

# Application du filtrage particulaire pour le suivi de véhicules

Clara CARLIER

Ryme KABAK

Elyes KARRAY

Etienne LE NAOUR

## 1 Introduction

Le filtrage particulaire est une technique d'estimation de modèles séquentiels fondée sur la simulation, l'objectif est d'estimer la densité postérieure des variables d'état compte tenu des variables d'observation. Le filtre de particules est conçu pour un modèle de Markov caché, où le système se compose de variables cachées et observables (éventuellement bruitées).

Nous nous intéressons dans ce projet à l'une des application du filtrage particulaire, à savoir le suivi (tracking) de véhicules. Nous observons des variables telles que la position, la vitesse, la direction etc. Elles sont détectées grâce à des capteurs à bord du véhicule. Ces capteurs peuvent avoir différents modes de fonctionnement et délivrent des signaux de façon synchrone ou asynchrone. Dans ce projet, nous avons implémenté deux algorithmes de filtrage particulaire permettant de tracker le véhicule en tenant compte des spécificités des capteurs.

Le premier algorithme sera un filtre particulaire standard prenant en compte des observations asynchrones et comportant une étape de resampling. Le second algorithme sera un filtre particulaire appelé Algorithme Rao-Blackwell où nous utiliserons un filtre de Kalman "unscented" (UKF). Ce second algorithme prendra également en compte des observations asynchrones et possédera une étape de resampling.

L'implémentation effectuée est basée sur les travaux de recherche tirés d'un article : <http://www.stats.ox.ac.uk/~caron/Publications/IEEETSP2006.pdf>.

## 2 Modélisation

Le véhicule que nous souhaitons suivre est doté de trois capteurs qui délivrent l'information de différentes façons : un capteur de **vitesse** et un capteur de **direction** qui sont synchrones, ainsi qu'un capteur GPS de **position** qui est asynchrone par rapport aux deux autres. On note qu'on a environ 10 fois plus d'observations provenant des capteurs de direction et de vitesse que du capteur GPS.

Le vecteur d'état que l'on cherche à estimer est :

$$x_t = (X_t \ Y_t \ V_t \ \psi_t \ \beta_t \ \dot{\beta}_t)^T \quad (1)$$

avec :

- $X_t$  et  $Y_t$  sont les positions (Nord-Est) du véhicule à l'instant  $t$ ,
- $V_t$  la vitesse,  $\psi_t$  la direction du véhicule et  $\beta_t$  l'angle de braquage à l'instant  $t$ .

### 2.1 Le modèle de transition

Le modèle de transition est un modèle stochastique (bruité) qui est censé approximer (et représenter) les déplacements du véhicule pour chaque instant  $t$ . Le modèle conseillé par l'article que nous avons utilisé

est un modèle cinématique qui s'écrit de la manière suivante (approximation au 1<sup>er</sup> ordre avec T petit) :

$$\begin{aligned}
X_{t+1} &= X_t + T \left[ V_t \cos \psi_t + (-a \sin \psi_t + b \cos \psi_t) \frac{V_t}{L} \tan \beta_t \right] \\
Y_{t+1} &= Y_t + T \left[ V_t \sin \psi_t + (a \cos \psi_t + b \sin \psi_t) \frac{V_t}{L} \tan \beta_t \right] \\
V_{t+1} &= V_t + T \dot{V}_t & \psi_{t+1} &= \psi_t + T \frac{V_t}{L} \tan(\beta_t) \\
\beta_{t+1} &= \beta_t + T \dot{\beta}_t & \dot{\beta}_{t+1} &= \dot{\beta}_t + T \ddot{\beta}_t
\end{aligned} \tag{2}$$

La composante stochastique consiste en  $(\dot{V}_t \ddot{\beta}_t)$ . Celle-ci est un bruit gaussien centré de variance estimée à partir des mesures.

## 2.2 Les modèles d'observation

En supposant un fonctionnement nominal, les observations issues du capteur GPS de position sont modélisées par :

$$z_{1,t} = h_1(x_t) + w_{1,t} = \begin{pmatrix} X_t \\ Y_t \end{pmatrix} + w_{1,t} \quad \text{où} \quad w_{1,t} \sim \mathcal{N}\left(0, \begin{pmatrix} 0.002 & 0 \\ 0 & 0.002 \end{pmatrix}\right) \tag{3}$$

Il est possible que le capteur GPS prenne un autre état de fonctionnement caractérisé par une variance plus grande.

Les observations issues du capteur de vitesse et de direction sont modélisées par :

$$z_{2,t} = h_2(x_t) + w_{2,t} = \left(1 + \tan(\beta_t) \frac{H}{L}\right) v_t + w_{2,t} \quad \text{où} \quad w_{2,t} \sim \mathcal{N}(0, 0.1) \tag{4}$$

$$z_{3,t} = h_3(x_t) + w_{3,t} = \beta_t + w_{3,t} \quad \text{où} \quad w_{3,t} \sim \mathcal{N}(0, 0.002) \tag{5}$$

Les modèles d'observation servent à mettre à jour nos poids dans nos algorithmes de filtre particulaire. Plus nos particules sont éloignées de la valeur recensée par le capteur, plus le nouveau poids associé à cette particule sera faible.

Dans notre cas, le filtrage particulaire (dans sa version simple et Rao-Blackwellisée) a pour but principal d'estimer le vecteur d'état  $x_t$  à chaque instant  $t$  étant donné le modèle de transition ainsi que les observations des positions, des vitesses et des directions.

### 3 Filtrage particulaire standard

#### 3.1 Description de l'algorithme

Décrivons le principe du premier algorithme de filtre particulaire que nous avons implémenté. On notera qu'une version détaillée et plus mathématique de cet algorithme est présentée en Annexe A.

##### Algorithme de filtrage particulaire standard

- Dans un premier temps, durant la phase d'**initialisation**, nous générons  $N$  particules de façon uniforme afin que l'espace de vie des variables d'état soit bien recouvert. On assigne à chaque particule un poids initial de  $1/N$ .
- Puis, pour chaque instant  $t$  issu du vecteur temps, défini au préalable, nous allons effectuer les étapes suivantes :
  1. On fait bouger nos  $N$  particules selon le **modèle de transition** (2).
  2. Puis on détermine si pour cet instant  $t$  l'observation est issue du capteur GPS ou bien des capteurs de vitesse et de direction.
  3. En fonction de l'information précédente, nous mettons nos **poids à jour** en tenant compte de l'erreur entre la/les variable(s) concernée(s) du vecteur d'état (1) et des données observées.
  4. Par la suite, on passe à l'étape de **resampling** où nous allons "tuer" les particules les moins probables, celles avec des poids faibles, puis dupliquer les autres particules de façon à ce que le nombre initial de particules soit conservé. À la fin du resampling, on va à nouveau assigner des poids de  $1/N$  à chaque particule.
  5. Enfin, on passe à l'étape de **prédiction** qui consiste simplement à stocker la moyenne des particules dans un vecteur.
- On recommence ce processus jusqu'à avoir parcouru tout le vecteur temps.

#### 3.2 Résultats

Premièrement, nous allons regarder les différences entre le trajet prédit et le trajet réellement recensé par le GPS pour sur tout le temps de l'expérience.

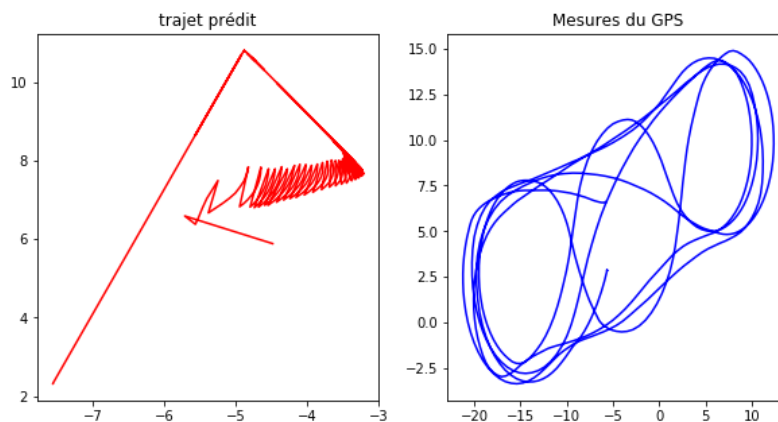


FIGURE 1 – Trajet prédit et trajet réel pour  $N = 100\,000$  particules

On se rend compte que la prédiction est mauvaise, la trajectoire semble converger vers un point au milieu de l'espace et n'évolue quasiment pas à chaque instant. Cela doit être notamment dû aux particules qui ont tendance à dégénérer au bout d'un certain temps.

Notre algorithme comportant une grande part d'aléatoire, nous devons le faire tourner plusieurs fois afin d'avoir une bonne estimation de sa performance. Nous prenons comme indicateur de performance la moyenne des différences entre une mesure GPS et la position prédite. Pour 10 exécutions consécutives de notre algorithme et  $N = 10\,000$  particules, la moyenne des erreurs est de  $1.16457 \pm 0.14596$  mètres.

Avec cet algorithme, en moyenne, l'erreur de prédiction pour un point est de 1.16457 mètres ce qui traduit des résultats non probants. Nous allons à présent tester un nouvel algorithme qui va améliorer ces résultats.

## 4 Algorithme Rao-Blackwellisé

### 4.1 Description de l'algorithme

Le changement majeur dans ce nouvel algorithme par rapport au précédent consiste à utiliser un **filtre de Kalman "unscented"** (UKF) afin d'estimer le vecteur d'état. Sa version détaillée est visible en Annexe B.

L'UKF permet d'approcher la densité postérieure du modèle d'état, sachant les observations, par une densité de loi normale qui est représentée par des points choisis de façon déterministe et optimale. Appliquer le modèle de transition à ces points va nous permettre d'estimer la moyenne et la variance de la loi normale que nous cherchons à déterminer. Pour plus de détails, se référer à l'**Annexe C** dans laquelle il est expliqué la méthode d'estimation de la moyenne et de la variance à travers le temps ainsi que comment sont choisis les poids.

### 4.2 Résultats

Premièrement, nous allons regarder la différence entre le trajet prédit et le trajet réellement recensé par le GPS.

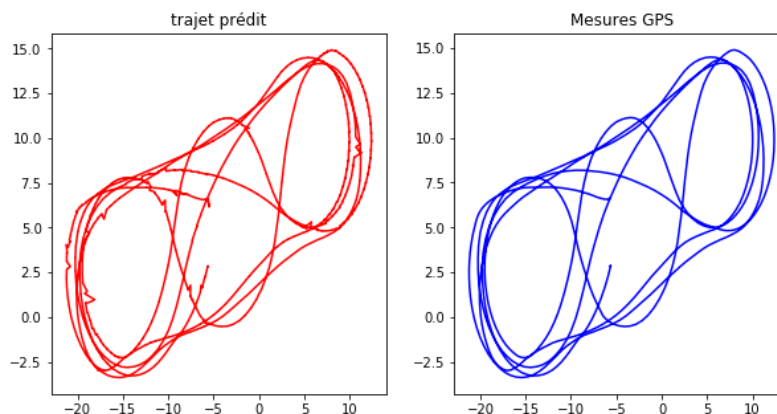


FIGURE 2 – Trajet prédit et trajet réel pour  $N = 1\,000$  particules

Cette fois-ci, la prédiction semble très bonne.

Comme précédemment, nous allons exécuter 10 fois cet algorithme afin d’avoir un résultat moyen. Lorsque nous réalisons ceci, pour  $N = 100$  particules, la moyenne est de **0.00193**  $\pm$  **1.6935e-05** mètres.

## 5 Conclusion

Ce rapport a permis de présenter rapidement notre système de filtrage particulaire visant à suivre des véhicules dotés de capteurs de vitesse, de position et de direction délivrant l’information de façon synchrone et asynchrone. Il est composé de deux modèles : un modèle de transition et un modèle d’observation. Nous avons, dans un second temps, présenté une variante basée sur les filtres de Kalman uncested et permettant d’avoir un meilleur résultat pour notre trajet prédit. L’état du GPS a été pris en compte dans le deuxième algorithme mais pas dans le premier car les particules dégénéraient trop vite. Tout cela a été inspiré d’un article de recherche à partir duquel nous avons essayé d’implémenter les deux approches et de les comparer. Néanmoins, nous n’avons malheureusement pas pu prendre réellement en compte l’état de fonctionnement du GPS. En effet, comme il a été annoncé lors des modèles d’observation, le capteur GPS possède deux états de fonctionnement : un état défectueux et un état normal. Seul ce dernier a été modélisé. Un travail futur serait de penser un nouveau modèle où en plus d’estimer le vecteur d’état  $x_t$  nous estimons l’état de fonctionnement du GPS.

## A Filtre particulaire standard avec changement du modèle d'observation

---

**Algorithm 1** Algorithme de filtrage particulaire standard avec changement du modèle d'observation

---

% == Initialisation == %

**for** particules  $i \in \{1, \dots, N\}$  **do**

Tirer :  $\hat{x}_0^{(i)} \sim p_0(\hat{x}_0)$ ,  $\sigma_0 \sim p_0(\sigma_0)$ ,  $\alpha_0^{(i)} \sim p_0(\alpha_0|\sigma_0^{(i)})$  et  $c_0^{(i)} \sim p_0(c_0|\alpha_0^{(i)})$

**end for**

% == Itérations == %

**for**  $t \in \{1, 2, \dots\}$  **do**

Attendre l'arrivée d'une mesure  $z_{k,t}$  réalisée par le capteur  $k$  :

**for** particules  $i \in \{1, \dots, N\}$  **do**

**if** arrivée de mesures  $z_{2,t}$  et  $z_{3,t}$  **then**

Mise à jour de :  $c_{1,t}^{(i)} \leftarrow c_{1,t-1}^{(i)}$ ,  $\alpha_{1,t}^{(i)} \leftarrow \alpha_{1,t-1}^{(i)}$  et  $\sigma_{1,t}^{\alpha(i)} \leftarrow \sigma_{1,t-1}^{\alpha(i)}$

Tirer la variable de l'état du véhicule :  $\tilde{\mathbf{x}}_t^{(i)} \sim q(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)}, \mathbf{z}_{1:t})$

Mise à jour des poids :  $\tilde{w}_t^{(i)} = w_{t-1}^{(i)} p(z_{2,t}|\mathbf{x}_t) p(z_{3,t}|\mathbf{x}_t)$  puis les renormaliser telle que leur somme soit égale à 1.

**end if**

**if** arrivée d'une mesure  $z_{1,t}$  du capteur GPS **then**

Tirer la variable décrivant l'état du capteur :  $\tilde{c}_{1,t}^{(i)} \sim q(c_{1,t}|\mathbf{x}_{t-1}^{(i)}, \alpha_{1,t-1}^{(i)}, \mathbf{z}_{1,t})$

Tirer la variable de l'état du véhicule :  $\tilde{x}_t^{(i)} \sim q(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)}, \tilde{c}_{1,t}^{(i)}, \mathbf{z}_{1,t})$

Tirer les probabilités :  $\tilde{\alpha}_{1,t}^{(i)} \sim q(\alpha_{1,t}|\alpha_{1,t-1}^{(i)}, \tilde{c}_{1,t}^{(i)}, \sigma_{1,t-1}^{(i)})$

Tirer l'hyperparamètre :  $\tilde{\sigma}_{1,t}^{\alpha(i)} \sim q(\sigma_{1,t}^{\alpha(i)}|\sigma_{1,t-1}^{\alpha(i)}, \tilde{\alpha}_{1,t}^{(i)}, \alpha_{1,t-1}^{(i)})$

Mise à jour des poids : ( $h$  définie ci-dessous, équation (6))

$$\tilde{w}_t^{(i)} \propto h(w_{t-1}^{(i)}, z_{k,t}, \tilde{c}_{k,t}^{(i)}, \tilde{\alpha}_{k,t}^{(i)}, \alpha_{k,t-1}^{(i)}, \tilde{\alpha}_t^{(i)}, \alpha_{k,t-1}^{(i)}, \tilde{x}_t^{(i)}, x_{t-1}^{(i)}, \tilde{\sigma}_{k,t}^{(i)}, \sigma_{k,t-1}^{(i)}, \tilde{\sigma}_{k,t}^{\alpha(i)}|\sigma_{k,t-1}^{\alpha(i)})$$

puis les renormaliser telle que leur somme soit égale à 1.

**end if**

**end for**

**end for**

% == Rééchantillonnage == %

Calculer  $N_{\text{eff}} = \left[ \sum_{i=1}^N \left( \tilde{w}_t^{(i)} \right)^2 \right]^{-1}$  puis pour  $\eta = 0.8 \times N$  :

**if** ( $N_{\text{eff}} \leq \eta$ ) **then**

Rééchantillonnage des particules : duplication des particules qui ont un poids élevé et suppression des autres, les particules résultantes sont notées  $x_t^{(i)}$  et leur poids sont  $w_t^{(i)} = \frac{1}{N}$

**end if**

---

La fonction  $h$  est donnée par :

$$\begin{aligned}
& h(w_{t-1}^{(i)}, z_{k,t}, \tilde{c}_{1,t}^{(i)}, \tilde{\alpha}_{1,t}^{(i)}, \alpha_{t-1}^{(i)}, \tilde{\alpha}_t^{(i)}, \alpha_{1,t-1}^{(i)}, \tilde{x}_t^{(i)}, x_{t-1}^{(i)}, \tilde{\sigma}_{1,t}^{(i)}, \sigma_{1,t-1}^{(i)}, \tilde{\sigma}_{1,t}^{\alpha (i)} | \sigma_{1,t-1}^{\alpha (i)}) \\
&= w_{t-1}^{(i)} \frac{p(z_{1,t} | \tilde{x}_t^{(i)}, \tilde{c}_{1,t}^{(i)}) p(\tilde{x}_t^{(i)} | x_{t-1}^{(i)})}{q(\tilde{x}_t^{(i)} | x_{t-1}^{(i)}, \tilde{c}_{1,t}^{(i)}, z_{1,t})} \\
&\times \frac{p(\tilde{c}_{1,t}^{(i)} | \tilde{\alpha}_{1,t}^{(i)}) p(\tilde{\alpha}_{1,t}^{(i)} | \alpha_{1,t-1}^{(i)}, \tilde{\sigma}_{1,t}^{(i)})}{q(\tilde{c}_{1,t}^{(i)} | x_{t-1}^{(i)}, \alpha_{1,t-1}^{(i)}, z_{1,t}) q(\tilde{\alpha}_{1,t}^{(i)} | \alpha_{1,t-1}^{(i)})} \\
&\times \frac{p(\tilde{c}_{1,t}^{(i)} | \tilde{\alpha}_{1,t}^{(i)}) p(\tilde{\sigma}_{1,t}^{\alpha (i)} | \sigma_{1,t-1}^{\alpha (i)})}{q(\tilde{\sigma}_{1,t}^{\alpha (i)} | \sigma_{1,t-1}^{\alpha (i)}, \tilde{\alpha}_t^{(i)}), \alpha_{t-1}^{(i)}}
\end{aligned} \tag{6}$$

## B Algorithme de filtrage particulaire Rao-Blackwellisé avec changement du modèle d'observation et un UKF

---

**Algorithm 2** Algorithme de filtrage particulaire Rao-Blackwellisé avec changement du modèle d'observation et un UKF

---

% == Initialisation == %

**for** particules  $i \in \{1, \dots, N\}$  **do**

Tirer :  $\hat{x}_{0|0}^{(i)} \sim p_0(\hat{x}_{0|0})$ ,  $\Sigma_{0|0}^{(i)} \sim p_0(\Sigma_{0|0})$ ,  $\sigma_{1,0}^{\alpha(i)} \sim p_0(\sigma_{1,0}^{\alpha})$ ,  $\alpha_{1,0}^{(i)} \sim p_0(\alpha_{1,0}|\sigma_{1,0}^{(i)})$  et  $c_0^{(i)} \sim p_0(c_0|\alpha_0^{(i)})$

**end for**

% == Itérations == %

**for**  $t \in \{1, 2, \dots\}$  **do**

Attendre l'arrivée d'une mesure  $z_{k,t}$  réalisée par le capteur  $k$  :

**for** particules  $i \in \{1, \dots, N\}$  **do**

**if** arrivée de mesures  $z_{2,t}$  et  $z_{3,t}$  **then**

Mise à jour de la moyenne  $\hat{x}_{t|t}^{(i)}$  et de la matrice de covariance  $\Sigma_{t|t}^{(i)}$  à l'aide de UKF :

$$\left(\hat{x}_{t|t}^{(i)}, \Sigma_{t|t}^{(i)}\right) = \text{UKF}(\hat{x}_{t-1|t-1}^{(i)}, \Sigma_{t-1|t-1}^{(i)}, z_{2,t}, z_{3,t})$$

Mise à jour des variables liées au capteur GPS

Mise à jour de :  $c_{1,t}^{(i)} \leftarrow c_{1,t-1}^{(i)}$ ,  $\alpha_{1,t}^{(i)} \leftarrow \alpha_{1,t-1}^{(i)}$  et  $\sigma_{1,t}^{\alpha(i)} \leftarrow \sigma_{1,t-1}^{\alpha(i)}$

Mise à jour des poids :  $\tilde{w}_t^{(i)} = w_{t-1}^{(i)} p(z_{2,t}|z_{1:t-1}) p(z_{3,t}|z_{1:t-1})$  puis les renormaliser telle que leur somme soit égale à 1

**end if**

**if** arrivée d'une mesure  $z_{1,t}$  du capteur GPS **then**

Tirer la variable décrivant l'état du capteur :  $\tilde{c}_{1,t}^{(i)} \sim q(c_{1,t}|\hat{x}_{t-1|t-1}^{(i)}, \Sigma_{t-1|t-1}^{(i)}, \alpha_{1,t-1}^{(i)}, z_{1,t})$

Tirer les probabilités :  $\tilde{\alpha}_{1,t}^{(i)} \sim q(\alpha_{1,t}|\alpha_{1,t-1}^{(i)}, \tilde{c}_{1,t}^{(i)}, \sigma_{1,t-1}^{\alpha(i)})$

Tirer l'hyperparamètre :  $\tilde{\sigma}_{1,t}^{\alpha(i)} \sim q(\sigma_{1,t}^{\alpha}|\sigma_{1,t-1}^{\alpha(i)}, \tilde{\alpha}_{1,t}^{(i)}, \alpha_{1,t-1}^{(i)})$

Mise à jour de la moyenne  $\hat{x}_{t|t}^{(i)}$  et de la matrice de covariance  $\Sigma_{t|t}^{(i)}$  à l'aide de UKF :

$$\left(\hat{x}_{t|t}^{(i)}, \Sigma_{t|t}^{(i)}\right) = \text{UKF}(\hat{x}_{t-1|t-1}^{(i)}, \Sigma_{t-1|t-1}^{(i)}, \tilde{c}_{1,t}^{(i)}, z_{1,t})$$

Mise à jour des poids : ( $f$  définie ci-dessous, équation (7))

$$\tilde{w}_t^{(i)} \propto f(w_{t-1}^{(i)}, z_{1,t}, z_{1:t-1}, \hat{x}_{t-1|t-1}^{(i)}, \Sigma_{t-1|t-1}^{(i)}, \tilde{c}_{1,t}^{(i)}, \tilde{\alpha}_{1,t}^{(i)}, \alpha_{1,t-1}^{(i)}, \tilde{\sigma}_{1,t}^{\alpha(i)}, \sigma_{1,t-1}^{\alpha(i)})$$

puis les renormaliser telle que leur somme soit égale à 1

**end if**

**end for**

**end for**

---



---

% == Rééchantillonnage == %

Calculer  $N_{\text{eff}} = \left[ \sum_{i=1}^N \left( \tilde{w}_t^{(i)} \right)^2 \right]^{-1}$  puis pour  $\eta = 0.8 \times N$  :

**if** ( $N_{\text{eff}} \leq \eta$ ) **then**

Rééchantillonnage des particules : duplication des particules qui ont un poids élevé et suppression des autres, les particules résultantes sont notées  $x_t^{(i)}$  et leur poids sont  $w_t^{(i)} = \frac{1}{N}$

**end if**

---

La fonction  $f$  est donnée par :

$$\begin{aligned}
 f(w_{t-1}^{(i)}, z_{1,t}, z_{1:t-1}, \tilde{c}_{1,1:t}^{(i)}, \tilde{\alpha}_{1,t}^{(i)}, \alpha_{1,t-1}^{(i)}, \hat{x}_{t-1|t-1}^{(i)}, \Sigma_{t-1|t-1}^{(i)}, \tilde{\sigma}_{1,t}^{\alpha(i)}, \sigma_{1,t-1}^{\alpha(i)}) \\
 = w_{t-1}^{(i)} \frac{p(z_{1,t} | \tilde{c}_{1,1:t}^{(i)}, z_{1:t-1}) p(\tilde{c}_{1,t}^{(i)} | \tilde{\alpha}_{1,t}^{(i)})}{q(\tilde{c}_{1,t}^{(i)} | \hat{x}_{t-1|t-1}^{(i)}, \Sigma_{t-1|t-1}^{(i)}, \alpha_{1,t-1}^{(i)}, z_{1,t})} \\
 \times \frac{p(\tilde{\alpha}_{1,t}^{(i)} | \alpha_{1,t-1}^{(i)}, \tilde{\sigma}_{1,t}^{\alpha(i)}) p(\tilde{\sigma}_{1,t}^{\alpha(i)} | \sigma_{1,t-1}^{\alpha(i)})}{q(\tilde{\alpha}_{1,t}^{(i)} | \alpha_{1,t-1}^{(i)}, \tilde{c}_{1,t}^{(i)}, \sigma_{1,t-1}^{\alpha(i)}) q(\tilde{\sigma}_{1,t}^{\alpha(i)} | \sigma_{1,t-1}^{\alpha(i)}, \tilde{\alpha}_{1,t}^{(i)}, \alpha_{1,t-1}^{(i)})}
 \end{aligned} \tag{7}$$

## C Quelques précisions sur le filtre de Kalman "unscented"

Le filtre de Kalman "unscented" repose sur la **transformée "unscented"**. Il s'agit d'une méthode qui permet de calculer la moyenne et la variance d'une variable aléatoire qui a subi une transformation non-linéaire. Nous notons  $g$  l'application non-linéaire,  $X$  la variable aléatoire transformée qui admet comme moyenne  $m_X$  et comme matrice de covariance  $\Sigma_X$ . On pose :  $Y = g(X)$ .

Pour calculer les deux premiers moments de  $Y$ , on utilise la transformée "unscented" de la sorte :

- Choisir de **façon déterministe**  $2n_X + 1$  particules pondérées  $(\mathcal{X}_i, w_i)$  afin qu'elles représentent correctement la moyenne  $m_X$  et la covariance  $\Sigma_X$  :

$$\left\{ \begin{array}{ll} \text{Pour } i = 0 : & \mathcal{X}_0 = m_X \quad w_0 = \frac{\kappa}{n_X + \kappa} \\ \text{Pour } i \in \{1, \dots, n_X\} : & \mathcal{X}_i = m_X + \left( \sqrt{(n_X + \kappa) \Sigma_X} \right)_i \quad w_i = \frac{1}{2(n_X + \kappa)} \\ \text{Pour } i \in \{n_X + 1, \dots, 2n_X\} : & \mathcal{X}_i = m_X - \left( \sqrt{(n_X + \kappa) \Sigma_X} \right)_i \quad w_i = \frac{1}{2(n_X + \kappa)} \end{array} \right.$$

où  $\kappa$  est le paramètre d'échelle tel que  $\kappa + n_X \neq 0$  et  $\left( \sqrt{(n_X + \kappa) \Sigma_X} \right)_i$  est la  $i^{\text{ème}}$  ligne de la matrice  $L$  telle que  $L^T L = (n_X + \kappa) \Sigma_X$ . Les poids  $w_i$  doivent être normalisés.

- Propager les particules à l'aide de la fonction non-linéaire  $g$  :  $\mathcal{Y}_i = g(\mathcal{X}_i)$  pour  $i \in \{0, 1, \dots, 2n_X\}$
- Les deux premiers moments de  $Y$  sont alors donnés par :

$$\hat{m}_Y = \sum_{i=0}^{2n_X} w_i \mathcal{Y}_i \tag{8}$$

$$\Sigma_Y = \sum_{i=0}^{2n_X} w_i (\mathcal{Y}_i - \hat{m}_Y)(\mathcal{Y}_i - \hat{m}_Y)^T \tag{9}$$

Considérons maintenant un problème de filtrage non-linéaire classique : les variables  $Z$  sont observées et nous cherchons à estimer  $x$ . Nous supposons que la densité postérieure au temps  $t - 1$  est gaussienne et est définie par :

$$p(x_{t-1}|Z_{t-1}) = \mathcal{N}(x_{t-1}; \hat{x}_{t-1|t-1}, \Sigma_{t-1|t-1}) \quad (10)$$

La première étape consiste à représenter cette densité à l'aide de  $N$  particules  $\mathcal{X}_{t-1}^{(i)}$  et leur poids  $w_{t-1}^{(i)}$  pour  $i \in \{0, 1, \dots, N-1\}$ . L'étape suivante est appelée prédiction et nécessite de calculer :

$$\hat{x}_{t|t-1} = \sum_{i=0}^{N-1} w_{t-1}^{(i)} f_{t-1}(\mathcal{X}_{t-1}^{(i)}) \quad (11)$$

$$\Sigma_{t|t-1} = Q_{t-1} + \sum_{i=0}^{N-1} w_{t-1}^{(i)} \left( f_{t-1}(\mathcal{X}_{t-1}^{(i)}) - \hat{x}_{t|t-1} \right) \left( f_{t-1}(\mathcal{X}_{t-1}^{(i)}) - \hat{x}_{t|t-1} \right)^T \quad (12)$$

La densité prédite  $p(x_t|Z_{t-1}) \approx \mathcal{N}(x_t; \hat{x}_{t|t-1}, \Sigma_{t|t-1})$  est représentée par un ensemble de  $N$  particules  $\mathcal{X}_{t|t-1}^{(i)} = f_{t-1}(\mathcal{X}_{t-1}^{(i)})$ . La mesure prédite est alors  $\hat{z}_{t|t-1} = \sum_{i=0}^{N-1} w_{t-1}^{(i)} h_t(\mathcal{X}_{t|t-1}^{(i)})$ .

L'étape de mise à jour des variables est donnée par :

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t(z_t - \hat{z}_{t|t-1}) \quad (13)$$

$$\Sigma_{t|t} = \Sigma_{t|t-1} - K_t S_t K_t^T \quad (14)$$

où  $K_t = \Sigma_{xz} S_t^{-1}$ ,  $S_t = R_t + \Sigma_{zz}$  et :

$$\Sigma_{xz} = \sum_{i=0}^{N-1} w_{t-1}^{(i)} \left( \mathcal{X}_{t|t-1}^{(i)} - \hat{x}_{t|t-1} \right) \left( h_t(\mathcal{X}_{t|t-1}^{(i)}) - \hat{z}_{t|t-1} \right)^T \quad (15)$$

$$\Sigma_{zz} = \sum_{i=0}^{N-1} w_{t-1}^{(i)} \left( h_t(\mathcal{X}_{t|t-1}^{(i)}) - \hat{z}_{t|t-1} \right) \left( h_t(\mathcal{X}_{t|t-1}^{(i)}) - \hat{z}_{t|t-1} \right)^T \quad (16)$$