq \frac{1}{N} \sum_{i=1}^N \delta_{\xi^i}$$

- convergence assez lente (en  $\frac{1}{\sqrt{N}}$ ) – mais ne dépend pas de la dimension
- comment faire évoluer un échantillon en temps ?

# Monte Carlo séquentiel

- supposons que l'on dispose d'un  $N$ -échantillon  $\xi_{k-1}^{1:N}$  de la loi  $(x_{k-1} | y_{1:k-1})$ , i.e.

$$\pi_{k-1} \simeq \pi_{k-1}^N = \frac{1}{N} \sum_{i=1}^N \delta_{\xi_{k-1}^i}$$

- **prédiction:** on simule indépendamment

$$\xi_{k-}^i \sim Q_k(\xi_{k-1}^i, dx) = \mathcal{N}(f_k(\xi_{k-1}^i), Q_k^w)$$

i.e.  $\xi_{k-}^i = f_k(\xi_{k-1}^i) + [Q_k^w]^{1/2} \text{randn}()$ , et

$$\pi_{k-} \simeq \pi_{k-}^N = \frac{1}{N} \sum_{i=1}^N \delta_{\xi_{k-}^i}$$

# Monte Carlo séquentiel (suite)

- **correction:** on pondère les particules en fonction de leur vraisemblance

$$\omega_k^i \propto \psi_k(\xi_{k-}^i) = \exp\left(-\frac{1}{2}[y_k - h_k(\xi_{k-}^i)] [Q_k^v]^{-1} [v]^*\right)$$

(et  $\sum_{i=1}^N \omega_k^i = 1$ )

- **ré-échantillonnage:**

$$\xi_k^{1:N} \stackrel{\text{iid}}{\sim} \sum_{i=1}^N \omega_k^i \delta_{\xi_{k-}^i}$$

- éviter la dégénérescences des poids
- implémentation spécifique

# Monte Carlo séquentiel (suite)

- on doit
  - +/- savoir simuler l'équation d'état
  - calculer la vraisemblance
  - utiliser une procédure de rééchantillonnage efficace
- avantages
  - simple à développer et à mettre en œuvre (intuitif)
  - beaucoup de variations
  - réellement non linéaire
- inconvénient
  - peut-être lent (beaucoup de particules)

# filtres de Kalman d'ensemble

# biblio

- Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics, *Journal of Geophysical Research*, 1994.
- Burgers, van Leeuwen, Evensen. On the Analysis Scheme in the Ensemble Kalman Filter, *Monthly Weather Review*, 1998.
- Evensen. The ensemble Kalman filter: theoretical formulation and practical implementation, *Ocean Dynamics*, 2004.
- Pham. Stochastic methods for sequential data assimilation in strongly nonlinear systems, *Monthly Weather Review*, 2001.

# acquisition de données

- prévision d'ensemble
  - prévision météo avec 1 seule simulation: impossible
  - on génère plusieurs simulation issues de conditions initiales différentes
  - interprétation probabiliste: fréquence relative des événements
- acquisition de données
  - c'est un problème de filtrage (prévision/prédiction + analyse/correction)
  - dimension de l'espace d'état très grande

# Kalman étendu

- système  $x_k = f_k(x_{k-1}) + w_k$  et  $y_k = h_k(x_k) + v_k$
- FKE

$$\hat{x}_{k-} = f_k(\hat{x}_{k-1})$$

$$P_{k-} = F_k P_{k-1} F_k^* + Q_k^w \quad \text{où } F_k = \partial f_k(\hat{x}_{k-1})$$

$$K_k = P_{k-} H_k^* [H_k P_{k-} H_k^* + Q_k^v]^{-1}$$

$$\hat{x}_k = \hat{x}_{k-} + K_k [y_k - H_k \hat{x}_{k-}]$$

$$P_k = [I - K_k H_k] P_{k-}$$

- deux problèmes concernant la covariance
  - place mémoire
  - calcul prédiction/correction

# représentation d'ensemble

- représentation de la loi

$$\text{loi}(x_k | y_{1:k}) \longleftrightarrow \xi_k^{1:N}$$

- moyenne empirique:  $\mu(\xi_k^{1:N}) = \frac{1}{N} \sum_{i=1}^N \xi_k^i$
- covariance empirique:  $\Gamma(\xi_k^{1:N}) = \frac{1}{N-1} \sum_{i=1}^N [\xi_k^i - \mu(\xi_k^{1:N})] [\xi_k^i - \mu(\xi_k^{1:N})]^*$
- on dispose de  $N$  états : **attention  $N$  est petit**

# prévision (prédiction)

- on se donne un ensemble d'état à l'instant  $k - 1$ :  $\xi_{k-1}^{1:N}$
- dans l'étape de prévision:

$$\xi_{k-}^i = \underbrace{f_k(\xi_{k-1}^i)}_{\text{code de simu}} + w_k^i \quad i = 1 \dots N$$

- Monte Carlo

## analyse (correction)

- on s'inspire de

$$\text{“ } \hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k-} + P_{k-} H_k^* [H_k P_{k-} H_k^* + Q_k^v]^{-1} [y_k - H_k \hat{\mathbf{x}}_{k-}] \text{”}$$

- ensemble de mesures

$$y_k^i = y_k + v_k^i$$

- analyse

$$\xi_k^{1:N} = \xi_{k-}^{1:N} + \Gamma(\xi_{k-}^{1:N}) H_k^* \left[ H_k \Gamma(\xi_{k-}^{1:N}) H_k^* + \Gamma(v_k^{1:N}) \right]^{-1} \left[ y_k^{1:N} - H_k \xi_{k-}^{1:N} \right]$$

## analyse (correction) (suite)

- l'inversion de

$$M = H_k \Gamma(\xi_k^{1:N}) H_k^* + \Gamma(v_k^{1:N})$$

décomposition

$$M = Z \Lambda Z$$

( $\Lambda$  matrice diagonale) le rang de  $M$  est  $\leq N$ , donc moins de  $N$  termes sont non nuls dans  $\Lambda$

- on ne conserve en mémoire que les  $N$  premières colonnes de  $Z$

# Pham

- échantillonnage au second ordre:

$$w^{1:N} \stackrel{\text{iid}}{\sim} \mathcal{N}(0, Q) \quad \text{tels que} \quad \begin{cases} \mu(w^{1:N}) = 0 \\ \Gamma(w^{1:N}) = Q \end{cases}$$

(réduire le nombre de particules sans dégrader les performances)