

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي و البحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique



Mémoire de Master

Filière Électronique
Spécialité Instrumentation

présenté par

Boumrar amira

&

Laari safa

Validation des Signaux par Filtres Particulaires

Proposé par : Mr. Khentout Nourddine & Mr. Ykhlef Farid

Année Universitaire 2020-2021

Remerciements

A l'issue de cette fin de travail, nous remercions le Dieu tout puissant pour la volonté, la santé et la patience qu'il nous a donné durant toutes ces longues années d'études.

Le mémoire de master présenté ici est le résultat d'un travail continu pendant six mois.

Il n'aurait pas pu être élaboré sans l'aide et la collaboration, de près ou de loin, de nombreuses personnes que nous tenons ici à remercier.

Nous remercions chaleureux à nos parents, qui ont été toujours à nos côtés pour nous encourager à atteindre nos objectifs académique et personnel. Nous souhaitons qu'ils soient fiers de nous.

Nous tenons à exprimer nos remerciements à nos grandes familles, qu'ont été aussi parmi les partenaires qui nous encourager.

Nous exprimons nos sincères remerciements à Monsieur *Khentout Nourddine*, chercheur au *Centre de la Recherche Nucléaire de Birine (CRNB)* qui a initié et dirigé cette activité scientifique. Ses connaissances du sujet, ses conseils et son aide continus ont été essentiels pour les résultats de notre travail.

Nous souhaiterons exprimer nos sincères remerciements à Monsieur *Farid Yekhlef*, notre professeur d'Enseignement Supérieur à *l'université SAAD Dahleb Blida 01 (USDB)*. Nous voudrions aussi lui exprimer nos profondes reconnaissances pour la confiance qu'il nous a toujours témoignée.

Nous adressons nos remerciements à tous les professeurs qui nous ont enseigné pendant notre périple académique.

Nous avons été sensibles à l'honneur que nous fait l'ensemble des professeurs de participer aux jurys et nous les remerciant sincèrement.

Enfin, nos reconnaissances s'adressent à tous nos camarades d'Université pour les bons moments que nous avons passés en leur compagnie et nos ami(e)s.

ملخص

الهدف الأساسي من هذه الأطروحة هو التحقق من صحة الإشارات (البيانات) واكتشاف الأعطال في الأنظمة غير الخطية ذات التشويش غير الغوسي. الطريقة المختارة لهذا التطبيق هي التقدير التكراري باستخدام أخذ عينات مونت كارلو المتسلسل ، وغالبًا ما تُعرف بمرشحات الجسيمات أو مرشحات الجسيمات. يمكن تعميم هذا النهج على الإشارات المتعددة وأعطال الحدوث المتزامن

كلمات مفتاحية : التحقق من الصحة ، اكتشاف الاعطال ، تقدير الإشارة ، مرشح الجسيمات ، أخذ عينات مونت كارلو ، تقدير بايزي

Résumé

L'objectif fondamental de ce mémoire est de valider les signaux (données) et de détecter les défauts dans les systèmes non-linéaires avec un bruit *non-Gaussien*. La méthode désignée pour cette application est une estimation récursive en utilisant l'échantillonnage de Monte-Carlo séquentiel, connu souvent par filtres particuliers ou filtres à particules. Cette approche pourra être généralisée aux multi-signaux et aux défauts d'occurrence simultanée

Mots clés : Validation, Détection des Défauts, Estimation du Signal, Filtre Particulaire, Echantillonnage de Monte-Carlo, Estimation Bayésienne

Abstract

This thesis deals with a method to validate signals (data) and to detect faults in non-linear systems with *non-Gaussian* noise. The method designated for this application is a recursive estimation using sequential Monte Carlo sampling, often known as particle filters. This approach can be generalized to multi-signals and to simultaneous occurrence of faults.

Keywords : Validation, fault Detection, Signal Estimation, Particule Filter, Monte-Carlo Sampling, Bayesian Estimation

Listes des acronymes et abréviations

Notation	Signification	
	Français	Anglais
CM	Chaîne de Markov, Modèle de Markov	Markov Chain, Markov Model
CMC	Chaîne de Markov Cachée	Hidden Markov Chain
CMO	Chaîne de Markov Observable	Observable Markov Chain
DC	Densité Conditionnelle	Conditional Density
EQM	Erreur Quadratique Moyen	Mean Squared Error
FB	Filtre Bayésien	Bayesian Filter
FK	Filtre de Kalman	Kalman Filter
FKE	Filtre de Kalman Etendu	Extended Kalman Filter
FKNP	Filtre de Kalman Non-Parfumé	Unscented Kalman Filter
FP	Filtres Particulaire ou Filtre à Particules	Particle Filter
<i>i.e.</i>	Erreur Indépendant	Independent Error
<i>i.i.d.</i>	Indépendant et Identiquement Distribués	Independent and Identically Distributed
IMC	Intégration Monte Carlo	Monte Carlo Integration
IS	Echantillonnage d'Importance	Importance Sampling
MAP	<i>a Postérieur</i> Maximum	Maximum <i>a Posteriori</i>
MC	Monte-Carlo	Monte-Carlo
MCMC	Chaîne Markovien Monte Carlo	Markov Chain Monte Carlo
MMC	Méthode des Moindres Carrés	Least Squares Method
MCS	Monte Carlo Séquentiel	Sequential Monte Carlo
MMSE	Erreur Quadratique Moyen Minimal	Minimum Mean Squared Error
NG	Non-Gaussien	non-Gaussian
NL	Non-Linéaire	Non-Linear
PDF	Function de Densité de Probabilité	Probability Density Function
RV	Rapport de Vraisemblance	Likelihood Report
TB	Théorème de Bayes (TB)	Bayes Theorem
<i>v.a.</i>	Variable aléatoire	Random Variable

Table des matières

Remerciements	2
ملخص.....	3
Résumé.....	3
Abstract	4
Listes des acronymes et abréviations.....	5
Table des matières	6
Liste des figures.....	9
Introduction Générale.....	10
Chapitre 1 Model d'Espace d'Etat	12
1.1. Modèle Linéaire.....	12
1.2. Modèle Non-Linéaire.....	12
1.3. Modèle de Markov	13
1.3.1. Chaîne de Markov Observable	15
1.3.2. Chaîne de Markov Caché.....	16
1.3.2.1. Modèle d'état.....	17
1.3.2.2. Canal sans mémoire	18
Chapitre 2 Estimation d'Etat et Filtrage	19
2.1. Estimation.....	19
2.1.1. Définition.....	19
2.1.2. Applications.....	19
2.1.3. Caractéristiques	19
2.1.3.1. Biais	19
2.1.3.2. Erreur quadratique moyenne.....	20
2.1.3.3. Convergence.....	20
2.1.3.4. Efficacité	20
2.1.3.5. Robustesse.....	20
2.1.4. Types d'estimation	21
2.1.4.1. Estimation des Paramètres	21

2.1.4.2.	Estimation d'état	21
2.1.5.	Méthodes d'estimation	21
2.1.5.1.	Moindres carrés.....	21
2.1.5.2.	Observateur de Luenberger	22
2.1.5.3.	Méthode du maximum de vraisemblance	23
2.1.5.4.	Méthode des moments.....	24
2.2.	Filtrage.....	25
2.2.1.	Filtrage particulière	26
2.2.2.	Estimation Réursive	27
Chapitre 3	Théorie Bayésienne.....	29
3.1.	Introduction.....	29
3.2.	Définition.....	29
3.3.	Théorème de Bayés.....	31
3.3.1.	Formule	31
3.4.	Filtre Bayésien	32
3.4.1.	Définition.....	32
3.4.2.	Filtre de Kalman.....	33
3.4.3.	Filtre de Kalman Etendu	34
3.4.4.	Filtre Particulaire	34
3.5.	Estimation bayésienne séquentielle.....	34
3.5.1.	Principe.....	34
3.5.2.	Etapes	36
3.5.2.1.	Prédiction	36
3.5.2.2.	Mise à jour.....	37
Chapitre 4	Théorie Monte-Carlo.....	39
4.1.	Principe.....	39
4.1.1.	Algorithme d'Acceptation ou Rejet.....	40
4.2.	Intégration Monte-Carlo	41
4.3.	Echantillonnage de Monte Carlo.....	42
4.3.1.	Échantillonnage d'Importance	43
4.4.	Simulation de Monte-Carlo	45
4.5.	Utilisation	45
Chapitre 5	Filtre Particulaire.....	46
5.1.	Introduction.....	46
5.2.	Définition.....	46
5.3.	Représentation.....	47
5.3.1.	Particules	49
5.3.2.	Poids	50

5.4.	Etapes	51
5.4.1.	Algorithme de Base	52
5.4.2.	Calcul des Poids des Particules.....	53
5.4.3.	Ré-échantillonnage.....	53
5.5.	Modélisation.....	54
5.6.	Approximation de MC	55
Chapitre 6	Détection des Défauts et Validation des Signaux	56
6.1.	Introduction.....	56
6.2.	FPs pour la détection des défauts	56
6.3.	Algorithme Proposé.....	57
6.4.	Détection des Défauts et Validation des Signaux par FPs	58
6.4.1.	Algorithme.....	59
Chapitre 7	Programme	61
7.1.	Introduction.....	61
7.2.	Problématique.....	61
7.3.	Développement de la méthode	61
7.4.	Application et résultats	63
7.5.	Conclusion	67
Conclusion Générale	68
Bibliographie	69
Annexe 1- Notation de Probabilités	72
Annexe 2 – programme MATLAB	74

Figure 1-1 : les étapes de chaîne de Markov

Figure 1-2 : (a) Présentation d'une Chaîne de Markov, (b) matrice de transition et les probabilités du départ.

Figure 1-3 : canal sans mémoire

Figure 2-1 : observateur de Luenberger

Figure 3-1 : filtre Bayésien

Figure 3-2 Filtre Bayésien séquentiel.

Figure 4-1 : échantillonnage d'importance

Figure 5-1 : filtre à particule

Figure 7-1 : Organigramme présente les étapes d'organisation de programme MATLAB

Figure 7-2 : évolution de la densité d'état

Figure 7-3 : L'état en fonction état estimé par particule et chemins de particules

Figure 7-4 : L'observation en fonction d'observation filtrée par filtre à particules

Figure 7-5 : L'observation.

Figure 7-6 : L'observation filtrée par filtre à particules

Figure 7-7 : L'erreur d'estimation d'observation par le filtre à particules

Introduction Générale

Ce mémoire vise à fournir une introduction au *Filtres Particulaires* ou *Filtres à Particules (FPs)* qui sont devenues de plus en plus populaires au cours des dernières décennies qui permettent l'implémentation de l'inférence Bayésienne dans des modèles d'espace d'états dynamiques complexes [1].

Les *FPs* ou *Monte Carlo séquentiel (MCS)* [16] sont une classe des techniques basées sur la simulation. Les *FPs* sont généralement appliqués à des problèmes dits de filtrage [1], où l'objectif est d'estimer en ligne les états latents d'un processus stochastique, de telle sorte qu'après chaque observation séquentielle, l'état à l'origine de cette observation soit estimé.

Les problèmes d'estimation d'état, sont d'un grand intérêt dans d'innombrables applications pratiques. Dans tels types de problèmes, les données mesurées disponibles sont utilisées conjointement avec des connaissances préalables sur les phénomènes physiques et les dispositifs de mesure, afin de produire séquentiellement des estimations des variables dynamiques souhaitées. Ceci est accompli de telle manière que l'erreur est minimisée statistiquement.

Les problèmes d'estimation d'état sont résolus avec les filtres dits Bayésiens [2]. Une approche statistique Bayésienne essaye d'utiliser toutes les informations disponibles afin de réduire la grandeur d'incertitude présente dans un problème d'inférence ou de prise de décision. Au fur et à mesure que de nouvelles informations sont obtenues, elles sont combinées avec des informations antérieures pour former la base des procédures statistiques. Le mécanisme formel utilisé pour combiner les nouvelles informations avec les informations précédemment disponibles est connu sous le nom de *Théorème de Bayes (TB)*.

La méthode de *filtrage Bayésien (FB)* la plus connue est le *Filtre de Kalman (FK)*. Cependant, l'application du *FK* est limitée aux modèles linéaires avec des bruits additifs *Gaussiens*. Des extensions du *FK* ont été développées dans le passé pour des cas moins restrictifs en utilisant des techniques de linéarisation. De même, des méthodes de *Monte Carlo séquentiel (MC)* ont été développées afin de représenter la *densité postérieure* en termes d'échantillons aléatoires et de poids associés. Ces méthodes de *MC* sont généralement désignées par *FPs* et ne nécessitent

pas les hypothèses restrictives du *FK*. Par conséquent, les *FPs* peuvent être appliqués à des modèles *NLs* avec des erreurs (bruits) *non-Gaussiennes (NGs)*.

Ce mémoire comporte, outre cette *introduction*, *sept chapitres* et une *conclusion générale*.

Le *premier chapitre* expose le modèle d'espace d'état d'un système linéaire et d'un système *NL*. Ensuite, nous faisons une description du modèle de *Markov* y compris la chaînes de *Markov* ; cachée et observable.

Le *second chapitre* est composé de deux sections : l'estimation d'état et le filtrage, destinés pour l'implémentation de la détection et la validation des signaux. Pour l'estimation, nous avons cité ses applications, ses caractéristiques et ses différents types ; et quelques méthodes d'estimation. Pour le filtrage, nous avons concentré sur le filtrage Bayésien et particulaire.

Le *troisième chapitre*, relatif à la théorie Bayésienne, aborde la règle de Bayés ; les filtres Bayésiens tels que le *FK*, *FK étendu* et *FP* ; et enfin, l'estimation Bayésienne séquentielle ou récursive.

Le *quatrième chapitre* présente un développement de la théorie de monte Carlo qui englobe l'intégration Monte-Carlo, l'*Echantillonnage d'Importance (EI)* et la simulation de *Monte-Carlo*.

Le *cinquième chapitre* présente les *FPs* pour les systèmes *NLs*, leur définition, modélisation, étapes et modélisation.

Le *sixième chapitre* donne des illustrations sur l'aspect de détection des défauts et la validation des signaux en utilisant les *FPs*.

Le *dernier chapitre* est consacré, à son tour, à la discussion et l'analyse des résultats d'un programme que nous avons proposé. Ce programme s'exécute sous *MATLAB* et est destiné à l'estimation des signaux. Cette estimation est exploitée par la suite dans la validation des signaux et la détection des défauts.

Finalement, nous achevons ce mémoire par une conclusion générale qui résume nos contributions et efforts que nous avons fournis pour l'élaboration de ce mémoire.

1.1. Modèle Linéaire

La représentation d'état des systèmes linéaires en temps continu, s'écrit :

$$\begin{cases} \dot{x}(t) = F_t(t)x(t) + G_t x(t) \\ y(t) = H_t x(t) + D_t x(t) \end{cases} \quad (1)$$

- $x(t) \in R^n$ colonne qui représente les n variables d'état ;
- $u(t) \in R^m$ colonne qui représente les m commandent ;
- $y(t) \in R^p$ colonne qui représente les p observations ;
- $F_t \in R^{n \times n}$ matrice d'état variant dans le temps ;
- $G_t \in R^{n \times m}$ matrice de commande variant dans le temps ;
- $H_t \in R^{p \times n}$ matrice d'observation variant dans le temps ;
- $D_t \in R^{p \times m}$ matrice d'action directe variant dans le temps.

Dans le cas d'un système linéaire en temps discret, obtenu par la discrétisation du système à temps continu, la représentation d'état s'écrit de la manière suivante [12] :

$$\begin{aligned} x_{k+1} &= F_k x_k + G_k u_k \\ y_k &= H_k x_k + D_k u_k \end{aligned} \quad (2)$$

1.2. Modèle Non-Linéaire

On considère le problème simple suivant, où il s'agit d'estimer l'état X_k d'un système position au vu d'observations (Y_0, \dots, Y_k) recueillies avec un bruit blanc additif [5], pas nécessairement *Gaussien*. En général, les observations seules ne permettent pas d'estimer l'état caché, il est nécessaire de disposer d'un modèle *a priori*, suffisamment réaliste, pour décrire l'évolution de l'état entre deux instants consécutifs d'observation.

$$\begin{aligned} X_k &= F_k (X_k, W_k) \\ Y_k &= h_k (X_k) + V_k \end{aligned} \quad (3)$$

où $\{W_k\}$ et $\{V_k\}$ sont des bruits blancs indépendants. L'incertitude sur l'état initial est décrite par la distribution de probabilité $X_0 \sim \mu_0(dx)$.

1.3. Modèle de Markov

Une chaîne de *Markov* est une suite variables aléatoires $(X_n, n \in \mathcal{M})$ qui permettent de modéliser l'évolution dynamique d'un système aléatoire : X_n représente l'état du système à l'instant n . La propriété fondamentale de la *Chaîne de Markov (CM)*, dite *propriété de Markov*, est que son évolution future ne dépend du passé qu'au travers de sa valeur actuelle. Autrement dit, conditionnellement à X_n (X_0, X_1, \dots, X_n) et $(X_{n+k}, k \in \mathcal{M})$ sont indépendantes.

Un processus stochastique est un phénomène où intervient le hasard. Ce processus est dit *Markovien* si son évolution ne dépend pas de son passé, mais uniquement de son état présent (Ceci est appelé la propriété de *Markov*) [8].

Un *modèle de Markov*, aussi appelé *CM*, est un modèle statistique composé d'états et de transitions [7]. Une transition matérialise la possibilité de passer d'un état à un autre. Dans une *CM*, les changements des états sont nommés transitions et qui sont unidirectionnelles. Chaque état a des transitions vers tous les états, y compris eux-mêmes. Chaque transition est associée à une probabilité nommée *probabilité de transition* qui peut éventuellement être nulle. L'état initial caractérisé par les *probabilités du départ*, Π , et la matrice de transition décrivant les probabilités des transitions caractérisent un *modèle de Markov*.

Considérons un processus stochastique $\{X_n, n = 0, 1, 2, \dots\}$. Notons cet ensemble de valeurs possibles du processus par l'ensemble des entiers non-négatifs $\{0, 1, \dots\}$. Si $X_n = i$, alors le processus est dit être dans l'état i au temps n . On suppose que chaque fois que le processus est dans l'état i , il y a une probabilité P_{ij} qu'il sera ensuite dans l'état j . C'est-à-dire que nous supposons que [9] :

$$\begin{aligned} P_{ij} &= P\{X_{n+1} = j | X_n = i, X_{n-1} = i-1, \dots, X_1 = 1, X_0 = 0\} \\ &= P\{X_{n+1} = j | X_n = i\} \end{aligned} \tag{4}$$

Un tel processus stochastique est connu comme *CM*. L'équation ci-dessus peut être énoncée comme, pour une *CM*, la *DC* de tout état futur X_{n+1} étant donné les états passés X_0, \dots, X_{n-1} et l'état actuel X_n , est indépendant des états passés et ne dépend que de la état actuel.

La valeur P_{ij} représente la probabilité que le processus, lorsqu'il est dans l'état i , faire ensuite une transition vers l'état j . Comme les probabilités sont positives et le processus doit faire une transition vers un certain état, nous aurons :

$$P_{ij} \geq 0, i, j \geq 0; \quad (5)$$

et

$$\sum_{j=0}^{\infty} P_{ij} = 1, \quad i = 0, 1, \dots \quad (6)$$

Soit P la *matrice des probabilités de transition* à une étape P_{ij} , alors

$$P = \begin{pmatrix} p_{00} & p_{01} & p_{02} & \dots \\ p_{10} & p_{11} & p_{12} & \dots \\ \vdots & \vdots & \vdots & \vdots \\ p_{i0} & p_{i1} & p_{i2} & \dots \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix} \quad (7)$$

L'utilisation de la *CM* se fait séquentiellement dans le temps. A chaque "unité de temps", on opère une transition, ce qui génère finalement une séquence d'états :



Figure 1-1 : Les étapes de chaîne de Markov

On remarque qu'au moment $t - 1$ on était sur l'état 1, puis au moment t sur l'état 2, puis au moment $t + 1$ sur l'état 3. Comme une transition peut être au niveau d'un état, il est possible que cet état soit plusieurs fois affilée dans la séquence.

Figure 1-2 représente une simple *CM* avec 3 états, sa *matrice de transition* A et la probabilité du départ.

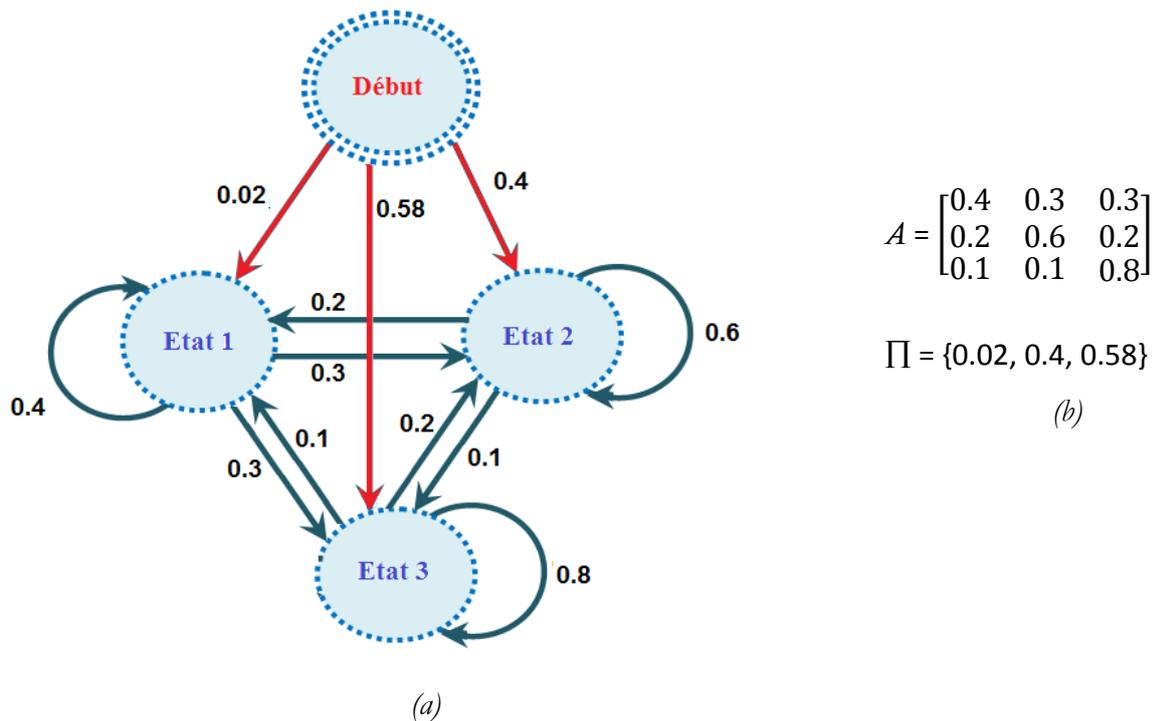


Figure 1-2 : (a) Présentation d'une Chaîne de Markov, (b) matrice de transition et les probabilités du départ. La somme des éléments de chaque ligne de A prend la valeur 1 et la somme des éléments de Π prend la valeur 1.

Ce modèle est composé d'un état "Début" qui sert à présenter les probabilités de départ des états de la CM : 40% (0.4) de chances de commencer par l'état 1 et 60% (0.6) de chances de commencer par l'état 2. Par définition, on ne revient jamais à l'état de départ, raison pour laquelle il n'y a jamais de transition vers cet état. On remarque également que la somme des probabilités des transitions partant d'un état est toujours égale à 1 (100%).

Un processus *Markovien* peut être modélisé théoriquement par un modèle dit de *Markov*. Il existe deux types de ce modèle : (a) *observable*, lorsque les états de ce processus sont complètement observables et (b) *caché*, lorsque ce processus possède des états qui sont partiellement observables seulement. En d'autres termes, les observations sont liées aux états du système, mais elles sont généralement insuffisantes pour déterminer précisément l'état.

1.3.1. Chaîne de Markov Observable

L'évolution du processus de *Markov* peut être représentée par un graphe de transitions d'états qui fait apparaître la structure du processus selon les règles suivantes [8] :

- Les états sont représentés par des sommets (État n). Nous parlons d'alphabet des états : $S = \{s_1, s_2, \dots, s_m\}$ les états de la CM.
- Les transitions (passage d'un état à un autre) sont représentées par des arêtes. Elles sont pondérées par leur probabilité (Flèche). Les probabilités sont regroupées dans une *matrice de*

transition de type carré, appelé aussi *matrice de probabilités*, *matrice de substitution* ou *matrice de Markov*. Si la probabilité de bouger de l'état i à l'état j en une étape de temps est a_{ij}

$$A = \{a_{ij} = P(s_j/s_i)\} \quad (8)$$

vue que la somme des probabilités des transitions à partir d'un état i vers tous les autres état doit être 1, $\sum a_{ij} = 1$.

- Les *probabilités de départ* : ce sont les probabilités de débuter dans un état ou un autre (point 0). Elles sont regroupées dans un vecteur d'initialisation :

$$\Pi = \{\pi_i = P(s_i)\} \sum_{i=1}^N \pi_i = 1. \quad (9)$$

Un modèle λ est dit observable car les états sont directement observables. Il est caractérisé par une matrice de transition A et un vecteur d'initialisation Π , nous notons :

$$\lambda = \{\Pi, A\} \quad (10)$$

1.3.2. Chaîne de Markov Caché

Dans une *CMC* les états $S = \{s_1, s_2, \dots, s_m\}$ sont non-observables. Cependant, ils émettent des signaux observables $O = (o_1, o_2, \dots, o_k)$ qui sont pondérés par leurs probabilités.

Une *CMC* est décrite par deux ensembles de probabilités : (a) *probabilités de transition* qui sont les probabilités de passer d'un état à l'autre, (b) *probabilités d'émission* qui sont les probabilités d'observer un symbole pour un état donné.

Le modèle *CMC*, λ , peut être représenté par [8] :

- La *matrice de transitions* :

$$A = \{a_{ij} = P(s_j/s_i)\} \text{ avec } \sum a_{ij} = 1. \quad (11)$$

- Le *vecteur d'initialisation* :

$$\Pi = \{\pi_i = P(S_i)\} \text{ avec } \sum_{i=1}^N \pi_i = 1. \quad (12)$$

- Les probabilités que l'état s_i émet le signal d'observation o_k sont regroupées dans une *matrice d'émission* (probabilité que le signal o_k soit observé dans l'état i):

$$E = \{e_i(o_k) = P(o_k/s_i)\} \text{ avec } \sum_{j=1}^N e(o_j) = 1. \quad (13)$$

Une *CMC* λ est caractérisé par une matrice de transition A , une matrice d'émission E et un vecteur d'initialisation Π , nous notons :

$$\lambda = \{\Pi, A, E\} \quad (14)$$

Une *CMO* peut être modélisée sous forme d'une *CMC* où les états correspondent aux événements observés, c'est-à-dire que chaque état s_i a une probabilité l_i d'émettre l'observation b_i . Les *CMC* possèdent un très grand pouvoir de modélisation, et permettent de décrire très simplement des structures de dépendances temporelles complexes. Pratiquement, il suffit de spécifier les transitions $X_k|X_{k-1} = x$ et les *densités d'émission* $Y_k|X_k = y$.

Les *modèles de Markov Cachés (MMC)* définissent un processus stochastique : (a) *non-déterministe* (une même séquence peut être générées de plusieurs manières différentes), (b) *Markovien* : le chemin (la séquence des états) constitue une *CM* simple puisque la probabilité de transition vers un état ne dépend que de l'état actuel et non des états rencontrés précédemment, (c) *caché* : on observe la séquence générée par le modèle mais pas la séquence des états qui génèrent ces observations.

1.3.2.1. Modèle d'état

On suppose que la suite des états cachés $\{X_k\}$ forme une *CM* sur un espace E . Cet espace dépend du temps, c'est-à-dire où X_k prend ses valeurs dans un espace E_k , i.e. conditionnellement aux états passés (X_0, \dots, X_{k-1}) , l'état présent X_k ne dépend que de l'état précédent X_{k-1} .

On en déduit [5] :

$$\mathbb{P}[X_0 \in dx_0, \dots, X_n \in dx_n] = \underbrace{\mathbb{P}[X_0 \in dx_0]}_{\mu_0(dx_0)} \prod_{k=1}^n \underbrace{\mathbb{P}[X_k \in dx_k | X_{k-1} = x_{k-1}]}_{Q_k(x_{k-1}, dx_k)} \quad (15)$$

Cette situation est complètement décrite par la donnée de la distribution initiale et de la caractéristique locale

$$\begin{aligned} \mathbb{P}[X_0 \in dx] &= \mu_0(dx) \\ \mathbb{P}[X_k \in dx' | X_{k-1} = x] &= Q_k(x, dx') \end{aligned} \quad (16)$$

En général, il n'existe pas d'expression explicite pour le noyau de transition $Q_k(x, dx')$ ou bien cette expression est si compliquée que *calculer* une intégrale telle que

$$\mu Q_k(dx') = \int E\mu(dx) Q_k(x, dx') \quad (17)$$

est impossible. Dans la plupart des cas pratiques, la distribution de probabilité *PDF* $Q_k(x, dx')$ n'admet déjà pas de densité par rapport à la mesure de Lebesgue sur E . $\int E\mu(dx) Q_k\mu(dx) Q_k(x, dx')$.

En revanche, il est souvent très facile de *simuler* une *v.a.* selon la *PDF* $Q_k(x, dx')$. Par conséquent, on fera l'hypothèse que pour tout point de départ $x \in E$, il est facile de *simuler* une *v.a.* selon la *PDF* $Q_k(x, dx')$.

1.3.2.2. Canal sans mémoire

L'hypothèse usuellement faite dans les *CMC* pour relier les observations aux états cachés, est de supposer que les observations sont indépendantes, sachant les états cachés, et que la *probabilité d'émission* de Y_k ne dépend que de X_k [5].

$$\begin{aligned}
 & \mathbb{P}[Y_k \in dy_0, \dots, Y_n \in dy_n | X_0 = x_0, \dots, X_n = x_n] \\
 &= \prod_{k=0}^n \mathbb{P}[Y_k \in dy_k | X_0 = x_0, \dots, X_n = x_n] \\
 &= \underbrace{\prod_{k=0}^n \mathbb{P}[Y_k \in dy_k | X_k = x_k]}_{g_k(x_k, y_k) \succ_k(dy_k)} \tag{18}
 \end{aligned}$$

La fonction de vraisemblance est alors définie par [10]:

$$\Psi_k(x) = g_k(x, y_k) \tag{19}$$

Cette situation est complètement décrite par le diagramme de dépendance ci-dessous [5] :

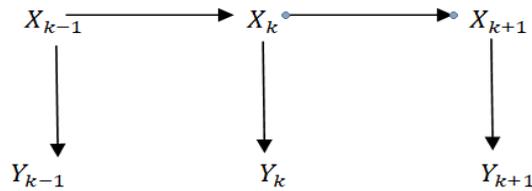


Figure 1-3 : Canal sans mémoire

et par la donnée de la distribution initiale et des caractéristiques locales [10] :

$$\begin{aligned}
 & \mathbb{P}[X_0 \in dx] = \mu_0(dx) \\
 & \mathbb{P}[X_k \in dx' | X_{k-1} = x] = Q_k(x, dx') \\
 & \mathbb{P}[Y_k \in dy' | X_k = x'] = g_k(x', y') \succ_k(dy')
 \end{aligned} \tag{20}$$

On peut vérifier que le modèle (18) rentre bien dans ce cadre.

Dans tous les modèles où le bruit d'observation est additif, avec une densité connue explicitement, il est en principe facile de calculer la fonction de vraisemblance (19) mais ce calcul peut parfois être couteux. Si le bruit d'observation n'est pas additif, une solution possible consiste à considérer un modèle perturbé, incluant un petit bruit d'observation de densité connue [6].

Chapitre 2 Estimation d'Etat et Filtrage

2.1. Estimation

2.1.1. Définition

L'estimation, nommée aussi prédiction [12], fait partie des mathématiques statistiques et qui permet, à partir des observations effectuées sur un système, de prévoir par des valeurs proches les différents paramètres de ce système qui sont supposés inconnus.

Soit θ un paramètre dont la vraie valeur, considérée inconnue, est θ_0 . Un estimateur est une fonction des observations (statistiques) de l'échantillon dont la valeur est utilisée à la place de la valeur vraie θ_0 du paramètre θ [13]. La valeur prise par un estimateur sur un échantillon donné est appelé une estimation (de θ_0).

2.1.2. Applications

L'estimation est largement utilisée, elle se trouve dans de nombreux domaines, à savoir: (a) *statistique* : estimer le nombre de la population dans une région ; (b) *détection des défauts* : estimer les défauts des capteurs ou des actionneurs. Dans notre travail, on s'intéresse sur la détection par estimation ; (c) *économie* : estimer la resette annuelle d'une industrie; (d) *navigation* : le but du radar est estimer l'objet (avion, bateaux, missile, ... etc.).

2.1.3. Caractéristiques

Un estimateur du paramètre θ est une valeur $\hat{\theta}$ calculée sur un échantillon tiré au hasard. La valeur $\hat{\theta}$ est une *v.a.* possédant une espérance $E[\hat{\theta}]$ et une variance $[\hat{\theta}]$. On comprend alors que sa valeur puisse fluctuer selon l'échantillon [12]. Elle a de très faibles chances de coïncider exactement avec la valeur θ qu'elle est censée représenter. L'objectif est alors de minimiser l'erreur entre la valeur réelle θ et son estimation $\hat{\theta}$. Un estimateur est caractérisé par sa *convergence, biais, efficacité, erreur quadratique et robustesse*.

2.1.3.1. Biais

Le biais d'estimation $\hat{\theta}$ d'un paramètre est défini par :

$$Bais[\hat{\theta}] = E[\hat{\theta}] - \theta \quad (21)$$

On dit que l'estimateur $\hat{\theta}$ est :

- sans biais (non-biaisé) si $Biais[\hat{\theta}] = 0$, c'est-à-dire si $E[\hat{\theta}] = \theta$.
- avec biais (biaisé) si $Biais[\hat{\theta}] \neq 0$, c'est-à-dire si $E[\hat{\theta}] \neq \theta$.

2.1.3.2. Erreur quadratique moyenne

La précision globale d'un estimateur est une mesure par son *Erreur Quadratique Moyenne (EQM)*.

Soit $\hat{\theta}$ un estimateur d'un paramètre θ , *EQM* de $\hat{\theta}$ est défini par :

$$EQM[\hat{\theta}] = [\hat{\theta} - \theta]^2 \quad (22)$$

On a toujours :

$$EQM[\hat{\theta}] = Var[\hat{\theta}] + [Biais[\hat{\theta}]]^2 \quad (23)$$

Lorsque $\hat{\theta}$ est sans biais, on a :

$$Biais[\hat{\theta}] = 0, \text{ et (23) sera : } EQM[\hat{\theta}] = Var[\hat{\theta}] \quad (24)$$

Le critère de *EQM* permet de comparer les estimateurs d'un même paramètre. Ainsi, si $\hat{\theta}_1$ et $\hat{\theta}_2$ sont deux estimateurs d'un paramètre θ , le plus efficace des deux est celui ayant la plus petite *EQM*. Si on considère que $\hat{\theta}_1$ est plus efficace que $\hat{\theta}_2$ alors :

$$EQM[\hat{\theta}_1] < EQM[\hat{\theta}_2] \quad (25)$$

2.1.3.3. Convergence

Un estimateur $\hat{\theta}$ est convergent si sa distribution tend à se concentrer autour de la valeur inconnue à estimer θ , à mesure que la taille d'échantillon augmente, c'est-à-dire si :

$$\lim_{n \rightarrow \infty} Var[\hat{\theta}] = 0 \quad (26)$$

2.1.3.4. Efficacité

Un estimateur $\hat{\theta}$ sans biais est efficace si sa variance est la plus faible parmi les variances des autres estimateurs sans biais. Ainsi, si $\hat{\theta}_1$ et $\hat{\theta}_2$ sont deux estimateurs du paramètre θ , l'estimateur $\hat{\theta}_1$ est plus efficace que celui $\hat{\theta}_2$ si :

$$Var[\hat{\theta}_1] < Var[\hat{\theta}_2] \text{ et } E[\hat{\theta}_1] = E[\hat{\theta}_2] = \theta \quad (27)$$

2.1.3.5. Robustesse

On dit qu'un estimateur est robuste, si les valeurs de l'estimateur varient d'une manière très faible.

2.1.4. Types d'estimation

Deux types d'estimation sont couramment utilisés; estimation des paramètres et estimation d'état.

2.1.4.1. Estimation des Paramètres

Un estimateur du paramètre inconnu θ d'un modèle est une fonction qui fait correspondre à une suite d'observations x_1, x_2, \dots, x_n , issues du modèle, la valeur nommée estimée ou estimation [12].

$$\hat{\theta} = f(x_1, x_2, \dots, x_n) \quad (28)$$

Un estimateur ne doit évidemment jamais dépendre de l'historique de θ , il ne dépend que des observations empiriques

2.1.4.2. Estimation d'état

L'estimation d'état est basée sur la représentation d'état qui décrit le processus physique fondamentale de l'application, alors cette estimation consiste à contracter l'ensemble $x(t)$ a temps continu ou x_k à temps discret, en utilisant l'information fournie par les mesures.

On remarque qu'il y a une différence fondamentale entre les problèmes d'estimation du paramètre d'une part, et l'estimation de l'état d'autre part. Il s'agit de la commande en temps (ou en espace) dans l'estimation d'état, qui est absent dans l'estimation des paramètres.

2.1.5. Méthodes d'estimation

Nous présentons quelques méthodes plus connues d'estimation qui peuvent être appliquées dans la détection des défauts des signaux.

2.1.5.1. Moindres carrés

Historiquement, la *méthode des moindres carrés (MMC)* est le premier outil d'estimation. Elle reste encore très utile et appréciée pour son caractère intuitif et le peu de concepts mathématiques et d'hypothèses qu'elle nécessite.

Le principe de cette *MMC* est de minimiser une fonction de l'erreur quadratique qu'on l'appelle fonction de coût définie par :

$$J = \sum_{k=1}^N [\varepsilon(k)]^2 \quad (29)$$

où $\varepsilon(k)$ représente l'erreur de prédiction commise sur l'estimation.

Parmi les avantages de la *MMC* est sa simplicité à implémenter, et son traitement du problème d'estimation de la même manière que celui d'optimisation déterministe, ce qui ne nécessite pas des hypothèses stochastiques.

Le principal inconvénient de la *MMC* est le fait que les performances d'estimation sont basés sur la nécessité de la connaissance des caractéristiques du bruit (moyenne, covariance) [12]. En plus, elle utilise une large mémoire dans ses calculs, car en fait elle exploite à chaque instant toutes les données de l'archive calculées auparavant.

2.1.5.2. Observateur de Luenberger

Soit le système linéaire discret décrit par les équations suivante [12] :

$$\begin{cases} x_{k+1} = Fx_k + Gu_k \\ y_k = Hx_k + Du_k \end{cases} \quad (30)$$

Les matrices F , G et H sont constantes. On se place dans le cadre déterministe (les bruits sont négligés). L'observabilité du système doit être confirmée en premier lieu, par le critère de *Kalman* qui dit que le système est *observable*, si et seulement si.

$$\text{rang} \begin{bmatrix} H \\ HF \\ \vdots \\ HF^{n-2} \\ HF^{n-1} \end{bmatrix} = n \quad (31)$$

où n est la dimension de l'espace d'état.

L'observateur linéaire de *Luenberger* est composé d'un estimateur, fonctionnant en boucle ouverte, caractérisé par la même dynamique que celle du système. La structure fonctionnant en boucle fermée obtenue par le biais du gain, détermine la dynamique propre de cet observateur

L'erreur $e = (y - \hat{y}) = H(x - \hat{x})$ assure le fonctionnement en boucle fermée. Elle est multipliée par une matrice de gain K et est envoyée à l'entrée de l'observateur afin de corriger les variables d'états estimées.

L'observateur de *Luenberger* est décrit par :

$$\hat{x}_{k+1} = (F - KH)\hat{x}_k + Gu_k \quad (32)$$

Un choix judicieux du gain K tel que les valeurs propres de la matrice $(F - KH)$ soient à parties réelles strictement négatives, peut modifier la dynamique de l'observateur et de la vitesse de convergence de l'erreur vers zéro.

Les différentes grandeurs citées sont :

- U : vecteur d'entrée du système et de l'observateur.
- X : vecteur d'état.
- Y : vecteur de sortie du système.
- \hat{X} : vecteur d'état reconstruit par l'observateur.
- \hat{Y} : vecteur de sortie de l'observateur.

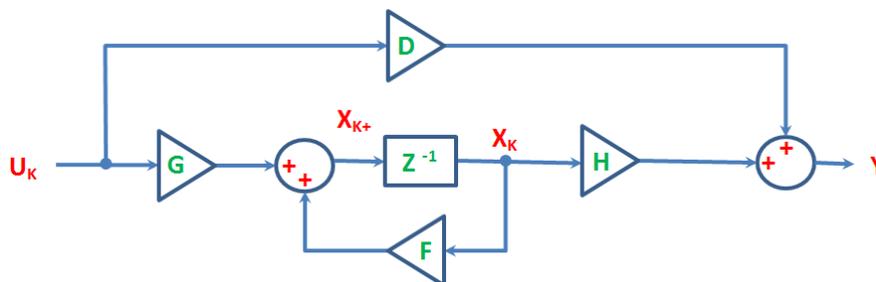


Figure 2-1 : Observateur de Luenberger

2.1.5.3. Méthode du maximum de vraisemblance

Cette méthode consiste, étant donné un échantillon de valeurs x_1, x_2, \dots, x_n , à prendre comme estimation de θ la valeur θ qui rend maximale la *vraisemblance* [15] :

$$L(x_1, x_2, \dots, x_n; \theta) \quad (33)$$

En pratique on prend comme estimation θ une solution de l'équation :

$$\frac{\partial}{\partial \theta} \ln L(X - \theta) = 0 \quad (34)$$

dite "équation de la vraisemblance"

Intuitivement, puisque L représente une densité de probabilité, cela revient à supposer que l'événement qui s'est produit était le plus probable.

2.1.5.4. Méthode des moments

C'est la méthode la plus naturelle, l'idée de base est d'estimer une espérance mathématique par une moyenne empirique, une variance par une variance empirique, etc...

Si le paramètre à estimer est l'espérance de la loi des X_i , alors on peut l'estimer par la moyenne empirique de l'échantillon. Autrement dit, si $\theta = E[X]$, alors l'estimateur de θ par la méthode des moments (EMM) est [15] :

$$\hat{\theta} = \bar{X}_n = \frac{\sum_{i=1}^n X_i}{n} \quad (35)$$

Plus généralement, pour $\theta \in R$, si $E(X) = \varphi(\theta)$ où φ est une fonction inversible, alors l'estimateur θ de par la méthode des moments est $\tilde{\theta} = \varphi^{-1}(\bar{X}^n)$. Aussi, on estime la variance de la loi des X_i par la variance empirique de l'échantillon :

$$Var[\hat{\theta}] = S_n^2 = \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X}_i)^2 = \frac{1}{n} \sum_{i=1}^n (X_i^2 - \bar{X}_i^2) \quad (36)$$

Plus généralement, si la loi des X_i a deux paramètres θ_1 et θ_2 , tels que

$$(E(X), Var(X)) = \varphi(\theta_1, \theta_2) \quad (37)$$

où φ est une fonction inversible, alors les estimateurs de θ_1 et θ_2 par la méthode des moments sont:

$$(\hat{\theta}_{1n}, \hat{\theta}_{2n}) = \varphi^{-1}(\bar{X}_i, S_n^2) \quad (38)$$

Ce principe peut naturellement se généraliser aux moments de tous ordres, centrés ou non-centrés :

$$\begin{aligned} E[(X - E(X))^K] \\ E(X)^K, K \geq 1 \end{aligned} \quad (39)$$

Cette méthode est avantageuse par la simplicité de sa mise en œuvre, ainsi qu'elle nécessite pas d'hypothèses sur la loi X_i . Par contre le majeur inconvénient de cette méthode c'est l'absence d'optimalisation et la nécessité d'analyse statistique cas par cas.

2.2. Filtrage

Les algorithmes de filtrages permettent l'approximation des séquences de distributions $p(X_{1:T})$ conditionnellement aux observations $\{Y_{1:T}\}$ pour $t = 1, 2, \dots, T$. Lorsque les distributions f et g sont linéaires, que ϵ_x et ϵ_y sont additifs et distribués selon une loi normale, les FKs permettent une estimation optimale de la distribution $p(X_{1:T})$. Dès que ces hypothèses ne sont pas vérifiées, il est préférable d'utiliser les méthodes d'inférence Bayésienne avec $p(x_{1:T} | Y_{1:T})$ comme distribution *a posteriori*, la distribution des $x_{1:T}$, $p(x_{1:T})$ comme distribution *a priori* et $p(Y_{1:T} | x_{1:T})$ comme vraisemblance. En utilisant ces notations, il est possible de redéfinir la distribution *a priori* du processus à étudier X_t , soit ; [16]

$$p(X_{1:T}) = h(x_1) \prod_{t=2}^T f(x_t | x_{t-1}) \quad (40)$$

ainsi que la vraisemblance Bayésienne

$$p(y_{1:T} | x_{1:T}) = \prod_{t=1}^T h(y_t | x_t) \quad (41)$$

Selon la formule de Bayes, la distribution de l'ensemble des réalisations $x_{1:T}$ conditionnées sur celle des observations $y_{1:T}$ s'écrit :

$$p(x_{1:T} | y_{1:T}) = \frac{p(x_{1:T}) p(y_{1:T} | x_{1:T})}{\int p(x_{1:T}) p(y_{1:T} | x_{1:T}) dx_{1:T}} \quad (42)$$

où $p(x_{1:T} | y_{1:T})$ correspond à la fonction *a posteriori* non-normalisée utilisée dans le reste du document.

Dans des situations de non-linéarité, de non-normalité ou de haute dimensionnalité, l'intégrale au dénominateur de l'équation (42) est difficile voire impossible à calculer analytiquement. Pour évaluer ces quantités, les méthodes MC sont généralement utilisées. Il est à noter que pour alléger le document, la notation suivante est utilisée,

$$p(x_{1:T}, y_{1:T}) = p(x_{1:T}) p(y_{1:T} | x_{1:T}) \text{ et } p(y_{1:T}) = \int p(x_{1:T}, y_{1:T}) dx_{1:T} \quad (43)$$

De plus, il est parfois préférable d'indexer la distribution *a posteriori* selon un paramètre ϕ . Si ce paramètre est connu, alors la distribution *a posteriori* $p_\phi(x_{1:T}, y_{1:T})$ est proportionnelle à la distribution,

$$p_\phi(x_{1:T}, y_{1:T}) = g_\phi \prod_{t=2}^T f_\phi(x_t | x_{t-1}) \prod_{t=1}^T h_\phi(y_t | x_t) \quad (44)$$

Dans le cas contraire, cette distribution est proportionnelle à la distribution bivariée,

$$p(\phi, x_{1:T}|y_{1:T}) \propto p_\phi(x_{1:T}, y_{1:T})p(\phi) \quad (45)$$

Qui est alors utilisée comme distribution *a posteriori* dans *l'inférence Bayésienne*. Puisque l'ajout de l'indexation aux résultats généraux est direct, par mesure d'économie, l'indexation est omise des formulations utilisées dans ce document.

En résumé, le but est de déterminer la distribution des réalisations du processus X_t au fur et à mesure que nous disposons de nouvelles informations liées aux observations Y_t , donc au temps t , nous estimons $p(x_{1:T}, y_{1:T})$ et $p(y_{1:T})$. À partir de cette approximation, il est relativement simple de faire une estimation des moments de $p(x_{1:T})$ par échantillonnage de $p(x_{1:T})$ et l'utilisation de la moyenne empirique.

2.2.1. Filtrage particulière

Le *FP* est une méthode numérique de filtrage adaptée aux cas de modèles *NLS* munis des bruits *NGs*.

Il est parfois difficile, voire impossible, d'établir explicitement les filtres optimaux. L'idée du *FP* [38-39] est d'approximer à chaque instant n , le filtre $p_k(x)$ par une suite de loi discrète

$$p_k^N(x) = \sum_{i=1}^N w_k^i \delta_{\varepsilon_k^i}(x) \quad (46)$$

où δ est la mesure de Dirac.

Dans cette méthode, dite méthode de *MCS*, les étapes de prédiction et correction sont représentées par des systèmes de particules ε_k^i où $\varepsilon_k^i \sim p_k(x)$.

A chaque simulation,

$$p_k^N(\varepsilon_k^i) = w_k^i \quad (47)$$

est le poids de la particule ε_k^i .

Il se peut que dans la simulation, des particules ont des poids w_k^i très faibles ou presque nuls et ne peuvent plus contribuer à l'approximation. Cette dégénérescence de particules implique une redistribution. Les particules de poids négligeables devront être remplacées par duplication des particules de poids plus importants non-négligeables. On procède alors à un regroupement de particules dans une région d'état pour ralentir la dégénérescence. Une redistribution des particules de poids plus importants est nécessaire pour compléter le nombre total des particules.

Cette redistribution ne doit pas se faire sur un petit nombre de particules sinon on perd l'idée de convergence de la méthode.

On peut considérer un ré-échantillonnage multinomial : à partir des particules $\varepsilon_k^1, \dots, \varepsilon_k^N$ avec les probabilités w, \dots, w_k^N , on fait un échantillonnage de N particules de poids $\frac{1}{N}$. Cela revient à multiplier les particules les plus importantes.

2.2.2. Estimation Réursive

Dans de nombreuses applications, les observations sont disponibles consécutivement. Cela motive un cadre récursif pour permettre l'estimation en ligne.

La densité du filtre peut être mise à jour de manière récursive à l'aide des équations suivantes [29]:

$$p(x_t|y_{0:t}) = \frac{p(x_t|y_t)p(x_t|y_{0:t-1})}{p(y_t|y_{0:t-1})} \quad (48)$$

$$p(y_t|y_{0:t-1}) = \int p(y_t|x_t)p(x_t|y_{0:t-1})dx_t \quad (49)$$

$$p(x_{t+1}|y_{0:t}) = \int p(x_{t+1}, x_t)p(x_t|y_{0:t})dx_t \quad (50)$$

Ce schéma consiste en une mise à jour de la mesure selon (48), (49) et une mise à jour de probabilité selon (50).

Utilisation de la propriété de la *CM* et du théorème de Bayés

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} \quad (51)$$

$$p(y) = \int p(x, y)dx = \int p(y|x)p(x)dx \quad (52)$$

On peut montrer que

$$p(x_t|y_{0:t}) = p(x_t|y_t, y_{0:t-1}) = \frac{p(y_t|x_t, y_{0:t-1})p(x_t|y_{0:t-1})}{p(y_t|y_{0:t-1})} = \frac{p(y_t|x_t)p(x_t|y_{0:t-1})}{p(y_t|y_{0:t-1})} \quad (53)$$

Aussi,

$$p(y_t, x_t|y_{0:t-1}) = p(y_t|x_t, y_{0:t-1})p(x_t|y_{0:t-1}) = p(y_t|x_t)p(x_t|y_{0:t-1}) \quad (54)$$

$$p(x_{t+1}, x_t|y_{0:t}) = p(x_{t+1}|x_t, y_{0:t})p(x_t|y_{0:t}) = p(x_{t+1}|x_t)p(x_t|y_{0:t}) \quad (55)$$

ce qui implique que (intégrant les deux côtés)

$$p(y_t|y_{0:t-1}) = \int p(y_t|x_t)p(x_t|y_{0:t-1})dx_t \quad (56)$$

$$p(x_{t+1}|y_{0:t}) = \int p(x_{t+1}|x_t)p(x_t|y_{0:t})dx_t \quad (57)$$

Cependant, la récursivité est un résultat plutôt théorique puisque les intégrales (multidimensionnelles) impliquées ne permettent pas généralement une solution analytique. Le cas linéaire, *Gaussien* est une exception et sa solution est connue sous le nom de *FK*.

Les *FPs* approchent numériquement la *PDF postérieure* $p(x_{t:0}|y_{0:t})$ en utilisant des techniques de *MC*. L'approximation de la densité de filtrage $p(x_t|y_{0:t})$ est obtenue par marginalisation.

3.1. Introduction

Supposons que les événements dans une partition de l'espace d'échantillonnage sont choisis afin de réduire une source importante d'incertitude si nous connaissons quel événement s'est produit. Après lectures des informations, il reste une incertitude sur quel événement de la partition s'est produit. Le résultat suivant, appelé *TB*, donne une règle simple de calcul [9].

Le *TB* est une extension relativement mineure de la définition de la probabilité conditionnelle. Si nous connaissons $P(A|b_i)$ pour tout i , ce théorème nous permet de calculer des *probabilités conditionnelles* dans l'autre sens, c'est-à-dire que nous peut trouver $P(b_i|A)$ du $P(A|b_i)$'s

TB Soit les événements b_1, \dots, b_k former une partition de l'espace échantillon S tel que $P(A) > 0$. Alors [9], pour $i = 1, \dots, k$

$$P(b_i|A) = \frac{P(b_i) p(A|b_i)}{\sum_{j=1}^k P(b_j) p(A|b_j)} \quad (58)$$

Remarque Version conditionnelle du *TB*.

$$P(b_i|AC) = \frac{P(b_i|C) p(A|b_iC)}{\sum_{j=1}^k P(b_j|C) p(A|b_jC)} \quad (59)$$

Le *TB* est une branche de la théorie mathématique des probabilités qui permet de modéliser l'incertitude et les résultats qui nous intéressent en intégrant connaissances préalables et données d'observation

L'analyse *Bayésienne*, l'interprétation de la probabilité comme la mesure conditionnelle de l'incertitude, sont des méthodes populaires pour résoudre les problèmes inverses [18]. Avant de nous lancer dans l'inférence et l'estimation *Bayésiennes*, nous présentons d'abord quelques statistiques *Bayésiennes* fondamentales.

3.2. Définition

Soit $p(x|y)$ représentent la *PDF* de x conditionnés sur des mesures y . La statistique, $\Psi(x)$, est dit être "suffisante" si la distribution de x conditionnellement à Ψ ne dépend pas de y . En d'autres termes, $p(x|y) = p(x|y')$ pour tous deux ensembles y et y' on trouve que $\Psi(y) = \Psi(y')$ [18]

La statistique suffisante $\Psi(x)$ contient l'ensemble des informations portées par x sur y .

Il existe trois types de problèmes insolubles liés à la statistique *Bayésienne* [18] :

- *Normalisation* : Étant donné $p(x)$ préalable et la probabilité $p(y|x)$, la partie *postérieure* $p(x|y)$ est obtenu par le produit préalable et la probabilité divisée par un facteur de normalisation, comme ;

$$P(x|y) = \frac{p(y|x)p(x)}{\int p(y|x)p(x)dx} \quad (60)$$

- *Marginalisation* : Compte tenu de la conjointe *a posteriori* (x, y) , la marginale *a posteriori* est

$$P(x|y) = \int p(x, y|y) dy \quad (61)$$

- *Attente* : Compte tenu de la *PDF* conditionnelle, certaines statistiques moyennes d'intérêts peuvent être calculées

$$\mathbb{E}_{p(x|y)} [f(x)] = \int f(x) P(x|y) dx \quad (62)$$

Avec l'inférence *Bayésienne*, toutes les incertitudes (y compris les états, les paramètres qui sont soit variant dans le temps ou lois *a priori* fixes mais inconnues) sont traitées comme des variables aléatoires. La déduction (inférence) effectuée dans le cadre *Bayésien*, donne toutes les l'information disponible. L'objectif de l'inférence *Bayésienne* est d'utiliser les lois *a priori* et les connaissances de cause à effet, quantitativement et qualitativement, pour déduire la *DC*.

Généralement, Il y a trois niveaux de raisonnement probabiliste dans l'analyse *Bayésienne* (*i.e.*, analyse *Bayésienne* hiérarchique) :

- Choix du modèle, compte tenu des données et des lois *a priori* assumées.
- Estimation des paramètres pour ajuster les données compte tenu du modèle et des lois *a priori*.
- Mise à jour des paramètres préalable.

L'optimisation et l'intégration sont deux problèmes numériques fondamentaux qui se posent dans l'inférence statistique.

3.3. Théorème de Bayés

En statistique et en théorie des probabilités, le *TB*, également connu sous le nom de règle de *Bayes*, est une formule mathématique utilisée pour déterminer la probabilité conditionnelle d'événement [19]. Essentiellement, le *TB* décrit la probabilité d'un événement sur la base de la connaissance préalable des conditions qui pourraient être pertinentes pour l'événement.

3.3.1. Formule

Le *TB* s'exprime dans la formule suivante [16]:

$$p(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (63)$$

où:

- $P(A)$ (ou $P(B)$) est la probabilité de l'événement A (ou B)
- $P(A|B)$ (ou $P(B|A)$) est la probabilité que l'événement A (ou B) se produit, étant donné que l'événement B (ou A) s'est produit.

Un cas particulier du *TB* est lorsque l'événement A est une variable binaire. Dans un tel cas, le théorème s'exprime de la manière suivante :

$$p(A|B) = \frac{P(B|A)P(A)}{P(B|A^-)P(A^-) + P(B|A^+)P(A^+)} \quad (64)$$

où:

- A^- et A^+ sont des événements résultats mutuellement exclusifs de l'événement A .
- $P(B|A^-)$ est la probabilité que l'événement B se produise étant donné que l'événement A^- s'est produit.
- $P(B|A^+)$ – la probabilité que l'événement B se produise étant donné que l'événement A^+ s'est produit.

Les formules de *Bayes* permettent de déterminer les *DC* en fonction de la densité conjointe et des lois marginales.

Considérons une *v.a.* X a valeurs dans \mathbb{R}^d et de densité $p(x)$ [16]:

$$E\phi(X) = \int_{\mathbb{R}^d} \phi(x)p_x(x)dx \quad \forall \phi \in C_b(\mathbb{R}^d) \quad (65)$$

Par définition, la densité conjointe $p_{x,y}$ de X et Y est telle que $\forall \phi \in C_b(\mathbb{R}^d \times \mathbb{R}^d)$

$$E\phi(X, Y) = \int_{\mathbb{R}^d} \int_{\mathbb{R}^d} \phi(x, y)p_{x,y}(x, y)dxdy \quad (66)$$

On définit la *DC* de X sachant que $Y = y$, notée $P_{X|Y=y}$ telle que : $\forall \emptyset \in C_b(\mathbb{R}^d)$.

$$E[\emptyset(X) | Y = y] = \int_{\mathbb{R}^d} \emptyset(x) p_{X|Y=y}(x) dx = \frac{\int_{\mathbb{R}^d} \emptyset(x) p_{x,y}(x,y) dx}{\int_{\mathbb{R}^d} p_{x,y}(x,y) dy} \quad (67)$$

Et :

$$p_{X|Y=y}(x) = \frac{p_{x,y}(x,y)}{p_y(y)} = \frac{p_{x,y}(x,y)}{\int_{\mathbb{R}^d} p_{x,y}(x,y) dx} = \frac{p_{Y|X=x}(y) p_x(x)}{p_y(y)} \quad (68)$$

$$= \frac{p_{Y|X=x}(y)}{p_y(y)} p_x(x) = \frac{p_{Y|X=x}(y)}{\int_{\mathbb{R}^d} p_{x,y}(x,y) dx} \int_{\mathbb{R}^d} p_{x,y}(x,y) dy \quad (69)$$

3.4. Filtre Bayésien

3.4.1. Définition

Le *FB* c'est une méthode statistique permettant d'estimer l'état d'un objet (d'un système) qui n'est pas directement observable. Il est utilisé dans de applications très variées [32].

- *Suivi de cibles* : Position et vitesse d'un objet volant, détection et suivi de personnages dans une image et estimation de la position d'un véhicule ;
- *Filtrage* : Dé bruitage des signaux ;
- *Analyse* : Estimation spectrale non-stationnaire ;
- *Surveillance industrielle* : Détection précoce de panne de boites de vitesses.

Les *FBs* sont regroupés en deux majeurs ensembles : continus et discrets. Les *FKs* sont des filtre Gaussiens paramétriques, tandis que les *FPs* sont filtres non-paramétriques (*Figure 3-1*).

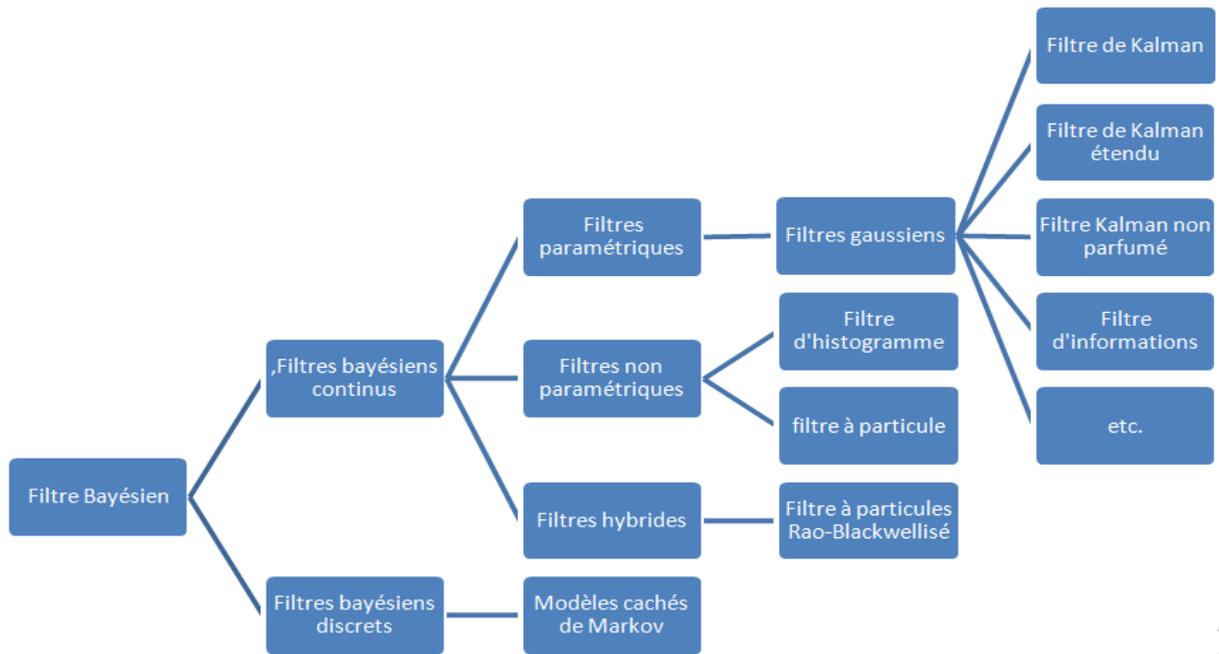


Figure 3-1 : Filtre Bayésien.

3.4.2. Filtre de Kalman

Le *FK* représente une extension de la théorie de Wiener applicable aux signaux non-stationnaires sous leur forme temporelle en présence des conditions initiales et d'entrées déterministes. Il résout le problème de filtrage linéaire et se présente sous la forme d'un ensemble d'équations récurrentes plus faciles à résoudre sur calculateur numérique, ce qui explique le succès remarquable de l'approche proposée par *Kalman*. Sa réalisation fournit non seulement l'estimation optimale, mais aussi la variance de l'erreur de l'estimation. C'est un observateur donnant la prédiction ou l'estimation du vecteur d'état. Lorsque les variances des bruits [18] sont connues, c'est le meilleur observateur linéaire, de plus, si les bruits sont blancs *Gaussiens*, linéaires ou non, c'est lui dont la variance de l'erreur d'estimation est la plus faible.

Le *FK* s'applique au système dynamique linéaire, continu ou discret, où le bruit de mesure est blanc.

L'estimation de *Kalman* et ses nombreuses dérivées ont été développées avec succès dans le monde de l'aérospatial à de nombreux problèmes, comme la poursuite radar, l'estimation de trajectoires d'orbites...

Après la parution des publications fondamentales de *Kalman* et *Bucy* [33] pour la théorie de filtrage linéaire, un nombre très important de travaux fut consacré à ce problème, chaque auteur décrivant son algorithme en exploitant des méthodes comme les *moindres carrées*, le maximum de vraisemblance et d'autres méthodes statistiques classiques. Cependant, ces méthodes statistiques laissent dans l'ombre la structure probabiliste du problème de filtrage, structure qui est fondamentale.

3.4.3. Filtre de Kalman Etendu

L'une des hypothèses de base du *FK* est qu'il est conçu pour évaluer les états d'un système linéaire à partir de mesures qui sont une fonction linéaire des états. Malheureusement, dans beaucoup de situations où nous voudrions utiliser un *FK*, nous avons un modèle de système *NL* et ou une équation de mesure *NL*. Plus précisément, le modèle du système est une fonction *NL* des états et ou les mesures sont des fonctions *NLs* des Etats. Habituellement, les non-linéarités ne s'étendent pas aux perturbations du réseau et du bruit de mesure.

En raison de l'attractivité du *FK*, les concepteurs ont mis au point un ensemble mathématiques à fin d'étendre la théorie du *FK* à des situations où le modèle du système et / ou du modèle de mesure sont des fonctions *NLs* de l'état [18]. Le *FK* qui en résulte est appelé le *FK* étendu.

FK étendu utilise le modèle de système *NL* pour le calcul de l'estimation de l'état prédit $x((k + 1)|k)$, et le modèle de mesure *NL* pour former la mesure prédite, $y((k + 1)|k)$.

3.4.4. Filtre Particulaire

Le *FP* connaît actuellement un intense développement dans de nombreux domaines tels que les sciences de l'ingénieur et les sciences techniques de l'information et de la communication comme la localisation, la robotique, le traitement du signal audio ou encore la communication numérique.

Le *FP* ou méthode de *MCS* est une technique très efficace pour estimer séquentiellement l'état caché d'un système dynamique *NL* [34]. Le calcul de cette estimation dépend de la *PDF a posteriori* de l'état caché sachant les observations.

Cependant, dans le cas d'un système dynamique *NL* cette densité ne peut pas être déterminée analytiquement. Il faut utiliser des approximations numériques. Le *FP* fait approximer la densité *a posteriori* avec des échantillons pondérés, appelés particules, qui évoluent dans le temps selon les lois dynamiques du système considéré. Cette évolution permet l'estimation séquentielle de l'état caché.

3.5. Estimation bayésienne séquentielle

3.5.1. Principe

L'estimation Bayésienne séquentielle fournit un *filtrage Bayésien* séquentiel à partir de deux hypothèses [41],

- L'état suit un processus de *Markov* de premier ordre :

$$p(x_k|x_{0:k-1}) = p(x_k|x_{k-1}) \quad (70)$$

- Les observations sont indépendantes conditionnellement aux états :

$$p(y_k | x_k, y_{k-1}, \dots, y_1) = p(y_k | x_k) \quad (71)$$

Ainsi, nous obtenons la *PDF* pour des observations disponibles en adoptant la méthodologie suivante :

$$p(x_k | y_{1:k}) = \frac{p(y_{1:k} | x_k) p(x_k)}{p(y_{1:k})} = \frac{p(y_k, y_{1:1-k} | x_k) p(x_k)}{p(y_k, y_{1:k-1})} \quad (72)$$

Or :

$$p(y_k, y_{1:1-k} | x_k) = p(y_k | y_{1:1-k}, x_k) \cdot p(y_{1:1-k} | x_k) \quad (73)$$

Donc la relation (72) devient :

$$p(x_k | y_{1:k}) = \frac{p(y_k | y_{1:k-1}, x_k) p(y_{1:k-1} | x_k) p(x_k)}{p(y_k | y_{1:k-1}) p(y_{1:k-1})} \quad (74)$$

On sait que :

$$p(y_{1:1-k} | x_k) = \frac{p(x_k | y_{1:k-1}) p(y_{1:k-1})}{p(x_k)} \quad (75)$$

Alors, la relation (74) devient :

$$p(x_k | y_{1:k}) = \frac{p(y_k | y_{1:k-1}, x_k) p(x_k | y_{1:k-1}) p(y_{1:k-1}) p(x_k)}{p(y_k | y_{1:k-1}) p(y_{1:k-1}) p(x_k)} \quad (76)$$

En utilisant l'indépendance entre les observations conditionnellement aux états et en simplifiant on obtient :

$$p(x_k | y_{1:k}) = \frac{p(y_k | x_k) p(x_k | y_{1:k-1})}{p(y_k | y_{1:k-1})} \quad (77)$$

La densité *a posteriori* (77) est caractérisée par trois termes :

- L'information *a priori* $p(x_k | y_{1:k-1})$ définissant la connaissance du modèle

$$p(x_k | y_{1:k-1}) = \int p(x_k | x_{k-1}) p(x_{k-1} | y_{1:k-1}) dx_{k-1} \quad (78)$$

Dans la relation (78), $p(x_k | x_{k-1})$ désigne une *PDF* représentant la transition entre les états.

- La *vraisemblance* est définie par $p(x_k | y_k)$, la *PDF* associée à la mesure.
- L'*évidence* est définie par

$$p(y_k | y_{1:k-1}) = \int p(x_k | y_k) p(x_k | y_{1:k-1}) dx_k. \quad (79)$$

La *PDF a posteriori* complète $p(x_{0:k} | y_{1:k})$, ou marginale $p(x_k | y_{1:k})$, permet d'estimer la valeur du vecteur caché x_k . Le cadre Bayésien facilite l'estimation de deux grandeurs :

- le *MMSE*
- le *MAP*

Le MMSE est défini par :

$$\hat{x}_{k/k}^{MMSE} = \mathbb{E}[x_k | y_{1:k}] = \int x_k p(x_{0:k} | y_{1:k}) dx_{0:k} = \int x_k p(x_k | y_{1:k}) dx_k \quad (80)$$

Le MAP est donné par :

$$\hat{x}_{k/k}^{MAP} = \operatorname{argmax}_{x_k} [p(x_k | z_{1:k})] \quad (81)$$

La formule (77) fournit l'estimation Bayésienne séquentielle. En l'introduisant dans l'*EI* on obtient l'*EI Bayésien*.

3.5.2. Etapes

Soit un modèle d'observation donné par l'équation de mesure $y_k = h_k(x_k, u_k)$ et un modèle d'évolution donné par $x_k = f_k(x_{k-1}, u_{k-1})$. La fonction h_k est appelée fonction d'observation et la fonction f_k fonction d'évolution. Nous nous plaçons dans un cadre *Bayésien* où l'objectif est de calculer la loi *a posteriori* $p(x_k | y_k)$. On peut ensuite en déduire des estimations ponctuelles de x_k . L'algorithme permettant d'estimer le vecteur d'état x_k récursivement opère en deux étapes [18]: une étape de *prédiction* où l'on prédit la valeur de x_k à partir de x_{k-1} , et une étape de *correction* ou *mise à jour* où l'on corrige la valeur prédit de x_k par utilisation de l'observation reçue, y_k , à l'instant k afin d'obtenir une estimation plus précise (*figure 3-2*).



Figure 3-2 Filtre Bayésien séquentiel.

3.5.2.1. Prédiction

On connaît $p(x_k | x_{k-1})$ donnée par le modèle d'évolution et $p(x_{k-1} | y_{1:k-1})$, la loi *a posteriori* à l'instant $k - 1$, avec $y_{1:k-1}$ l'ensemble des mesures obtenues jusqu'à l'instant $k - 1$. Le but est d'obtenir la *PDF* $p(x_k | y_{1:k-1})$ [35].

Considérons $p(x_k, x_{k-1} | y_{1:k-1})$ comme la densité conjointe (conditionnelle) alors on a [35] :

$$p(x_k | y_{1:k-1}) = \int_{x_{k-1}} p(x_k, x_{k-1} | y_{1:k-1}) p(x_{k-1} | y_{1:k-1}) dx_{k-1} \quad (82)$$

Soit en appliquant la loi de *Bayes* dans (82) ; Comme x_{k-1} contiennent déjà les informations apportées par $y_{1:k-1}$, il vient [18]:

$$p(x_k | x_{k-1}, y_{1:k-1}) = p(x_k | x_{k-1}) \quad (83)$$

D'où

$$p(x_k | y_{1:k-1}) = \int p(x_k | x_{k-1}) p(x_{k-1} | y_{1:k-1}) dx_{k-1} \quad (84)$$

Si on approche :

$$p(x_k | y_{1:k-1}) \approx \sum_{i=1}^N w_{k-1}^i \delta(x_{k-1} = x_{k-1}^i) \quad (85)$$

On a alors :

$$p(x_k | y_{1:k-1}) \approx \sum_{i=1}^N w_{k-1}^i p(x_k | x_{k-1}^i) \quad (86)$$

La loi de densité prédite ($x_k | y_{1:k-1}$) s'approche par

$$p(x_k | y_{1:k-1}) \approx \sum_{i=1}^N w_{k|k-1}^i \delta(x_k = X_{k|k-1}^i) \quad (87)$$

Avec $x_{k|k-1}^i$ qui sont obtenues par des réalisations indépendantes de la loi de transition

$p(x_k | x_{k-1}^i)$ Et les poids $w_{k|k-1}^i$ ne changent pas $w_{k|k-1}^i = w_{k-1}^i$

Les w_{k-1}^i sont des poids positifs de somme 1, $x_{k-1}^1, \dots, x_{k-1}^i$ sont des vecteurs de \mathbb{R}^d appelés "particules".

3.5.2.2. Mise à jour

A l'étape de correction, on passe de la loi de densité prédite $p(x_k | y_{1:k-1})$ à la loi de *DC* $p(x_k | y_{1:k})$ grâce à la vraisemblance $g(Y_k - H_k(x_{k|k-1}^i))$ donnée par le modèle du bruit de mesure.

$$\begin{aligned} p(x_k | y_{1:k}) &= p(x_k | y_{1:k-1} y_k) \\ &= \frac{p(x_k, y_k | y_{1:k-1})}{p(y_k | y_{1:k-1})} = \frac{p(y_k | x_k, y_{1:k-1}) p(x_k | y_{1:k-1})}{\int p(x_k, y_k | y_{1:k-1}) dx_k} \end{aligned} \quad (88)$$

Comme l'intégrale dans le dénominateur de (88) est indépendante de x_k , on peut la considérer constante. De plus, $p(y_k | x_k, y_{1:k-1}) = p(y_k | x_k)$ on obtient [18]:

$$p(x_k | y_{1:k}) = \frac{p(y_k | x_k) p(x_k | y_{1:k-1})}{cste} \quad (89)$$

La loi de *DC* est alors approximée par une autre distribution de *Dirac* de support $x_k^i = x_{k|k-1}^i$: [18]

$$p(x_k | y_{1:k}) \approx \sum_{i=1}^N w_k^i \delta(x_k = x_k^i) \quad (90)$$

et de poids :

$$w_k^i = \frac{w_{k/k-1}^i g(y_k - H_k(x_{k|k-1}^i))}{\sum_{j=1}^N w_{k|k}^i g(y_k - H_k(x_{k|k-1}^j))} \quad (91)$$

4.1. Principe

Le principe de *MC* admet à exploiter un ensemble d'échantillons discrets obtenus par simulation pour approcher une *PDF* $p(x)$, dont on ne connaît pas l'expression analytique.

Le problème de calculer des intégrales de grandes dimensions pourront être résolus grâce à cet ensemble. Soit $\{x^{(i)}\}_{i=1, \dots, N}$, un ensemble de réalisations indépendantes et identiquement distribuées selon la densité $p(x)$ définie sur un espace x . Cette densité peut être approximée par la loi discrète $p_N(x)$ [31] :

$$p_N(x) = \frac{1}{N} \sum_{i=1}^N \delta_{x^{(i)}}(x) \quad (92)$$

Où $\delta_{x^{(i)}}$ désigne la mesure de Dirac prise en la valeur $x^{(i)}$.

A partir de cette densité empirique, il est possible d'approcher des intégrales suivant :

$$I_N(\phi(x)) = E_{p(x)}[\phi(x)] \quad (93)$$

par des sommes pondérées $I_N(\phi(x))$, telles que :

$$I_N(\phi(x)) = \frac{1}{N} \sum_{i=1}^N \phi(x^i), \text{ si } N \rightarrow \infty \quad I_N(\phi(x)) = \int \phi(x)p(x)dx \quad (94)$$

L'estimateur $I_N(\phi(x))$ est non-biaisé et la convergence presque sûre est démontrée par la loi forte des grands nombres.

Si de plus la variance de $\phi(x)$ notée est finie, δ_ϕ^2 alors la variance de l'estimateur $I_N(\phi(x))$ est égale à δ_ϕ^2/N , et le théorème central limite nous garantit la convergence en loi de l'erreur d'estimation :

$$\sqrt{N} \left(I_N(\phi(x)) - I_N(\phi(x)) \right), \text{ si } N \rightarrow \infty \quad \mathcal{N}(0, \delta_\phi^2) \quad (95)$$

Cependant il sera ainsi possible d'estimer toute quantité la forme $E_{p(x)}[\phi(x)]$ à partir des échantillons $\{x^{(i)}\}$ où $i = 1, \dots, N$, la vitesse de convergence de l'estimateur ne dépend que de N . Contrairement aux méthodes d'intégration numériques déterministes, la vitesse de convergence ne dépend pas de la taille de l'espace \mathcal{X} .

Quand la loi objectif $p(x)$, n'est pas connue, et qu'il n'est pas possible d'en tirer des échantillons, d'autre alternative sont proposés pour apporter des solutions en utilisant les techniques de *MC* [31]. L'algorithme d'acceptation ou rejet est décrit ci-dessus ;

4.1.1. Algorithme d'Acceptation ou Rejet

Comme son nom l'indique, l'objectif de l'algorithme d'acceptation/rejet consiste à accepter ou à rejeter les particules générées [43]. Afin d'expliquer son fonctionnement, considérons que $p(x)$ représente la densité à partir de laquelle on souhaite générer x . Cependant, supposons que $p(x)$ n'est connue qu'à une constante γ_1 près et qu'il existe une densité de probabilité $p_1(x)$ connue telle que $p(x) = \gamma_1 p_1(x)$. Considérons qu'il soit possible de générer x à partir de $q(x)$ mais que $q(x)$ n'est aussi connue qu'à une constante γ_2 près. Cela signifie qu'il existe une densité de probabilité $q_1(x)$ telle que : $q(x) = \gamma_2 q_1(x)$ Donc le rapport $\frac{p(x)}{q(x)}$ devient égal à γ

$\frac{p_1(x)}{q_1(x)}$ avec $\gamma = \frac{\gamma_1}{\gamma_2}$. Soit $h(x) = \frac{p(x)}{q(x)}$ le quotient des deux lois, lorsqu'il est possible de bor-

ner $h(x)$ en tout x par la constante c_1 tel que $\frac{p_1(x)}{q_1(x)} \leq c_1$ ce qui est équivalent à $\frac{p(x)}{q(x)} \leq c$, alors

la procédure d'échantillonnage acceptation/rejet est donnée par :

- Générer une variable aléatoire uniforme $u \sim [0, 1]$.
- Générer une variable aléatoire $x \sim q(x)$.
- Si $u \leq \frac{p(x)}{cq(x)}$ retourner x sinon aller à la première étape.

L'algorithme d'acceptation/rejet [41] montre que la probabilité d'acceptation d'une *v.a.* générée est inversement proportionnelle à la constante c . Une faible valeur de c conduit à l'acceptation des échantillons peu représentatifs en raison du taux de rejet faible.

Tandis qu'une très forte valeur de c rend l'algorithme inefficace compte tenu du taux d'acceptation faible. L'algorithme acceptation/rejet produit des résultats satisfaisants lorsque la borne supérieure de la distribution est connue. Les particules générées au-dessous de la courbe $cq(x)$ seront rejetées lorsqu'elles se situent entre $p(x)$ et $cq(x)$. En revanche, lorsque les particules se situent en dessous de $p(x)$, elles seront acceptées.

Les méthodes de générations des lois [10] introduites précédemment sont inexploitable en temps-réel en raison de l'absence de récursivité. Leurs évolutions vers des versions séquentielles passent par le filtrage *Bayésien*.

4.2. Intégration Monte-Carlo

Pour approximer les intégrales dans la solution *Bayésienne*, des méthodes appelées *MC* sont utilisés. Les méthodes *MC*, basées sur la simulation, sont une alternative aux méthodes déterministes. Ils visent à approximer des intégrales [20].

$$I = \int g(s)p(s)ds \quad (96)$$

où $s \in \mathbb{R}^n$ et $p(s)$ sont supposés être interprétés comme la densité de probabilité sous-jacente telle que

$$\int p(s)ds = 1 ; p(s) > 0, \forall s. \quad (97)$$

S'il est possible de prélever N échantillons $\{s^i\}_{i=1}^N$ selon $p(s)$, alors, l'intégrale peut alors être approximé par la somme :

$$\hat{I} = \frac{1}{N} \sum_{i=1}^N g(s^i) \quad (98)$$

Les échantillons $\{s^i\}_{i=1}^N$ proviennent automatiquement des régions de l'espace d'état qui sont important pour le résultat de l'intégration.

L'estimation dans le cadre de l'inférence *Bayésienne* nécessite le calcul d'intégrales de type [21] :

$$E [g(x_{0:k})] = \int g(x_{0:k}) p(x_{0:k}|y_{1:k}) dx_{0:k} \quad (99)$$

Dans le cas de *PDF* complexes, le calcul analytique de cette intégrale n'est pas envisageable, et les techniques classiques d'intégration, comme la discrétisation déterministe sur une grille, sont inappropriées car elles sont imprécises et trop coûteuses en temps de calcul. En effet, le nombre d'éléments augmente de façon exponentielle avec la dimension du vecteur à estimer, et la taille de la grille peut ne pas être adaptée à la forme de la densité *a posteriori*.

Le principe de l'*Intégration Monte Carlo (IMC)* consiste à générer N réalisations de *v.a.* $x_{0:k}$ distribuées selon $p(x_{0:k}|y_{1:k})$, et notées $\tilde{x}_{0:k}^{(i)}$, soit :

$$\tilde{x}_{0:k}^{(i)} \sim p(x_{0:k}|y_{1:k}) \text{ pour } i = 1, \dots, N \quad (100)$$

L'approximation de *MC* d'une intégrale est donnée par :

$$E [g (x_{0:k})] \approx \frac{1}{N} \sum_{i=1}^N g(\tilde{x}_{0:k}^{(i)}) \quad (101)$$

On obtient ainsi beaucoup de réalisations dans les régions de forte probabilité et peu (ou pas) dans les régions de faible probabilité. L'échantillonnage est aléatoire et adapté à la forme de la densité *a posteriori*.

L'*IMC* est fondée sur l'approximation de la distribution $p(dx_{0:k}|y_{1:k})$ par la distribution empirique:

$$p_N(dx_{0:k}|y_{1:k}) = \frac{1}{N} \sum_{i=1}^N \delta_{\tilde{x}_{0:k}^{(i)}}(dx_{0:k}) \quad (102)$$

où $\delta(\cdot)$ est la fonction de Dirac au point x . La *PDF a posteriori* est ainsi approximée par un ensemble de points $\tilde{x}_{0:k}^{(i)}$ ayant chacun un poids $\frac{1}{N}$.

Pratiquement, il est souvent impossible de générer directement des échantillons distribués selon la *PDF a posteriori* $p(x_{0:k}|y_{1:k})$ [21] qui peut être sous forme complexe, non-standard (*NG*, gamma, uniforme...) ou seulement connue à une constante près. Les méthodes de *MCSs*, à l'aide de l'*EI*, permettent de générer les échantillons en ligne de façon récurrente.

4.3. Echantillonnage de Monte Carlo

Considérons pour une toute fonction f bornée de $R^d \rightarrow R$ et une densité de probabilité $p(x)$ connue, l'espérance de $f(x)$ est donnée par [10] :

$$\mathbb{E}_p[f(x)] = \int f(x)p(x)dx \quad (103)$$

Le but consiste à déterminer l'intégrale donc à calculer l'espérance de $f(x)$. Cependant, la complexité de la *PDF* $p(x)$ rend difficile l'évaluation analytique de cette intégrale. Cela conduit à recourir à l'échantillonnage de *MC*. Soit x une variable aléatoire dans \mathbb{R}^d , distribuée selon la *PDF* $p(x)$, et soient $(x^{(i)})$, $i = 1, \dots, N_p$ des variables aléatoires *Indépendant et Identiquement Distribués (i.i.d.)* sur \mathbb{R}^d et disposant la même loi. Le principe de l'échantillonnage de *MC* consiste à générer N_p réalisations de la variable aléatoire x distribuée selon la *PDF* $p(x)$ notées :

$$(x^{(i)})_{i=1,\dots,N_p} \sim p(x) \quad (104)$$

et ceci pour remplacer la distribution continue $p(x)$ par une distribution empirique désignée par $p(x)$ comme suit :

$$P(x) = \frac{1}{N_p} \sum_{i=1}^{N_p} \delta(x - x^{(i)}) \quad (105)$$

Pour que la distribution empirique $P(x)$ remplace correctement la distribution continue $p(x)$ il faut que le nombre de réalisations N_p soit élevé. En utilisant (105), l'intégrale (103) devient [44]:

$$\begin{aligned} \mathbb{E}_p[f(x)] &= \int f(x)P(x)dx \\ &= \int f(x) \frac{1}{N_p} \sum_{i=1}^{N_p} \delta(x - x^{(i)}) dx \\ &= \frac{1}{N_p} \sum_{i=1}^{N_p} \int f(x) \delta(x - x^{(i)}) dx \\ &= \frac{1}{N_p} \sum_{i=1}^{N_p} f(x^{(i)}) = \hat{f}_{N_p}(x) \end{aligned} \quad (106)$$

La loi des grands nombres indique que lorsque le nombre d'échantillons N_p est important la moyenne empirique tend vers l'espérance *presque sûrement* [42]:

$$\hat{f}_{N_p}(x) = \frac{1}{N_p} \sum_{i=1}^{N_p} f(x^{(i)}) \xrightarrow{N_p \rightarrow +\infty} \mathbb{E}_p[f(x)] \quad (107)$$

On montre que [41] :

$$\begin{aligned} \mathbb{E}[\hat{f}_{N_p}] &= \mathbb{E}[f], \text{Var}[\hat{f}_{N_p}] \approx \frac{1}{N_p} \text{Var}[f] = \frac{\sigma^2}{N_p}, \sigma^2 \\ &= \int [f(x) - E[f(x)]]^2 p(x) dx \end{aligned} \quad (108)$$

ou : σ^2 désigne la *variance* de f .

4.3.1. Échantillonnage d'Importance

L'*EI* est une étape nécessaire dans le cas où la loi est difficilement simulable voire impossible. Une autre loi connue et plus facile à manier, intervient dans la simulation. Le choix de la loi de remplacement, généralement, est justifié par le fait d'avoir une variance plus petite [16-20].

L'objectif de l'*EI*, consiste à proposer une distribution, désignée par distribution proposée (*proposal distribution*) $q(x)$ qui entraîne la réalisation de valeurs influentes prises par une *v.a.* dans le but d'approximer la densité de probabilité cible (*target distribution*) $p(x)$. Les raisons de la substitution de $p(x)$ par $q(x)$ en vue de générer des variables aléatoires ayant des valeurs influentes sont souvent dictées par la difficulté de connaître avec précision $p(x)$. L'obtention des valeurs influentes des variables aléatoires générées à partir de $q(x)$ impose de construire cette

dernière de façon à ce que son support couvre celui de $p(x)$. Pour obtenir une somme pondérée qui approxime (103), réécrivons (103) sous forme [10-20] :

$$\begin{aligned}\mathbb{E}_p[f(x)] &= \int f(x)p(x)dx \\ &= \int f(x) \frac{p(x)}{q(x)} q(x)dx \\ &= \int f(x)w(x)q(x) dx\end{aligned}\tag{109}$$

avec : $w(x) = \frac{p(x)}{q(x)}$

En générant, N_p particules indépendantes et distribuées selon la densité de probabilité $q(x)$ notées par: $(x^{(i)})_{i=1,\dots,N_p} \sim q(x)$ selon le même cheminement de calcul de (109) on aura:

$$\hat{f}_{N_p} = \frac{1}{N_p} \sum_{i=1}^{N_p} w(x^{(i)}) f(x^{(i)})\tag{110}$$

où $w(x^{(i)}) = \frac{p(x^{(i)})}{q(x^{(i)})}$ est appelé le poids d'importance. Pour s'assurer que la somme des poids vaut 1, $\sum_{i=1}^{N_p} w(x^{(i)}) = 1$, le poids d'importance est normalisé :

$$\hat{f}_{N_p} = \frac{\frac{1}{N_p} \sum_{i=1}^{N_p} w(x^{(i)}) f(x^{(i)})}{\frac{1}{N_p} \sum_{j=1}^{N_p} w(x^{(j)})} = \sum_{i=1}^{N_p} \tilde{w}(x^{(i)}) f(x^{(i)})\tag{111}$$

La relation précédente fournit une estimation de l'espérance de f , avec:

$$\tilde{w}(x^{(i)}) = \frac{w(x^{(i)})}{\sum_{i=1}^{N_p} w(x^{(i)})}\tag{112}$$

On montre que la *variance* de cette estimation est donnée par :

$$\text{var}[\hat{f}] = \frac{1}{N_p} \text{var}[f(x)w(x)] = \frac{1}{N_p} \int \left[\frac{(f(x)p(x))^2}{q(x)} \right] dx - \frac{(\mathbb{E}_p[f(x)])^2}{N_p}\tag{113}$$

L'équation (112) montre que la réduction de la variance de l'estimation \hat{f}_{N_p} s'effectue lorsque la distribution proposée $q(x)$ est choisie de façon à coïncider avec $p(x)$. La qualité de l'estimation est conditionnée par l'écart qui existe entre les deux distributions $p(x)$ et $q(x)$.

Lorsque cet écart est important les poids deviennent peu représentatifs et plusieurs particules auront une contribution négligeable dans l'approximation de la *PDF* $p(x)$ [10-42].

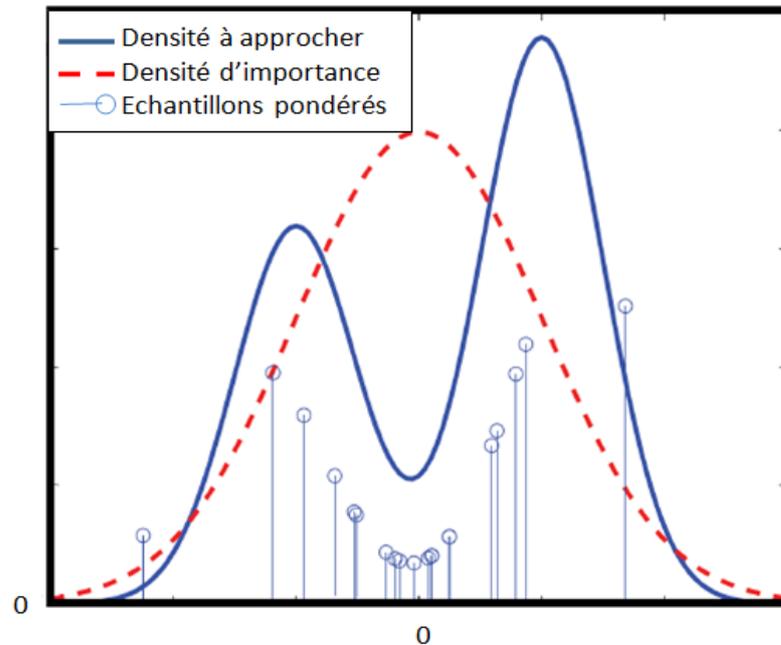


Figure 4-1 : Échantillonnage d'importance

4.4. Simulation de Monte-Carlo

L'estimation récursive donnée ci-dessus conduit à des expressions impossibles à évaluer analytiquement à moins que des hypothèses solides ne soient formulées. Dans un cadre plus général, ce problème est surmonté en utilisant des méthodes de *MC*, où N échantillons aléatoires indépendants et *i.i.d.* $\{s_t^i; i = 1 : N\}$.

Une *PDF* simulée est donnée par l'expression suivante [20] :

$$\tilde{p}(s_t | e_{1:t}) = \frac{1}{N} \sum_{i=1}^N \delta(s_t - s_t^i) \quad (114)$$

Il existe également plusieurs inconvénients qui empêchent l'utilisation de cette méthode $P(s_t | e_{1:t})$ est généralement complexe, multi varié et n'est connu que jusqu'à un constante de proportionnalité.

4.5. Utilisation

Evaluer une valeur numérique par des procédés aléatoires peut conduire à utiliser les méthodes de *MC*. Elles exploitent des échantillons simulés pour approximer, par exemple, une fonction densité de probabilité [41]. Lorsque les densités de probabilités ne peuvent être approximées par des méthodes analytiques, l'utilisation des techniques de *MC* peut se révéler particulièrement efficace. Les méthodes de *MC* se répartissent en trois catégories [10]:

- L'échantillonnage de *MC* : dédié à l'estimation.
- Le calcul de *MC* : pour la conception des générateurs de *v.a.*
- L'optimisation de *MC* : pour l'optimisation des problèmes mathématiques.

5.1. Introduction

L'objectif d'un *FP* est d'estimer *la densité postérieure* des variables d'état compte tenu des variables d'observation. Le *FP* est conçu pour un modèle de *MC*, où le système se compose de variables cachées et observables. Les variables observables (processus d'observation) sont liées aux variables cachées (processus d'état) par une forme fonctionnelle connue [46]. De même, le système dynamique décrivant l'évolution des variables d'état est également connu de façon probabiliste.

Un *FP* générique estime *la distribution postérieure* des états cachés en utilisant le procédé de mesure d'observation. Considérons un espace d'état illustré par le diagramme ci-dessus [3] :

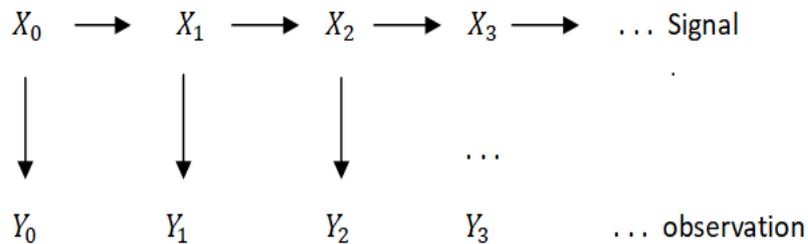


Figure 5.1 : filtre à particule

Le problème de filtrage consiste à estimer séquentiellement les valeurs des états cachés X_k , compte tenu des valeurs du processus d'observation Y_0, \dots, Y_k , à tout moment (étape) k [3].

Toutes les estimations *Bayésiennes* X_k de *la densité postérieure* $p(x_k | Y_0, \dots, Y_k)$.

5.2. Définition

Le *FP* est une méthode numérique de filtrage adaptée aux cas de modèles *NLs* munis des bruits *NGs*.

En utilisant un *FP* ou *MCS* nous visons à suivre diverses distributions qui surviennent dans les modèles dynamiques d'espace d'état. Alors, l'objectif du *FP* est l'estimation séquentiel des distributions des états, y compris la distribution du filtrage $p(x_t | y_{1:t})$ la distribution prédictive $p(x_t | y_{1:t-1})$, ou la distribution de lissage $p(x_t | y_{1:T})$, où $t < T$. Le suivi est effectué en explo-

rant l'espace des états avec des échantillons générés aléatoirement (également appelés particules).

D'une autre façon, les *PF* effectuent le suivi de x_n en mettant à jour une mesure aléatoire $\{x_n^{(m)}, w_n^{(m)}\}_{m=1}^M$ composé de M particules $x_n^{(m)}$ et de leurs poids $w_n^{(m)}$ définis à l'instant n , récursivement dans le temps [46-36-7]. Les distributions d'intérêt sont approximées par les particules générées ainsi que les poids attribués aux particules.

5.3. Représentation

On peut présenter un *FP* classique de deux façons différentes [23] :

a) Première approche du *FP* : La première façon est basée sur l'approximation d'une densité par une combinaison convexe de Dirac. On suppose qu'à l'instant $k - 1$, on a une approximation de la distribution conditionnelle $p(x_{k-1}|y_{1:k-1})$, de l'état x_{k-1} sachant les mesures $y_{1:k-1} = y_1, \dots, y_{k-1}$ de la forme :

$$p(x_{k-1}|y_{1:k-1}) \approx \sum_{i=1}^N w_{k-1}^i \delta(x_{k-1} = x_{k-1}^i) \quad (115)$$

où w_k^i sont des poids positifs de somme 1, $x_{k-1}^1, \dots, x_{k-1}^N$ sont des vecteurs de \mathbb{R}_d appelés particules.

• **Etape de prédiction :** Considérons $p(x_k, x_{k-1}|y_{1:k-1})$ comme la densité Conjointe (conditionnelle), alors on a :

$$p(x_k|y_{1:k-1}) = \int p(x_k, x_{k-1}|y_{1:k-1}) dx_{k-1} \quad (116)$$

Selon la définition de la probabilité conditionnelle, on a :

$$p(x_k, x_{k-1}|y_{1:k-1}) = \int p(x_k|x_{k-1}, y_{1:k-1}) p(x_{k-1}|y_{1:k-1}) dx_{k-1} \quad (117)$$

Par hypothèse, on a :

$$p(x_k|x_{k-1}, y_{1:k-1}) = p(x_k|x_{k-1}) \quad (118)$$

d'où :

$$p(x_k|y_{1:k-1}) = \int p(x_k|x_{k-1}) p(x_{k-1}|y_{1:k-1}) dx_{k-1} \quad (119)$$

Si on fait l'approximation :

$$p(x_{k-1}|y_{1:k-1}) \approx \sum_{i=1}^N w_{k-1}^i \delta(x_{k-1} = x_{k-1}^i) \quad (120)$$

on aura alors :

$$p(x_k | y_{1:k-1}) = \sum_{i=1}^N w_{k-1}^i \delta(x_k = x_{k|k-1}^i) \quad (121)$$

avec $x_{k|k-1}^i$ sont obtenues par des réalisations indépendantes de la loi de transition $p(x_k | x_{k|k-1}^i)$ et les poids w_{k-1}^i ne changent pas $w_{k|k-1}^i = w_{k-1}^i$

• *Etape de correction* à l'étape de correction, on passe de la loi de densité prédite $p(x_k | y_{1:k-1})$ à la loi de densité conditionnelle $p(x_k | y_{1:k})$ grâce à la vraisemblance $g(y_k - H_k(x_{k|k-1}^i))$ donnée par le modèle du bruit de mesure. La loi de densité conditionnelle est alors approximée par une autre distribution de Dirac de support $x_k^i = x_{k|k-1}^i$;

$$p(x_k | y_{1:k}) \approx \sum_{i=1}^N w_k^i \delta(x_k = x_k^i) \quad (122)$$

et de poids ;

$$w_k^i = \frac{w_{k|k-1}^i g(y_k - H_k(x_{k|k-1}^i))}{\sum_{j=1}^N w_{k|k-1}^j g(y_k - H_k(x_{k|k-1}^j))} \quad (123)$$

b) *Seconde approche du FP* : La seconde façon de présenter le FP est basée sur l'échantillonnage pondéré séquentiel. On notera $x_k = (x_0, \dots, x_k)^T$, la trajectoire de l'état de l'instant initial x_0 à l'instant final x_k . Soit à calculer ;

$$\begin{aligned} \mathbb{E}[f(x_{0:k} | y_{1:k})] &= \int f(x_{0:k}) p(x_{0:k} | y_{1:k}) dx_{0:k} \\ &= \int f(x_{0:k}) \frac{p(x_{0:k} | y_{1:k})}{q(x_{0:k} | y_{1:k})} q(x_{0:k} | y_{1:k}) dx_{0:k} \end{aligned} \quad (124)$$

D'après la méthode d'échantillonnage pondéré séquentiel ;

$$\omega_k = \frac{p(x_{0:k} | y_{1:k})}{q(x_{0:k} | y_{1:k})} \quad (125)$$

On a alors :

$$\mathbb{E}[f(x_{0:k} | y_{1:k})] = \sum_{i=1}^N f(x_{0:k}^i) \omega(x_{0:k}^i) / \sum_{j=1}^N \omega(x_{0:k}^j) \quad (126)$$

On a :

$$p(x_{0:k} | y_{1:k}) = \frac{p(x_{0:k-1} | y_{1:k-1}) p(x_k | x_{0:k-1}, y_{1:k-1}) p(y_k | x_k)}{p(y_k | y_{1:k-1})} \quad (127)$$

vu le caractère *Markovien* de x_k , on peut écrire :

$$p(x_k | x_{0:k-1}, y_{1:k-1}) = p(x_k | x_{k-1}) \quad (128)$$

D'où :

$$p(x_{0:k} | y_{1:k}) = \frac{p(x_{0:k-1} | y_{1:k-1}) p(x_k | x_{k-1}) p(y_k | x_k)}{p(y_k | y_{1:k-1})} \quad (129)$$

Afin d'obtenir un algorithme récursif, la fonction d'importance q doit vérifier la relation suivante :

$$q(x_{0:k} | y_{1:k}) = q(x_{0:k-1} | y_{1:k-1}) q(x_k | x_{0:k-1}, y_{1:k}) \quad (130)$$

Si on remplace la relation (125), par (129) et (130), on aura la relation récurrente suivante :

$$\omega_k \propto \omega_{k-1} \frac{p(x_k | x_{0:k-1}) p(y_k | x_k)}{q(x_k | x_{0:k-1}, y_{1:k})} \quad (131)$$

Qui réactualise les poids ω_k en fonction de $q(x_k | x_{0:k-1}, y_{1:k})$.

Si g désigne la densité de la loi du bruit de mesure, alors ;

$$p(y_k | x_k) = g(y_k - H_k(x_k | x_{k-1})) \quad (132)$$

et on a :

$$\omega_k^i \propto \omega_{k-1}^i \frac{g(y_k - H_k(x_k^i | x_{k-1}^i)) p(x_k^i | x_{k-1}^i)}{q(x_k^i | x_{0:k-1}^i, y_{1:k})} \quad (133)$$

L'estimée de l'état \hat{x}_k est alors égale à :

$$\hat{x}_k \approx \sum_{i=1}^N \omega_k^i x_k^i \quad (134)$$

et la matrice de covariance est estimée par :

$$P_k \approx \sum_{i=1}^N \omega_k^i (x_k^i - \hat{x}_k) (x_k^i - \hat{x}_k)^T \quad (135)$$

5.3.1. Particules

Présentation de chaque particule à l'instant k [25]

Instant $k = 0$: $\{x_0^{(i)}, w_0^{(i)}\}$

où (i) est indice de la particule $\{1, \dots, N\}$, $x_0^{(i)}$ est un support et $w_0^{(i)}$ est le poids.

Instant k : $\{x_{0 \rightarrow k}^{(i)}, w_k^{(i)}\}$

$x_{0 \rightarrow k}^{(i)}$ Support = trajectoire testée

$w_k^{(i)}$ le poids = crédibilité de la trajectoire testée

(Trajectoire du vecteur)

$$x_{0:k}^{(i)} = (x_0^{(i)}, x_1^{(i)}, \dots, x_k^{(i)})$$

Qu'est-ce qu'un « Dirac situé en une trajectoire » ?

$$\delta(x_{0 \rightarrow k} - x_{0 \rightarrow k}^{(i)}) = \delta(x_0 - x_0^{(i)}) \delta(x_1 - x_1^{(i)}) \dots \delta(x_k - x_k^{(i)}) \quad (136)$$

Cette fonction de plusieurs variables x_0, \dots, x_k :

- vaut ∞ si : $x_0 = x_0^{(i)}, x_1 = x_1^{(i)}, \dots, x_k = x_k^{(i)}$.
- vaut 0 si autrement.

On a :

$$\int_{x_0} \dots \int_{x_k} \delta(x_{0:k} - x_{0:k}^{(i)}) dx_0 \dots dx_k = 1 \quad (137)$$

$$\int_{x_0} \dots \int_{x_{k-1}} \delta(x_{0:k} - x_{0:k}^{(i)}) dx_0 \dots dx_{k-1} = \delta(x_k - x_k^{(i)})$$

$$p(x_{0 \rightarrow k} / y_{0 \rightarrow k}) \approx \sum_{i=1}^N w_k^{(i)} \cdot \delta(x_{0 \rightarrow k} - x_{0 \rightarrow k}^{(i)})$$

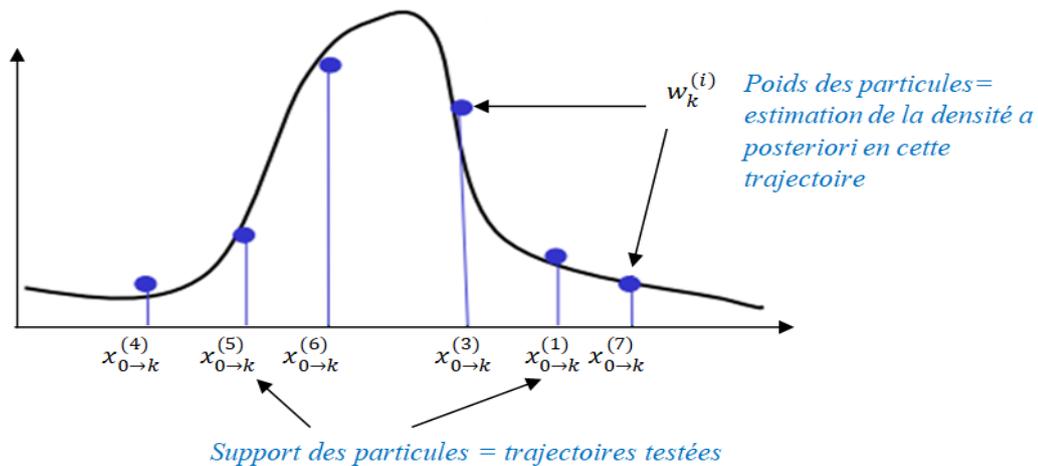


Figure 5-1 : Trajectoire des poids des particules

5.3.2. Poids

Dans le cas de l'échantillonnage par importance, tous les poids d'importance doivent être recalculés lorsqu'une nouvelle mesure devient disponible. Soit $x_k = \{x^j\}_{j=1}^N$ représentent tous les états cibles jusqu'à l'instant k . $p(x_k | y_k)$ est alors la densité postérieure jointe au temps k , qui peut être approximé par [20] :

$$p(x_k | y_k) = \sum_{i=1}^N w_k^i (x_k - x_k^i) \quad (138)$$

où les poids normalisés sont choisis en fonction de l'importance de l'échantillonnage et les échantillons sont extrais à partir de la densité d'importance $q(x_k|y_k)$, les poids sont données par:

$$w_k^i \propto \frac{p(x_k^i|y_k)}{q(x_k^i|y_k)} \quad (139)$$

Si la densité d'importance est supposée être sous la forme :

$$\pi(x_k|y_k) = \pi(x_k|x_{k-1}, y_k) \pi(x_{k-1}|y_{k-1}) \quad (140)$$

il est possible d'évaluer l'importance des poids de manière récursive l'orsque de nouvelles mesures deviennent disponibles. Pour dériver l'équation de mise à jour des poids, la *PDF* $p(x_k|y_k)$ est exprimée en termes de $p(x_{k-1})$, $p(y_k|x_k)$, et $p(x_k|y_{k-1})$:

$$\begin{aligned} p(x_k|y_k) &= \frac{p(x_k|x_{k-1}, y_k) P(x_k|y_{k-1})}{P(y_k|y_{k-1})} \\ &= \frac{p(y_k|x_k, y_{k-1}) p(x_k|x_{k-1}, y_{k-1}) p(x_{k-1}|y_{k-1})}{P(y_k|y_{k-1})} \\ &= \frac{p(y_k|x_k) p(x_k|x_{k-1})}{P(y_k|y_{k-1})} p(x_{k-1}|y_{k-1}) \\ &= \alpha p(y_k|x_k) P(x_k|x_{k-1}) p(x_{k-1}|y_{k-1}) \end{aligned} \quad (141)$$

En insérant $\alpha p(y_k|x_k) P(x_k|x_{k-1}) p(x_{k-1}|y_{k-1})$ et $\pi(x_k|y_k) = \pi(x_k|x_{k-1}, y_k) \pi(x_{k-1}|y_{k-1})$

dans $w_k^i \propto \frac{p(x_k^i|y_k)}{\pi(x_k^i|y_k)}$, la mise à jour récursive des poids peut être présentée par :

$$w_k^i \propto w_{k-1}^i \frac{p(y_k|x_k^i) p(x_k^i|x_{k-1}^i)}{\pi(x_k^i|x_{k-1}^i, y_k)} \quad (142)$$

La densité filtrée postérieure peut être, alors, approximée par :

$$p(x_k|y_k) \approx \sum_{i=1}^N w_k^i \delta(x_k - x_k^i) \quad (143)$$

5.4. Etapes

Il existe de nombreuses méthodes de *FP*, et presque toutes sont basé sur trois opérations : (a) génération (propagation) des particules, (b) calcul des poids, et (c) ré-échantillonnage comme est illustré sur la *figure* ci-dessous.

La propagation des particules et le calcul du poids se résument à la génération de particules et à l'affectation du poids, tandis que le ré-échantillonnage remplace un ensemble de particules et leurs poids par un autre ensemble.

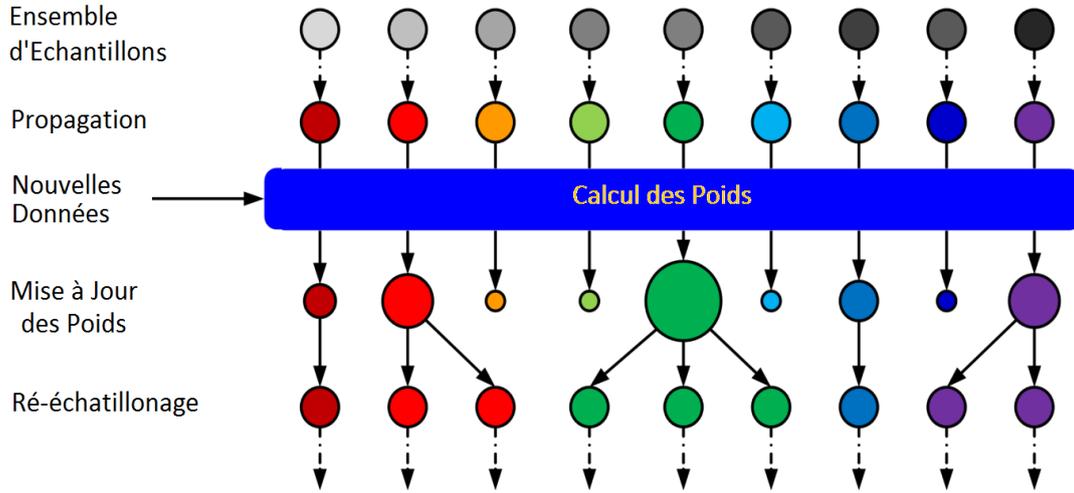


Figure 5-2 : Etapes du FP.

5.4.1. Algorithme de Base

Soit une distribution de proposition $q(x_{k+1}|x_{k:1}, y_{k+1})$, une stratégie de ré-échantillonnage et un nombre de particules N [4].

- **Initialisation** : Générer $x_1^i \sim p_{x_0}$, $i = 1, \dots, N$ et soit $w_{1|0}^i = 1/N$.
- **Itération** : Pour $k = 1, 2, \dots$
 - Mise à jour de la mesure : Pour $i = 1, 2, \dots, N$,

$$w_{k|k}^i = \frac{1}{c_k} w_{k|k-1}^i p(y_k | x_k^i) \quad (144)$$

où le poids de normalisation est donné par

$$c_k = \sum_{i=1}^N w_{k|k-1}^i p(y_k | x_k^i) \quad (145)$$

- **Estimation** : La densité de filtrage est approximée par

$$\hat{p}(x_{k:1} | y_{1:k}) = \sum_{i=1}^N w_{k|k}^i \delta(x_{1:k} - x_{1:k}^i) \quad (146)$$

et la moyenne est approximée par

$$\hat{x}_{k:1} \approx \sum_{i=1}^N w_{k|k}^i x_{1:k}^i \quad (147)$$

- **Ré-échantillonnage** : éventuellement à chaque fois, prélever N échantillons avec remplacement de l'ensemble $\{x_{1:k}^i\}_{i=1}^N$ où la probabilité de prélever l'échantillon i est $w_{k|k}^i = 1/N$.
- **Mise à jour du temps** : générer des prédictions en fonction de la distribution de la proposition :

$$x_{k+1}^i \sim q(x_{k+1}|x_k^i, y_{k+1}) \quad (148)$$

Et compenser le poids d'importance ;

$$w_{K+1|k}^i = w_{K|k}^i \frac{p(x_{k+1}^i|x_k^i)}{q(x_{k+1}^i|x_k^i, y_{k+1})} \quad (149)$$

5.4.2. Calcul des Poids des Particules

La seconde étape de base des *FPs* est le calcul des poids des particules. Pour avoir une inférence à partir des particules générées, la théorie montre que ces particules générées à partir de $q(x_t)$, différent de $p(x_t|y_{1:t})$, doivent être liées à des poids, qui peuvent être calculés récursivement selon :

$$w_t^{(m)} \propto w_{t-1}^{(m)} \frac{q(y_t|x_t^{(m)})p(x_t^{(m)}|x_{t-1}^{(m)})}{q(x_t^{(m)})} \quad (150)$$

Idéalement, les poids des particules sont tous les mêmes. D'autre part, il est indésirable si tous les particules ont des poids nuls ou peut de particules ont de grands poids et les reste des particules sont négligeables. Ce phénomène est connu par l'approvisionnement est d'habitude nommé dégénérescence.

5.4.3. Ré-échantillonnage

L'étape de ré-échantillonnage est une opération cruciale et coûteuse en calcul dans un *FP*, car avec le temps, un petit nombre de poids dominant les poids restants, ce qui conduit à une mauvaise approximation de la densité postérieure et par conséquent à des mauvaises estimations [36].

Sans ré-échantillonnage, un *PF* produira rapidement un ensemble dégénéré de particules, c'est-à-dire un ensemble dans lequel quelques particules dominant le reste des particules avec leur poids.

Avec l'étape de ré-échantillonnage, les particules avec des poids forts (importants) sont répliquées (dupliquées) et celles avec des poids trop faibles (négligeables) sont supprimées (éliminées) [26]. En d'autres termes, les particules de poids négligeables sont remplacées par de nouvelles particules à proximité des particules de poids plus élevé afin de ne conserver que les particules les plus significatives.

Notez que le ré-échantillonnage est la seule étape *PF* qui ne dépend pas de l'application ou du modèle d'espace d'état. Alors, l'analyse et les algorithmes de ré-échantillonnage sont généraux [30].

Par conséquent, le ré-échantillonnage a fait l'objet de recherches approfondies et de divers schémas de ré-échantillonnage ont été proposés.

Dans la littérature, différentes méthodes de ré-échantillonnage différentes peuvent être trouvées. Les algorithmes les plus fréquemment rencontrés sont le ré-échantillonnage multinomial [45], le ré-échantillonnage stratifié [36], le ré-échantillonnage systématique et le ré-échantillonnage résiduel. Un aperçu sur les méthodes de ré-échantillonnage peuvent être trouvées dans [36-40].

Les algorithmes de ré-échantillonnage sont divisés en algorithme séquentiel et à distribution/parallèle. Les plus courants sont les algorithmes séquentiels, où le ré-échantillonnage est effectué à partir de la distribution approximée en utilisant les derniers poids. Nous appelons également ces algorithmes traditionnels.

Le critère d'implémentation du ré-échantillonnage est d'habitude basé sur la variation des poids qui reflète le degré de dégénérescence. La taille effective de l'échantillon notée N_{eff} est un indicateur de la dégénérescence du filtre. N_{eff} est relié à la variance des poids normalisés à l'instant k , $\{w_k^i\}_{i:0}^N$ via l'égalité [30] :

$$N_{eff} = \frac{N}{1 + Var_{p_{0:k}}(w_k(x_{0:k}))} \quad (151)$$

Ainsi, plus la variance des poids est élevée, plus la taille effective de l'échantillon est faible. Par ailleurs, N_{eff} est relié à la divergence du χ^2 entre $p_{0:k}$ et $q_{0:k}$ via l'égalité :

$$\frac{N}{N_{eff}} - 1 = \chi^2(p_{0:k}, q_{0:k}) = \int \frac{(p_{0:k}(x_{0:k}) - q_{0:k}(x_{0:k}))^2}{q_{0:k}(x_{0:k})} dx_{0:k} \quad (152)$$

On peut remarquer que $1 \leq N_{eff} \leq N$, autrement dit, N_{eff} est un indicateur du nombre d'échantillons qui contribuent significativement à l'estimation de la loi $p(x_k | y_{0:k})$. En pratique, lorsque N_{eff} est inférieur à un seuil prédéfini N_{th} , on applique le ré-échantillonnage. Pour cela, l'approximation suivante de N_{eff} est souvent utilisée :

$$N_{eff} \approx \frac{1}{\sum_{i=1}^N (w_k^i)^2} \quad (153)$$

5.5. Modélisation

Les méthodes de *FPS* supposent souvent que X_k et les observations Y_k peuvent être modélisées sous la forme [42]. x_0, x_1, \dots est un processus de *Markov* sur \mathbb{R}^{d_x} (pour certain $d_x \geq 1$) qui évolue en fonction de la densité de suivi de transition $p(x_k | x_{k-1})$. Ce modèle est également souvent écrit de manière synthétique comme:

$$X_k | X_{k-1} = x_{k-1} \sim p(x_k | x_{k-1}) \quad (154)$$

Avec une densité de conversation initiale $p(x_0)$.

Les observations y_0, y_1, \dots , prennent des valeurs dans un certain espace d'état sur \mathbb{R}^{d_y} (pour certain $d_y \geq 1$) sont conditionnellement indépendantes à condition que x_0, x_1, \dots , soient connus. En d'autres termes, chaque Y_k dépend uniquement de X_k . En outre, nous supposons une distribution conditionnelle pour Y_k donné $X_k = x_k$ sont absolument continus, et de manière synthétique, nous avons ;

$$X_k | Y_k = y_k \sim p(y_k | x_k) \quad (155)$$

5.6. Approximation de MC

Les *FPs*, comme toutes les méthodes à base d'échantillonnages (telles que les *MCMC*), créent un ensemble d'échantillons qui approchent la distribution de filtrage $p(x_k | Y_0, \dots, Y_k)$ [43]. Ainsi, avec N échantillons, les valeurs espérées vis-à-vis de la distribution de filtrage sont approximées par :

$$\int f(x_k) p(x_k | Y_0, \dots, Y_k) dx_k \approx \frac{1}{N} \sum_{L=1}^N f(x_k^{(L)}) \quad (156)$$

où $x_k^{(L)}$ est la $L^{\text{ième}}$ particule à l'instant k ; et $f(\cdot)$, de la façon habituelle des méthodes *MC*, peut donner toutes les données de la distribution (moments, *etc.*) jusqu'à un certain degré d'approximation. Généralement, l'algorithme est répété itérativement pour un nombre donné de valeurs.

Chapitre 6 Détection des Défauts et Validation des Signaux

6.1. Introduction

La phase de détection permet de décider si le système se trouve ou non dans un état de fonctionnement normal. Pour cela, il suffit de tester la valeur des résidus s'il ne sont pas nulles afin de décider de l'apparition d'un défaut [44]. En pratique, le problème n'est pas si simple, car les grandeurs mesurées sont toujours entachées de bruits. D'autre part, le système surveillé est toujours soumis à des perturbations non nécessairement mesurables et le modèle utilisé n'est qu'une représentation approximative du système réel. Ce pour cela les résidus peuvent être non-nuls même en l'absence d'un défaut. Donc, il est nécessaire de définir des seuils de détection.

En particulier, nous utilisons le *FP*, un estimateur d'état largement utilisé pour les systèmes *NLs* stochastiques généraux. Le *FP* est un estimateur d'état flexible et largement applicable, et il a été utilisé pour la détection des défauts en fournissant des « *analyses redondantes* ».

6.2. FPs pour la détection des défauts

L'avantage d'utiliser la *PDF* complète de l'état du système dans un schéma de détection de défaut est supérieur à celui qui utilise les approximations, comme dans *l'FKE*. Étant donné les mesures $z_k = \{y_1, y_2, \dots, y_k\}$, la vraisemblance du paramètre θ du modèle peut être exprimé la non-nullité en utilisant les *DC* comme suit [27] :

$$L(\theta) = p(y_1, y_2, \dots, y_k | \theta) = \prod_{j=1}^k p(y_j, y_1, \dots, y_{j-1}, \theta) = \prod_{j=1}^k p(y_j | z_{j-1}, \theta) \quad (157)$$

Dans la méthode utilisée, la fonction de décision est choisie comme la somme *des log-vraisemblances* du paramètre système θ du mode système l sur une 'fenêtre glissante' de largeur M qui est calculé comme :

$$d_k = \sum_{j=k-M+1}^k \ln(p(y_j | z_{j-1}, \theta)) \quad (158)$$

Dans la mise en œuvre de la méthode utilisée, le calcul de la fonction de décision (158) est réalisé en réutilisant la *vraisemblance* de chaque particule *a priori* calculée lors du filtrage de particules.

6.3. Algorithme Proposé

L'algorithme de détection du défaut développé dans cette section devrait identifier tout changement dans le modèle. Une manière pour surveiller les changements de θ est de surveiller la fonction de *vraisemblance* des mesures, $p(y_{1:k}|\theta)$, sous les hypothèses suivantes[22] :

- $H_0 : \theta = \theta_0$ Hypothèse nulle.
- $H_1 : \theta \neq \theta_0$ Hypothèse alternative.

où θ_0 représente la vraie valeur de θ et si en fait $\theta = \theta_0$, alors on dit que l'hypothèse nulle est vraie. Une condition de défaut est une hypothèse dans laquelle l'hypothèse alternative est considérée comme vrai. Une approche populaire pour tester les hypothèses ci-dessus est pour effectuer le test du *Rapport de Vraisemblance (RV)*. Les bases l'idée derrière ce test est de surveiller le logarithme du rapport entre la vraisemblance des mesures sous zéro hypothèse et celle sous hypothèse alternative. En fait, qu'il y des différentes variantes de la statistique de test pour le *RV*, nous utilisons :

$$L = \log \left[\frac{p(y_{1:k}|\theta_0)}{\sup_{\theta} p(y_{1:k}|\theta)} \right] \quad (159)$$

Afin d'évaluer cette statistique de test, il faut estimer la fonction de *vraisemblance*, $p(y_{1:k}|\theta)$. Cette évaluation n'est pas simple en raison des états cachés dans le modèle de processus [17]. La fonction de *vraisemblance* a été étendue comme suit :

$$p(y_{1:k}|\theta_0) = p(y_1|\theta_0) \prod_{i=2}^k p(y_i|y_{1:i-1}, \theta) \quad (160)$$

et $p(y_i|y_{1:i-1}, \theta)$ a été approximé à l'aide de:

$$p(y_i|y_{1:i-1}, \theta) = p(h(y_i|y_{1:i-1}, \theta)|\theta) \left| \frac{\partial h}{\partial y_i} \right| \quad (161)$$

où h est la fonction inverse de g qui fournit une estimation du bruit de mesure w_i . Il y a deux problèmes avec (161). Cette formule ne s'applique qu'aux fonctions monotones, h , et il n'est souvent pas possible d'obtenir une expression analytique pour h . Afin de surmonter ces problèmes, nous proposons d'utiliser la fonction de *vraisemblance* conjointe suivante qui rassemble les états et les mesures afin de créer une nouvelle statistique de test qui remplace le *RV* standard ;

$$L_{new}(m,k) = \log \left[\frac{E[P(y_{k-m:k}, x_{k-m:k} | \theta_0)]}{\sup_{\theta} E[P(y_{k-m:k}, x_{k-m:k} | \theta)]} \right] \quad (162)$$

où E désigne l'espérance par rapport à la valeur de l'états cachée $x_{k-m:k}$. L'entier, m désigne la fenêtre de temps sur laquelle la nouvelle statistique est calculée. La proposition le logarithme de *vraisemblance* est le logarithme du rapport entre fonction de *vraisemblance* conjointe évaluée à la vraie valeur de la vecteur, et celui évalué à la valeur maximale de θ . Par conséquent, L_{new} sera toujours négatif. La nouvelle statistique L_{new} peut être facilement calculé en utilisant une approximation de particules de l'intégrale correspondante ;

$$E[P(y_{k-m:k}, x_{k-m:k} | \theta)] = \int [P(x_{k-m:k} | y_{k-m:k}, \theta)] \log P(y_{k-m:k}, x_{k-m:k} | y_{k-m:k}, \theta) dx_{k-m:k} \quad (163)$$

Pour plus de simplicité, nous utilisons l'approximation à base de particules suivante qui tient compte de la probabilité d'occurrence des états,

$$E[P(y_{k-m:k}, x_{k-m:k} | \theta)] \approx \int \sum_{i=1}^N w_{k-m:k}^i \log P(x_{k-m:k}^i, y_{k-m:k} | y_{k-m:k}, \theta) \quad (164)$$

où $w_{1:k}^i$ sont des poids proportionnels à la probabilité d'occurrence d'estimations de séquence d'états, $x_{1:k}^i$, obtenus à travers des *FPs*. La propriété de *Markov* du *NL* du modèle d'espace d'état permet d'étendre l'état conjoint et probabilité de mesure comme suit :

$$p(x_{k-m:k}^i, y_{k-m:k} | y_{k-m:k}, \theta) = (x_1^i | y_{k-m:k}, \theta) \prod_{t=k-m}^k P(x_t^i | x_{t-1}^i, \theta) \prod_{t=k-m}^k P(y_t | x_{t-1}^i, \theta) \quad (165)$$

et l'utiliser dans (163). Avec cette approximation, il est possible d'évaluer la statistique du *RV* proposée dans (164). Le processus est considéré comme ayant présenté un défaut si la nouvelle statistique, l_{new} est inférieure à une sensibilité prédéterminée. Idéalement, on aimerait choisir ce niveau de sensibilité telle que la probabilité d'erreur de type *I* est minimisée, et est souvent choisi à l'aide d'une fonction de puissance. Il est difficile de dériver une fonction puissance pour la statistique proposée, cependant, on peut utiliser un niveau de sensibilité empirique pour détecter les défauts. Une sensibilité de niveau α est choisie par l'utilisateur de telle sorte que lorsque $l_{new} < \alpha$, le processus est sensé avoir rencontré un défaut.

6.4. Détection des Défauts et Validation des Signaux par FPs

La détection des défauts et la prévision des pannes est une tâche cruciale tant du point de vue économique que sécurité. Pour les systèmes *NLs* et *NGs*, la prédiction pour la détection des défauts et des pannes est une tâche difficile, car les algorithmes existants sont destinés aux systèmes linéaires et *Gaussiens* seuls. Ainsi, en appliquant ces algorithmes pour les systèmes

NLs certaines hypothèses sont faites qui ne donnent pas des résultats précis. Divers algorithmes ont été développés pour l'estimation d'état *NL*, parmi eux on trouve la méthode du *FP*, adoptées pour la prédiction et la détection des défauts dans ce mémoire [28].

6.4.1. Algorithme

L'application de la *PDF* de l'état du système pour l'estimation et la détection des défauts est plus avantageuse que d'utiliser les approximations. La méthode est basée sur la règle de décision pour la faute détection, construite en utilisant la vraisemblance du paramètre du système, calculée en utilisant les informations de la *PDF* d'état représentées par les particules du *FP*. Les estimations d'état sont faites en utilisant le modèle d'évolution :

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1}) \quad (166)$$

Ces estimations d'état sont utilisées pour prédire les valeurs d'observation :

$$\hat{y}_k = h_k(\hat{x}_{k|k-1}) \quad (167)$$

Les mesures jusqu'à l'instant k sont supposées être données :

$$y_k = \{y_1, y_2, \dots, y_k\} \quad (168)$$

Le résidu ou l'innovation est donné par :

$$r_k = y_k - \hat{y}_k \quad (169)$$

La vraisemblance du paramètre du modèle peut être exprimée en utilisant la *DC*

$$L(\theta) = p(y_1, y_2, \dots, y_k | \theta) \quad (170)$$

Algorithm :

- *Etat de prédiction* : Un ensemble de N échantillons $\{x_k^*(i) = 1, 2, \dots, N\}$ sont générés comme dans l'étape de prédiction du *FP*.
- *Évaluation de la vraisemblance* : les résidus des particules antérieures sont évalués à l'aide de l'équation de mesure de sortie et la mesure $y(k)$ reçue. Ces résidus sont utilisés pour calculer la *vraisemblance* des particules :

$$l_k(i) = p(y | x_k^*(i), \theta) \text{ où } (i = 1, 2, \dots, N) \quad (171)$$

- *Détection de défaut* : les échantillons prédits peuvent être considérés comme les échantillons de la *PDF* $p(x_j | y_{j-1}, \theta)$ ainsi les termes individuels dans la fonction de décision peuvent être évalué par l'*IMC*:

$$L_j(\theta) \triangleq p(x_j | y_{j-1}, \theta) = \int p(y_j | x_j, \theta) p(x_j | y_{j-1}, \theta) dx_j \quad (172)$$

$$\approx \frac{1}{N} \sum_{i=1}^N p(y_k | x_k^*(i), \theta) = \frac{1}{N} \sum_{i=1}^N l_j(i)$$

- *Mise à jour de l'état* : les poids $q_k(i)$ pour les échantillons $\{x_k^* : i = 1, 2, \dots, N\}$ sont calculés comme à l'étape de *mise à jour* du *FP* :

$$q_k(i) = \frac{l_k(i)}{NL_j(\theta)} \quad (i = 1, 2, \dots, N) \quad (173)$$

- *Ré échantillonnage* : les échantillons $\{x_k^* : i = 1, 2, \dots, N\}$ sont obtenus par ré-échantillonnage.
- Ces étapes sont répétées récursivement pour chaque instant k .

7.1. Introduction

Les techniques d'estimation de données qui traitent le problème de la détection des défauts et la validation des signaux sont parfois lourdes à mettre en œuvre et ne permettent pas toujours d'avoir de bons résultats. Alors, on applique dans ce mémoire une méthode très utilisée pour l'estimation des signaux dans les systèmes *NL* en *NBs*.

La section suivante expose un exemple d'un système *NL* munis des bruits *NGs*. Les résultats sont obtenus par simulation par *MATLAB*. Sachant que *MATLAB* est un logiciel de calcul matriciel à syntaxe simple. Avec ses fonctions spécialisées, ce logiciel peut être aussi considéré comme un langage de programmation adapté pour les problèmes scientifiques.

7.2. Problématique

Dans un système *NL*, il s'agit d'estimer l'état X_k du système étant donné les observations (Y_0, \dots, Y_k) recueillies dans un bruit additif, pas nécessairement *Gaussien*. En général, les observations seules ne permettent d'estimer l'état caché, et il est nécessaire de disposer d'un modèle a priori, suffisamment réaliste, pour décrire l'évolution de l'état entre deux instants successifs d'observation [5]

Nous cherchons à détecter les défauts. Ceci revient à trouver une technique qui s'adapte mieux au signal. Le *FP* comme a été présenté en chapitre 5, répond bien à cette exigence, si en le comparant avec d'autres approches.

7.3. Développement de la méthode

Quand le système est affecté par un défaut additif entachant les mesures, l'équation de modèle d'espace d'état du système *NL* est donnée par :

$$\begin{aligned}x_k &= \frac{1}{2} x_{k-1} + 15 \times \frac{x_{k-1}}{1+x_{k-1}} + 4 \times \sin(1.155 \times k) \times \cos(2.57115 \times k) + u_k; \\y_k &= \frac{1}{0.2} x_k + v_k\end{aligned}\tag{174}$$

L'estimation récursive *Bayésienne* est implémentée par échantillonnage de *MC*, plutôt qu'en résolvant les intégrales dans les équations (119) :

$$p(x_k | y_{1:k}) = \frac{p(y_k | x_k) p(x_k | y_{1:k-1})}{p(y_k | y_{1:k-1})} \quad (175)$$

A chaque pas du temps k , les deux informations requises pour estimer la *PDF* sont les échantillons x_k^i et leurs poids associés w_k^i . Les échantillons x_k^i sont supposés être générés à partir d'une densité connue appelée fonction de densité d'importance, $q(x_k | y_{1:k})$, et les poids correspondants aux échantillons.

La densité d'importance ne dépend plus que de x_{k-1} et y_k . Ceci est particulièrement utile dans le cas où seule une estimation filtrée de $p(x_k | y_{1:k})$ est requise à chaque pas du temps.

En utilisant les hypothèses d'espace d'état, les poids d'importance peuvent être estimés récursivement, et la densité filtrée *postérieure* $p(y_k | y_{1:k})$ peut être approximée par :

$$p(y_k | y_{1:k}) \approx \sum_{i=1}^N w_k^i \delta(x_k - x_k^i) \quad (176)$$

où δ est la fonction delta de Dirac, x_k^i est le i ème échantillon qui se rapproche de la distribution, et le coefficient w_k^i est le poids associé à l'échantillon correspondant. Ça peut être montré que comme $N \rightarrow \infty$, l'approximation dans l'équation (176) se rapproche de la vraie *densité postérieure* $p(x_k | y_{1:k})$.

Après quelques itérations, les poids de la plupart des particules deviennent insignifiants en raison des erreurs de modélisation et du bruit qui font dériver les particules de l'état réel. Par conséquent, la plupart des particules n'ont aucune influence sur la partie fonction postérieure de densité, et l'approximation des distributions cibles ne sont alors déterminés que par quelques particules. Ce problème est appelé *dégénérescence* et ne peut être évité. Cette dégénérescence implique que les performances de cet algorithme peuvent affaiblir l'échantillonnage *MC* et les ressources de calcul peuvent être gaspillées à mettre à jour des particules avec peu ou pas de pertinence pour l'approximation de la fonction de densité postérieure $(x_k | y_{1:k})$.

Ce problème de dégénérescence peut être résolu en utilisant une technique appelée *ré-échantillonnage*. L'idée de base de la méthode de cette méthode est d'éliminer les trajectoires qui ont de petits poids d'importance et de se concentrer sur les trajectoires avec gros poids.

7.4. Application et résultats

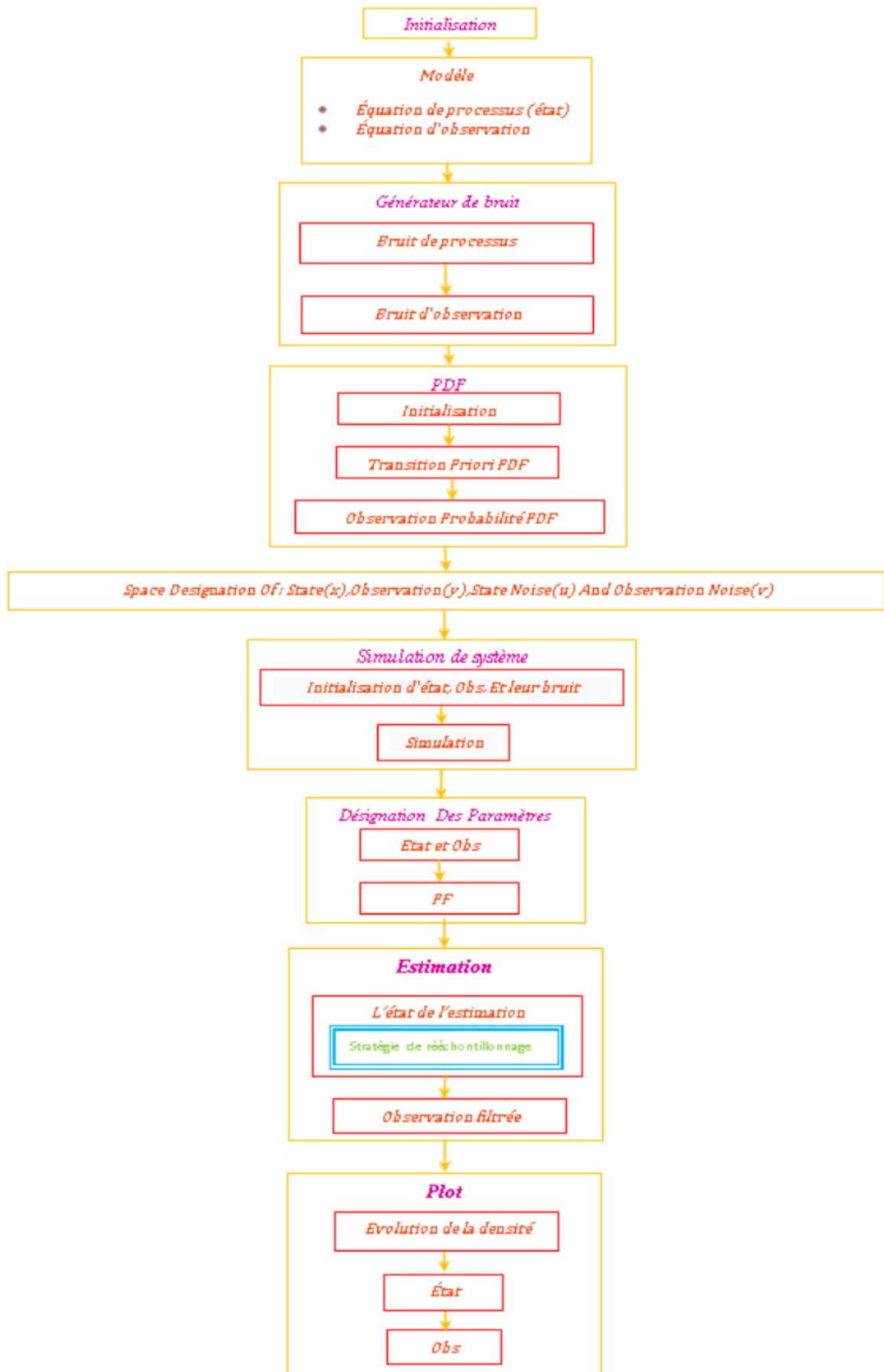


Figure 7-1 : Organigramme présente les étapes d'organisation de programme MATLAB

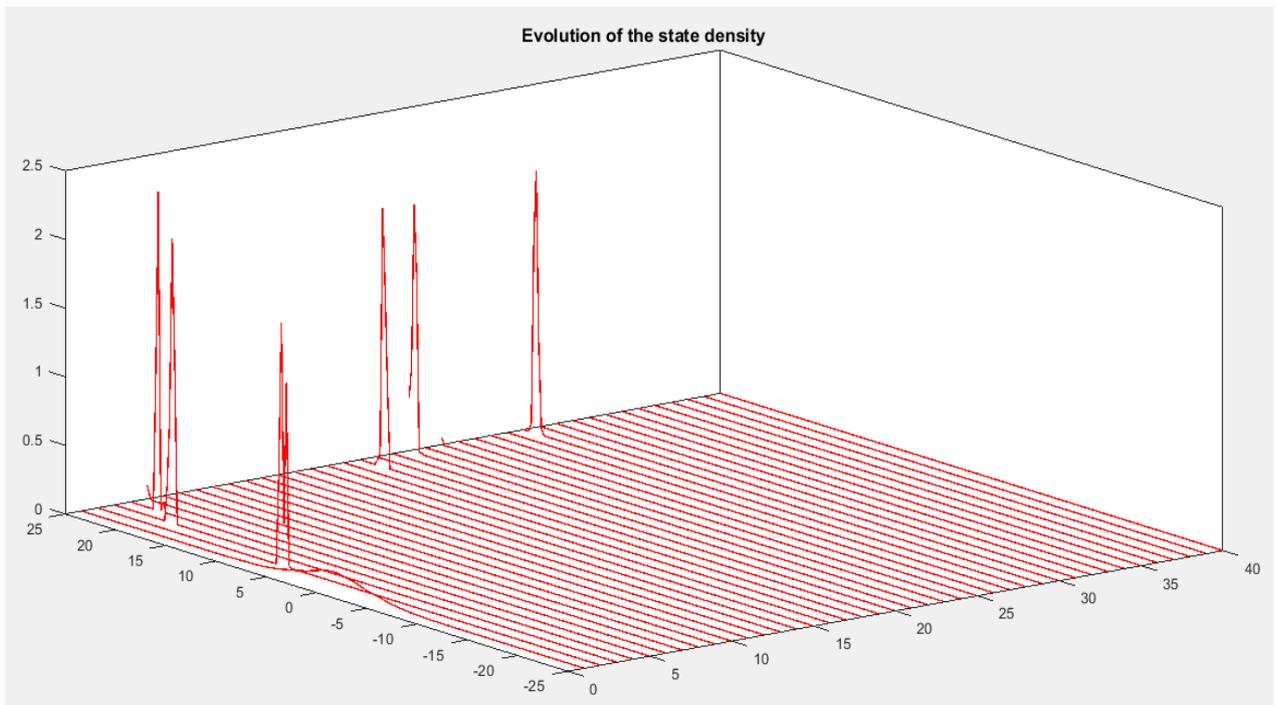


Figure 7-2 : évolution de la densité d'état

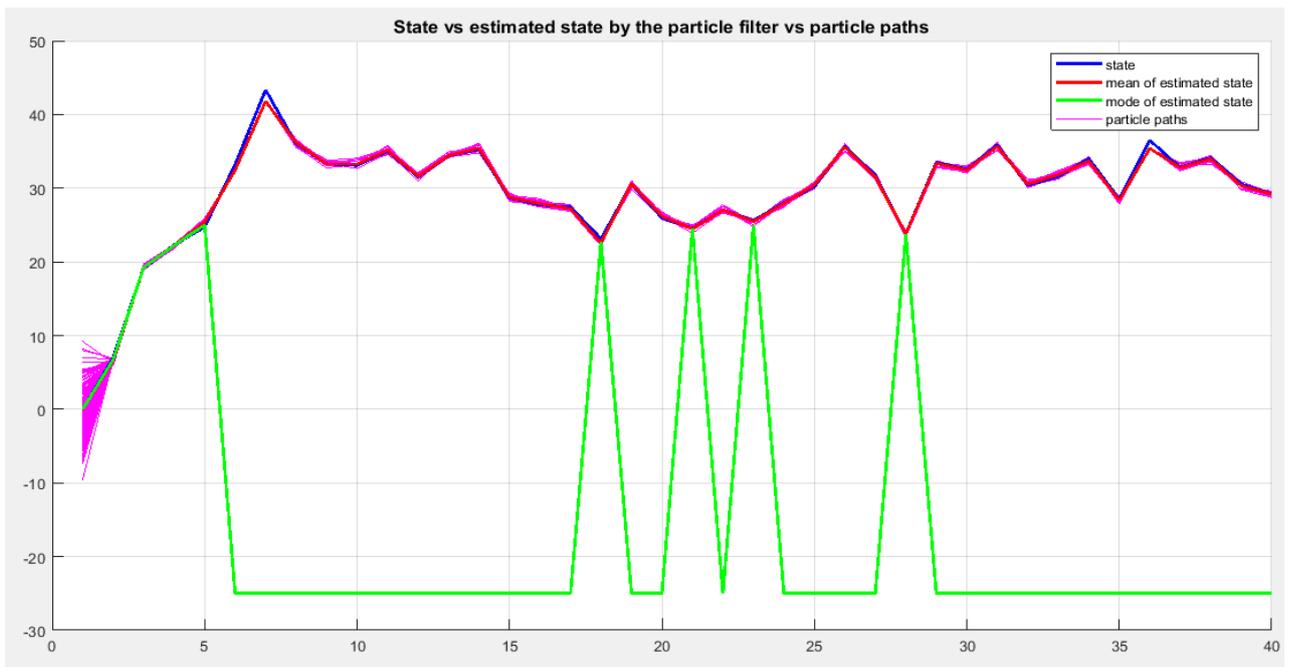


Figure 7-3 : L'état en fonction état estimé par particule et chemins de particules

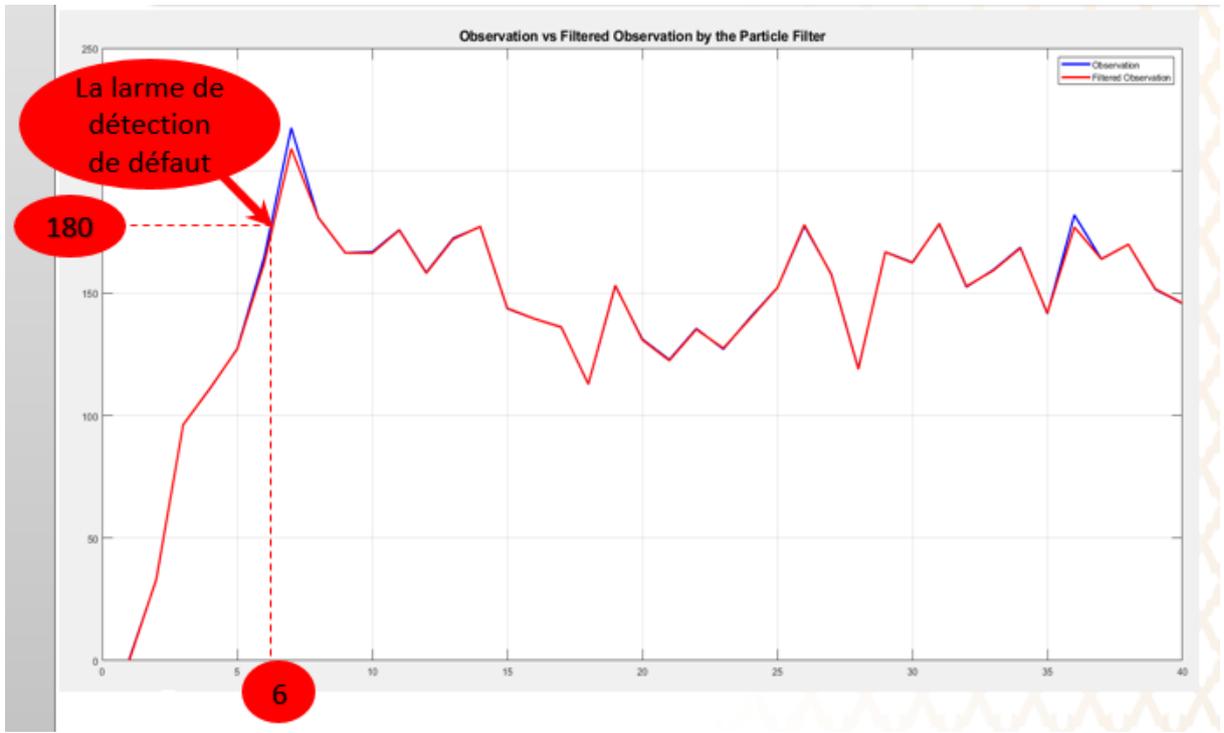


Figure 7-4 : L'observation en fonction d'observation filtrée par filtre à particules

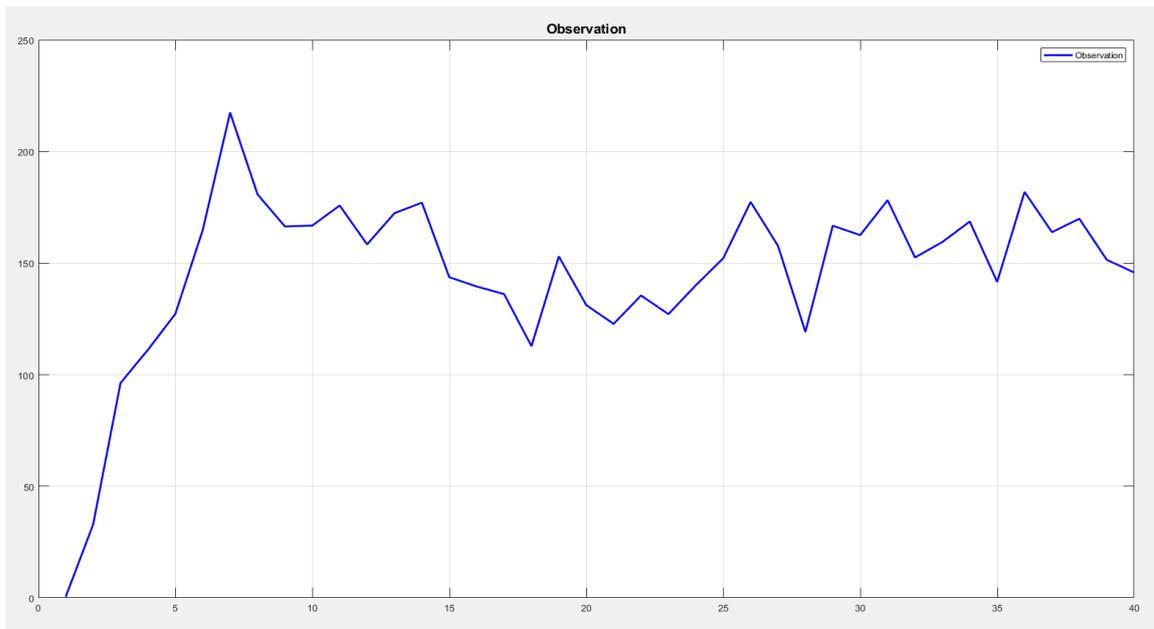


Figure 7-5 : L'observation.

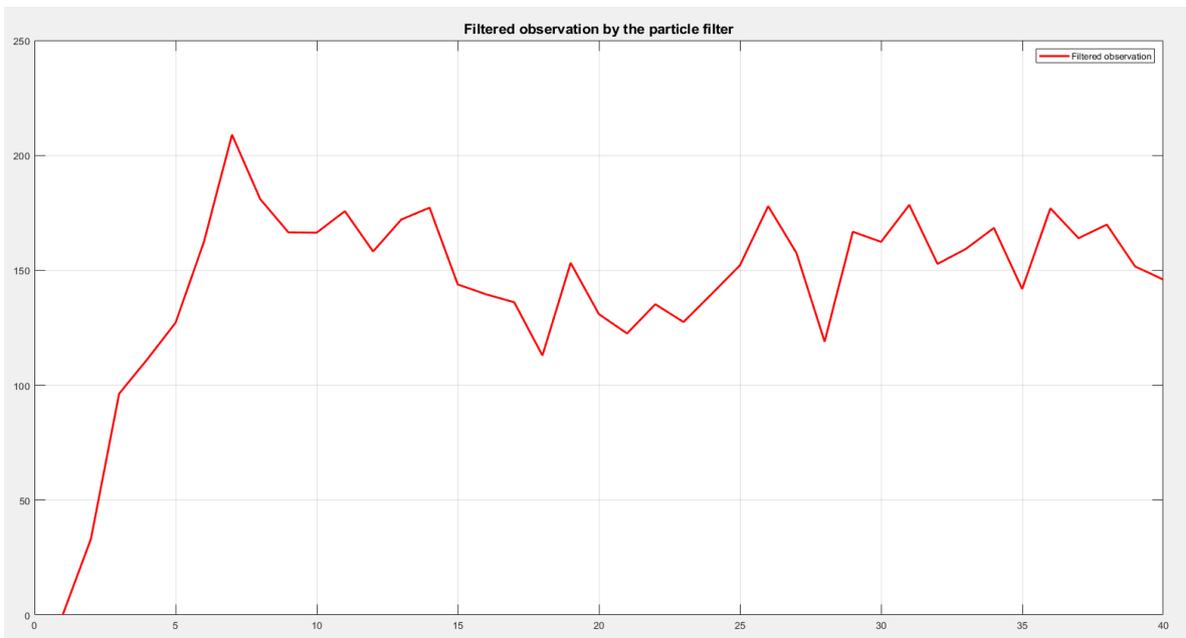


Figure 7-6 : L'observation filtrée par filtre à particules

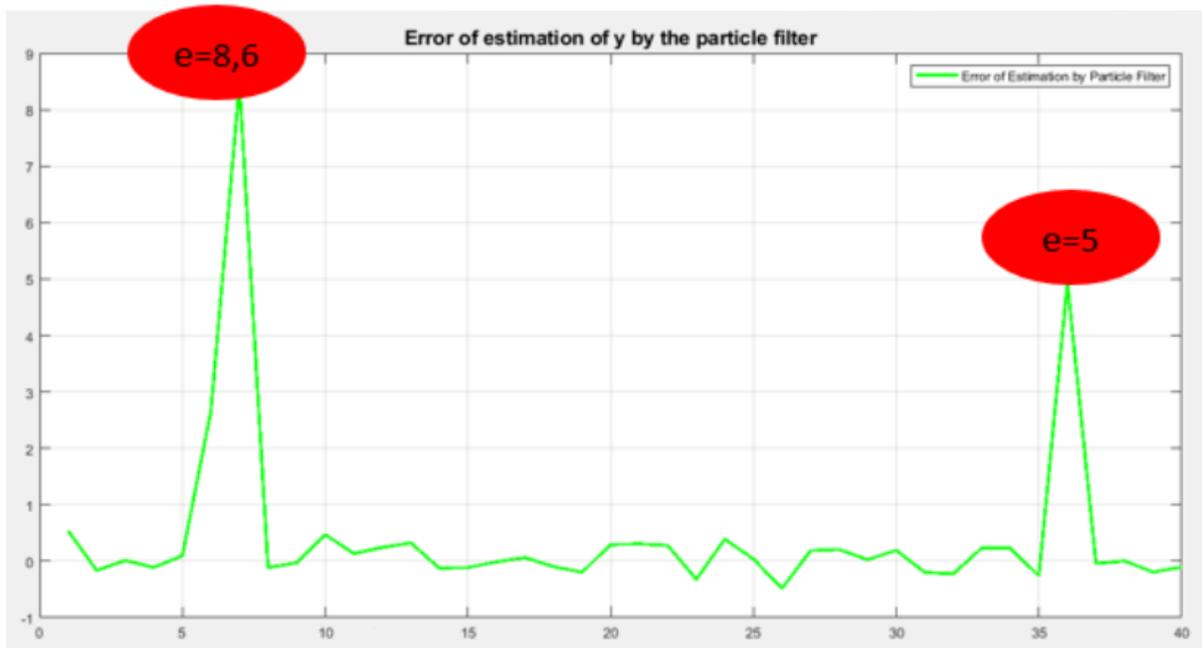


Figure 7-7 : L'erreur d'estimation d'observation par le filtre à particules

Figure 7.2 : On a effectué une "fonction de lissage" de particule et estimation de densité x_k dans chaque instant.

Figure 7.3 : On remarque que les états et les états estimés par *FP* dans l'intervalle [10, 35] sont proches. Les états et les états estimés sont presque superposés, ce qui signifie que l'estimateur à base de *FP* est bien choisi.

Par contre dans l'intervalle [5, 10] et [35, 40] les états et les états estimés sont légèrement différents à cause de l'apparition du défaut.

Figure 7-4, 7-5, 7-6 : On remarque que les observations et observations estimées par *FP* sont différentes à cause de l'apparition de défaut dans l'intervalle [5, 10] et [35, 40], cette différence sera importante si l'amplitude de défaut est grande.

A $T=10$ jusque à $T=35$; les états d'observation et observation filtrée par *FP* sont proches et ce dû à la correction du système par retour d'état. Les états d'observation et observation filtrée sont presque superposés, ce qui signifie l'absence de défaut.

Figure 7-7 : On remarque que dans l'intervalle [10, 35], il y'a pas d'erreur et que les signaux ne sont pas affectés par des défauts ce qui signifie le signal est validé (i.e., l'absence d'erreur signifie l'absence de défaut donc le signal est validé).

Dans l'intervalle [5, 10] et [35, 40], les signaux sont affectés par des défauts ce qui signifie le signal n'est pas la validé (i.e., la présence d'erreur signifie la présence de défaut donc le signal n'est pas validé).

Calcul De L'erreur Maximale :

D'après la figure 7-4 :

A l'intersection de l'axe x et l'axe y on a :

- Le temps de l'alarme est $T=6s$.
- La valeur de seuil égale à 180

Ce point définit le seuil de l'alarme de détection des défauts. Par conséquent, si le signal dépasse cette valeur du seuil, l'alarme se déclenche ce qui signifie une détection de défaut. Deux valeurs importantes de ce seuil sont constatées, alors on prend la valeur maximale. A partir de la figure 7-7, la valeur du seuil de détection est fixée à l'erreur maximale qui est 8.6.

7.5. Conclusion

Dans ce chapitre nous avons analysé et présenté les résultats de simulation pour un système prototype de nature *NL* et *NG*, en utilisant le *FP* comme une méthode d'estimation. On remarque que ce type de filtre fait la prédiction avec une bonne précision, ce qui permet d'avoir des résultats meilleurs pour les applications de la détection des défauts et de validation des signaux.

Dans ce mémoire, notre travail était consacré à la validation des signaux (données) et la détection des défauts dans les systèmes dynamiques décrits par un modèle discret NLs munis d'un bruit NG . Cette action est accomplie par l'application du FB , précisément la méthode de Monte-Carlo séquentielle appelée aussi FPs . L'utilisation de ces filtres ont eu une bonne réputation dans tels systèmes et dans différents domaines d'estimation, vue la qualité de leurs résultats.

Un FP est un FB récursive qui utilise l'échantillonnage ou simulation de Monte Carlo. Il est considéré une alternatif à l'approche de *Kalman* pour l'estimation d'état et paramètres.

Le FP permet d'avoir une solution à l'estimation du vecteur d'état compte tenu des observations (mesures ou échantillons) enregistré au cours du temps. En effet, cette technique permet de gérer les modèles d'évolutions et les densités de probabilités multimodales et NGs . Les FPs approximent la distribution de fonction probabilité de l'état d'un système par un ensemble d'échantillons ou particules. Ces particules se déplacent selon des réalisations indépendantes à partir de l'équation d'état. Elles sont corrigées ou pondérées en fonction de leur cohérence avec les observations (quantifiée par la fonction de vraisemblance). Par conséquent, un FP opère sur trois principales opérations : génération des particules (propagation), calcul des poids et ré-échantillonnage. Cette dernière étape est essentielle et de haute importance, car elle empêche l'apparition de la dégénérescence des particules, i.e., peu de particule domine avec leurs poids élevés, ce qui rend l'estimation obtenue non-précise. Par conséquent, le ré-échantillonnage est extensivement recherché et comme résultat, divers algorithmes ont été proposés.

Pour la mise en œuvre de ce travail, nous avons développé un programme sous *MATLAB* qui permet la simulation d'un système de détection des défauts et de validation des signaux à base d'estimation. Après l'acquisition des données, bruitées, un prétraitement était nécessaire avant leur utilisation. Enfin, les résultats d'estimation, en utilisant les FPs , pour la validation des signaux et la détections des défauts étaient bons et convaincantes.

- [1] M. Speekenbrink, « A Tutorial on Particle Filters », *Journal of Mathematical Psychology*, august 2016.
- [2] B. Orlande, « Tutorial 10 Kalman and Particle filters », 2011.
- [3] A. Doucet, et al., « On Sequential Monte-Carlo Sampling Methods for Bayesian filtering », 2000.
- [4] F. Gustafsson, « Particle Filter Theory and Practice with Positioning Applications », IEEE, 2010.
- [5] F. Legland, « Filtrage particulaire », IRISA / INRIA Campus de Beaulieu, 35042 RENNES Cédex, France.
- [6] P. Del Moral, J. Jacod « Interacting particle filtering with discrete observations », chapter 3, pp 43-75 in A. Doucet et al., "Sequential Monte Carlo Methods in Practice", *Statistics for Engineering and Information Science*, Springer – Verlag, NewYork, 2001.
- [7] J. Rodon, « Hidden Markov Model - Description du modèle », //igm.univ-mlv.fr/~dr/XPOSE2012/HiddenMarkovModel/description.html#hmm.
- [8] B. Hamrouni, « Désambiguïsation dans la typification de données textuelles à des fins d'analyses catégorielles », *Mémoire de la Maîtrise en Mathématiques*, Université du Québec à Trois-Rivières, mars 2008.
- [9] Fang-I Chu, « From Bayesian to Particle Filter », *Master thesis, Faculty of San Francisco State University*, 2009.
- [10] Amadou-Abdoulaye, « Contribution à la surveillance d'un processus de forage pétrolier », *Thèse de Dكتورat*, École Nationale Supérieure d'Arts et Métiers, 2010.
- [11] D. Crisan et A. Doucet, « Convergence of Sequential Monte Carlo Methods », *Computer Science*, 2007.
- [12] F. Touam et F. Hadj Mihoub Sidi Moussa, « Détection et localisation des défauts des capteurs », *Mémoire de Master II, universite saad dahleb, blida*, 2012/2013.
- [13] O. Gaudoin : "Principes et Méthodes Statistiques", *Cours, Grenoble*.
- [14] A. Maddi, A Guessoum et al., « Etude de la methode de moindre carre recursive et application au signal de parole », *3rd International Conference: Sciences of Electronic, Technologies of Information and Telecommunications March 17-21, Tunisia*, 2005.
- [15] G. Faporta, « Probabilité, analyse des données et statistique », Paris, 2006.
- [16] G.Touzin, « Etude des méthodes de monte-carlo et de leurs efficacités relatives », *Mémoire de la Maîtrise en Mathématiques, Université du Quebec à trois-rivières*, Avril 2013.
- [17] R. Matio et Hobihery, « Estimation et stabilite en filtrage non lineaire : approximations particulaires », *Thèse de Doctorat, Université d'Antananarivo*, Decembre 2014.
- [18] M. Bourahla et H. Bachenine, « Filtrage particulaire: application à la de modulation de phase », *Mémoire Master II, université Blida 1*, 2012/2013.

- [19] « Bayes' Theorem », //corporatfinanceinstitute.com/resources/knowledge/other/bayes-theorem/.
- [20] J. Torstensson et M. Trieb, « Particle Filtering for Track Before Detect Applications », Master's Thesis, Linköping University, Sweden 2005.
- [21] P. Simon, « Détection robuste et précoce des pannes oscillatoires dans les systèmes de commandes de vol », Université de Bordeaux, Ecole doctorale des sciences physiques et de l'ingénieur, France, 2011.
- [22] F. Alrowaie, K.E. Kwok, R.B. Gopaluni, « An Algorithm for Fault Detection in Stochastic Non-linear State-Space Models Using Particle Filters », Proceedings of the 2011 4th International Symposium on Advanced Control of Industrial Processes, Thousand Islands Lake, Hangzhou, P.R. China, May 23-26, 2011
- [23] K. Dahia, « Nouvelles méthodes en filtrage particulaire: Application au recalage de navigation inertielle par mesure altimétriques », Thèse de Doctorat, Université Joseph Fourier, Janvier 2005.
- [24] P. Smith, « Particle filters in practice », Data Assimilation Research Centre, University of Reading, UK, 5 - 8 March 2019.
- [25] N. Paul, « Filtrage Particulaire », Séminaire de Statistique Conservatoire National des Arts et Métiers, jeudi 2 Mars 2006.
- [26] M. Bolic, P. M. Djuric et al., « Resampling Algorithms for Particle Filters: A Computational Complexity Perspective ». EURASIP Journal on Applied Signal Processing 2004.
- [27] V. Kadirkamanathan et al., 2002 « Particle filtering-based fault detection in non-linear stochastic systems », International Journal of Systems Science, 2002.
- [28] M. Mishra, « Model Based Approach For Fault Detection And Prediction Using Particle Filters », Master of Science Thesis, University Of Hawai, may 2008.
- [29] J. D. Hol, T. B. Schon, et al., « On resampling algorithms for particle filters », Linköpings university, Sweden, 2004.
- [30] A. Murangira, « Nouvelles approches en filtrage particulaire. Application au recalage de la navigation inertielle », Hal, 4 Jul 2014.
- [31] L. Ayadi, « étude dun système de capteur de mouvement (MOCAP) utilisant le filtrage particulaire », universite mouhamed khider biskra, 08/12/2015.
- [32] M. Davy, « filtrage Bayésien non lineaire et non gaussien : filtres particulaires », seminaire IRCCyN, cambridgc university engineering de partenet, 6 Septembre 2001.
- [33] W.Sorensen et al., « least-squares estimation from gauss to kalman », IEEE Spectrum Vol 7, pp. 63 - 68, University of California, Jul 1970.
- [34] M. S. Arulampalam, S. Maskell et al., « A Tutorial on particle filters for online non linear / non-gaussian Bayesian tracking », IEEE Transition on Signal Prpcessing, Vol 50, no.2, February 2002.
- [35] G. Field, « Particle filters for state estimation of confined a quifers », Master of Science Thesis, university of North Florida, UNF, 2018.
- [36] A. Doucet et al., « Sequential Monte Carlo Methods in Practice », Statistics for Engineering and Information Science, Springer – Verlag, NewYork, 2001.
- [37] A. Doucet, A. M. Johansen, « A Tutorial on Particle Filtering and Smoothing », Article citations, 2011.
- [38] C. Baehr, « Filtres particulaires et turbulence. Problèmes posés par le filtrage en grande dimension », Météo-France / IMT-LSP Univ. Paul Sabatier, 25 mars 2009.
- [39] T. Bertozzi, « these de doctorat application du filtrage particulaire aux communication numeriques

», paris, 2003.

- [40] J. D. Hol, T. B. Schön, F. Gustafsson, "On resampling algorithms for particle filters," *Proc. IEEE Nonlinear Statistical Signal Processing Workshop*, 2006, pp. 79–82.
- [41] C. Zhiqi, « A Theory of international strategic alliance », *Review of International Economics*, 11(5), 758–769, 2003.
- [42] Doucet et al., « *Sequential Monte-carlo Methode in practice* », Springer, January 2001.
- [43] H. Tanizaki, « On the nonlinear and nonnormal filter using rejection sampling », *IEEE Transactions on Automatic Control* (44), 314–319, 1999.
- [44] P. Billingsley, « *Probability and Measure* », Third Edition, John Wiley & Sons, 26 mai 1995.
- [45] N. J. Gordon, D. J. Salmond, A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proceedings*, vol. 140, no. 2, pp. 107–113, 1993.
- [46] M. S. Arulampalam, « A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking », *IEEE transactions on signal processing* vol 50 No 2, February 2002.

Annexe 1- Notation de Probabilités

Notation	Signification	
	Français	Anglais
x_k	Vecteur d'état	State vector
\hat{x}_k	Estimation d'état	condition estimate
w_k	Vecteur bruit d'état	State noise vector
y_k	Vecteur de mesure	Measurement vector
v_k	Vecteur bruit de mesure	Measurement noise vector
x_k^i	Vecteur d'état de la particule i à l'itération k	State vector of particle i at iteration k
N	Nombre de particules.	Number of particles.
$p(x_0 z_0) = p(x_0)$	État initial également connu sous le nom d'avant	Initial state, Prior
$p(x_k x_{k-1})$	PDF de Transition, Densité de transition d'état, Distribution de transition, Densité de processus, modèle de mouvement	Transition PDF, Transition density of the state, Transition Distribution, Process density Motion Model
$p(y_k x_k)$	Fonction de vraisemblance Modèle Perceptuel, Distribution	The likelihood function Perceptual Model, Distribution
$p(x_k y_k)$ $p(x_{k-1} y_{1:k-1})$	Densité postérieure ou croyance	Posterior Density or Belief
$p(x_k y_{1:k})$ $p(x_k y_k)$ $p(x(k) y(k))$	PDF postérieur, Densité antérieure, Distribution marginale, Etat de croyance	Posterior PDF, Prior density, Marginal Distribution, Belief state
$p(x_k)$	Distribution de filtrage	Filtering distribution
$q(x)$	Distribution d'échantillonnage.	Sampling distribution
$p(y_{1:k} K \geq 1)$	Probabilités marginales	Marginal likelihoods
$p(x_k y_{k-1})$	Distribution préalable de la PDF ou de	Prior PDF or filtering distribution.

	<i>filtrage.</i>	
$p(x_k y_{1:k-1})$ $p(x_k y_0, y_1, y_2, \dots, y_k)$	<i>Filtre prédit, Filtre prédiction, PDF priori, distribution priori. Distribution de filtrage</i>	<i>Filter Predict, Filter Prediction, Prior PDF, Prior Distribution. Filtering Distribution</i>
$p(x_{0:k} y_{1:k-1})$	<i>État a priori de la PDF</i>	<i>A priori state PDF</i>
$p(x_{0:k} y_{0:k})$, $p(x_{1:k} y_{1:k})$ $p(x_{0:k} y_{1:k})$, $p(X_k Y_k)$	<i>Distribution postérieure / densité. Répartitions conjointes Probabilité Jointes a postérieure. Distribution postérieure marginale</i>	<i>Posteriori Distribution / Density. Joint Distributions A posteriori Joint Probability Marginal Posterior Distribution.</i>
$p(x_k y_{0:k})$, $p(x_k y_{1:k})$, or $p(x_k Y_k)$	<i>Distribution marginale, distribution filtrante (la densité conditionnelle)</i>	<i>Marginal distribution, Filtering distribution (conditional density)</i>
$p(X_{k-1} Y_{k-1})$	<i>Densité de probabilité conditionnelle:</i>	<i>Densité de probability conditionnelle:</i>
$p(y_k x_k)$	<i>Fonction de vraisemblance, Densité d'observation</i>	<i>Likelihood function, Observation density</i>
$\{x_k^i\}_{Ni=1}$	<i>Posterior samples</i>	<i>Posterior samples</i>
$p(y_k y_{0:k-1})$	<i>Evidence</i>	<i>Evidence</i>
$q(x_{0:k} y_{1:k})$	<i>Fonction importante,</i>	<i>Importance function,</i>
$q(x_k y_{1:k})$	<i>Importance (proposition) densité (répartition)</i>	<i>Importance (proposal) density (distribution)</i>
$P(A)$	<i>probabilité de l'événement A</i>	<i>probability of event A</i>
$P(A B)$	<i>Probabilité que l'événement A se produise, étant donné que l'événement B s'est produit</i>	<i>Probability that event A will occur, given that event B has occurred</i>

Annexe 2 – programme MATLAB

```
%% ***** Initialisation *****

clear;    % clear memory

clc;      % clear screen,

close all; % close all figures

%

%% ***** Model *****

% ***** Process equation  $x[k] = \text{sys}(k, x[k-1], u[k])$  -----
sys = @(k, xkm1, uk) xkm1/2 + 15*xkm1/(1+xkm1^1) + 4*sin(1.155*k)*cos(2.57115*k) + uk; % {V Good}

    % Equ d'Etat

    % km1: k minus 1

nx = 1;    % number of states

% ***** Observation equation  $y[k] = \text{obs}(k, x[k], v[k])$  ***** %

%obs = @(k, xk, vk) xk^2/20 + vk; % (returns column vector)

obs = @(k, xk, vk) xk^1/0.2 + vk; % {Good}

    % Equ d'Observabilité

ny = 1;    % number of observations

%

%% ***** Noise Generator Function, PDF of process noise *****

%

nu = 1;    % size of the vector of process noise

sigma_u = sqrt(10);          % variance du bruit du systeme

gen_sys_noise = @(u) normrnd(0, sigma_u); % normrnd: Normal random numbers.

    % sample from p_sys_noise (returns column vector)

p_sys_noise = @(u) normpdf(u, 0, sigma_u); % normpdf: Normal Probability Density Function.

    % the PDF of process noise

%

%% ***** Noise Generator Function, PDF of observation noise *****

%
```

```

nv = 1;                % size of the vector of observation noise
sigma_v = sqrt(1);     % sigma: variance du bruit d'observation
%
gen_obs_noise = @(v) normrnd(0, sigma_v); % Noise Generator Function of observation
                % sample from p_obs_noise (returns column vector)
%
p_obs_noise = @(v) normpdf(v, 0, sigma_v); % PDF of observation noise
%
%% ***** Initial PDF *****
p_x0 = @(x) normpdf(x, 0, sqrt(10)); % initial pdf-
gen_x0 = @(x) normrnd(0, sqrt(10)); % sample from p_x0 (returns column vector)
%
%% ***** Transition prior PDF p(x[k] | x[k-1]) *****
%
% -----(under the suposition of additive process noise)-----
p_xk_given_xkm1 = @(k, xk, xkm1) p_sys_noise(xk - sys(k, xkm1, 0));

%% ***** Observation likelihood PDF: p(y[k] | x[k])(4) *****
%
% ----- (under the suposition of additive process noise)-----
p_yk_given_xk = @(k, yk, xk) p_obs_noise(yk - obs(k, xk, 0));

%% ***** Number of time steps *****
T = 40;
%
%% ***** Separate memory space *****
x = zeros(nx,T);
y = zeros(ny,T);
u = zeros(nu,T);
v = zeros(nv,T);
%% ***** Simulate system *****
% ***** Initialisation *****
xh0 = 0;                % initial state
u(:,1) = 0;            % initial process noise

```

```

v(:,1) = gen_obs_noise(sigma_v); % initial observation noise
        % gen_obs_noise
        % sigma_v

x(:,1) = xh0;
y(:,1) = obs(1, xh0, v(:,1));
% ***** simulation *****

for k = 2:T
    % here we are basically sampling from p_xk_given_xkm1 and from p_yk_given_xk
    u(:,k) = gen_sys_noise(); % simulate process noise, gen_sys_noise:
    v(:,k) = gen_obs_noise(); % simulate observation noise, gen_obs_noise:
    x(:,k) = sys(k, x(:,k-1), u(:,k)); % simulate state
    y(:,k) = obs(k, x(:,k), v(:,k)); % simulate observation
end
%
%% ***** Parameters Designation *****
xh = zeros(nx, T);
xh(:,1) = xh0;
%
yh = zeros(ny, T);
yh(:,1) = obs(1, xh0, 0);
%
pf.k      = 1; % initial iteration number
pf.Ns     = 200; % number of particles
pf.w      = zeros(pf.Ns, T); % weights
pf.particles = zeros(nx, pf.Ns, T); % particles
pf.gen_x0  = gen_x0; % function for sampling from initial pdf p_x0
pf.p_yk_given_xk = p_yk_given_xk; % function of the observation likelihood PDF p(y[k] | x[k])
pf.p_xk_given_xkm1 = p_xk_given_xkm1; % transition prior PDF p(x[k] | x[k-1])
pf.gen_sys_noise = gen_sys_noise; % function for generating system noise
pf.p_x0 = p_x0; % initial prior PDF p(x[0])
%
%% ***** Estimate state *****

for k = 2:T
    fprintf('Iteration = %d/%d\n',k,T);

```

```

% state estimation

pf.k = k;

[xh(:,k), pf] = particle_filter(sys, y(:,k), pf, 'multinomial_resampling');
%[xh(:,k), pf] = particle_filter(sys, y(:,k), pf, 'systematic_resampling');

% filtered observation

yh(:,k) = obs(k, xh(:,k), 0);

end

%

%% ***** Make plots of the evolution of the density *****

figure

hold on;

xi = 1:T;

yi = -25:0.25:25;

[xx,yy] = meshgrid(xi,yi);

den = zeros(size(xx));

xhmode = zeros(size(xh));

%

for i = xi

    % for each time step perform a kernel "smoothing funct." density estimation, xk

    den(:,i) = ksdensity(pf.particles(1,:,i), yi,'kernel','epanechnikov');

    %[~,idx] = max(den(:,i));

    % estimate the mode of the density

    %xhmode(i) = yi(idx);

    plot3(repmat(xi(i),length(yi),1), yi, den(:,i),'r','LineWidth',1);

        %repmat: replicat and tile array

end

%

view(3);

box on;

title('Evolution of the state density','FontSize',12)

%

figure

```

```

mesh(xx,yy,den);

title('Evolution of the state density','FontSize',12)

%

%% **** plot of the state vs estimated state by the particle filter vs particle paths

figure

hold on;

h1 = plot(1:T,squeeze(pf.particles),'m');
h2 = plot(1:T,x,'b','LineWidth',2);
h3 = plot(1:T,xh,'r','LineWidth',2);
h4 = plot(1:T,xhmode,'g','LineWidth',2);

legend([h2 h3 h4 h1(1)],'state','mean of estimated state','mode of estimated state','particle paths');

title('State vs estimated state by the particle filter vs particle paths','FontSize',12);

grid;

%

%% **** plot of the observation vs filtered observation by the particle filter

figure

plot(1:T,y,'b', 1:T,yh,'r','LineWidth',2);

legend('Observation','Filtered Observation');

title('Observation vs Filtered Observation by the Particle Filter','FontSize',14);

grid;

%

figure

plot(1:T,y,'b','LineWidth',2)% 1:T,yh,'r','LineWidth',2);

title('Observation','FontSize',14);

legend('Observation');

grid;

%

figure

plot(1:T,yh,'r','LineWidth',2);

legend('Filtered observation');

title('Filtered observation by the particle filter','FontSize',14);

grid;

%

figure

```

```

plot(1:T,y-yh,'g','LineWidth',2);
legend('Error of Estimation by Particle Filter');
title('Error of estimation of y by the particle filter','FontSize',14);
grid;
%
return;
%% ***** Generic particle filter *****
%
% Note: when resampling is performed on each step this algorithm is called
% the Bootstrap particle filter
% (P100)
% Usage:
% [xhk, pf] = particle_filter(sys, yk, pf, resampling_strategy)
%
% Inputs:
% sys = function handle to process equation
% yk = observation vector at time k (column vector)
% pf = structure with the following fields
% .k = iteration number
% .Ns = number of particles
% .w = weights (Ns x T)
% .particles = particles (nx x Ns x T)
% .gen_x0 = function handle of a procedure that samples from the initial pdf p_x0
% .p_yk_given_xk = function handle of the observation likelihood PDF p(y[k] | x[k])
% .gen_sys_noise = function handle of a procedure that generates system noise
% resampling_strategy = resampling strategy. Set it either to
% 'multinomial_resampling' or 'systematic_resampling'
%
% Outputs:
% xhk = estimated state
% pf = the same structure as in the input but updated at iteration k
%
%% *****
function [xhk, pf] = particle_filter(sys, yk, pf, resampling_strategy)

```

```

k = pf.k;
if k == 1
    error('error: k must be an integer greater or equal than 2');
end
%
%% ***** Initialize variables *****
Ns = pf.Ns;          % number of particles
nx = size(pf.particles,1); % number of states
wkm1 = pf.w(:, k-1); % weights of last iteration
%
if k == 2
    for i = 1:Ns      % simulate initial particles
        pf.particles(:,i,1) = pf.gen_x0(); % at time k=1
    end
    wkm1 = repmat(1/Ns, Ns, 1); % all particles have the same weight
end
%
%% *****
% The importance sampling function:
% PRIOR: (this method is sensitive to outliers) THIS IS THE ONE USED HERE
% q_xk_given_xkm1_yk = pf.p_xk_given_xkm1;

% OPTIMAL:
% q_xk_given_xkm1_yk = q_xk_given_xkm1^i_yk;
% Note this PDF can be approximated by MCMC methods: they are expensive but
% they may be useful when non-iterative schemes fail

%% ***** Separate memory *****
xkm1 = pf.particles(:,k-1); % extract particles from last iteration;
xk = zeros(size(xkm1)); % = zeros(nx,Ns);
wk = zeros(size(wkm1)); % = zeros(Ns,1);
%
%% ***** Algorithm *****
for i = 1:Ns

```

```

% xk(:,i) = sample_vector_from q_xk_given_xkm1_yk given xkm1(:,i) and yk
% xk(:,i)=pf.p_xk_given_xkm1(k,xk(:,i),xkm1(:,i));
% Using the PRIOR PDF: pf.p_xk_given_xkm1"funct (1)" not defined.
xk(:,i) = sys(k, xkm1(:,i), pf.gen_sys_noise()); % calcul xk(i)a postriori
% ----- = p(xk/xk-1).
%
% wk(i) = wkm1(i) * p.p_yk_given_xk(yk,
xk(:,i))*p_xk_given_xkm1(k,xk(:,i),xkm1(:,i))/q_xk_given_xkm1_yk(xk(:,i), xkm1(:,i), yk);
%
% weights (when using the PRIOR pdf)
wk(i) = wkm1(i) * pf.p_yk_given_xk(k, yk, xk(:,i));
%
% weights (when using the OPTIMAL pdf)
% wk(i) = wkm1(i) * p_yk_given_xkm1(yk, xkm1(:,i)); % (1) we do not know this PDF
end;
%
%% ***** Normalize weight vector *****
wk = wk./sum(wk);
%
%% ***** Calculate effective sample size*****
Neff = 1/sum(wk.^2);
%
% (p102)
%% ***** Resampling *****
% remove this condition and sample on each iteration:
% [xk, wk] = resample(xk, wk, resampling_strategy);
%if you want to implement the bootstrap particle filter
resample_percentaje = 0.50;
Nt = resample_percentaje*Ns;
if Neff < Nt
    disp('Resampling ...')
    [xk, wk] = resample(xk, wk, resampling_strategy);
    % {xk, wk} is an approximate discrete representation of p(x_k | y_{1:k})
end

```

```

%
%% ***** Compute estimated state *****
xhk = zeros(nx,1);
for i = 1:Ns
    xhk = xhk + wk(i)*xk(:,i); % (somme des xk )/200
end
%
%% ***** Store new weights and particles *****
pf.w(:,k) = wk;
pf.particles(:,k) = xk;
%
return; % bye, bye!!!
%
%% ***** Resampling function *****
function [xk, wk, idx] = resample(xk, wk, resampling_strategy)
Ns = length(wk); % Ns = number of particles
%
% wk = wk./sum(wk); % normalize weight vector (already done)
%
switch resampling_strategy
    case 'multinomial_resampling' %called from (*)P.2 of Princ./Prog.
        with_replacement = true;
        idx = randsample(1:Ns, Ns, with_replacement, wk); % Ns nombre of Particle
% générer un vecteur ligne "indx" avec un nombre d'élément: Ns (=200)
%         chaque elemnty prend une valeur (1, Ns)
%         avec repeton: with_replacement = true
% 1 element de "wk" représente la probabilité d'apparition de la valeur de son index " k"
%{
    THIS IS EQUIVALENT TO:
    edges = min([0 cumsum(wk)],1); % protect against accumulated round-off
    edges(end) = 1;          % get the upper edge exact
    % this works like the inverse of the empirical distribution and returns
    % the interval where the sample is to be found
    [~, idx] = histc(sort(rand(Ns,1)), edges);

```

```

%}

case 'systematic_resampling'

    % this is performing latin hypercube sampling on wk
    edges = min([0 cumsum(wk)],1); % protect against accumulated round-off
    edges(end) = 1;          % get the upper edge exact
    u1 = rand/Ns;
    % this works like the inverse of the empirical distribution and returns
    % the interval where the sample is to be found
    %[~,idx] = histc(u1:1/Ns:1, edges);

% case 'regularized_pf'    TO BE IMPLEMENTED
% case 'stratified_sampling' TO BE IMPLEMENTED
% case 'residual_sampling' TO BE IMPLEMENTED

otherwise

    error('Resampling strategy not implemented')
end

%

%xk = xk(:,idx);          % extract new particles

%mwk = repmat(1/Ns, 1, Ns); % now all particles have the same weight

%

return;

```

