

Nom, prénom :

NGUYEN Trung Duong

Responsable pédagogique UTT :

Antoine GRALL

Anne BARROS

Branche : Master OSS

Année : 2013

Semestre : P13

## Titre du stage

Approche de type filtre particulaire pour le problème d'estimation de la vie résiduelle (RUL)

## Résumé

Mon stage s'est déroulé au sein du Laboratoire de Modélisation et Sécurité des Systèmes, Université de Technologie de Troyes sous la direction de M. Antoine GRALL et Mme. Anne BARROS.

L'objectif de ce stage est d'estimer l'état de dégradation d'un système à un instant donné en basant sur les observations contaminées disponibles et puis prévoir l'évolution de la dégradation dans le futur pour calculer la vie résiduelle. L'outil utilisé est le filtre particulaire.

Dans le cadre de ce stage, j'ai modélisé la dégradation par un processus Gamma. La base théorique et les algorithmes du filtre particulaire sont révisés. L'implémentation du filtre particulaire est visualisée par une instruction sur la programmation en Matlab. Après avoir estimé l'état de dégradation courant et la RUL correspondant, la loi de temps d'atteinte est étudiée. De plus, j'ai effectué une comparaison sur la performance du filtre particulaire lors d'une variation des paramètres du processus Gamma. Finalement, les résultats de simulation sont vérifiés.

Entreprise : LM2S – ICD - UTT

Lieu : Troyes

Responsable : Antoine GRALL

Anne BARROS

Mots clés (CF Thésaurus)

Mot clé n°1 : Modélisation probabiliste

Mot clé n°2 : processus Gamma, dégradation

Mot clé n°3 : simulation de Monte Carlo, filtre particulaire

Mot clé n°4 : estimation de paramètres, RUL

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Problématique . . . . .	2
1.2	Modèle d'espace d'état . . . . .	2
<b>2</b>	<b>Base mathématique du filtre particulière</b>	<b>4</b>
2.1	Estimation Bayésienne . . . . .	4
2.2	Méthode de Monte Carlo . . . . .	5
2.2.1	Échantillonnage d'importance . . . . .	5
2.2.2	Échantillonnage d'importance séquentielle . . . . .	7
2.2.3	Algorithme du SIS filtre particulière . . . . .	10
<b>3</b>	<b>Simulation en Matlab du SIS filtre particulière</b>	<b>11</b>
3.1	Processus Gamma . . . . .	11
3.2	Loi d'importance populaire . . . . .	11
3.3	Implémentation Matlab . . . . .	12
3.3.1	Création des valeurs de mesures ( $Y_t$ ) . . . . .	12
3.3.2	Étape d'initialisation . . . . .	13
3.3.3	Étape de prédiction . . . . .	13
3.3.4	Étape de correction et d'estimation . . . . .	14
3.4	Principe du SIS filtre particulière . . . . .	15
3.5	Distribution des états estimés . . . . .	20
<b>4</b>	<b>Filtre particulière SISR et l'estimation de la RUL</b>	<b>23</b>
4.1	Filtre Bootstrap . . . . .	23
4.2	Filtre SISR avec redistribution d'adaptation . . . . .	25
4.3	Estimation de la RUL . . . . .	29
4.4	Résultats de simulation . . . . .	30
4.4.1	Quand le nombre d'échantillons ( $N_s$ ) augmente . . . . .	30
4.4.2	Quand l'écart-type du bruit de mesures ( $\sigma_\varepsilon$ ) varie . . . . .	32
4.4.3	Loi du temps d'atteinte . . . . .	33
<b>5</b>	<b>Conclusion</b>	<b>36</b>

<b>Bibliography</b>	<b>37</b>
---------------------	-----------

## Notations

Pour mieux comprendre le rapport, il faut faire attention à des notations suivantes :

- $X_t = \{x_0, x_1, \dots, x_t\}$  représente l'ensemble des états réels (cachés) du processus de dégradation jusqu'à l'instant  $(t)$ .
- $Y_t = \{y_0, y_1, \dots, y_t\}$  représente l'ensemble des valeurs de mesures correspondant respectivement à des états  $(x_t)$ . En supposant que au début  $(t = 0)$ , on n'effectue pas aucune mesure, donc  $Y_t = \{y_1, \dots, y_t\}$  effectivement.
- $(x_t^i, i = 1 : N_s)$  désigne l'échantillon  $(i)$  de l'état  $(x_t)$ .
- $\{x_t^i\}_{i=1}^{N_s}$  désigne un ensemble contenant  $N_s$  échantillons.
- Dans ce rapport, quand on parle d'un échantillon, on souhaite parler de la position de cet échantillon. Et quand cet échantillon est assigné d'un poids, il devient une *particule*.
- Dans quelques articles sur le filtre particulaire, le terme “*filtering distribution*” est utilisé pour désigner la loi marginale  $p(x_t | Y_t)$  de la loi a posteriori  $p(X_t | Y_t)$ . Pour la raison de simplicité, dans ce rapport on utilise abusivement le terme loi *a posteriori* pour indiquer cette loi marginale. En effet, avant d'estimer la vie résiduelle à un instant donné  $(t)$ , il s'agit d'un problème d'estimer le niveau de dégradation  $(x_t)$  à cet instant en tenant compte des valeurs de mesures  $(Y_t)$ . Ce travail préliminaire peut être réalisé par l'étude de la loi *a posteriori*  $p(x_t | Y_t)$ .

# 1 Introduction

## 1.1 Problématique

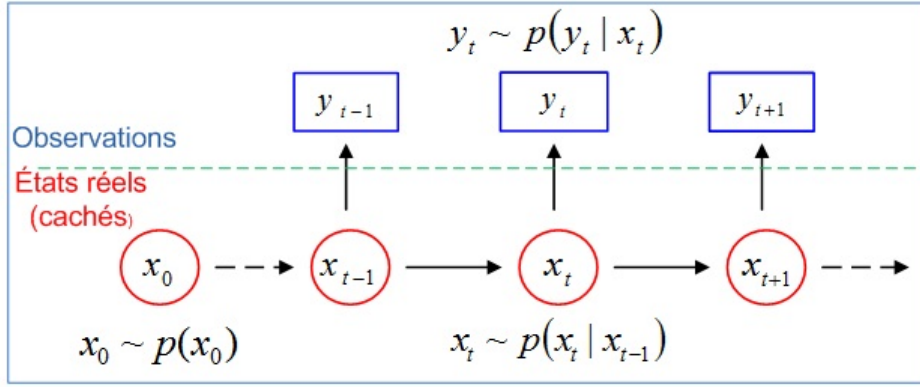
La durée de vie résiduelle (RUL) est définie comme le temps de fonctionnement restant d'un composant (ou d'un système) avant la date de défaillance. Ainsi la prédiction de cette durée joue un rôle important dans les stratégies de maintenance. En général, la RUL est estimée en s'appuyant sur des données obtenues à partir des mesures du niveau de dégradation jusqu'à l'instant courant. Pourtant, à cause de l'imparfait des capteurs ainsi que l'impact de l'environnement de travail, les données surveillées sont souvent contaminées. Cela pose des difficultés pour le diagnostic (retrouver le niveau de dégradation actuel) et par conséquent le pronostic (prévoir l'évolution dans le future de l'état de dégradation). Dans le cadre de ce rapport on s'intéresse en particulier à l'approche de type filtrage particulaire pour traiter ce problème.

Le filtre particulaire est une implémentation de l'estimation Bayésienne récursive en utilisant la méthode de Monte Carlo séquentielle (voir chapitre 2). L'idée est d'approcher à chaque instant ( $t$ ), la distribution *a posteriori*  $p(x_t | Y_t)$  du niveau de dégradation par une distribution discrète formée d'un ensemble d'échantillons avec leurs poids associés, qu'on appelle sous le terme *particules*  $\{x_t^i, W_t^i\}_{i=1}^{N_s}$ . Selon la loi de grand nombre, lorsque le nombre d'échantillons est grand, ils caractérisent bien la distribution *a posteriori*. Une fois qu'on a obtenu la loi *a posteriori*, on peut estimer le niveau de dégradation réel ( $x_t$ ). Finalement, après avoir estimé ( $x_t$ ), le calcul de la RUL est réalisé à l'aide de la simulation.

## 1.2 Modèle d'espace d'état

Le modèle d'espace d'état est l'outil fondamental pour représenter notre problème. Ce modèle constitue deux équations: l'équation d'état et l'équation d'observation. La première décrit le processus de dégradation en le modélisant par une suite des états dépendant du temps tandis que la deuxième donne la valeur de mesure *bruitée* du niveau de dégradation courant. Un tel modèle est montré ci-dessous:

$$\begin{cases} x_t = f(x_{t-1}, v_t) \\ y_t = h(x_t, \varepsilon_t) \end{cases} \quad (1.1)$$



**Figure 1.1:** Modèle d'espace d'état

où  $f(\cdot)$  et  $h(\cdot)$  sont deux fonctions (linéaires ou non linéaires) supposées connues.  $(v_t) \sim p(v)$  et  $(\varepsilon_t) \sim q(\varepsilon)$  sont des bruits indépendants (pas nécessairement Gaussien) de processus et de mesure respectivement.

On suppose que la suite  $X_t = \{x_0, x_1, \dots, x_t\}$  est Markovien, c'est à dire conditionnellement aux états passés  $X_{t-1} = \{x_0, x_1, \dots, x_{t-1}\}$ , l'état présent ( $x_t$ ) ne dépend que de l'état précédent ( $x_{t-1}$ ) à travers la *densité de transition* (ou *noyau de transition*):

$$p(x_t | X_{t-1}) = p(x_t | x_{t-1}) \leftrightarrow f(x_t | x_{t-1}) \quad (1.2)$$

Une autre hypothèse importante est que la valeur de mesure ( $y_t$ ) sachant les états passés ( $X_{t-1}$ ) et les observations passées ( $Y_{t-1}$ ) ne dépend que de l'état présent ( $x_t$ ). Une telle relation est représentée par la *fonction de vraisemblance*:

$$p(y_t | Y_{t-1}, X_{t-1}) = p(y_t | x_t) \leftrightarrow h(y_t | x_t) \quad (1.3)$$

Par ailleurs, on suppose aussi que l'état initial ( $x_0$ ) est distribué selon une *loi initiale*  $p(x_0)$ .

La description probabiliste formée de trois éléments  $p(x_t | x_{t-1})$ ,  $p(y_t | x_t)$  et  $p(x_0)$  du modèle d'espace d'état est montrée dans la figure 1.1.

## 2 Base mathématique du filtre particulaire

Dans ce chapitre on présente l'approche Bayésienne pour le problème d'estimation ainsi que l'utilisation d'une technique d'approximation numérique de type Monte Carlo. Enfin, un algorithme général du filtre particulaire est donné dans la section 2.2.3.

### 2.1 Estimation Bayésienne

Le but est d'estimer de façon récursive l'état caché  $(x_t)_{t \geq 0}$  au vu des observations  $(y_0, \dots, y_t)$ . L'estimation Bayésienne récursive consiste deux phases: prédiction et correction (mise à jour). Dans la phase de prédiction, suppose que l'on dispose déjà la loi a posteriori  $p(X_{t-1} | Y_{t-1})$ , on peut trouver la loi a priori  $p(x_t | Y_{t-1})$  à l'instant  $(t)$  en utilisant l'équation Chapman - Kolmogorov:

$$p(x_t | Y_{t-1}) = \int p(x_t | x_{t-1}, Y_{t-1}) \times p(x_{t-1} | Y_{t-1}) dx_{t-1} = \int p(x_t | x_{t-1}) \times p(x_{t-1} | Y_{t-1}) dx_{t-1} \quad (2.1)$$

Dans la deuxième phase, afin de corriger la prédiction, on fait appel à la fonction de vraisemblance qui tient compte de la nouvelle valeur de mesure  $(y_t)$ . La loi *a posteriori*  $p(x_t | Y_t)$  est calculée à l'aide de la formule suivante:

$$p(x_t | Y_t) = \frac{p(y_t | x_t) \times p(x_t | Y_{t-1})}{p(y_t | Y_{t-1})} = \frac{p(y_t | x_t) \times p(x_t | Y_{t-1})}{\int p(y_t | x_t) \times p(x_t | Y_{t-1}) dx_t} \quad (2.2)$$

Le principe de l'estimation Bayésienne est présenté dans la figure 2.1.

Enfin, après avoir obtenu la loi a posteriori, le niveau de dégradation  $(x_t)$  est déterminé par un

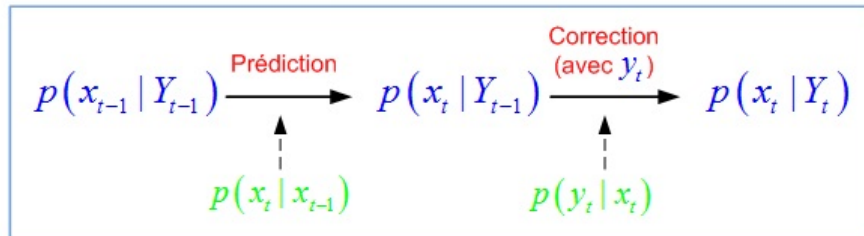


Figure 2.1: Deux phases de l'estimation Bayésienne

estimateur dit minimum de l'erreur quadratique moyenne:

$$x_{MMSE} = \int x_t \times p(x_t | Y_t) dx_t$$

Pourtant, on n'obtient que le calcul analytique des intégrations (2.1) et (2.2) dans très peu de cas particuliers (par exemple  $f(\cdot)$  et  $h(\cdot)$  sont linéaires,  $(v_t)$  et  $(\varepsilon_t)$  sont Gaussiens). Donc, on se propose d'utiliser la méthode de Monte Carlo qui offre une approximation des intégrations. La section 2.2 donne des interprétations mathématiques de cette méthode, ce qui nous aide à comprendre comment le filtre particulaire est construit.

## 2.2 Méthode de Monte Carlo

### 2.2.1 Échantillonnage d'importance

L'idée de base de la méthode de Monte Carlo consiste à utiliser la technique d'*échantillonnage d'importance*. Cette technique permet d'approcher l'intégration d'une densité de probabilité grâce à un ensemble des variables aléatoires générées selon une loi appelée *loi d'importance*.

Soit  $p(x)$  une densité de probabilité et  $f(x)$  une fonction quelconque, on considère l'intégration suivante:

$$E_p \{f(x)\} = \int f(x) \times p(x) dx \quad (2.3)$$

Si  $p(x)$  est plutôt complexe, il est nécessaire d'approcher cette intégration au lieu d'intégrer directement. Comme il est difficile d'échantillonner selon  $p(x)$ , on peut échantillonner selon une loi  $q(x)$  qui est plus simple, i.e:

$$X(i) \sim q(x), i = 1 : N_s$$

Ainsi:

$$q(x) \approx \frac{1}{N_s} \sum_{i=1}^{N_s} \delta(x - X(i)) \quad (2.4)$$

où  $\delta$  est la distribution de Dirac.

La loi des grands nombres assure que l'approximation (2.4) converge presque sûr à  $q(x)$  lorsque  $N_s \rightarrow \infty$ :

$$\frac{1}{N_s} \sum_{i=1}^{N_s} \delta(x - X(i)) \rightarrow q(x)$$



En utilisant (2.4) et la propriété de la distribution Dirac, l'intégration (2.3) est équivalent à:

$$\begin{aligned}
E_p \{f(x)\} &= \int f(x) \times p(x) dx = \int f(x) \times \left( \frac{p(x)}{q(x)} \right) \times q(x) dx \\
&\approx \int f(x) \times \left( \frac{p(x)}{q(x)} \right) \times \frac{1}{N_s} \sum_{i=1}^{N_s} \delta(x - X(i)) dx \\
&\approx \frac{1}{N_s} \sum_{i=1}^{N_s} f(X(i)) \times \left( \frac{p(X(i))}{q(X(i))} \right)
\end{aligned} \tag{2.5}$$

Dans ce cas  $q(x)$  est appelé loi d'importance. Le choix de la loi d'importance contribue le plus à la performance de la technique d'échantillonnage d'importance. En effet, à condition de pouvoir simuler selon  $q(x)$ , plus  $q(x)$  approche  $p(x)$ , plus le filtre particulaire est efficient.

De même, considère l'intégration:

$$E_p \{f(X_t)\} = \int f(X_t) \times p(X_t | Y_t) dX_t$$

On souhaite approcher cette intégration alors que l'on sait seulement échantillonner selon  $q(X_t | Y_t)$  au lieu de  $p(X_t | Y_t)$ . L'intégration ci-dessus peut être ré-écrit:

$$E_p \{f(X_t)\} = \int f(X_t) \times \left[ \frac{p(X_t | Y_t)}{q(X_t | Y_t)} \right] \times q(X_t | Y_t) dX_t$$

Grâce à la formule de Bayes:

$$p(X_t | Y_t) = \frac{p(Y_t | X_t) \times p(X_t)}{p(Y_t)}$$

on obtient:

$$\begin{aligned}
E_p \{f(X_t)\} &= \frac{1}{p(Y_t)} \int f(X_t) \times \left[ \frac{p(Y_t | X_t) \times p(X_t)}{q(X_t | Y_t)} \right] \times q(X_t | Y_t) dX_t \\
&= \frac{1}{\int p(Y_t | X_t) \times p(X_t) dX_t} \times \int f(X_t) \times \left[ \frac{p(Y_t | X_t) \times p(X_t)}{q(X_t | Y_t)} \right] \times q(X_t | Y_t) dX_t
\end{aligned}$$

En introduisant un facteur dit *poids d'importance* égale à la partie dans le crochet:

$$\omega_t = \frac{p(Y_t | X_t) \times p(X_t)}{q(X_t | Y_t)} \tag{2.6}$$

l'intégration au dessus peut être simplifiée à:

$$E_p \{f(X_t)\} = \frac{\int \omega_t \times f(X_t) \times q(X_t | Y_t) dX_t}{\int \omega_t \times q(X_t | Y_t) dX_t} \tag{2.7}$$

Si on génère  $(N_s)$  échantillons selon la loi d'importance  $q(X_t | Y_t)$ :  $X_t^i \sim q(X_t | Y_t)$ , d'où  $q(X_t | Y_t) \approx$

$\frac{1}{N_s} \sum_{i=1}^{N_s} \delta(X_t - X_t^i)$ , en reprenant le résultat (2.5) l'espérance (2.7) devient:

$$E_p \{f(X_t)\} \approx \frac{\int \omega_t \times f(X_t) \times \frac{1}{N_s} \sum_{i=1}^{N_s} \delta(X_t - X_t^i) dX_t}{\int \omega_t \times \frac{1}{N_s} \sum_{i=1}^{N_s} \delta(X_t - X_t^i) dX_t} \approx \frac{\sum_{i=1}^{N_s} \omega_t^i \times f(X_t^i)}{\sum_{i=1}^{N_s} \omega_t^i}$$

De plus, si le *poids d'importance normalisé* est défini telle que:

$$W_t^i = \frac{\omega_t^i}{\sum_{i=1}^{N_s} \omega_t^i}$$

où  $\omega_t^i = \frac{p(Y_t | X_t^i) \times p(X_t^i)}{q(X_t^i | Y_t)}$ , on obtient finalement l'approximation:

$$E_p \{f(X_t)\} = \int f(X_t) \times p(X_t | Y_t) dX_t \approx \sum_{i=1}^{N_s} W_t^i \times f(X_t^i)$$

à partir de laquelle on peut déduire l'approximation de la loi a posteriori:

$$p(X_t | Y_t) \approx \sum_{i=1}^{N_s} W_t^i \times \delta(X_t - X_t^i) \quad (2.8)$$

avec  $\sum_{i=1}^{N_s} W_t^i = 1$

On peut voir que la loi a posteriori  $p(X_t | Y_t)$  est approchée par une distribution discrète pondérée caractérisée par un ensemble des particules  $\{X_t^i, W_t^i\}_{i=1}^{N_s}$ .

### 2.2.2 Échantillonnage d'importance séquentielle

La technique d'échantillonnage d'importance est inappropriée pour le problème d'estimation récursive à cause du calcul coûteux du poids d'importance. En effet, chaque fois qu'une nouvelle valeur de mesure ( $y_t$ ) est disponible, le re-calcul du poids d'importance selon la formule (2.6) est vraiment pénible car on doit prendre en compte tous les valeurs de mesures dans le passé ( $Y_{t-1}$ ) ainsi que tous les états ( $X_t$ ). La technique d'échantillonnage d'importance séquentielle (*sequential important sampling* - SIS en anglais) est introduite pour traiter ce problème.

L'idée de la technique SIS est interprétée comme suit: Suppose que l'on dispose un ensemble de particules  $\{X_{t-1}^i, W_{t-1}^i\}_{i=1}^{N_s}$  qui approche la distribution a posteriori  $p(X_{t-1} | Y_{t-1})$ . À l'acquisition de ( $y_t$ ), on souhaite approcher  $p(X_t | Y_t)$  en propageant chaque particule de l'ensemble  $\{X_{t-1}^i, W_{t-1}^i\}_{i=1}^{N_s}$  suivant deux étapes: muter (prédire) les échantillons  $\{X_{t-1}^i\}_{i=1}^{N_s}$ , puis mettre à jour leurs poids d'importances  $\{W_{t-1}^i\}_{i=1}^{N_s}$  correspondants. De cette manière, la technique d'échantillonnage d'importance doit être modifiée de telle sorte que l'on peut approcher  $p(X_{t-1} | Y_{t-1})$  à l'aide des poids d'importances  $\{W_{t-1}^i\}_{i=1}^{N_s}$  et puis  $p(X_t | Y_t)$  avec des nouveaux poids  $\{W_t^i\}_{i=1}^{N_s}$ . Cette procédure est réalisé séquentiellement en faisant appel à une suite de loi d'importance telle que  $X_{t-1}^i \sim q(X_{t-1} | Y_{t-1})$ ,  $X_t^i \sim$

$q(X_t | Y_t)$  et ainsi de suite.

On s'intéresse maintenant à comment entraîner l'approximation séquentielle de la loi a posteriori. Reprend l'équation (2.6), on décompose la forme simplifiée de son dénominateur:

$$\begin{aligned}
 q(X_t | Y_t) &= q(x_t, X_{t-1} | Y_t) = \frac{q(x_t, X_{t-1}, Y_t)}{q(Y_t)} \\
 &= \frac{q(x_t | X_{t-1}, Y_t) \times q(X_{t-1}, Y_t)}{q(Y_t)} \\
 &= q(x_t | X_{t-1}, Y_t) \times \frac{q(X_{t-1} | Y_t) \times q(Y_t)}{q(Y_t)} \\
 &= q(x_t | X_{t-1}, Y_t) \times q(X_{t-1} | Y_t) \\
 &= q(x_t | X_{t-1}, Y_t) \times q(X_{t-1} | y_t, Y_{t-1})
 \end{aligned}$$

Comme les états au passé ( $X_{t-1}$ ) sont indépendants de l'observation courante ( $y_t$ ), donc  $q(X_{t-1} | y_t, Y_{t-1}) = q(X_{t-1} | Y_{t-1})$ . Alors:

$$q(X_t | Y_t) = q(X_{t-1} | Y_{t-1}) \times q(x_t | X_{t-1}, Y_t)$$

En utilisant cette équation, on obtient l'extension de (2.6):

$$\begin{aligned}
 \omega_t^i &= \frac{p(Y_t | X_t^i) \times p(X_t^i)}{q(X_t^i | Y_t)} = \frac{p(Y_t | X_t^i) \times p(X_t^i)}{q(X_{t-1}^i | Y_{t-1}) \times q(x_t^i | X_{t-1}^i, Y_t)} \\
 &= \frac{[p(Y_{t-1} | X_{t-1}^i) \times p(X_{t-1}^i)]}{q(X_{t-1}^i | Y_{t-1})} \times \frac{p(Y_t | X_t^i) \times p(X_t^i)}{[p(Y_{t-1} | X_{t-1}^i) \times p(X_{t-1}^i)] \times q(x_t^i | X_{t-1}^i, Y_t)} \\
 &= \omega_{t-1}^i \times \frac{p(Y_t | X_t^i) \times p(X_t^i)}{p(Y_{t-1} | X_{t-1}^i) \times p(X_{t-1}^i) \times q(x_t^i | X_{t-1}^i, Y_t)} \tag{2.9}
 \end{aligned}$$

où on peut facilement trouver que le poids d'importance est mise à jour récursivement.

On fait attention à deux densités  $p(Y_t | X_t^i)$  et  $p(X_t^i)$  du numérateur. Tout d'abord, on décompose  $p(X_t^i)$  en faisant appel à la propriété Markovien (1.2) de ( $X_t$ ):

$$p(X_t^i) = p(x_t^i, X_{t-1}^i) = p(x_t^i | X_{t-1}^i) \times p(X_{t-1}^i) = p(x_t^i | x_{t-1}^i) \times p(X_{t-1}^i) \tag{2.10}$$

Par ailleurs, l'extension de la formule de Bayes  $p(AB | C) = p(A | BC) \times p(B | C)$  nous donne l'expression:

$$p(Y_t | X_t^i) = p(y_t, Y_{t-1} | x_t^i, X_{t-1}^i) = p(y_t | Y_{t-1}, x_t^i, X_{t-1}^i) \times p(Y_{t-1} | x_t^i, X_{t-1}^i)$$

L'hypothèse (1.3) et le fait que des observations au passé ( $Y_{t-1}$ ) sont indépendantes avec l'état courant

$(x_t)$  favorisent respectivement des simplifications suivantes:

$$\begin{cases} p(y_t | Y_{t-1}, x_t^i, X_{t-1}^i) &= p(y_t | x_t^i) \\ p(Y_{t-1} | x_t^i, X_{t-1}^i) &= p(Y_{t-1} | X_{t-1}^i) \end{cases}$$

Par conséquent:

$$p(Y_t | X_t^i) = p(y_t | Y_{t-1}, x_t^i, X_{t-1}^i) \times p(Y_{t-1} | x_t^i, X_{t-1}^i) = p(y_t | x_t^i) \times p(Y_{t-1} | X_{t-1}^i) \quad (2.11)$$

Substituer (2.10) et (2.11) à (2.9), on obtient:

$$\begin{aligned} \omega_t^i &= \omega_{t-1}^i \times \frac{p(Y_t | X_t^i) \times p(X_t^i)}{p(Y_{t-1} | X_{t-1}^i) \times p(X_{t-1}^i) \times q(x_t^i | X_{t-1}^i, Y_t)} \\ &= \omega_{t-1}^i \times \frac{p(y_t | x_t^i) \times p(Y_{t-1} | X_{t-1}^i) \times p(x_t^i | x_{t-1}^i) \times p(X_{t-1}^i)}{p(Y_{t-1} | X_{t-1}^i) \times p(X_{t-1}^i) \times q(x_t^i | X_{t-1}^i, Y_t)} \\ &= \omega_{t-1}^i \times \frac{p(y_t | x_t^i) \times p(x_t^i | x_{t-1}^i)}{q(x_t^i | X_{t-1}^i, Y_t)} \end{aligned}$$

En outre, on peut reformuler  $q(x_t^i | X_{t-1}^i, Y_t) = q(x_t^i | x_{t-1}^i, y_t)$  en supposant que la loi d'importance ne dépend que de l'état précédent  $(x_{t-1})$  et de l'observation courante  $(y_t)$ . Une telle supposition est raisonnable et paraît très commun dans le contexte de filtrage. En effet, ce qu'on veut savoir véritablement est une approximation de la loi a posteriori "*marginale*"  $p(x_t | Y_t)$  à chaque instant  $(t)$ . On peut voir dans le chapitre 3, que l'on n'est pas strictement obligé de sauvegarder ni la trajectoire passée  $\{X_{t-1}^i\}_{i=1}^{N_s}$  ni les observations passées  $\{Y_{t-1}\}$  lors de la programmation. Alors, l'expression de calcul du poids d'importance est:

$$\omega_t^i = \omega_{t-1}^i \times \frac{p(y_t | x_t^i) \times p(x_t^i | x_{t-1}^i)}{q(x_t^i | x_{t-1}^i, y_t)} \quad (2.12)$$

La loi a posteriori est une modification de (2.8):

$$p(x_t | Y_t) \approx \sum_{i=1}^{N_e} W_t^i \times \delta(x_t - x_t^i) \quad (2.13)$$

dans laquelle  $W_t^i = \frac{\omega_t^i}{\sum_{i=1}^{N_s} \omega_t^i}$  avec  $(\omega_t^i)$  précisé dans l'équation (2.12).

Le niveau de dégradation estimé est donc:

$$x_{MMSE} = \int x_t \times p(x_t | Y_t) dx_t \approx \sum_{i=1}^{N_s} x_t \times (W_t^i \times \delta(x_t - x_t^i)) \approx \sum_{i=1}^{N_s} W_t^i \times x_t^i$$

**Algorithme 2.1** Algorithme du SIS filtre particulaireÀ l'instant ( $t = 0$ )**Initialisation** ( $i = 1 : N_s$ )

Générer un ensemble d'échantillons i.i.d selon la loi initiale:

$$x_0^i \sim p(x_0)$$

Assigner les poids d'importances:

$$\omega_0^i = \frac{1}{N_s}$$

À partir de l'instant ( $t \geq 1$ ), sachant que l'on dispose un ensemble de particules  $\{x_{t-1}^i, \omega_{t-1}^i\}_{i=1}^{N_s}$ **Échantillonnage d'importance**

Générer un nouvel ensemble des échantillons selon la loi d'importance:

$$x_t^i \sim q(x_t^i | x_{t-1}^i, y_t), i = 1 : N_s$$

Déterminer la fonction de vraisemblance  $p(y_t | x_t^i)$  et le noyau de transition  $p(x_t^i | x_{t-1}^i)$ , puis calculer les poids d'importances selon la formule (2.12).**Normaliser**

$$W_t^i = \frac{\omega_t^i}{\sum_{i=1}^{N_s} \omega_t^i}, i = 1 : N_s$$

**Estimer le niveau de dégradation**

$$x_{MMSE} \approx \sum_{i=1}^{N_s} W_t^i \times x_t^i$$

**2.2.3 Algorithme du SIS filtre particulaire**

On est maintenant capable de montrer dans l' **Algorithme 2.1** les composants principaux d'un filtre particulaire SIS. On trouve que c'est un algorithme de type génétique où un ensemble de particules est propagé avec le temps.

L'avantage du filtre particulaire est qu'il n'existe aucune restriction de la forme de deux fonctions  $f(\cdot)$  et  $h(\cdot)$  (linéaire ou non linéaire) ainsi que le type de bruit ( $v_t$ ) et ( $\varepsilon_t$ ) (Gaussien ou non Gaussien) dans le modèle d'espace d'état (1.1). Pour mettre en oeuvre l'algorithme décrit ci dessus, il est nécessaire de savoir:

- simuler selon la loi initiale  $p(x_0)$
- simuler selon la loi d'importance  $q(x_t | x_{t-1}, y_t)$
- calculer la vraisemblance  $p(y_t | x_t)$  pour tout ( $x_t$ ) à l'acquisition de ( $y_t$ )

## 3 Simulation en Matlab du SIS filtre particulaire

Le troisième chapitre présente dans un premier temps le modèle de dégradation utilisé. Puis, l'implémentation Matlab du SIS filtre particulaire est donnée sous forme d'un tutoriel. Au cours de ce tutoriel, le principe ainsi que des problèmes concernant le filtre sont clarifiés. Finalement, une comparaison de l'effet du SIS filtre particulaire sur deux processus Gamma de l'incrément différents est étudiée.

### 3.1 Processus Gamma

Dans notre cas d'étude, la dégradation du système est modélisée par un processus Gamma alors que le bruit de mesure est supposé de type Gaussien additif de moyenne nulle. Alors, le modèle d'espace d'état prend la forme suivante:

$$\begin{cases} x_t = x_{t-1} + v_t \\ y_t = x_t + \varepsilon_t \end{cases} \text{ avec } \begin{cases} v_t \sim \Gamma(k, \theta) \\ \varepsilon_t \sim N(0, \sigma_\varepsilon^2) \end{cases} \quad (3.1)$$

$(v_t)$  est une variable aléatoire représente l'incrément du processus Gamma, donc dans l'équation d'état il joue le rôle du bruit de processus.

$(k)$  est le paramètre de forme,  $(\theta)$  est le paramètre d'intensité de la distribution Gamma.

$(\sigma_\varepsilon)$  est l'écart-type de la distribution Normale.

Le problème est d'estimer le niveau de dégradation (caché)  $(x_t)$  grâce aux observations  $(Y_t)$  en supposant que le modèle 3.1 est connu.

### 3.2 Loi d'importance populaire

Choisir la loi d'importance est l'étape fondamentale lorsque l'on veut mettre en application un filtre particulaire. En général, effectuer le choix comme suit:

$$q(x_t | x_{t-1}, y_t) = p(x_t | x_{t-1})$$

permet de mettre à jour simplement le poids d'importance grâce à la fonction de vraisemblance:

$$\omega_t^i = \omega_{t-1}^i \times \frac{p(y_t | x_t^i) \times p(x_t^i | x_{t-1}^i)}{q(x_t^i | x_{t-1}^i, y_t)} = \omega_{t-1}^i \times p(y_t | x_t^i) \quad (3.2)$$

La manière de choisir la loi d'importance correspond identiquement au noyau de transition paraît comme le plus populaire. C'est aussi la loi d'importance qu'on utilise désormais dans ce rapport. Le point positif de cette approche est qu'elle facilite le calcul récursif du poids d'importance. Pourtant, le fait que la loi d'importance ne tient pas compte des observations, en particulier celle la plus récente ( $y_t$ ), va causer un filtre sensible avec horsains. Plus l'observation est bruitée, plus le SIS filtre particulaire exprime une pauvre qualité. De toute façon, il est nécessaire de remarquer que l'on dispose d'autres solutions plus sophistiquées dans le choix de la loi d'importance. Celle que l'on prend ci dessus est la plus simple pour l'étude d'un filtre particulaire.

### 3.3 Implémentation Matlab

#### 3.3.1 Création des valeurs de mesures ( $Y_t$ )

D'abord on génère une suite de variables aléatoires  $(x_t)_{t>0}$  représentant le niveau de dégradation réel en utilisant le noyau de transition. Puis, à chaque instant ( $t$ ),  $(x_t)$  est additionné d'un bruit Gaussien pour donner une valeur de mesure correspondante. En pratique, le filtre particulaire va re-estimer l'état  $(x_t)$  (caché) en basant sur ces valeurs de mesures. Le programme Matlab qui réalise la création de  $(Y_t)$  est montré ci-dessous:

```
T = 500; % L'instant final
%% À chaque instant, on fait ...
for t = 1:T
    % Mise à jour le niveau de dégradation accumulé
    x = x + gamrnd(a,b);
    % Créer la valeur de mesure correspondante
    y = x + dev_noise*randn;
end
```

Note que:

- $\text{gamrnd}(a, b)$  génère une variable aléatoire selon la loi  $\Gamma(k, \theta)$
- $\text{dev\_noise}$  représente l'écart-type ( $\sigma_\varepsilon$ ) du bruit de mesure
- $\text{dev\_noise} * \text{randn}$  génère une variable aléatoire selon la loi  $N(0, \sigma_\varepsilon^2)$

### 3.3.2 Étape d'initialisation

À l'instant ( $t = 0$ ), on doit initialiser le filtre, c'est-à-dire générer selon la loi initiale  $p(x_0)$  un ensemble d'échantillons indépendants  $\{x_0^i\}_{i=1}^{N_s}$  associés d'un même valeur de poids d'importance  $\omega_0^i = \frac{1}{N_s}$ . Quand le temps passe, ces particules  $\{x_0^i, \omega_0^i\}_{i=1}^{N_s}$  vont s'évoluer indépendamment. Et les poids  $\{\omega_t^i\}_{i=1}^{N_s}$  se diffèrent, ce qui met en évidence "l'importance" d'un échantillon par rapport aux autres. Après quelques itérations ( $t$ ), on constate que le grand écart entre "l'importance" des échantillons cause une estimation fragile de l'état de dégradation. Ce phénomène qu'on appelle *dégénérescence de poids* sera étudié après.

Supposer que au début le système est neuf ( $x_0 = 0$ ) et on ne fait aucune observation, donc ( $y_0$ ) n'existe pas. Supposer aussi que la loi initiale est la loi Normale  $p(x_0) = N(0, \sigma_0^2)$ . L'étape d'initialisation est montrée dans le tableau suivant:

```
%% Initialiser le filtre particulaire
x = 0; % L'état initial
x_P = zeros(1,Ns); % L'ensemble des échantillons représentant l'état de dégradation
P_w = zeros(1,Ns); % Les poids d'importances associés à des échantillons
% Simuler selon la loi initiale
for i = 1:Ns
    x_P(i) = x + dev_init_sample*randn;
    P_w(i) = 1/Ns;
end
```

Note que:

- *dev\_init\_sample* représente l'écart-type ( $\sigma_0$ ) de la loi initiale
- *dev\_init\_sample\*randn* génère une variable aléatoire selon la loi  $N(0, \sigma_\varepsilon^2)$

On dispose maintenant un ensemble des particules  $\{x_0^i, P\_w_0^i\}_{i=1}^{N_s}$  qui représentent l'état initial. Ces particules caractérisent la loi a priori sous le point de vue de l'estimation Bayésienne. À l'acquisition de la valeur de mesure ( $y_1$ ), le filtre particulaire se met en marche. Le corps du filtre constitue deux phases qui servent à générer les échantillons selon la loi d'importance, puis les pondérer (mettre à jour leurs poids d'importances) pour former un nouvel ensemble  $\{x\_P\_update_1^i, P\_w_1^i\}_{i=1}^{N_s}$ , qui approche la loi a posteriori  $p(x_1 | y_1)$ .

### 3.3.3 Étape de prédiction

Car le noyau de transition est choisi comme la loi d'importance, donc la première phase, prédiction (échantillonnage) consiste à muter les échantillons à l'aide de ce noyau. On obtient un nouvel ensemble d'échantillons indépendants  $\{x\_P\_update_1^i\}_{i=1}^{N_s}$ .



```

T = 500; % L'instant final
%% À chaque instant, on fait ...
for t = 1:T
    % Initialiser le filtre particulaire
    % ...
    % Le corps du filtre particulaire
    % Chaque particule s'évolue indépendamment
    for i = 1:Ns
        % Phase de prédiction
        % Muter l'échantillon à l'aide du noyau de transition
        x_P_update(i) = x_P(i) + gamrnd (a,b);
        % Phase de correction
        % ...
    end
    % ...
end

```

### 3.3.4 Étape de correction et d'estimation

Lorsque la première valeur de mesure ( $y_1$ ) est disponible, on évalue la vraisemblance de chaque échantillon ( $x_{P\_update_1^i}, i = 1 : N_s$ ) et calcule le poids correspondant à chacun selon la formule (3.2). Le poids quantifie l'adéquation de chaque échantillon par rapport à l'observation courante. L'échantillon qui offre une valeur (trouvée grâce à l'équation d'observation) plus proche de ( $y_1$ ) va être assigné un poids plus significatif. Pour effectuer un tel calcul, on doit tout d'abord déterminer la fonction de vraisemblance  $p(y_1 | x_1)$ . Grâce à l'équation d'observation du modèle (3.1), on trouve facilement que la fonction de vraisemblance correspond à la loi  $N(x_t, \sigma_\varepsilon^2)$ . Donc:

$$p(y_1 | x_1^i) = \frac{1}{\sigma_\varepsilon \sqrt{2\pi}} \times \exp\left(-\frac{(x_1^i - y_1)^2}{2\sigma_\varepsilon^2}\right)$$

Enfin, les travaux restent à normaliser les poids d'importances  $W_t^i = \frac{\omega_t^i}{\sum_{i=1}^{N_s} \omega_t^i}$  afin d'obtenir un ensemble de particules  $\{x_{P\_update_1^i}, P\_w_1^i\}_{i=1}^{N_s}$  et puis à estimer le niveau de dégradation  $x_{MMSE} \approx \sum_{i=1}^{N_s} W_t^i \times x_t^i$ .

```

T = 500; % L'instant final
%% À chaque instant, on fait ...
for t = 1:T
    % ...
    % Le corps du filtre particulaire
    % Chaque particule s'évolue indépendamment
    for i = 1:Ns
        % Phase de prédiction
        % ...
        % Phase de correction
        y_update(i) = x_P_update(i);
        P_w(i) = P_w(i) * (1/(dev_noise*sqrt(2*pi)))*exp(-(y_update(i)-y)^2/(2*dev_noise^2));
    end
    % Normaliser les poids d'importances
    P_w = P_w./sum(P_w);
    % Estimer le niveau de dégradation
    for i = 1:Ns
        x_estimate = x_estimate + P_w(i)*x_P_update(i);
    end
    % Propager les échantillons à l'instant suivant
    x_P = x_P_update;
end

```

L'instruction  $y\_update(i) = x\_P\_update(i)$  fait référence à l'équation d'observation  $y_t = x_t + \varepsilon$ . Pour notre problème, heureusement cette équation est simple. Mais pour les autres cas, par exemple les autres techniques de mesures, elle peut être non linéaire. Donc au lieu d'utiliser directement  $x\_P\_update(i)$  dans le calcul des poids d'importances, on ajoute l'instruction ci-dessus pour démontrer l'algorithme générale du SIS filtre particulaire. A la fin du cycle, les échantillons  $\{x\_P\_update_i^i\}_{i=1}^{N_s}$  doivent être propagés à l'instant ( $t = 2$ ) pour caractériser la loi a priori  $p(x_2 | y_1)$ .

Lors de la programmation, on est capable de sauvegarder les données concernant la propagation des particules, i.e  $\{x\_P_t^i\}_{i=1}^{N_s}$ ,  $\{x\_P\_update_t^i\}_{i=1}^{N_s}$ ,  $\{P\_w_t^i\}_{i=1}^{N_s}$  pour l'investigation si nécessaire.

### 3.4 Principe du SIS filtre particulaire

Maintenant on effectue la simulation pour visualiser les démarches du SIS filtre particulaire .

La figure 3.1 montre la répartition des échantillons selon la loi a priori  $p(x_0)$ . Une cinquantaine d'échantillons sont distribués aléatoirement selon la loi  $N(0, \sigma_0^2)$  avec  $\sigma_0 = 4$ . On peut voir dans la première image qu'il y a des "bons" échantillons qui se localisent près de l'état réel ( $x_0 = 0$ ) ainsi que

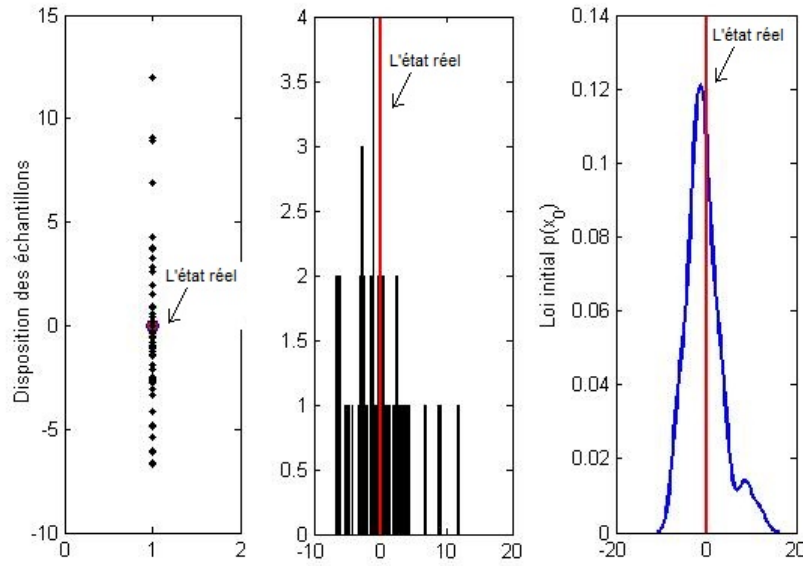


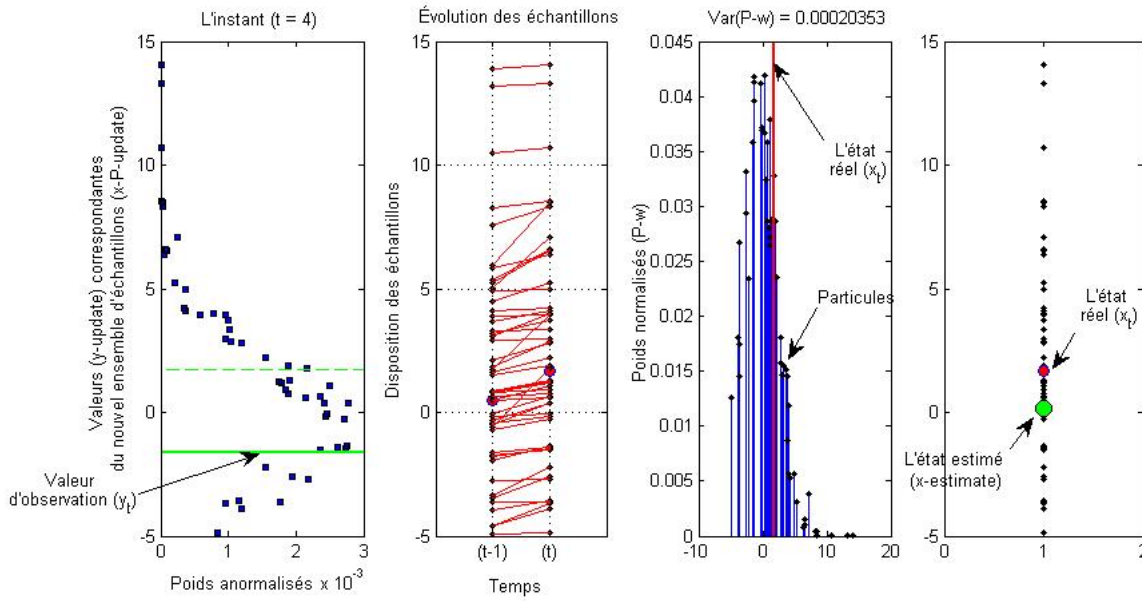
Figure 3.1: Loi a priori

des “mauvais” échantillons qui sont très distants de  $(x_0)$ . La densité de probabilité de la distribution des échantillons est dessinée dans le troisième image.

La figure 3.2 nous fait savoir ce qui se passe à l’instant  $(t = 4)$ . La première image démontre la pondération des échantillons. La deuxième image illustre la phase de prédiction: chaque échantillon  $(x_{-P_4^i} \equiv x_{-P\_update_3^i}, i = 1 : N_s)$  est muté vers sa nouvelle position  $(x_{-P\_update_4^i}, i = 1 : N_s)$ . L’ensemble de particules est montré dans le troisième image. La quatrième image localise en même temps l’état estimé  $(x_{-estimate_4})$  et l’état réel  $(x_4)$ .

On peut apercevoir qu’à l’instant  $(t = 4)$ , le niveau de dégradation réel atteint évidemment une valeur positive tandis que la valeur de mesure  $(y_4)$  donne une valeur négative à cause du bruit de mesure. À cet instant, comme le filtre vient de démarrer, on peut voir dans la première image que la plupart des échantillons  $(x_{-P\_update_4^i}, i = 1 : N_s)$  qui offrent une valeur  $(y_{-update_4^i}, i = 1 : N_s)$  proche de  $(y_4)$  vont être assignés des poids forts. Pourtant, il existe des cas exceptionnels. En effet, on trouve que certains échantillons prennent un poids plus fort même si ils sont des “mauvais” échantillons, autrement dit ils se trouvent plus distants de l’état réel que quelques autres échantillons. Et puisque l’état estimé est la somme pondérée des échantillons, ces “mauvais” échantillons contribuent plus significativement à l’estimation de l’état réel (caché) que quelques “bons” échantillons. Cela engendre le décalage entre l’état estimé et l’état réel. Comme on a mentionné précédemment, l’origine de ce défaut est le choix de la loi d’importance correspondant au noyau de transition, sans tenant compte de l’observation.

La raison de l’assignation inappropriée des poids d’importances est la mise à jour par récurrence selon la formule (3.2) ( $\omega_t$  en fonction de  $\omega_{t-1}$  et ainsi de suite) pendant que un échantillon ayant un poids fort à l’instant  $(t = 1)$  peut être probablement assigné un poids beaucoup plus faible à l’instant  $(t = 2 \text{ ou } 3)$ . Ce phénomène est observable dans la troisième image.



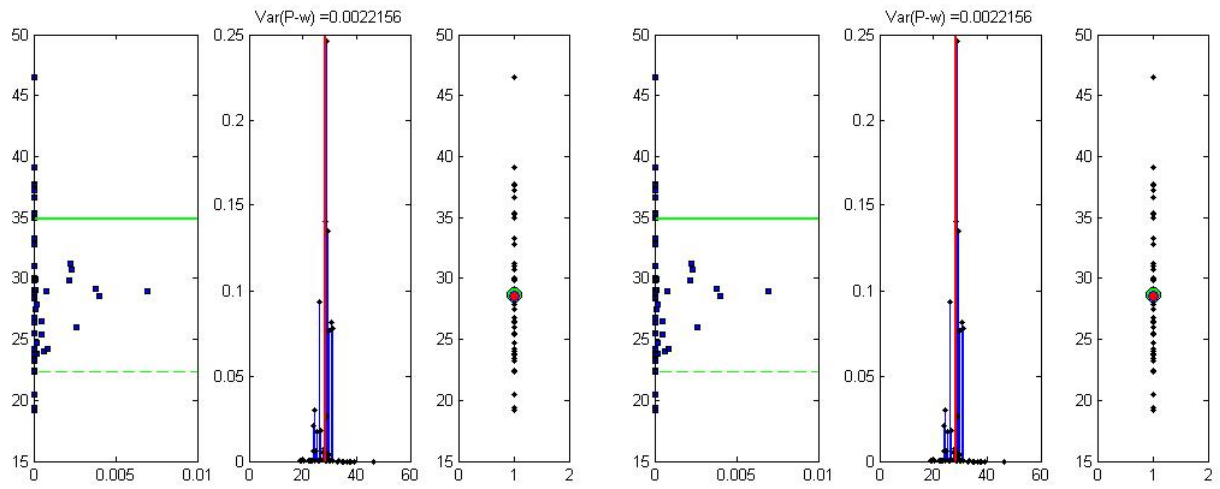
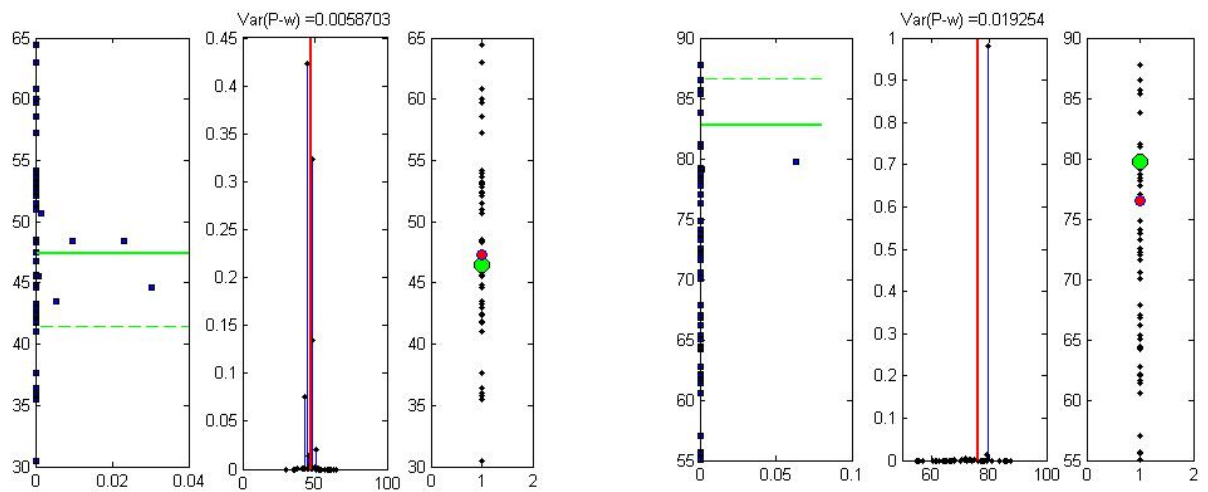
**Figure 3.2:** Évolution des particules à l'instant ( $t = 4$ )

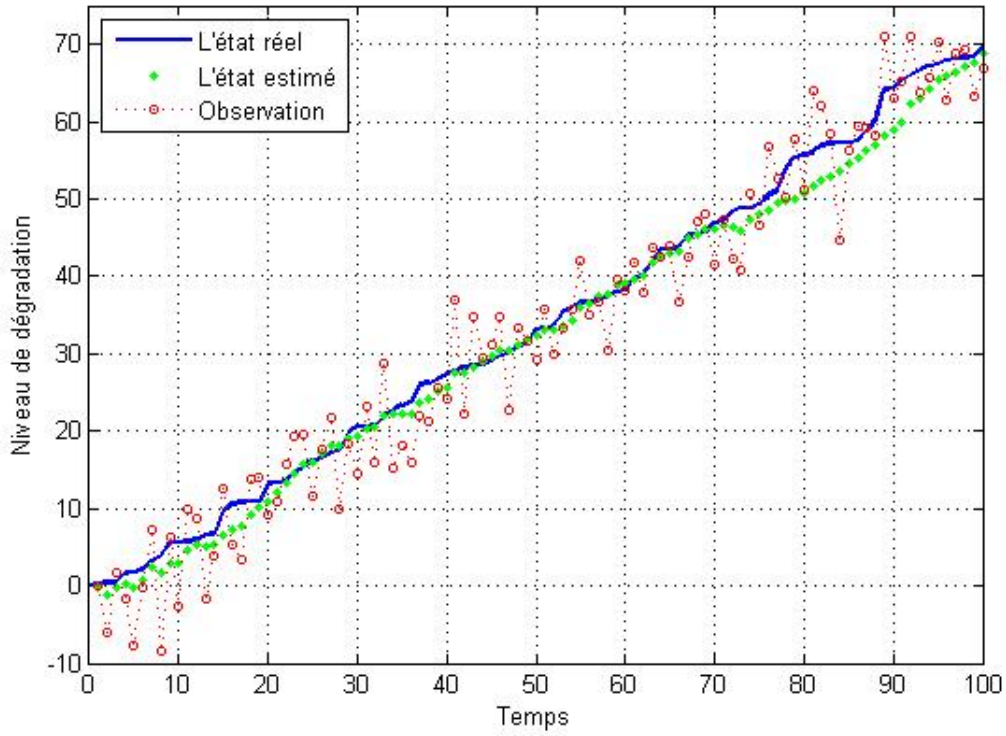
De plus, comme les poids d'importances sont normalisés à chaque itération (donc  $P\_W_t^i < 1$ ), la tendance est que le nombre d'échantillons dont le poids est significativement supérieur à zéro diminue. Après quelques itérations, tous les poids sont presque nuls sauf un. Cela entraîne le fait que l'approximation de la loi a posteriori dépend presque tout à fait du seul échantillon ayant le poids non négligeable. Une fois que cet échantillon n'approche pas l'état réel, on n'arrive pas à obtenir une estimation efficace. On appelle un tel problème sous le terme dégénérescence de poids. La figure 3.3 permet de visualiser ce problème.

On constate que la variance des poids d'importances tend à augmenter après chaque itération, la dégénérescence de poids est inévitable et cause une mauvaise approximation de la loi a posteriori et donc une estimation fragile du niveau de dégradation. De plus, ce problème implique que la plupart des ressources du centre de calcul est affecté à propager les particules peu intéressantes.

La figure 3.4 montre la trajectoire de l'état estimé obtenu après 100 premières itérations. Les valeurs caractérisant l'évolution de 6 premières particules sont enregistrées dans le tableau 3.1.

Au début, tous les échantillons  $\{x_0^i\}_{i=1}^{N_s}$  sont assignés une même valeur de poids d'importance  $\omega_0^i = \frac{1}{N_s}$ . Quand le filtre démarre, les particules sont propagées indépendamment. Après 155 itérations, on constate qu'il ne reste qu'un seul échantillon qui prend le poids approximativement à 1, alors que les poids des autres échantillons sont négligeables. On note aussi que un échantillon peut prendre le rôle le plus "important" durant quelques itérations avant de dégénérer.

(a) ( $t = 16$ ) et ( $t = 43$ )(b) ( $t = 71$ ) et ( $t = 108$ )**Figure 3.3:** Problème de dégénérescence de poids

**Figure 3.4:** Trajectoire de l'état estimé

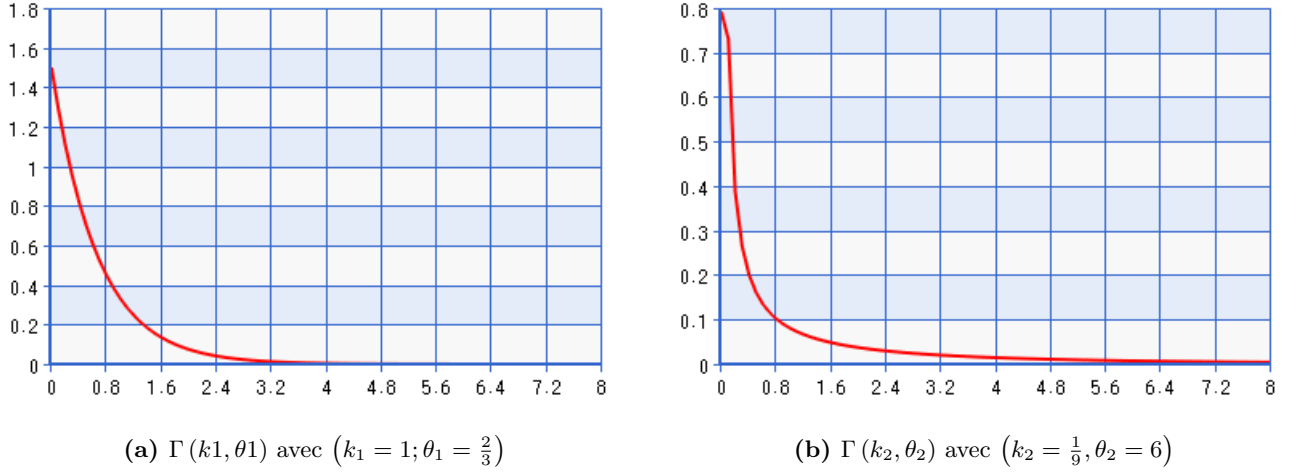
(a) Position des échantillons

$i$	$(t = 0)$	$(t = 1)$		$(t = 2)$		$(t = 3)$	
	$xP_0^i \equiv x_{-}P_1^i$	$x_{-}P_{-}update_1^i$	$xP_2^i$	$x_{-}P_{-}update_2^i$	$xP_3^i$	$x_{-}P_{-}update_3^i$	...
1	-0.5785	-0.4927	-0.4927	0.6983	0.6983	1.4469	...
2	-0.7938	1.1958	1.1958	1.2958	1.2958	1.8700	...
3	-1.2378	-1.2223	-1.2223	-0.6981	-0.6981	-0.4150	...
4	9.0948	9.3925	9.3925	9.6794	9.6794	9.8049	...
5	-6.7253	-6.5783	-6.5783	-5.9894	-5.9894	-5.5064	...
6	0.9647	1.2471	1.2471	1.6316	1.6316	1.9371	...
...	...	...	...	...	...	...	...

(b) Poids associés à chaque échantillon

$i$	$(t = 0)$	$(t = 1)$	$(t = 2)$	...	$(t = 77)$	$(t = 78)$	...	$(t = 155)$	$(t = 156)$	...
1	0.02	0.024931	0.021084	...	0.027794	0.025749	...	2.26E-10	2.07E-10	...
2	0.02	0.024488	0.017501	...	9.56E-40	6.78E-40	...	3.06E-40	3.09E-40	...
3	0.02	0.024253	0.028749	...	1.19E-07	1.42E-07	...	1.75E-09	1.47E-09	...
4	0.02	0.004440	6.64E-05	...	3.47E-44	3.97E-44	...	3.60E-71	1.13E-71	...
5	0.02	0.010324	0.021677	...	0.007096	0.007282	...	2.71E-63	6.20E-64	...
6	0.02	0.024431	0.015785	...	0.000803	0.000994	...	0.998909	0.999447	...
...	...	...	...	...	...	...	...	...	...	...

**Table 3.1:** Évolutions de 6 premières particules



**Figure 3.5:** Densité de probabilité de la distribution Gamma

### 3.5 Distribution des états estimés

Dans cette section, on étudie l'effet du SIS filtre particulaire sur deux processus Gamma de l'incrément différents:  $\Gamma(k_1, \theta_1)$  avec  $(k_1 = 1; \theta_1 = \frac{2}{3})$  et  $\Gamma(k_2, \theta_2)$  avec  $(k_2 = \frac{1}{9}, \theta_2 = 6)$ . On peut observer la densité de probabilité de ces deux distributions Gamma dans la figure 3.5:

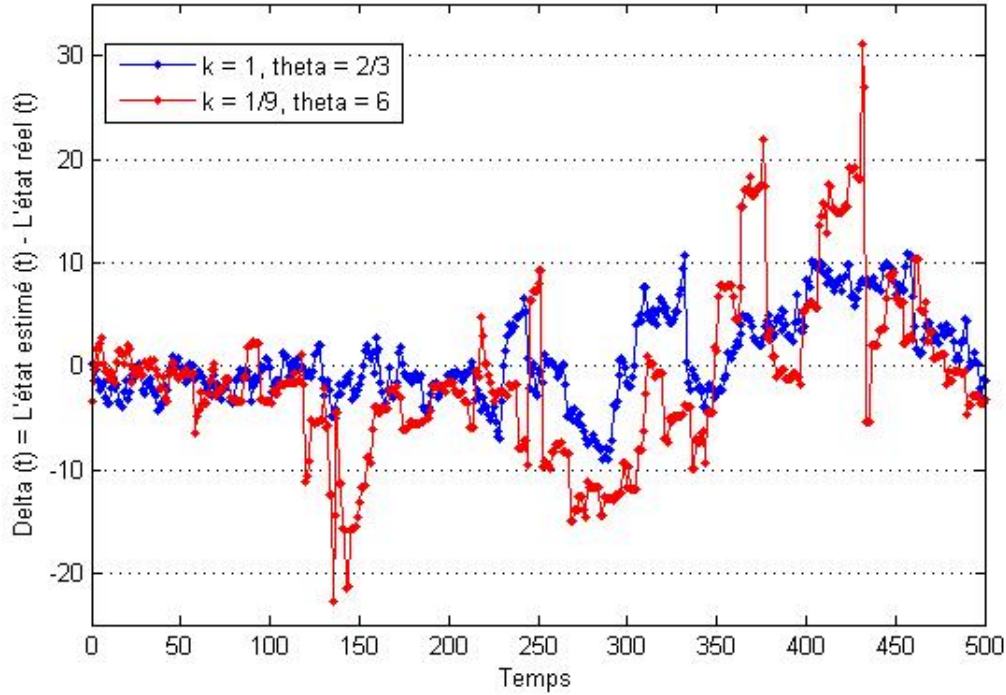
Ces deux distributions Gamma ont le même moyenne alors que la variance de  $\Gamma(k_2, \theta_2)$  est 9 fois plus grande que celle de  $\Gamma(k_1, \theta_1)$ :

$$\begin{cases} \text{moyenne} &= k_1 \times \theta_1 = k_2 \times \theta_2 = \frac{2}{3} \\ \text{variance1} &= k_1 \times \theta_1^2 = 1 \times \left(\frac{2}{3}\right)^2 = \frac{4}{9} \\ \text{variance2} &= k_2 \times \theta_2^2 = \frac{1}{9} \times (6)^2 = 4 \end{cases}$$

La figure 3.6 montre l'erreur de l'estimation  $(d_t = (x_{MMSE})_t - x_t)$  à chaque l'instant  $(t)$  dans les deux cas  $\Gamma(k_1, \theta_1)$  et  $\Gamma(k_2, \theta_2)$  avec un bruit Gaussien de moyenne nulle, l'écart-type  $\sigma_\varepsilon = 5$ . On trouve que avec une même valeur de la variance ( $\sigma_\varepsilon^2$ ) du bruit de mesure, si l'incrément du processus Gamma est plus incertain, la performance du filtre particulaire SIS décroît fortement. La raison est que quand on a un plus grand degré de variabilité sur la trajectoire de l'état de dégradation réel  $(x_t)$ , les valeurs  $\{x\_update_t^i\}_{i=1}^{N_s}$  et donc,  $\{y\_update_t^i\}_{i=1}^{N_s}$  se dispersent davantage. Par conséquent, les poids d'importances associés à ces échantillons sont gravement écartés. Cela conduit à une accélération du problème de dégénérescence de poids.

Pour visualiser la vitesse de dégénérescence de poids, on utilise l'estimateur Maximum a Posteriori  $(x_{MAP})$ :

$$(x_{MAP})_t = x_t^i \mid W_t^i = \max \{W_t^i\}_{i=1}^{N_s}$$



**Figure 3.6:** Erreur de l'estimation

Cet estimateur désigne à chaque instant ( $t$ ) l'échantillon ( $x_t^i$ ) ayant le poids le plus fort, autrement dit l'échantillon le plus "important" au sens de la contribution à l'estimation de l'état réel. Lorsqu'il ne reste qu'un seul échantillon intéressant, ( $x_{MAP}$ ) et ( $x_{MMSE}$ ) est presque identique. La figure 3.7 illustre cette idée.

On peut trouver clairement que dans le cas d'un processus Gamma avec la variance de l'incrément plus élevée, les poids dégénèrent plus rapide. Cela diminue considérablement la qualité du filtre particulaire SIS.

Chaque fois on fait la simulation avec la même donnée de mesure, on obtient une trajectoire de l'état estimé tout à fait différente. Donc, pour étudier la distribution des états estimés à tous les instants, on répète par exemple 300 fois la procédure de filtrage. La figure 3.8 montrent la distribution des états estimés correspondante à certains instants donnés pour les deux incréments.



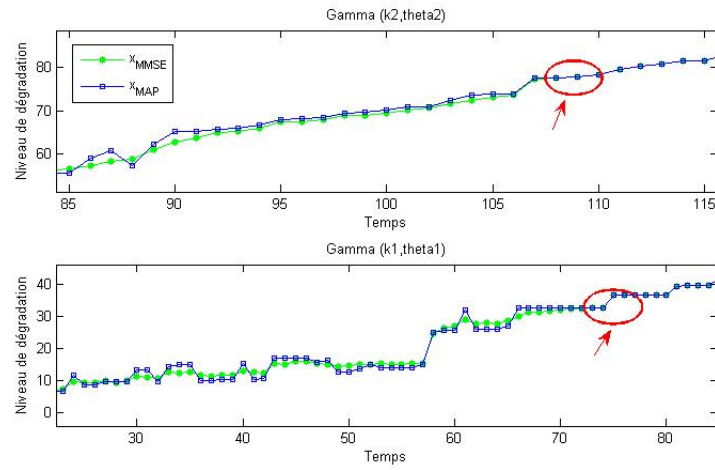


Figure 3.7: Vitesse de dégénérescence de poids

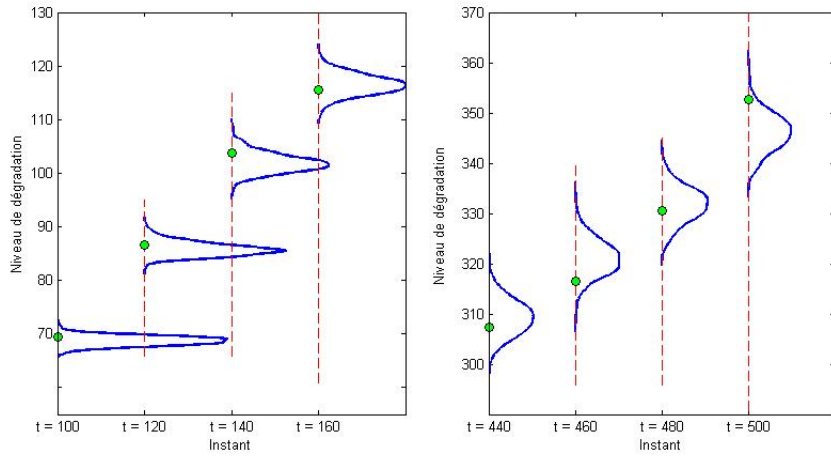
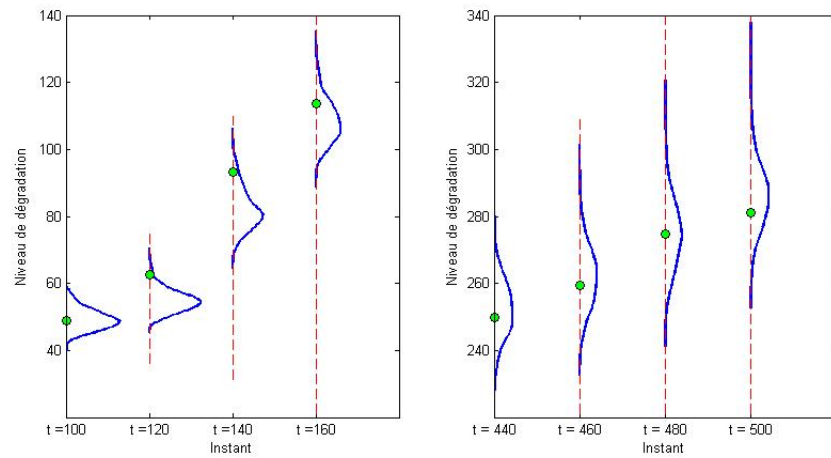
(a)  $\Gamma(k_1, \theta_1)$  avec  $(k_1 = 1, \theta_1 = \frac{2}{3})$ (b)  $\Gamma(k_2, \theta_2)$  avec  $(k_2 = \frac{1}{9}, \theta_2 = 6)$ 

Figure 3.8: Distribution des états estimés

## 4 Filtre particulaire SISR et l'estimation de la RUL

Dans le quatrième chapitre, on présente d'abord le filtre particulaire SISR (*Sequential Importance Sampling and Resampling* en anglais) développé basant sur le SIS filtre particulaire. Ensuite, on parle de l'estimation de la RUL et puis on analyse des résultats de simulation et donne des remarques importantes. La loi du temps d'atteinte est aussi abordée dans ce chapitre.

### 4.1 Filtre Bootstrap

Comme on a vu dans le chapitre 3, un grave défaut du filtre particulaire SIS est le dégénérescence de poids qui mène à une représentation inadéquate de la loi a posteriori. Alors, il est indispensable de penser à une technique qui vise à ré-initialiser le filtre régulièrement pour prévenir ce phénomène. L'idée est d'introduire après l'estimation de l'état ( $x_t$ ), une étape supplémentaire dite *ré-échantillonnage* (ou *redistribution*). Dans cette étape, les échantillons ayant un poids faible sont éliminés tandis que ceux dont le poids est fort sont dupliqués et donc auront plus de chance d'approcher la loi a posteriori à l'instant suivant. En général, la redistribution peut être considérée comme une autre étape d'échantillonnage d'importance dans laquelle la distribution discrète pondérée  $\{x_t^i, W_t^i\}_{i=1}^{N_s}$  est approchée par un ensemble des échantillons non pondérés (appelés les descendants)  $\{x_t^{i*}\}_{i*=1}^{N_s}$ . De cette manière l'écart entre les poids d'importances à chaque instant ( $t$ ) est réduit et les particules peuvent être mieux profités. Un algorithme de SIS filtre particulaire avec la redistribution à chaque instants ( $t$ ) est introduit dans [10] sous le nom *filtre bootstrap*. Cet algorithme que l'on décrit dans [Algorithme 4.1](#) prend aussi le noyau de transition comme la loi d'importance.

Note que après l'étape de redistribution à l'instant  $(t - 1)$  les échantillons ont des poids uniformes, donc il n'est pas nécessaire de sauvegarder ces poids pour l'instant ( $t$ ). Alors, la formule qui sert à pondérer les échantillons à l'instant ( $t$ ) est très simple  $\omega_t^i = p(y_t | x_t)$  (pas de multiplication récursive). L'utilisation du noyau de transition comme la loi d'importance est très populaire car quand le problème change (l'équation d'état change), on ne doit que corriger l'expression du noyau de transition et de la fonction de vraisemblance dans le programme.

Selon [6], [5] et [13], l'étape de redistribution engendre extra Monte Carlo variance de l'estimateur. Donc, on est recommandé d'effectuer préalablement l'estimation de l'état de dégradation à chaque instant à l'aide des échantillons pondérés.

Le mécanisme de redistribution est illustrée dans la figure [4.1](#).

**Algorithme 4.1** Filtre bootstrap

À l'instant ( $t = 0$ )

**Initialisation:**  $x_0^i \sim p(x_0)$ ,  $i = 1 : N_s$

À partir de l'instant ( $t \geq 1$ )

**Échantillonnage d'importance** ( $i = 1 : N_s$ )

$$\begin{cases} x_t^i \sim p(x_t^i | x_{t-1}^i) \\ \omega_t^i = p(y_t | x_t^i) \end{cases}$$

**Normaliser**

$$W_t^i = \frac{\omega_t^i}{\sum_{i=1}^{N_s} \omega_t^i}, i = 1 : N_s$$

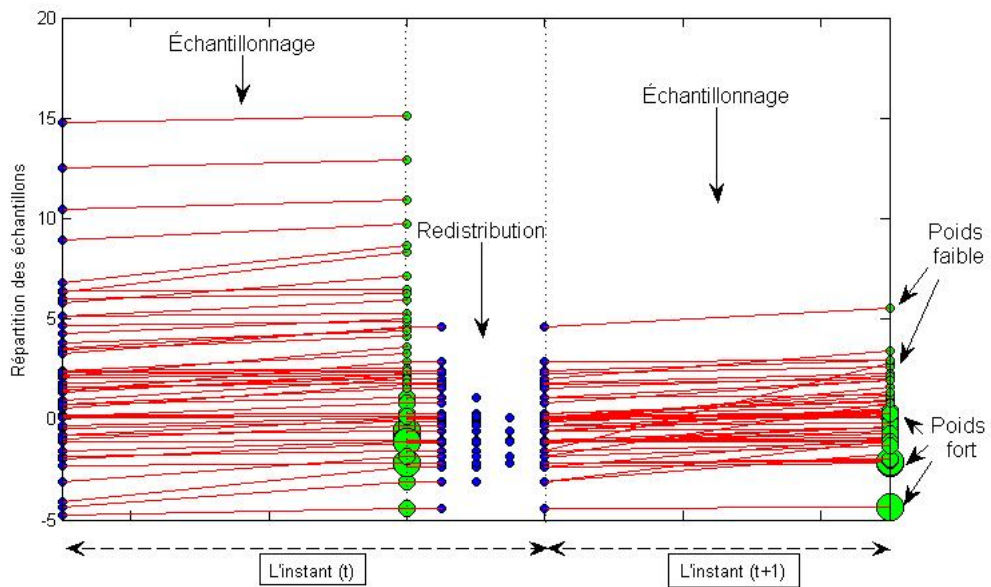
**Estimer le niveau de dégradation**

$$x_{MMSE} = \sum_{i=1}^{N_s} W_t^i \times x_t^i$$

**Redistribution** ( $i = 1 : N_s$ )

$$x_t^{i*} = \text{redistribution}(x_t^i, W_t^i)$$

$$(x_t^i = x_t^{i*})$$



**Figure 4.1:** Mécanisme de redistribution

En général, le filtre bootstrap ne donne plus une approximation efficace de la loi a posteriori quand le temps  $T$  ( $t = 1 : T$ ) est très grand à cause des redistributions successives. En effet, le mécanisme de redistribution réduit la diversité des particules, c'est-à-dire quand le temps passe, l'ensemble des particules contient seulement quelques unes "distinguées". La plupart entre eux sont des descendants d'un même ancêtre. C'est évidemment la conséquence d'effectuer l'échantillonnage d'une distribution discrète au lieu d'une distribution continue. Ce phénomène qu'on appelle *dégénérescence des positions* (*sample impoverishment* en anglais) a le même effet que la dégénérescence de poids mentionnée dans le chapitre 3. D'après [2], si le bruit du processus (l'incrément  $\Gamma(k, \theta)$  du processus Gamma) a une variance suffisamment grande, les échantillons non pondérés  $\{x_t^{i*}\}_{i=1}^{N_s}$  peuvent être propagés, dans l'étape de prédiction à l'instant suivant, de sorte qu'ils peuvent maintenir une diversité adéquate au sein de l'ensemble d'échantillons. Au contraire, si  $\{x_t^{i*}\}_{i=1}^{N_s}$  ne sont pas suffisamment contrebalancés par le bruit du processus, la qualité de l'approximation de la loi a posteriori est dégradée. Pour résoudre ce problème, des mesures sont proposées et récapitulées dans [10], [1], [11], [16] et [9].

## 4.2 Filtre SISR avec redistribution d'adaptation

Pour obtenir une estimation du niveau de dégradation plus exacte à chaque instant ( $t$ ), il est possible de modifier l'Algorithme 4.1 sur les points suivants:

- La redistribution multinomiale est la plus simple technique par l'utilisation de laquelle  $N_s$  nouveaux échantillons (les descendants) sont sélectionnées à partir de  $N_s$  anciens échantillons avec la probabilité égale à leurs poids normalisés:  $p(x_t^{i*} = x_t^i) = W_t^i$ . En choisissant d'autres techniques de redistribution on peut réduire la variance introduite par l'étape de redistribution. L'une est la technique de redistribution systématique proposée par Kitagawa [14] et l'autre est la redistribution des résidus introduite par Liu et Chen [18]. Une comparaison sur la qualité et la complexité de ces trois techniques est effectué dans [7] et [12]. Dans ce rapport les résultats de la simulation sont obtenus avec l'utilisation de la technique de redistribution de Kitagawa dont le code est donné dans [3].
- On peut améliorer la performance du filtre particulaire en augmentant le nombre d'échantillons. Néanmoins, cela ne paraît pas comme une solution pratique. Il est préférable de choisir une meilleure loi d'importance. [4] donne un bon résumé sur cette alternative. Une loi d'importance optimale obtenu dans un cas particulier est représenté dans [8] et [1].
- On sait maintenant que l'on ne doit pas redistribuer à chaque instant mais de le faire de telle manière que la redistribution puisse prévenir la dégénérescence de poids et ne cause pas une dégénérescence des positions trop grave. Autrement dit, les échantillons sont redistribués seulement quand un grand déséquilibre de leurs poids d'importances est constaté. Alors, on souhaite de disposer une critère qui nous permet de déterminer s'il est nécessaire de redistribuer les échantillons ou s'il faut les conserver. [15], [17] quantifient le

niveau de déséquilibre des poids d'importances en utilisant un coefficient appelé nombre d'échantillons effectives:

$$N_{eff} \approx \frac{1}{\sum_{i=1}^{N_s} (W_t^i)^2} \in [1, N_s]$$

Lorsque  $N_{eff}$  est proche de  $N_s$ , les échantillons contribuent de façon égale à l'approximation de la loi a posteriori tandis que une valeur proche de 1 indique une dégénérescence de poids sévère. La redistribution est faite seulement quand  $N_{eff}$  est inférieur à un seuil  $N_{thresh}$  prédéterminé, typiquement  $N_{thresh} = \frac{N_s}{2}$ .

Tenir compte de la premier et de la troisième modification mentionnées au dessus, un algorithme du filtre SISR avec redistribution d'adaptation est présenté dans l'[Algorithme 4.2](#).

Comme on a dit au dessus, la redistribution donne effectivement des “bruits” additionnels. Cependant, redistribuer des échantillons nous aide à éviter l'accumulation des erreurs avec le temps et donc rendre l'approximation de la loi a posteriori plus stable [9] , [4]. C'est la raison pour laquelle la redistribution est largement utilisée. En pratique, implémenter correctement l'étape de redistribution peut améliorer vivement la performance du filtre particulaire. Pour illustrer cette idée, on définit tout d'abord un indicateur qui désigne l'erreur quadratique moyenne de l'estimation à chaque instant ( $t$ ):

$$RMSE_t = \sqrt{\frac{1}{300} \times \sum_{k=1}^{300} \left( (x_{MMSE})_t^k - x_t \right)^2} \quad (4.1)$$

avec l'ensemble  $\{(x_{MMSE})_t^k\}_{k=1}^{300}$  contient 300 valeurs estimées du niveau de dégradation réel ( $x_t$ ) à chaque instant ( $t = 1 : 500$ ), obtenues après  $k = 300$  itérations.

En utilisant les algorithmes [Algorithme 2.1](#) et [Algorithme 4.2](#), on obtient pour chacun les valeurs  $RMSE_t$  ainsi que les variance  $var \left( \{(x_{MMSE})_t^k\}_{k=1}^{300} \right)$  à chaque instant ( $t = 1 : 500$ ) correspondantes. Les avantages de l'utilisation d'un filtre SISR avec redistribution d'adaptation au lieu d'un SIS filtre particulaire est indiqué dans la figure [4.2](#).

La trajectoire de  $RMSE_t$  dans la figure [4.2](#) n'est pas monotone, ce qui nous met en difficulté à donner une conclusion certaine sur l'exactitude de l'estimation du niveau de dégradation. Plus précisément, il est probable qu'un grand nombre de mesures effectuées n'assurent pas une amélioration de la qualité de l'estimation. En effet, l'utilisation beaucoup de données bruitées peuvent diminuer la performance du filtre particulaire. Alors, on pense à utiliser de manière plus intelligemment un moindre données pour les profiter mais sans avoir perturber le filtre particulaire.

---

**Algorithme 4.2** Filtre SISR avec redistribution d'adaptation.

---

À l'instant ( $t = 0$ )

$$\textbf{Initialisation:} \begin{cases} x_0^i & \sim p(x_0) \\ \omega_0^i & = 1/N_s \end{cases}, i = 1 : N_s$$


---

À partir de l'instant ( $t \geq 1$ )

**Échantillonnage d'importance**

$$\begin{cases} x_t^i & \sim p(x_t^i | x_{t-1}^i) \\ \omega_t^i & = \omega_t^i \times p(y_t | x_t^i) \end{cases}$$

**Normaliser**

$$W_t^i = \frac{\omega_t^i}{\sum_{i=1}^{N_s} \omega_t^i}, i = 1 : N_s$$

**Estimer le niveau de dégradation**

$$x_{MMSE} = \sum_{i=1}^{N_s} W_t^i \times x_t^i$$

**Calculer le nombre d'échantillons effectives**

$$N_{eff} = \frac{1}{\sum_{i=1}^{N_s} (W_t^i)^2}$$

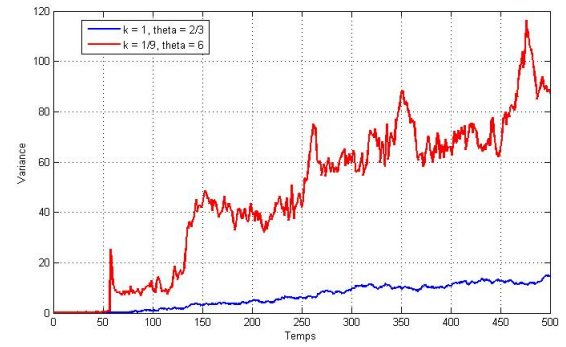
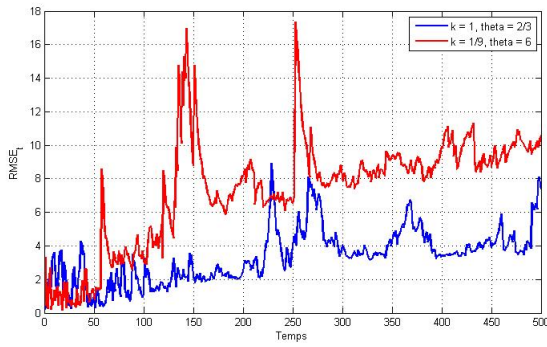
Si  $N_{eff} \leq N_{thresh}$  alors

**Redistribution** ( $i = 1 : N_s$ )

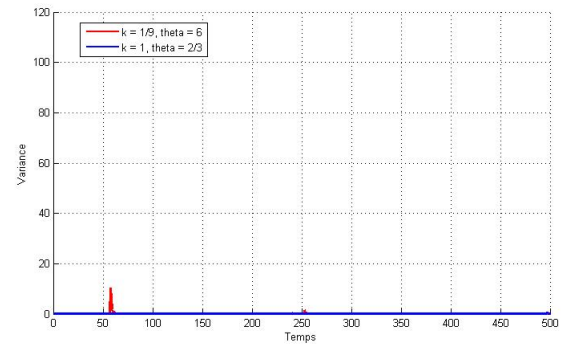
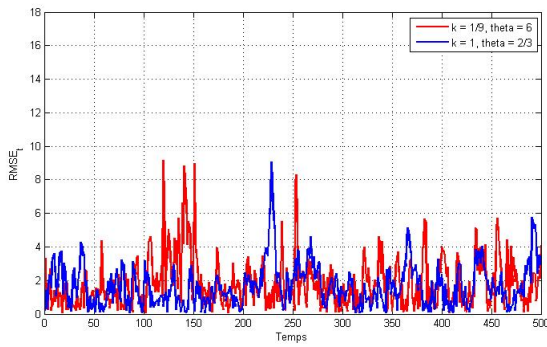
$$x_t^{i*} = \text{redistribution}(x_t^i, W_t^i)$$

$$\begin{cases} x_t^i = x_t^{i*} \\ \omega_t^i = 1/N_s \end{cases}$$


---



(a) SIS filtre particulaire



(b) Filtre SISR avec redistribution adaptative

**Figure 4.2:**  $RMSE_t$  et  $var\left(\left\{(x_{MMSE})_t^k\right\}_{k=1}^{300}\right)$  à chaque instant ( $t = 1 : 500$ )

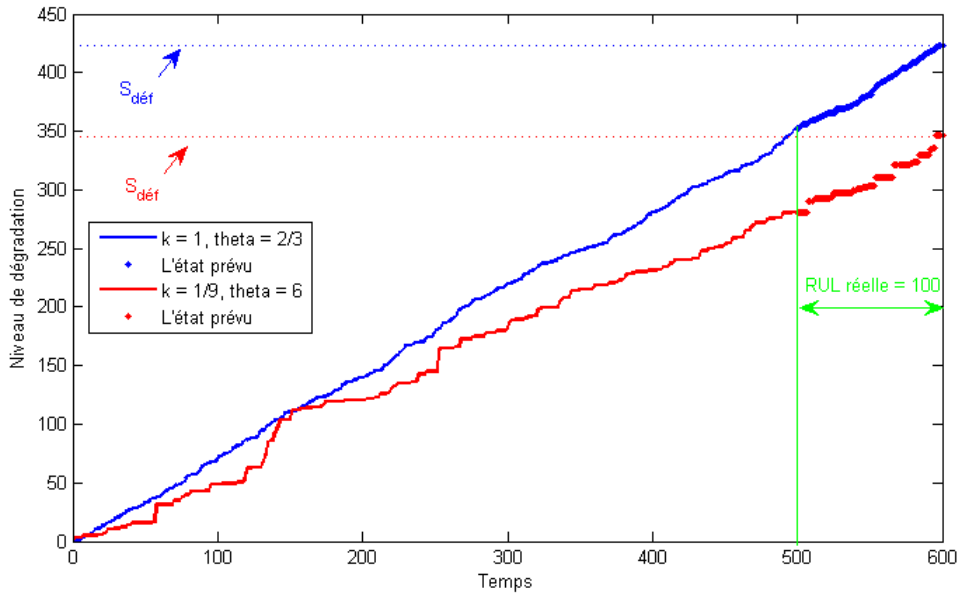
### 4.3 Estimation de la RUL

Rappelons que l'instant courant est ( $t_n = 500$ ). Pour calculer la vie résiduelle, on propage le processus Gamma avec le niveau de dégradation initial  $(x_{MMSE})_{t=500}^k, k = 1 : 300$  jusqu'au moment ( $t$ ) où  $(x_{MMSE})_t^k$  atteint un seuil prédéterminé appelé seuil de défaillance. Alors, si la date de défaillance prévue est notée  $T_{déf}$ , la RUL correspondant est:

$$RUL = T_{déf} - t_n = T_{déf} - 500$$

En répétant cette procédure  $k = 300$  fois, on obtient la distribution de la RUL.

Avant de faire la simulation, il est indispensable de déterminer le seuil de défaillance  $S_{déf}$ . Suppose que la date de défaillance réel est  $T_{déf} = 600$ , donc la RUL réelle est 100. La figure 4.3 montre que le seuil de défaillance correspondant de deux processus Gamma de l'incrément  $\Gamma(k_1 = 1, \theta_1 = 2/3)$  et  $\Gamma(k_2 = 1/9, \theta_2 = 6)$  est 424.5 et 347.4 respectivement.



**Figure 4.3:** Seuil de défaillance

L'évolution du niveau de dégradation est montrée dans la figure 4.4. La distribution de la RUL estimée est dessinée en couleur verte.



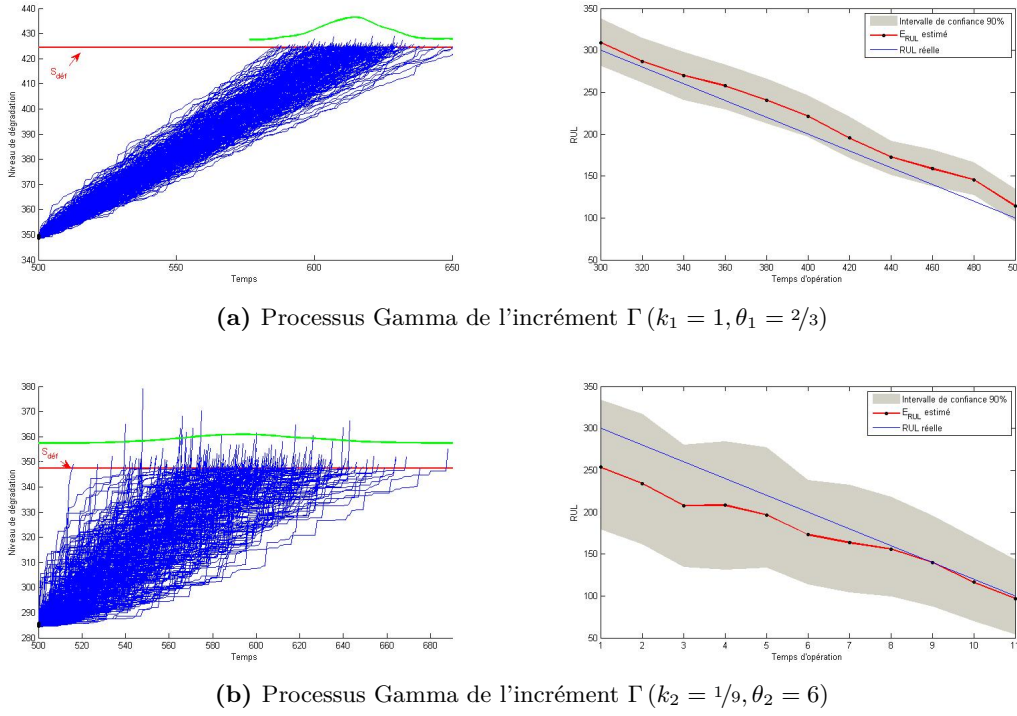


Figure 4.4: Estimation de la RUL

## 4.4 Résultats de simulation

Dans cette section, on cherche à comprendre l'impact de deux paramètres: nombre d'échantillons ( $N_s$ ) et l'écart-type du bruit de mesures ( $\sigma_\varepsilon$ ) sur la performance du filtre particulaire.

### 4.4.1 Quand le nombre d'échantillons ( $N_s$ ) augmente

D'abord, on introduit un nouveau indicateur  $RMSE_{globale}$  pour avoir une vue globale sur l'importance des paramètres comme le nombre d'échantillons  $N_s$  et l'écart-type du bruit de mesures  $\sigma_\varepsilon$ :

$$RMSE_{globale}^k = \sqrt{\frac{1}{T} \sum_{t=1}^T \left( (x_{MMSE})_t^k - x_t \right)^2}$$

Car à chaque simulation ( $k$ ), le filtre particulaire produit une différente trajectoire du niveau de dégradation estimé  $\{x_{MMSE}\}_{t=1}^T$ ,  $T = 500$ , donc les valeurs  $RMSE_{globale}$  affichées dans les tableaux suivants est la moyenne des valeurs  $RMSE_{globale}^k$  obtenues après ( $k = 300$ ) simulations. Plus on obtient un  $RMSE_{globale}$  petit, plus il est probable que l'on peut avoir une estimation précise.

Calculée selon la formule (4.1) avec ( $t = 500$ ),  $RMSE_{t_n=500}$  désigne l'erreur de l'estimation du niveau de dégradation à l'instant courant.

Un troisième indicateur utilisé est  $RMSE_{RUL}$  qui indique l'erreur de l'estimation de la RUL par

rapport à la RUL réelle (=100):

$$RMSE_{RUL} = \sqrt{\frac{1}{300} \times \sum_{k=1}^{300} ((RUL)_k - 100)^2}$$

Les résultats de simulation sont présentés dans les tableaux suivantes:

$N_s$	$\Gamma(k_1 = 1, \theta_1 = 2/3), \sigma_\varepsilon = 5$					
	SIS filtre particulaire			Filtre SISR avec redistribution d'adaptation		
	$RMSE_{globale}$	$RMSE_{t_n=500}$	$RMSE_{RUL}$	$RMSE_{globale}$	$RMSE_{t_n=500}$	$RMSE_{RUL}$
50	4.832	7.963	37.771	2.074	3.705	34.473
100	4.390	8.114	36.933	2.047	3.643	34.430
300	4.030	6.798	35.375	2.033	3.494	35.741
500	3.890	7.604	38.099	2.031	3.490	35.005
1000	3.671	7.394	37.463	2.027	3.448	33.568
2000	3.539	7.232	36.946	2.028	3.473	35.628
4000	3.354	7.357	36.725	2.025	3.444	33.866
7000	3.240	7.140	36.827	2.025	3.440	34.400
10000	3.179	6.623	35.948	2.025	3.429	33.925

$N_s$	$\Gamma(k_2 = 1/9, \theta_2 = 6), \sigma_\varepsilon = 5$					
	SIS filtre particulaire			Filtre SISR avec redistribution d'adaptation		
	$RMSE_{globale}$	$RMSE_{t_n=500}$	$RMSE_{RUL}$	$RMSE_{globale}$	$RMSE_{t_n=500}$	$RMSE_{RUL}$
50	11.237	19.822	44.955	2.498	4.608	41.677
100	9.980	14.886	44.364	2.392	4.144	40.025
300	8.801	11.447	42.990	2.311	4.117	40.475
500	8.361	11.570	42.283	2.298	4.116	42.015
1000	7.955	10.604	43.700	2.282	4.086	41.021
2000	7.467	10.499	39.610	2.278	4.117	38.559
4000	7.171	9.137	45.841	2.275	4.136	42.825
7000	6.846	9.406	43.610	2.274	4.126	40.738
10000	6.680	8.471	42.690	2.274	4.127	38.779

Avec le SIS filtre particulaire, on trouve que plus ( $N_s$ ) est grand, plus  $RMSE_{globale}$  décroît. Néanmoins, pour la raison spécifiée dans la section 4.2, une valeur  $RMSE_{globale}$  petite n'assure pas une erreur de l'estimation petite. Par ailleurs, avec un grand ( $N_s$ ), le filtre particulaire donne des valeurs  $RMSE_{globale}$  et  $RMSE_{t_n=500}$  plus robuste.

En utilisant un SIS filtre particulaire, le processus Gamma dont l'incrément  $\Gamma(k_2 = 1/9, \theta_2 = 6)$  ayant une variance plus grande offre des erreurs  $RMSE_{globale}$ ,  $RMSE_{t_n=500}$  et dans une certaine mesure,

$RMSE_{RUL}$  plus grandes. C'est parce que un tel processus subit une dégénérescence de poids plus sévère. Avec l'algorithme du filtre SISR avec redistribution d'adaptation, la différence entre les deux processus Gamma au vu des indicateurs  $RMSE_{globale}$  et  $RMSE_{t_n=500}$  est réduite. Pourtant,  $RMSE_{RUL}$  reste encore différé significativement car l'estimation de la RUL est fortement influencé par la variance de l'incrément (voir figure 4.4). C'est aussi la raison pour laquelle un bon diagnostic n'assure pas un bon résultat de pronostic, notamment dans le cas du processus Gamma avec un incrément très varié ( $\Gamma(k_2 = 1/9, \theta_2 = 6)$ ).

Par rapport au SIS filtre particulaire, l'implémentation de l'étape de redistribution d'adaptation apporte une erreur  $RMSE_{globale}$  beaucoup plus petite. De plus, comme l'erreur de l'estimation du niveau de dégradation courante (le diagnostic) est plus petite, on obtient une légère amélioration du résultat de pronostic. Enfin, on constate que l'augmentation de  $N_s$  n'apporte pas une amélioration importante pour les résultats  $RMSE_{globale}$  et  $RMSE_{t_n=500}$ .

#### 4.4.2 Quand l'écart-type du bruit de mesures ( $\sigma_\varepsilon$ ) varie

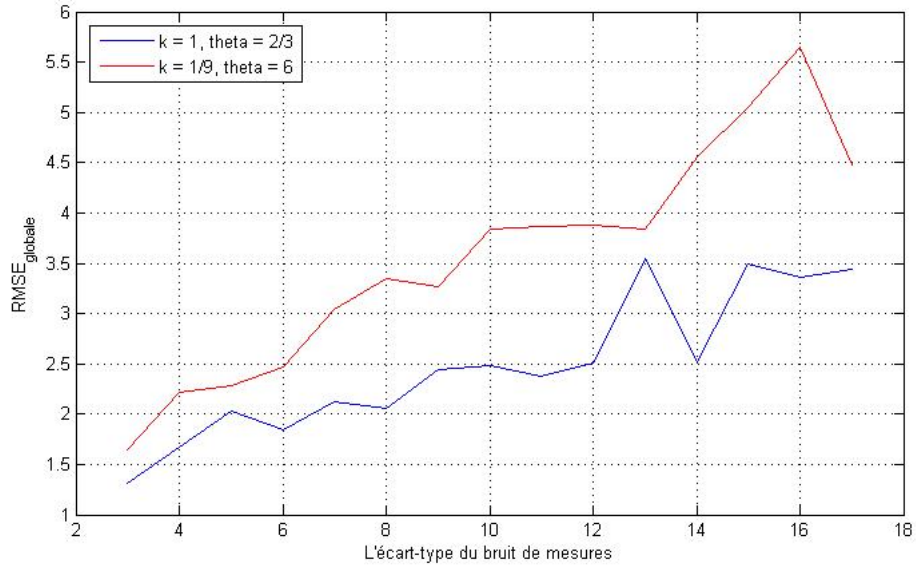
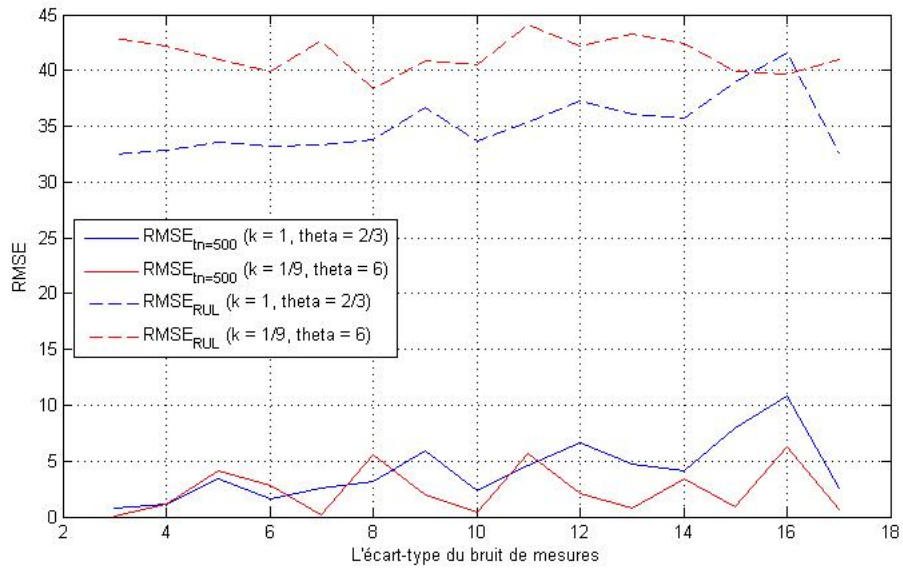
$\sigma_\varepsilon$	Filtre SISR avec redistribution d'adaptation, $N_s = 1000$					
	$\Gamma(k_1 = 1, \theta_1 = 2/3)$			$\Gamma(k_2 = 1/9, \theta_2 = 6)$		
	$RMSE_{globale}$	$RMSE_{t_n=500}$	$RMSE_{RUL}$	$RMSE_{globale}$	$RMSE_{t_n=500}$	$RMSE_{RUL}$
3	1.315	0.731	32.448	1.651	0.088	42.868
4	1.672	1.096	32.862	2.222	1.089	42.196
5	2.027	3.448	33.568	2.282	4.086	41.021
6	1.838	1.622	33.164	2.474	2.778	39.873
7	2.130	2.543	33.274	3.047	0.139	42.628
8	2.052	3.207	33.810	3.353	5.528	38.409
9	2.444	5.939	36.720	3.268	2.024	40.908
10	2.479	2.305	33.661	3.840	0.395	40.508
11	2.371	4.618	35.367	3.862	5.658	44.118
12	2.512	6.597	37.228	3.873	2.052	42.148
13	3.546	4.706	36.088	3.839	0.720	43.288
14	2.526	4.077	35.697	4.562	3.367	42.462
15	3.489	7.957	38.998	5.043	0.923	39.909
16	3.354	10.794	41.591	5.645	6.238	39.731
17	3.445	2.592	32.583	4.482	0.628	40.944

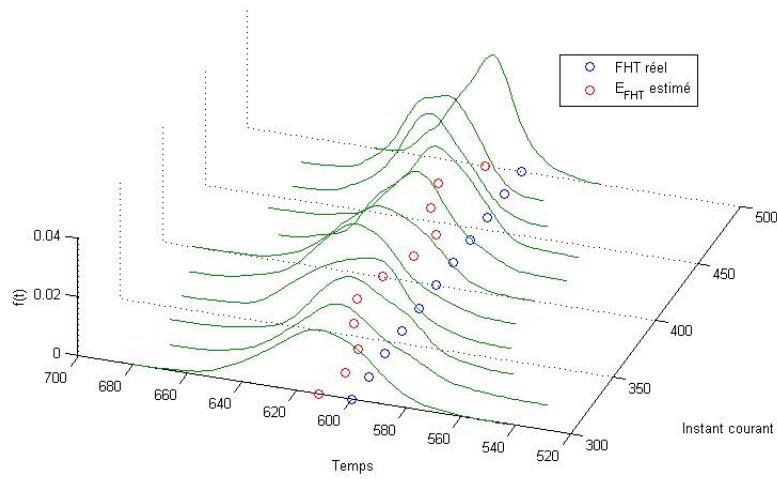
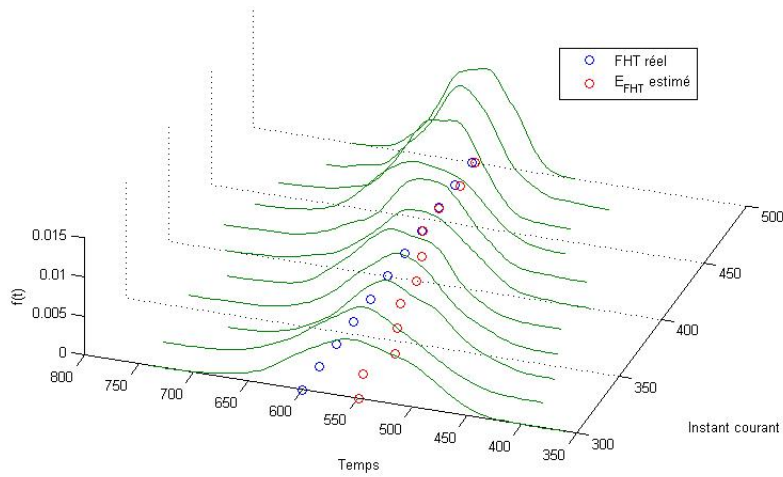
Comme on a dit dans la section 3.2, choisir la loi d'importance sous la forme du noyau de transition emmène à un filtre particulièrement sensible avec des observations bruitées. Par conséquent, la qualité de l'estimation est instable (cette idée est illustrée dans la figure 4.5a où les graphes  $RMSE_{globale}$  sont non linéaires, non monotones).

D'autre part, on constate que même si  $\sigma_\varepsilon$  est grand, le filtre particulaire peut apporter un bon diagnostic si la variance de la distribution  $\Gamma(k, \theta)$  est suffisamment petite. Alors que si cette variance est grande, le diagnostic est arbitrairement de mauvaise qualité même si  $\sigma_\varepsilon$  est petit. Par exemple, avec l'incrément  $\Gamma(k_1 = 1, \theta_1 = 2/3)$ ,  $RMSE_{t_n=500} = 4.077$  quand  $\sigma_\varepsilon = 14$  tandis que  $RMSE_{t_n=500} = 4.086$  quand  $\sigma_\varepsilon = 8$  avec l'incrément  $\Gamma(k_2 = 1/9, \theta_2 = 6)$ .

#### 4.4.3 Loi du temps d'atteinte

Le temps d'atteinte (*first hitting time* - FHT en anglais) est l'instant où le niveau de dégradation dépasse le seuil de défaillance. La densité de probabilité de FHT en fonction du temps est montrée dans la figure 4.6. On trouve que cette densité de probabilité devient plus étroite quand l'instant courant approche la date de défaillance  $T_{déf} = 600$ . Cela signifie que l'estimation de FHT est plus exacte quand le temps passe.

(a) Variation de  $RMSE_{globale}$  lorsque  $(\sigma_\epsilon)$  change(b) Variation de  $RMSE_{tn=500}$  et  $RMSE_{RUL}$  lorsque  $(\sigma_\epsilon)$  change**Figure 4.5:** Impact de l'écart-type du bruit de mesures  $(\sigma_\epsilon)$

(a) Incrément  $\Gamma (k_1 = 1, \theta_1 = 2/3)$ (b) Incrément  $\Gamma (k_2 = 1/9, \theta_2 = 6)$ **Figure 4.6:** Densité de probabilité de la loi du temps d'atteinte

## 5 Conclusion

L'objectif de ce stage durant six mois est d'étudier une approche stochastique dite filtre particulaire pour résoudre le problème de diagnostic et pronostic de défaillance. Le processus Gamma est choisi pour modéliser la dégradation. Après avoir chercher à comprendre comment un filtre particulaire est construit, on met en application des algorithmes fondamentales afin de réaliser l'estimation du niveau de dégradation actuel et puis calculer la durée de vie résiduelle du système d'intérêt. Dans la simulation, on étudie la performance du filtre particulaire lors d'une variation des paramètres importants. De plus, on considère aussi le comportement du filtre particulaire envers deux processus Gamma de l'incrément différents. Des remarques importantes sont retirées en analysant les résultats de simulation.

Dans le rapport, des extension sophistiquées du filtre particulaire sont également relevées et d'autres travaux sont en cours pour améliorer sa performance. Par exemple, il est souhaitable d'augmenter vivement la qualité de l'estimation en utilisant habilement les données disponibles puisque un grand nombre de valeurs de mesures incertaines peuvent perturber le filtre particulaire. En effet, les mesures peuvent être effectuées périodiquement ou suivant une loi de probabilité quelconque, c'est à dire l'instant de faire la mesure est une variable aléatoire. Cela nous aide à économiser la dépense.

Les algorithmes du filtre particulaire sont vérifiés par la simulation avec des données créées artificiellement. Il est plus convaincant s'il on a des données réelles acquis d'un vrai processus de dégradation pour examiner l'efficience de ces algorithmes.

# Bibliography

- [1] M Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *Signal Processing, IEEE Transactions on*, 50(2):174–188, 2002.
- [2] D Avitzour. Stochastic simulation bayesian approach to multitarget tracking. In *Radar, Sonar and Navigation, IEE Proceedings-*, volume 142, pages 41–44. IET, 1995.
- [3] Fabien Campillo. Filtrage particulaire et modèles de markov cachés. *Cours Master recherche*, 2006.
- [4] Olivier Cappé, Simon J Godsill, and Eric Moulines. An overview of existing methods and recent advances in sequential monte carlo. *Proceedings of the IEEE*, 95(5):899–924, 2007.
- [5] James Carpenter, Peter Clifford, and Paul Fearnhead. Improved particle filter for nonlinear problems. *IEE Proceedings-Radar, Sonar and Navigation*, 146(1):2–7, 1999.
- [6] George Casella, Juan Ferrándiz, Daniel Peña, David Rios Insua, José M Bernardo, PA García-López, A González, J Berger, AP Dawid, Thomas J Diciccio, et al. Statistical inference and monte carlo algorithms. *Test*, 5(2):249–344, 1996.
- [7] Randal Douc and Olivier Cappé. Comparison of resampling schemes for particle filtering. In *Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium on*, pages 64–69. IEEE, 2005.
- [8] Arnaud Doucet, Simon Godsill, and Christophe Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and computing*, 10(3):197–208, 2000.
- [9] Arnaud Doucet and Adam M Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of Nonlinear Filtering*, 12:656–704, 2009.
- [10] Neil J Gordon, David J Salmond, and Adrian FM Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE Proceedings F (Radar and Signal Processing)*, volume 140, pages 107–113. IET, 1993.
- [11] Fredrik Gustafsson, Fredrik Gunnarsson, Niclas Bergman, Urban Forssell, Jonas Jansson, Rickard Karlsson, and P-J Nordlund. Particle filters for positioning, navigation, and tracking. *Signal Processing, IEEE Transactions on*, 50(2):425–437, 2002.
- [12] Jeroen D Hol, Thomas B Schon, and Fredrik Gustafsson. On resampling algorithms for particle filters. In *Nonlinear Statistical Signal Processing Workshop, 2006 IEEE*, pages 79–82. IEEE, 2006.



- [13] Adam M Johansen, Ludger Evers, and N Whiteley. Monte carlo methods. *Lecture notes, Bristol: University of Bristol*, 2007.
- [14] Genshiro Kitagawa. Monte carlo filter and smoother for non-gaussian nonlinear state space models. *Journal of computational and graphical statistics*, 5(1):1–25, 1996.
- [15] Augustine Kong, Jun S Liu, and Wing Hung Wong. Sequential imputations and bayesian missing data problems. *Journal of the American statistical association*, 89(425):278–288, 1994.
- [16] François Legland. Filtrage particulaire. In *Proceedings 19eme Colloque GRETSI sur le Traitement du Signal et des Images*, volume 1, pages 1–8, 2003.
- [17] Jun S Liu and Rong Chen. Blind deconvolution via sequential imputations. *Journal of the American Statistical Association*, 90(430):567–576, 1995.
- [18] Jun S Liu and Rong Chen. Sequential monte carlo methods for dynamic systems. *Journal of the American statistical association*, 93(443):1032–1044, 1998.