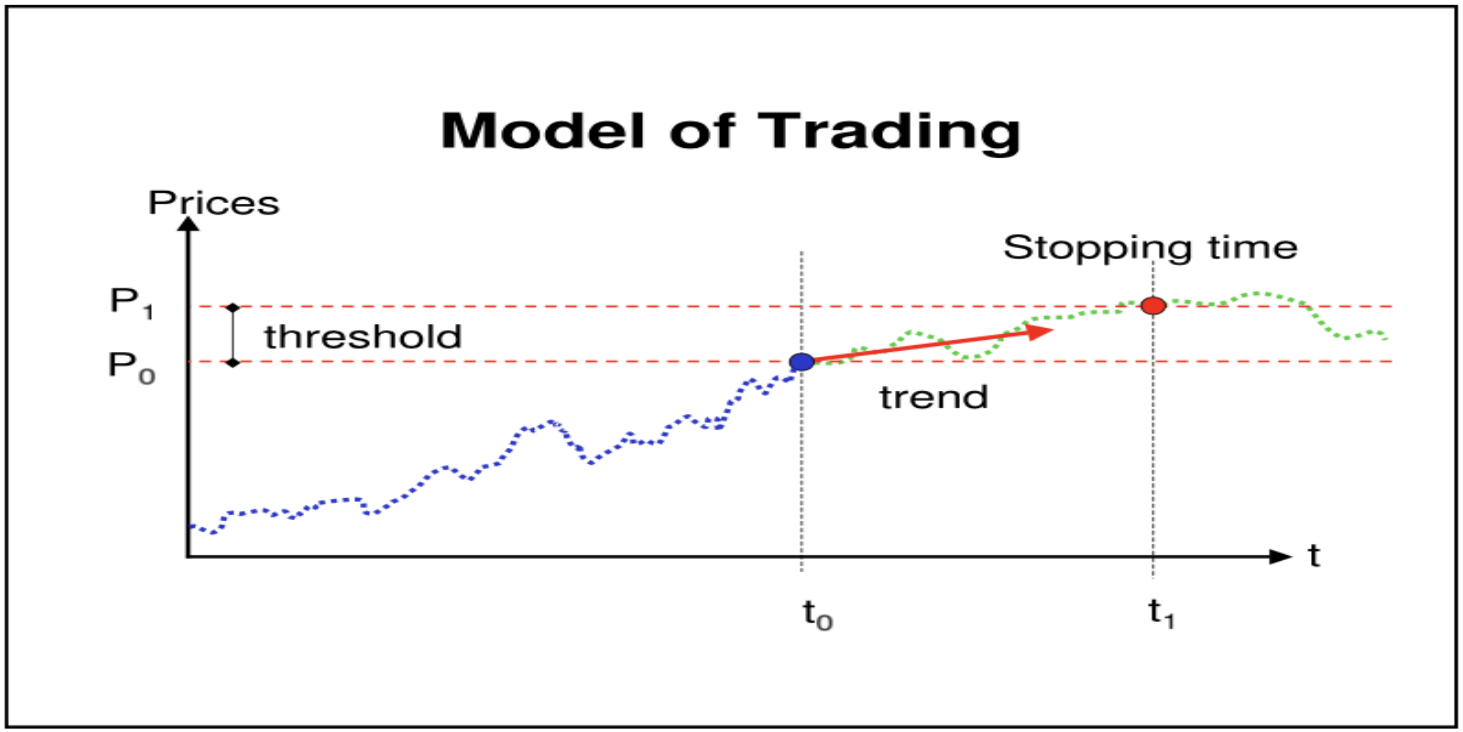
Forecasting "High" and "Low" of financial time series by Particle systems and Kalman filters

S. ANANDARAJAH, S. BOUTIGNY, F. DI FABIO, N. LATRACHE

Introduction

- But de l'article : prédiction du premier "stop-loss time" évalué à partir de "tick data" et d'un modèle de trading.



Introduction

- But de l'article : prédiction du premier "stop-loss time" évalué à partir de "tick data" et d'un modèle de trading.
- Utilisation des fonctions RBF :

$$f(x_k) = \sum_{i=1}^{I} \lambda_{ik}^f exp\left[-\frac{1}{\sigma_{ik}^f}(x_k - c_{ik}^f)^t(x_k - c_{ik}^f)\right]$$

$$h(x_k) = \sum_{j=1}^{J} \lambda_{jk}^h exp\left[-\frac{1}{\sigma_{jk}^h}(x_k - c_{jk}^h)^t(x_k - c_{jk}^h)\right]$$

- Utilisation du filtre à particules pour la prédiction

Table des matières

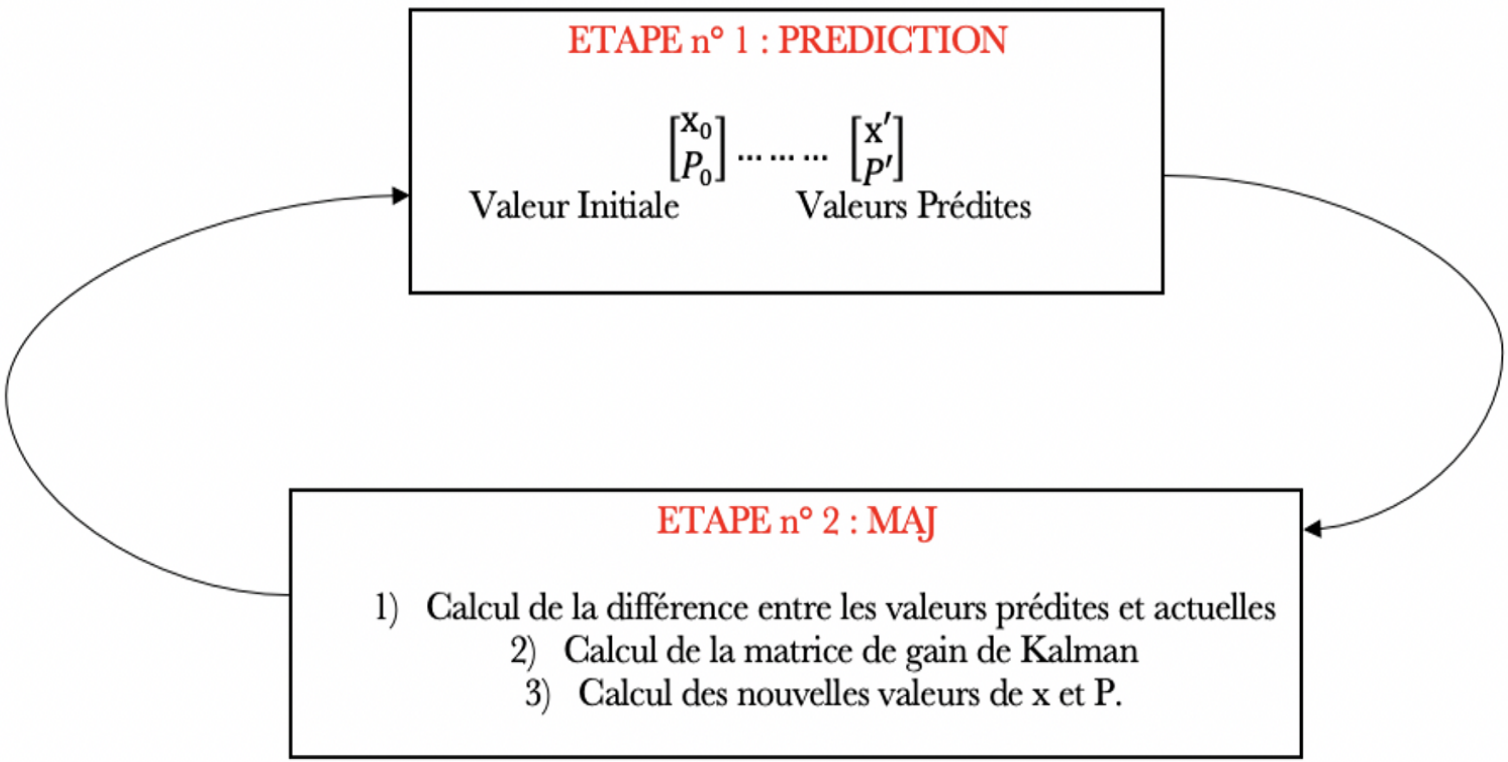
Linear Kalman Filter

Dans le cadre du LKF, le système d'équations est linéaire. Soit le système d'équations linéaires suivant :

$$\begin{cases} x_{k+1} = A_k x_k + B_k u_k + G_k v_k \\ y_k = C_k x_k + n_k \end{cases}$$

avec v_k et n_k des bruits gaussiens.

Linear Kalman Filter



Extended Kalman Filter

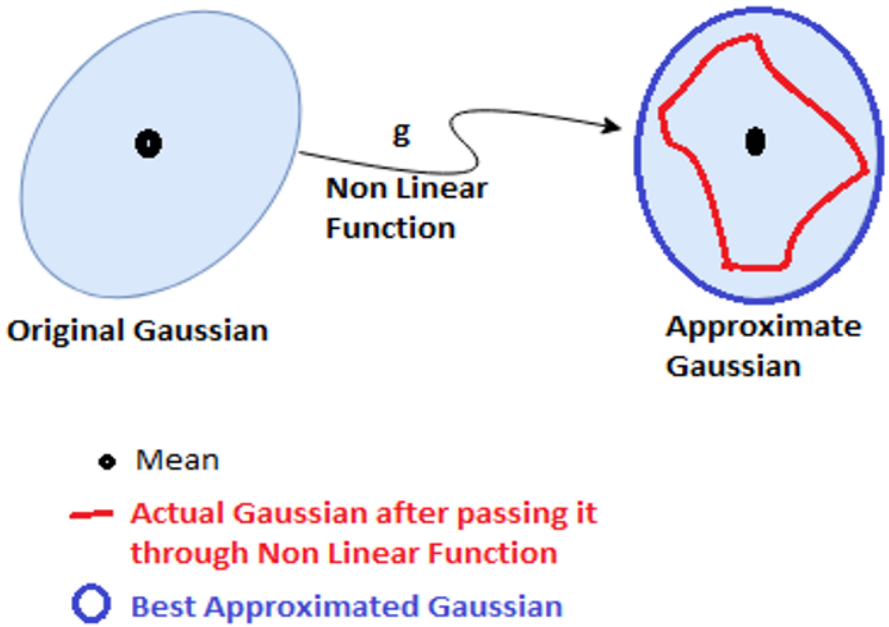
- Deux fonctions non linéaires :

$$\begin{cases} x_{k+1} = f_k(x_k, u_k, v_k; w) \\ y_k = h_k(x_k, v_k; w) \end{cases}$$

- Principe :
  - Approximation linéaire de premier ordre des fonctions de transition et de mesure à l'aide de matrices jacobiennes
  - Application du filtre de Kalman : Etapes de prévision et de mise à jour

Extended Kalman Filter : limites

- Limites :



Extended Kalman Filter : limites

- Limites :
  - Erreurs engendrées par la linéarisation
  - Divergence de l'EKF
  - Matrice jacobienne compliquée à calculer pour certains modèles

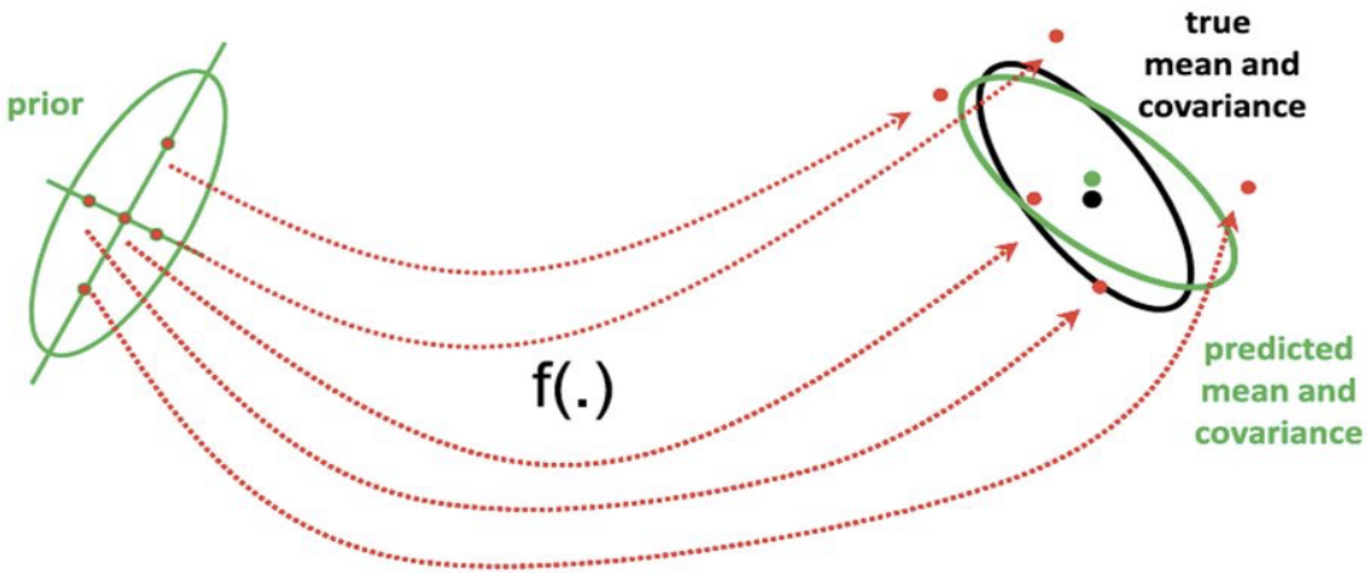
Une extension : Consistent Extended Kalman Filter

- Le but est de rendre la méthode EKF cohérente en concevant et ajoutant les paramètres suivants :
  - ΔQ_k aux calculs de $P_{k|k-1}, \Sigma_{k|k-1}, K_k, \hat{x}_{k|k}, P_{k|k}$
  - ΔR_k au calcul de $P_{k|k}$
- Les expressions de Q_k et R_k sont choisies de telle sorte que le filtre CEKF soit le plus performant
- Un filtre est cohérent si : $\mathbb{E}\left[\hat{X}_k - X_k\right] = 0$ et $\mathbb{E}\left[(\hat{X}_k - X_k)(\hat{X}_k - X_k)^t\right] = P_k$

1. erreur d'estimation suffisamment petite
2. erreur quadratique moyenne de l'estimateur égale à celle du filtre de Kalman

Unscented Kalman Filter

- Technique de filtrage qui permet de traiter des problèmes non linéaires sans utiliser la linéarisation de l'EKF.
- Utilisation de l'unscented transformation.



Calcul des Sigma-Points

$$\chi^0 = \mu$$
$$\chi^i = \mu + \gamma(\sqrt{\Sigma})_i \ \forall i = 1, \cdots, L$$
$$\chi^i = \mu - \gamma(\sqrt{\Sigma})_i \ \forall i = L+1, \cdots, 2L$$

où

- L : dimension du vecteur d'état augmenté
- λ : paramètre d'échelle (nous indique de combien on s'éloigne de la moyenne)

Calcul de la matrice racine carrée

On a deux méthodes pour calculer la matrice racine carrée :

1. Par diagonalisation : $\sqrt{\Sigma} = P\bar{D}P^{-1}$ où

$$\bar{D} = \begin{bmatrix} \sqrt{d_{11}} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sqrt{d_{nn}} \end{bmatrix}$$

En effet, soit $S = P\bar{D}P^{-1}$. Alors :

$$\begin{aligned} S \times S &= P\bar{D}P^{-1} \times P\bar{D}P^{-1} \\ &= P\bar{D}\bar{D}P^{-1} \\ &= PDP^{-1} \\ &= \Sigma \end{aligned}$$

Calcul de la matrice racine carrée

On a deux méthodes pour calculer la matrice racine carrée :

1. Par diagonalisation : $\sqrt{\Sigma} = P\bar{D}P^{-1}$ où

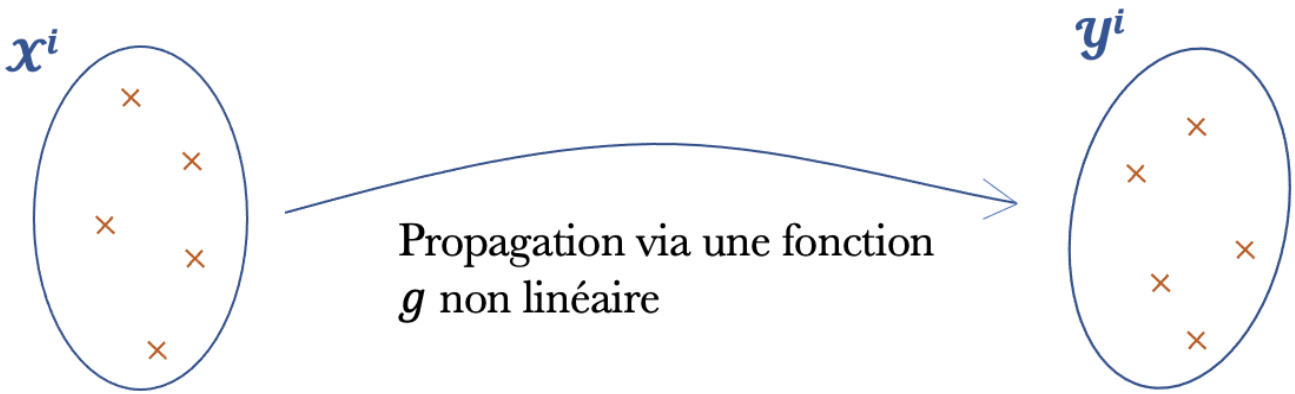
$$\bar{D} = \begin{bmatrix} \sqrt{d_{11}} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sqrt{d_{nn}} \end{bmatrix}$$

2. Par la transformation de Cholesky : $\Sigma = L \times L^t \Rightarrow \sqrt{\Sigma} = L$

Poids des Sigma-Points

- $w_m^0 = \frac{\lambda}{L+\lambda}$ Poids pour le calcul de la moyenne
- $w_c^0 = \frac{\lambda}{L+\lambda} + (1 - \alpha^2 + \beta)$ Poids pour le calcul de la matrice de covariance
- $w_m^i = \frac{\lambda}{2(L+\lambda)} \ \forall i = 1, \cdots, 2L$ Poids pour le calcul de la moyenne
- $w_c^i = \frac{\lambda}{2(L+\lambda)} \ \forall i = 1, \cdots, 2L$ Poids pour le calcul de la matrice de covariance

Propagation des Sigma-Points et calcul de la moyenne et covariance



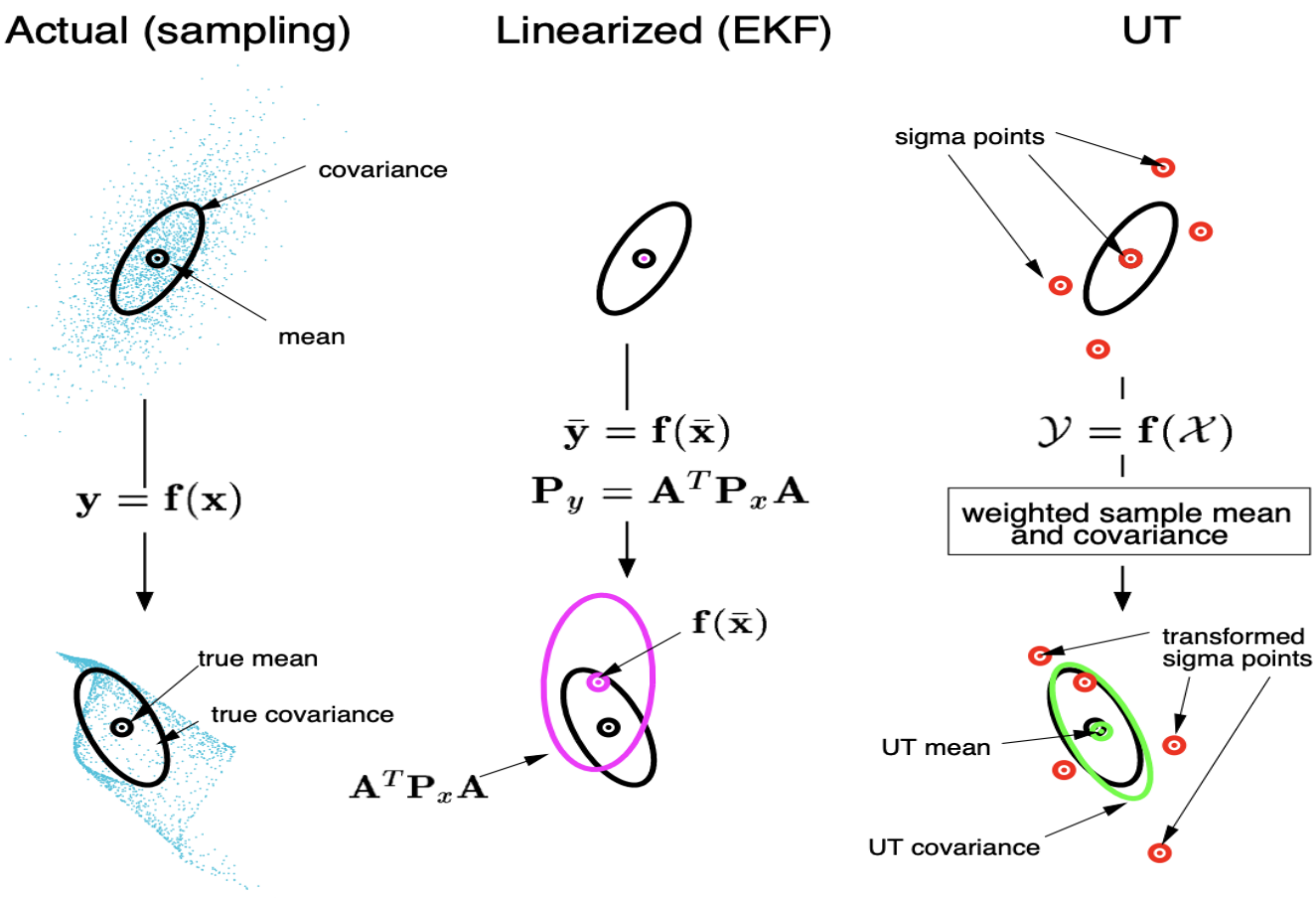
- $\tilde{\mu} = \sum_{i=0}^{2L} w_m^i f(\chi^i)$
- $\tilde{\Sigma} = \sum_{i=0}^{2L} w_c^i (f(\chi^i) - \tilde{\mu})(f(\chi^i) - \tilde{\mu})^t$

LKF vs EKF vs UKF

La mise en place de ces filtres est critiquable :

- Cadre gaussien et fonctions linéaires relativement peu probables.
- Introduction de fonctions non linéaires, plus proche de la réalité, indispensable.
- EKF le permet et fait appel à une linéarisation qui posera à son tour problème.
- La distribution est approchée avec plusieurs points.

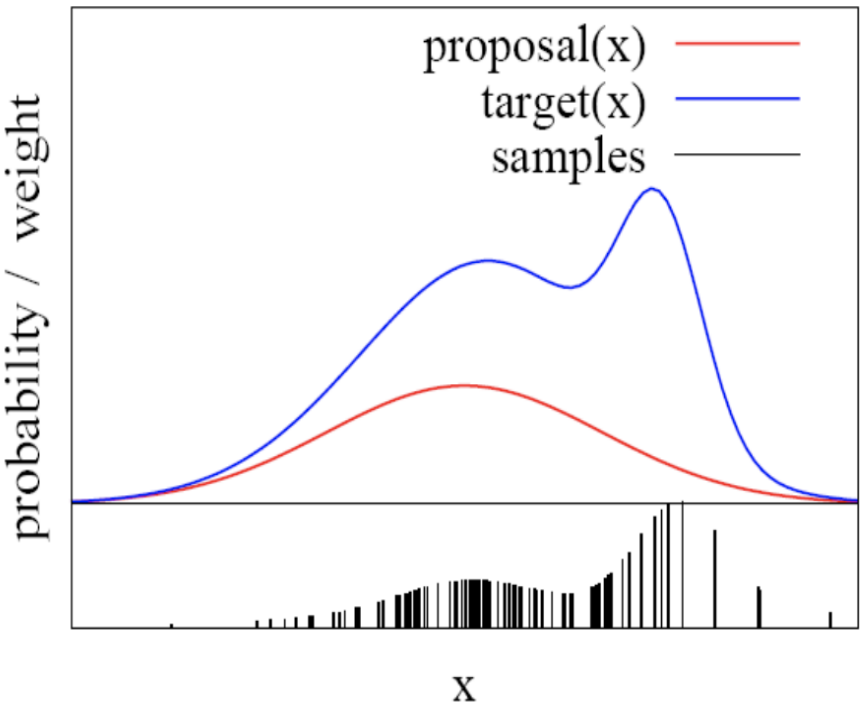
EKF vs UKF


Actual (sampling)
Linearized (EKF)
UT
covariance
mean
$\mathbf{y} = \mathbf{f}(\mathbf{x})$
true mean
true covariance
$\bar{\mathbf{y}} = \mathbf{f}(\bar{\mathbf{x}})$
$\mathbf{P}_y = \mathbf{A}^T\mathbf{P}_x\mathbf{A}$
$\mathbf{f}(\bar{\mathbf{x}})$
$\mathbf{A}^T\mathbf{P}_x\mathbf{A}$
sigma points
$\mathcal{Y} = \mathbf{f}(\mathcal{X})$
weighted sample mean and covariance
transformed sigma points
UT mean
UT covariance

Particle Filter

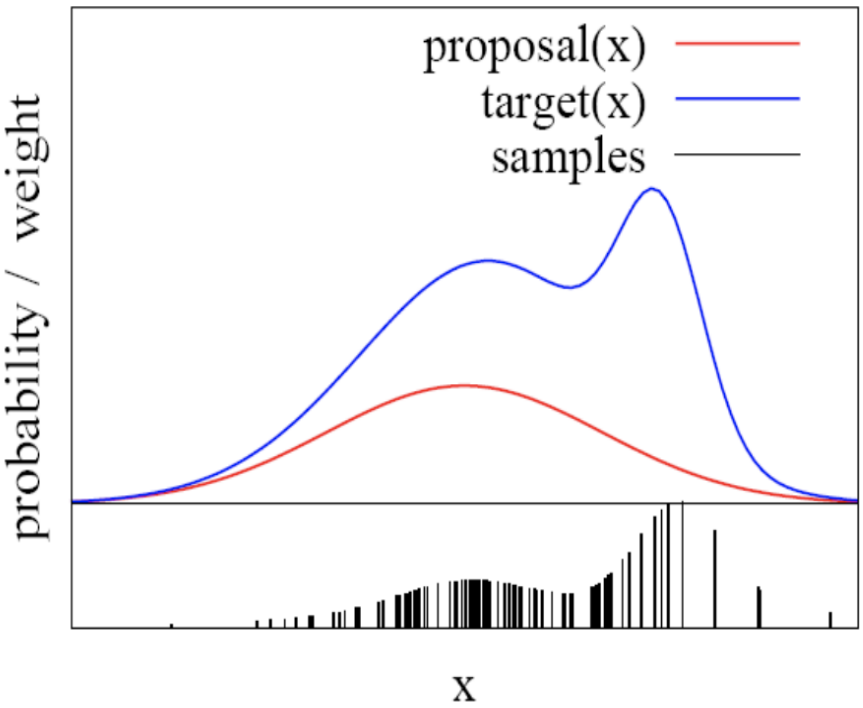
- LKF, EKF, UKF vs PF : plus aucune hypothèse sur la distribution
- On utilise deux techniques :
  1. Sequential Importance Sampling
  2. Resampling

Sequential Importance Sampling



$$\Longrightarrow w = \frac{target}{proposal}$$

Sequential Importance Sampling



$$w_k^i = w_{k-1}^i \frac{p(y_k|x_k^i)p(x_k^i|x_{k-1}^i)}{\pi(x_k^i|x_{0:k-1}^i, y_{1:k})}$$

Resampling

Après avoir simulé N points au temps t, on double le nombre de points ayant un grand poids et on élimine ceux ayant un petit poids :

Estimation en t

$$\begin{bmatrix} w_1 & x_{1,t} \\ w_2 & x_{2,t} \\ \vdots & \vdots \\ w_N & x_{N,t} \end{bmatrix}$$

Triage du vecteur

$$\begin{bmatrix} \tilde{w}_1 & \tilde{x}_{1,t} \\ \tilde{w}_2 & \tilde{x}_{2,t} \\ \vdots & \vdots \\ \tilde{w}_N & \tilde{x}_{N,t} \end{bmatrix}$$

$$\tilde{w}_i > \tilde{w}_j$$

pour $i > j$

Resampling

Elimination petits poids,

doublement grands poids

$$\begin{bmatrix} \tilde{w}_{N-j} & \tilde{x}_{N-j,t} \\ \tilde{w}_{N-j+1} & \tilde{x}_{N-j+1,t} \\ \vdots & \vdots \\ \tilde{w}_N & \tilde{x}_{N,t} \\ \tilde{w}_{j+1} & \tilde{x}_{j+1,t} \\ \vdots & \vdots \\ \tilde{w}_N & \tilde{x}_{N,t} \end{bmatrix}$$

Calcul de l'estimateur

$$\hat{x}_t = \frac{1}{N} \sum_i \tilde{x}_{i,t}$$

Sigma-Point Particle Filter

1. Etape 1 : Initialisation $\rightarrow x_i \sim$ *proposal*
2. Etape 2 : UKF $\rightarrow (\bar{x}_t^i, S_t^i)$
3. Etape 3 : Tirage $\rightarrow \tilde{x}_t^i = \bar{x}_t^i + S_t^i \times v_t$ avec $v_t \sim N(0,1)$
4. Etape 4 : Calcul des poids

$$w_t^i = w_{t-1}^i \frac{\text{vraisemblance} \times \text{transition prior}}{\text{proposal}}$$

5. Etape 5 : Resampling

Application filtres UKF et SPPF

Considérons un SSM avec les deux équations d'état et de mesure suivantes :

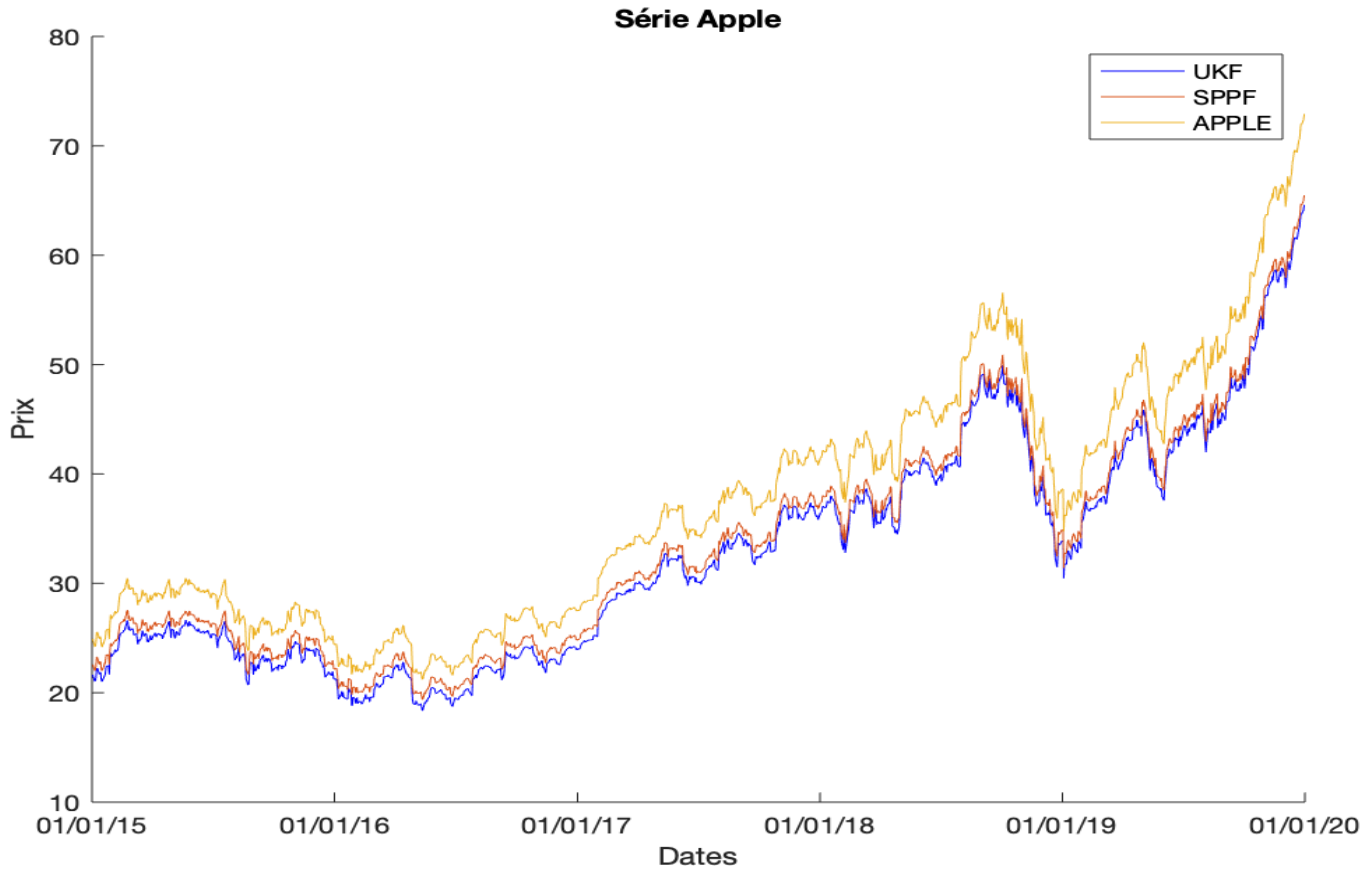
$$f(x_t) = \lambda^f \left[\frac{1}{\sigma^f} (x_t - c^f)^t (x_t - c^f) \right]$$

$$h(x_t) = \lambda^h \left[\frac{1}{\sigma^h} (x_t - c^h)^t (x_t - c^h) \right]$$

Nous avons appliqué les filtres UKF et SPPF afin de reconstruire la série Apple (02-01-2015/31-12-2019).

Application filtres UKF et SPPF

Résultats :



Application filtres EKF et UKF

Considérons un SSM avec les deux équations d'état et de mesure suivantes :

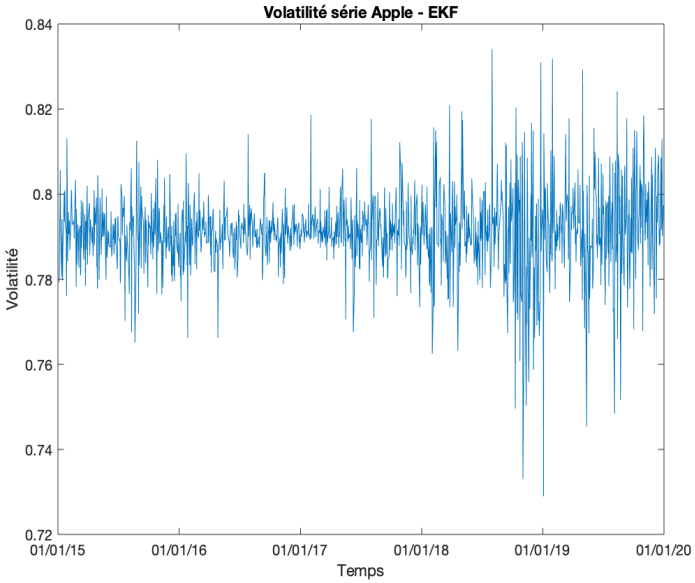
$$dVt = k(\theta - V_t)dt + \eta\rho(d\ln S_t - (\mu - \frac{1}{2}V_t)dt) + \eta\sqrt{V_t}\sqrt{1-\rho^2}dBt$$

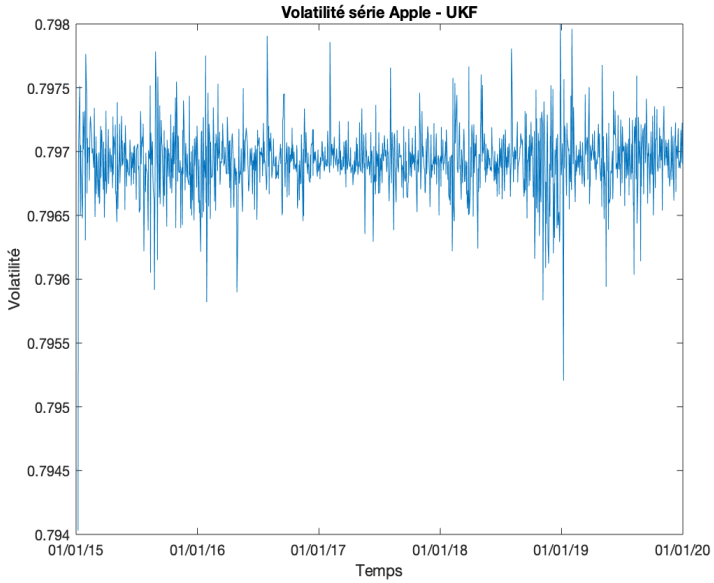
$$d\ln S_t = (\mu - \frac{1}{2}V_t)dt + \sqrt{V_t}dW_t$$

Nous avons appliqué les filtres UKF et SPPF afin d'extraire la volatilité de la série Apple (02-01-2015/31-12-2019).

Application filtres EKF et UKF et SPPF

Résultats :





Application filtres EKF et UKF et SPPF

Résultats :